

Dokumentacija softverskog rada



C#/.NET aplikacija za moderniju i efikasniju katalogizaciju
i pretraživanje građe u knjižnicama

Izradio: Ivan Patarčić, OŠ Matije Gupca

Mentorica: Marina Antolović

Vanjski mentor: Dražen Penić

Županijsko natjecanje u kategoriji Razvoj softvera

U Zagrebu, 9. ožujka, 2022.

1. Svrha programa

1.1. Osnovna svrha programa

Svrha *ShElfa* je efikasno upravljanje katalogom knjižnica i distribucijom građe, sa naglaskom na školske knjižnice.

Daljnji principi po kojima bi rješenje bilo napravljeno su:

- mora biti web-rješenje za svrhe praktičnosti;
- mora imati mogućnost jedne baze podataka za više lokacija;
- mora biti dizajnirano za efikasno snalaženje u programu;
- program mora biti održiv kako bi bilo jednostavno napraviti cijeli ekosustav oko temeljne aplikacije.
- program mora udovoljavati zahtjevima ne samo korisnika, već i knjižničarske struke (suradnja i s knjižničarima i s korisnicima pri razvoju aplikacije)

1.2. Slična rješenja na tržištu

Hrvatsko tržište za upravljanje katalogima knjižnica u hrvatskoj je zasićeno zastarjelim i/ili nepraktičnim rješenjima, od kojih se gotovo sve hrvatske škole služe Metel(Win)om, programom razvijanim od 1990-ih od tvrtke Point.

Slična rješenja u Hrvatskoj (*ZaKi Book* – program KGZ-a, *Buki* – program NSK) nisu dovoljno prilagođena školskim knjižnicama.

Metel, iako sa puno značajki koje u stručnom smislu odgovaraju zahtjevima školskih knjižničara je prezastarjele arhitekture, a mnoge značajke nisu dobro implementirane te ih tako većina korisnika ne koristi s lakoćom.



To je vidljivo na primjeru integracije više Metel programa sa zajedničkom bazom podataka (Metel ovo radi tako što računala dijele baze podataka ne sa centralnom bazom podataka, već sa slanjem podataka o knjigama međusobno), koju malo stvarnih korisnika koristi.

Na primjeru moje matične škole, OŠ Matije Gupca, škola za svrhu konzistencije u dostupnosti informatičkog sustava dijeli ono što bi trebala biti jedna baza podataka na dvije na dva različita uređaja – jedan sa knjigama i korisnicima internacionalnog programa, drugi sa knjigama i korisnicima nacionalnog programa.

Zbog ovoga učenici nacionalnog programa škole od početka korištenja *Metela*, 2004. g., ne mogu na jednom korisničkom računu posuđivati knjige na engleskom jeziku, makar su često za svrhe čitanja u slobodno vrijeme popularnije od knjiga na hrvatskom.



Drugi primjer značajke *Metela* koja postoji samo na papiru je *online* katalog knjižnice, koji zahtjeva prijavu e-mail adresa članova knjižnice od strane knjižničara, što većina knjižničara ni ne popisuje, a većina učenika ne koristi. Najkorisnija značajka sličnog sustava koji bi dobro funkcionirao bi bila rezervacija knjiga, koja bi bila posebno bitna u vrijeme posuđivanja lektire, u koje knjižnice sa manjkom lektirnog fonda *Metel*, ako je korisnik registriran, nudi opciju rezerviranja, ali ona samo šalje elektronskom poštom obavijest knjižničaru.

Nadalje, učenici nisu zainteresirani za korištenje aplikacija sa zastarjelim i kompliciranim sučeljem i opcijama.

1.3. Ideja i plan razvoja (koncept) programa

NAPOMENA: sljedeći tekst nije prikaz trenutnog programa, već je planirani krajnji program. Za trenutno implementirane planove pogledati poglavlje 2.

Program (tj. ekosustav programa za upravljanje katalogom knjižnice) treba imati jednu centralnu bazu podataka kako bi podržavalo sve eksterne¹ aplikacije.

1.4 Temeljni dio aplikacije *ShElf*- aplikacija za knjižničare

(*ShElf*) Temeljni dio aplikacije koji služi za katalogizaciju knjiga, posudbu uživo i upravljanje ostalim aspektima knjižničnih baza podataka (provjera stanja, određivanje zakasnine, provjera lektire) je server-side web-aplikacija.

Sadrži dio za katalogizaciju knjiga baziran na metodi upisivanja ISBN-a i automatskog popunjavanja ostatka polja putem nekog otvorenog API-a kako bi skeniranjem standardnih ISBN barkodova svakog izdanja knjige knjižničari mogli odmah upisati cijelu knjigu, te automatski generirati redalicu. Polja se sva, naravno, mogu ručno izmijeniti u slučaju pogrešaka u API-u.

Ima mogućnost migracije starih baza podataka *Metela*, što bi ga činilo jedinim proizvodom na tržištu s tom opcijom.

Rješenje problema više knjižnica na jednom mjestu bi radilo na principu računara koji ima mogućnost upravljati više knjižnica te svrstavati knjige pod bilo koju od upisanih knjižnica.

Budući da je aplikacija namijenjena za korištenje u hrvatskim školama, uz upisivanje učenika pod razrede u određenoj školskoj godini planirana je integracija sa CARNet-ovim E-dnevnikom za učenike, prijave putem AAI@Edu platforme, što bi eliminiralo veliku količinu problema sa učeničkim pristupima katalogu školske knjižnice i online pristupu posudbama.

Također, knjižničari bi mogli imati uvid u plan lektire za cijelu godinu (koje profesori hrvatskog upisuju u E-dnevnik) kako bi izbjegli dva razreda koji u isto vrijeme posuđuju istu knjigu, što se često događa i završava tako da puno učenika nema knjige za vrijeme lektira.



1.5 Lokator2, on-site aplikacija za lociranje knjižne građe na policama

Prva eksterna aplikacija je on-site mobilna aplikacija za nalaženje knjiga na digitalnoj karti knjižnice. Sastoji se od programa za iscrtavanje 2D karte knjižnice i katalogiziranja po mjestu u knjižnici.

S ovom 2D kartom članovi knjižnice mogu kroz sustav jednostavne navigacije po policama odmah pronaći knjige, bez upotrebe kompliciranih knjižničnih kataloga, što bitno olakšava nalaženje knjiga učenicima i eliminira velik dio posla posuđivanja knjiga knjižničarima.

Karta je u školama namijenjena za prikaz na školskim tabletima, ili drugim uređajima koje ne-školske knjižnice žele koristiti.

Pojednostavljeno, prilikom upisivanja tražene građe na karti knjižnice kao rezultat bi bilo označeno mjesto na točno određenoj polici u knjižnici gdje se traženi primjerak nalazi.

¹ – pod eksterno se misli na sve aplikacije koja nisu temeljna aplikacija (*Shelf*) u ekosustavu.

² – lokator i mobilna aplikacija još nisu imenovani

1.6 Mobilna aplikacija za korištenje knjižnice za članove

Treća eksterna aplikacija je mobilna aplikacija koju koriste članovi knjižnica kako bi:

- provjerili dostupnost knjiga u knjižnici;
- našli najbližu knjižnicu sa određenom knjigom;
- rezervirali knjige;
- mogli provjeriti svoje posuđene knjige;
- općenito imali kontrolnu ploču svojeg članstva u knjižnicama;
- sastavili listu knjiga koje žele pročitati (wishlist) i dobili obavijesti o dostupnosti tih knjiga u blizini korisnika;
- recenzirali knjige.

Neke od pogodnosti ove aplikacije i njene integracije za knjižnice su:

- lakša komunikacija sa članovima (upozorenja o kašnjenju, dostupnost knjiga, etc.)
- lakše popisivanje članova
- sustav povratnih informacija za članove, nešto slično svojevrsnoj kreditnoj sposobnosti, ali za knjige, kako bi članovi sa kojima je bilo posebice ugodno raditi (ne kasne, dolaze često, posuđuju knjige) mogli dobiti pogodnosti (smanjenje zakasnine, prioritarna rezervacija) za poticaj čitanja i poticaj odgovornog članstva.

Zadnja ideja je nazvana Mjera Aktivnosti i Pouzdanosti Člana te će se u ostatku dokumentacije kratiti sa MAPČ.



MAPČ se sastoji od više različitih pokazatelja aktivnosti člana, te pokazatelj koji se jednostavno naziva aktivnost (broj posuđenih knjiga u nekom vremenskom periodu) može biti korišten za socijalni element aplikacije i poticanje čitanja.

Mobilna aplikacija je isto tako povezana sa AAI@Edu uslugama, kako bi bila pristupačna učenicima, pogotovo učenicima razredne nastave, koji većinom nemaju E-mail adrese.

Prijava u neku knjižnicu bi bila vrlo jednostavna, sa dodiranjem mobitela na neki NFC tag ili skeniranje QR koda u knjižnici te prihvaćanjem člana u knjižnicu od strane knjižničara.



Bili korisnici AAI@Edu registrirani ili registrirani putem standardne e-mail adrese, mogu povezati AAI@Edu račun za funkcionalnosti provjere lektire, rezervacije i njene dostupnosti.

2. Strukture podataka u trenutnoj implementaciji

Svi podaci u aplikaciji ovise o sljedećim klasama:

2.1. Klase povezane s knjigama (Domain/Entities/Knjige)

- I. Klasa knjige (**Knjiga**) se sastoji od sljedećih polja:
 - a) cijeli broj **Id**, interni kataložni broj knjige;
 - b) cijeli broj **KnjiznicaId**, identifikacijski broj knjižnice kojoj knjiga pripada;
 - c) niz znakova **Naslov**, naslov knjige;
 - d) niz znakova **Redalica**, redalica knjige;
 - e) niz znakova **Autor**, autor knjige;
 - f) niz znakova **ISBN**, ISBN13 knjige;
 - g) niz znakova **ISBN10**, ISBN10 knjige;
 - h) niz znakova **Opis**, kratki opis knjige;
 - i) cijeli broj **IzdavačId**, identifikacijski broj/oznaka izdavača knjige;
 - j) *Izdavac* **Izdavac**, polje sačinjeno od klase *Izdavac*, izdavač knjige;
 - k) niz znakova **Godinaldanja**, godina izdanja knjige.

- II. Klasa izdavača (**Izdavac**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj/oznaka izdavača;
 - b) niz znakova **Naziv**, naziv izdavača;
 - c) *OIB* **Oib**, OIB izdavača;
 - d) *Adresa* **Adresa**, adresa izdavača.

- III. Klasa primjerka knjige (**Primjerak**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj primjerka;
 - b) cijeli broj **KnjigaId**, identifikacijski broj knjige primjerka;
 - c) *Knjiga* (II.I.I.) **Knjiga**, knjiga primjerka;
 - d) cijeli broj **InventurniBroj**, inventurni broj primjerka;
 - e) niz znakova **InventurnaOznaka**, inventurna oznaka primjerka;
 - f) niz znakova **BarCode**, barkod primjerka.
 - g)

- IV. Klasa autora knjige (**AutorKnjige**) sastoji se od sljedećih polja:
 - a) cijeli broj **AutorId**, identifikacijski broj autora knjige;
 - b) *Autor* **Autor**, autor knjige;
 - c) cijeli broj **KnjigaId**, identifikacijski broj knjige koju je napisao;
 - d) *Knjiga* **Knjiga**, knjiga koju je napisao;
 - e) cijeli broj **RedniBroj**, redni broj autora.

2.2. Klase povezane sa autorima (Domain/Entities/Autori)

- I. Klasa autora (**Autor**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj autora;
 - b) niz znakova **Ime**, ime autora;
 - c) niz znakova **Prezime**, prezime autora;
 - d) niz znakova **Opis**, kratki opis knjige.

2.3. Klase povezane s članovima (Domain/Entities/Clanovi)

- I. Klasa člana (**Clan**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj člana;
 - b) cijeli broj **Brojskaznice**, broj iskaznice člana;
 - c) *Knjiznica* **Knjižnica**, knjižnica kojoj član pripada;
 - d) cijeli broj **Oso bald**, identifikacijski broj osobe koja predstavlja člana;
 - e) *Osoba* **Osoba**, osoba koja predstavlja člana;
 - f) *DateOnly* **PocetniDatum**, datum početka članstva;
 - g) *DateOnly* **KrajClanstva**, datum kraja članstva;
 - h) Boolean **Aktivan**, status aktivnosti člana.

2.4. Klase povezane s knjižnicama (Domain/Entities/Knjiznice)

- I. Klasa knjižnice (**Knjiznica**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj knjižnice;
 - b) niz znakova **Naziv**, naziv knjižnice;
 - c) *Adresa* **Adresa**, adresa knjižnice;
 - d) cijeli broj **VlasnikId**, identifikacijski broj vlasnika knjižnice;
 - e) *Vlasnik* **Vlasnik**, vlasnik knjižnice.

2.5. Klase povezane sa korisnicima (Domain/Entities/Tenants)

- I. Klasa vlasnika (**Vlasnik**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj vlasnika;
 - b) niz znakova **Naziv**, naziv/ime vlasnika;
 - c) *Adresa* **Adresa**, adresa vlasnika.

2.6. Klase povezane sa školama (Domain/Entities/Skole)

- I. Klasa razreda (**Razred**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj razreda;
 - b) niz znakova **Naziv**, naziv razreda;
 - c) cijeli broj **SkolaId**, identifikacijski broj škole kojoj razred pripada;
 - d) *Vlasnik Skola*, škola prikazana kao klasa *Vlasnik* (II.V.).

- II. Klasa razrednog odjeljenja (**RazrednoOdjeljenje**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, identifikacijski broj razrednog odjeljenja;
 - b) niz znakova **Naziv**, naziv razrednog odjeljenja;
 - c) cijeli broj **RazredId**, identifikacijski broj razreda kojemu odjeljenje pripada;
 - d) cijeli broj **SkolskaGodinaId**, identifikacijski broj školske godine kojoj razred pripada;
 - e) *SkolskaGodina SkolskaGodina*, školska godina kojoj razred pripada.

- III. Klasa školske godine (**SkolskaGodina**) sastoji se od sljedećih polja:
 - a) cijeli broj **Id**, unikatni identifikacijski broj školske godine;
 - b) niz znakova **Naziv**, naziv školske godine;
 - c) *Period Period^{II.VII.IV}*, period/razdoblje između početka i kraja školske godine;
 - d) cijeli broj **SkolaId**, identifikacijski broj škole kojoj školska godina pripada;
 - e) *Vlasnik Skola*, škola kojoj školska godina pripada.

2.7. Vrijednosni objekti (Domain/Entities/ValueObjects)

- I. Klasa adrese (**Adresa**) sastoji se od sljedećih polja:
 - a) niz znakova **Mjesto**, grad/naselje/mjesto u kojem je adresa;
 - b) niz znakova **Ulica**, ulica adrese;
 - c) niz znakova **KucniBroj**, kućni broj adrese;
 - d) niz znakova **PttBroj**, poštanski broj mjesta;
 - a) niz znakova **NazivPoste**, naziv pošte adrese;
 - b) niz znakova **Drzava**, država adrese.

- II. Klasa boje (**Colour**) služi za usklađivanje boja putem statičnih polja koji vraćaju heksadecimalne oznake boja.

- III. Klasa OIB-a (**OIB**) se sastoji od sljedećih polja:
 - a) niz znakova **Oib**, koji označava OIB korisnika/vlasnika. OIB mora biti *non-null* i mora imati točno 11 znamenaka, te u protivnom slučaju baca *InvalidOibException*.



IV. Klasa razdoblja/perioda (**Period**) se sastoji od sljedećih polja:

- a) *DateOnly* **StartDate**, datum početka razdoblja;
- b) *DateOnly* **EndDate**, datum kraja razdoblja.

StartDate i **EndDate** su u slučaju školske godine automatski postavljeni na prvi ponedjeljak u rujnu i prvi petak u srpnju, kako bi odmah na početku pokrili cijelu (hrvatsku) školsku godinu.

3. Objašnjenje imena i svrha polja u klasama

Kako puno polja u klasama dijele imena jer slijede ista pravila imenovanja, dva istoimena polja dvije različite klase neće biti različito pojašnjena. Klasa *Colour* i njena polja su izuzeti iz sljedeće tablice.

Polja/klase	Klase/konteksti	Objašnjenje
Id, AutorId	Sve osim <i>OIB</i>	Interni identifikacijski ključ objekata. Kod klase <i>Autor</i> se koristi AutorId . (provjeriti)
KnjiznicaId	<i>Knjiga, Clan</i>	U klasama koje imaju polje KnjiznicaId polje služi za referiranje na knjižnicu kojoj pripadaju.
KnjigaId	<i>AutorKnjige, Primjerak</i>	Polje služi za referiranje knjige od koje su autor, odnosno primjerak.
SkolaId	<i>Razred, SkolskaGodina</i>	Polja označavaju kojoj školi pripadaju/su dostupni objekti.
SkolskaGodinaId	<i>RazrednoOdjeljenje</i>	Polje označava od kad do kad je <i>RazrednoOdjeljenje</i> razred koji je.

4. Tehnologije korištene za implementaciju temeljne aplikacije

UI/UX temeljne aplikacije je pisan u Microsoft Blazor *Framework*-u, kao *server-side* aplikacija. Neke od pogodnosti Blazor-a su:

- *back-end* i *front-end* aplikacije su jako blizu; *back-end* je efektivno isti kao i *front-end* aplikacije budući da je cijela aplikacija *server-side* WebAssembly. Ovo čini pristup bazi podataka vrlo jednostavnim, bez potrebe za dodatne API-eve, već se može uvijek napraviti direktno;



- Uz *server-side* arhitekturu jednostavnosti pristupa bazi podataka pridodaje činjenica da je *back-end* (programa i Blazora) pisan u C#/.NET jeziku/*frameworku*.
- zbog Microsoftovog razvoja Blazora, eventualna distribucija kao PWA (posebice za Windows) bi bila jednostavna;
- općenita pogodnost web-aplikacija kao najefikasniji način distribucije, zbog obvezne *client-side* ažurnosti programa.

Za organiziranje komponenata i stiliziranje u skladu sa Bootstrap standardom je korišten Bootstrap, a za komponente sâme, uz standardne Blazor komponente je korišten RadzenBlazor. RadzenBlazor je jedina Open Source *library* komponenata koja podržava određene mogućnosti poput odabira datuma sa specifičnim formatom.

```
<div class="col-sm-3 col-md-3 inputdiv">  
  <InputText id="isbn" class="form-control" placeholder="Upiši ISBN10" @bind-Value="@_knjiga.ISBN10" @onkeydown="KeyboardEventHandlerAutoFill"/>  
</div>
```

Za samu bazu podataka je korišten PostgreSQL, kako bi aplikacija mogla raditi sa kompliciranijim sustavima podataka kao što mora koristiti aplikacija za pohranu knjiga.



5. Implementacija rješenja po modulima (stranicama)

Prototipna verzija u vrijeme pisanja dokumentacije se sastoji od modula za:

- upisivanje i izmjenu podataka o vlasniku;
- upisivanje i izmjenu podataka o školskim godinama;
- upisivanje i izmjenu podataka o razredima;
- upisivanje i izmjenu podataka o razrednim odjeljenjima;
- upisivanje i izmjenu podataka o knjižnicama;
- upisivanje i izmjenu podataka o članovima;
- upisivanje i izmjenu podataka o knjižnicama;
- katalogizaciju i izmjenu podataka o knjigama;
- zaduživanje i razduživanje knjiga;

5.1. Osnovni princip upisivanja podataka

Svaki od navedenih modula ima formu sastavljenu za upisivanje svih polja u njihovim klasama. Forma koja se koristi za sve što ima adresu (*Vlasnik, Knjižnica*) je sa nazivom **AdresaEdit** (`WebUI/shared/AdresaEdit.razor`) te ima elemente za unos svih polja u klasi *Adresa*.

Unatoč ovome, većina formi za unos su trenutno (zbog dovoljne raznolikosti podataka) napisane unutar samih (`.razor`) stranica.

Sva polja putem Blazor sustava validacije, koji je usko povezan sa samim metodama prenošenja u bazu podataka novostvorenih objekata. Način na koji Blazor validator radi je sljedeći:

- element `EditForm` – uzima parametar `Model`, koji uzima objekt klase koju objekt predstavlja/upisuje
- `EditForm` „obgrljuje“ sve ostale elemente forme, te stvara `EditContext`, koji djeci označava model klase
- prije elemenata za unos se stavlja `CustomValidation`, koji po `EditContext`-u označuje cijelu formu za validaciju podataka
- svi elementi dalje mogu imati **for** (provjeriti) parametre, koji označuju koje polje u klasi element mora popuniti
- validator zatim provjerava podatke po pravilima obično definiranima u naredbama za dodavanje/mijenjanje postojećeg (eng. *Upsert*) objekta (npr. `UpsertRazredCommandValidator.cs`)
- Validator ima naredbu koju po uspješnoj validaciji treba izvršiti, obično prije spomenuti *UpsertRazredComand*, i tako su podaci pročišćeni te upisani.
- Ukoliko nisu svi podaci zadovoljavajući po standardima validacije, Validator vraća poruke u zadnji element u formi, `ValidationSummary`.



Osoba - Novi

Prezime i ime	<input type="text" value="Prezime"/>	<input type="text" value="Ime"/>
Email	<input type="text" value="e-mail"/>	
Telefon	<input type="text" value="telefon"/>	
Mjesto	<input type="text" value="Mjesto"/>	
Ulica i kuć. broj	<input type="text" value="Ulica"/>	<input type="text" value="Kbr"/>
Poštanski broj	<input type="text" value="Ptt broj"/>	<input type="text" value="Posta"/>
Napomena	<input type="text" value="Napomena"/>	

Ostala polja služe za upis imena, prezimena, kontakt podataka i, opcionalno, napomena.

5.2.5. Razredi (ruta /razredi/create)

Razredi, kao što je prije spomenuto u objašnjenjima klasa, se sastoje isključivo od naziva, budući da su čisto kategorije u koje svrstavamo razredna odjeljenja. Naziv je izrečen nizom znakova zbog fleksibilnosti u nestandardnim oblicima razreda.

Razred - Novi

Naziv

5.2.6. Razredna odjeljenja (ruta razrednaodjeljenja/create/)

Razredno odjeljenje se sastoji od naziva, školske godine kojoj pripada, te razreda kojemu pripada. Školska godina i razred kojemu pripadaju moraju već biti stvoreni, tj. ne upisuju ih pri stvaranju razrednog odjeljenja učenik.

Razredno odjeljenje - Novi

Naziv

Školska godina

Razred

5.2.7. Knjižnice (ruta /knjiznice/create)

Knjižnice se sastoje od naziva knjižnice i AdresaEdit komponente.

Knjižnica - Novi

Naziv

Mjesto

Ulica i kuć. broj

Poštanski broj

5.2.8. Knjiga (ruta /knjige/create)

Knjiga je jedina forma sa automatskim svih polja. Knjižničari za čitanje prvog polja, polja sa ISBN-om knjige (može biti u formatu ISBN13 ili ISBN10) mogu koristiti skener barkoda. Skener automatski ispisuje ISBN i imitira pritisak tipke Enter.

Kako skener pritišće tipku Enter, ISBN polje detektira pritisak tipke, te program napravi API poziv i sa *Google Books* API-a preuzima sve knjige koje su označene sa tim ISBN-om. Ova metoda radi za **veliku** većinu knjiga koje su u knjižnicama, s time da zna vraćati krive knjige ukoliko je knjiga apsolutno opskurno.

Neke knjige, pak, mogu imati krivi ISBN označen na koricama i unutrašnjosti (jedina takva knjiga pronađena tijekom testiranja je *Englesko-hrvatski informatički rječnik* za školu i dom Miroslava Kiša, a ISBN konflikt između nje i druge knjige nisam mogao razriješiti ni ručno, budući da postoje izvori koji pokazuju isti ISBN na obje nađene knjige).

Neke starije knjige (kao izdanje Školske knjige Novakovih Posljednjih Stipančića i Velegradskog podzemlja iz 1977.) nemaju uopće ISBN. U testiranju je takvih nađena samo jedna.

Unatoč gornjim slučajevima, većinom je metoda nalaženje putem ISBN-a precizna i efikasna, a definitivno je od dostupnih rješenja za brzo nalaženje knjiga najsigurnija.

Upis knjiga

ISBN za automatsku popunu

ISBN10

Redalica

Autor i naslov

Knjižnica

Godina izdanja

Reset

5.2.9. Posudba (ruta /posudba)

Posudba se sastoji od upisa identifikacijskih brojeva osobe i knjige. Kada knjižničar upiše identifikacijski broj člana, dobije pregled trenutno zaduženih knjiga na člana, kako može znati može li korisnik posuditi knjigu.

Posudba

Osoba

Knjiga

Redalica	Naziv knjige	Datum posuđenja	Dani od posuđenja
DIVKOVIĆ, Mirko	Latinsko-hrvatski rječnik	06.03.2022.	3

Posuđene knjige člana **Mirko Divković**

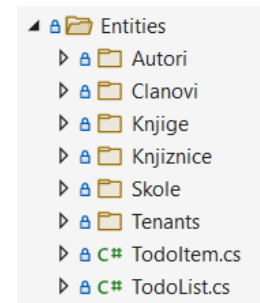
6. Back-end implementacija po projektima

Back-end, kao što je opisano u tehnologijama implementacije, je zbog praktičnosti napravljen u C#/.NET, i dijeljen je na više projekata po sadržaju i svrsi datoteka unutar njih.

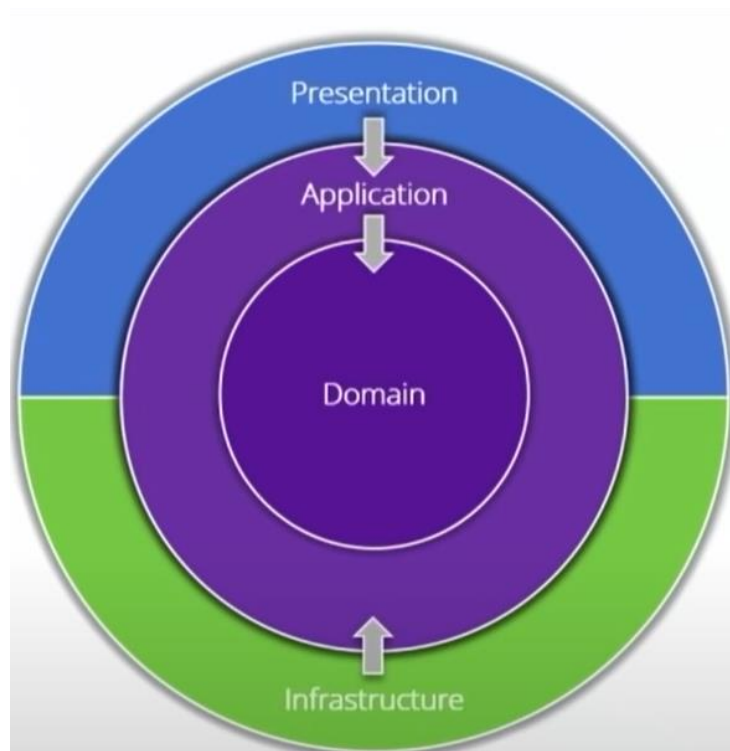
6.1. Domain (domena)

Kao što samo ime, domena, nalaže, domena sadrži sve klase i objekte koje program treba za uspješno pokretanje. Najvažniji dio domene je *folder* Entities (entiteti), koji sadrži sve informacije za klase korištene u programu.

Drugi važan dio je Exceptions, koji sadrži sve posebne *exceptione* koji bi se mogli dogoditi u izvršavanju programa.



6.2. Application (aplikacija)





7. Plan razvoja

7.1. Stanje i plan razvoja temeljnog programa

U vrijeme pisanja dokumentacije i županijskog natjecanja implementirana je rudimentarna verzija temeljnog programa *ShElf* te je mali dio planiranih krajnjih značajki.

Uz to što je sada napravljen samo *ShElf* temeljni program, puno značajki planirane za *ShElf* zbog nedostatka vremena nisu implementirane. Te značajke su:

- AAI@Edu integracija;
- API za ostatak ekosustava;
- automatsko unaprjeđivanje razreda;
- dodati statistiku knjiga;
- dodati ispisivanje naljepnica;
- dodati kontakt članova;
- alati za računovodstvo;
- paziti na naručivanje knjiga;
- dodati zajedničku bazu podataka o knjigama koja bi eventualno mogla suplementirati veće, e.g. Google Books, GoodReads, etc.

Gornje su značajke koje još nisu uopće implementirane, no puno značajki treba unaprijediti:

- uskladiti dizajn i promijeniti ga da bude distinktniji;
- dodati postavke;
- dodati lokalizaciju;
- dodati lokalizaciju jezika;
- dodati mogućnost navigacije tablica tipkovnicom.

Temeljna aplikacija je sada namijenjena za hrvatsko tržište, specifično škole, jer je tržište u kojemu sličnih programa apsolutno manjka. Metel je korišten u gotovo svim školama, a generalni konsenzus je da je zastario program. Nema adekvatne zamjene za hrvatske škole. Zato je (trenutno) prioritet značajke koje zahtijevaju hrvatske škole, no sa vremenom i sa građenjem ostatka ekosustava *ShElf* bi bio zadovoljavajuće rješenje i za strano, a ne samo hrvatsko tržište.

7.2. Plan razvoja mobilne aplikacije

Mobilna aplikacija za članove knjižnica treba imati sljedeće značajke:



- recenziranje knjiga;
- pregled lektira;
- nalaženje knjiga u knjižnicama u blizini;
- rezervacija knjiga;
- računanje kvantitativne vrijednosti aktivnosti i pouzdanosti člana;
- *Material Design (Material You)* dizajn;
- lagana prijava i registracija u knjižnice putem neke vrste kôda;
- društvene značajke za poticanje čitanja putem natjecanja;
- (anonimne) recenzije knjižnica;
- provjera dostupnosti mjesta u knjižničnim čitaonama;
- mogućnost komunikacije knjižnica sa članovima.

Mobilna aplikacija će biti izrađena u Flutteru, te se povezati sa API-em koji će biti dio *ShElf* aplikacije. Identitet mobilne aplikacije će biti relativno odvojen od *ShElf* aplikacije. *Metel* ima mobilnu aplikaciju (*MetLib*), ali pati od istih problema kao *online* katalog *Metel*-a, nije dovoljno dobro integriran, traži više od knjižničara, i većina korisnika koji bi ga voljeli koristiti ni ne znaju za njega. Aplikacija ima ukupno cca. 500 preuzimanja na *Google Play* platformi.



Aplikacija je predviđena kao *interface* između korisnika i cijelog *ShElf* ekosustava.

7.3. Plan razvoja lokatora

Lokator knjiga funkcionira na principu 2D tlocrta koji knjižničar radi u jednostavnom *plotter* programu, gdje uz pozicije ormara svaki entitet koji predstavlja ormara sadrži i informacije o visini ormara u policama.

Lokator iz API-a temeljne aplikacije preuzima podatke o svim knjigama, te ih poreda leksikografskim redom po prezimenima autora, kako bi knjige bile poredane na policama. Tako knjižničar može ići po policama te kada primijeti da jedna knjiga prelazi u drugi odjeljak police označi to u aplikaciji, te aplikacija kategorizira knjigu pod sektor i podsektor u 2D karti.

Kada korisnik dođe u knjižnicu može na nekom uređaju (za škole – vjerojatno školski tableti) upisati redalicu ili naslov knjige te mu lokator može reći ima li knjige trenutno u knjižnici i, ako ima, gdje se nalazi.

Lokator će vjerojatno biti izrađen kao PWA, da može biti distribuiran na raznim uređajima koje knjižnice mogu naći, bila to stara računala ili novokupljeni tableti ili ostali Android uređaji.



8. Krajnji cilj razvoja ShElf ekosustava; zaključak

Trenutno hrvatsko tržište predvođeno *Metelom* uz činjenicu da pati od nedovoljno dostupnih rješenja koja su dovoljno moderna i ugodna za koristiti, pati od razvedenosti podataka o knjigama i knjižnicama. Agregacija tih podataka radi inovativnih rješenja koja bi centralizirala informacije o knjižnicama kako bi pregledavanje materijala u knjižnicama, dostupnih knjižnica i općenito planiranje čitanja i posuđivanja štiva bilo puno lakše je vrlo teška ili nemoguća.

Od informatizacije knjižnica krajem 1990.-ih do danas tržište u hrvatskoj je imalo konzistentan period stagnacije i, makar su nekad rješenja bila sasvim zadovoljavajuća dulje vrijeme (*Metel* je u svojih 20-ak godina postojanja konzistentno surađivao sa knjižničarima i pomagao sa implementacijom značajki koje su njima bile važne), postoji puno neiskorištenog potencijala koje bi novo rješenje moglo pravilno iskoristiti.

Ovo je cilj *ShElf* ekosustava, znatno pomoći knjižničarima sa modernijim alatom, promijeniti način interakcije korisnika i knjižnica, pružati učenicama bolji način za planirati čitanje lektire, davati knjižnicama način za nagrađivanje aktivnih i pouzdanih korisnika, poticati čitanje, te napokon ujediniti veliku količinu informatičkih sustava koji nemaju kvalitetan način međusobne komunikacije i dati korisnicima veći uvid u dostupno štivo u blizini.