UNIVERSITATEA DE VEST DIN TIMIŞOARA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
PROGRAMUL DE STUDII DE LICENŢĂ: INFORMATICĂ

LUCRARE DE LICENŢĂ

COORDONATOR:
Lect. Dr. Adriana Tanasie

ABSOLVENT:
Bosna Marinel Lilian

TIMIŞOARA
2020

UNIVERSITATEA DE VEST DIN TIMIŞOARA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
PROGRAMUL DE STUDII DE LICENŢĂ INFORMATĂ

LUCRARE DE LICENŢĂ

COORDONATOR:
Lect. Dr. Adriana Tanasie

ABSOLVENT:
Bosna Marinel Lilian

TIMIŞOARA
2022

Abstract

E-library system makes the work of a person who is in charge of the library more convenient to search, arrange and make an inventory of the contents of the library. In this paper, a search system for an E-Library for the students and users who want to find an fast and good way to read a book. This system has some methods of searches and it is convenient for the users to find their required information. The data base contains Books, users and reviews . The administrator of the E-Library can add, update, and delete any information in the database easily

Cuprins

0.1 Rezumat

Prin COVID-19, devine vizibil ce înseamnă cu adevărat a fi digital. Nu este vorba doar de aplicații interesante, ci de a avea un lanț de soluții care traversează procese, oameni și tehnologie. Este o schimbare fundamentală în modul în care lucrăm, trăim și facem afaceri. O astfel de resursa limitata sunt librariile fizice care urmeaza un parcurs traditional, un astfel de eveniment major precum pandemia de COVID-19 a afectat bibliotecile de toate tipurile din intreaga lume . O solutionare al acestei probleme care a urmat tiparul digitalizarii este aplicatia Elib . Elib este o aplicatie care ofera utilizatorilor posibilitatea de a avea access la orice carte in format electronic printr-un sistem de cautare, sortare si filtrare astfel facilitand timpul petrecut intr-o librarie traditionala, resursele limitate, riscul deteriorarii unei carti s.a.m.d. .Aceasta aplicatie pune la dispozitie utilizatorilor doar rolul de user normal, ceea ce inseamna ca un utilizator normal poate doar sa downloadeze o carte fara sa faciliteze de operatii care pot crea,sterge sau modifica carti, lucru care duce la un management mai sigur al bibliotecii electronice .In realizarea acestei applicatii web sunt fololosite urmatoarele tehnologii: limbajul Java prin care s-a dezvoltat partea de back-end, baza de date relationlaa PostgreSQL iar front-end-ul este realizat prin cadrul frameworkului Angular.

Capitolul 1

Aspecte generale

1.1 Motivatie

Scopul acestei lucrări este de a revizui digitalul rolul bibliotecilor în sprijinirea e-learning-ului. Biblioteci digitale oferiți resurse și servicii informaționale bazate pe tehnologie permite cursanților să acceseze cunoștințele relevante oriunde oricând. Ca atare, bibliotecile digitale sunt inseparabile de procesul de învățare pentru că au făcut posibilă e-learning. Deși foarte relevant pentru e-learning, bibliotecile digitale nu au fost folosite în educație la capacitatea lor maximă datorită la dezvoltarea lor inegală la nivel mondial, juridic și tehnic bariere și faptul că sectorul industrial (comercial editori) gestionează în continuare accesul la conținutul digital utilizat în educație pe bază comercială (făcându-l inaccesibil pentru public larg fără a plăti o taxă). Prezenta si lucrarea rezultate din cercetarea bibliotecilor digitale și a acestora capacități de includere în educație . În plus, o bibliotecă electronică ar fi o modalitate excelentă pentru utilizatorii noștri de a o folosi scopuri diferite în funcție de nevoile fiecăruia, de exemplu pentru elevi pot fi a mare aplicație pentru cercetare bazată pe munca de clasă, o aplicație în care utilizatorul obișnuit pot găsi răspunsuri, lecturi, lucruri noi și așa mai departe.

Prin urmare, biblioteca virtuală devine o nouă resursă didactică care captivează şi motivează elevii/studenţii la studiu .

1.2 Descrierea problemei

Bibliotecile publice, un refugiu sigur și o resursă valoroasă pentru studenți și familii, nu au fost imune la efectele severe ale pandemiei de COVID-19. După declanșarea pandemiei în martie 2020, bibliotecile din Statele Unite au închis operațiunile în persoană. Aceste opriri au afectat negativ studenții și familiile care se bazează pe biblioteci ca locuri de studiu, acces la internet și socializare. Un sondaj realizat în martie 2020 de Asociația Americană de Biblioteci a constatat că 99dintre bibliotecile publice care au răspuns au fost închise din cauza pandemiei.

Bibliotecile au fost afectate în principal din cauza decalajului digital, a lipsei de politici, a problemelor legate de digitizare, în special a drepturilor de autor și a lipsei de personal calificat. —— Acest lucru a condus la imposibilitatea persoanelor de a avea access la resurse din libariile publice si au fost nevoiti sa gaseasca o alta modalitate de a-si obtine informatiile dorite . Asadar, aplicatia web creata care vine ca o

solutionare a acestei probleme este Elib. Ea are ca scop usurarea procesului urmat de diferiti utilizatori de a obtine diferite resurse in format digital. In aceasta aplicatie utilizatorii pot sa isi downloadeze resursele in format digital astfel avand access la orice tip de resursa indiferent de contextul in care ne aflam, context cum ar fi cel pandemic. In aplicatia Elib resursele puse la dispozitia utilizatorilor sunt adaugate de catre un administrator, persoana care poate avea controlul cartilor digitale, ea avand doar 2 tipuri de utilizatori, utilizatorii normali care pot folosi aplicatia pentru a vedea informatii despre resurse digitale sau pentru a le descarca si administratorii care sunt persoane autorizate, ei avand control asupra librariei electronice.

Realizarea acestei aplicat, ii web a avut la baza trei mari tehnologii: limbajul de programre Java, framework-ul Spring Boot, Angular si baza de date relationala PostgreSql.

Limbajul de programare Java, a fost folosit pe partea de back-end, pentru a creea API-uri, a face conexiunea cu baza de date si a oferi servicii pe parte clientului.

Cu ajutorul framework-ului Angular, a fost realizat a partea de front-end. Prin utili- zarea acestui framework, aceast a aplicat, ie este de tipul single page application, as,adar

va oferi o experient a mai bun a ˆım utilizare, asem an atoare ca fluiditate aplicat, iilor desktop. Baza de date a acestei aplicat, ii a fost creat a prin PostgreSql. Alegerea folosirii acestui tip de baze de date a constant ˆın faptul c a este una dintre cele mai populare din momentul actual s, i ofer a o multitudine de avantaje cum ar fi: suport pentru un num ar mare de limbaje de programare, suport pentru un num ar ridicat de tipuri de date s, i ofer a caracteristicile unei baze de date orientate pe obiecte.

1.3 Soluție propusa

Cu toate acestea, potrivit Asif și Singh (2020), bibliotecile tradiționale au fost transformate în biblioteci inteligente ca urmare a progreselor și dezvoltărilor tehnice. În scenariul pandemic de astăzi, bibliotecile au o mare varietate de resurse electronice, biblioteci digitale, servicii electronice și așa mai departe. Bibliotecile inteligente ale viitorului ar putea folosi cunoștințele și expertiza, precum și să distribuie informații în formate electronice, pentru a satisface nevoile de informații ale utilizatorilor moderni de biblioteci.

Bibliotecile digitale sunt colecții multilingve, multimedia, bogate în conținut documente care sunt distribuite și accesate în întreaga lume. Pentru solutionarea acestor problemel aplicația Elib își propune să ofere tuturor utilizatorilor access la resurse digitale de pe orice device cu acces la internet printr-o aplicație web de tip e-Library, lucru care va facilita economisirea timpului pierdut in librariile publice, accesul din orice colt al lumii indiferent de ora si de asemenea acces in orice context cum ar fi cel pandemic. Elib va fi o componentă importantă care va expune tuturor utilizatorii noștri nu numai la o perspectivă globală, ci și la accesul la cunoașterea lumii are de oferit .Elib va oferi utilizatorilor noștri acces la o multitudine de surse deschise care consta in zeci de mii de cărți digitale din întreaga lume.

1.4 Aplicaţii si soluţii existente

Cele mai populare 5 aplicații de tip e-Library sunt : Wifey online library, Maastricht university library, Open library, Online library, Z-library . În continuare o sa prezint cele mai importante plusuri si minusuri ale fiecărei aplicaţii .

1.4.1 Wifey online library

Wifey online library dispune atât de o aplicație web cât și de o aplicație mobilă care a fost descărcată de peste 1000 de persoane . Atât aplicația mobilă cât și cea web dispun de un sistem de securitate care pune la dispoziția utilizatorului atât un sistem de înregistrare cât și unul de autentificare, acest sistem oferă utilizatorului posibilitatea de a se autentifica cu o adresa instituțională, un lucru foarte benefic pentru utilizatori precum sunt studenții . Un alt plus al acestei aplicații este acela că putem avea o lista de favorite însă un minus destul de mare la o aplicație de acest tip este faptul că nu dispunem de un sistem de feedback sau review al unei cărți . De asemenea un alt plus foarte mare este sistemul de e-mail care ne pune la dispoziție posibilitatea de a fi informați în legătură cu noi cărți adăugate în sistem, actualizări legate de aplicație dar și de eventuale știri din acest domeniu al lecturii . Probabil cel mai important aspect când vine vorba de acest tip de aplicație de management este sistemul de căutare avansată pentru a putea găsi cartea dorită indiferent de criteriu de căutare cum ar fi: autorul, anul publicației, editura și așa mai departe . Această aplicație dispune de o căutare avansată a unei cărți foarte bine pusă la punct, deoarece putem găsi resursele căutate după: cuvinte cheie, articole, publicații și așa mai departe, lucru care ne oferă un răspuns foarte bun al căutării noastre chiar dacă informațiile pe care le dețîn sunt puține, de asemenea aplicația dispune de un volum foarte mare de resurse, peste 22.000 de cărți, peste 1.600 de jurnale si așa mai departe . În concluzie aplicația este una foarte complexă, foarte bine pusă la punct care poate fi utilizată cu ușurință de absolut oricine, singurele minusuri ar fi lipsa unui sistem de feedback si aplicația mobilă care este utilizată de foarte puțini utilizatori in comparație cu aplicația web.

1.4.2 Maastricht University

Maastricht University library dispune doar de o aplicație web care nu dispune o aplicație mobilă. Scopul acestei aplicații este să ofere studenților de la universitatea din Maastricht o metodă digitală și mai modernă de a avea acces la resurse, un foarte mare plus la acest tip de aplicații este faptul că aplicația dispune de un sistem prin care studenți de la această universitate pot împrumută o carte de la facultate direct din aplicație, evitând timpul petrecut în biblioteca, ei pot face o cerere de împrumut al unei cărți ,iar când ajung la biblioteca universitară au deja cartea pregătită în cazul în care stundeții își doresc cartea respectivă în format fizic. Fiind o aplicație a unei biblioteci universitare nu ne putem înregistra sau loga, acces având doar stundeții acestei facultăți pe baza e-mailului instituțional, ceea ce aduce anumite minusuri pentru un utilizator care nu face parte din această universitate deoarece nu o să primească nici un e-mail legat de anumite informații de care are nevoie, cum ar fi informații despre o carte pe care o caută, informații legate de utilizarea aplicației sau despre disponibilitatea unei resurse, iar automat nu dispune nici de un sistem de feedback sau review de la alți utilizatori, lucru care ne-ar putea pune probleme în alegerea unei resurse.

Dacă singurul nostru scop în vederea utilizării acestei aplicații este să găsim o anumită resursă, aplicația este foarte utilă deoarece are un sistem de căutare avansată atât pentru cărți cât și pentru jurnale foarte bine pus la punct, putem găsi orice resursă dorim după foarte multe criterii, chiar dacă știm și cel mai mic detaliu precum seria unei cărți, limba unei cărți, dată publicării și așa mai departe, putem găsii absolut orice detaliu pe care să îl filtram ,iar de asemenea putem folosi chiar și operatori în momentul căutării precum sau,și, etc. în cazul în care nu suntem siguri de o informație . În concluzie aplicația are un sistem de management perfect prin care este imposibil să nu găsim o carte despre care știm chiar și cele mai puține detalii, din păcate aplicația nu dispune de o aplicație mobilă si are foarte multe dezavantaje care o limitează în comparație cu alte aplicații în cazul în care utilizatorul acestei aplicații nu este de la Maastricht University.

1.4.3 Open Library

Open library dispune atât de o aplicație web cât și de o aplicație mobilă, din start avem un punct forțe al acestei aplicații și mai exact aplicația mobilă deoarece ea are peste 50.000 de utilizatori, fiind și cea mai populară aplicație mobilă dintre toate aplicațiile enumerate. Open library are un sistem de înregistrare și de autentificare care poate fi folosit de absolut orice utilizator, procesul prin care cineva se poate înregistra și autentifica pe această platforma este foarte simplu și rapid.Mulțumită sistemului de autentificare aplicația ne pune la dispoziție și un sistem de newsletter în care suntem înscriși automat în momentul înregistrării unui cont pe platforma lor, așadar o să putem primi informații pe adresa de e-mail legate de orice resursă nou adăugată, informații despre anumite cărți sau funcționalități ale aplicației și așa mai departe .Din păcate la fel că și în cazul celorlalte aplicații de până acum lipsește sistemul de feedback sau al realizării unei liste de favorite, acest lucru îngreunează găsirea resurselor potrivite utilizatorului deoarece această aplicație mai are un minus în comparație cu celalalte două aplicații de până acum, minusul pe care îl are este căutarea avansată . Open library dispune de un sistem de căutare avansată dar din păcate nu este foarte bine pus la punct, iar dacă nu cunoști câteva informații despre o anumită carte va fi destul de greu să găsești cartea respectivă deoarece aplicația nu dispune de funcționalități de căutare precum: căutare cu operatori logici, căutare după cuvinte cheie, căutare după resurse asemănătoare, căutare după intervale, de exemplu căutăm o carte care a fost publicată între anii 1963 - 1966 și așa mai departe . În concluzie aplicația Open library este una bună în cazul în care căutăm o aplicație cu o interfață ușoară, dorim să o folosim din orice ipostază de user, indiferent că deținem o adresa instituțională sau nu și cel mai important dacă cunoaștem detaliile exacte despre o anumită carte pe care dorim să o găsim.

1.4.4 Online Library

Online library este un serviciu dedicat studenților din comunitatea de învățământ de la distanță a Universității din Londra . Ea oferă resurse online, sprijin profesional și indrumare tuturor studenților din această universitate indiferent de momentul sau facultatea la care au ales să studieze. Online library dispune de o aplicație web dar nu și de una mobilă.Sistemul de autentificare este special pentru cei care dețin o adresa insituțională pentru Universitatea din Londra, această constrângere blochează

folosirea acestei aplicații la nivel optim de către utilizatorii care nu fac parte din această universitate deoarece înregistrarea se poate face doar pe baza adresei instituționale. În cazul în care suntem un utilizator obișnuit această aplicație nu este foarte optimă pentru noi deoarece nu deține resurse pentru cineva care caută informații din afară programei universitare deoarece aplicația deține strict informații de care studenții au nevoie pentru suport de curs, acest lucru este un advantaj foarte mare pentru utilizatorii care studiază în această universitate având suport la absolut orice materie din orice facultate, resursele sunt foarte bine structurate de la tipul facultății, anul universitar până la materie, în schimb pentru cineva care nu are nevoie de suport la o anumită materie din această universitate aplicația nu este una utilă.Un alt plus pe care aplicația îl oferă studenților este acela că ei pot downloada e-bookuri, jurnale și orice tip de resursă în orice tip de format ceea ce facilitează folosirea aplicației într-un mod cât mai ușor și interactiv conform bunului plac. Deoarece aplicația a fost concepută pentru studenții din această facultate, concluzia este că pentru un utilizator care nu face parte din această instituția aplicația nu este folositoare deoarece este mult prea restricționată în comparație cu cea de la Maastricht University

1.4.5 Z-Library

Z-Library este una dintre cele mai mari biblioteci online din lume, care conține peste 9.945.695 de cărți și 84.837.000 de articole. Ne propunem să facem literatura accesibilă tuturor.Z library dispune de o aplicație web cât și de una mobilă. Această aplicație poate fi folosită de absolut orice utilizator din orice colț al lumii . Ea dispune de un sistem de autentificare și de înregistrare foarte rapid și ușor de utilizat. Unul dintre punctele forțe ale acestei aplicații este sistemul prin care putem alege o anumită carte deoarece aplicația dispune de un sistem de feedback al unei cărți, feedback care poate fi oferit de absolut orice utilizator, un sistem de rating al fiecărei cărți care poate contribui la alegerea mai bună a unei cărți, lucru care facilitează și la durata pe care utilizatorul o petrece în căutarea unei anumite resurse deoarece cărțile sunt prezentate în ordinea ratingului fiind afișate în ordine descrescătoare. Pe lângă aceste funcționalități cu ajutorul cărora Z library începe să se diferențieze de alte aplicații de acest tip mai este și lista de favorite în care utilizatorul poate adaugă cărți pe care dorește să le aibă la îndemână din diferite motive. De asemenea aplicația ne facilitează și utilizarea ei indiferent de dispozitivul pe care îl folosim și ne oferă posibilitatea de a avea informații de îndemână în situații mai nefavorabile cum ar fi să nu dispunem de internet deoarece aplicația are un sistem prin care putem descarcă cărțile direct în format pdf, txt, rdf și așa mai departe în funcție de posibilitatea sau preferință noastră. Z library ne oferă și o alternativă pentru cititorii moderni prin care putem vizualiza conținutul unei cărți direct de pe un Kindle. Din moment ce ne logam pe aplicația lor putem interacționa cu diferiți utilizatori prin funcționalitatea de trimitere a unei cărți către alt utilizator, expunerea informațiilor dintr-o carte, păreri despre acea carte și așa mai departe, acest lucru îmbunătățește semnificativ experiență unui utilizator deoarece își poate formă un "cerc" cu alți utilizatori pasionați de aceași arie. Z library pare a fi cea mai complexă și completă aplicație dintre cele enumerate, după părerea mea are toate funcționalitățile de care ar trebui să dispună o astfel de aplicație, cel mai important aspect fiind chiar partea de căutare avansată, chiar dacă nu dispune de o căutări cu operatori precum și, sau etc., putem găsi atât articole cât și cărți chiar dacă știm foarte puține detalii cum ar fi: o secvență dintr-o carte sau un cuvânt cheie

cât de mic . Un mic dezavantaj ar fi faptul că resursele nu sunt grupate pe domenii, astfel încât dacă dorim să vizualizam o lista de cărți dintr-un anumit domeniu nu o putem face, trebuie să știm exact ce carte căutăm, de asemenea pagină web cred că ar putea fi "cosmetizată" pentru a oferi utilizatorului o experiență mai plăcută în utilizarea ei . În concluzie, cu excepția celor două mici dezavantaje aplicația este una excelență pentru un utilizator care caută o anumită carte, într-un anumit format dar spre exemplu pentru o persoană care are nevoie de suport pentru a performa într-o anumită arie si nu știe exact ce carte caută aplicația ar putea pune câteva probleme utilizatorului .

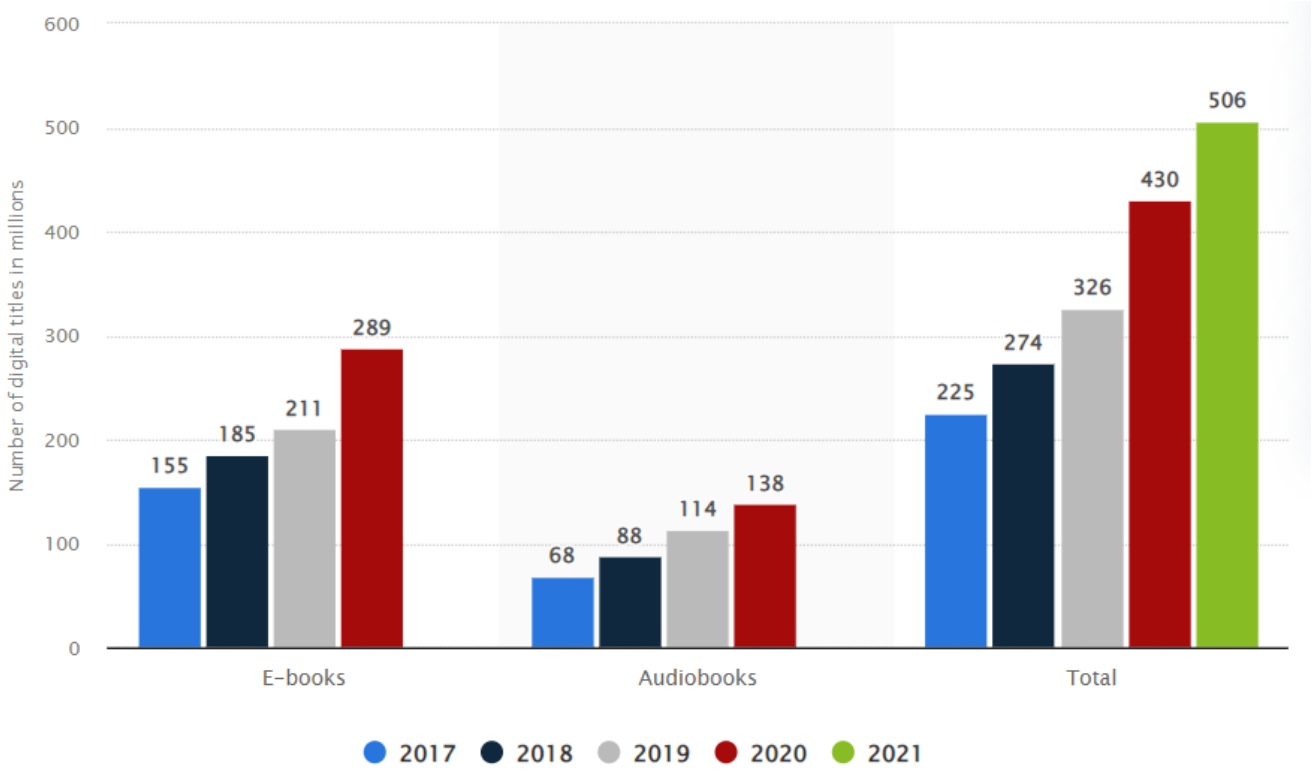
Capitolul 2

Prezentarea problematicii abordate

2.1 Importanta libariilor electronice

Ca porți către cunoaștere și cultură, bibliotecile joacă un rol fundamental în societate. Resursele și serviciile pe care le oferă creează oportunități de învățare, sprijină alfabetizarea și educația și ajută la modelarea noilor idei și perspective care sunt esențiale pentru o societate creativă și inovatoare.

Conform Institutului Nat, ional de Statistica, numărul de cărți digitale împrumutate de la biblioteci și școli a atins 506 milioane în 2020, în creștere cu 16Deși sursa nu a furnizat detalii cu privire la numărul de cărți electronice comparativ cu cărțile audio vândute în acel an, cărțile electronice au fost istoric mai populare printre împrumuta cții de cărți electronice comparativ cu cărțile audio vândute în acel an, cărțile electronice au fost istoric mai populare printre împrumuta cții de cărți digitale, cu pu pu țin sub 290 de miliobé de împru bimut și școli incluse în studiază în 2020.

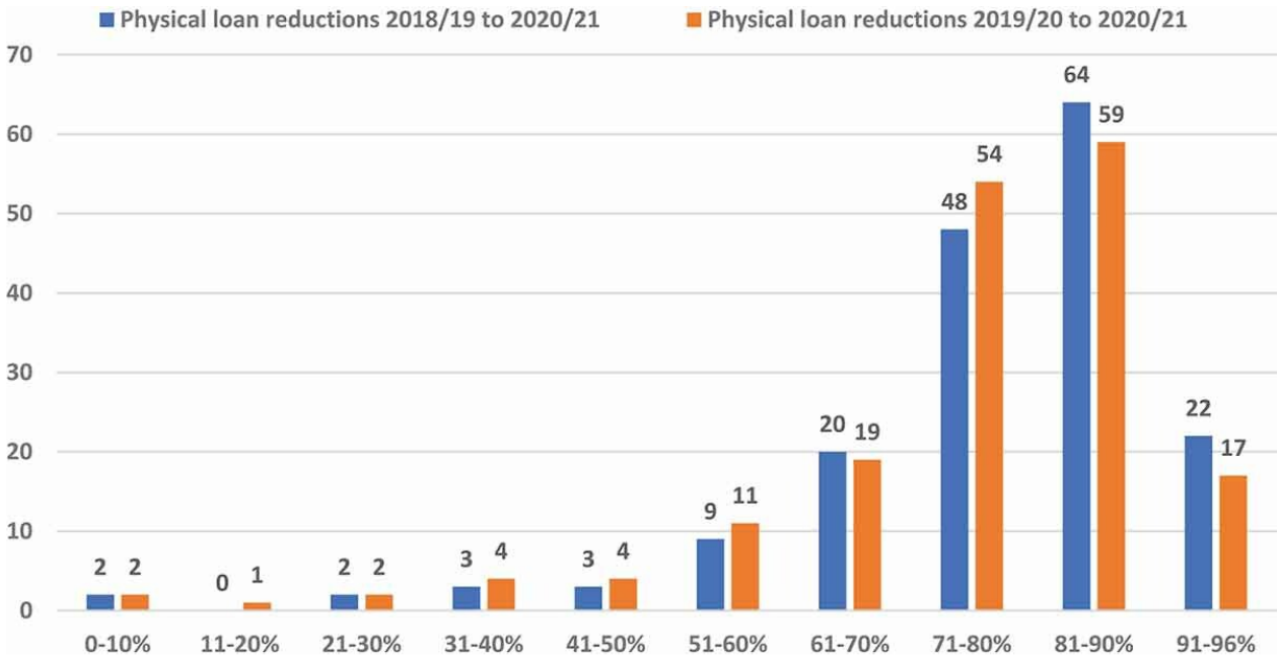


2.2 Impactul pandemic asupra librariilor publice

Pandemia COVID-19 a avut inevitabil un efect semnificativ asupra serviciilor bibliotecii publice de pe tot globul. Închiderea forțată a clădirilor bibliotecilor în mare parte din 2020 și 2021 a adus o nouă normalitate, forțând serviciile bibliotecilor să încurajeze utilizarea extinsă a altor servicii, cum ar fi împrumuturile electronice, pentru a răspunde nevoilor comunității. Această lucrare explorează acest impact, înainte de a examina mai în detaliu modul în care sa manifestat în serviciile bibliotecilor publice din intreaga lume.

Solicitări privind libertatea de informare au fost adresate tuturor serviciilor de biblioteci publice globale. Datele primite au indicat că aproape 65din serviciile bibliotecii globale au înregistrat o reducere a împrumuturilor fizice între 70și 90dintre nivelul de împrumut arătat, de asemenea, că aproape 120 de milioane de cărți care au fost emise în anii pre-pandemiei nu au fost emise în perioada de izolare 2020/21. Între timp, 47dintre serviciile de bibliotecă și-mut văzut furnizarea electronice a crescut între 100și 200la nivelurile pre-pandemie, deși aceste cifre au crescut de la o bază scăzută și au fost relativ mici atunci când sunt măsurate în raport cu pierderile fizice din împrumuturi de ase au evidențtcă membrii activi ai serviciilor bibliotecii publice (membrii care au împrumutat un articol în anul precedent) a scăzut la 40din nivelurile pre-pandemie din întreaga lume. de bibliotecă erau în mare parte indisponibile, această creștere a fost de la o bază foarte scăzută, iar această creștere a utilizării digitale nu a fost aproape de a atenua scăderea împrumuturilor fizice care a avut loc în întreaga lume. . Descoperirile lucrării sugerează că, chiar și atunci când publicul nu a avut de ales decât să treacă la digital, a făcut acest lucru în număr limitat în comparație cu utilizarea clădirilor și colecțiilor fizice ale bibliotecilor.

Mai jos putem vedea reducerea imprumuturilor cartilor in format fizic - 2018/2019 si 2019/2020 in comparatie cu 2020/2021



2.3 Care sunt advantajele unei aplicații de tip e-Library?

Scăderea vizitelor la bibliotecile convenționale sugerează că clienții din zilele noastre doresc să acceseze informații și să citească conținut fără a vizita o bibliotecă în persoană. Multe biblioteci mari și universități au început deja procesul de digitalizare pentru a face materialele accesibile membrilor și publicului larg . Corporațiile, de diferite dimensiuni, au început și ele să analizeze avantajele bibliotecii digitale și au început să adopte biblioteci digitale pentru angajații lor.

Acesta este motivul pentru care; digitizarea este una dintre cele mai populare tendințe în managementul bibliotecilor. Multe instituții de învățământ, în aceste zile, au investit în biblioteci digitale pentru a le permite studenților să acceseze cărți de top, de asemenea, recalifică și îmbunătățesc angajații, permițându-le acces la bibliotecile digitale. Pe lângă digitizarea serviciilor convenționale, bibliotecile digitale vă ajută organizația să vă ofere servicii îmbunătățite și inovatoare. Să înțelegem cele 10 avantaje majore ale bibliotecii digitale.

1. Permiteți cititorilor să acceseze materiale la cerere
2. Cititorii pot să găsească resurse instantaneu
3. Fără program de deschidere sau de închidere
4. Eliminați deteriorarea resurselor
5. Căutare simplificată
6. O gamă largă de conținut disponibil
7. Actualizat cu ușurință
8. Automatizarea managementului bibliotecilor
9. Preventie și Conservare a datelor
10. Costuri reduse

Concluzie: Nicio organizație nu poate rămâne relevantă într-o economie a cunoașterii fără a facilita accesul la bibliotecă digitală. Sistemele de management al învățării de nouă generație (LMS) facilitează accesul la cerere a resurselor digitale mari atât pentru întreprinderi, cât și pentru instituțiile de învățământ. Organizația dvs. poate investi cu ușurință în sistemul LMS potrivit pentru a profita de aceste avantaje ale bibliotecii digitale

Capitolul 3

Tehnologii folosite

3.1 Spring Boot

Pentru dezvoltarea părţii de back-end s-a folosit frameworkul Spring Boot scris în limbajul de programare Java.

Ce este Spring Boot?

Definiţia 3.1.1. Java Spring Framework (Spring Framework) este un cadru popular, cu sursă deschisă, la nivel de întreprindere, pentru crearea de aplicații autonome, la nivel de producție, care rulează pe Java Virtual Machine (JVM).

Java Spring Boot (Spring Boot) este un instrument care face dezvoltarea aplicațiilor web și a microserviciilor cu Spring Framework mai rapidă și mai ușoară prin intermediul a trei capabilități principale:autoconfigurare,o abordare cu opinie a configurației, abilitatea de a crea aplicații de sine stătătoare.Aceste caracteristici funcționează împreună pentru a vă oferi un instrument care vă permite să configurați o aplicație bazată pe Spring cu o configurație și o configurare minimă.

De ce am ales sa folosesc Spring Boot?

Puteți alege Spring Boot datorită caracteristicilor și beneficiilor pe care le oferă, așa cum sunt prezentate aici:

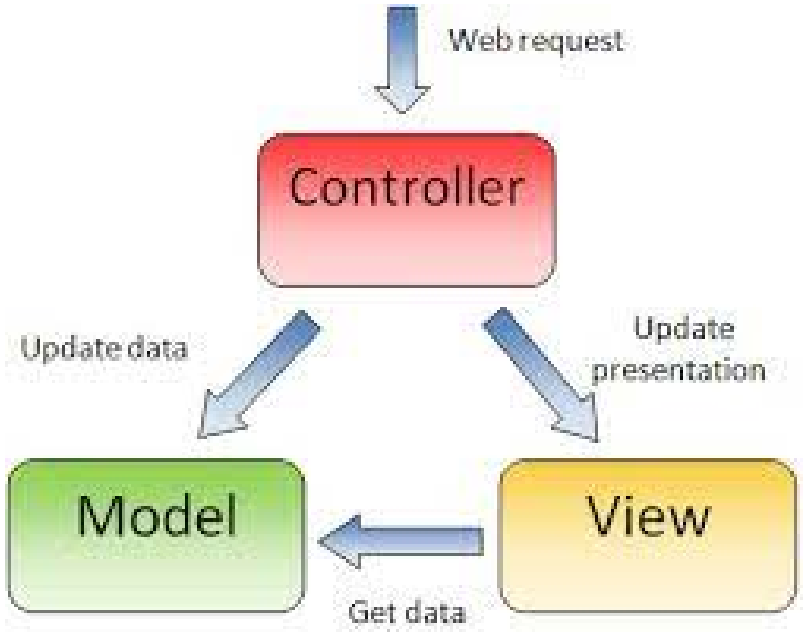
- Oferă o modalitate flexibilă de a configura Java Beans, configurații XML și tranzacții cu baze de date.
- Oferă o procesare puternică în lot și gestionează punctele finale REST.
- În Spring Boot, totul este configurat automat; nu sunt necesare configurații manuale.
- Oferă aplicație de Spring bazată pe adnotări
- Ușurează gestionarea dependenței
- Include Container Servlet încorporat

Alte cateva avantaje majore ar fi: suportul HTTP care ajuta controllerul sa gestioneze cererile pentru toate metodele de HTTP (GET,POST,PUT,DELETE,UPDATE), adnotariile de rest sunt un alt factor important pentru care am ales acest framework deoarece putem folosi o adnotare pentru intreaga clasa fara sa fie nevoie sa folosim

cate o adnotare pentru fiecare metoda in parte ceea ce duce la un cod mult mai curat si bine structurat, iar un ultim aspect pe care l-am luat in considerare este suportul pentru diferite tipuri de format JSON, XML, si HTML, deoarece un alt aspect cheie al serviciilor web RESTful este reprezentarea, ceea ce înseamnă că aceeași resursă poate fi reprezentată în diferite formate, cum ar fi JSON, XML, HTML etc.

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications.

The four components of Spring MVC are Model, View, Controller, and Front Controller (DispatcherServlet class). The four layers of Spring Boot are the Presentation layer, Business layer, Persistence layer, and Database layer. Spring MVC is only designed to develop dynamic web pages and RESTful web services.



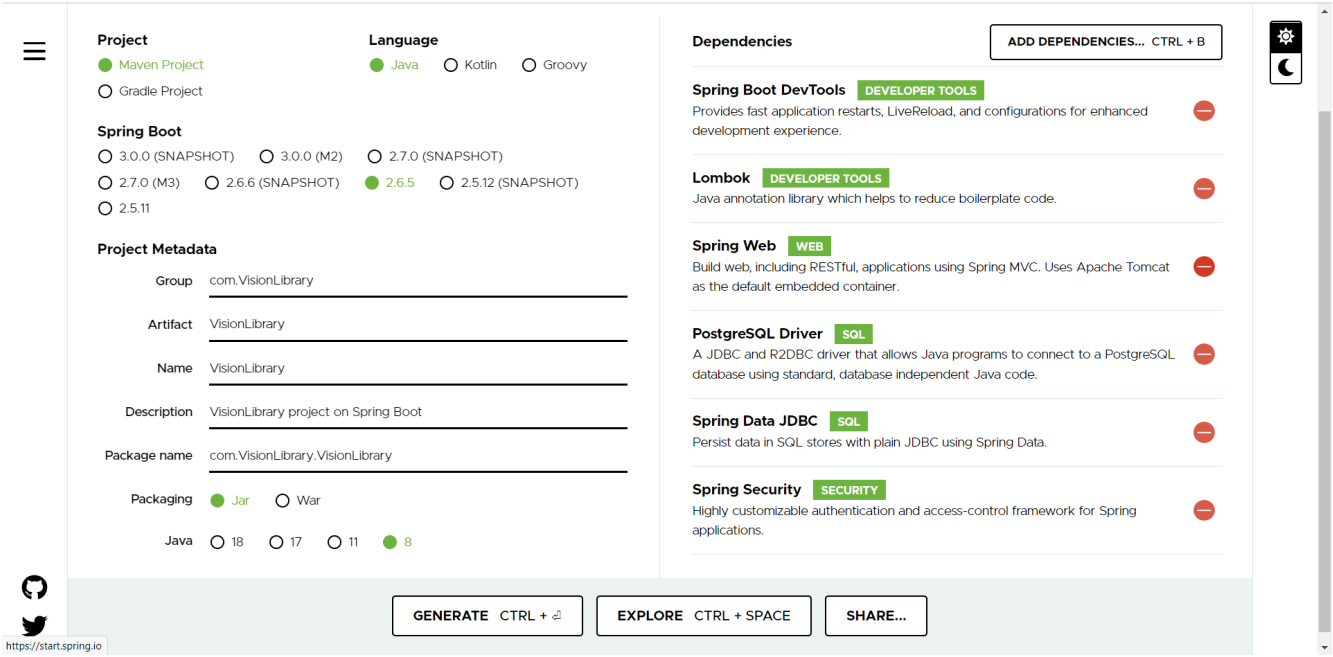
Arhitectura pe care am urmat-o si in realizarea aplicatiei Elib.

3.2 Spring Initializr

Ce este Spring Initializr?

Definiţia 3.2.1. Uneori, cea mai grea parte a oricărui proiect este începerea. Spring Initializr este în cele din urmă o aplicație web care poate genera un proiect Spring Boot structura pentru tine. Nu generează niciun cod de aplicație, dar vă va oferi un proiect de bază structură și fie o specificație de construcție Maven, fie Gradle cu care să vă construiți codul. Tot ce trebuie să faceți este să scrieți codul aplicației.

În imaginea de mai jos se poate observă cum am configurat proiectul pentru Vision Library la care am adaugat cateva dependinte necesare pentru inceperea proiectului sau versiunea de Java.



3.3 Spring Data JPA

Ce este Spring Data JPA?

Definiţia 3.3.1. Spring Data JPA, parte a familiei mai mari Spring Data, facilitează implementarea cu ușurință a depozitelor bazate pe JPA. Acest modul se ocupă de suport îmbunătățit pentru straturile de acces la date bazate pe JPA. Facilitează construirea de aplicații Spring-powered care utilizează tehnologii de acces la date.

Implementarea unui strat de acces la date al unei aplicații a fost greoaie de ceva vreme. Trebuie scris prea mult cod standard pentru a executa interogări simple, precum și pentru a efectua paginarea și auditarea. Spring Data JPA își propune să îmbunătățească în mod semnificativ implementarea straturilor de acces la date prin reducerea efortului la cantitatea necesară. În calitate de dezvoltator, vă scrieți interfețele de depozit, inclusiv metode de căutare personalizate, iar Spring va furniza automat implementarea. JpaRepository este o extensie specifică JPA (Java Persistence API) a Repository. Conține API-ul complet al CrudRepository și PagingAndSortingRepository. Deci conține API pentru operațiuni CRUD de bază și, de asemenea, API pentru paginare și sortare.

Modelul de depozit este unul dintre cele mai populare modele legate de persistență. Acesta ascunde detaliile specifice de implementare a depozitului de date și vă permite să implementați codul dvs. de afaceri la un nivel de abstractizare mai ridicat.

Implementarea acelui model nu este prea complicată, dar scrierea operațiilor standard CRUD pentru fiecare entitate creează o mulțime de cod repetitiv. Spring Data JPA vă oferă un set de interfețe de depozit pe care trebuie doar să le extindeți pentru a defini un anumit depozit pentru una dintre entitățile dvs. . De exemplu in cazul unei aplicatii de management cum este cea de fata pe care dorim sa o implementam avem nevoie de numeroase metode de sortare/filtrare pe care dorim sa le implementam JPA-ul ne ajuta foarte mult sa interpretam diferite query-uri pentru ajustarea logicii de afaceri pe care dorim sa o folosim in crearea metodelor noastre de rest .

Este important de știut că majoritatea caracteristicilor, cum ar fi maparea relațională obiect și capabilitățile de interogare, sunt definite și furnizate de specificația JPA și de implementările sale. Aceasta înseamnă că puteți utiliza toate caracteristicile implementării dvs. JPA preferate. Spring Data JPA simplifică utilizarea acestora.

3.4 Spring Data REST

Ce este Spring Data REST?

Definiţia 3.4.1. Spring Data REST face parte din proiectul umbrelă Spring Data și facilitează construirea de servicii web REST bazate pe hipermedia peste depozitele Spring Data.

Spring Data REST se bazează pe depozitele Spring Data, analizează modelul de domeniu al aplicației dvs. și expune resursele HTTP bazate pe hipermedia pentru agregatele conținute în model. O instanță de Resource este o reprezentare a obiectului Java a artefactului care este oferită ca răspuns la o solicitare de resursă din partea clientului .

De ce am ales sa folosesc REST webservices?

Unul dintre avantajele cheie ale API-urilor REST este că oferă o mare flexibilitate. Datele nu sunt legate de resurse sau metode, așa că REST poate gestiona mai multe tipuri de apeluri, poate returna diferite formate de date și chiar poate modifica structural cu implementarea corectă a hypermedia.

Cum am testat acest API de rest?

Pentru testarea implement arii de API-urilor s-a folosit programul Postman. Postman este un instrument de dezvoltare API (interfat, a de programare a aplicat, iilor) care ajut a la construirea, testarea s, i modificarea API-urilor. Aproape orice funct, ionalitate care ar putea fi necesar a de c atre orice dezvoltator este ˆıncapsulat a ˆın acest instrument. Este folosit de peste 20 de milioane de dezvoltatori pentru a us,ura dezvoltarea de API-uri. Are capacitatea de a face diferite tipuri de solicit ari HTTP (GET, POST, PUT, PATCH), salvˆand medii pentru utilizare ulterioar a, transformˆand API-ul ˆın cod pentru diferite limbaje

Pentru a vedea cum functioneaza api-ul meu am decis sa folosesc Swagger.

Ce este Swagger?

Swagger este un set de instrumente open-source construit în jurul specificației OpenAPI, care vă poate ajuta să proiectați, să construiți, să documentați și să utilizați API-uri REST. Principalele instrumente Swagger includ:

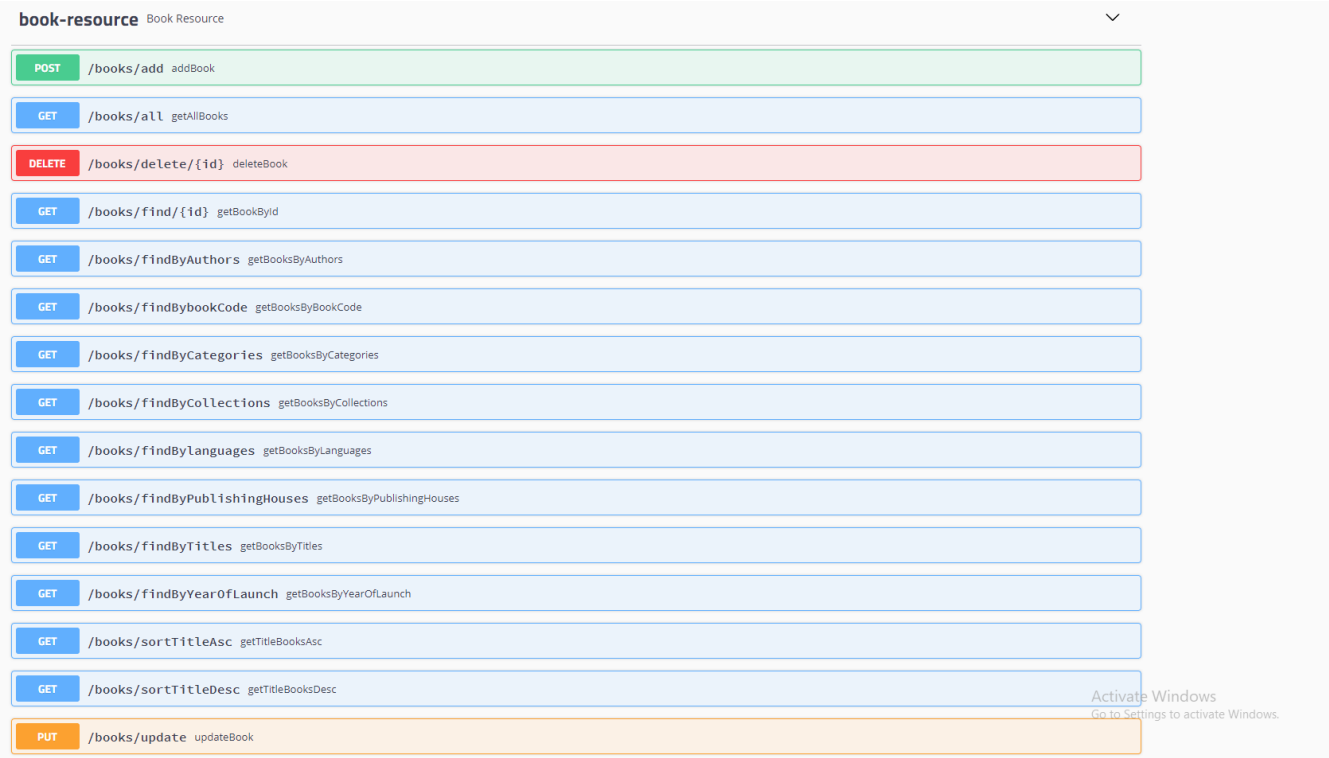
Swagger Editor – editor bazat pe browser în care puteți scrie specificații OpenAPI. Swagger UI – redă specificațiile OpenAPI ca documentație interactivă API. Swagger Codegen – generează stub-uri de server și biblioteci client dintr-o specificație OpenAPI.

Ce este OpenAPI?

Specificația OpenAPI (fosta Swagger Specification) este un format de descriere API pentru API-urile REST. Un fișier OpenAPI vă permite să descrieți întregul dvs. API, inclusiv: puncte finale disponibile (/books) și operații pe fiecare punct final (GET /books, POST /books) Parametri de operare Intrare și ieșire pentru fiecare operațiune Metode de autentificare Informații de contact, licență, termeni de utilizare și alte informații.

Accesand http://localhost:8080/swagger-ui.html dupa momentul in care aplicatia noastra ruleaza cu succes v-om avea access la api-ul nostru . De exemplu API-ul pentru

o carte putem utiliza api-ul de rest sau sa vedem descrierea endpointurilor arata in felul urmator:



3.5 Spring Security

Ce este Spring Security?

Definiţia 3.5.1. Spring Security este un cadru de autentificare și control al accesului puternic și extrem de personalizabil. Este standardul de facto pentru securizarea aplicațiilor bazate pe Spring.

Spring Security este un cadru care se concentrează pe furnizarea atât de autentificare, cât și de autorizare pentru aplicațiile Java. Ca toate proiectele Spring, puterea reală a Spring Security se găsește în cât de ușor poate fi extinsă pentru a îndeplini cerințele personalizate

Cum am implementat partea de securitate?

Am început să îmi implementez partea de securitate prin crearea unei noi entități numite AppUser, această entitate implementează interfață UserDetails . Această interfață oferă informații de baza pentru utilizator conținând metodele : getAuthorities() această metoda returnează doar un obiect Collection<GrantedAuthority> ea poate fi folosită la colectarea adecvată pentru a adăuga noua dvs. autoritate la acea colecție, getPassword() metodă care returnează parola, getUsername() metodă care returnează numele utilizatorului , isAccountNonExpired() returnează true dacă contul utilizatorului este valid (adică nu a expirat), false dacă nu mai este valid (adică a expirat) pentru a optimiza acest lucru am folosit un confirmationToken, isAccountNonLocked() care returnează true în cazul în care utilizatorul nu este blocat, false în caz contrar, isCredentialsNonExpired() returnează adevărat dacă datele de conectare ale utilizatorului sunt valide (adică nu au expirat), false dacă nu mai sunt valide (adică

au expirat) și isEnabled() care returnează true if the user is enabled, false otherwise . Implementările nu sunt utilizate direct de Spring Security în scopuri de securitate. Ele stochează pur și simplu informații despre utilizator care sunt ulterior încapsulate în obiecte de autentificare. Acest lucru permite ca informațiile despre utilizator care nu sunt legate de securitate (cum ar fi adrese de e-mail, numere de telefon etc.) să fie stocate într-o locație convenabilă. Entitatea utilizatorului arată în felul următor:

```
public class AppUser implements UserDetails {


    @SequenceGenerator(
            name = "user_sequence",
            sequenceName = "user_sequence",
            allocationSize = 1
    )
    @Id
    @GeneratedValue(
            strategy = GenerationType.SEQUENCE,
            generator = "user_sequence"
    )
    private Long id;
    2 usages
    private String firstName;
    2 usages
    private String lastName;
    2 usages
    private String email;
    2 usages
    private String password;
    2 usages
    @Enumerated(EnumType.STRING)
    private AppUserRole appUserRole;
    1 usage
    private Boolean locked = false;
    1 usage
    private Boolean enabled = false;

    1 usage
    public AppUser(String firstName,
                   String lastName,
                   String email,
                   String password,
                   AppUserRole appUserRole) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
        this.password = password;
        this.appUserRole = appUserRole;
    }
```

În imaginea de mai sus putem observă că appUserRole e de tip AppUserRole care reprezintă un enum cu rolurile de admin și utilizator. Următorul pas a fost să mă folosesc de JPA în crearea unei interfețe AppUserRepository în care am definit 2 metode findByEmail pentru a găsi un anumit user după emailul cu care s-a înregistrat în sistem și enableAppUser pentru a verifică dacă un user este activat sau dezactivat, lucru pe care îl verificăm în momentul în care un user face login, acțiune care nu este permisă pentru un user dezactivat . În continuare am creat o clasa AppUserService care reprezintă service-ul entității User, această clasa implementează interfață UserDetailsService . Această clasa conține 3 obiecte de tip AppUserRepository, BCryptPasswor-

dEncoder și ConfirmationTokenService .AppUserRepository este interfață menționată anterior, iar implementarea ei arată în felul următor:

```
1 usage
@Repository
@Transactional(readOnly = true)
public interface AppUserRepository
        extends JpaRepository<AppUser, Long> {

    2 usages
    Optional<AppUser> findByEmail(String email);

    1 usage
    @Transactional
    @Modifying
    @Query("UPDATE AppUser a " +
            "SET a.enabled = TRUE WHERE a.email = ?1")
    int enableAppUser(String email);

}
```

BCryptPasswordEncoder este implementarea PasswordEncoder care utilizează funcția de hashing puternică BCrypt. Clienții pot furniza opțional o „putere" și o instanță SecureRandom. Cu cât parametrul de putere este mai mare, cu atât va trebui depusă mai multă muncă (exponențial) pentru hashing parolele. Valoarea implicită este 10, iar în legătură cu ConfirmationTokenService urmează precizări mai detaliate despre implementare .

Ce este UserDetailsService?

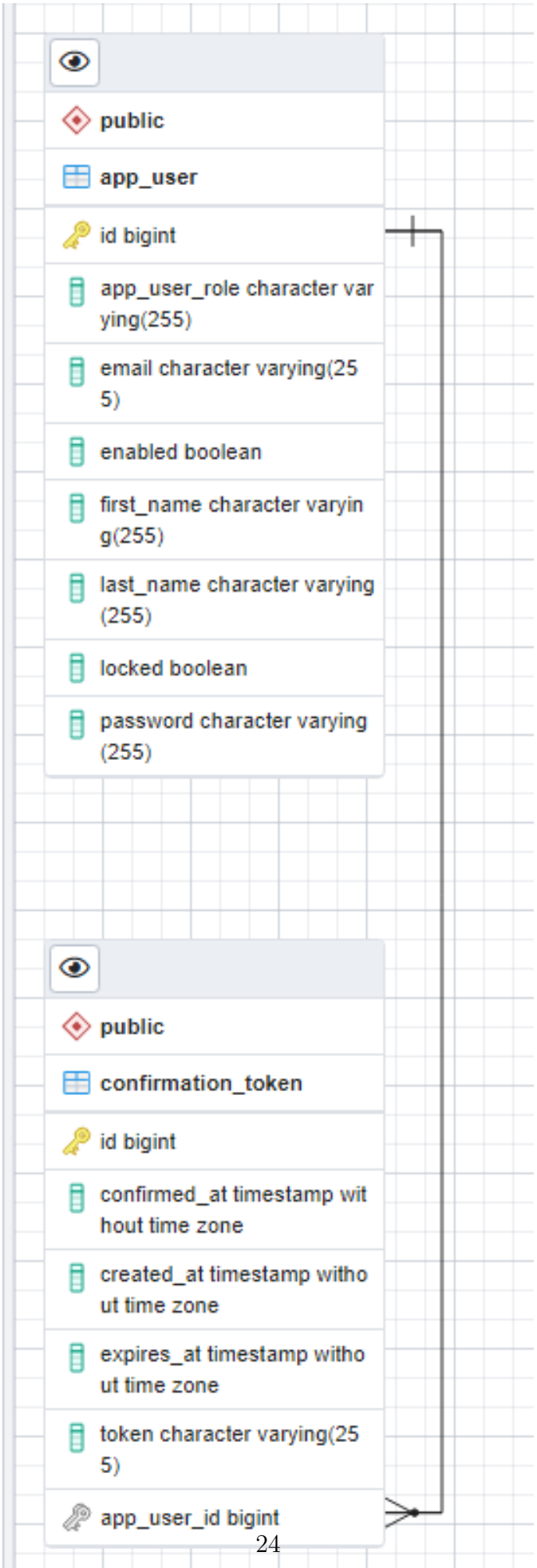
UserDetailsService este o interfață de bază în cadrul Spring Security, care este utilizată pentru a prelua informațiile de autentificare și autorizare ale utilizatorului. Această interfață are o singură metodă numită loadUserByUsername() pe care o putem implementa pentru a furniza informații despre client către API-ul de securitate Spring. Pe lângă metode loadUserByUsername din interfață UserDetailsService avem și implementarea metodei de sign în pentru un user. Implementarea metodelor din clasa AppUserService signUpUser și enableAppUser arată în felul următor:

```
36  @   public String signUpUser(AppUser appUser) {
37          boolean userExists = appUserRepository
38                  .findByEmail(appUser.getEmail())
39                  .isPresent();
40
41          if (userExists) {
42              // TODO check of attributes are the same and
43              // TODO if email not confirmed send confirmation email.
44
45              throw new IllegalStateException("email already taken");
46          }
47
48          String encodedPassword = bCryptPasswordEncoder
49                  .encode(appUser.getPassword());
50
51          appUser.setPassword(encodedPassword);
52
53          appUserRepository.save(appUser);
54
55          String token = UUID.randomUUID().toString();
56
57          ConfirmationToken confirmationToken = new ConfirmationToken(
58                  token,
59                  LocalDateTime.now(),
60                  LocalDateTime.now().plusMinutes(15),
61                  appUser
62          );
63
64          confirmationTokenService.saveConfirmationToken(
65                  confirmationToken);
66
67  //          TODO: SEND EMAIL
68
69          return token;
70      }
71
            1 usage
72      public int enableAppUser(String email) {
73          return appUserRepository.enableAppUser(email);
74      }
75  }
```

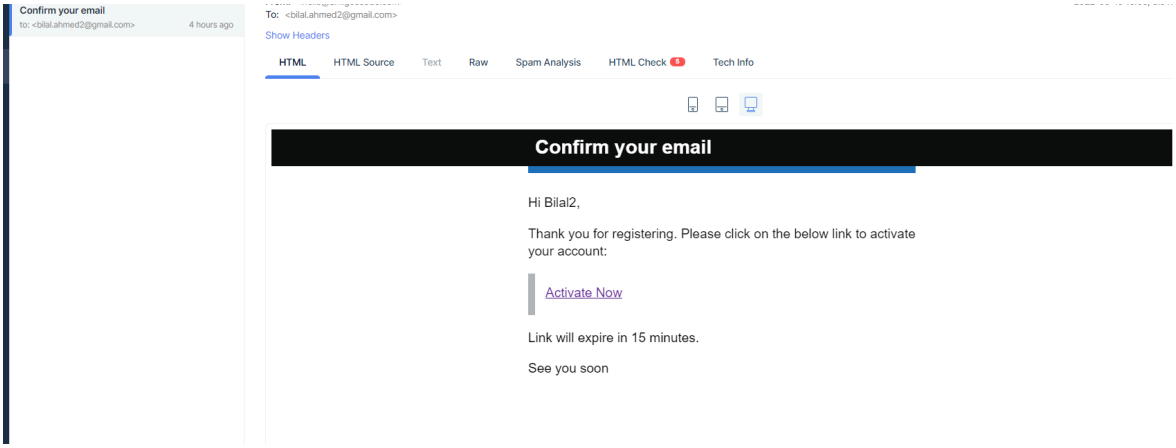
Putem observă la linia 57 că avem un obiect de tip ConfirmationToken care ne ajută la mecanismul de confirmare a înregistrării care obligă utilizatorul să răspundă la un e-mail „Confirmare înregistrare” trimis după înregistrarea cu succes pentru a-și verifica adresa de e-mail și a-și activa contul. Utilizatorul face acest lucru făcând clic pe un link unic de activare trimis prin e-mail.

```java
public class ConfirmationToken {

    @SequenceGenerator(
            name = "confirmation_token_sequence",
            sequenceName = "confirmation_token_sequence",
            allocationSize = 1
    )
    @Id
    @GeneratedValue(
            strategy = GenerationType.SEQUENCE,
            generator = "confirmation_token_sequence"
    )
    private Long id;

    1 usage
    @Column(nullable = false)
    private String token;

    1 usage
    @Column(nullable = false)
    private LocalDateTime createdAt;

    1 usage
    @Column(nullable = false)
    private LocalDateTime expiresAt;

    private LocalDateTime confirmedAt;

    1 usage
    @ManyToOne
    @JoinColumn(
            nullable = false,
            name = "app_user_id"
    )
    private AppUser appUser;
```

Putem observă la că appUser este însoțit de adnotarea @ManyToOne, ceea ce înseamnă că un user poate primii mai multe token-uri de confirmare pe email, lucru confirmat și de modelul bazei de date:


public
app_user
id bigint
app_user_role character varying(255)
email character varying(255)
enabled boolean
first_name character varying(255)
last_name character varying(255)
locked boolean
password character varying(255)
public
confirmation_token
id bigint
confirmed_at timestamp without time zone
created_at timestamp without time zone
expires_at timestamp without time zone
token character varying(255)
app_user_id bigint

În momentul în care utilizatorul își crează un cont pe site token-ul primim pe mail o să vină sub formă unui buton pe care scrie Activate Now . Pentru a fi sigur de bună funcționare a acestei implementări am folosit Mailtrap . Mailtrap este o soluție de server de e-mail de testare care permite testarea notificărilor prin e-mail fără a le trimite utilizatorilor reali ai aplicației dvs. . Mail-ul primit cu token-ul de confirmare în momentul înregistrării în sistem arată în felul următor:



Tokenul de confirmare are o valabilitate de doar 15 minute după care expiră și nu poate fi activat niciodată, pentru a retrimite un alt token de confirmare va fi necesar să ne înregistrăm în sistem din nou .

Cum am implementat acest sistem de înregistrare?

Pentru a implementa acest sistem de înregistrare a fost nevoie să creez o nouă clasă RegistrationService care conține 4 obiecte, unul de tipul AppUserSerivce, pentru a apela metodă signUpUser explicată mai sus, un alt obiect din această clasa este unul de tip ConfirmationTokenService pentru a folosi metodele : saveConfirmationToken(ConfirmationToken token) care salvează tokenul de confirmare în baza de date, getToken(Strîng token) care găsește un token în baza de date după primary key, iar ultima metodă este setConfirmedAt(Strîng token) care verifică dacă tokenul este expirat sau nu . Cele două noi obiecte din această clasa despre care nu am vorbit sunt EmailSender și EmailValidator . EmailValidator este o clasa care conține o metodă test care ne returnează true în cazul în care mail-ul este valid și fals în caz contrat, EmailSender este o interfață care conține o metodă de send, această metodă setează titlul mailului și de către cine a fost trimis. RegistrationService se folosește de aceste 4 obiecte pentru implementarea a trei metode: metodă de înregistrare, de confirmare a tokenului și de construire a mailului trimis. Mai jos este implementarea metodei register care are că parametru un RegistrationRequest care conține atributele: firstName, lastName, email și parolă, iar în corpul metodei avem verificarea mailului, crearea tokenului pentru un user care dorește să facă signUp și trimiterea tokenului pe email, iar tipul de return e Strîng deoarece returnează tokenul.

```java
public String register(RegistrationRequest request) {
    boolean isValidEmail = emailValidator.
            test(request.getEmail());

    if (!isValidEmail) {
        throw new IllegalStateException("email not valid");
    }

    String token = appUserService.signUpUser(
            new AppUser(
                    request.getFirstName(),
                    request.getLastName(),
                    request.getEmail(),
                    request.getPassword(),
                    AppUserRole.USER

            )
    );

    String link = "http://localhost:8080/api/v1/registration/confirm?token=" + token;
    emailSender.send(
            request.getEmail(),
            buildEmail(request.getFirstName(), link));

    return token;
}
```

Următoarea implementare e a metodei confirmToken(Strîng token) care verifică dacă token-ul a fost găsit în baza de date, dacă a mai existat un email identic confirmat anterior și dacă tokenul a expirat sau nu, dacă se trece cu succes de toate aceste verificări avem un token valabil pe care îl returnăm .

```java
@Transactional
public String confirmToken(String token) {
    ConfirmationToken confirmationToken = confirmationTokenService
            .getToken(token)
            .orElseThrow(() ->
                    new IllegalStateException("token not found"));

    if (confirmationToken.getConfirmedAt() != null) {
        throw new IllegalStateException("email already confirmed");
    }

    LocalDateTime expiredAt = confirmationToken.getExpiresAt();

    if (expiredAt.isBefore(LocalDateTime.now())) {
        throw new IllegalStateException("token expired");
    }

    confirmationTokenService.setConfirmedAt(token);
    appUserService.enableAppUser(
            confirmationToken.getAppUser().getEmail());
    return "confirmed";
}
```

În momentul în care utilizatorul trece de tot acest proces de înregistrare și autentificare

el va fi redirecționat către pagină de home, unde va avea access la toate metodele de căutare, filtrare și sortare .

3.6 Angular

3.7 node js

Capitolul 4

Arhitectura aplicatiei

4.1 1

4.2 2

4.3 3

Capitolul 5

Detalii de implementare

5.1 1.be

5.2 2.fe

Bibliografie

https://spring.io