

Univerzitet u Beogradu
Fakultet organizacionih nauka
Laboratorija za elektronsko poslovanje



E-biblioteka

Seminarski rad iz Internet Tehnologija

Profesor:
Marijana Despotović-Zrakić

Studenti:
Vladan Simonović 145/15
Milica Bogdanović 40/15
Marko Smoljanović 144/15

Februar, 2019.

Sadržaj

1.	Korisnički zahtev	4
2.	Opis sistema	6
2.1.	Slučajevi korišćenja	6
2.1.1.	SK Prijava korisnika	6
2.1.2.	SK Unos knjige	7
2.1.3.	SK Pretraživanje rezervacija	8
2.1.4.	SK Izmena knjige	9
2.1.5.	SK Unos nove slike u galeriju.....	10
2.1.6.	SK Unos nove rezervacije	11
2.1.7.	SK Brisanje rezervacija	12
2.1.8.	Prikaz grafika	13
2.1.9.	SK Prikaz knjiga.....	14
2.1.10.	SK Registrovanje korisnika	15
2.2.	Opis arhitekture	16
2.3.	Opis procesa-dijagrami sekvenci.....	17
2.3.1.	DS Prijava korisnika	17
2.3.2.	DS Unos knjige	18
2.3.3.	DS Pretraživanje rezervacija.....	19
2.3.4.	DS Izmena knjige	20
2.3.5.	DS Unos nove slike u galeriju	22
2.3.6.	DS Unos nove rezervacije.....	23
2.3.7.	DS Brisanje rezervacije	25
2.3.8.	DS Prikaz grafika	26
2.3.9.	DS Prikaz knjiga	27
2.3.10.	DS Registrovanje korisnika	28
2.4.	Model podataka	29
2.5.	Specifikacija REST API-ja.....	30
	Funkcija vratiZanrove()	30
	Funkcija vratiKnjige().....	31
	Funkcija ubaciKorisnika(data)	32
	Funkcija novaKnjiga(data)	33
3.	Opis korišćenih tehnologija.....	34

4. Korisničko uputstvo.....	35
5. Reprezentativni delovi koda	44
Prikaz svih knjiga	44
Prikaz mape.....	49
Izmena naziva knjige	50
Unos nove knjige.....	51
Unos nove slike u galeriju	51
Prikaz grafika	52
Registracija.....	55
Login.....	58
Prikaz svih rezervacija i brisanje rezervacije	58
Sopstveni servis.....	60

1. Korisnički zahtev

Potrebno je kreirati veb aplikaciju koja omogućava vođenje evidencije o članovima biblioteke, njihovim zahtevima, kao i knjigama. Korisnicima je potrebno omogućiti logovanje, a ukoliko korisnici ne postoje u bazi, omogućiti im registrovanje i upisivanje u bazu podataka. Korisnici koji imaju ulogu administratora, imaju mogućnost da unose nove knjige i menjaju naslove već unetih knjiga. Pored toga, omogućeno im je i brisanje rezervacija iz tabele rezervacije u bazi, kao i prikaz grafika, na kojem se prikazuju, procentualno, rezervacije knjiga. Administratori, kao i obični korisnici imaju mogućnost rezervacije. Administratori mogu i da ubacuju nove slike u galeriju slika.

Prisiput imaju **dve vrste korisnika**: administrator i korisnik

Administratoru omogućiti sledeće funkcionalnosti:

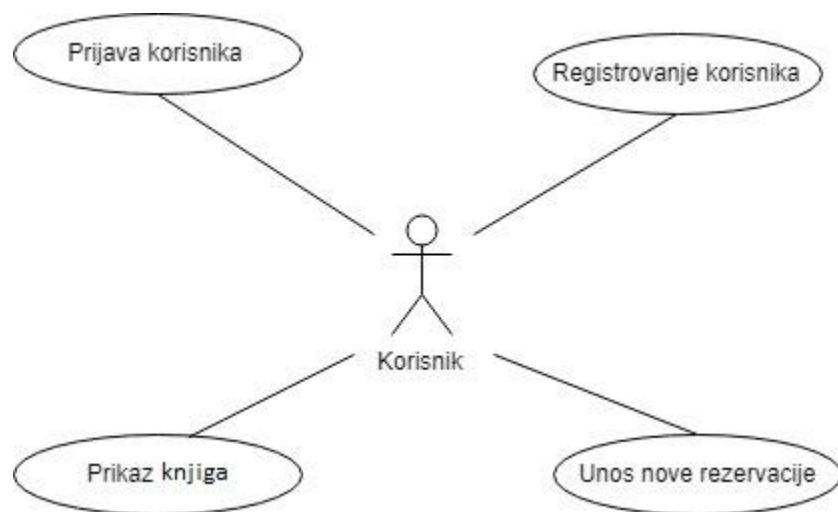
1. Prijava korisnika
2. Unos knjige
3. Izmena knjige
4. Pretraga rezervacija
5. Unos nove slike u galeriju
6. Unos nove rezervacije
7. Brisanje rezervacije
8. Prikaz grafika
9. Prikaz knjiga

Korisniku omogućiti sledeće funkcionalnosti:

1. Prijava korisnika
2. Registrovanje korisnika
3. Unos nove rezervacije
4. Prikaz knjiga



Slika 1. Slučajevi korišćenja za administratora



Slika 2. Slučajevi korišćenja za korisnika

2. Opis sistema

2.1. Slučajevi korišćenja

2.1.1. SK Prijava korisnika

Naziv SK

Prijavljivanje na sistem

Aktori Sk

Korisnik

Učesnici SK

Korisnik I sistem

Preduslov

Sistem je uključen

Osnovni scenario SK

1. Korisnik unosi korisničko ime I lozinku(APUSO)
2. Korisnik poziva sistem da se uloguje (proveri podatke) (APSO)
3. Sistem proverava podatke o korisniku (SO)
4. Sistem prikazuje korisniku poruku „Uspešno ulogovan korisnik“(IA)

Alternativni scenario

- 4.1. Ako sistem ne može da nađe korisnika, prikazuje poruku „Neuspešno prijavljivanje.“(IA)

2.1.2. SK Unos knjige

Naziv SK

Unos knjige

Aktori Sk

Administrator

Učesnici SK

Administrator i sistem

Preduslov

Sistem je uključen i administrator je ulogovan. Sistem prikazuje formu za rad sa knjigama.

Osnovni scenario SK

1. Administrator unosi podatke o knjizi (APUSO)
2. Administrator poziva sistem da zapamti podatke o knjizi (APSO)
3. Sistem pamti podatke o knjizi (SO)
4. Sistem prikazuje administratoru poruku „Uspešno uneta nova knjiga“ (IA)

Alternativni scenario

- 4.1. Ako sistem ne može da zapamti podatke o knjizi, administratoru se prikazuje poruka „Neuspešno unošenje nove knjige“ (IA)

2.1.3. SK Pretraživanje rezervacija

Naziv SK

Pretraživanje rezervacija

Aktori Sk

Administrator

Učesnici SK

Administrator i sistem

Preduslov

Sistem je uključen i administrator je ulogovan.

Osnovni scenario SK

1. Administrator unosi vrednost po kojoj pretražuje rezervacije (APUSO)
2. Administrator poziva sistem da nađe rezervacije po zadatoj vrednosti(APSO)
3. Sistem traži rezervacije po zadatoj vrednosti (SO)
4. Sistem prikazuje administratoru listu rezervacija (IA)

2.1.4. SK Izmena knjige

Naziv SK

Izmena knjige

Aktori Sk

Administrator

Učesnici SK

Administrator i sistem

Preduslov

Sistem je uključen i administrator je ulogovan. Učitana je lista knjiga.

Osnovni scenario SK

1. Administrator bira knjigu čiji naslov želi da menja (APUSO)
2. Administrator poziva sistem da prikaže izabranu knjigu (APSO)
3. Sistem traži podatke o knjizi (SO)
4. Sistem prikazuje izabranu knjigu (IA)
5. Administrator unosi nov naslov knjige (APUSO)
6. Administrator poziva sistem da zapamti naslov knjige (APSO)
7. Sistem pamti nov naslov knjige (SO)
8. Sistem prikazuje administratoru poruku „Uspešno izmenjen naslov knjige“ (IA)

Alternativni scenario

- 8.1. Ako sistem ne može da zapamti naslov knjige, prikazuje poruku „Neuspešno izmenjen naslov knjige.“(IA)

2.1.5. SK Unos nove slike u galeriju

Naziv SK

Unos nove slike

Aktori Sk

Administrator

Učesnici SK

Administrator i sistem

Preduslov

Sistem je uključen i administrator je ulogovan.

Osnovni scenario SK

1. Administrator bira novu sliku (APUSO)
2. Administrator poziva sistem da ubaci novu sliku u galeriju (APSO)
3. Sistem pamti novu sliku (SO)
4. Sistem prikazuje administratoru novu sliku u galeriji(IA)

2.1.6. SK Unos nove rezervacije

Naziv SK

Unos nove rezervacije

Aktori Sk

Korisnik (ili administrator)

Učesnici SK

Korisnik (ili administrator) i sistem

Preduslov

Sistem je uključen i korisnik je ulogovan. Sistem prikazuje formu za rad sa registracijama. Učitana je lista knjiga.

Osnovni scenario SK

1. Korisnik unosi podatke o rezervaciji (APUSO)
2. Korisnik poziva sistem da zapamti podatke o rezervaciji (APSO)
3. Sistem pamti podatke o rezervaciji (SO)
4. Sistem prikazuje korisniku poruku „Uspešno rezervisano“(IA)

Alternativni scenario

- 4.1. Ako sistem ne može da zapamti podatke o rezervaciji, korisniku prikazuje poruku „Neuspešna rezervacija“ (IA)

2.1.7. SK Brisanje rezervacija

Naziv SK

Brisanje rezervacija

Aktori Sk

Administrator

Učesnici SK

Administrator i sistem

Preduslov

Sistem je uključen i administrator je ulogovan. Sistem prikazuje formu za rad sa rezervacijama.

Osnovni scenario SK

1. Administrator unosi vrednost po kojoj pretražuje rezervacije (APUSO)
2. Administrator poziva sistem da pronađe rezervacije po zadatoj vrednosti (APSO)
3. Sistem traži rezervacije po zadatoj vrednosti (SO)
4. Sistem prikazuje administratoru listu rezervacija (IA)
5. Administrator bira rezervaciju koju će obrisati (APUSO)
6. Administrator poziva sistem da obriše izabranu rezervaciju (APSO)
7. Sistem briše rezervaciju (SO)

Alternativni scenario

- 7.1 Ako sistem ne može da obriše rezervaciju, administratoru prikazuje poruku „Došlo je do greške prilikom brisanja“ (IA)

2.1.8. Prikaz grafika

Naziv SK

Prikaz grafika

Aktori Sk

Administrator

Učesnici SK

Administrator i sistem

Preduslov

Sistem je uključen i administrator je ulogovan.

Osnovni scenario SK

1. Administrator poziva sistem da prikaže grafik (APSO)
2. Sistem obrađuje podatke i kreira grafik (SO)
3. Sistem prikazuje administratoru grafik (IA)

2.1.9. SK Prikaz knjiga

Naziv SK

Prikaz knjiga

Aktori Sk

Korisnik (ili administrator)

Učesnici SK

Korisnik (ili administrator)i sistem

Preduslov

Sistem je uključen.

Osnovni scenario SK

1. Korisnik poziva sistem da učitava listu knjiga (APSO)
2. Sistem učitava listu knjiga (SO)
3. Sistem prikazuje korisniku listu knjiga (IA)

2.1.10. SK Registrovanje korisnika

Naziv SK

Registrovanje korisnika

Aktori Sk

Korisnik

Učesnici SK

Korisnik i sistem

Preduslov

Sistem je uključen. Sistem prikazuje formu za registrovanje novog korisnika.

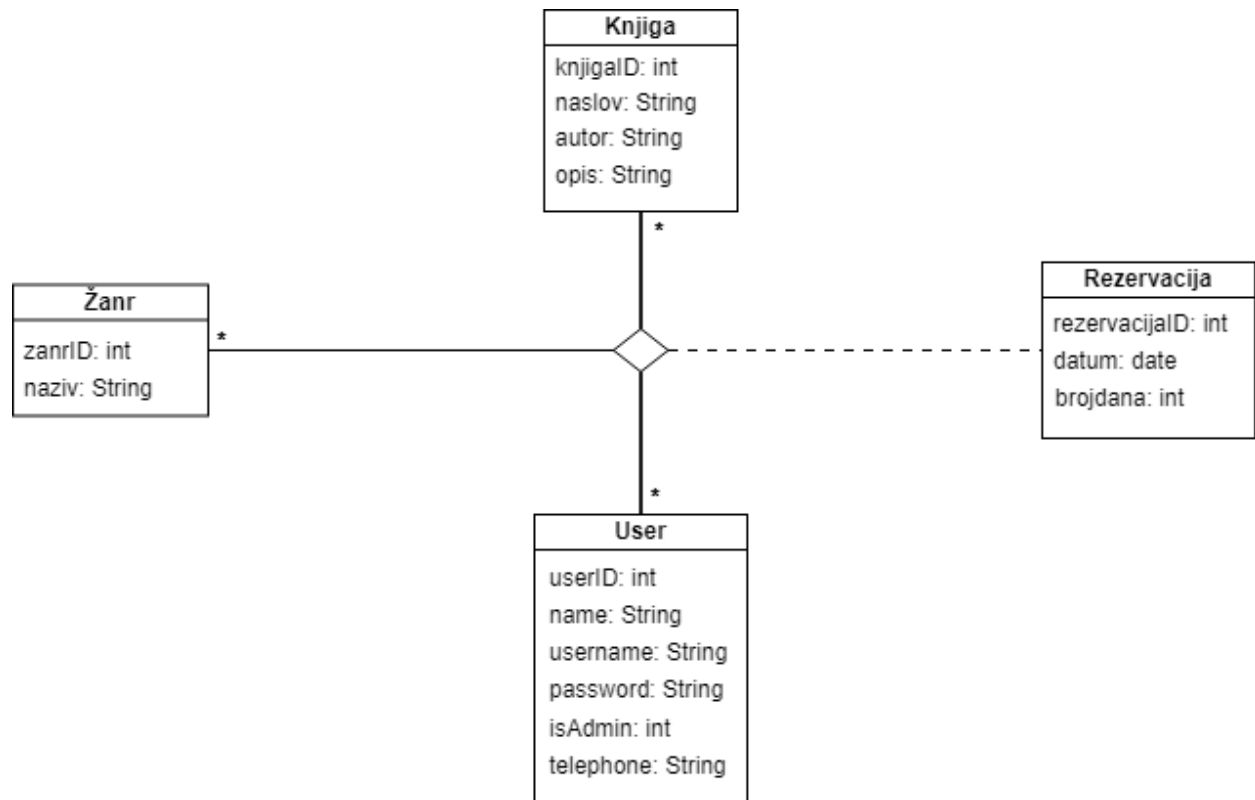
Osnovni scenario SK

1. Korisnik unosi podatke (APUSO)
2. Korisnik poziva sistem da zapamti podatke (APSO)
3. Sistem pamti podatke (SO)
4. Sistem prikazuje korisniku poruku „Uspešno registrovan korisnik“(IA)

Alternativni scenario

- 4.1. Ako sistem ne može da zapamti podatke o korisniku, prikazuje poruku „Neuspešna registracija korisnika.“(IA)

2.2. Opis arhitekture



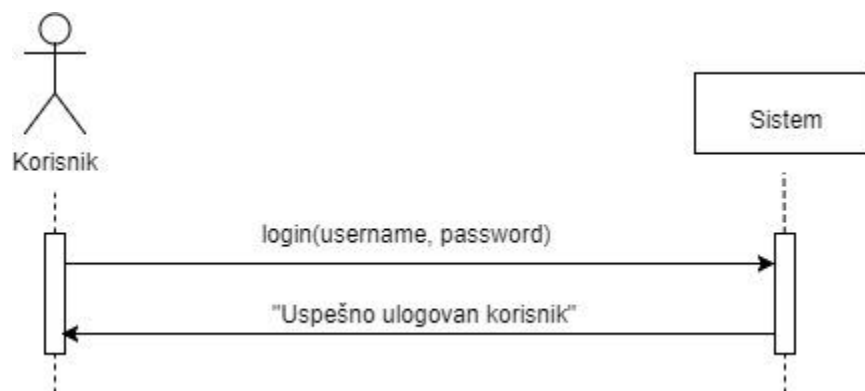
Slika 3. Dijagram klasa

2.3. Opis procesa-dijagrami sekvenci

2.3.1. DS Prijava korisnika

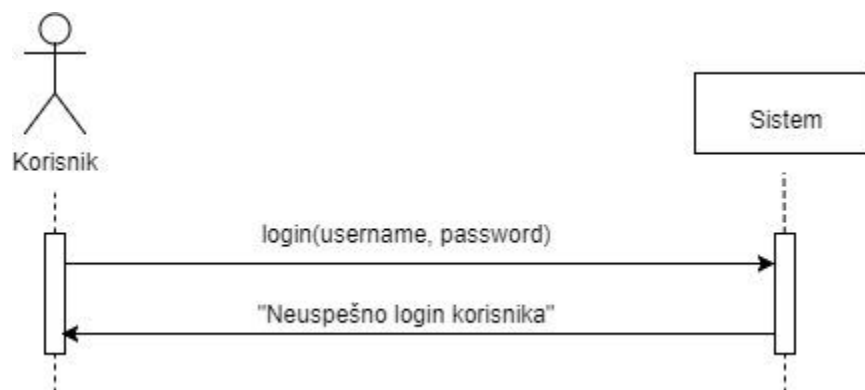
Osnovni scenario SK

1. Korisnik poziva sistem da se uloguje (proveri podatke) (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno ulogovan korisnik“ (IA)



Alternativni scenario

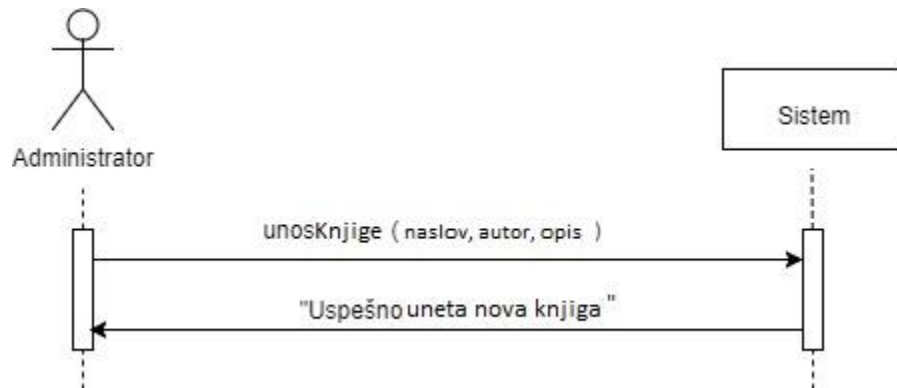
- 2.1. Ako sistem ne može da nađe korisnika, prikazuje poruku „Neuspešno login korisnika.“(IA)



2.3.2. DS Unos knjige

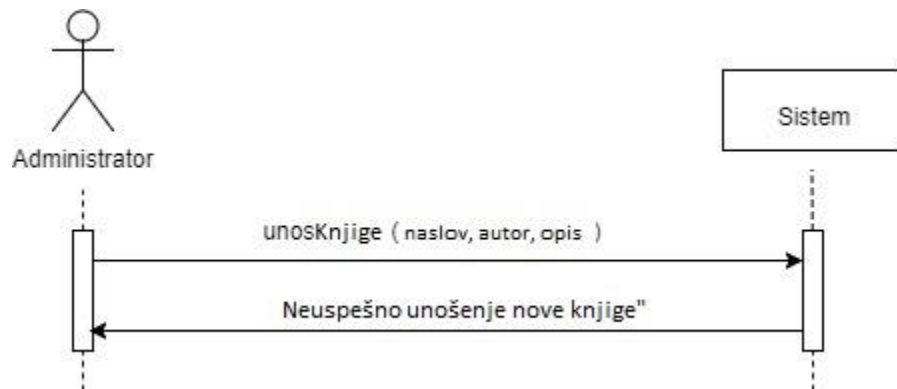
Osnovni scenario SK

1. Administrator poziva sistem da zapamti podatke o knjizi (APSO)
2. Sistem prikazuje administratoru poruku „Uspešno uneta nova knjiga“ (IA)



Alternativni scenario

- 2.1. Ako sistem ne može da zapamti podatke o knjizi, administratoru se prikazuje poruka „Neuspešno unošenje nove knjige“ (IA)



2.3.3. DS Pretraživanje rezervacija

Osnovni scenario SK

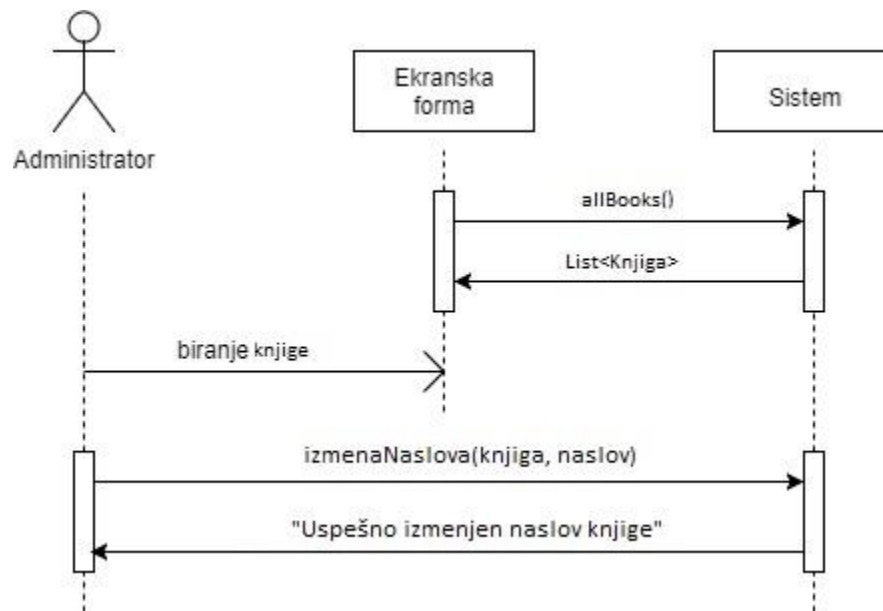
1. Administrator poziva sistem da nađe rezervacije po zadatoj vrednosti(APSO)
2. Sistem prikazuje administratoru listu rezervacija (IA)



2.3.4. DS Izmena knjige

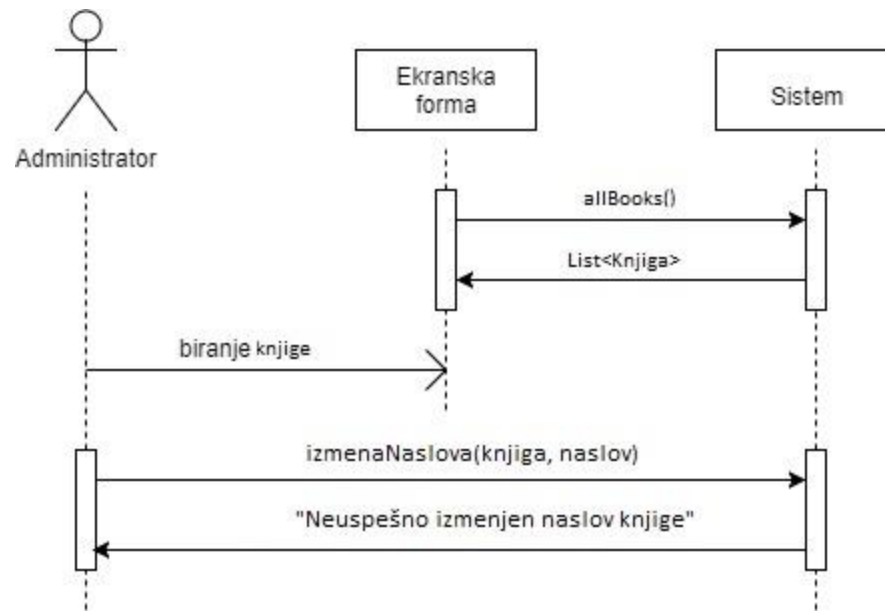
Osnovni scenario SK

1. Ekranska forma poziva sistem da učitá listu knjiga (APSO)
2. Sistem prikazuje listu knjiga (IA)
3. Administrator poziva sistem da prikaže izabranu knjigu (APSO)
4. Sistem prikazuje izabranu knjigu (IA)
5. Administrator poziva sistem da zapamti naslov knjige (APSO)
6. Sistem prikazuje administratoru poruku „Uspešno izmenjen naslov knjige“ (IA)



Alternativni scenario

- 6.1. Ako sistem ne može da zapamti naslov knjige, prikazuje poruku „Neuspešno izmenjen naslov knjige.“ (IA)



2.3.5. DS Unos nove slike u galeriju

Osnovni scenario SK

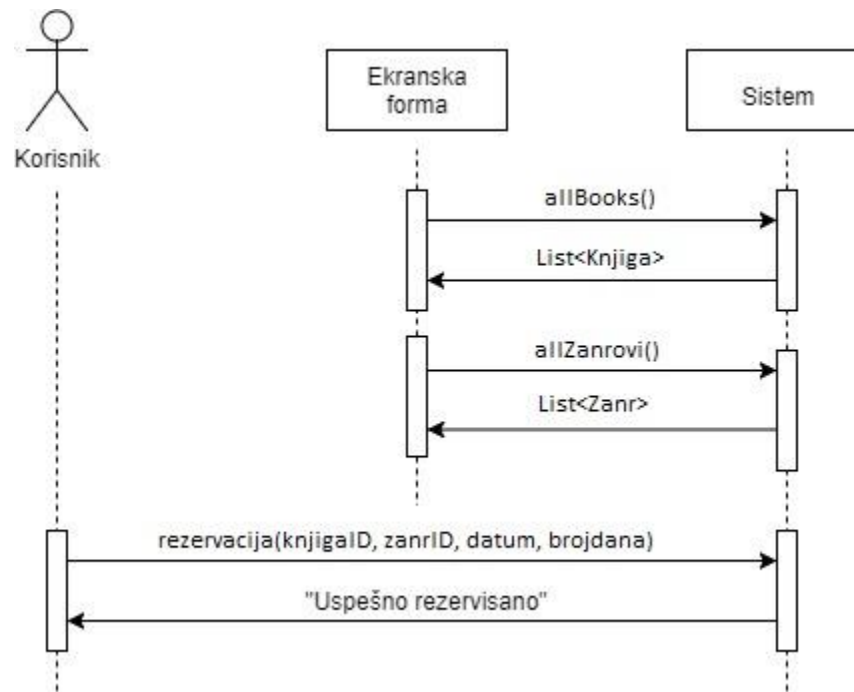
1. Administrator poziva sistem da ubaci novu sliku u galeriju (APSO)
2. Sistem prikazuje administratoru novu sliku u galeriji (IA)



2.3.6. DS Unos nove rezervacije

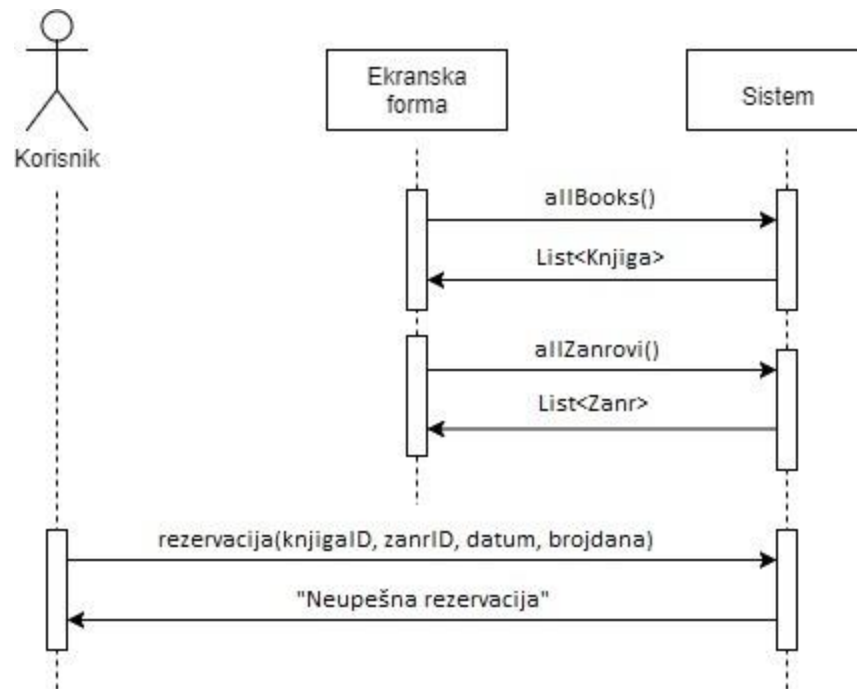
Osnovni scenario SK

1. Korisnik poziva sistem da zapamti podatke o rezervaciji (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno rezervisano“ (IA)



Alternativni scenario

- 2.1. Ako sistem ne može da zapamti podatke o rezervaciji, korisniku prikazuje poruku „Neuspešna rezervacija“ (IA)



2.3.7. DS Brisanje rezervacije

Osnovni scenario SK

1. Administrator poziva sistem da pronade rezervacije po zadatoj vrednosti (APSO)
2. Sistem prikazuje administratoru listu rezervacija (IA)
3. Administrator poziva sistem da obriše izabranu rezervaciju (APSO)



Alternativni scenario

- 3.1. Ako sistem ne može da obriše rezervaciju, administratoru prikazuje poruku „Došlo je do greške prilikom brisanja“ (IA)



2.3.8. DS Prikaz grafika

Osnovni scenario SK

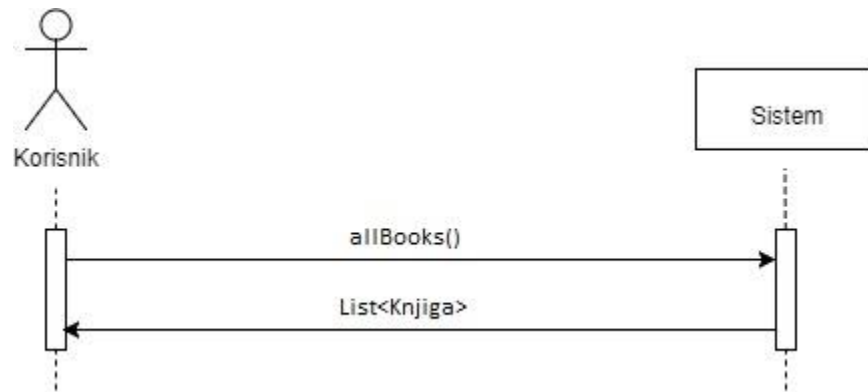
1. Administrator poziva sistem da prikaže grafik (APSO)
2. Sistem prikazuje administratoru grafik (IA)



2.3.9. DS Prikaz knjiga

Osnovni scenario SK

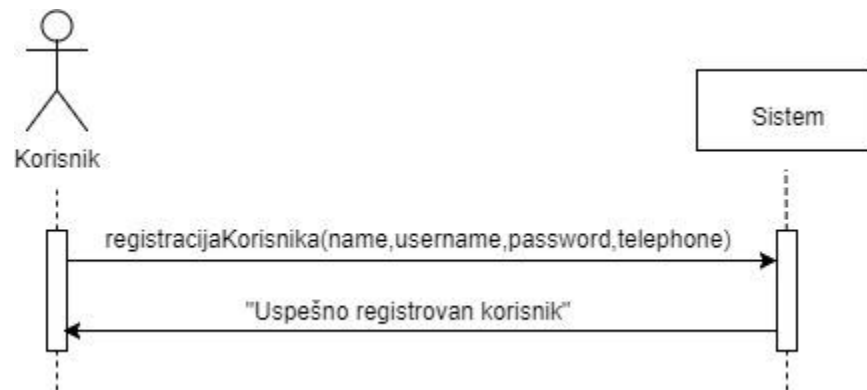
1. Korisnik poziva sistem da učitá listu knjiga (APSO)
2. Sistem prikazuje korisniku listu knjiga (IA)



2.3.10. DS Registrovanje korisnika

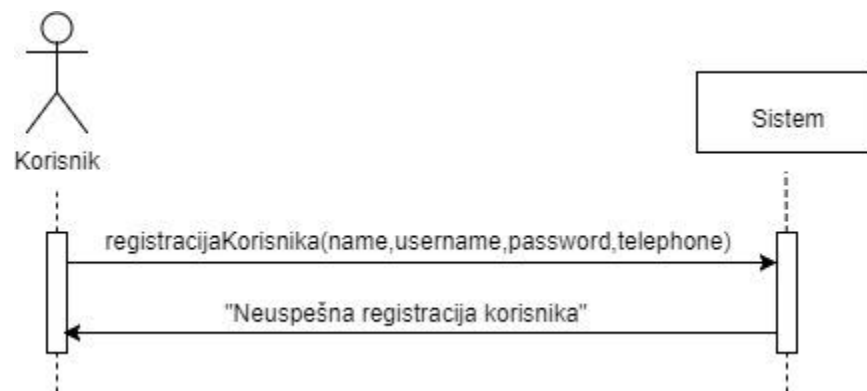
Osnovni scenario SK

1. Korisnik poziva sistem da zapamti podatke (APSO)
2. Sistem prikazuje korisniku poruku „Uspešno registrovan korisnik“ (IA)

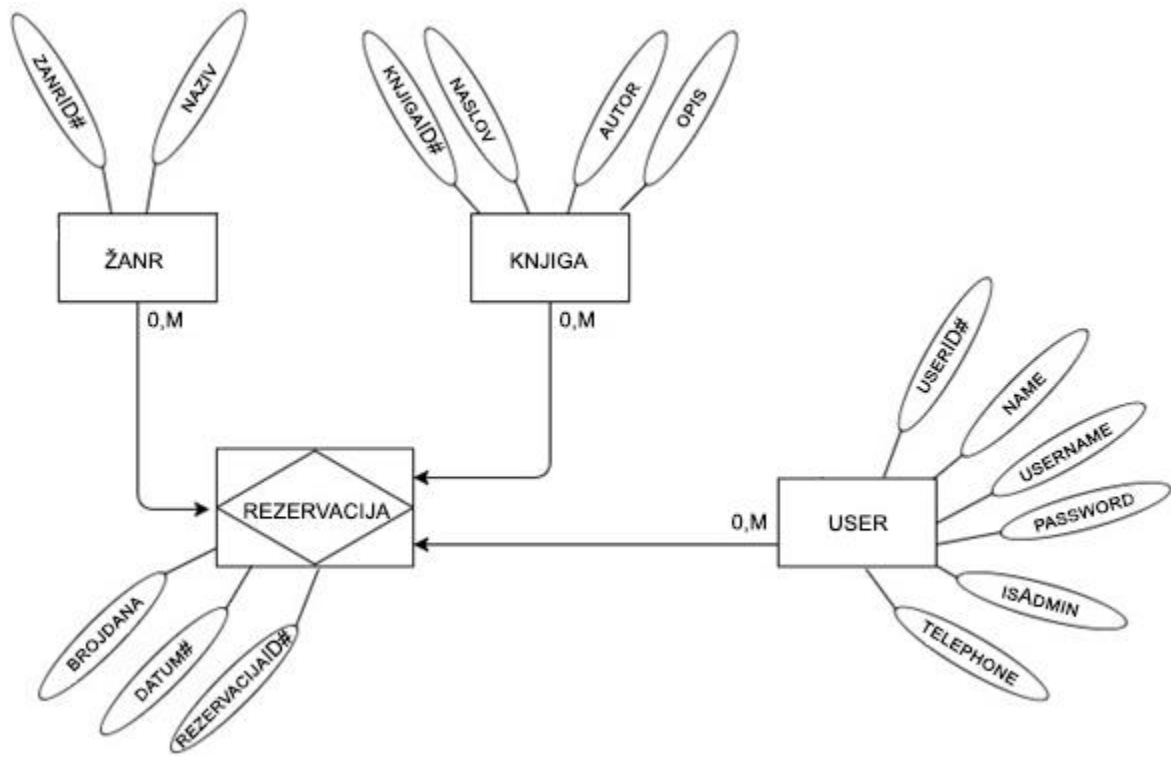


Alternativni scenario

- 2.1. Ako sistem ne može da zapamti podatke o korisniku, prikazuje poruku „Neuspešna registracija korisnika.“(IA)



2.4. Model podataka



Slika 4. Prošireni model objekti-veze

2.5. Specifikacija REST API-ja

Za kreiranje sopstvenog servisa korišćen je framework Flight. Servis ima 4 funkcije koje omogućavaju učitavanje liste žanrova i liste knjiga iz baze, kao i unošenje nove knjige i novog korisnika.

Funkcija vratiZanrove()

Opis funkcije	Vraćanje svih žanrova
HTTP metoda	GET
URL	/zanrovi.json
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	JSON objekat sa atributima zanrID[int], naziv[string]. Primer: <pre>{ "zanrID": "1", "naziv": "Istorija" }</pre>
Format izlaznih parametara	application/json

Funkcija vratiKnjige()

Opis funkcije	Vraćanje svih knjiga
HTTP metoda	GET
URL	/knjige.json
URL parametri	(nema)
HTTP body parametri	(nema)
Format HTTP body parametara	(nema)
Izlazni parametri	<p>JSON objekat sa atributimaknjigaID[int], naslov [string], autor[string], opis[string].</p> <p>Primer:</p> <pre>{ "knjigaID": "1", "naslov": "Na Drini cuprija", "autor": "Ivo Andric", "opis": "asdasdasd asdasdasdasdasd" }</pre>
Format izlaznih parametara	application/json

Funkcija ubaciKorisnika(data)

Opis funkcije	Dodavanje novog korisnika
HTTP metoda	POST
URL	/ubaciKorisnika.json
URL parametri	(nema)
HTTP body parametri	<p>JSON Objekat sa atributima userID[int], name[string], username[string], password[string], isAdmin[int], telephone[string].</p> <p>Primer:</p> <pre>{ "userID": "8", "name": "Jovana Milovanovic", "username": "jovanam", "password": "jovanam123" "isAdmin": "0" "telephone": "06251685261" }</pre>
Format HTTP body parametara	application/json
Izlazni parametri	<p>JSON objekat sa jednim atributom, poruka[string].</p> <p>Primer:</p> <pre>{ "poruka": "Uspesno!" }</pre>
Format izlaznih parametara	application/json

Funkcija novaKnjiga(data)

Opis funkcije	Dodavanje nove knjige
HTTP metoda	POST
URL	/knjige.json
URL parametri	(nema)
HTTP body parametri	<p>JSON objekat sa atributima knjigaID[int], naslov[string], autor[string], opis[string].</p> <p>Primer:</p> <pre>{ "knjigaID": "1", "naslov": "Na Drini cuprija", "autor": "Ivo Andric", "opis": "asdasdasd asdasdasdasdasd" }</pre>
Format HTTP body parametara	application/json
Izlazni parametri	<p>JSON objekat sa jednim atributom, poruka[string].</p> <p>Primer:</p> <pre>{ "poruka": "Uspesno!" }</pre>
Format izlaznih parametara	application/json

3. Opis korišćenih tehnologija

Za izradu korišćene su standardne tehnologije za izradu web tehnologija poput HTML-a, CSS-a, JavaScript-a, Bootstrap-a i PHP-a.

HTML, HyperText Markup Language, je jezik koji služi za struktuiranje tekstova, medija i ugrađenih objekata u web stranice. Sastoji se od teksta i tagova koji omogućavaju da internet pretraživač prepozna strukturu dokumenta i ispravno je prikaže korisniku.

CSS, Cascading Style Sheets, omogućava razdvajanje strukture stranice od njenog izgleda. CSS fajl treba da sadrži opis izgleda stranice i način na koji će informacije biti predstavljene.

Bootstrap je front-end framework koji služi za kreiranje web aplikacija. Sadrži HTML i CSS šablone za dizajn, kao i JavaScript ekstenzije. Jedna od najvažnijih uloga jeste omogućavanje responzivnosti sajta.

JavaScript je programski jezik HTML-a i weba. Bitan je deo web aplikacija jer omogućava interaktivnost aplikacija. JavaScript je skriptni jezik, odnosno sastoji se od niza komandi koje se očitavaju u interpretoru, bez prethodnog kompajliranja, i podržava objektno-orijentisano programiranje.

PHP je skriptni, open-source jezik koji služi za izradu dinamičkog web sadržaja i serverske strane aplikacije. Dozvoljava interakciju sa velikim brojem relacionih baza, kao što su MySQL, Oracle, itd. Poput JavaScripta, ne kompajlira se, već se interpretira pri svakom izvršavanju. Bazira se na proceduralnom programiranju ali omogućava i korišćenje klasa i drugih koncepata objektno-orijentisanog programiranja.

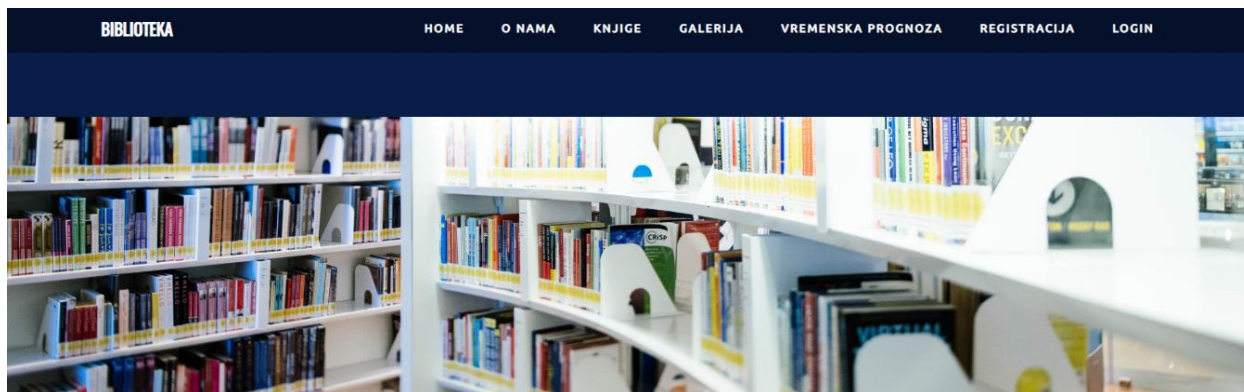
Pored ovih korišćene su i AJAX tehnologije i javni API-ji, kao i REST API za sopstveni servis.

AJAX, Asynchronous JavaScript And XML, predstavlja razvojnu tehniku za kreiranje interaktivnih web sadržaja. Omogućava asinhronnu komunikaciju između servera i korisnika. Zasniva se na razmeni male količine podataka između servera, čime dopušta korisniku konstantno korišćene aplikacije, bez čekanja na odgovor servera.

REST web servisi: REST (Representational State Transfer) predstavlja stil arhitekture aplikacije, koji se zasniva na često korišćenim tehnologijama i protokolima, od kojih je najvažniji HTTP u kombinaciji sa URL-om. Koristi osnovne HTTP-ove metode kao što su GET, POST, PUT, DELETE. Osnovni koncept REST-a je resurs, koji može biti informacija, document, slika, itd. Svaki resurs ima svoj identifikator, koji je URL, pomoću koga se pristupa i manipuliše tim resursom. Pri kreiranju sopstvenog REST servisa korišćen je framework Flight. Takođe korišćeni su dostupni API-ji za prikazivanje vremenske prognoze i generisanje lozinke pri registraciji novog korisnika.

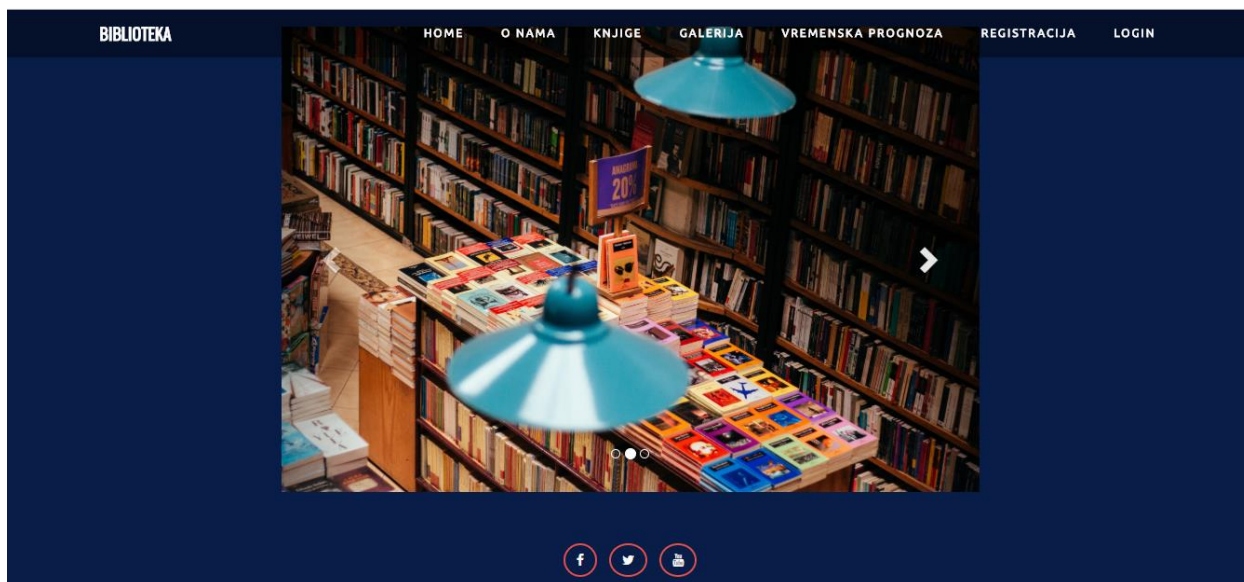
4. Korisničko uputstvo

Pri učitavanju aplikacije prikazuje se početna strana na kojoj se nalazi meni, odakle korisnik može izabrati karticu O NAMA gde može pogledati dodatne informacije, karticu KNJIGE gde može videti sve knjige koje postoje u biblioteci i mogu se rezervisati, karticu GALERIJA gde može pogledati slike biblioteke, karticu VREMENSKA PROGNOZA gde dobija informacije za Beograd za narednih 10 dana, karticu REGISTRACIJA gde se može prijaviti novi korisnik i karticu LOGIN gde se prijavljuje postojeći korisnik.



Slika 5. Meni

Na početnoj strani se nalazi i slajder koji sadrži tri slike koje se menjaju automatski ili kada korisnik pritisne strelice koje se nalaze sa leve i desne strane trenutne slike. Ispod slajdera nalaze se ikonice koje vode na društvene mreže biblioteke, Facebook, Twitter i Youtube.



Slika 6. Slajder i društvene mreže

Na početnoj strani se takođe nalazi i ponuda biblioteke u kojoj se nalaze knjige, galerija i registracija, a klikom na bilo koju od stavki, korisnik se preusmerava na odgovarajuću stranicu. Ovaj deo je sastavni deo svake stranice.



Slika 7. Ponuda

Na kraju početne strane, kao i svih ostalih, nalazi se footer koji sadrži podatke o kontaktu (email, broj, skype), kao i adresi biblioteke.



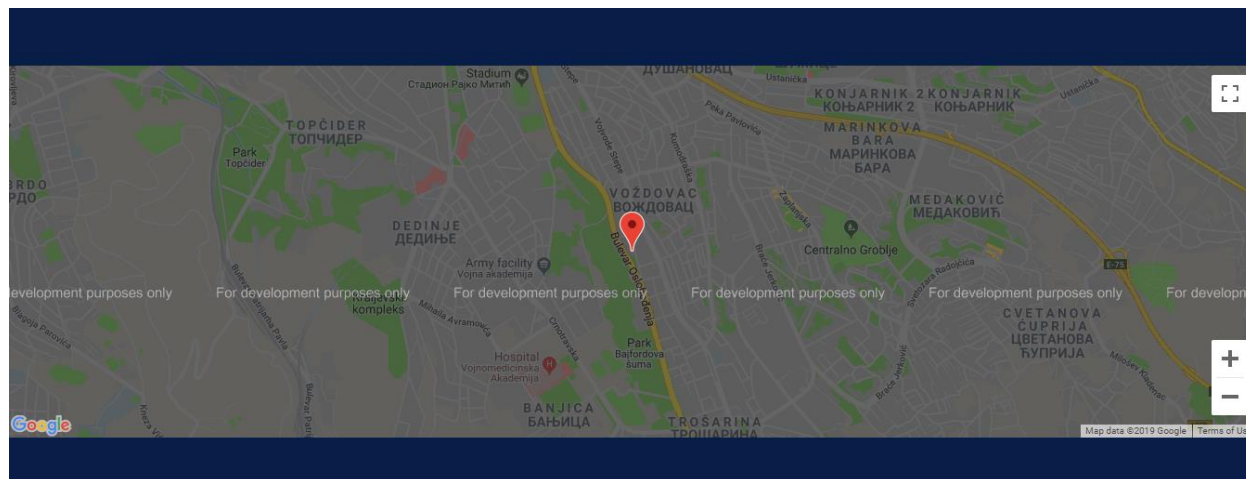
Slika 8. Footer

Na stranici O nama nalaze se osnovni podaci o biblioteci, uključujući istorijat, matičnu delatnost, dokumente, projekte i prostore.



Slika 9. Stranica O nama

U donjem delu stranice se nalazi mapa koja prikazuje lokaciju biblioteke.



Slika 10. Mapa

Na stranici Knjige nalazi se spisak koji sadrži sve knjige koje se nalaze u biblioteci i koje je moguće rezervisati. Spisak se sastoji od naslova dela, autora i kratkog opisa.

KNJIGE		
SHOW 10 ENTRIES	SEARCH: <input type="text"/>	
NASLOV	AUTOR	OPIS
BLAGO CARA RADOVANA	JOVAN DUCIC	QEWQEWQEWQWE
HAZARSKI RECNIK	MILORAD PAVIC	ASADSFGVCXSAGHNBFDSTJKHMBVCDSA
NA DRINI CUPRIJA	IVO ANDRIC	ASDASDASD ASDASDASDASDASD
NECISTA KRV	BORA STANKOVIC	RWEWERGFDG DFGDFGDF
RANI JADI	DANILO KIS	CBVCVB CBVCVCBVCVB
SEOBE	MILOS CRNJANSKI	CVXCVXCVXCVXCVXCVX
TVRDIJAVA	MESA SELIMOVIC	QWEQWEQWEQWE
SHOWING 1 TO 7 OF 7 ENTRIES		PREVIOUS 1 NEXT

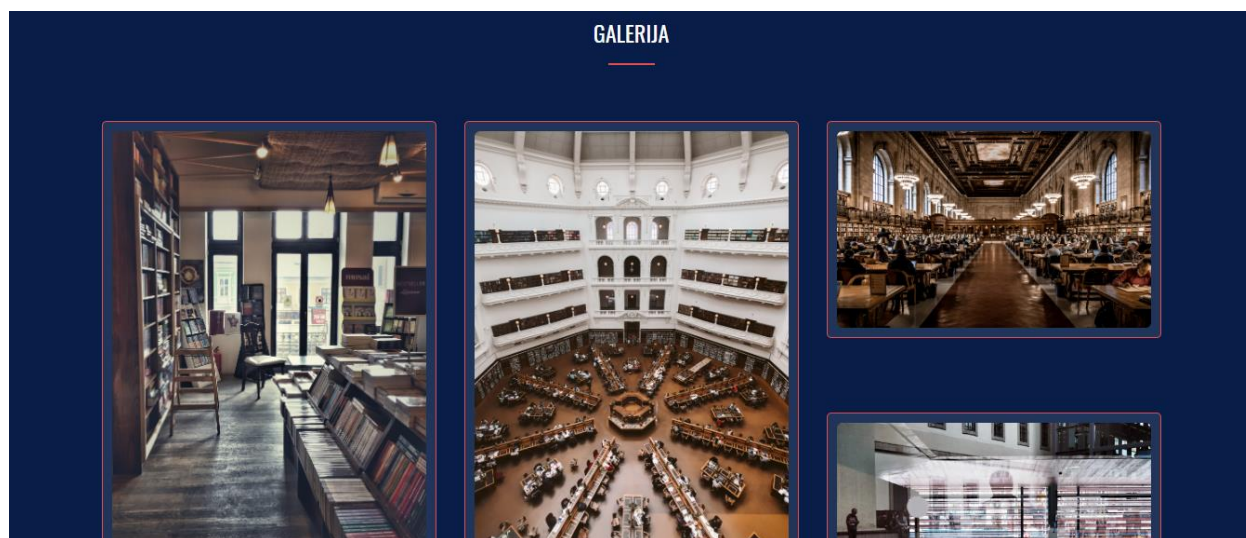
Slika 11. Knjige

Moguće je izvršiti pretragu knjiga u polju “Search”. Na primer, kucanjem “na” u tom polju, spisak se smanjuje i ostaju samo dve knjige, a to su “Blago cara Radovana” i “NA Drini ćuprija”.

KNJIGE		
SHOW 10 ENTRIES	SEARCH: na	
NASLOV	AUTOR	OPIS
BLAGO CARA RADOVANA	JOVAN DUCIC	QEWQEWQEWQWE
NA DRINI CUPRIJA	IVO ANDRIC	ASDASDASD ASDASDASDASDASD
SHOWING 1 TO 2 OF 2 ENTRIES (FILTERED FROM 7 TOTAL ENTRIES)		PREVIOUS 1 NEXT

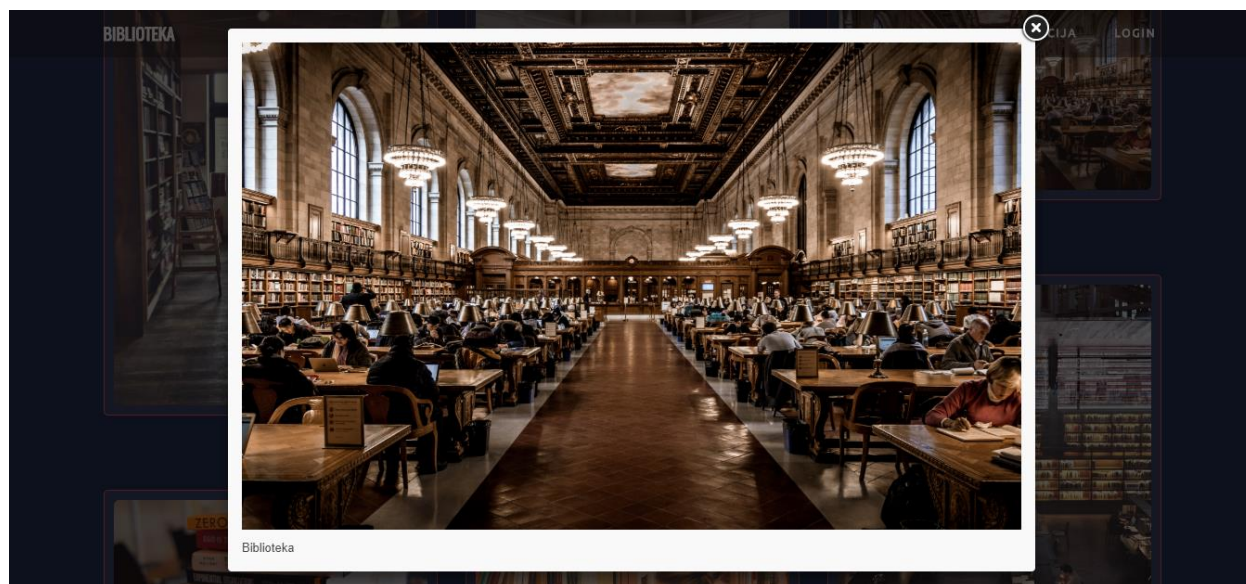
Slika 12. Pretraga knjiga

Na stranici Galerija nalaze se slike biblioteke.



Slika 13. Galerija

Izborom neke od slika iz galerija, ta slika se prikazuje u većem formatu.



Slika 14. Slika biblioteke

Na stranici Vremenska prognoza nalazi se vremenska prognoza za Beograd za narednih 10 dana.



Slika 15. Vremenska prognoza

Otvaranjem stranice Registracija, prikazuje se forma za registrovanje novih korisnika, gde je potrebno uneti ime i prezime, korisničko ime, lozinku i broj telefona. Lozinku nije neophodno uneti, već je moguće pritisnuti dugme „Generiši lozinku“ i lozinka će biti automatski kreirana.

REGISTRACIJA

IME I PREZIME

KORISNIČKO IME

LOZINKA

BROJ TELEFONA

Slika 16. Registracija

Na stranici Login se takođe nalazi forma, ali za prijavljivanje postojećih korisnika.



LOGIN FORMA

KORISNIČKO IME

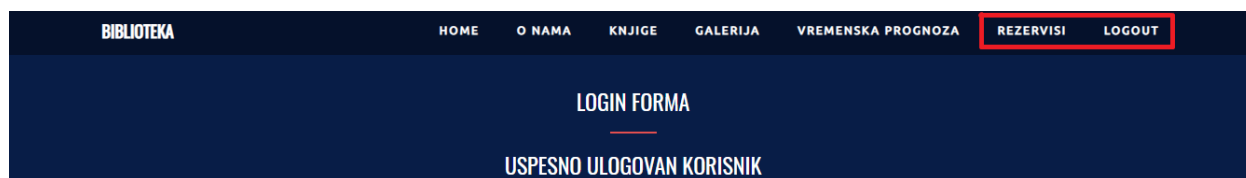
Lozinka

Lozinka

Login

Slika 17. Login forma

Nakon uspešnog logovanja prikazuje se poruka “uspešno ulogovan korisnik” i pojavljuju se dve nove opcije na meniju – Rezerviši i Logout.



BIBLIOTEKA

HOME

O NAMA

KNJIGE

GALERIJA

VREMENSKA PROGNOZA

REZERVISI

LOGOUT

LOGIN FORMA

USPEŠNO ULOGOVAN KORISNIK

Slika 18. Uspešno logovanje

Na stranici Rezerviši korisnik bira koju knjigu od postojećih želi da rezerviše, unosi datum rezervacije i broj dana na koji želi rezervaciju.



REZERVACIJA

KNJIGA

Na Drini cuprija

DATUM

Datum

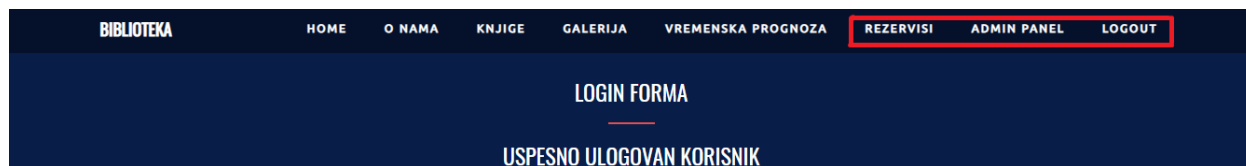
BROJ DANA

Broj dana

Rezerviši

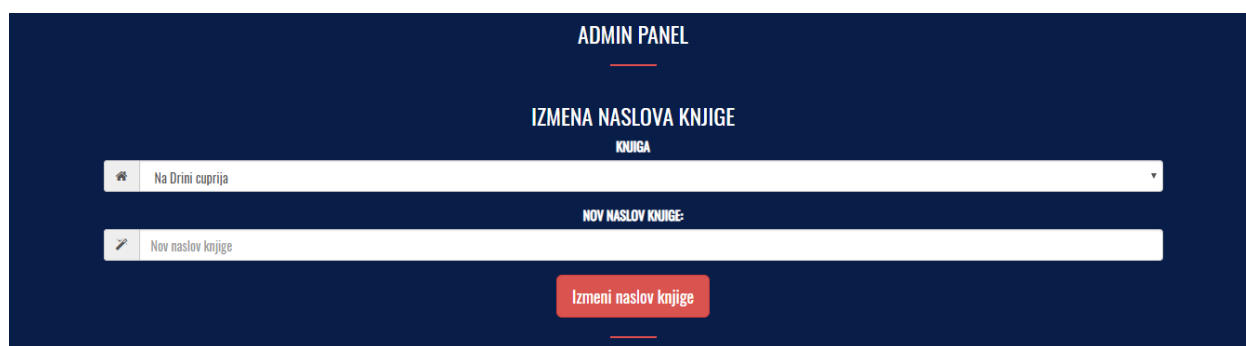
Slika 19. Rezervacija

Kada se uloguje korisnik koji je admin, njemu se u meniju pored opcija Rezerviši i Logout koje se pojavljuju svim korisnicima, pojavljuje i opcija Admin panel.



Slika 20. Logovanje admina

Na stranici Admin panel postoji više opcija koje admin može da uradi. Prva od njih je izmena naslova knjige izborom postojeće i unošenjem novog naslova.



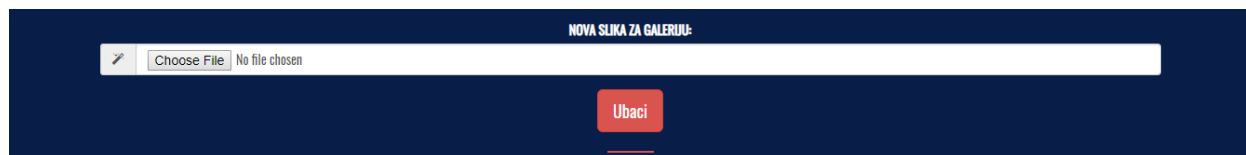
Slika 21. Izmjena naslova knjige

Admin ima uvid u sve rezervacije po knjigama, tj. koji korisnik je kada uzeo određenu knjigu i na koliko dana. On takođe može i da obriše rezervaciju.



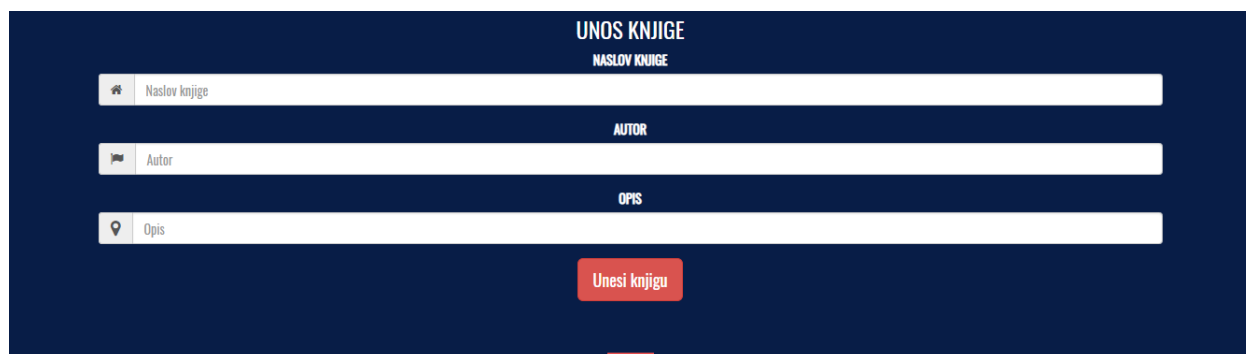
Slika 22. Pregled rezervacija po knjizi

Još jedna od mogućnosti admina je unos nove slike koja će biti dodata u galeriju.



Slika 23. Unos slike za galeriju

Admin takođe može da unosi nove knjige tako što će najpre uneti naslov knjige, a zatim autora i kratak opis.



UNOS KNJIGE

NASLOV KNJIGE

Naslov knjige

AUTOR

Autor

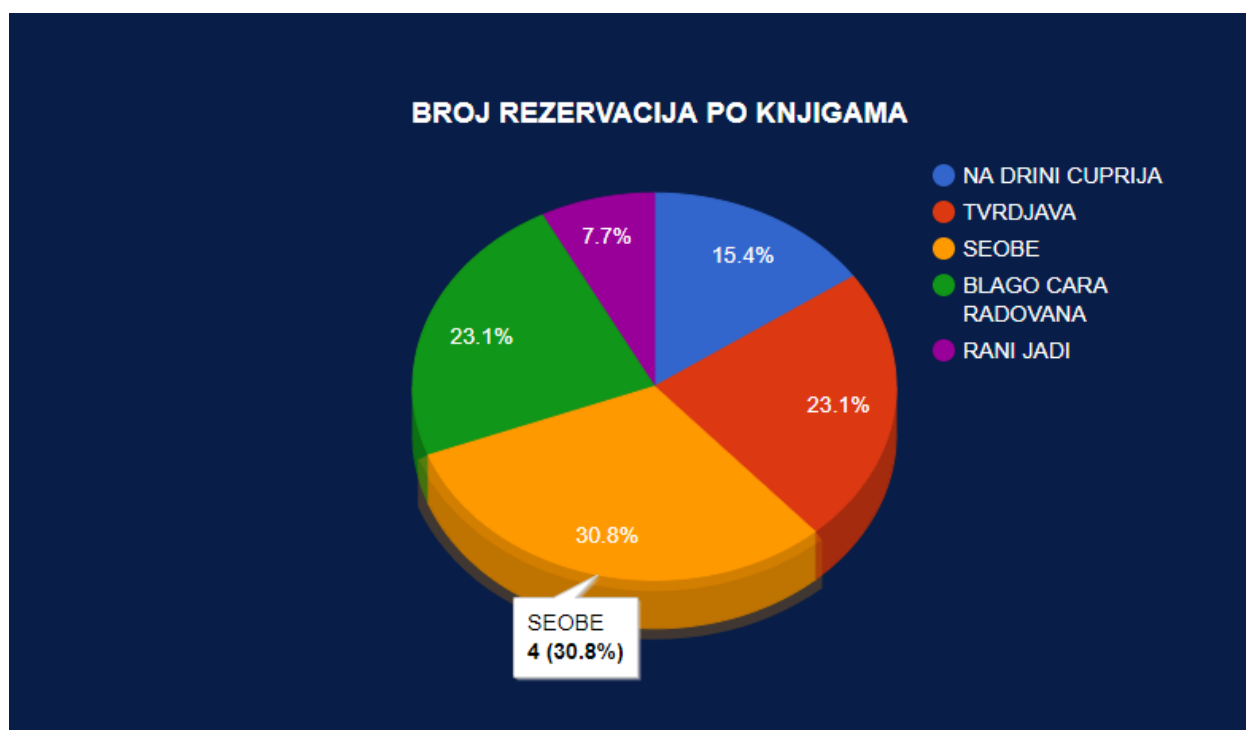
OPIS

Opis

Unesi knjigu

Slika 24. Unos knjige

Adminu se prikazuje grafik u obliku pite, na kom se procentualno vide rezervacije za svaku knjgu. Na primer, knjgu Seobe je rezervisalo 4 korisnika, što predstavlja 30.8% od svih rezervacija.



Slika 25. Broj rezervacija po knjigama

5. Reprezentativni delovi koda

Prikaz svih knjiga

Za prikaz knjiga korišćeni su PHP fajlovi knjige.php koji prikazuje podatke, init.php koji omogućava konekciju na bazu odakle se čitaju podaci i knjigaClass.php koji predstavlja klasu i sadrži attribute knjige i funkcije.

Deo koda knjige.php fajla za prikaz podataka:

```
<?php
    include("knjigaClass.php");
    $knjiga = new Knjiga($mysqli);
    $knjiga->allBooks();
    $result = $knjiga->getResult();

    if(count($result) >0) {
?>
<table class="table">
  <thead>
    <tr>
      <th class="text-center">Naslov</th>
      <th class="text-center">Autor</th>
      <th class="text-center">Opis</th>
    </tr>
  </thead>
  <tbody>
?>
    foreach ($result as $red ) {
      $naslov = $red['naslov'];
      $autor = $red['autor'];
      $opis = $red['opis'];
?>
    <tr style="color: #fff; background: #081d47;">

      <td><?php echo $naslov; ?></td>
      <td><?php echo $autor; ?></td>
      <td><?php echo $opis; ?></td>
    </tr>
```

```
<?php
```

```
}
```

```
?>
```

Takođe je korišćen DataTables za prikaz podataka u tabeli i pretragu, koji sadrži ujedno i AJAX tehnologiju.

```
<script type="text/javascript" src="https://cdn.datatables.net/v/dt/dt-1.10.16/datatables.min.js"></script>
```

```
<script>
```

```
$(document).ready(function() {
```

```
    $('#table').DataTable();
```

```
});
```

```
</script>
```

```
</body>
```

init.php fajl za konekciju i pokretanje sesije

```
<?php
```

```
error_reporting(E_ALL | E_STRICT);
```

```
ini_set("display_errors", 0);
```

```
ini_set("log_errors", 1);
```

```
ini_set("error_log", "logs.log");
```

```
//instace of mysqli
```

```
$mysqli = mysqli_connect("localhost", "root", "", "biblioteka");
```

```
//sesija korisnik sistema, inicijalizovanje user sesije
```

```
session_start();
```

```
if(!isset($_SESSION['user'])){
```

```
    $_SESSION['user'] = array();
```

```
}
```

```
?>
```

knjigaClass.php, za prikaz svih knjiga poziva se funkcija allBooks()

```
<?php
```

```
class Knjiga {  
    private $conn;  
    private $result;  
    public function __construct($conn) {  
        $this->conn = $conn;  
    }  
    public function getResult(){  
        return $this->result;  
    }  
    public function setResult($res){  
        $this->result = $res;  
    }  
    public function allBooks() {  
        $curl_zahtev = curl_init("http://localhost/biblioteka/api/knjige.json");  
        curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);  
        $curl_odgovor = curl_exec($curl_zahtev);  
        $json_objekat=json_decode($curl_odgovor, true);  
        curl_close($curl_zahtev);  
        $this->setResult($json_objekat);  
    }  
    public function allReservationBySearch($knjigaID) {  
        $knjigaID = mysqli_real_escape_string($this->conn,$knjigaID);
```

```
$q = mysqli_query($this->conn, "SELECT * FROM rezervacija r join knjiga k on  
r.knjigaID = k.knjigaID join user u on u.userID = r.userID where r.knjigaID = $knjigaID");
```

```
$this->setResult($q);
```

```
}
```

```
public function rezervacija($knjigaID,$datum,$dani) {
```

```
$knjigaID = mysqli_real_escape_string($this->conn,$knjigaID);
```

```
$datum = mysqli_real_escape_string($this->conn,$datum);
```

```
$dani = mysqli_real_escape_string($this->conn,$dani);
```

```
$userID=$_SESSION["user"]["userID"];
```

```
$timestamp = date('Y-m-d H:i:s', strtotime($datum));
```

```
$sql = "INSERT INTO rezervacija (datum,userID,knjigaID, brojDana) VALUES  
('$timestamp',$userID,$knjigaID,'$dani')";
```

```
if(mysqli_query($this->conn, $sql)){
```

```
$this->setResult(true);
```

```
}else{
```

```
$this->setResult(false);
```

```
};
```

```
}
```

```
public function unosKnjige($naslov,$autor, $opis) {
```

```
$naslov = mysqli_real_escape_string($this->conn,$naslov);
```

```
$autor = mysqli_real_escape_string($this->conn,$autor);
```

```
$opis = mysqli_real_escape_string($this->conn,$opis);
```

```
$data = Array (
```

```
    "naslov" => $naslov,
```

```
    "autor" => $autor,
```

```
    "opis" => $opis
```

```
);
```

```

        $zaSlanje = json_encode($data);
        $curl_zahtev = curl_init("http://localhost/biblioteka/api/knjige.json");
        curl_setopt($curl_zahtev, CURLOPT_POST, TRUE);
        curl_setopt($curl_zahtev, CURLOPT_POSTFIELDS, $zaSlanje);
        curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);
        $curl_odgovor = curl_exec($curl_zahtev);
        $json_objekat=json_decode($curl_odgovor, true);
        curl_close($curl_zahtev);
        if($json_objekat == "Uspesno!") {
            $this->setResult(true);
        }
        else {
            $this->setResult(false);
        }
    }

    public function izmenaNaslova($knjigaID,$naslov) {
        $knjigaID = mysqli_real_escape_string($this->conn,$knjigaID);
        $naslov = mysqli_real_escape_string($this->conn,$naslov);
        if(mysqli_query($this->conn, "UPDATE knjiga SET naslov='$naslov' where
knjigaID=$knjigaID")){
            $this->setResult(true);
        }else{
            $this->setResult(false);
        }
    }
}

?>

```


Prikaz mape

Za prikaz mape koristi se prognoza.php fajl koji koristi javni API .

Deo koda onama.php koji omogućava korišćenje API-ja

```
<div id="map"></div>

<!-- Skripta za pokretanje mape i setovanje položaja -->

<script>

    function initMap() {

        var položaj = {lat: 44.772841, lng: 20.475185 };

        var mapa = new google.maps.Map(document.getElementById('map'), {

            zoom: 14,

            center: položaj

        });

        var marker = new google.maps.Marker({

            position: položaj,

            map: mapa

        });

    }

</script>

<!-- Oдавde se skida mapa sa google api-a -->

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=AlzaSyA4xClapNfTV8lONxfoc4svvKN0SukizIM&callb
ack=initMap">

</script>
```

Izmena naziva knjige

Za izmenu naziva potrebni su admin.php i knjigaClass.php.

Iz knjigaClass.php fajla pozivaju se funkcije allBooks() i izmenaNaslova(knjigaID,naslov).

Delovi koda admin.php za izmenu imena

```
<select class="form-control" name="knjiga" >
    <?php
        $knjiga = new knjiga($mysqli);
        $knjiga->allBooks();
        $result = $knjiga->getResult();

        foreach ($result as $red ) {
            $id = $red['knjigaID'];
            $naslov = $red['naslov'];

            ?>
            <option value="<?php echo $id;?>"><?php echo $naslov;?></option>
        <?php
    }
    ?>
</select>

if(isset($_POST['izmeni'])) {

    $knjiga = new Knjiga($mysqli);
    $knjiga->izmenaNaslova(trim($_POST['knjiga']),trim($_POST['naslov']));
    if($knjiga->getResult()){
        $msg="Uspesno izmenjen naslov knjige! ";
    }else{
        $msg="Neuspesno izmenjen naslov knjige. ";
    }
}
```

Unos nove knjige

Za unos knjige potrebni su takođe admin.php i knjigaClass.php fajl.

Iz knjigaClass.php fajla se poziva funkcija unosKnjige(naslov,autor,opis).

Deo admin.php fajla koji omogućava unos nove knjige

```
if(isset($_POST['unosKnjige'])) {  
  
    $knjiga = new Knjiga($mysqli);  
    $knjiga->  
>unosKnjige(trim($_POST['naslov']),trim($_POST['autor']),trim($_POST['opis']));  
    if($knjiga->getResult()){  
        $msg="Uspesno uneta nova knjiga!";  
    }else{  
        $msg="Neuspesno unosenje nove knjige.";  
    }  
}
```

Unos nove slike u galeriju

Za unos nove slike potrebni su admin.php, novaSlikaGalerija.php i galerija.php fajl koji služi za prikaz slika.

Deo koda admin.php

```
<form name="galerija" method="post" action="novaSlikaGalerija.php"  
enctype="multipart/form-data">  
    <div class="form-group">  
        <label for="file" class="cols-sm-2 control-label">Nova slika za galeriju:</label>  
        <div class="cols-sm-10">  
            <div class="input-group">  
                <span class="input-group-addon"><i class="fa fa-magic fa" aria-  
hidden="true"></i></span>  
                <input type="file" class="form-control" name="file" placeholder="Ubacite  
sliku za galeriju"/>  
            </div>  
        </div>  
        <div class="form-group ">  
            <input type="submit" name="file" class="btn btn-danger btn-lg " value="Ubaci">  
        </div>  
</form>
```

novaSlikaGalerija.php

```
<?php
    ini_set('display_errors', 1);

    include("init.php");
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {

        $name    = $_FILES['file']['name'];
        $tmpName  = $_FILES['file']['tmp_name'];
        $error    = $_FILES['file']['error'];
        $size     = $_FILES['file']['size'];
        $ext      = strtolower(pathinfo($name, PATHINFO_EXTENSION));

        $targetPath = dirname( __FILE__ ) . DIRECTORY_SEPARATOR. 'assets'.
        DIRECTORY_SEPARATOR. 'galerija' . DIRECTORY_SEPARATOR. $name;
        move_uploaded_file($tmpName,$targetPath);
        header( 'Location: galerija.php' );
        exit;
    }

?>
```

Prikaz grafika

Grafik treba procentualno da prikaže broj rezervacija svake knjige pojedinačno. Za to su korišćeni admin.php i podaciGrafik.php fajlovi kao i ajax tehnologija.

Deo fajla admin.php

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>

<script type="text/javascript">

google.load('visualization', '1', {'packages':['corechart']});
google.setOnLoadCallback(grafik);

function grafik() {

    var jsonData = $.ajax({
        url: "podaciGrafik.php",
        dataType:"json",
        async: false
```

```

}).responseText;

var data = new google.visualization.DataTable(jsonData);
var options = {'title':'Broj rezervacija po knjigama',
  backgroundColor: { fill:'transparent' },
    titleTextStyle: {
      textAlign: 'center',
      color: 'white',
      fontSize: 18},
    'width':800,
    'height':500,
    is3D:true,
    legend: {
      textStyle: {
        color: 'white'
      }
    },
};

var chart = new google.visualization.PieChart(document.getElementById('divGrafik'));
function selectHandler() {
    var selectedItem = chart.getSelection()[0];
    if (selectedItem) {
        alert( data.getValue(selectedItem.row,0));
    }
    google.visualization.events.addListener(chart, 'select', selectHandler);
    chart.draw(data, options);
}
</script>

```

podaciGrafik.php

```
<?php
```

```
include("init2.php");
```

```
    $array['cols'][] = array('label' => 'Knjiga','type' => 'string');
```

```
    $array['cols'][] = array('label' => 'Broj rezervacija po knjigama', 'type' => 'number');
```

```
    $sql="SELECT k.naslov, COUNT(r.rezervacijaID) AS Broj FROM knjiga k INNER JOIN  
rezervacija r ON k.knjigaID = r.knjigaID GROUP BY k.knjigaID";
```

```
    if (!$q=$mysqli->query($sql)){
```

```
        echo '{"greska":"Nastala je greška pri izvršavanju upita."}';
```

```
    exit();
```

```
    } else {
```

```
        if ($q->num_rows>0){
```

```
            $niz[] = array();
```

```
            while ($red=$q->fetch_object()){
```

```
                $array['rows'][] = array('c' => array( array('v'=>$red-  
>naslov),array('v'=>(int)$red->Broj)) );
```

```
            }
```

```
            $niz_json = json_encode ($array);
```

```
            print ($niz_json);
```

```
        } else {
```

```
            //ako nema rezultata u bazi
```

```
            echo '{"greska":"Nema rezultata."}';
```

```
        }
```

```
    }
```

```
?>
```

Registracija

Registracija omogućava pamćenje novog korisnika. Potrebni fajlovi su register.php, userClass.php koji predstavlja klasu User-a i lozinka.php koji omogućava korišćenje javnog API-ja za generisanje lozinke.

Deo register.php

```
$msg = "";

if(isset($_POST['register'])) {
    require('userClass.php');
    $user = new User($mysqli);
    $user->
    >registracijaKorisnika(trim($_POST['name']),trim($_POST['username']),trim($_POST['password']),trim($_POST['telephone']));
    if($user->getResult()){
        $msg="Uspesno registrovan korisnik";
    }else{
        $msg="Neuspesna registracija korisnika ";
    }
}
```

Deo koji omogućava korišćenje API-ja

<script>

```
$(document).ready(function() {

    $('#generisiLozinku').click(function() {
        $.ajax({
            url: 'lozinka.php',
            dataType: 'json',
            success: function(json) {

                $('#password').val(json);

            },
            error: function(json) {
                console.log(json,password + ' error');
            }
        });
    });
});
</script>
```

lozinka.php

```
<?php
    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL,
    "http://www.sethcardoza.com/api/rest/tools/random_password_generator");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

    $output = curl_exec($ch);

    curl_close($ch);

    echo json_encode($output);
?>
```

userClass.php

```
<?php

class User {

    private $conn;
    private $result;

    public function __construct($conn) {
        $this->conn = $conn;
    }
    public function getResult(){
        return $this->result;
    }
    public function setResult($res){
        $this->result = $res;
    }
    public function registracijaKorisnika($ime,$username,$password,$telephone) {
        $ime = mysqli_real_escape_string($this->conn,$ime);
        $username = mysqli_real_escape_string($this->conn,$username);
        $password = mysqli_real_escape_string($this->conn,$password);
        $telephone = mysqli_real_escape_string($this->conn,$telephone);
        $data = Array (
            "name" => $ime,
            "username" => $username,
```



```

        "password" => $password,
        "telephone"=> $telephone
    );

    $zaSlanje = json_encode($data);
    $curl_zahtev =
curl_init("http://localhost/hotel/api/ubaciKorisnika.json");
    curl_setopt($curl_zahtev, CURLOPT_POST, TRUE);
    curl_setopt($curl_zahtev, CURLOPT_POSTFIELDS, $zaSlanje);
    curl_setopt($curl_zahtev, CURLOPT_RETURNTRANSFER, 1);
    $curl_odgovor = curl_exec($curl_zahtev);
    $json_objekat=json_decode($curl_odgovor, true);
    curl_close($curl_zahtev);

    if($json_objekat == "Uspesno!") {
        $this->setResult(true);
    }
    else {
        $this->setResult(false);
    }

}

public function login($username,$password) {
    $username = mysqli_real_escape_string($this->conn,$username);
    $password = mysqli_real_escape_string($this->conn,$password);
    $q = mysqli_query($this->conn, "select * from user where
username='$username' and password='$password' limit 1");
    if(mysqli_num_rows($q)>0){
        while($red = mysqli_fetch_assoc($q)) {
            $_SESSION['user'] = $red;
        }
        $this->setResult(true);
    }else{
        $this->setResult(false);
    };
}

}

?>

```

Login

Omogućava prijavu korisnika koji postoje u bazi. Fajlovi koji su potrebni jesu login.php i userClass.php koji poziva funkciju login(username, password).

Deo koda login.php

```
$msg = "";
if(isset($_POST['login'])) {
    require('userClass.php');
    $user = new User($mysqli);
    $user->login(trim($_POST['username']),trim($_POST['password']));
    if($user->getResult()){
        $msg="Uspesno ulogovan korisnik";
    }else{
        $msg="Neuspesno login korisnika ";
    }
}
```

Prikaz svih rezervacija i brisanje rezervacije

Korišćeni su admin.php, knjigaClass.php fajl koji poziva funkciju allReservationBySearch(knjigaID) koji pronalazi rezervacije za određenu knjigu, generisiPodatkeSearch.php koji kreira klasu knjige i prikuplja podatke i obriši.php koji omogućava asinhrono brisanje rezervacije.

Deo admin.php fajla

```
<script>
    function search(){
        var event = $("#knjigaSearch").val();
        $.ajax({
            url: "generisiPodatkeSearch.php",
            data: "knjigaID="+event,
            success: function(result){
                var text = '<table
class="table"><thead><tr><th>Korisnik</th><th>Datum rezervacije</th><th>Broj
dana</th><th>Brisanje</th></tr></thead><tbody>';
                $.each($.parseJSON(result), function(i, val) {
                    text += '<tr>';
                    text += '<td>'+val.name+'</td>';
                    text += '<td>'+val.datum+'</td>';
                    text += '<td>'+val.brojdana+'</td>';
                    text += '<td><a
href="obrisi.php?id='+val.rezervacijaID+'">Obrisi</a></td>';
```

```

                                text += '</tr>';
                                });
                                text+='</tbody></table>';
                                $('#tabela').html(text);
                                });
                                }
</script>
<script>
                                $( document ).ready(function() {
                                search();
                                });
</script>

```

generisiPodatkeSearch.php

```

<?php
    include("init.php");
    require('knjigaClass.php');
    $knjigaID=$_GET['knjigaID'];
    $knjiga = new Knjiga($mysqli);
    $knjiga->allReservationBySearch($knjigaID);
    $result = $knjiga->getResult();
    $niz = array();
    $iterator = 0;
    while($red = mysqli_fetch_assoc($result)) {
        $niz[$iterator] = $red;
        $iterator++;
    }
    echo(json_encode($niz));
?>

```

obrisi.php

```

<?php
include("init.php");
$id=mysqli_real_escape_string($mysqli,$_GET["id"]);
if(mysqli_query($mysqli, "delete from rezervacija where rezervacijaID=$id")){
    header("Location: index.php");
}else{
    echo("Doslo je do greske prilikom brisanja");}; ?>

```

Sopstveni servis

database.php

```
<?php
class Database {
    private $hostname = "localhost";
    private $username = "root";
    private $password = "";
    private $dbname = "biblioteka";
    private $dblink;
    private $result = true;
    private $records;
    private $affectedRows;

    function __construct($dbname)
    {
        $this->$dbname = $dbname;
        $this->Connect();
    }

    public function getResult()
    {
        return $this->result;
    }

    function __destruct()
    {
        $this->dblink->close();
    }

    function Connect()
    {
        $this->dblink = new mysqli($this->hostname, $this->username, $this->password,
$this->dbname);
        if($this->dblink->connect_errno)
        {
            printf("Konekcija neuspesna: %s\n", $mysqli->connect_error);
            exit();
        }
        $this->dblink->set_charset("utf8");
    }
}
```

```

function novaKnjiga($data) {
    $mysqli = new mysqli("localhost", "root", "", "biblioteka");
    $naslov = mysqli_real_escape_string($mysqli,$data["naslov"]);
    $autor = mysqli_real_escape_string($mysqli,$data["autor"]);
    $opis = mysqli_real_escape_string($mysqli,$data["opis"]);
    $sql = "INSERT INTO knjiga (naslov, autor, opis) VALUES ('$naslov','$autor',
'$opis')";

    if($mysqli->query($sql))
    {
        $this->result = true;
    }
    else
    {
        $this->result = false;
    }
    $mysqli->close();
}

function ubaciKorisnika($data) {
    $mysqli = new mysqli("localhost", "root", "", "biblioteka");
    $cols = '(name, username, password, isAdmin, telephone)';

    $name = mysqli_real_escape_string($mysqli,$data['name']);
    $username = mysqli_real_escape_string($mysqli,$data['username']);
    $password = mysqli_real_escape_string($mysqli,$data['password']);
    $telephone = mysqli_real_escape_string($mysqli,$data['telephone']);

    $query = "INSERT into user (name, username, password, isAdmin,
telephone) VALUES ('$name','$username','$password','0','$telephone')";
    if($mysqli->query($query))
    {
        $this->result = true;
    }
    else
    {
        $this->result = false;
    }
    $mysqli->close();
}

function vratiKnjige() {
    $mysqli = new mysqli("localhost", "root", "", "biblioteka");
    $q = 'SELECT * FROM knjiga';

```

```

        $this->result = $mysqli->query($q);
        $mysqli->close();
    }

    function vratiZanrove() {
        $mysqli = new mysqli("localhost", "root", "", "biblioteka");
        $q = 'SELECT * FROM zanr';
        $this->result = $mysqli->query($q);
        $mysqli->close();
    }

    function ExecuteQuery($query)
    {
        if($this->result = $this->dblink->query($query)){
            if (isset($this->result->num_rows)) $this->records = $this->result-
>num_rows;
            if (isset($this->dblink->affected_rows)) $this->affected = $this->dblink-
>affected_rows;
            return true;
        }
        else{
            return false;
        }
    }
}
?>

```

index.php

```

<?php
require 'flight/Flight.php';
require 'jsonindent.php';

Flight::route('/', function(){
    echo('Popis ruta: <br>');
    echo('----- <br>');
    echo('GET zanrovi.json <br>');
    echo('GET knjige.json <br>');
    echo('POST ubaciKorisnika.json <br>');

```

```

        echo('POST knjige.json <br>');
    });
    Flight::register('db', 'Database', array('niz'));
    Flight::route('GET /zanrovi.json', function()
    {
        header("Content-Type: application/json; charset=utf-8");
        $db = Flight::db();
        $db->vratiZanrove();
        $niz = array();
        $iterator = 0;
        while ($red = $db->getResult()->fetch_object())
        {
            $niz[$iterator] = $red;
            $iterator += 1;
        }
        echo indent(json_encode($niz));
    });

```

```

    Flight::route('GET /knjige.json', function()
    {
        header("Content-Type: application/json; charset=utf-8");
        $db = Flight::db();
        $db->vratiKnjige();
        $niz = array();
        $iterator = 0;
        while ($red = $db->getResult()->fetch_object())
        {
            $niz[$iterator] = $red;

```

```

        $iterator += 1;
    }
    echo indent(json_encode($niz));
});

Flight::route('POST /ubaciKorisnika.json', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $post_data = file_get_contents('php://input');
    $json_data = json_decode($post_data,true);
    $db->ubaciKorisnika($json_data);
    if($db->getResult())
    {
        $response = "Uspesno!";
    }
    else
    {
        $response = "Greska!";
    }
    echo indent(json_encode($response));
});

```

```

Flight::route('POST /knjige.json', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $post_data = file_get_contents('php://input');

```



```
$json_data = json_decode($post_data,true);
$db->novaKnjiga($json_data);
if($db->getResult())
{
    $response = "Uspesno!";
}
else
{
    $response = "Neuspesno";
}
echo indent(json_encode($response));
});
```

```
Flight::start();
```

```
?>
```