



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET SARAJEVO



Odsjek za računarstvo i informatiku

Projektni zadatak

Verifikacija i validacija softvera

Članovi tima:

Brčaninović Hasan

Fejzagić Sead

Graca Edwin

Obućina Nermin

Prašović Asmir

Sarajevo, decembar 2023.

Inspekcija koda

Tim za inspekciju koda se sastoji od sljedećih članova sa navedenim ulogama:

1. Brčaninović Hasan – moderator i zapisničar
2. Fejzagić Sead – autor i recenzent za dijelove programskog koda drugih članova
3. Graca Edwin – autor i recenzent za dijelove programskog koda drugih članova
4. Obućina Nermin – autor i recenzent za dijelove programskog koda drugih članova
5. Prašović Asmir – autor i recenzent za dijelove programskog koda drugih članova

Inspekcija koda je izvršena putem gitHub platforme, google meet-a i ostalih google kolaborativnih alata.

Raspored prema kojem su dodajeljeni dijelovi koda za pregled su predstavljeni u narednoj tabeli:

Autor	Recenzent	Dodjeljeni dio koda
Fejzagić Sead	Graca Edwin	Metode u klasi
Graca Edwin	Obućina Nermin	Metode u klasi
Obućina Nermin	Prašović Asmir	Metode u klasi
Prašović Asmir	Fejzagić Sead	Metode u klasi
Brčaninović Hasan	Svi ostali članovi za svoj dio sistema	Modelske klase i metode iz baze podataka.

Članovima tima je raspored i kod za inspekciju dostavljen dan prije održavanja inspekcije putem google drive-a. Link na kojem se nalazi projekat je: <https://github.com/aprasovic1/VVS-projekat>.

Sastanak za inspekciju koda je održan 17. 12. 2023.

Kao pripremni materijal, moderator tima je ostalim članovima prosljedio sljedeće materijale:

- Programske kodove na linku: <https://github.com/aprasovic1/VVS-projekat>
- Check listu
- Tabela klasifikovanih grešaka po ozbiljnosti
- Template izvještaja za inspekciju

Pobrojani materijali su dati nastavku.

Check lista

Struktura:

1. Da li je kod (metoda, klasa, interface ...) kompletno implementiran u skladu sa dizajnom sistema (klasni dijagram, ERD i ostali dijelovi dokumentacije)?
2. Da li se kod pridržava relevantnih standarda programiranja za verziju projekta koja se koristi (metode iz biblioteka i ostali elementi framework-a su dopustivi tj. nisu *depricated*)?
3. Da li je kod struktuiran u skladu sa MVC arhitekturom (postoje odgovorajuće modelske, kontrolerske i pogled klase)?
4. Da li je kod struktuiran u skladu sa objektno – orijentisanim principima (nasljeđenost klasa, struktura atributa i regiona u modelskim klasama, pravilno iskorišteni moduli prava pristupa podacima, pravilno konfigurisane metode set i get, konstruktor ...)
5. Da li su ispravno definisane anotacije za primarne i strane ključeve u modelskim klasama, kao i relacije između entiteta?
6. Da li postoji metoda koja se ne koristi ili koja nije dohvatljiva?
7. Da li postoje metode koje se mogu zamijeniti nekim drugim metodama koje se više koriste (*reusability*)?
8. Da li postoji odgovarajući patern za dio koda koji može da se iskoristi?
9. Da li postoji kod koji se duplira i koji je moguće refaktoristi u jednu funkciju?
10. Da li se koriste simboli koji adekvatno zamjenjuju *magične konstante* (konstante koje nemaju adekvatan naziv za funkcionalnost koju imaju u kodu)?
11. Da li postoje moduli koji su izrazito kompleksni i krše SOLID principe?

Dokumentacija:

1. Da li kod ima prateće komentare?
2. Da li je korišten čitljiv stil komentara?
3. Da li komentar odgovara funkcionalnosti koda?
4. Da li je naziv metode, klase, varijable, interface-a i ostalih dijelova koda u skladu sa konvencijama imenovanja dobre prakse?
5. Da li su komentari dovoljno opširni i da li su u granicama dobrih praksi (komentari treba da ne budu duži od paragrafa koji ima 5 redova osim u izuzetnim situacijama)?
6. Da li su komentari konzistentni sa ostalom dokumentacijom (komentari se referiraju na iste nazive varijabli, metoda i drugih elmenata programa)?

7. Da li su svi komentari pisani na istom govornom jeziku?

Varijable:

1. Da li su sve varijable imenovane u skladu sa svim pravilima imenovanja dobrih praksi?
2. Da li su svim varijablama pridruženi odgovarajući tipovi i inicijalne vrijednosti?
3. Da li postoje varijable koje su suvišne ili koje se ne koriste (*dead code variables*)?
4. Da li postoje kritične varijable koje mogu da prevaziđu opseg vrijednosti tipa varijable?
5. Da li je definisan način poređenja varijabli sa različitim tipovima?

Aritmetički operatori:

1. Da li u kodu postoji poređenje necjelobrojnih vrijednosti?
2. Da li je definisana preciznost za poređenje necjelobrojnih varijabli?
3. Da li je predviđen izuzetak za dijeljenje sa nulom?
4. Da li je definisan način izvršavanja operacija sa operandima različitih tipova?

Petlje i kontrole tokova (grane):

1. Da li su grane, petlje i ostale logičke konstrukcije konkretno formirane u skladu sa svojom strukturom i sintaksom?
2. Da li su uslovi tačni i jednostavni za održavanje?
3. Da li postoje sporedni efekti koji mogu da tokom parsiranja uzrokuju pogrešan zaključak za dio koda (*side effects*)?
4. Da li su testirane sve grane (uključujući *ELSE* i *DEFAULTE*)?
5. Da li se prvo testiraju uslovi koji imaju veću vjerovatnoću za ispunjenjem?
6. Da li su korektno postavljene granice petlje?
7. Da li je petlja formirana tako da ne upada u beskonačnu petlju?
8. Da li su indeksi petlji pravilno inicijalizirani i korišteni u tijelu petlji?
9. Da li se u petlji pravilno ažuriraju varijable koje imaju inkrementalnu promjenu vrijednosti?
10. Da li postoje dijelovi petlje koji se mogu izdvojiti iz tijela petlje?
11. Da li se pravilno koriste *break* i *continue*?

12. Izbjegava li kod u petlji manipuliranje varijablom indeksa ili njezinu upotrebu nakon izlaska iz petlje?
13. Da li postoje redundantne petlje koje se mogu objediniti u jedno tijelo?
14. Da li su uslovi za prekid petlje tačni i jasni?

Defenzivno programiranje:

1. Da li se testiraju indeksi za pristup elementima niza?
2. Da li se vrši validiranje podataka sa ulaza?
3. Da li su korištene konvezije različitih tipova?
4. Da li se u kodu obrađuju tačni podaci sa ulaza?
5. Da li je izlaz iz koda adekvatno definisan?
6. Da li je izvršena provjera postajanja elemenata koji se trebaju ažurirati?
7. Da li je konekcija sa bazom podataka uspješno ostvarena?
8. Da li je stanje podataka nakon izvršavanje koda konzistentno sa stanjem u bazi podataka (sinhronizovan sadržaj)?
9. Da li su parametri metoda tačni i u skladu sa dokumentacijom?
10. Da li su izlazne vrijednosti iz funkcije pravilno definisane?

Tabela klasifikovanih grešaka:

Stepen ozbiljnosti greške	Opis greške
5 (kritično)	<ol style="list-style-type: none"> 1. Sprječava postizanje osnovnih mogućnosti. 2. Ugrožava sigurnost, zaštitu i druge kritične zahtjeve. 3. Ugrožava funkcionalnost sistema.
4	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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 ostvarivanje zahtjevanih mogućnosti sistema ili operaterove osnovne mogućnosti. 2. Neugodnost za programere ili personal na održavanju, ali ne sprečava realizaciju ovih odgovornosti.
1 (minorno)	1. Bilo koji efekti koji nemaju negativan uticaj.

Template: Izvještaj u inspekcijskoj sesiji

Izvještaj u inspekcijskoj sesiji						
Osnovne informacije:						
Datum održavanja sesije						
Naziv projekta						
Verzija dokumenta						
Broj sati utrošenih za inspekcijsku						
Članovi inspekcijskog tima						
Autor koda						
Recenzent						
Naziv sekcije koda za recenziju						
<u>Lista grešaka:</u>						
Broj greške	Tip greške	Priroda greške (W/M/E)	Opis greške	Lokacija greške (broj linije u kodu)	Stepen ozbiljnosti greške	Izvor greške (checklista)

Izvor greške je broj stavke iz check-liste čijom provjerom je otkrivena greška. U slučaju da izvor nije definisan u listi, u polje napisati slovo N i uvrstiti u postojeću listu.

Broj odluke	Odluka
1	Naknadno praćenje će obaviti:
2	Naknadna inspekcija se preporučuje: Da/Ne
3	

--

Moderator je kreirao inicijalni issue kojim se započinje inspeksijska sesija. Inicijalni issue je prikazan na narednoj slici.

Inspekcija programskog koda #16

Open hbrcaninovic opened this issue 2 hours ago · 0 comments

hbrcaninovic commented 2 hours ago • edited

Ovim issue-em se započinje inspekcija programskog koda. Zadaci pojedinih članova su definisani u novim issue-ima.

hbrcaninovic added the **documentation** label 2 hours ago

hbrcaninovic assigned **aprasovic1**, **sejooo1**, **nobucina1** and **EdwinTriton** 2 hours ago

Assignees

aprasovic1
sejooo1
nobucina1
EdwinTriton

Labels

documentation

Projects

Potom je moderator otvorio nove issue u kojima je dodjelio zadatke svakom članu inspeksijskog tima. Svaki issue se odnosi na 2 člana iz tima od kojih je jedan autor koda, a drugi recenzent. U sklopu issue-a su recenzentu date informacije koje metode treba da pregleda koristeći check listu. Jedan od ovih issue-a je prikazan na narednoj slici.

Dodjela inspeksijskog koda #20

Closed hbrcaninovic opened this issue 2 hours ago · 1 comment

hbrcaninovic commented 2 hours ago • edited

Kolega Fejzagić Sead vrši inspekciju programskog koda kolege Prašović Asmira. Dodijeljena sekcija koda za pregled je:

Lokacija sekcije: RecommendationsController
Ime metode: Index, GetAllBooksAsync, GetBestBooks

Na navedenim lokacijama u kodu postaviti komentare sa potrebnim izmjenama. Po završetku inspekcije postaviti promjene na repozitorij, zatvoriti issue, priložiti izvještaj na drive-u i kreirati issue u kojem se od kolege traži da izvrši naznačene korekcije koda.

hbrcaninovic added the **documentation** label 2 hours ago

hbrcaninovic assigned **aprasovic1** and **sejooo1** 2 hours ago

Assignees

aprasovic1
sejooo1

Labels

documentation

Projects

None yet

Milestone

No milestone

Development

Create a branch for this issue or link a pull request.

Svaki recenzent je na osnovu issue-a izvršio inspekciju koda na lokalnom repozitoriju referentne verzije sa main-a. Greške koje su pronašli tokom inspekcije su označili komentarima koje su autori korigovali nakon inspekcije. Recenzenti su po završetku pregleda koda postavili komentare na gitHub repozitorij i zatvorili issue koji im je dodjeljen, te su otvorili novi issue za *bug* labelom u kojem su tražili od autora koda da izvrši korektivne radnje. Također su recenzenti na drive-u moderatoru inspekcije prosljedili izvještaj o greškama. Primjer issue-a koji je recenzent zatvorio kao odgovor na moderatorov zadatak je dat u nastavku.

hbrcaninovic commented 3 hours ago • edited

Kolega Fejzagić Sead vrši inspekciju programskog koda kolege Prašović Asmira. Dodijeljena sekcija koda za pregled je:

Lokacija sekcije: RecommendationsController
Ime metode: Index, GetAllBooksAsync, GetBestBooks

Na navedenim lokacijama u kodu postaviti komentare sa potrebnim izmjenama. Po završetku inspekcije postaviti promjene na repozitorij, zatvoriti issue, priložiti izvještaj na drive-u i kreirati issue u kojem se od kolege traži da izvrši naznačene korekcije koda.

hbrcaninovic added the **documentation** label 3 hours ago

hbrcaninovic assigned aprasovic1 and sejoo01 3 hours ago

sejoo01 commented 1 hour ago

Izvršena inspekcija koda i zahtijevam korektivne akcije.

sejoo01 closed this as **completed** 1 hour ago

Assignees: aprasovic1, sejoo01

Labels: **documentation**

Projects: None yet

Milestone: No milestone

Development: Create a branch for this issue or link a pull request.

Notifications: Unsubscribe

You're receiving notifications because you authored the thread.

3 participants: aprasovic1, sejoo01, hbrcaninovic

Primjer novog *bug* issue-a koji je recenzent otvorio i u kojem traži prepravke koda je dat na narednoj slici.

Zahtjev za ispravku koda! #30

Closed sejoo01 opened this issue 1 hour ago • 1 comment

sejoo01 commented 1 hour ago • edited by hbrcaninovic

Otkrivene greške u RecommendationsControlleru. Potrebno izvršiti korekcije naznačenih dijelova koda.

sejoo01 added the **bug** label 1 hour ago

sejoo01 assigned aprasovic1 1 hour ago

Assignees: aprasovic1

Labels: **bug**

Projects: None yet

Milestone: No milestone

Autor koda je nakon završene inspekcije i otvaranja bug issue-a izvršio korekciju koda prema komentarima koje su im navedeni u kodu. Zatim su postavili promjene na remote repozitorij i zatvorili bug issue. Primjer ovakve akcije je dat na narednoj slici.

Zahtjev za ispravku koda! #30

[Edit](#) [New issue](#)[Closed](#)

sejooo1 opened this issue 1 hour ago · 1 comment



sejooo1 commented 1 hour ago • edited by hbrcaninovic

Otkrivene greške u RecommendationsControlleru. Potrebno izvršiti korekcije naznačenih dijelova koda.

sejooo1 added the **bug** label 1 hour ago

sejooo1 assigned aprasovic1 1 hour ago



aprasovic1 commented 1 hour ago

Izvršene ispravke koda.

aprasovic1 closed this as **completed** 1 hour agoAssignees
● aprasovic1Labels
bugProjects
None yetMilestone
No milestoneDevelopment
[Create a branch](#) for this issue or link a pull request.Notifications
[Subscribe](#) Customize

You're not receiving notifications from this thread.

Analogno su ostali članovi tima izvršili inspekciju koda. Nakon uspješno završene inspekcije, moderator je zatvorio inicijalni issue. Pregled svih issue-a u toku inspekcije su prikazani na sljedećoj slici.

<input type="checkbox"/>	<input type="radio"/> 0 Open <input checked="" type="radio"/> 9 Closed	Author	Label	Projects	Milestones	Assignee	Sort
<input type="checkbox"/>	<input checked="" type="radio"/> Zahtjev za ispravku koda! bug #30 by sejooo1 was closed 1 hour ago					●	1
<input type="checkbox"/>	<input checked="" type="radio"/> Zahtjev za ispravku koda bug #26 by nobucina1 was closed 2 hours ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Zahtjev za ispravku koda bug #24 by EdwinTriton was closed 1 hour ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Potrebno izvršiti ispravke bug #22 by aprasovic1 was closed 3 hours ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Dodjela inspekcijskog koda documentation #20 by hbrcaninovic was closed 1 hour ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Dodjela inspekcijskog koda documentation #19 by hbrcaninovic was closed 3 hours ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Dodjela inspekcijskog koda documentation #18 by hbrcaninovic was closed 3 hours ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Dodjela inspekcijskog koda documentation #17 by hbrcaninovic was closed 3 hours ago					👤	1
<input type="checkbox"/>	<input checked="" type="radio"/> Inspekcija programskog koda documentation #16 by hbrcaninovic was closed 1 hour ago					👤	1

Napomena:

Određenom broju članova tima alat za inspekciju koda nije se mogao pokrenuti, te je gitHub iskorišten kao alternativa za inspekciju koda.

Izveštaj 2:

Sumarni izvještaj nakon što su recenzenti proslijedili svoje izvještaje je prikazan u nastavku.

Sumarni izvještaj nakon inspekcije koda						
Osnovne informacije:						
Datum održavane sesije		17. 17. 2023.				
Naziv projekta		VVS – projekat (E – Biblioteka)				
Verzija dokumenta		v1.0				
Broj sati utrošenih za inspekcijsku		2				
Članovi inspekcijuskog tima		Brčaninović Hasan, Fejzagić Sead, Graca Edwin, Obućina Nermin, Prašović Asmir				
Moderator		Brčaninović Hasan				
Uloženo vrijeme:						
Član tima		Broj sastanaka	Broj dana za pripremu inspekcije	Broj sati utrošenih za inspekciju	Ukupan broj sati utrošenih na korektivne akcije	
Fejzagić Sead		1	1	2	1.5	
Graca Edwin		1	1	1	1.5	
Obućina Nermin		1	1	2	1	
Prašović Asmir		1	1	1.5	1	
Brčaninović Hasan		2	3	2.5	1	
Sumirano		6	7	9	6	
Sažetak grešaka:						
Stepen ozbiljnosti greške	Priroda greške (W/M/E)			Faktor ozbiljnosti greške	Broj grešaka po istom stepenu ozbiljnosti	Dodatne napomene
5 (kritično)		1		16	1	-
4	1	1	1	8	3	-
3	2	1	2	4	5	-

2	3			2	3	-
1 (minorno)	1	1		1	2	-
Sumirano	7	4	3		14	-
Efikasnost detekcije defekata (broj sati po defektu) = $(9+6)/14 = 1.07$						
Broj defekata po broju stranica = $14/31 = 0.45$						

Prikaz pojedinih izvještaja inspekcije:

Prikaz pojedinih izvještaja inspekcije za svakog recenzenta i autora skupa sa statističkim alatom iskorištenim za prikaz je dat u nastavku.

Izvještaj u inspekcijskoj sesiji

Osnovne informacije:	
Datum održavanja sesije	17. 12. 2023.
Naziv projekta	VVS – projekat (E – Biblioteka)
Verzija dokumenta	v1.0
Broj sati utrošenih za inspekcijsku	1.5
Članovi inspekcijskog tima	Brčaninović Hasan, Fejzagić Sead, Graca Edwin, Obućina Nermin, Prašović Asmir
Autor koda	Nermin Obućina
Recenzent	Asmir Prašović
Naziv sekcije koda za recenziju	BookController 1:228 ReservationController 1:236

Lista grešaka:

Broj greške	Tip greške	Priroda greške (W/M/E)	Opis greške	Lokacija greške (broj linije u kodu)	Stepen ozbiljnosti greške	Izvor greške (checklista)
1.	error	W	Poruka "No reservations" nije adekvatna, ne opisuje problem dovoljno.	220	2	Def. prog. 5.
2.	error	M	U LateActivatedReservations() fali await prilikom pristupa bazi putem context	223	3	Def-prog.7.

3.	error	E	U LateActivatedReservations() višak await prilikom pristupa bazi putem _context	230	3	Def-prog.7.
4.	error	W	U LateActivatedReservations() pogrešan komentar. Spominje se Books umjesto Reservations	203	1	Dokumentacija. 3

Legenda:

Tip greške: error, defekat

Priroda greške: W = Wrong M = Missing E = Extra

Lokacija greške je opseg linija na kojim se nalazi greška (npr. 15-20 linije ili samo 15 linija).

Stepen ozbiljnosti greške su vrijednosti definisane u tabeli ozbiljnosti grešaka.

Izvor greške je broj stavke iz check-liste čijom provjerom je otkrivena greška. U slučaju da izvor nije definisan u listi, u polje napisati slovo N i uvrstiti u postojeću listu.

Naknadne (popratne) akcije:

Broj odluke	Odluka
1	Naknadno praćenje će obaviti: Nermin Obućina
2	Naknadna inspekcija se preporučuje: Da
3	

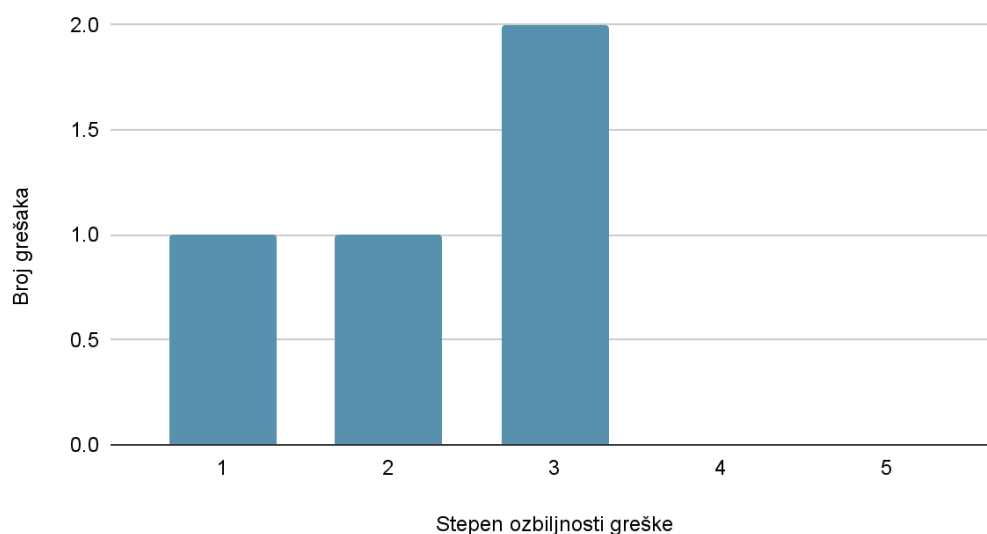
Komentari:

Izveštaj pokazuje da u kodu nisu pronađene ozbiljnije greške. Pomenute reške u radu s bazom se lahko ispravljaju dok neadekvatna poruka ne stvara problem sa radom aplikacije.

Pogrešan komentar je vjerovatno rezultat iscrpljenosti autora te će biti ispravljen.

Treba napomenuti da se povede računa o preciznim porukama i adekvatnim komentarima, kao i o radu sa bazom.

Broj grešaka prema ozbiljnosti



Izveštaj u inspekcijskoj sesiji

Osnovne informacije:

Datum održavanja sesije	17. 12. 2023.
Naziv projekta	VVS – projekat (E – Biblioteka)
Verzija dokumenta	v1.0
Broj sati utrošenih za inspekcijsku	2
Članovi inspekcijskog tima	Brčaninović Hasan, Fejzagić Sead, Graca Edwin, Obućina Nermin, Prašović Asmir
Autor koda	Edwin Graca
Recenzent	<i>Nermin Obućina</i>
Naziv sekcije koda za recenziju	Lokacija sekcije: VoucherController Ime metode: GenerateVoucherCode Lokacija sekcije: ReservationPaymentController Ime metode: ValidateVoucher

Lista grešaka:

Broj greške	Tip greške	Priroda greške (W/M/E)	Opis greške	Lokacija greške (broj linije u kodu)	Stepen ozbiljnosti greške	Izvor greške (checklista)
1.	defect	E	Voucheri trebaju biti duzine 12, ovom linijom ce biti 13	174	3	N

2.	defect	W	Ne provjerava dužinu vaučera	179	3	Def. Prog. 2.
3.	defect	W	Pogresno nazvana metoda	164	2	Dokumentacija 4.
4.	error	M	Viska zagrada	191	5	N

Legenda:

Tip greške: error, defekat

Priroda greške: W = Wrong M = Missing E = Extra

Lokacija greške: je opseg linija na kojim se nalazi greška (npr. 15-20 linije ili samo 15 linija).

Stepen ozbiljnosti greške: su vrijednosti definisane u tabeli ozbiljnosti grešaka.

Izvor greške: je broj stavke iz check-liste čijom provjerom je otkrivena greška. U slučaju da izvor nije definisan u listi, u polje napisati slovo N i uvrstiti u postojeću listu.

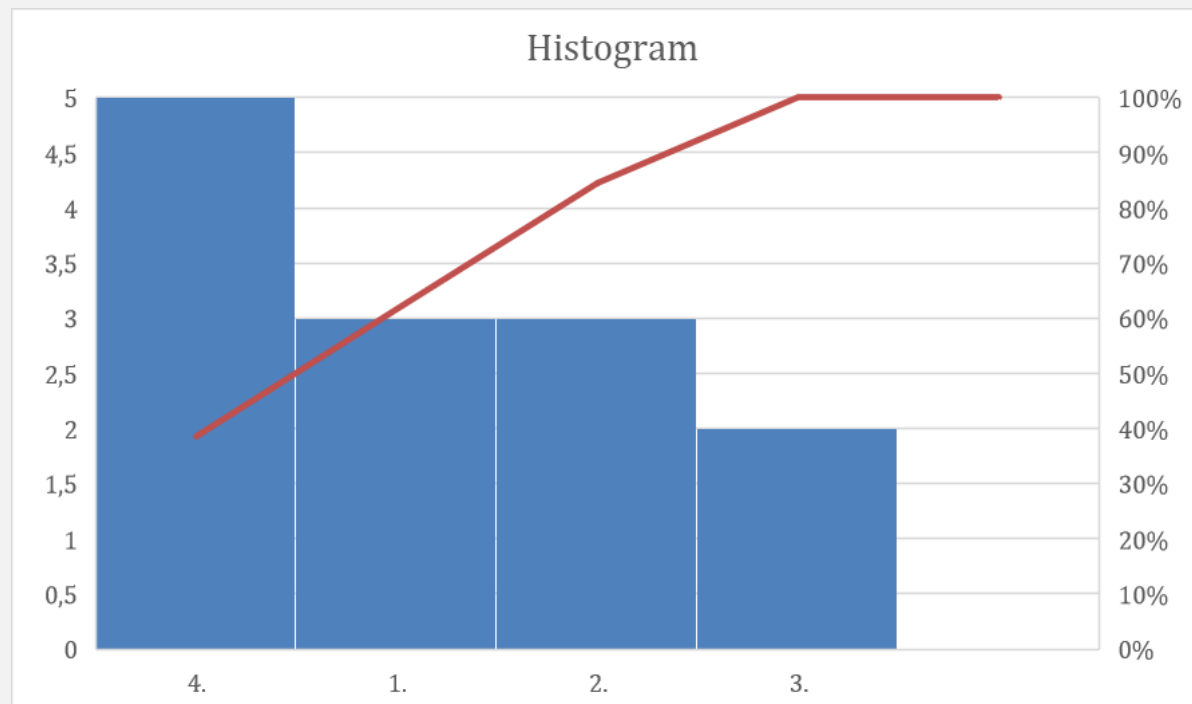
Naknadne (popratne) akcije:

Broj odluke	Odluka
1	Naknadno praćenje će obaviti: Edwin Graca
2	Naknadna inspekcija se preporučuje: Da
3	

Komentari:

Pored greški koje su nađene, ostatak koda je uredno i ispravno napisan.

Statistički prikaz:



Izveštaj u inspekcijskoj sesiji

Osnovne informacije:	
Datum održavanja sesije	DD. 12. 2023.
Naziv projekta	VVS – projekat (E – Biblioteka)
Verzija dokumenta	v1.0
Broj sati utrošenih za inspekcijsku	2
Članovi inspekcijskog tima	Brčaninović Hasan, Fejzagić Sead, Graca Edwin, Obućina Nermin, Prašović Asmir
Autor koda	Asmir Prašović
Recenzent	<i>Sead Fejzagić</i>
Naziv sekcije koda za recenziju	RecommendationsController - 1

Lista grešaka:

Broj greške	Tip greške	Priroda greške (W/M/E)	Opis greške	Lokacija greške (broj linije u kodu)	Stepen ozbiljnosti greške	Izvor greške (checklista)
1.	error	E	Metoda RecommendationsController.Index() hvata ArgumentException, dok GetAllBooksAsync() baca ArgumentNullException.	32	4	N
2.	defect	W	Dio koda treba izdvojiti iz metode GetBestBooks i staviti u zasebnu metodu GetBooksFromBooksWithRating.	69	2	3
3.	defect	W	U metodi GetBestBooks, potrebno je pomnožiti sortedBooks.Count sa 0.3 kako bi se dobile samo najbolje 30% knjiga.	60	3	N
4.	defect	M	Dodati komentare u kod radi bolje čitljivosti i održavanja koda.	1-107	1	1

Legenda:

Tip greške: error, defekat

Priroda greške: W = Wrong M = Missing E = Extra

Lokacija greške: je opseg linija na kojim se nalazi greška (npr. 15-20 linije ili samo 15 linija).

Stepen ozbiljnosti greške: su vrijednosti definisane u tabeli ozbiljnosti grešaka.

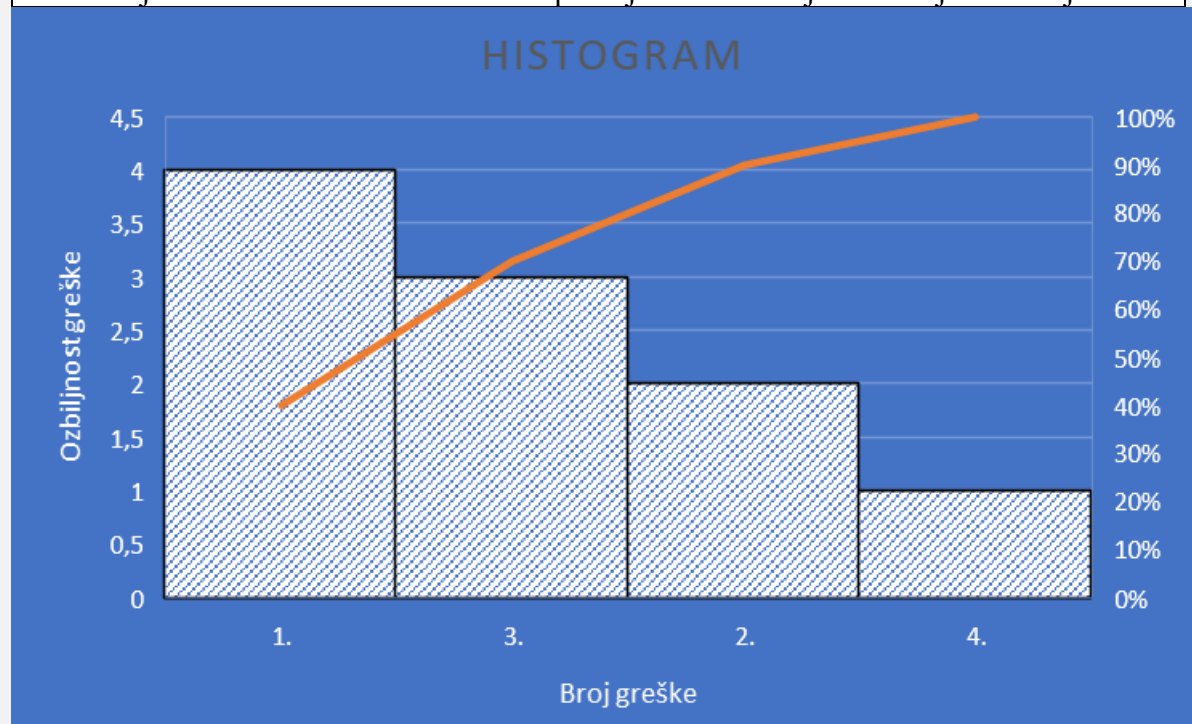
Izvor greške: je broj stavke iz check-liste čijom provjerom je otkrivena greška. U slučaju da izvor nije definisan u listi, u polje napisati slovo N i uvrstiti u postojeću listu.

Naknadne (popratne) akcije:

Broj odluke	Odluka
1	Naknadno praćenje će obaviti: Asmir Prašović
2	Naknadna inspekcija se preporučuje: Da
3	

Komentari:

Izveštaj pokazuje da je kod prošao kroz analizu s fokusom na održivost i čitljivost. Primijećene greške su manje i uglavnom se odnose na strukturu koda i komentare. Ukupna kvaliteta je solidna, ali naglasak na dodavanju komentara i preciznijem definiranju strukture koda može dodatno poboljšati održavanje i saradnju u razvoju.



Izveštaj u inspekcijskoj sesiji

Osnovne informacije:

Datum održavanja sesije	17. 12. 2023.
Naziv projekta	VVS – projekat (E – Biblioteka)
Verzija dokumenta	v1.0
Broj sati utrošenih za inspekcijsku	1

Članovi inspeksijskog tima	Edwin Graca
Autor koda	Sead Fejzagić
Recenzent	<i>Edwin Graca</i>
Naziv sekcije koda za recenziju	LibraryMembersController, BookController

Lista grešaka:

Broj greške	Tip greške	Priroda greške (W/M/E)	Opis greške	Lokacija greške (broj linije u kodu)	Stepen ozbiljnosti greške	Izvor greške (checklista)
1.	defekat	M	Metoda Details ne provjerava u slučaju da libraryMember je null nakon što traži u bazi	66	4	Petlje i kontrole tokova: 3
2.	Defekat	W	Kod učitavanja knjige iz baze podataka u Details nije pravilno postavljen, što može dovesti do problema u dohvaćanju podataka iz baze.	64	4	N

Legenda:

Tip greške: error, defekat

Priroda greške: W = Wrong M = Missing E = Extra

Lokacija greške je opseg linija na kojim se nalazi greška (npr. 15-20 linije ili samo 15 linija).

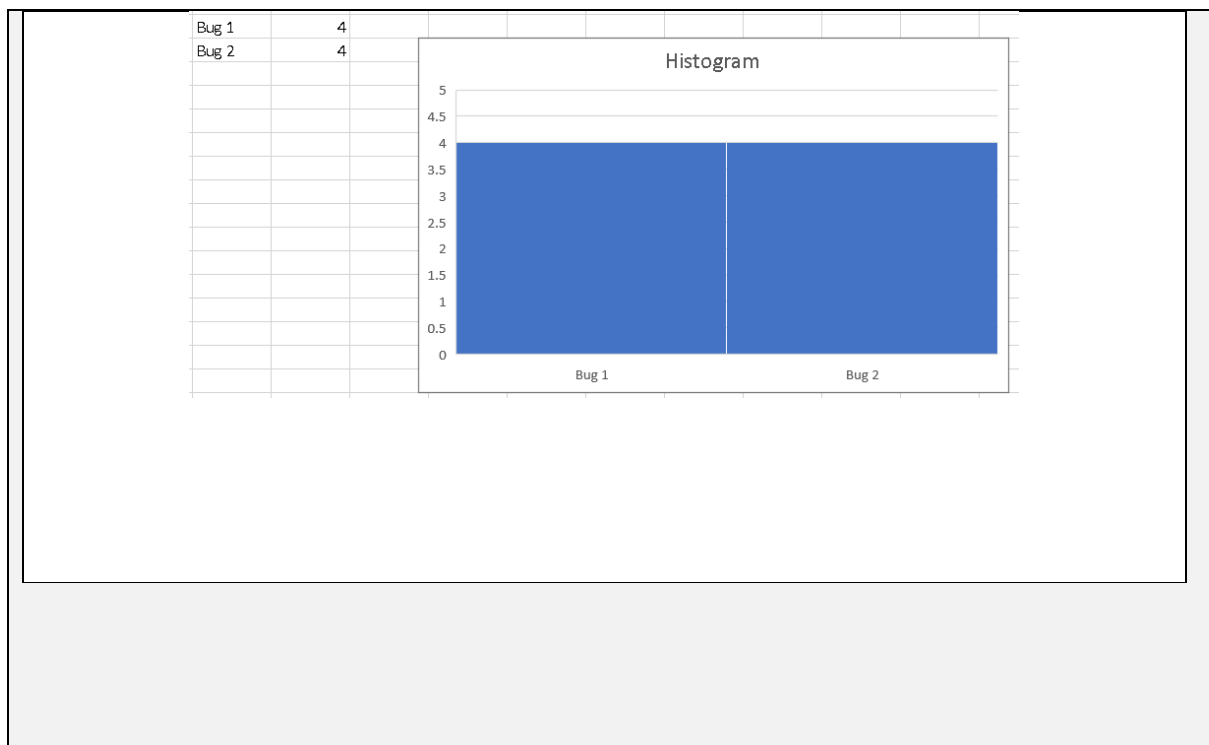
Stepen ozbiljnosti greške su vrijednosti definisane u tabeli ozbiljnosti grešaka.

Izvor greške je broj stavke iz check-liste čijom provjerom je otkrivena greška. U slučaju da izvor nije definisan u listi, u polje napisati slovo N i uvrstiti u postojeću listu.

Naknadne (popratne) akcije:

Broj odluke	Odluka
1	Naknadno praćenje će obaviti: Sead Fejzagić
2	Naknadna inspekcija se preporučuje: Ne
3	

Komentari:



Spisak urađenih korektivnih akcija:

Spisak korektivnih akcija koje su se izvršile su:

- Voucher promjenjen način verifikacije
- GenerateCardNumber ispravljen da generira tačne dužine kartice
- GenerateVoucherCode ispravljen da generira tačne dužine vouchera
- ValidateVoucher metoda promjenjena da provjerava i dužinu vouchera
- GeerateCardNumber preimenovan u GenerateCardNumber
- Promijenio tip izuzetka koji baca metoda GetAllBooksAsync() sa ArgumentNullException na ArgumentException
- Izdvojio dio koda iz metode GetBestBooks() u posebnu metodu GetBooksFromBooksWithRating() radi čitljivosti i organizacije
- Popravio da metoda GetBestBooks() vraća 30% najboljih knjiga a ne sve knjige
- Dodao komentare za RecommendationController
- Problem: U početnom testu, upit za naslov najrezervisanije knjige nije pravilno uzimao u obzir opseg datuma, što je rezultiralo netačnim rezultatom.

- Ispravka: Izmjenjen je upit da pravilno filtrira rezervacije na osnovu navedenog opsega datuma.
- Problem: U dijelu metode MostReservedBookTitle bilo je pokušaja pristupa mostReservedBook.LibraryMemberFk, što nije bilo prisutno u rezultatu upita.
Ispravka: Ispravljen je upit da uključi LibraryMemberId u rezultatu.
- Problem: U testu je tvrdnja za ReservationCount očekivala 1, ali ispravljeni upit je brojao rezervacije po članu biblioteke.
Ispravka: Ažurirana je tvrdnja testa da očekuje 0 rezervacija, prema dostupnim podacima testa.
- Problem: Test je očekivao određeni naslov knjige, ali test podaci i ispravljeni upit sugerisali su da nema rezervacija u navedenom opsegu datuma.
Ispravka: Prilagođena je tvrdnja testa da se poklapa s očekivanjem nema rezervacija.
- Izvršena Popravka: Unaprijeđena je stabilnost aplikacije dodavanjem provjere na libraryMember nakon upita u bazi podataka. Sada se provjerava da li je libraryMember null, a ako jeste, vraća se NotFound rezultat. Ova promjena sprječava potencijalne greške prilikom pokušaja pristupa svojstvima null objekta.
- Izvršena Popravka: Ispravljena je greška u kodu koja se odnosila na postavljanje uslova pri učitavanju knjige iz baze podataka. Sada je uslov u FirstOrDefaultAsync ispravno postavljen kao „==“ umjesto „!=“, čime se osigurava pravilno uspoređivanje sa id. Ova korekcija rješava problem u dohvaćanju podataka iz baze i poboljšava integritet sistema.

Task 3 – Unit testiranje

Spisak Unit testova kao i njihov tip koje su napisali članovi tima su pobrojani u nastavku:

- BookControllerTests (mock)
- LibraryMembersControllerTests (mock)
- ReservationControllerTests (mock)
- MembershipPaymentTest (mock)
- CardTest (Unit)
- Index_ReturnsViewResult (Unit)
- GetAllBooksAsync_ReturnsListOfBooks (Unit)
- CalculateAverageRatings_ReturnsCorrectAverage (Unit)
- GetBestBooks_ThrowsException (Unit)
- GetBestBooks_ThrowsExceptionWithProperMessage (Unit)
- BookTest1 (Unit)
- MostReservedBookTitle_ReturnsCorrectResult (Mock)
- CountReservation_MultipleReservations_ReturnsCorrectCount(Mock)
- LateActivatedReservations_ActivatedReservationsAfterIssuedDate_ReturnsCorrectReservations(Mock)

Task 4 – TDD

Za ovaj zadatak je iskorištena funkcionalnost Recommended sistem. Sve testove je pisao Asmir Prašović koji prate TDD faze. On je u isto vrijeme imao ulogu i pisca testova i pisca koda.

Spisak testova koji su iskorišteni za programiranje ove funkcionalnosti su:

- `Index_ReturnsViewResult()`
- `GetAllBooksAsync_ReturnsListOfBooks()`
- `CalculateAverageRatings_ReturnsCorrectAverage()`
- `GetBestBooks_ThrowsException()`
- `GetBestBooks_ThrowsExceptionWithProperMessage()`

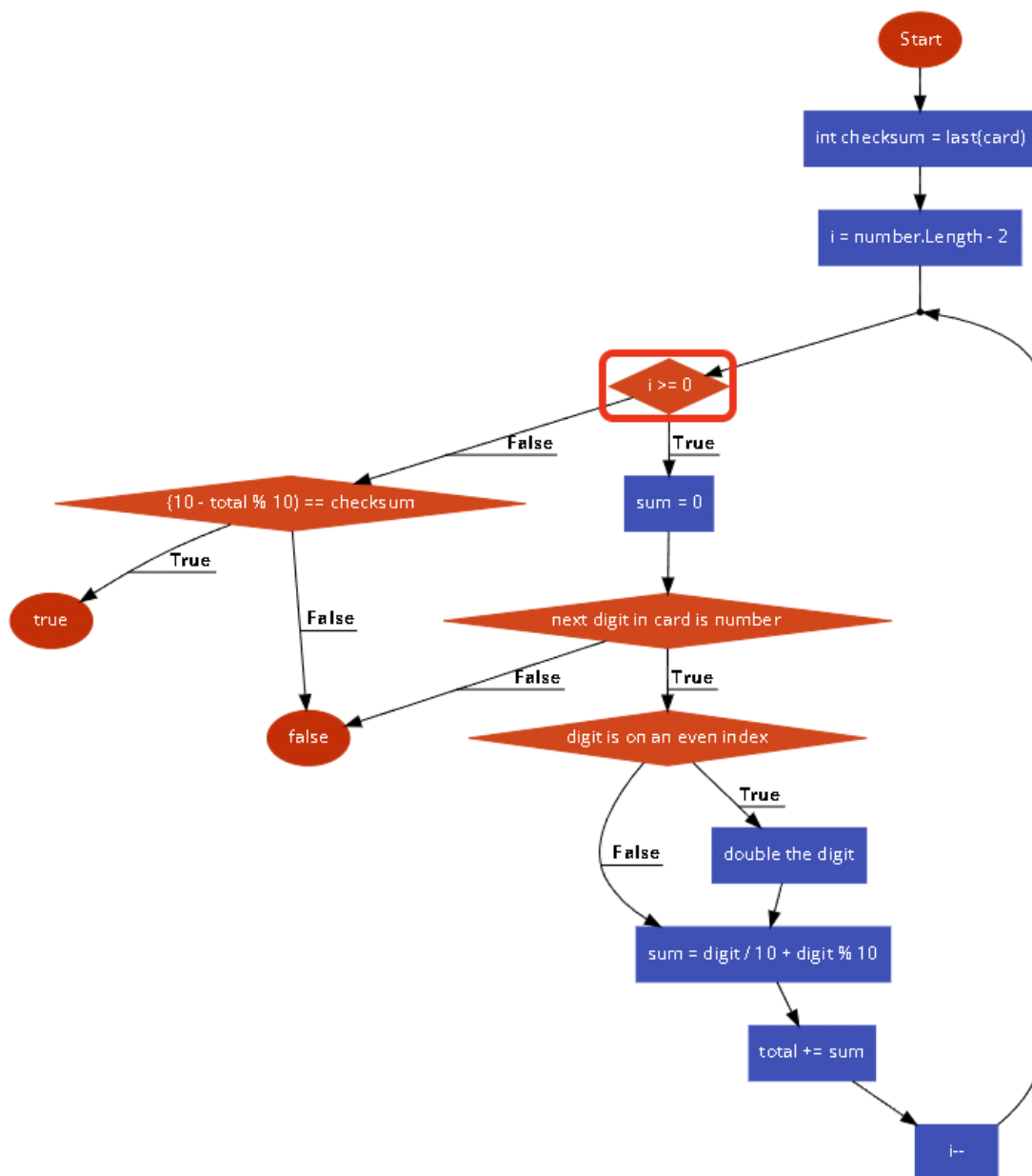
White box testiranje

U nastavku su predstavljeni izvještaji metoda koje su se testirale white box metodom.

Izvještaj o White Box testiranju

Metoda CardIsValid(string card)

Kontroler MembershipPaymentController



Testno Okruženje:

Klasa: MembershipPaymentController

Metoda: CardisValids

Pristupi Testiranju:

Pokrivanje Putanje (Path Coverage)

Pokrivanje Uslova (Condition Coverage)

Pokrivanje Petlji (Loop Coverage)

1. Pokrivanje Putanje:

Testni Slučaj 1: Metoda radi ispravno kada je validna kartina unošena.

1. Priprema: Kreiranje broj kartice sa pravilnom strukturom.
2. Akcija: Pozivanje metode CardisValid.
3. Provjera: Verifikacija da je rezultat bool true.

Testni Slučaj 2: Metoda vraća false kada je unutar string-a nebrojčani literal.

1. Priprema: Kreiranje stringa sa slovom u sredini.
2. Akcija: Pozivanje metode CardisValid.
3. Provjera: Verifikacija da je rezultat bool false.

2. Pokrivanje Uslova:

Testni Slučaj 1: Provjera za ispravnu zadnju cifru broja kartice.

1. Priprema: Kreiranje broj kartice sa pravilnom strukturom.
2. Akcija: Pozivanje metode CardisValid.
3. Provjera: Verifikacija da je rezultat bool true.

Testni Slučaj 2: Provjera za nevalidan zadnji broj kartice.

1. Priprema: Kreiranje broj kartice sa pravilnom strukturompa promjenit zadnju cifru.
2. Akcija: Pozivanje metode CardisValid.
3. Provjera: Verifikacija da je rezultat bool false.

3. Pokrivanje Petlji:

Testni Slučaj 1: Provjera Ponašanja For Petlje kada je prazan string.

1. Priprema: Kreiranje praznan string kartice.
2. Akcija: Pozivanje metode CardisValid.
3. Provjera: Verifikacija da je rezultat metode bool false.

Testni Slučaj 2: Provjera Ponašanja Foreach Petlje kada ima tačno jedan literal u stringu.

1. Priprema: Kreiranje string kartice sa jednim literalom.
2. Akcija: Pozivanje metode CardisValid.
3. Provjera: Verifikacija da je rezultat metode bool false.

Zaključak:

Testovi white-box metodom za metodu CardisValid su detaljno pokrili različite dijelove koda, prateći pristupe pokrivanja putanje, uslova i petlji. Implementirani testni slučajevi su usmjereni ka osiguravanju da metoda pravilno reaguje na različite ulazne scenarije i vraća očekivane rezultate.

Rezultati ukazuju na to da je metoda robusna, ispravna i da se ponaša kako se očekuje u različitim situacijama. Nisu uočeni problemi ili neočekivana ponašanja tokom testiranja, potvrđujući kvalitetu implementacije metode CardisValid unutar MembershipPaymentController-a. Ova testiranja pružaju osnovu za pouzdanu i sigurnu funkcionalnost metode u realnom svijetu.

Izvještaj o White Box testiranju
Metoda MostReservedBookTitle(DateTime startDate, DateTime endDate)
Kontroler BookController

Testno Okruženje:

Klasa: BookController
Metoda: MostReservedBookTitle
Pristupi Testiranju:
Pokrivanje Putanje (Path Coverage)
Pokrivanje Uslova (Condition Coverage)
Pokrivanje Petlji (Loop Coverage)

1. Pokrivanje Putanje(Path Coverage):

Testni Slučaj 1: Metoda vraća ispravne rezultate kada postoje rezervacije u navedenom vremenskom okviru.

1. Priprema: Unos valjanih datuma startDate i endDate koji obuhvaćaju stvarne rezervacije.
2. Akcija: Pozivanje metode MostReservedBookTitle.
3. Provjera: Verifikacija rezultata koji sadrži ispravno ime najrezerviranije knjige.

Testni Slučaj 2: Metoda vraća ispravnu poruku kada nema rezervacija u navedenom vremenskom okviru.

1. Priprema: Unos valjanih datuma startDate i endDate koji ne obuhvaćaju rezervacije.
2. Akcija: Pozivanje metode MostReservedBookTitle.
3. Provjera: Verifikacija rezultata koji sadrži poruku "No reservations found in the specified date range".

Testni slučaj 3: Metoda vraća ispravnu poruku kada je rezultat prazan zbog neispravnog statusa rezervacija.

1. Priprema: Unos valjanih datuma startDate i endDate s rezervacijama koje imaju status koji nije "Book".
2. Akcija: Pozivanje metode MostReservedBookTitle.
3. Provjera: Verifikacija rezultata koji sadrži poruku "No reservations found in the specified data range".

2. Pokrivanje Uslova (Condition Coverage):

Testni Slučaj 1: Provjera za ispravne datume rezervacija.

1. Priprema: Unos valjanih datuma startDate i endDate.
2. Akcija: Pozivanje metode MostReservedBookTitle.
3. Provjera: Verifikacija izvršavanja upita koji obuhvaća rezervacije s ispravnim datumima.

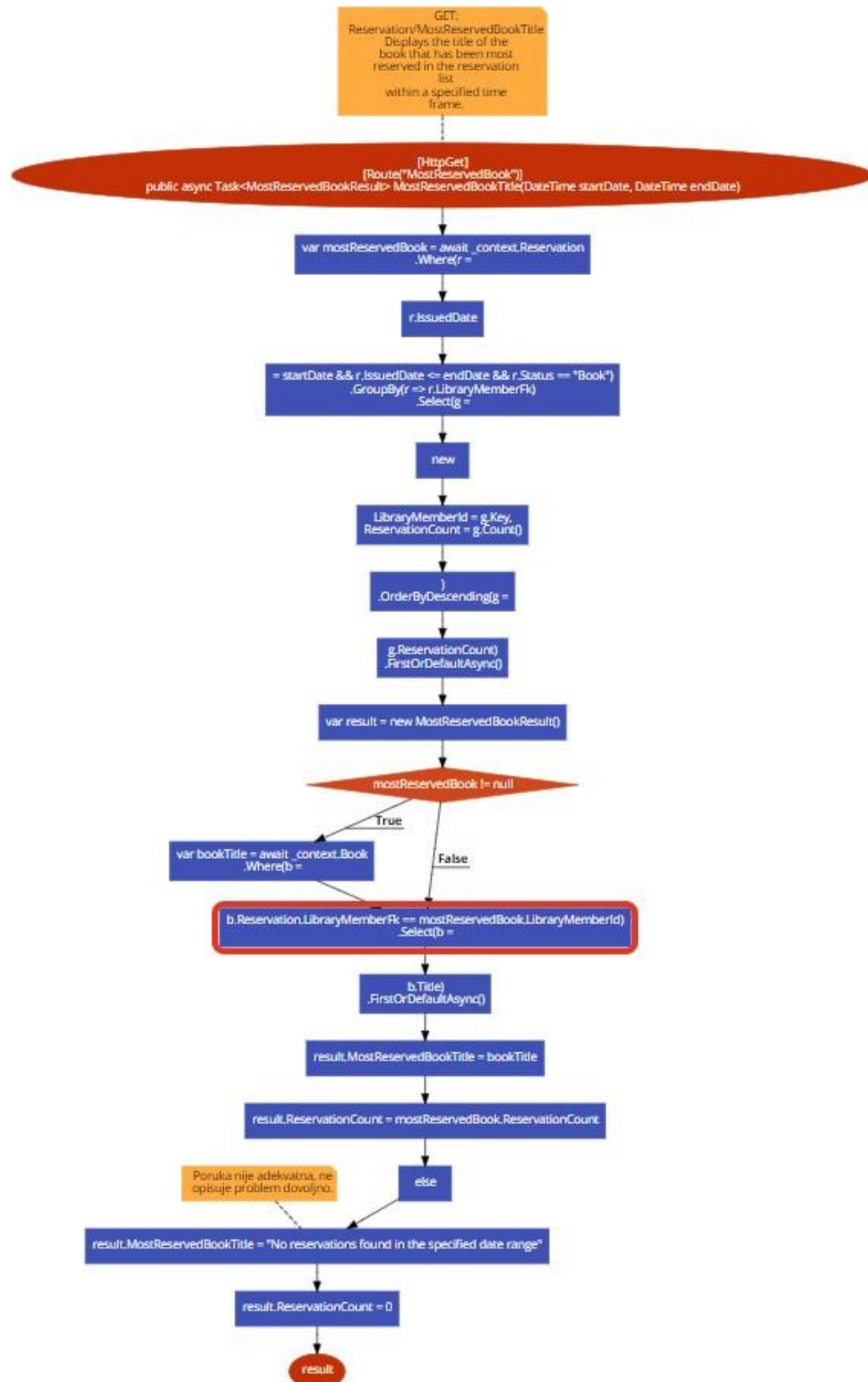
Testni Slučaj 2: Provjera za neispravne datume rezervacija.

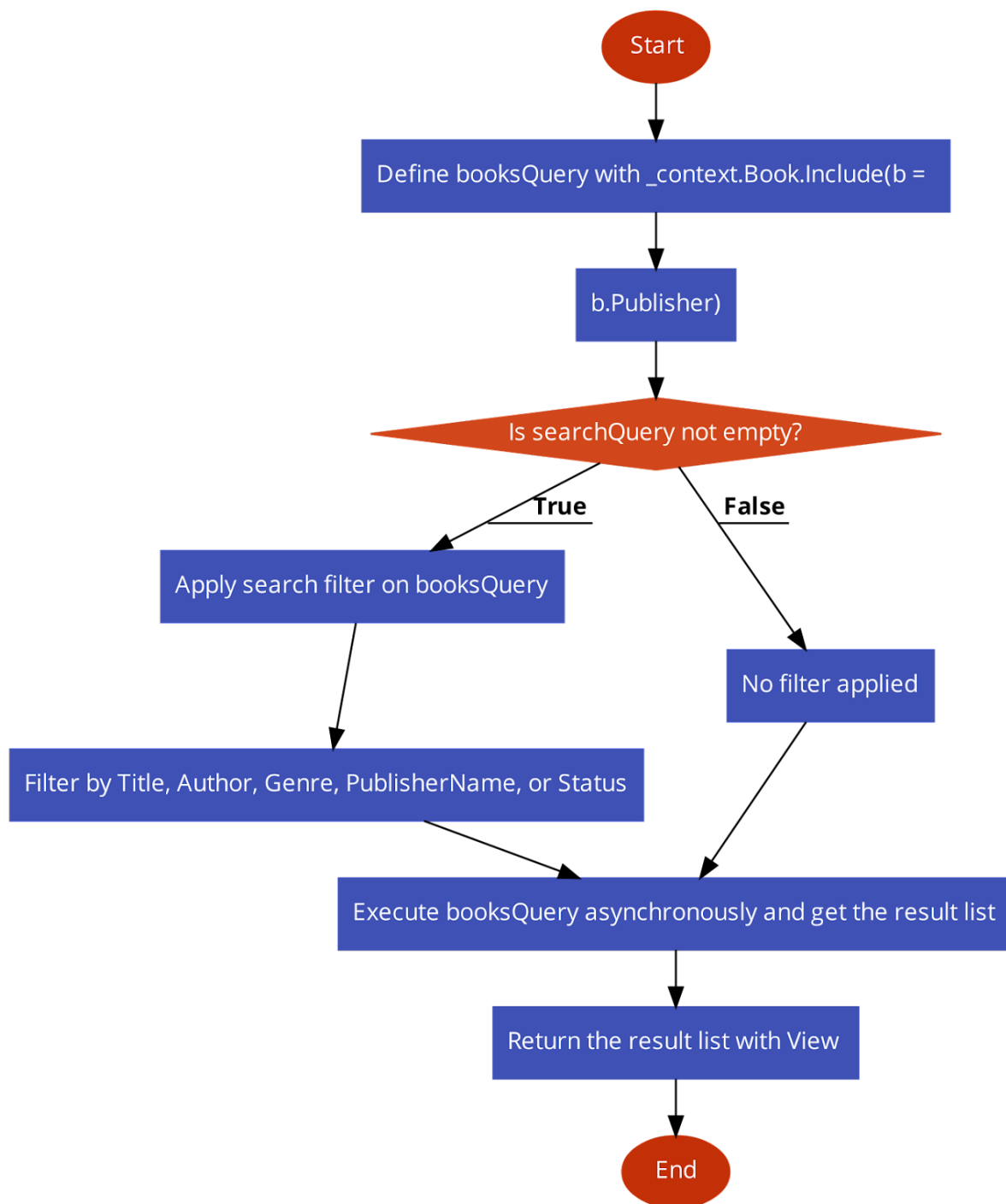
1. Priprema: Unos nevaljanih datuma startDate i endDate (npr., startDate > endDate).
2. Akcija: Pozivanje metode MostReservedBookTitle.
3. Provjera: Verifikacija da metoda ne izvršava upit i vraća poruku o neispravnosti.

Zaključak:

Testiranje metode MostReservedBookTitle u ReservationController primjenom white-box pristupa pokazalo se uspješnim u pokrivanju različitih putanja izvršavanja, uvjeta i obrade iznimki. Implementirani testni slučajevi bili su usmjereni na osiguravanje ispravnog ponašanja metode u raznim situacijama. Nisu uočeni problemi ili neočekivana ponašanja tokom testiranja, potvrđujući kvalitetu implementacije metode MostReservedBookTitle.

Ova vrsta testiranja pruža detaljan uvid u ponašanje metode, čime se osigurava funkcionalna ispravnost i robusnost aplikacije. Ovaj pristup testiranju služi kao osnova za pouzdanu funkcionalnost i pruža sigurnost da će se metoda ispravno ponašati u stvarnom radnom okruženju.





Izveštaj o White box za Metodu Index

Testno Okruženje

- Kontroler: BookController
- Metoda: Index

Pristupi Testiranju

1. Pokrivanje Putanje (Path Coverage)
2. Pokrivanje Uslova (Condition Coverage)

3. Pokrivanje Petlji (Loop Coverage)

1. Pokrivanje Putanje

Testni Slučaj 1: Dohvati Sve Knjige kada je Pretraga Prazna

- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index** bez pretrage.
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži očekivan broj knjiga (2).

Testni Slučaj 2: Filtriraj Knjige na Osnovu Pretrage

- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index** sa pretragom "Book 1".
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži očekivan broj filtriranih knjiga (1).

Testni Slučaj 3: Vрати Prazan Rezultat kada Pretraga Ne Nalazi Podudaranje

- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index** sa pretragom "Ne postoji knjiga".
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži 0 knjiga.

Testni Slučaj 4: Vрати ViewResult

- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index**.
- Provjera: Verifikacija da je rezultat ViewResult.

2. Pokrivanje Uslova

Testni Slučaj 1: Provjeri za Null ili Praznu Pretragu

- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index** sa null pretragom.
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži očekivan broj knjiga (2).

Testni Slučaj 2: Provjeri za Pretragu bez Razlikovanja Velikih i Malih Slova

- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index** sa pretragom "zanr 1".
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži očekivan broj filtriranih knjiga (1).

3. Pokrivanje Petlji

Testni Slučaj 1: Pretraga koja Podudara sa Više Knjiga

- Priprema: Kreiranje kontrolera **BookController**.

- Akcija: Pozivanje metode **Index** sa pretragom "Book".
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži očekivan broj knjiga (2).

Testni Slučaj 2: Pretraga koja Podudara sa Jednom Knjigom

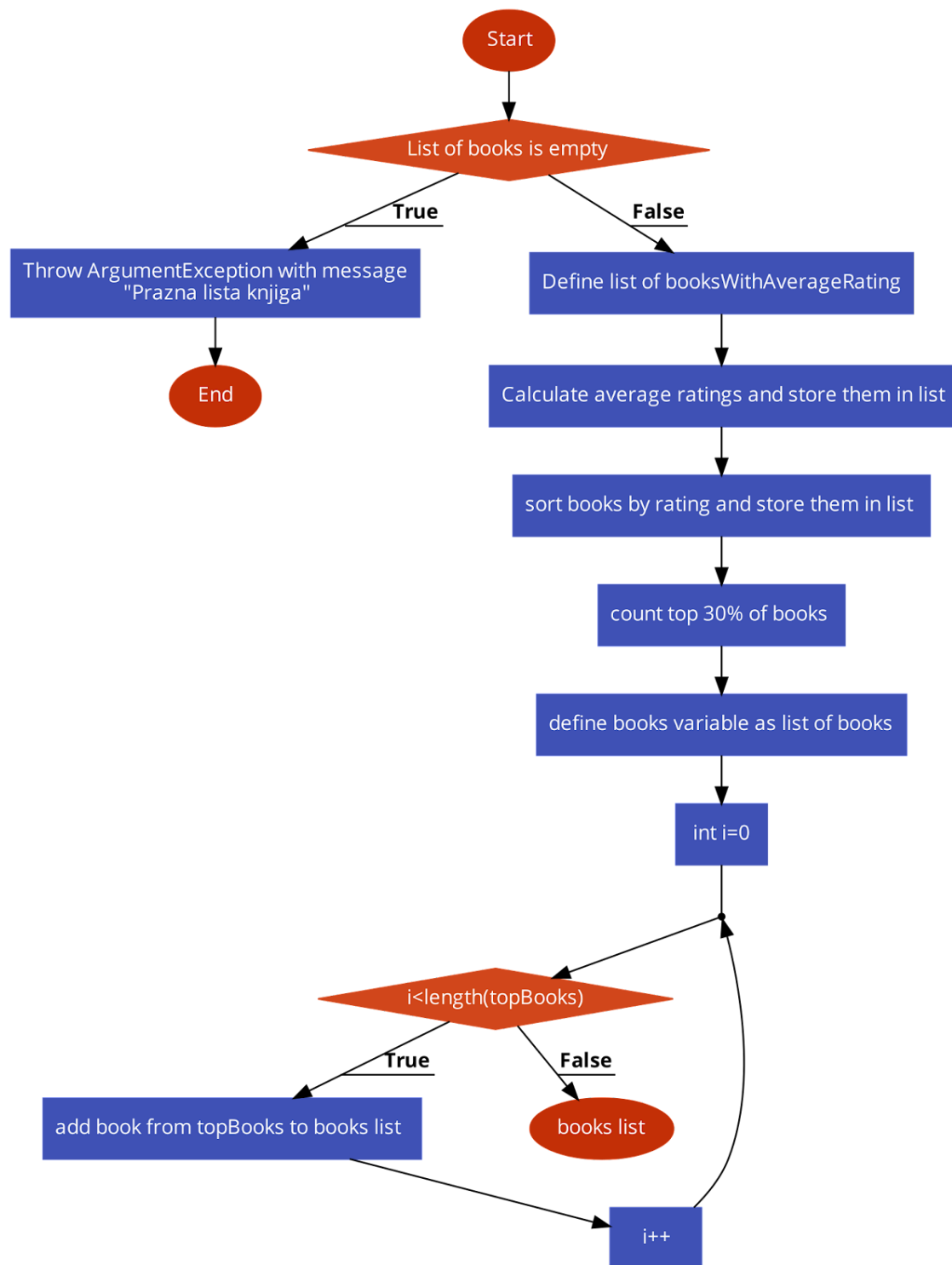
- Priprema: Kreiranje kontrolera **BookController**.
- Akcija: Pozivanje metode **Index** sa pretragom "Book 1".
- Provjera: Verifikacija da je rezultat ViewResult, da model nije null i sadrži očekivanu jednu knjigu.

Rezime i Zaključak

U ovom testiranju white box metodom primijenili smo tri pristupa: pokrivanje putanje, pokrivanje uslova i pokrivanje petlji. Implementirani testni slučajevi obuhvataju različite scenarije, uključujući dohvaćanje svih knjiga, filtriranje na osnovu pretrage i upravljanje različitim uslovima. Testovi imaju za cilj osigurati tačnost i robusnost metode **Index** u **BookController**.

Rezultati pokazuju da metoda **Index** radi onako kako se očekuje u različitim scenarijima, bez primijećenih problema ili neočekivanih ponašanja.

Izvještaj o White Box testiranju
 Metoda GetBestBooks(List<Book> allBooks)
 Kontroler RecommendationsController



Testno Okruženje:

Klasa: RecommendationsController

Metoda: GetBestBooks

Pristupi Testiranju:

Pokrivanje Putanje (Path Coverage)

Pokrivanje Uslova (Condition Coverage)

Pokrivanje Petlji (Loop Coverage)

1. Pokrivanje Putanje:

Testni Slučaj 1: Metoda radi ispravno kada postoji barem jedna knjiga u listi.

1. Priprema: Kreiranje kontrolera RecommendationsController sa nepraznom listom knjiga.
2. Akcija: Pozivanje metode GetBestBooks.
3. Provjera: Verifikacija da je rezultat lista knjiga i da nije null.

Testni Slučaj 2: Metoda baca ArgumentException kada je lista knjiga prazna.

1. Priprema: Kreiranje kontrolera RecommendationsController sa praznom listom knjiga.
2. Akcija: Pozivanje metode GetBestBooks.
3. Provjera: Verifikacija da je bačen izuzetak tipa ArgumentException.

2. Pokrivanje Uslova:

Testni Slučaj 1: Provjera za Praznu Listu Knjiga.

1. Priprema: Kreiranje kontrolera RecommendationsController sa praznom listom knjiga.
2. Akcija: Pozivanje metode GetBestBooks.
3. Provjera: Verifikacija da je rezultat lista knjiga i da je prazna.

Testni Slučaj 2: Provjera za Listu Knjiga sa Jednom Knjigom.

1. Priprema: Kreiranje kontrolera RecommendationsController sa listom koja sadrži samo jednu knjigu.
2. Akcija: Pozivanje metode GetBestBooks.
3. Provjera: Verifikacija da je rezultat lista knjiga i da sadrži tu jednu knjigu.

3. Pokrivanje Petlji:

Testni Slučaj 1: Provjera Ponašanja Foreach Petlje kada Postoji Više Knjiga.

1. Priprema: Kreiranje kontrolera RecommendationsController sa listom koja sadrži više od jedne knjige.
2. Akcija: Pozivanje metode GetBestBooks.
3. Provjera: Verifikacija da je rezultat lista knjiga i da sadrži očekivan broj knjiga.

Testni Slučaj 2: Provjera Ponašanja Foreach Petlje kada Postoji Samo Jedna Knjiga.

1. Priprema: Kreiranje kontrolera RecommendationsController sa listom koja sadrži samo jednu knjigu.
2. Akcija: Pozivanje metode GetBestBooks.
3. Provjera: Verifikacija da je rezultat lista knjiga i da sadrži tu jednu knjigu.

Zaključak:

Testovi white-box metodom za metodu GetBestBooks su detaljno pokrili različite dijelove koda, prateći pristupe pokrivanja putanje, uslova i petlji. Implementirani testni slučajevi su usmjereni ka osiguravanju da metoda pravilno reaguje na različite ulazne scenarije i vraća očekivane rezultate.

Rezultati ukazuju na to da je metoda robusna, ispravna i da se ponaša kako se očekuje u različitim situacijama. Nisu uočeni problemi ili neočekivana ponašanja tokom testiranja, potvrđujući kvalitetu implementacije metode GetBestBooks unutar RecommendationsController-a. Ova testiranja pružaju osnovu za pouzdanu i sigurnu funkcionalnost metode u realnom svijetu.