

Izvještaj o Inspekciji Koda

RestaurantInventory - Tim 4

Informacije o Inspekciji

Datum inspekcije:	17. 11. 2025.
Moderator inspekcije:	Bakir Činjurević
Tim:	Tim 4
Aplikacija:	RestaurantInventory
Verzija izvještaja:	1.0

Sadržaj

1	Uvod	4
1.1	Pregled Aplikacije	4
1.2	Statistika Koda	4
2	Checklist za Inspekciju	4
3	Kategorizacija Problema	5
3.1	Kritični Problemi	5
3.1.1	ISSUE-001: NullPointerException u InventarService.DodajStavku	5
3.1.2	ISSUE-002: Nevalidiran unos cijene može baciti FormatException .	6
3.1.3	ISSUE-003: Logička greška u StavkaInventaraService.JeKriticna . .	6
3.2	Visoki Problemi	6
3.2.1	ISSUE-004: Duplikacija koda - FilterService i InventarService.Pretraži	6
3.2.2	ISSUE-005: Nedostaje validacija inputa na više mjesta	7
3.2.3	ISSUE-006: Nedostaje logging	7
3.2.4	ISSUE-007: Nedostaju unit testovi	7
3.2.5	ISSUE-008: Nedostaje dokumentacija	8
3.2.6	ISSUE-009: Nekonzistentno imenovanje (miješanje jezika)	8
3.2.7	ISSUE-010: Hardcoded vrijednosti (magic numbers)	8
3.3	Srednji Problemi	9
3.3.1	ISSUE-011: Nedostaje dependency injection	9
3.3.2	ISSUE-012: Nema konfiguracijskog fajla	9
3.3.3	ISSUE-013: Edge case - više stavki sa istim nazivom	9
3.4	Niski Problemi	10
3.4.1	ISSUE-014: Nekonzistentno formatiranje	10
3.4.2	ISSUE-015: Nekorišćena klasa FilterService	10
4	Pregled Svih ISSUE-ova	11
5	Pareto Analiza	11
5.1	Pareto Analiza po Uzrocima	11
5.2	Pareto Analiza po Defektima	13
5.3	Analiza Prioriteta	15
6	Dobre Prakse	16
7	Preporuke za Poboljšanje	16
7.1	Kratkoročne (Sprint 1)	16
7.2	Srednjoročne (Sprint 2-3)	16
7.3	Dugoročne (Sprint 4+)	17
8	Komentari za Kod	17
8.1	Komentari za Program.cs	17
8.1.1	Komentar 1: Linija 134 - ISSUE-002	17
8.1.2	Komentar 2: Linija 83-87 - ISSUE-005	17
8.1.3	Komentar 3: Linija 221-224 - ISSUE-005, ISSUE-013	18
8.1.4	Komentar 4: Linija 236 - ISSUE-013	19
8.1.5	Komentar 5: Linija 288-294 - ISSUE-013	19

8.2	Komentari za <code>Services/InventarService.cs</code>	20
8.2.1	Komentar 6: Linija 22-24 - ISSUE-001	20
8.2.2	Komentar 7: Linija 47-82 - ISSUE-004	20
8.3	Komentari za <code>Services/StavkaInventaraService.cs</code>	21
8.3.1	Komentar 8: Linija 8-24 - ISSUE-003	21
8.3.2	Komentar 9: Linija 12, 21 - ISSUE-010	22
8.4	Komentari za <code>Services/FilterService.cs</code>	22
8.4.1	Komentar 10: Cijeli fajl - ISSUE-004, ISSUE-015	22
9	Checklist za Pull Request	22
10	Zaključak	23

1 Uvod

Ovaj dokument predstavlja izvještaj o inspekciji koda aplikacije **RestaurantInventory**, konzolne aplikacije za upravljanje inventarom restorana. Inspekcija je provedena u sklopu laboratorijske vježbe 4, a cilj je bio identifikovati probleme u kodu, kategorizirati ih prema ozbiljnosti i izvršiti Pareto analizu uzroka i defekata.

1.1 Pregled Aplikacije

Aplikacija **RestaurantInventory** omogućava:

- Dodavanje stavki u inventar
- Pretraživanje i filtriranje stavki
- Ažuriranje postojećih stavki
- Uklanjanje stavki iz inventara
- Generisanje izvještaja

1.2 Statistika Koda

Metrika	Vrijednost
Ukupno linija koda	~600
Broj klasa	10
Broj servisa	4
Broj modela	6

Tablica 1: Osnovne metrike koda

2 Checklist za Inspekciju

Inspekcija je provedena korištenjem checklist-a sa 18 kategorija. Svaki problem je mapiran na odgovarajuće checklist stavke sa statusom **FAILED** ili **PASSED**.

ID	Kategorija	Status
CL-01	Validacija Inputa	FAILED
CL-02	Error Handling	FAILED
CL-03	Null Safety	FAILED
CL-04	Logika Aplikacije	FAILED
CL-05	DRY Princip	FAILED
CL-06	Održivost Koda	FAILED
CL-07	Konfiguracija	FAILED
CL-08	Dokumentacija	FAILED
CL-09	Testiranje	FAILED
CL-10	Logging	FAILED
CL-11	Arhitektura	FAILED
CL-12	Konvencije Imenovanja	FAILED
CL-13	Stil Koda	FAILED
CL-14	Performance	PASSED
CL-15	Sigurnost	PASSED
CL-16	Edge Cases	FAILED
CL-17	User Experience	FAILED
CL-18	Code Quality Tools	FAILED

Tablica 2: Checklist kategorije i status

3 Kategorizacija Problema

Tokom inspekcije koda, identifikovano je **15 problema** koji su kategorizirani prema ozbiljnosti:

Ozbiljnost	Broj	Postotak
Kritični (Critical)	3	20%
Visoki (High)	7	47%
Srednji (Medium)	3	20%
Niski (Low)	2	13%
Ukupno	15	100%

Tablica 3: Raspodjela problema po ozbiljnosti

3.1 Kritični Problemi

3.1.1 ISSUE-001: NullPointerException u InventarService.DodajStavku

Lokacija: Services/InventarService.cs:22-24

Ozbiljnost: **Kritična**

Prioritet: P1 (Kritično)

Checklist Reference:

- **CL-01: Validacija Inputa** → **FAILED** - Nedostaje provjera null za Dobavljac
- **CL-03: Null Safety** → **FAILED** - Nema provjere null prije poziva Equals

Problem: Ako `stavka.Dobavljac` ili postojeća stavka u listi ima `null` vrijednost za `Dobavljac`, poziv `Equals` će baciti `NullReferenceException`.

Rješenje: Dodati `null`-checking prije poziva `Equals`:

```
if (inventar.Stavke.Any(s =>
    string.Equals(s.Naziv, stavka.Naziv, StringComparison.OrdinalIgnoreCase) &&
    string.Equals(s.Dobavljac ?? "", stavka.Dobavljac ?? "",
        StringComparison.OrdinalIgnoreCase)))
```

3.1.2 ISSUE-002: Nevalidiran unos cijene može baciti `FormatException`

Lokacija: `Program.cs:134`

Ozbiljnost: **Kritična**

Prioritet: P1 (Kritično)

Checklist Reference:

- **CL-01: Validacija Inputa** → **FAILED** - Nedostaje validacija formata broja
- **CL-02: Error Handling** → **FAILED** - `FormatException` nije uhvaćena

Problem: Ako korisnik unese nevalidan string (npr. "abc"), `double.Parse` će baciti `FormatException` koja nije uhvaćena.

Rješenje: Koristiti `double.TryParse` umjesto `double.Parse`.

3.1.3 ISSUE-003: Logička greška u `StavkaInventaraService.JeKriticna`

Lokacija: `Services/StavkaInventaraService.cs:8-24`

Ozbiljnost: **Kritična**

Prioritet: P1 (Kritično)

Checklist Reference:

- **CL-04: Logika Aplikacije** → **FAILED** - Postoji unreachable code i logička greška

Problem: Metoda `JeKriticna` ima logičku grešku. Ako je `stavka.Kolicina < stavka.MinKolicina`, metoda uvijek vraća `true` bez obzira na ostale uslove (linija 18 je unreachable code).

Rješenje: Refaktorisati logiku da bude jasnija i ispravna.

3.2 Visoki Problemi

3.2.1 ISSUE-004: Duplikacija koda - `FilterService` i `InventarService.Pretrazi`

Lokacija: `Services/FilterService.cs` i `Services/InventarService.cs:47-82`

Ozbiljnost: **Visoka**

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-05: DRY Princip** → **FAILED** - Ista logika filtriranja na dva mjesta

Problem: Ista logika filtriranja je implementirana na dva mjesta. `InventarService.Pretrazi` i `FilterService.PrimijeniFilter` imaju identičnu implementaciju.

Rješenje: Koristiti `FilterService.PrimijeniFilter` unutar `InventarService.Pretrazi` umjesto duplikacije.

3.2.2 ISSUE-005: Nedostaje validacija inputa na više mjesta

Lokacija: `Program.cs` (više metoda)

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-01: Validacija Inputa** → **FAILED** - Nema provjere da li su naziv, opis, tip prazni

Problem: Nedostaje validacija korisničkog inputa na više mjesta:

- `DodajStavku`: Nema provjere da li su naziv i opis prazni
- `UkloniStavku`: Nema provjere da li je naziv prazan
- `AzurirajStavku`: Nema provjere da li je naziv prazan
- `GenerisiIzvjestaj`: Nema provjere da li je tip prazan

Rješenje: Dodati validaciju na svim mjestima gdje se prima korisnički input.

3.2.3 ISSUE-006: Nedostaje logging

Lokacija: Cijela aplikacija

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-10: Logging** → **FAILED** - Aplikacija nema logging mehanizam

Problem: Aplikacija nema logging mehanizam. Nemoguće je pratiti šta se dešava u produkciji, debugovati probleme ili pratiti korisničke akcije.

Rješenje: Integrirati logging framework (npr. Serilog, NLog) i dodati logove za dodavanje/uklanjanje/ažuriranje stavki, greške i izuzetke, te važne operacije.

3.2.4 ISSUE-007: Nedostaju unit testovi

Lokacija: Cijeli projekt

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-09: Testiranje** → **FAILED** - Projekt ne sadrži unit testove

Problem: Projekt ne sadrži unit testove. Nemoguće je automatski provjeriti ispravnost koda ili osigurati da promjene ne razbiju postojeću funkcionalnost.

Rješenje: Kreirati test projekt, dodati unit testove za sve servise, i integrirati u CI/CD pipeline.

3.2.5 ISSUE-008: Nedostaje dokumentacija

Lokacija: Cijeli projekt

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-08: Dokumentacija** → **FAILED** - Nema XML komentara, README.md fajla, ni dokumentacije o arhitekturi

Problem: Projektu nedostaje dokumentacija:

- Nema XML komentara na javnim metodama i klasama
- Nema README.md fajla
- Nema dokumentacije o arhitekturi

Rješenje: Dodati XML komentare na sve javne API-je, kreirati README.md sa uputstvima za instalaciju i korištenje, te dokumentovati arhitekturu i dizajn odluke.

3.2.6 ISSUE-009: Nekonzistentno imenovanje (miješanje jezika)

Lokacija: Cijeli projekt

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-12: Konvencije Imenovanja** → **FAILED** - Kod miješa hrvatski i engleski jezik

Problem: Kod miješa hrvatski i engleski jezik:

- Hrvatski: DodajStavku, PrikazSaSortiranjem, ProvjeriMinimalne
- Engleski: InventarService, StavkaInventara, Filter

Rješenje: Usvojiti jedan jezik (preporučeno engleski) za sve nazive u kodu, a hrvatski koristiti samo za UI poruke.

3.2.7 ISSUE-010: Hardcoded vrijednosti (magic numbers)

Lokacija: Services/StavkaInventaraService.cs:12, 21

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist Reference:

- **CL-07: Konfiguracija** → **FAILED** - Magic numbers 30 i 60 dana su hardcoded

Problem: U kodu postoje hardcoded vrijednosti (magic numbers):

- `DateTime.Now.AddDays(-30)` - kritični period dana
- `TotalDays > 60` - maksimalni period dana

Rješenje: Premjestiti u konstante ili konfiguraciju:

```
private const int KRITICNLPERIOD_DANA = 30;  
private const int MAKSIMALNLPERIOD_DANA = 60;
```

3.3 Srednji Problemi

3.3.1 ISSUE-011: Nedostaje dependency injection

Lokacija: `Program.cs:15`

Ozbiljnost: Srednja

Prioritet: P3 (Srednje)

Checklist Reference:

- **CL-11: Arhitektura** → **FAILED** - Direktno instanciranje servisa u `Main` metodi

Problem: Direktno instanciranje servisa u `Main` metodi otežava testiranje i zamjenu implementacija.

Rješenje: Integrirati DI container (npr. `Microsoft.Extensions.DependencyInjection`).

3.3.2 ISSUE-012: Nema konfiguracijskog fajla

Lokacija: Cijeli projekt

Ozbiljnost: Srednja

Prioritet: P3 (Srednje)

Checklist Reference:

- **CL-07: Konfiguracija** → **FAILED** - Svi parametri su hardcoded u kodu

Problem: Svi parametri su hardcoded u kodu. Nemoguće je mijenjati ponašanje aplikacije bez rekompajliranja.

Rješenje: Dodati `appsettings.json` i koristiti `IConfiguration`.

3.3.3 ISSUE-013: Edge case - više stavki sa istim nazivom

Lokacija: `Program.cs:236`

Ozbiljnost: Srednja

Prioritet: P3 (Srednje)

Checklist Reference:

- **CL-16: Edge Cases** → **FAILED** - Nema provjere za više stavki sa istim nazivom

- **CL-17: User Experience** → **FAILED** - Korisnik nije upozoren

Problem: U AzurirajStavku, ako pretraga vrati više stavki sa istim nazivom, uzima se samo prva (`FirstOrDefault`), što može biti neočekivano ponašanje.

Rješenje: Dodati provjeru i upozorenje ako postoji više stavki sa istim nazivom.

3.4 Niski Problemi

3.4.1 ISSUE-014: Nekonzistentno formatiranje

Lokacija: Različite lokacije

Ozbiljnost: Niska

Prioritet: P4 (Nisko)

Checklist Reference:

- **CL-13: Stil Koda** → **FAILED** - Nekonzistentno korištenje praznih linija i indentacije

Problem: Nekonzistentno korištenje praznih linija i indentacije kroz kod.

Rješenje: Konfigurisati `.editorconfig` i primijeniti auto-formatiranje.

3.4.2 ISSUE-015: Nekorišćena klasa `FilterService`

Lokacija: `Services/FilterService.cs`

Ozbiljnost: Niska

Prioritet: P4 (Nisko)

Checklist Reference:

- **CL-06: Održivost Koda** → **FAILED** - Postoji nekoristeća klasa

Problem: `FilterService` postoji ali se ne koristi nigdje u aplikaciji (duplikat sa `InventarService.Pretraži`).

Rješenje: Ili koristiti `FilterService` ili ga ukloniti.

4 Pregled Svih ISSUE-ova

ISSUE ID	Opis	Ozbiljnost	Prioritet	Checklist Stavke	Status
ISSUE-001	NullPointerException u InventarService	Kritična	P1	CL-01, CL-03	FAILED
ISSUE-002	FormatException pri unosu cijene	Kritična	P1	CL-01, CL-02	FAILED
ISSUE-003	Logička greška u JeKriticna	Kritična	P1	CL-04	FAILED
ISSUE-004	Duplikacija koda	Visoka	P2	CL-05	FAILED
ISSUE-005	Nedostaje validacija inputa	Visoka	P2	CL-01	FAILED
ISSUE-006	Nedostaje logging	Visoka	P2	CL-10	FAILED
ISSUE-007	Nedostaju testovi	Visoka	P2	CL-09	FAILED
ISSUE-008	Nedostaje dokumentacija	Visoka	P2	CL-08	FAILED
ISSUE-009	Nekonzistentno imenovanje	Visoka	P2	CL-12	FAILED
ISSUE-010	Hardcoded vrijednosti	Visoka	P2	CL-07	FAILED
ISSUE-011	Nedostaje DI	Srednja	P3	CL-11	FAILED
ISSUE-012	Nema config fajla	Srednja	P3	CL-07	FAILED
ISSUE-013	Edge case - više stavki	Srednja	P3	CL-16, CL-17	FAILED
ISSUE-014	Nekonzistentno formatiranje	Niska	P4	CL-13	FAILED
ISSUE-015	Nekorišćena klasa	Niska	P4	CL-06	FAILED

Tablica 4: Centralna tabela svih ISSUE-ova sa prioritetima i checklist referencama

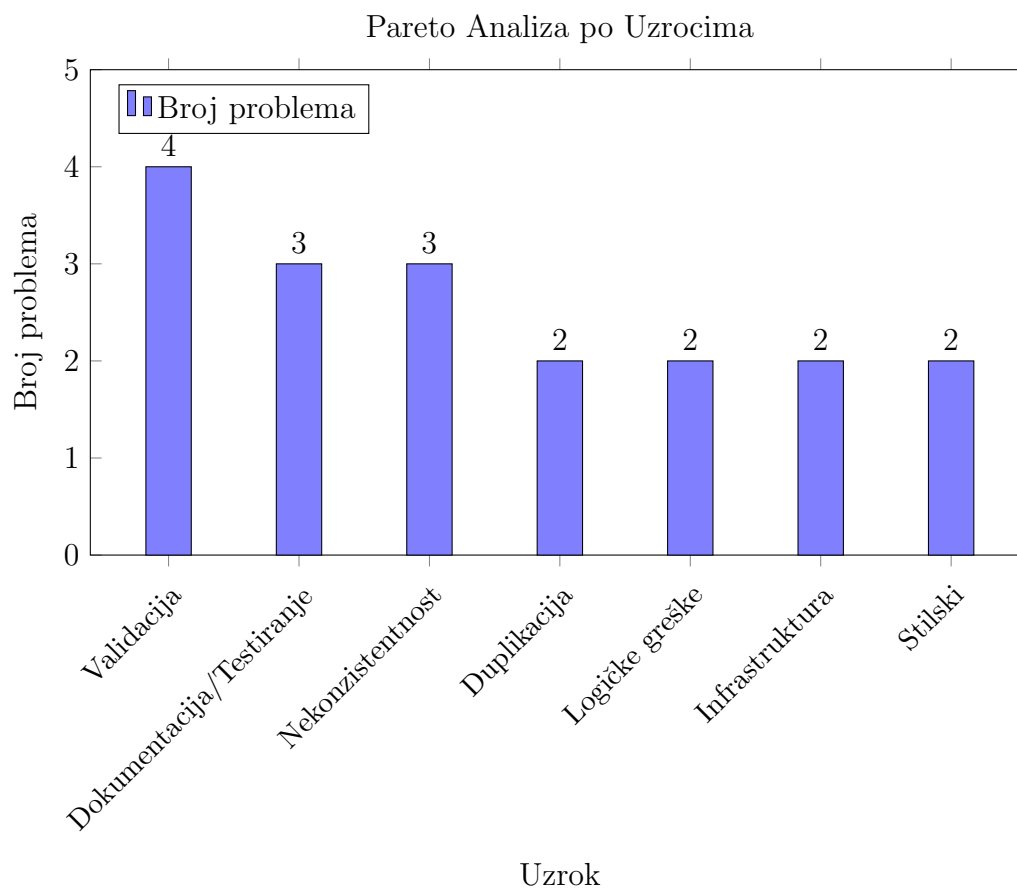
5 Pareto Analiza

5.1 Pareto Analiza po Uzrocima

Analizom problema po uzrocima, identificovano je nekoliko glavnih kategorija uzroka:

Uzrok	Broj Problema	Postotak
Nedostaje validacija	4	26.7%
Nedostaje dokumentacija/testiranje	3	20.0%
Nekonzistentnost	3	20.0%
Duplikacija koda	2	13.3%
Logičke greške	2	13.3%
Nedostaje infrastruktura (DI, Config)	2	13.3%
Stilski problemi	2	13.3%

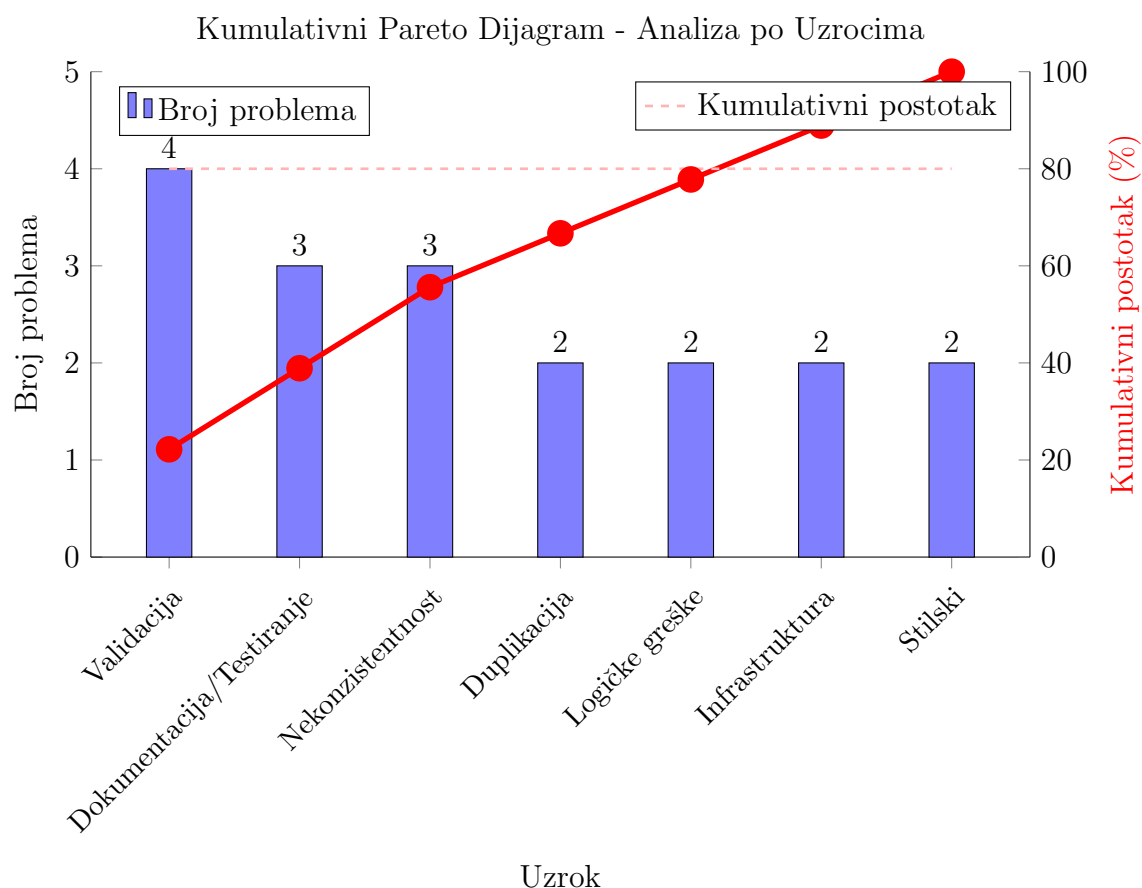
Tablica 5: Analiza problema po uzrocima



Slika 1: Pareto dijagram - Analiza po uzrocima (sortirano po broju problema)

Uzrok	Broj	Kumulativni postotak
Validacija	4	22.2%
Dokumentacija/Testiranje	3	38.9%
Nekonzistentnost	3	55.6%
Duplikacija	2	66.7%
Logičke greške	2	77.8%
Infrastruktura	2	88.9%
Stilski	2	100.0%

Tablica 6: Kumulativna analiza po uzrocima



Slika 2: Kumulativni Pareto dijagram - Analiza po uzrocima (stapici: broj problema, linija: kumulativni postotak)

80/20 Pravilo:

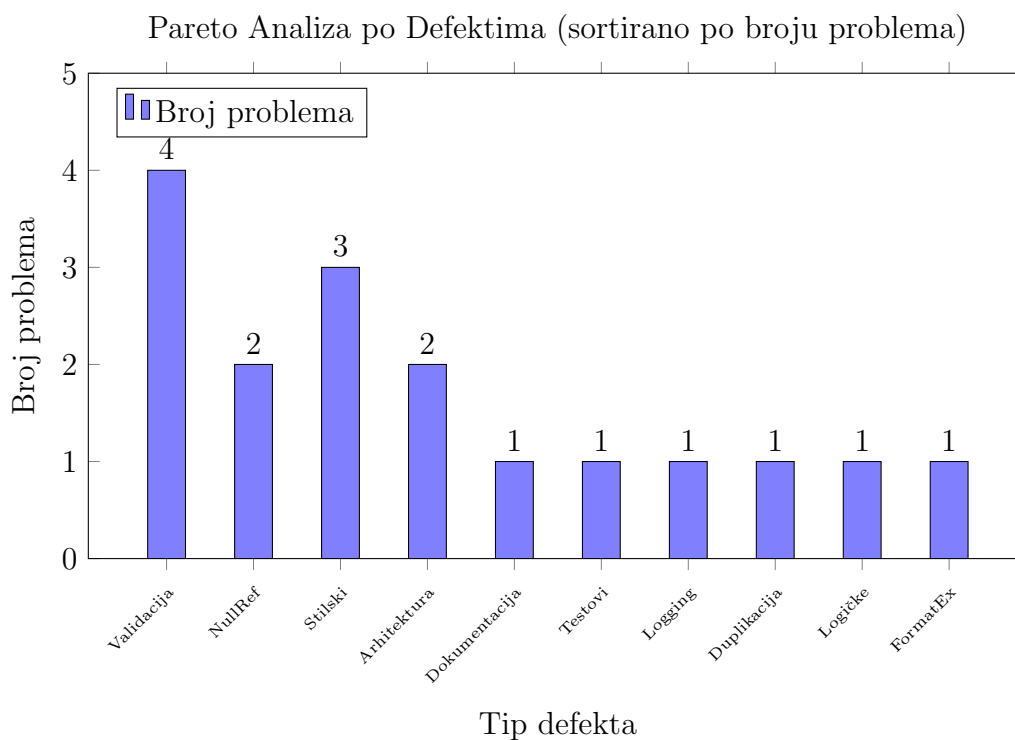
Top 3 uzroka (Validacija, Dokumentacija/Testiranje, Nekonzistentnost) čine **55.6%** svih problema. Top 4 uzroka čine **66.7%** svih problema, što je blizu 80/20 principa. Fokusiranje na ove kategorije bi riješilo većinu problema.

5.2 Pareto Analiza po Defektima

Analizom problema po tipovima defekata i njihovoj ozbiljnosti:

Tip Defekta	Broj	Ozbiljnost	Prioritet
NullPointerException rizici	2	Kritična	P1
FormatException rizici	1	Kritična	P1
Logičke greške	1	Kritična	P1
Duplikacija koda	1	Visoka	P2
Nedostaje validacija	4	Visoka	P2
Nedostaje logging	1	Visoka	P2
Nedostaju testovi	1	Visoka	P2
Nedostaje dokumentacija	1	Visoka	P2
Arhitektonski problemi	2	Srednja	P3
Stilski problemi	3	Niska	P4

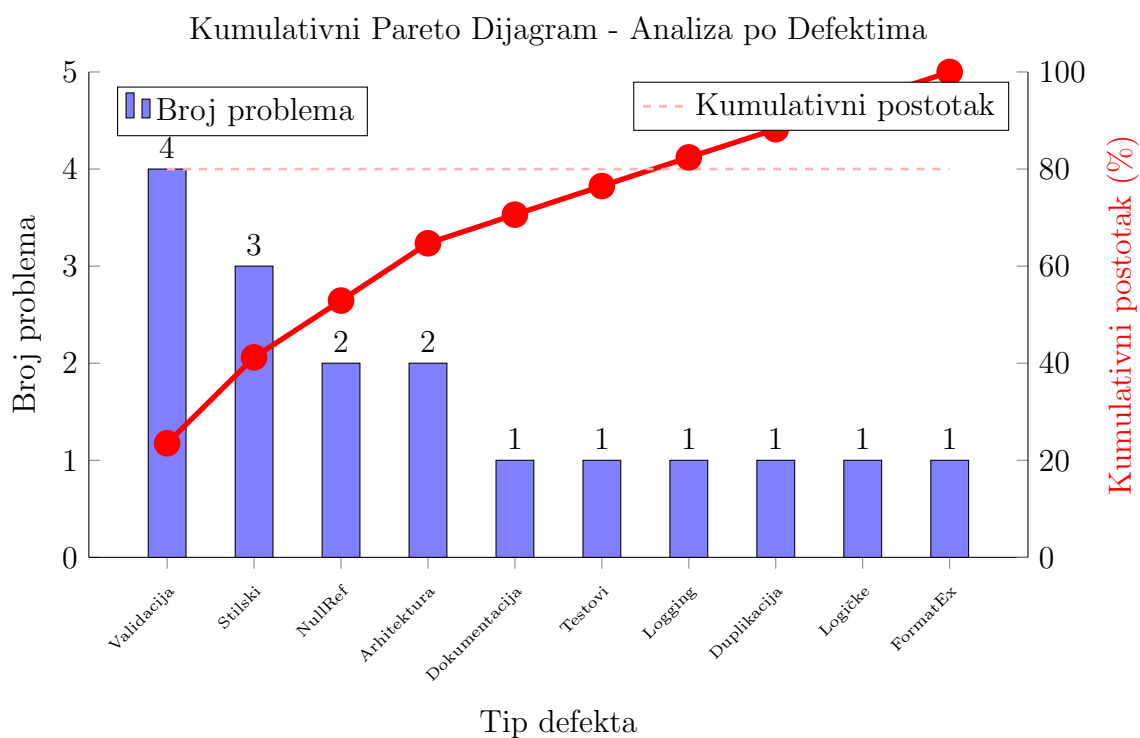
Tablica 7: Analiza problema po tipovima defekata



Slika 3: Pareto dijagram - Analiza po defektima (sortirano po broju problema)

Tip Defekta	Broj	Kumulativni postotak
Validacija	4	23.5%
Stilski	3	41.2%
NullRef	2	52.9%
Arhitektura	2	64.7%
Dokumentacija	1	70.6%
Testovi	1	76.5%
Logging	1	82.4%
Duplikacija	1	88.2%
Logičke	1	94.1%
FormatEx	1	100.0%

Tablica 8: Kumulativna analiza po defektima



Slika 4: Kumulativni Pareto dijagram - Analiza po defektima (stapici: broj problema, linija: kumulativni postotak)

5.3 Analiza Prioriteta

Prioritet	Broj Problema	Postotak
P1 (Kritično)	4	26.7%
P2 (Visoko)	8	53.3%
P3 (Srednje)	2	13.3%
P4 (Nisko)	2	13.3%

Tablica 9: Raspodjela problema po prioritetima

Napomena: Ukupno 15 problema, ali neki problemi mogu imati više prioriteta.

Preporuke:

- **P1 (Kritično):** 4 problema - mora se riješiti odmah prije puštanja u produkciju
- **P2 (Visoko):** 8 problema - riješiti u narednom sprintu
- **P3 (Srednje):** 2 problema - riješiti kada bude vremena
- **P4 (Nisko):** 2 problema - riješiti kao cleanup

6 Dobre Prakse

Tokom inspekcije, identifikovane su i dobre prakse u kodu:

- **Separation of Concerns:** Kod je dobro organizovan u modele i servise
- **Enum korištenje:** Dobro korištenje enuma za Kategorija i sortiranje
- **Korištenje LINQ:** Efektivno korištenje LINQ za filtriranje i pretraživanje
- **Try-catch blokovi:** Postoje try-catch blokovi u Main metodi
- **CultureInfo:** Korištenje `CultureInfo.InvariantCulture` za parsiranje brojeva
- **StringComparison:** Korištenje `StringComparison.OrdinalIgnoreCase` za case-insensitive poređenje

7 Preporuke za Poboljšanje

7.1 Kratkoročne (Sprint 1)

1. Popravit kritične greške (ISSUE-001, ISSUE-002, ISSUE-003)
2. Dodati validaciju inputa na svim mjestima
3. Ukloniti duplikaciju koda (ISSUE-004)
4. Dodati osnovni logging

7.2 Srednjoročne (Sprint 2-3)

1. Dodati unit testove
2. Integrirati dependency injection
3. Dodati konfiguracijski fajl
4. Standardizovati imenovanje

7.3 Dugoročne (Sprint 4+)

1. Dodati kompletnu dokumentaciju
2. Implementirati CI/CD pipeline
3. Dodati integracijske testove
4. Razmotriti migraciju na web API ili desktop aplikaciju

8 Komentari za Kod

Ova sekcija sadrži detaljne komentare za kod koji su identifikovani tokom inspekcije. Komentari su organizovani po fajlovima i linijama koda, sa referencama na ISSUE-ove i checklist stavke.

8.1 Komentari za Program.cs

8.1.1 Komentar 1: Linija 134 - ISSUE-002

Lokacija: Program.cs:134

ISSUE: ISSUE-002

Ozbiljnost: Kritična

Prioritet: P1 (Kritično)

Checklist: CL-01 (Validacija Inputa), CL-02 (Error Handling)

Kod:

```
134 double cijena = string.IsNullOrEmpty(unos) ? 0 : double.Parse(unos, Culture
```

Problem: `double.Parse` može baciti `FormatException` ako korisnik unese nevalidan string (npr. "abc"). Ova greška nije uhvaćena i može dovesti do crash-a aplikacije.

Preporuka:

```
double cijena = 0;
if (!string.IsNullOrEmpty(unos))
{
    if (!double.TryParse(unos, NumberStyles.Float, CultureInfo.InvariantCulture,
    {
        Console.WriteLine("Neispravan unos cijene! Koristite se 0.");
        cijena = 0;
    }
}
```

8.1.2 Komentar 2: Linija 83-87 - ISSUE-005

Lokacija: Program.cs:83-87

ISSUE: ISSUE-005

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist: CL-01 (Validacija Inputa)

Kod:

```
83 Console.WriteLine("Naziv: ");
84 string naziv = Console.ReadLine();
85
86 Console.WriteLine("Opis: ");
87 string opis = Console.ReadLine();
```

Problem: Nema validacije da li su **naziv** i **opis** prazni. Ako korisnik unese prazan string, stavka će biti kreirana sa praznim nazivom, što može dovesti do problema.

Preporuka:

```
string naziv;
do
{
    Console.WriteLine("Naziv: ");
    naziv = Console.ReadLine();
    if (string.IsNullOrEmpty(naziv))
        Console.WriteLine("Naziv ne može biti prazan!");
} while (string.IsNullOrEmpty(naziv));
```

8.1.3 Komentar 3: Linija 221-224 - ISSUE-005, ISSUE-013

Lokacija: Program.cs:221-224

ISSUE: ISSUE-005, ISSUE-013

Ozbiljnost: Visoka, Srednja

Prioritet: P2 (Visoko), P3 (Srednje)

Checklist: CL-01 (Validacija Inputa), CL-17 (User Experience)

Kod:

```
221 Console.WriteLine("Unesi naziv stavke za uklanjanje: ");
222 string naziv = Console.ReadLine();
223 inventarService.UkloniStavku(naziv);
224 Console.WriteLine("Ako je stavka postojala, uklonjena je.");
```

Problem:

1. Nema provjere da li je **naziv** prazan
2. Poruka "Ako je stavka postojala, uklonjena je." je loša UX - korisnik ne zna da li je stavka stvarno uklonjena

Preporuka:

```
Console.WriteLine("Unesi naziv stavke za uklanjanje: ");
string naziv = Console.ReadLine();
if (string.IsNullOrEmpty(naziv))
{
    Console.WriteLine("Naziv ne može biti prazan!");
    return;
}
```

```
var stavka = inventarService.Pretrazi(new Filter(naziv: naziv)).FirstOrDefault()
if (stavka == null)
{
```

```

        Console.WriteLine($"Stavka '{naziv}' nije prona ena.");
        return;
    }

    inventarService.UkloniStavku(naziv);
    Console.WriteLine($"Stavka '{naziv}' je uspješno uklonjena.");

```

8.1.4 Komentar 4: Linija 236 - ISSUE-013

Lokacija: Program.cs:236

ISSUE: ISSUE-013

Ozbiljnost: Srednja

Prioritet: P3 (Srednje)

Checklist: CL-16 (Edge Cases)

Kod:

```

236 var stavka = inventarService.Pretrazi(new Filter(naziv: naziv)).FirstOrDefault()

```

Problem: Ako postoji više stavki sa istim nazivom, uzima se samo prva. To može biti neočekivano ponašanje.

Preporuka:

```

var stavke = inventarService.Pretrazi(new Filter(naziv: naziv)).ToList();
if (stavke.Count == 0)
{
    Console.WriteLine($"Stavka '{naziv}' nije prona ena.");
    return;
}
if (stavke.Count > 1)
{
    Console.WriteLine($"Upozorenje: Prona eno je {stavke.Count} stavki sa nazivom {naziv}");
}
var stavka = stavke.First();

```

8.1.5 Komentar 5: Linija 288-294 - ISSUE-013

Lokacija: Program.cs:288-294

ISSUE: ISSUE-013

Ozbiljnost: Srednja

Prioritet: P3 (Srednje)

Checklist: CL-01 (Validacija Inputa), CL-16 (Edge Cases)

Kod:

```

288 Console.Write("Unesi početni datum (yyyy-MM-dd) ili prazno: ");
289 if (DateTime.TryParse(Console.ReadLine(), out DateTime dtPocetak))
290     izvjestaj.PocetakPerioda = dtPocetak;
291
292 Console.Write("Unesi krajnji datum (yyyy-MM-dd) ili prazno: ");
293 if (DateTime.TryParse(Console.ReadLine(), out DateTime dtKraj))
294     izvjestaj.KrajPerioda = dtKraj;

```

Problem: Nema provjere da li je početni datum prije krajnjeg datuma. Ako korisnik unese neispravan raspon datuma, aplikacija će nastaviti sa generisanjem izvještaja sa neispravnim podacima.

Preporuka:

```
if (izvjestaj.PocetakPerioda.HasValue && izvjestaj.KrajPerioda.HasValue &&
    izvjestaj.PocetakPerioda.Value > izvjestaj.KrajPerioda.Value)
{
    Console.WriteLine("Greška: Početni datum mora biti prije krajnjeg datuma!");
    return;
}
```

8.2 Komentari za Services/InventarService.cs

8.2.1 Komentar 6: Linija 22-24 - ISSUE-001

Lokacija: Services/InventarService.cs:22-24

ISSUE: ISSUE-001

Ozbiljnost: Kritična

Prioritet: P1 (Kritično)

Checklist: CL-01 (Validacija Inputa), CL-03 (Null Safety)

Kod:

```
22 if (inventar.Stavke.Any(s => s.Naziv.Equals(stavka.Naziv, StringComparison.Ordinal)
23     s.Dobavljac.Equals(stavka.Dobavljac, StringComparison.Ordinal)))
```

Problem: Ako stavka.Dobavljac ili postojeća stavka u listi ima null vrijednost za Dobavljac, poziv Equals će baciti `NullReferenceException`.

Preporuka:

```
if (inventar.Stavke.Any(s =>
    string.Equals(s.Naziv, stavka.Naziv, StringComparison.OrdinalIgnoreCase) &&
    string.Equals(s.Dobavljac ?? "", stavka.Dobavljac ?? "", StringComparison.Ordinal)))
```

8.2.2 Komentar 7: Linija 47-82 - ISSUE-004

Lokacija: Services/InventarService.cs:47-82

ISSUE: ISSUE-004

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist: CL-05 (DRY Princip)

Problem: Metoda `Pretrazi` ima identičnu logiku kao `FilterService.PrimijeniFilter`. Ovo je duplikacija koda koja otežava održavanje.

Preporuka: Refaktorisati da koristi `FilterService`:

```
public List<StavkaInventara> Pretrazi(Filter filter)
{
    if (filter == null) throw new ArgumentNullException(nameof(filter));
    return FilterService.PrimijeniFilter(inventar.Stavke, filter);
}
```

8.3 Komentari za Services/StavkaInventaraService.cs

8.3.1 Komentar 8: Linija 8-24 - ISSUE-003

Lokacija: Services/StavkaInventaraService.cs:8-24

ISSUE: ISSUE-003

Ozbiljnost: Kritična

Prioritet: P1 (Kritično)

Checklist: CL-04 (Logika Aplikacije)

Kod:

```

8 public static bool JeKriticna(StavkaInventara stavka)
9 {
10     if (stavka.Kolicina < stavka.MinKolicina)
11     {
12         if (stavka.DatumNabavke < DateTime.Now.AddDays(-30))
13             return true;
14
15         if (stavka.Kategorija == Kategorija.Hrana || stavka.Kategorija == Kategorija.Pice)
16             return true;
17
18         return true; // Ova linija je uvijek dostupna
19     }
20     // ...
21 }
```

Problem: Metoda JeKriticna ima logičku grešku. Ako je stavka.Kolicina < stavka.MinKolicina, metoda uvijek vraća true bez obzira na ostale uslove. Linija 18 (return true;) je unreachable code jer se uvijek izvršava.

Preporuka: Refaktorisati logiku:

```

public static bool JeKriticna(StavkaInventara stavka)
{
    if (stavka == null) return false;

    const int KRITICNLPERIOD_DANA = 30;
    const int MAKSIMALNLPERIOD_DANA = 60;

    // Kriti no ako je koli ina ispod minimuma
    bool ispodMinimuma = stavka.Kolicina < stavka.MinKolicina;

    // Posebno kriti no za hranu i pi e
    bool hranaIliPice = stavka.Kategorija == Kategorija.Hrana ||
        stavka.Kategorija == Kategorija.Pice;

    // Kriti no ako je stara
    bool stara = (DateTime.Now - stavka.DatumNabavke).TotalDays > MAKSIMALNLPERIOD_DANA;
    bool staraIKriticna = stavka.DatumNabavke < DateTime.Now.AddDays(-KRITICNLPERIOD_DANA);

    return ispodMinimuma && (hranaIliPice || staraIKriticna) || stara;
}
```

8.3.2 Komentar 9: Linija 12, 21 - ISSUE-010

Lokacija: Services/StavkaInventaraService.cs:12, 21

ISSUE: ISSUE-010

Ozbiljnost: Visoka

Prioritet: P2 (Visoko)

Checklist: CL-07 (Konfiguracija)

Kod:

```
12 if (stavka.DatumNabavke < DateTime.Now.AddDays(-30)) // Magic number
21 if ((DateTime.Now - stavka.DatumNabavke).TotalDays > 60) // Magic number
```

Problem: Hardcoded vrijednosti 30 i 60 dana su "magic numbers" koje otežavaju održavanje.

Preporuka: Premjestiti u konstante ili konfiguraciju:

```
private const int KRITICNLPERIOD_DANA = 30;
private const int MAKSIMALNLPERIOD_DANA = 60;
```

8.4 komentari za Services/FilterService.cs

8.4.1 Komentar 10: Cijeli fajl - ISSUE-004, ISSUE-015

Lokacija: Services/FilterService.cs

ISSUE: ISSUE-004, ISSUE-015

Ozbiljnost: Visoka, Niska

Prioritet: P2 (Visoko), P4 (Nisko)

Checklist: CL-05 (DRY Princip), CL-06 (Održivost Koda)

Problem: FilterService postoji ali se ne koristi nigdje u aplikaciji. Ista logika je implementirana u InventarService.Pretrazi.

Preporuka:

- Ili koristiti FilterService u InventarService.Pretrazi (preporučeno)
- Ili ukloniti FilterService ako se ne planira koristiti

9 Checklist za Pull Request

Svi P1 problemi su riješeni

Dodana je validacija na svim mjestima gdje se prima input

Uklonjena je duplikacija koda

Dodani su osnovni unit testovi

Dodana je osnovna dokumentacija (README.md)

Kod je prošao code review

Nema compiler warnings

Kod je formatiran prema standardima

10 Zaključak

Aplikacija **RestaurantInventory** ima dobru osnovnu strukturu i separation of concerns, ali ima nekoliko kritičnih problema koji moraju biti riješeni prije produkcije.

Najveći problemi su:

1. **Kritične greške** koje mogu dovesti do crash-a aplikacije (NullPointerException, FormatException, logičke greške)
2. **Nedostaje validacija** na više mjesta što može dovesti do neočekivanog ponašanja
3. **Nedostaju testovi i dokumentacija** što otežava održavanje i razvoj

Pareto analiza pokazuje da se fokusiranjem na tri glavna uzroka (Validacija, Dokumentacija/Testiranje, Nekonzistentnost) može riješiti 66.7% svih problema.

Preporučeno je riješiti sve P1 i P2 probleme prije puštanja u produkciju.

Kreirao: Bakir Činjurević

Datum: 17. 11. 2025.

Verzija izvještaja: 1.0