

# Dokumentacija Task 2

- Članovi tima

1. Ahmed Raščić
2. Nejra Stršević
3. Emina Peljto
4. Danijel Mioković
5. Eman Hodžić

- Lista provjere koda

- ☐ Da li dati kod uspješno implementira opisanu funkcionalnost?
- ☐ Može li se kod pojednostaviti?
- ☐ Da li kod uvodi nove biblioteke ili međuovisnosti u sistem?
- ☐ Je li kod jednostavan za shvatiti?
- ☐ Koji dijelovi koda su zbunjujući i teški za shvatiti?
- ☐ Je li kod na pravilnom nivou apstrakcije?
- ☐ Mogu li se poboljšati održivost, performans i sigurnost koda?
- ☐ Može li se poboljšati čitljivost koda?
- ☐ Da li se kod pridržava svih SOLID principa?

- Tabela ozbiljnosti

Korištena je ista tabela ozbiljnosti kao što je prikazana na predavanju.

Ozbiljnost	Opis
5 (kritično)	(1) Sprječava postizanje osnovnih mogućnosti. (2) Ugrožava sigurnost, zaštitu i druge kritične zahtjeve.
4	(1) Nepovoljno utiče na postizanje osnovnih mogućnosti, kada rješenje kojim se greška može izbjeći nije poznato. (2) Nepovoljno utiče na tehnički, troškovni i rasporedni rizik projekta ili sistemskog održavanja, kada rješenje kojim se greška može izbjeći nije poznato.
3	(1) Nepovoljno utiče na postizanje osnovnih mogućnosti, ali je rješenje kojim se greška može izbjeći poznato. (2) Nepovoljno utiče na tehnički, troškovni i rasporedni rizik projekta ili sistemskog održavanja, ali je rješenje kojim se greška može izbjeći poznato.
2	(1) Korisnička/operatorska neudobnost koja ne utiče na zahtijevanu misiju ili operaterove osnovne mogućnosti. (2) Neugodnost za programere ili personal na održavanju, ali ne sprečava realizaciju ovih odgovornosti.
1 (minorno)	Bilo koji drugi efekat.

- Opis sesije pregleda

Sesija provjere se odvijala tako što je svaki član tima dobio dio implementacije za čiji pregled je bio zadužen. Moderator inspekcije (Ahmed Raščić) je kreirao odgovarajuće liste provjere kao i format izgleda izvještaja za pojedinačni pregled. Potom su članovi pregledali njima zadate segmente koda, popunili svoje liste provjere te individualne izvještaje.

Uporedo, koristili smo GitHub za funkcionalnosti dodavanja komentara na kod (tako što su se komentari dodavali na određene committe), kao i funkcionalnosti kreiranja issue kartica. Članovi su opisali pronađene greške uz dodijeljene stepene ozbiljnosti, te na osnovu toga generisali histograme pronađenih grešaka po ozbiljnosti

koje su priložili uz svoje izvještaje. Predlagali su korektivne akcije, te su iste implementirali i postavili na GitHub u vidu pull request-ova (zahtjeva za pregled) koje je moderator pregledao i potvrdio.

# Izvještaj o Inspekcijskoj Sesiji

**Datum:** 11.decembar.2023.

**Opis projekta:** Informacioni sistem za biblioteku

**Tim:** Ahmed Raščić (moderator), Nejra Stršević, Emina Peljto,  
Danijel Mioković, Eman Hodžić

## Sažetak pronađenih grešaka

Tip	Opis	Lokacija	Ozbiljnost
Stil koda	Sve rute nisu imenovane konzistentnim stilom	Imena ruta u UserController.cs	2
Ponavljanje koda	Ime i tip rute se može dati skupa	Ruta login i ostale	2
Čitljivost	Manjak opisnih i dokumentirajućih komentara	UserController.cs	1
Stil koda	Sve rute nisu imenovane konzistentnim stilom	Imena ruta u BookController.cs	2
Ponavljanje koda	Ime i tip rute se može dati skupa	Ruta addBook i ostale	2
Dizajn koda	Logika posuđivanja se treba odvojiti od same CRUD logike knjiga	Rute loanBook i getBook	3
Čitljivost	Manjak opisnih i dokumentirajućih komentara	BookController.cs	1
Upravljanje greškama	Greške se vraćaju kao odgovor	Rute loanBook i getBookBack	2
Upravljanje greškama	Dodavanje praćenja grešaka	BookController.cs, metode SearchBook i GetBookDetails	2
Čitljivost	Drugi način provjere da li je poslana null vrijednost	BookController.cs	1
Upravljanje greškama	Nije definisano šta će se desiti ukoliko se metodi proslijede parametri koji su null	BookController.cs metoda ApplyCategorySpecificBenefits	3
Čitljivost	Ponavljanje Price i Days atributa Loan modela unutar switch konstrukcije	ApplyCategorySpecificBenefits u BookController.cs	1

## Metrike

**Prosječna ozbiljnost greške: 1.8**

**Broj grešaka po metodi u UserController.cs: 0.44**

**Broj grešaka po metodi u BookController.cs: 0.28**

**Prosječan broj otkrivenih grešaka po recenzentu: 2.8**

# Izveštaj o pronađenim greškama

Datum: 2.decembar.2023.

Ime projekta: Biblioteka

Pregledani fajlovi: BookController.cs

Izveštaj pripremio: Ahmed Raščić

Verzija: 14b4102

## Lista grešaka

Tip	Opis	Lokacija	Ozbilnost
Stil koda	Sve rute nisu imenovane konzistentnim stilom	Imena ruta u BookController.cs	2
Ponavljanje koda	Ime i tip rute se može dati skupa	Npr. ruta addBook i ostale	2
Dizajn koda	Logika posuđivanja se treba odvojiti od same CRUD logike knjiga	Rute loanBook i getBookBack	3
Čitljivost	Manjak opisnih i dokumentujućih komentara	BookController.cs	1
Upravljanje greškama	Greške se vraćaju kao odgovor	Rute loanBook i getBookBack	2

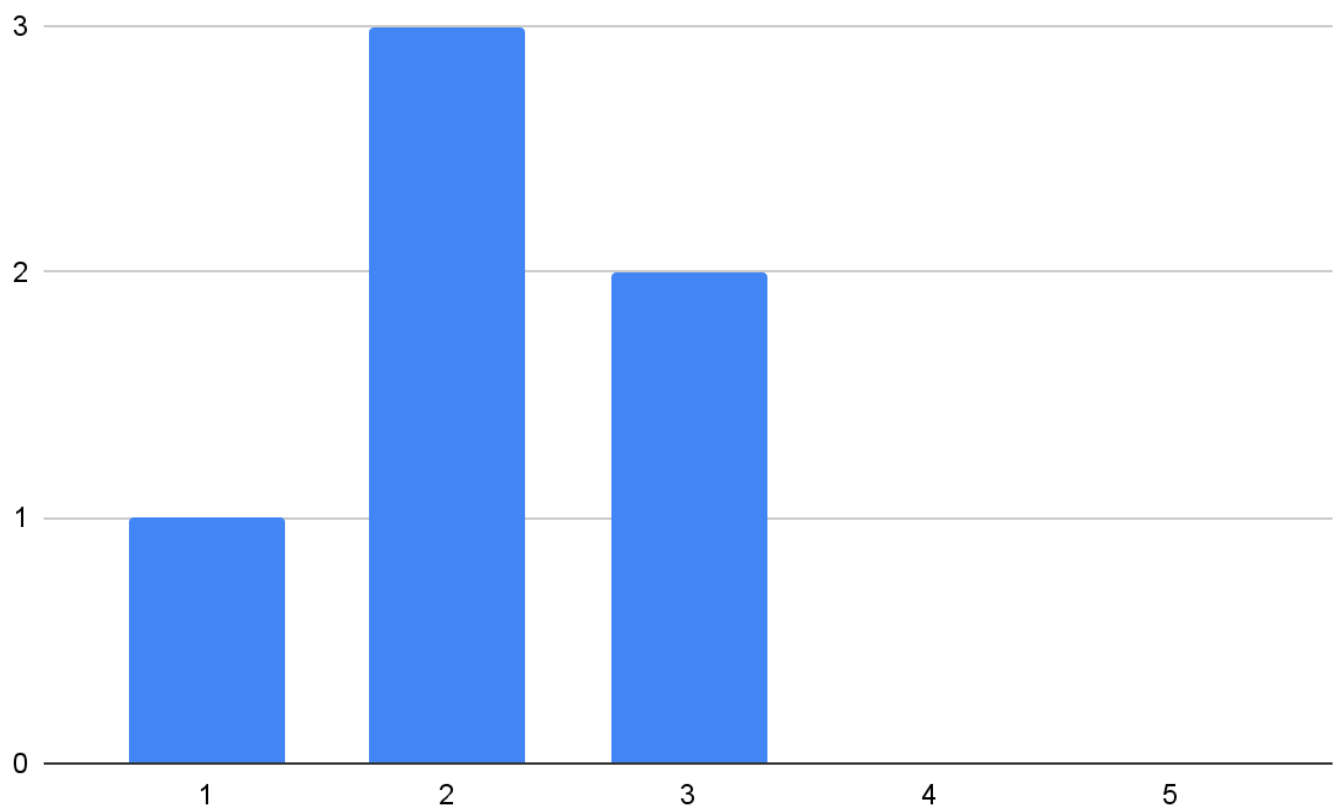
## Donesene odluke

Ujednačiti stil imenovanja ruta te ukloniti objediniti stereotype za tip i ime rute
Izdvojiti logiku posuđivanja i vraćanja u zaseban servis, CRUD logiku knjiga ne mijenjati
Spriječiti dostizanje izuzetaka do korisnika, vratiti odgovarajuće HTTP kodove

## Komentari

Većina grešaka su niske ozbilnosti i neće ugroziti rad ni funkcionalnost sistema. Većina grešaka se odnosi na poboljšanje održivosti postojećeg koda kao i poboljšanje korisničkog iskustva. Korektivne radnje su relativno jednostavne i lako izvodive.

Slijedi histogramski statistički prikaz broja pronađenih grešaka po njihovoj ozbiljnosti.



# Code Review Checklist

Funkcionalnost: posuđivanje i razduživanje knjiga

- ☒ Da li dati kod uspješno implementira opisanu funkcionalnost?

Da, izmijenjeni kod implementira funkcionalnosti zaduživanja i razduživanja specifičnih knjiga kao i pregled svih zaduženih knjiga po korisniku te zaduženosti specifične knjige (u sklopu pregleda detalja knjige).

- ☒ Može li se kod pojednostaviti?

Može - uvesti konzistentno imenovanje ruta, ukloniti dupliciranje u navođenju tipa rute i njenog imena odvojeno (to dvoje se može spojiti) kao i odvojiti metode specifično vezano za knjige u zaseban servis.

- ☒ Da li kod uvodi nove biblioteke ili međuovisnosti u sistem?

Ne - izmjena je fokusirana samo na rute i logiku funkcionalnosti, nikakve dodatne biblioteke ili međuovisnosti se ne dodaju.

- ☒ Je li kod jednostavan za shvatiti?

Kod je uglavnom čitljiv i razumljiv - potencijalno bi se mogli dodati komentari koji bi dodatno pojasnili određene metode i segmente.

- ☒ Koji dijelovi koda su zbunjujući i teški za shvatiti?

Kod je jasan i nema zbunjujućih dijelova.

- ☒ Je li kod na pravilnom nivou apstrakcije?

Većinom jeste - izuzev metode za obračun popusta koja je više vezana za poslovnu logiku i mogla bi se izdvojiti iz kontrolera.

- ☒ Mogu li se poboljšati održivost, performansi i sigurnost koda?

Kod je za navedene funkcionalnosti zadovoljavajući u pogledu održivosti, čitljivosti, sigurnosti i performansi.



- ☒ Može li se poboljšati čitljivost koda?

Može - davanje konzistentnih imena i uvođenje komentara bi poboljšale čitljivost.

- ☒ Da li se kod pridržava svih SOLID principa?

Kod se pravilno pridržava svih SOLID principa i ne krši nijedan.

## Izvještaj o pronađenim greškama

**Datum:** 13.decembar.2023.

**Izvještaj pripremila:** Nejra Stršević

**Ime projekta:** Biblioteka

**Pregledani fajlovi:** BookController.cs

### Lista grešaka

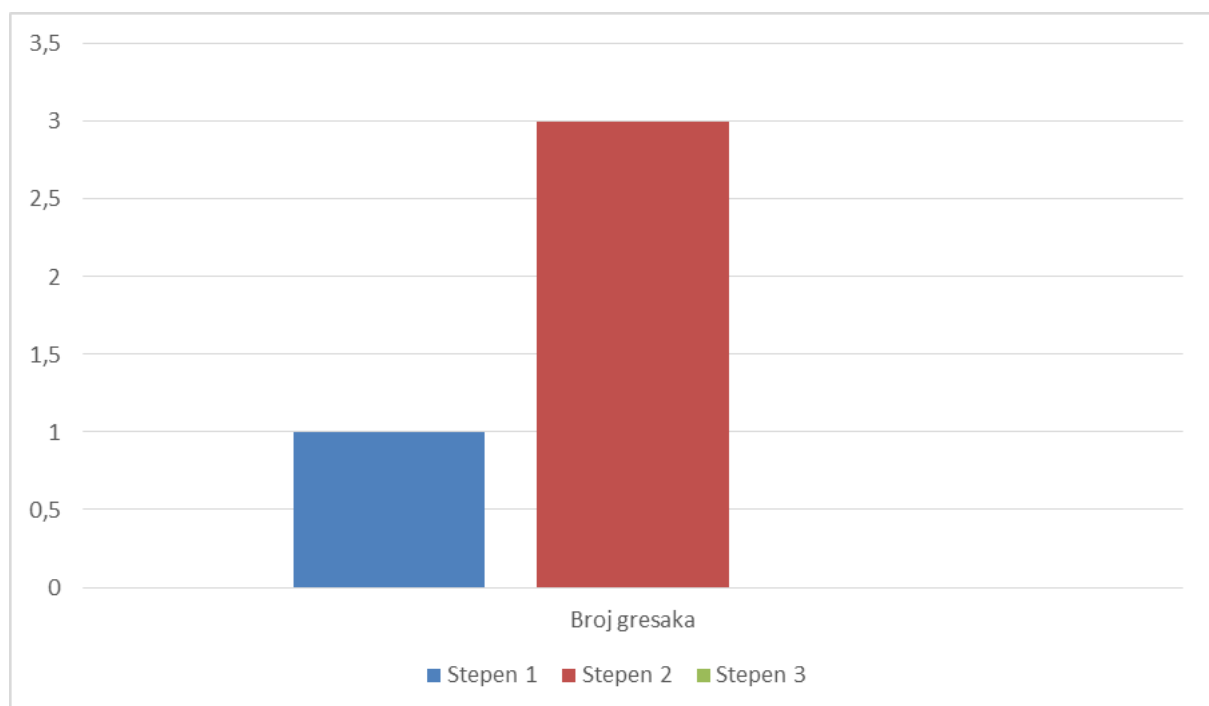
Tip	Opis	Lokacija	Ozbiljnost
Imenovanje varijabli	Sve rute nisu imenovane konzistentnim stilom	Imena ruta u BookController.cs	2
Ponavljanje koda	Ime i tip rute se može dati skupa	Npr. ruta addBook i ostale	2
Upravljanje greškama	Dodavanje praćenja greški	U SearchBook i GetBookDetails, ako knjiga ne postoji ili ne posjeduje detalje	2
Čitljivost	Dodati komentare radi lakšeg razumijevanja koda	BookController.cs	1
Čitljivost	Drugi način provjere da li je poslana null vrijednost	BookController.cs	1

## Donesene odluke

Promijeniti imena varijabli radi bolje čitljivosti
Dodati zapisivanje grešaka radi lakšeg praćenja problema u produkciji
Izmjena načina rukovanja s izuzecima, radi lakše čitljivosti

## Komentari

Greške nisu ozbiljnije i većina popravki je fokusirana na poboljšanje čitljivosti koda, postizanje clean coda i lakše razumijevanje funkcionalnosti koje trebaju da budu postignute s ovim metodama. Kod funkcionise onako kako bi trebao, ove promjene bi samo učinile kod kvalitetnijim.



# Code review checklist

## Funkcionalnost:

***Pretraživanje i pregled knjiga*** - sistem će omogućiti pretragu naslova biblioteke po nekoliko kriterija te će također podržavati detaljniji pregled svake pojedinačne knjige

1. Da li dati kod uspješno implementira opisanu funkcionalnost?

Da, date funkcije uspješno implementiraju opisane funkcionalnosti za pretraživanje i pregled knjiga.

2. Može li se kod pojednostaviti?

Može, bilo bi poželjno bolje imenovanje varijabli, također ne bi bilo loše provjeriti na početku da li je knjiga koje se pretražuje null vrijednost.

3. Da li kod uvodi nove biblioteke ili međuovisnosti u sistem?

Dodatne biblioteke ili međuovisnosti se ne dodaju u sistem dodavanjem ove dvije funkcionalnosti.

4. Je li kod jednostavan za shvatiti?

Ove dvije metode su poprilično jednostavne i čitljive, tako da nije teško shvatiti koje funkcionalnosti treba da zadovolje.

Korištenje komentara u kodu bi itekako pomoglo čitljivosti i bržem razumijevanju koda.

5. Koji dijelovi koda su zbunjujući i teški za shvatiti?

Svi dijelovi koda su prilično jednostavni, nisu teški za shvatiti.

6. Je li kod na pravilnom nivou apstrakcije?

Da, kod je na pravilnom nivou apstrakcije.

7. Mogu li se poboljšati održivost, performansi i sigurnost koda?

Održivost i performanse koda su zadovoljenje, sigurnost koda bi se mogla popraviti tako da pazimo da su pristupne informacije bazi podataka sigurno pohranjene i nisu izložene u kodu, te pažljivo rukovanje iznimkama i bez suvišnih informacija o greškama klijentu u produkcijskom okruženju.

8. Može li se poboljšati čitljivost koda?

Može - davanje konzistentnih imena i uvođenje komentara bi poboljšale čitljivost.

## 9. Da li se kod pridržava svih solid principa?

Funkcija trenutno radi više od jedne stvari: dohvaća knjigu, zatim dohvaća vezane informacije o zajmu i korisniku. Ovaj princip se često naziva "single responsibility principle" (srp). Preporučljivo je razdvojiti te zadatke u zasebne metode kako bi svaka metoda obavljala jedan konkretni zadatak.

# Izveštaj o pronađenim greškama

Datum: 14.12.2023.

Ime projekta: Biblioteka

Pregledani fajlovi: BookController.cs

Izveštaj pripremila: Emina Peljto

Verzija: e6cc45d

## Lista grešaka

Tip	Opis	Lokacija	Ozbiljnost
Upravljanje greškama	Nije definisano šta će se desiti ukoliko se metodi proslijede parametri koji su null.	Metoda ApplyCategorySpecific Benefits u BookController.cs	3
Čitljivost	Ponavljanje Price i Days atributa Loan modela unutar switch konstrukcije.	Metoda ApplyCategorySpecific Benefits u BookController.cs	1

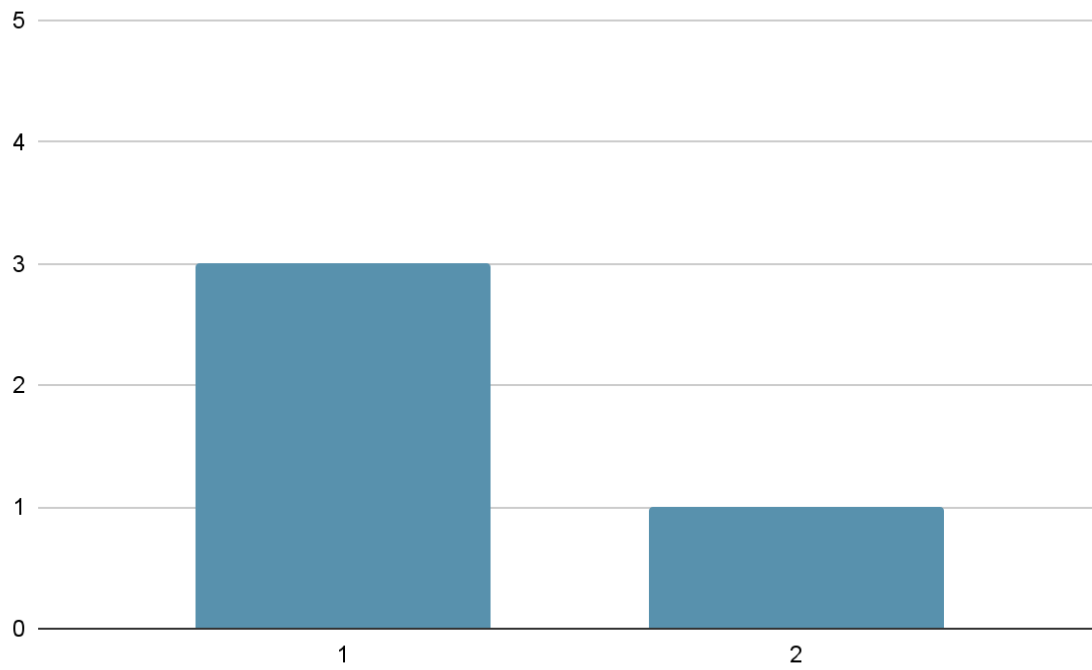
## Donesene odluke

Baciti exception ukoliko su parametri null.
Dodati pomoćnu metodu koja će ispravno postavljati Price i Days attribute Loan modela te nju pozivati unutar switch konstrukcije.

## Komentari

Korektivne akcije za navedene greške su poznate i jednostavno izvodive. Prva greška može ugroziti rad sistema.

Histogram pronađenih grešaka po ozbiljnosti:



# Code Review Checklist

Funkcionalnost: posebne kategorije korisnika

- ☒ Da li dati kod uspješno implementira opisanu funkcionalnost?

Da, dati kod uspješno implementira opisanu funkcionalnost. Funkcija prima korisnika i loan te primjenjuje određene pogodnosti na osnovu tipa korisnika. Svaka kategorija korisnika ima svoje posebne pogodnosti.

- ☒ Može li se kod pojednostaviti?

Kod se može pojednostaviti uvođenjem nove privatne metode koja bi primala parametre loan, discount percentage i loan days te primjenjivala pogodnosti na loan koristeći argumente koje je dobila. Na ovaj način kod će biti čistiji te će se smanjiti ponavljanje koda.

- ☒ Da li kod uvodi nove biblioteke ili međuovisnosti u sistem?

Ne, kod ne uvodi nikakve nove biblioteke ili međuovisnosti u sistem.

- ☒ Je li kod jednostavan za shvatiti?

Da, kod je jednostavan za shvatiti. Koristi dobre prakse u smislu imenovanja varijabli i funkcija. Također koristi pomoćnu funkciju što olakšava održavanje i razumijevanje.

- ☒ Koji dijelovi koda su zbunjujući i teški za shvatiti?

Kod je jasan, nema dijelova koji su zbunjujući i teški za shvatiti.

- ☒ Je li kod na pravilnom nivou apstrakcije?

Uglavnom da, obračun popusta se može izdvojiti iz kontrolera jer je više vezan za poslovnu logiku.

- ☒ Mogu li se poboljšati održivost, performans i sigurnost koda?

Održivost i performans koda za navedenu funkcionalnost su dobri i ne mogu se značajno poboljšati. Sigurnost se može poboljšati tako što će se uvesti validacija ulaznih podataka.



- ☒ Može li se poboljšati čitljivost koda?

Čitljivost koda se može poboljšati uvođenjem nove metode koja je spomenuta kod pojednostavljivanja koda.

- ☒ Da li se kod pridržava svih SOLID principa?

Da, kod se pridržava svih SOLID principa.

# Izveštaj o pronađenim greškama

**Datum:** 3.decembar.2023.

**Ime projekta:** Biblioteka

**Pregledani fajlovi:** UserController.cs

**Izveštaj pripremio:** Danijel Mioković

**Verzija:** 14b4102

## Lista grešaka

Tip	Opis	Lokacija	Ozbiljnost
Stil koda	Sve rute nisu imenovane konzistentnim stilom	Imena ruta u UserController.cs	2
Ponavljanje koda	Ime i tip rute se može dati skupa	Ruta login i ostale	2
Čitljivost	Manjak opisnih i dokumentujućih komentara	UserController.cs	1

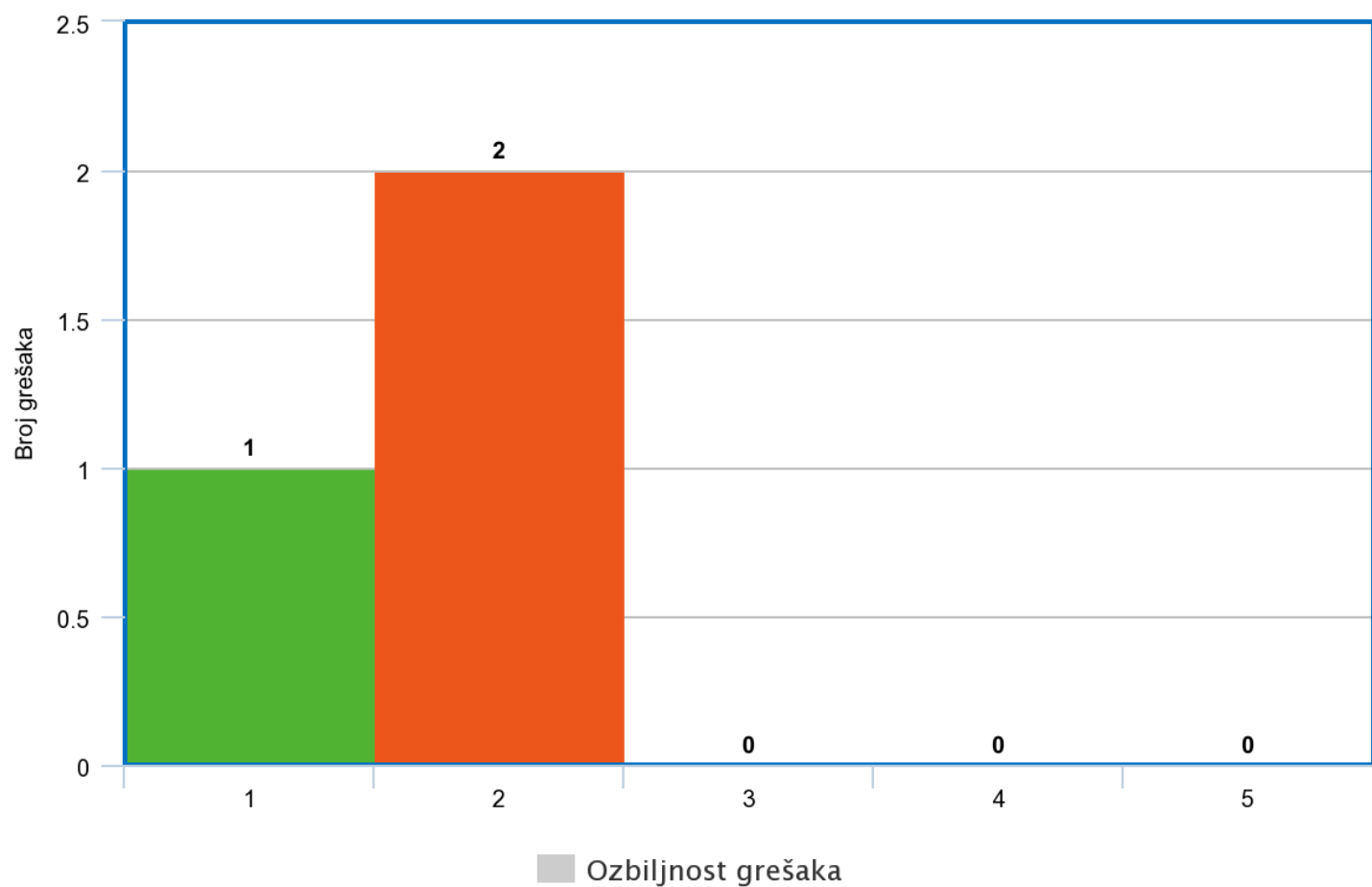
## Donesene odluke

Ujednačiti stil imenovanja ruta te objediniti stereotipe za tip i ime rute
Spriječiti dostizanje izuzetaka do korisnika, vratiti odgovarajuće HTTP kodove

## Komentari

Većina grešaka su niske ozbiljnosti i neće ugroziti rad ni funkcionalnost sistema. Većina grešaka se odnosi na poboljšanje održivosti postojećeg koda. Korektivne radnje su relativno jednostavne i lako izvodive.

Slijedi histogramski statistički prikaz broja pronađenih grešaka po njihovoj ozbiljnosti.



# Code Review Checklist

Funkcionalnost: prijava i registracija korisnika

- ☒ Da li dati kod uspješno implementira opisanu funkcionalnost?

Da, izmijenjeni kod implementira funkcionalnosti prijave i registracije korisnika, te mogućnost odjave korisnika.

- ☒ Može li se kod pojednostaviti?

Može - ukloniti dupliciranje u navođenju tipa rute i njenog imena odvojeno (to dvoje se može spojiti).

- ☒ Da li kod uvodi nove biblioteke ili međuovisnosti u sistem?

Ne - izmjena je fokusirana samo na rute i logiku funkcionalnosti, nikakve dodatne biblioteke ili međuovisnosti se ne dodaju.

- ☒ Je li kod jednostavan za shvatiti?

Kod je uglavnom čitljiv i razumljiv - potencijalno bi se mogli dodati komentari koji bi dodatno pojasnili određene metode i segmente.

- ☒ Koji dijelovi koda su zbunjujući i teški za shvatiti?

Kod je jasan i nema zbunjujućih dijelova.

- ☒ Je li kod na pravilnom nivou apstrakcije?

Da.

- ☒ Mogu li se poboljšati održivost, performansi i sigurnost koda?

Kod je za navedene funkcionalnosti zadovoljavajući u pogledu održivosti, čitljivosti, sigurnosti i performansi.

☒ Može li se poboljšati čitljivost koda?

Može - davanje konzistentnih imena i uvođenje komentara bi poboljšale čitljivost.

☒ Da li se kod pridržava svih SOLID principa?

Kod se pravilno pridržava svih SOLID principa i ne krši nijedan.

# Izveštaj o pronađenim greškama

Datum: 17.12.2023.

Izveštaj pripremio: Eman Hodžić

Ime projekta: Biblioteka

Pregledani fajlovi: UserController.cs

## Lista grešaka

Tip	Opis	Lokacija	Ozbiljnost
Čitljivost	Nedostaju komentari	UserController.cs	1
Čitljivost	Glavna metoda sadrži 3 potencijalne pomoćne metode	UserController.cs	1
Nestabilnost aplikacije	Bacanje izuzetaka	UserController.cs	3

## Donesene odluke

Dodati komentare kako bi se povećala čitljivost koda
Izdvojiti 3 pomoćne metode iz glavne metode
Kreirati rezultatsku klasu umjesto bacanja izuzetaka

## Komentari

Greške nisu ozbiljnije i većina popravki je fokusirana na poboljšanje čitljivosti koda, postizanje clean coda i lakše razumijevanje funkcionalnosti koje trebaju da budu postignute s ovim metodama. Kod funkcionije onako kako bi trebao, ove promjene bi samo učinile kod kvalitetnijim.

# Code Review Checklist

## Funkcionalnost:

*Obnova članarine - korisnici će preko sistema moći pratiti status svoje članarine te istu po želji obnoviti.*

### Da li dati kod uspješno implementira opisanu funkcionalnost?

Da, posmatrana metoda u klasi UserController uspješno implementira opisanu funkcionalnost.

### Može li se kod pojednostaviti?

Ovaj kod se može pojednostaviti na način da se izdvoje dijelovi koda u zasebne metode :

- GetUserById(RenewalRequest request)
- ValidateMonths(RenewalRequest request)
- UpdateUserExpirationDate(User user, int months)

### Da li kod uvodi nove biblioteke ili međuovisnosti u sistem?

Ne, ovaj kod ne uvodi nove biblioteke ili međuovisnosti u sistem.

### Je li kod jednostavan za shvatiti?

Da, kod je jednostavan za shvatiti čak i prije pojednostavljenja.

### Koji dijelovi koda su zbunjujući i teški za shvatiti?

/

### Je li kod na pravilnom nivou apstrakcije?

Da, kod je na pravilnom nivou apstrakcije za svoju složenost i funkcionalnost.

### Mogu li se poboljšati održivost, performans i sigurnost koda?

Ukoliko bismo koristeći 'async' i 'await' za olakšavanje asinhronog programiranja poboljšali bismo responsivnost aplikacije, jer asinhronne metode omogućavaju da se izvršavaju aplikaciji da ostane responsivna dug se izvršavaju duge operacije poput čekanja na odgovor baze podataka. Također poboljšale bi se performanse i efikasnost, jer bi aplikacija mogla raditi druge zadatke dok čeka na dugu operaciju.

### **Može li se poboljšati čitljivost koda?**

Čitljivost koda se može poboljšati na način da umjesto bacanja izuzetaka *ArgumentException* i *HttpRequestException*, kreira nova klasa *RenewMembershipResult* koja bi za attribute imala npr.

- bool Success
- string Message

Ova promjena poboljšava upravljanje kontrolom toka i omogućava lakše upravljanje raznim scenarijima.

### **Da li se kod pridržava svih SOLID principa?**

Ukoliko bismo primjenili sve ranije navede preporuke o promjeni koda, ova metoda bi bila dobro struktuirana i poštovala bi sve SOLID principe.



- Spisak urađenih korektivnih akcija

1. Ujednačen stil imenovanja ruta
2. Uklonjeno nepotrebno dupliciranje stereotipa za ime ruta
3. Izdvojena logika posuđivanja i vraćanja knjige u zaseban servis
4. Spriječeno dolaženje izuzetaka do korisničke strane, vraćanje pravilnih odgovora
5. Poboljšana imena varijabli radi čitljivosti
6. Dodani dokumentirajući komentari radi lakšeg i bržeg razumijevanja koda
7. Uvedeno zapisivanje grešaka za praćenje eventualnih problema u produkciji
8. Strožije validacije ulaznih parametara u metodama gdje je bilo neophodno