

Univerzitet Crne Gore

**ELEKTROTEHNIČKI FAKULTET**

Studije primijenjenog računarstva

\*Softversko inženjerstvo: Dokumentacija mini-projekta\*

**Tema projekta: Biblioteka knjiga**

**Projekat radili:**

Nadja Radonjic 24/23

Bozidar Janjusevic 100/23

Viktor Stijepovic 89/23

Nikola Knezevic 67/23

Podgorica, --.06.2025. godine

---

# Dokumentacija projekta: Biblioteka knjiga

---

## Sadržaj

1. Izvještaj o problemu
2. Tehnologije
3. Funkcionalni zahtjevi
4. Use Case specifikacije, struktura I funkcionalnosti
5. Model domena
6. Dijagram toka
7. Organizacija koda
8. Moguće nadogradnje
9. Uputstvo za korisnike
10. Detaljna analiza GUI elemenata
- 11..Tok rada aplikacije
12. Analiza funkcionalnosti aplikacije
13. Test slcajevi I validacija funkcionalnosti
14. Tehnicki predlozi I mogucnost dodavanja sadrzaja(dopuna)
15. Zanimljivi djelovi koda
16. Zaključak

## 1. Izvještaj o problemu

Aplikacija “Biblioteka knjiga” simulira osnovne funkcionalnosti male biblioteke u vidu jednostavnog desktop GUI programa. Omogućava korisnicima da: - Pregledaju sve knjige - Pretražuju po naslovu i autoru - Pozajmljuju i vraćaju knjige - Sortiraju i filtriraju spisak - Sačuvaju i učitaju trenutno stanje

Problem koji rešava ovaj softver je ručno upravljanje evidencijom knjiga u malim bibliotekama, učionicama ili čitaonicama. Digitalna verzija značajno olakšava evidenciju dostupnosti, kao i pregled i statistiku.

---

## 2. Tehnologije

- **Python 3** – osnovni programski jezik
  - **customtkinter** – biblioteka za pravljenje modernog korisničkog interfejsa
  - **json** – za serijalizaciju i deserializaciju stanja
  - **os** – za rad sa fajlovima i proveru postojanja fajlova
  - **Tkinter (osnova)** – za grafički prikaz, nasleđena u customtkinter-u
- 

## 3. Funkcionalni zahtjevi

ID	Prioritet	Zahtjev
REQ1	5	Korisnik može pretraživati knjige po naslovu ili autoru
REQ2	5	Prikaz sve literature u biblioteci
REQ3	4	Mogućnost pozajmljivanja knjiga
REQ4	4	Mogućnost vraćanja knjiga
REQ5	3	Filtriranje dostupnih/pozajmljenih knjiga
REQ6	2	Sortiranje knjiga po naslovu ili autoru
REQ7	3	Statistika prikaza knjiga
REQ8	5	Čuvanje i učitavanje stanja biblioteke
REQ9	2	Pregled rezultata kroz tekstualno polje (log aktivnosti)

---

## 4. Struktura i funkcionalnosti

### 4.1 Glavni podaci:

- `knjige_autori` – rečnik sa knjigama i autorima
- `status_knjige` – status knjige (dostupna / pozajmljena)

- `trenutno_prikazane_knjige` – lista trenutno prikazanih knjiga (npr. rezultat pretrage)

#### 4.2 GUI komponente:

- Unosni prozor za pretragu
- Dugmad za pretragu, prikaz svih knjiga, sortiranje, filtriranje, pozajmljivanje, vraćanje
- Tekstboks za ispis rezultata i obaveštenja

#### 4.3 Funkcionalnosti:

- **Prikaz knjiga:** prikazuje sve knjige sa statusom
- **Pretraga:** prema nazivu ili autoru
- **Pozajmljivanje knjiga:** kroz novi prozor sa čekboksovima za odabir
- **Vraćanje knjiga:** slično kao pozajmljivanje
- **Sortiranje:** po naslovu ili autoru
- **Filtriranje:** dostupne / pozajmljene
- **Statistika:** ukupno, dostupne i pozajmljene knjige
- **Sačuvaj/Učitaj stanje:** snima se u fajl `stanje_biblioteke.json`

---

## 5. Pokretanje aplikacije i organizacija

Instalirati biblioteku `customtkinter`:

```
pip install customtkinter
```

Pokrenuti Python fajl:

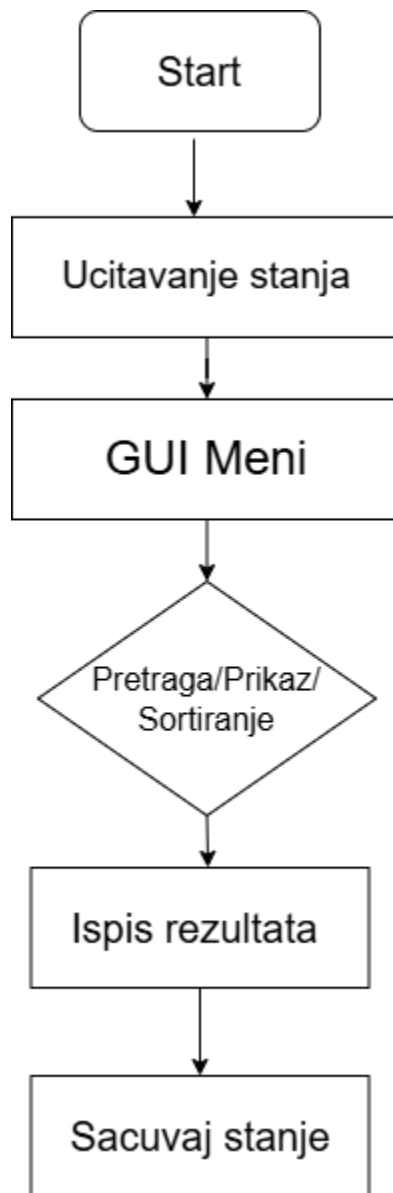
```
python biblioteka.py
```

---

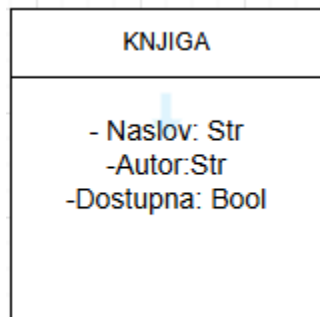
### Organizacija koda

- Sve funkcije su definisane u glavnom fajlu.
  - Globalne promenljive služe za rad sa prikazanim knjigama.
  - Interfejs se gradi korak po korak uz jasno definisane komponente i dugmad.
  - Funkcije su modularne i imaju logične nazive: `prikazi_knjige`, `pretrazi_knjigu`, `sortiraj_po_naslovu`, `otvori_pozajmljivanje_prozor`, itd.
-

## 6. Dijagram toka



## UML Klasa



## 7. Organizacija koda (dopuna)

Kod je pisan modularno, svaka funkcionalnost je u posebnoj funkciji. GUI elementi su grupisani u "frame" kontejnere, što olakšava snalaženje.

U tekstualni deo interfejsa se ispisuju poruke korisniku: - Šta je pozajmljeno - Koji su rezultati pretrage - Statistika biblioteke - Obaveštenja o učitavanju i čuvanju stanja

---

## 8. Moguće nadogradnje

- Dodavanje baze podataka umesto JSON fajla
- Autentikacija korisnika
- Evidencija ko je pozajmio knjigu
- Višekorisnička podrška
- Automatizovano testiranje GUI elemenata
- Sinhronizacija sa online bazom
- Uvođenje administratorskog panela
- Istorija pozajmica i datum vraćanja
- Automatsko obaveštavanje korisnika (e-mail/sistem poruke)

---

## 9. Uputstvo za korisnike

- Pokrenite aplikaciju
- U polje za unos unesite naslov knjige ili autora za pretragu
- Kliknite na dugme "Pozajmi" da biste odabrali dostupne knjige
- Kliknite na "Vrati" za vraćanje pozajmljenih knjiga
- Koristite "Sortiraj" i "Filtriraj" dugmad za dodatnu organizaciju

- Sačuvajte ili učitajte stanje po potrebi

Napomena: Sve funkcionalnosti su jasno naznačene na interfejsu kroz intuitivne nazive dugmadi i polja. Nije potrebno prethodno znanje iz oblasti biblioteka da bi se aplikacija koristila.

## 10. Detaljna analiza GUI elemenata

Aplikacija koristi CustomTkinter biblioteku za izradu modernog grafičkog korisničkog interfejsa. GUI je podijeljen u tri glavna panela:

- Lijevi meni: sadrži sve kontrole za pretragu, sortiranje, pozajmicu, vraćanje knjiga i statistiku.
- Centralni prikaz: prikazuje knjige (naslov, autor, status).
- Desni panel: omogućava prijavu i registraciju korisnika.

Korišćeni elementi uključuju: dugmad (CTkButton), tekstualna polja (CTkEntry), checkboxove za selekciju knjiga, i scrollable okvire za prikaz velikog broja knjiga.

## 11. Tok rada aplikacije

1. Pokretanje aplikacije kreira glavni prozor sa svim GUI komponentama.
2. Korisnik se prijavljuje ili registruje.
3. Po prijavi, dobija se pristup svim funkcijama biblioteke.
4. Pretraga i prikaz knjiga funkcioniše u realnom vremenu kroz filtriranje rečnika.
5. Pozajmljivanje i vraćanje knjiga otvara novi prozor sa checkbox izborom.
6. Stanje knjiga se ažurira odmah i prikazuje korisniku.
7. Statistika daje broj dostupnih i pozajmljenih knjiga.

## 12. Analiza funkcionalnosti aplikacije

- login(): Provjerava korisničko ime i lozinku i prijavljuje korisnika.
- signup(): Registruje novog korisnika i čuva podatke u JSON fajlu.
- prikazi\_knjige(): Prikazuje sve knjige sa statusima.
- pretrazi\_knjigu(): Omogućava pretragu po naslovu ili autoru.
- pozajmi\_knjige\_dialog(): Dijalog prozor za pozajmljivanje knjiga.
- vrati\_knjige\_dialog(): Dijalog za vraćanje knjiga.
- sortiraj\_naslov()/autor(): Sortiraju knjige po izabranom kriterijumu.
- prikazi\_statistiku(): Prikazuje ukupan broj, broj dostupnih i broj pozajmljenih knjiga.

## 13 Test slučajeви i validacija funkcionalnosti

REQ1 – Pretraga knjiga:

- Unijeti pojam u pretragu i verifikovati prikaz rezultata.

REQ3 – Pozajmljivanje:

- Prijaviti se, izabrati knjigu i potvrditi pozajmicu. Provjeriti da se status mijenja.
- REQ4 – Vraćanje:
- Prijaviti se, izabrati pozajmljenu knjigu, vratiti i provjeriti status.
- REQ8 – Čuvanje/učitavanje:
- Promijeniti stanje, restartovati aplikaciju i učitati JSON fajl.

## 14. Tehnički prijedlozi za nadogradnju(dopuna i vraćanje na prosle prijedloge)

- Čuvanje evidencije po korisniku (ko je pozajmio koju knjigu).
- Implementacija datuma pozajmice i očekivanog vraćanja.
- Dodavanje e-mail notifikacija o isteku roka.
- Upotreba SQLite baze umjesto JSON za bolju skalabilnost.
- Implementacija administratorskog interfejsa.

## 15. Zanimljivi djelovi koda

### 1. Pozajmljivanje knjiga – dinamički dijalog sa checkboxovima

Kod ispod prikazuje sve dostupne knjige kao checkboxove unutar novog dijaloga. Korisnik može izabrati više knjiga koje želi da pozajmi. Ove knjige se zatim označavaju kao pozajmljene i njihovo stanje se odmah ažurira u glavnom prikazu aplikacije.

```
for naslov, autor in knjige_autori.items():
    if status_knjige[naslov]:
        var = ctk.BooleanVar()
        check = ctk.CTkCheckBox(okvir, text=f"{naslov} -
{autor}", variable=var)
        check.pack(anchor="w", pady=2)
        check_vars[naslov] = var
```

### 2. Prikaz knjiga sa statusom dostupnosti

Ova funkcija centralno kontroliše prikaz knjiga u aplikaciji. Prvo briše postojeći prikaz, zatim redom prikazuje sve knjige uz ikonicu dostupnosti: ☐ za dostupne i ☐ za pozajmljene. Prikaz se koristi u gotovo svim funkcijama: prikazu, pretrazi, sortiranjima, itd.

```
def prikazi_knjige(knjige=None):
    for widget in knjige_list_frame.wininfo_children():
        widget.destroy()

    prikaz = knjige or knjige_autori
    for naslov, autor in prikaz.items():
        status = "☐ " if status_knjige[naslov] else "☐ "
```



```
        label = ctk.CTkLabel(knjige_list_frame, text=f"{naslov}
- {autor} ({status})", anchor="w")
        label.pack(fill="x", padx=10, pady=2)
```

### 3. Pretraga knjiga po nazivu ili autoru

Pretraga funkcioniše tako što korisnik unese tekstualni upit, a sistem traži sve naslove ili autore koji sadrže unijeti pojam. Pretraga je neosjetljiva na velika/mala slova. Ako se pronađu rezultati, prikazuju se kroz standardni prikaz knjiga.

```
def pretrazi_knjigu():
    upit = ctk.CTkInputDialog(title="Pretraga", text="Unesi
naziv knjige ili autora:").get_input()
    if upit:
        filtrirano = {k: v for k, v in knjige_autori.items() if
upit.lower() in k.lower() or upit.lower() in v.lower()}
        if filtrirano:
            prikazi_knjige(filtrirano)
        else:
            messagebox.showinfo("Rezultat", "Nema rezultata
pretrage.")
```

### 4. Sortiranje knjiga po autoru

```
def sortiraj_autor():
    sortirano = dict(sorted(knjige_autori.items(), key=lambda
item: item[1]))
    prikazi_knjige(sortirano)
```

#### **Opis:**

Funkcija koristi `sorted()` metod sa `lambda` funkcijom da sortira knjige po imenu autora. Rezultat se zatim prikazuje kroz `prikazi_knjige()`. Ova logika omogućava korisniku da lakše pronađe autora unutar velike kolekcije knjiga. Ovo je primjer upotrebe Python-ovih funkcionalnih metoda na jednostavnim strukturama podataka.

## 5. Prikaz statistike o knjigama

```
def prikazi_statistiku():
```

```
    ukupno = len(knjige_autori)
```

```
    dostupne = sum(1 for dostupna in status_knjige.values() if dostupna)
```

```
    pozajmljene = ukupno - dostupne
```

```
    messagebox.showinfo("□ Statistika", f"Ukupno knjiga: {ukupno}\nDostupne: {dostupne}\nPozajmljene: {pozajmljene}")
```

### Opis:

Kratka ali moćna funkcija koja korisniku pruža uvid u trenutno stanje biblioteke. Prikazuje broj svih knjiga, onih koje su slobodne za pozajmicu i onih koje su već izdate. Može se lako proširiti za statistiku po korisniku, vremenskim periodima itd.

## 6. Učitavanje korisnika iz JSON fajla

```
if os.path.exists(korisnici_fajl):
    with open(korisnici_fajl, "r") as f:
        korisnici = json.load(f)
else:
    korisnici = {}
```

### Opis:

Ovaj fragment omogućava aplikaciji da sačuva korisničke naloge čak i kada se aplikacija zatvori. Ako `korisnici.json` postoji, učitava se u memoriju; ako ne postoji, aplikacija počinje sa praznim skupom korisnika. To je osnovni mehanizam za rad sa trajnim podacima bez baze.

---

## 16. Zaključak

Biblioteka knjiga je jednostavna, ali moćna desktop aplikacija za simulaciju rada biblioteke. Idealna je za edukaciju, demonstraciju rada sa GUI-jem, i može biti osnov za složenije sisteme.

## 17. Literatura i izvori

- <https://customtkinter.tomschimansky.com/>
  - Python dokumentacija: <https://docs.python.org/3/>
  - Stack Overflow
  - Razni tutorijali na YouTube-u
-