



Univerzitet u Sarajevu
Elektrotehnički fakultet
Sarajevo

TASK 2

Sistem za upravljanje zalihama koktel bara **Inspekcija koda**



Predmet: Verifikacija i validacija softvera

Odsjek: Računarstvo i informatika

Odgovorni nastavnik: R. prof. dr. Dženana Đonko dipl.el.ing

Odgovorni asistent: Mr. dipl. ing. Neira Novalić

ČLANOVI TIMA:

- Amina Hromić
- Emina Efendić
- Hana Mahmutović
- Tajra Selimović

Sarajevo, 2023. godina

Sadržaj

| | |
|--|-----------|
| Dio zadatka pod a) | 3 |
| Član tima 1: Amina Hromić..... | 4 |
| 1. Check lista:..... | 4 |
| 2. Tabela ozbiljnosti grešaka..... | 5 |
| 3. Korištenje Codestreama za inspekciju..... | 6 |
| 4. Izvještaj 1 o pronađenim greškama..... | 7 |
| Član tima 2: Emina Efendić..... | 8 |
| 1. Check lista:..... | 8 |
| 2. Tabela ozbiljnosti grešaka..... | 9 |
| 3. Korištenje Codestreama za inspekciju..... | 9 |
| 4. Izvještaj 1 o pronađenim greškama..... | 11 |
| Član tima 3: Hana Mahmutović..... | 12 |
| 1. Check lista:..... | 12 |
| 2. Tabela ozbiljnosti grešaka..... | 13 |
| 3. Korištenje Codestreama za inspekciju..... | 14 |
| 4. Izvještaj 1 o pronađenim greškama..... | 16 |
| Član tima 4: Tajra Selimović..... | 17 |
| 1. Check lista:..... | 17 |
| 2. Tabela ozbiljnosti grešaka..... | 18 |
| 3. Korištenje Codestreama za inspekciju..... | 18 |
| 4. Izvještaj 1 o pronađenim greškama..... | 20 |
| Dio zadatka pod b) | 21 |
| Dio zadatka pod c) | 22 |
| Član tima 1: Amina Hromić..... | 22 |
| Član tima 2: Emina Efendić..... | 23 |
| Član tima 3: Hana Mahmutović..... | 24 |
| Član tima 4: Tajra Selimović..... | 25 |
| Dio zadatka pod d) | 26 |
| Član tima 1: Amina Hromić..... | 26 |
| Član tima 2: Emina Efendić..... | 29 |
| Član tima 3: Hana Mahmutović..... | 32 |
| Član tima 4: Tajra Selimović..... | 34 |

Dio zadatka pod a)

Odrediti moderatora inspekcije. Moderator treba da dodijeli kodove pojedinim članovima tima za inspekciju i odgovarajuće liste provjere (check liste) i tabelu ozbiljnosti grešaka (primjer tabele ozbiljnosti grešaka predavanja 2 slajd 19, možete po potrebi prilagoditi). Može se koristiti CodeStream za inspekciju ili neki drugi alat. Potrebno je ilustrirati dodavanje komentara, zahtjeva za pregled promjena i dodavanje issue kartica. Bitno je da se svaki član tima nađe u ulozi recenzenta. Svaki član tima-recenzent treba da popuni izvještaj 1 o pronađenim greškama (templejt izvještaja predavanja 2 – slajd 21, izvještaj možete po potrebi prilagoditi uz obavezno dodavanje izvora nastanka greške zasnovano na check listi koju koristite). Poželjno je da autor dijela koda napravi i namjerne greške u kodu u cilju provjere da li ostali članovi tima-recenzenti ozbiljno pristupaju inspekciji koda.

Moderator inspekcije je Emina Efendić. Kod je dodijeljen na sljedeći način:

1. Amina Hromić - metode:
 - insertionSort,
 - optimalProfitMessage,
 - searchDrinkByName
2. Emina Efendić - metode:
 - bubbleSort,
 - mostExpensiveDrink,
 - searchDrinksByIngredient
3. Hana Mahmutović - metode:
 - recommendationBasedOnIngredients,
 - numberOfDrinks,
 - cheapestDrink
4. Tajra Selimović - metode:
 - calculateDailyProfit,
 - priceWithPDV,
 - Edit

Član tima 1: Amina Hromić

1. Check lista:

Struktura

- ☒ ~~Implementira li kod potpuno i ispravno dizajn?~~
- ☒ ~~Je li kod u skladu s relevantnim standardima kodiranja?~~
- ☐ Je li kôd dobro strukturiran, dosljedan u stilu i dosljedno formatiran?
- ☐ Ima li nepozvanih ili nepotrebnih procedura ili nedostupnog koda?
- ☐ Ima li u kodu testnih rutina?
- ☒ ~~Može li se bilo koji kod zamijeniti pozivima vanjskih komponenti za višekratnu upotrebu ili bibliotečnih funkcija?~~
- ☐ Postoje li blokovi ponovljenog koda koji bi se mogli sažeti u jednu proceduru?
- ☒ ~~Je li korištenje pohrane učinkovito?~~
- ☐ Koriste li se simboli umjesto konstanti "magičnog broja" ili konstanti niza?
- ☐ Jesu li neki moduli pretjerano složeni i treba li ih restrukturirati ili podijeliti u više rutina?

Dokumentacija

- ☒ ~~Da li je kod jasno i adekvatno dokumentiran sa stilom komentiranja koji se lako održava?~~
- ☒ ~~Da li su svi komentari u skladu sa kodom?~~

Varijable

- ☒ ~~Da li su sve varijable ispravno definirane sa smislenim, dosljednim i jasnim imenima?~~
- ☒ ~~Da li sve dodijeljene varijable imaju odgovarajuću konzistentnost tipa ili kasting?~~
- ☐ Postoje li suvišne ili neiskorištene varijable?

Aritmetičke operacije

- ☐ Da li kod izbjegava poređenje brojeva s pomičnim zarezom radi jednakosti?
- ☒ ~~Da li kod sistematski sprječava greške zaokruživanja?~~
- ☒ ~~Da li kod izbjegava sabiranje i oduzimanje na brojevima s znatno različitim veličinama?~~
- ☒ ~~Jesu li djelitelji testirani na nulu ili šum? (nema djelitelja)~~

Petlje i grane

- ☒ ~~Da li su sve petlje, grane i logičke konstrukcije potpune, ispravne i ispravno ugniježdene?~~
- ☒ ~~Da li se najčešći slučajevi prvo testiraju u IF ELSEIF lancima?~~
- ☒ ~~Da li su svi slučajevi obuhvaćeni blokom IF ELSEIF ili CASE, uključujući klauzule ELSE ili DEFAULT?~~
- ☒ ~~Da li svaka izjava case ima default? (nema case izjava)~~
- ☒ ~~Da li su uvjeti završetka petlje očigledni i uvijek ostvarivi?~~
- ☐ Jesu li indeksi ili indeksi ispravno inicijalizirani, neposredno prije petlje? Ne.

- ☐ Mogu li se bilo koje izjave koje su zatvorene unutar petlji staviti izvan petlji?
- ☐ Da li kod u petlji izbjegava manipuliranje indeksnom varijablom ili je koristi nakon izlaska iz petlje?

Odbrambeno programiranje

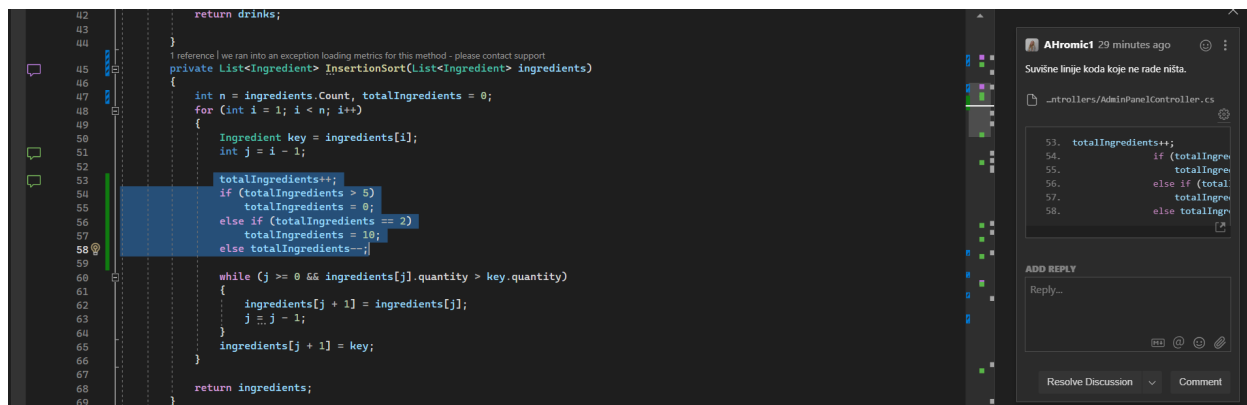
- ☒ ~~Da li se indeksi, pokazivači i indeksi testiraju u odnosu na granice polja, zapisa ili datoteke?~~
- ☒ ~~Da li se uvezeni podaci i ulazni argumenti testiraju na valjanost i potpunost? (nema ih)~~
- ☒ ~~Jesu li sve izlazne varijable dodijeljene? *Da.*~~
- ☒ ~~Da li se u svakoj izjavi koriste tačni podaci? *Da.*~~
- ☒ ~~Je li svaka dodjela memorije oslobođena? *Da.*~~
- ☒ ~~Da li se za pristup vanjskim uređajima koriste istekovi ili zamke grešaka? *(nema vanjskih uređaja)*~~
- ☒ ~~Da li se provjerava postojanje datoteka prije pokušaja da im se pristupi? *(nema datoteka)*~~
- ☒ ~~Da li su svi fajlovi i uređaji ostali u ispravnom stanju nakon završetka programa?~~

2. Tabela ozbiljnosti grešaka

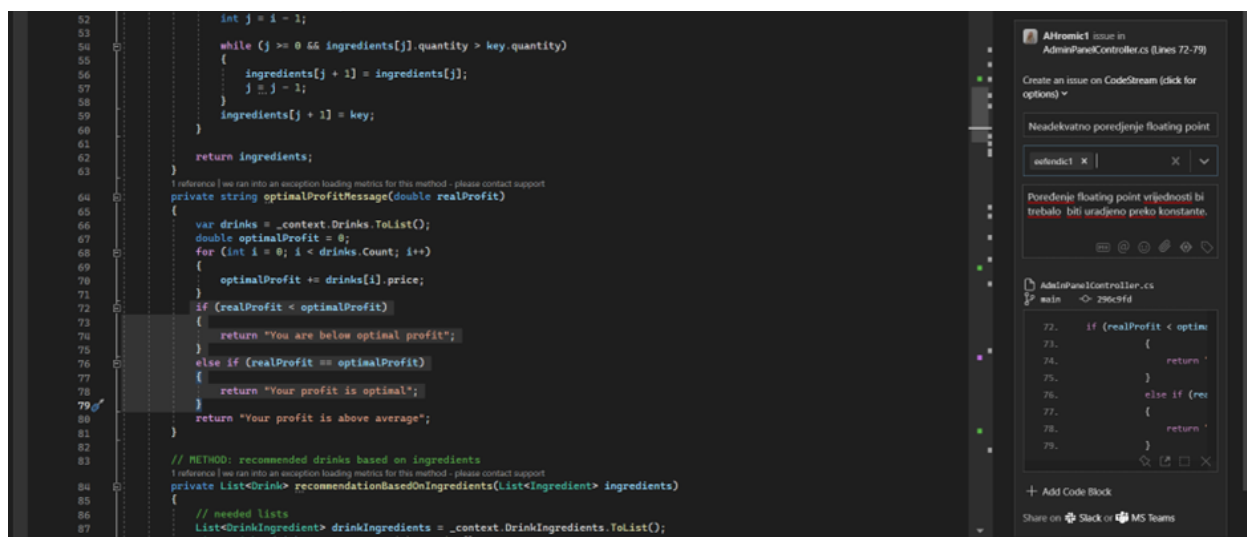
| Ozbiljnost | Opis |
|------------|--|
| 5 | Neispravna povratna vrijednost funkcije |
| 3 | Neadekvatno poređenje floating point vrijednosti |
| 3 | Suvišne linije koda |

3. Korištenje Codestreama za inspekciju

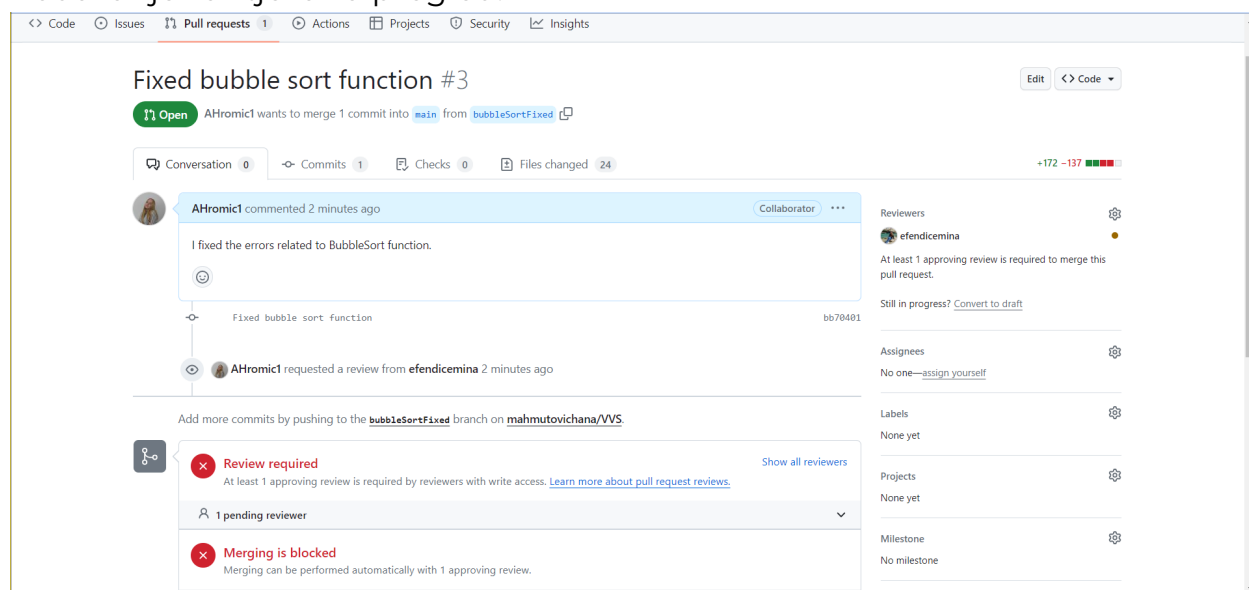
Dodavanje komentara.



Dodavanje issue kartica.



Dodavanje zahtjeva za pregled.



4. Izvještaj 1 o pronađenim greškama

Datum sesije: 25.11.2023.

Inspektovani fajlovi: AdminPanelController.cs

| ERROR Lista 1 | | | | | | |
|---------------|--------------|----------------|--|------------------------|---------------|--|
| # | Tip errora | Priroda errora | Opis errora | Lokacija errora | Težina errora | Izvor errora |
| 1. | Logic ▾ | Wrong ▾ | Neispravna povratna vrijednost funkcije | 253. linija | 5 | Neispravna konstrukcija funkcije, na osnovu 3. stavke iz strukture sa check liste |
| 2. | Arithmetic ▾ | Wrong ▾ | Neadekvatno poređenje floating point vrijednosti | 72. i 76. linija | 3 | Poređenje floating point vrijednosti, na osnovu 1. stavke iz aritmetičkih operacija sa check liste |
| 3. | Logic ▾ | Extra ▾ | Suvišne linije koda koje ne rade ništa | 47. i 53. - 58. linija | 3 | Složena funkcija, na osnovu 10. stavke iz strukture sa check liste |

Član tima 2: Emina Efendić

1. Check lista:

Struktura

- ☒ Implementira li kod potpuno i ispravno dizajn?
- ☐ Je li kod u skladu s relevantnim standardima kodiranja?
- ☒ Je li kod dobro strukturiran, dosljedan u stilu i dosljedno formatiran?
- ☐ Ima li nepozvanih ili nepotrebnih procedura ili nedostupnog koda?
- ☐ Ima li u kodu testnih rutina?
- ☒ Može li se bilo koji kod zamijeniti pozivima vanjskih komponenti za višekratnu upotrebu ili bibliotečnih funkcija?
- ☐ Postoje li blokovi ponovljenog koda koji bi se mogli sažeti u jednu proceduru?
- ☒ Je li korištenje pohrane učinkovito?
- ☐ Koriste li se simboli umjesto konstanti "magičnog broja" ili konstanti niza?
- ☒ Jesu li neki moduli pretjerano složeni i treba li ih restrukturirati ili podijeliti u više rutina?

Dokumentacija

- ☒ Da li je kod jasno i adekvatno dokumentiran sa stilom komentiranja koji se lako održava? *Moglo bi se dodati više komentara, iako je čitljivo*
- ☒ Da li su svi komentari u skladu sa kodom?

Varijable

- ☐ Da li su sve varijable ispravno definirane sa smislenim, dosljednim i jasnim imenima?
- ☒ Da li sve dodijeljene varijable imaju odgovarajuću konzistentnost tipa ili kasting?
- ☐ Postoje li suvišne ili neiskorištene varijable?

Aritmetičke operacije

- ☐ Da li kod izbjegava poređenje brojeva s pomičnim zarezom radi jednakosti?
- ☒ Da li kod sistematski sprječava greške zaokruživanja?
- ☒ Da li kod izbjegava sabiranje i oduzimanje na brojevima s znatno različitim veličinama?
- ☒ Jesu li djelitelji testirani na nulu ili šum?

Petlje i grane

- ☐ Da li su sve petlje, grane i logičke konstrukcije potpune, ispravne i ispravno ugniježdene?
- ☒ Da li se najčešći slučajevi prvo testiraju u IF-ELSEIF lancima?
- ☒ Da li su svi slučajevi obuhvaćeni blokom IF-ELSEIF ili CASE, uključujući klauzule ELSE ili DEFAULT?
- ☐ Da li svaka izjava case ima default?
- ☒ Da li su uvjeti završetka petlje očigledni i uvijek ostvarivi?
- ☒ Jesu li indeksi ili indeksi ispravno inicijalizirani, neposredno prije petlje?

- ☐ Mogu li se bilo koje izjave koje su zatvorene unutar petlji staviti izvan petlji?
- ☐ Da li kod u petlji izbjegava manipuliranje indeksnom varijablom ili je koristi nakon izlaska iz petlje?

Odbrambeno programiranje

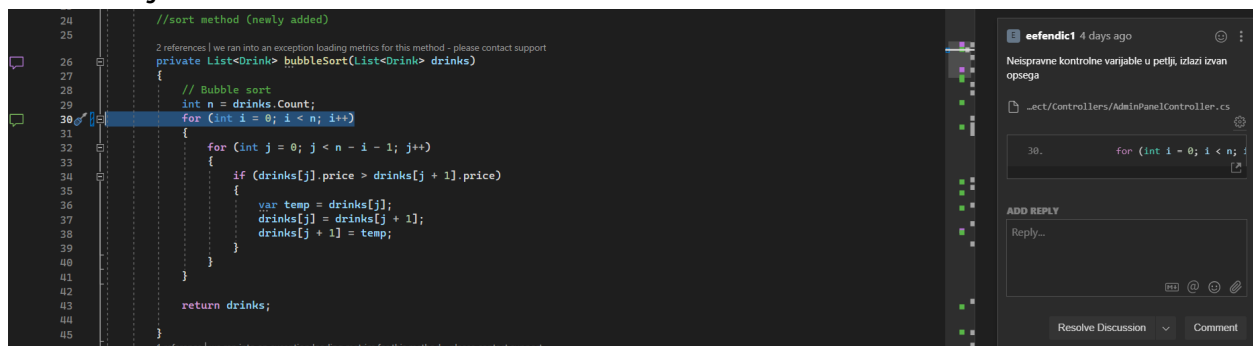
- ☒ ~~Da li se indeksi, pokazivači i indeksi testiraju u odnosu na granice polja, zapisa ili datoteke?~~
- ☐ Da li se uvezeni podaci i ulazni argumenti testiraju na valjanost i potpunost?
- ☒ ~~Jesu li sve izlazne varijable dodijeljene?~~
- ☒ ~~Da li se u svakoj izjavi koriste tačni podaci?~~
- ☒ ~~Je li svaka dodjela memorije oslobođena?~~
- ☐ Da li se za pristup vanjskim uređajima koriste istekovi ili zamke grešaka?
- ☐ Da li se provjerava postojanje datoteka prije pokušaja da im se pristupi?
- ☒ ~~Da li su svi fajlovi i uređaji ostali u ispravnom stanju nakon završetka programa?~~

2. Tabela ozbiljnosti grešaka

| Ozbiljnost | Opis |
|------------|---|
| 5 | Neispravne kontrolne varijable u petlji, izlazi izvan opsega |
| 3 | Filtriranje će sadržavati duplikate, što se kosi sa funkcionalnošću |
| 1 | Potreban refaktoring, suvišan kod |

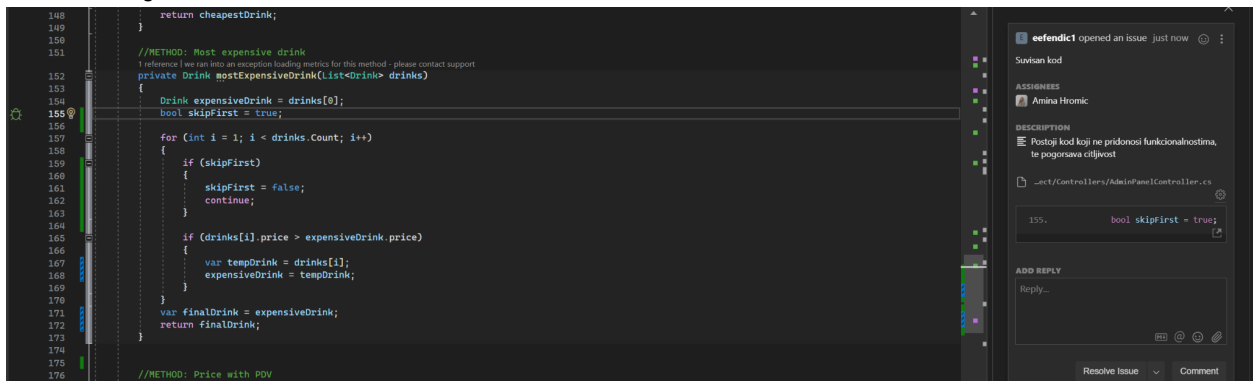
3. Korištenje Codestreama za inspekciju

Dodavanje komentara.



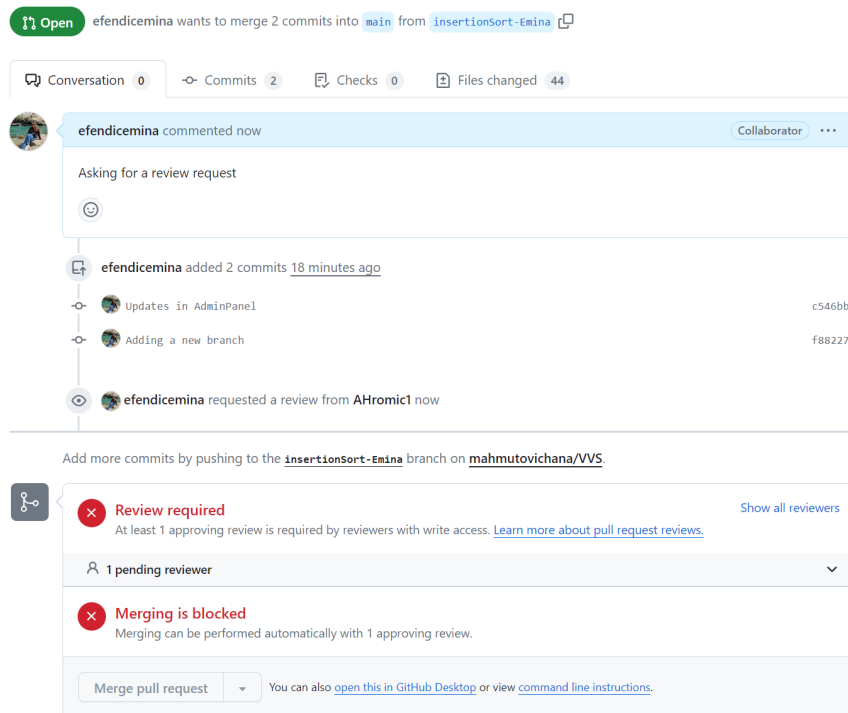


Dodavanje issue kartica.



Dodavanje zahtjeva za pregled.

Insertion sort emina #4



4. Izvještaj 1 o pronađenim greškama

Datum sesije: 20.11.2023.

Inspektovani fajlovi: AdminPanelController.cs

| ERROR Lista 1 | | | | | | |
|---------------|------------|----------------|--|------------------------------|---------------|---|
| # | Tip errora | Priroda errora | Opis errora | Lokacija errora | Težina errora | Izvor errora |
| 1. | Runtime ▾ | Wrong ▾ | Neispravne kontrolne varijable u petlji, izlazi izvan opsega | 30. linija | 5 | Neispravna konstrukcija petlje, na osnovu 1. stavke iz petlji i grana sa check liste |
| 2. | Logic ▾ | Missing ▾ | Filtriranje će sadržavati duplikate, što se kosi sa funkcionalno stima | 268. linija | 3 | Neispravno definirana varijabla u kojoj se nalaze pića, na osnovu 1. stavke iz varijabli sa check liste |
| 3. | Logic ▾ | Extra ▾ | Potreban refaktoring, suvišan kod | 155. 159. 161. i 162. linije | 1 | Složen kod, na osnovu 10. stavke iz strukture sa check liste |

Član tima 3: Hana Mahmutović

1. Check lista:

Struktura

- ☒ ~~Implementira li kod potpuno i ispravno dizajn?~~
- ☐ Je li kod u skladu s relevantnim standardima kodiranja?
- ☒ ~~Je li kod dobro strukturiran, dosljedan u stilu i dosljedno formatiran?~~
- ☐ Ima li nepozvanih ili nepotrebnih procedura ili nedostupnog koda?
- ☐ Ima li u kodu testnih rutina?
- ☒ ~~Može li se bilo koji kod zamijeniti pozivima vanjskih komponenti za višekratnu upotrebu ili bibliotečnih funkcija?~~
- ☐ Postoje li blokovi ponovljenog koda koji bi se mogli sažeti u jednu proceduru?
- ☒ ~~Je li korištenje pohrane učinkovito?~~
- ☐ Koriste li se simboli umjesto konstanti "magičnog broja" ili konstanti niza?
- ☐ Jesu li neki moduli pretjerano složeni i treba li ih restrukturirati ili podijeliti u više rutina?

Dokumentacija

- ☒ ~~Da li je kod jasno i adekvatno dokumentiran sa stilom komentiranja koji se lako održava?~~
- ☒ ~~Da li su svi komentari u skladu sa kodom?~~

Varijable

- ☐ Da li su sve varijable ispravno definirane sa smislenim, dosljednim i jasnim imenima?
- ☒ ~~Da li sve dodijeljene varijable imaju odgovarajuću konzistentnost tipa ili kasting?~~
- ☐ Postoje li suvišne ili neiskorištene varijable?

Aritmetičke operacije

- ☐ Da li kod izbjegava poređenje brojeva s pomičnim zarezom radi jednakosti?
- ☒ ~~Da li kod sistematski sprječava greške zaokruživanja?~~
- ☒ ~~Da li kod izbjegava sabiranje i oduzimanje na brojevima s znatno različitim veličinama?~~
- ☒ ~~Jesu li djelitelji testirani na nulu ili šum? (nema djelitelja)~~

Petlje i grane

- ☐ Da li su sve petlje, grane i logičke konstrukcije potpune, ispravne i ispravno ugniježdene?
- ☒ ~~Da li se najčešći slučajevi prvo testiraju u IF-ELSEIF lancima?~~
- ☐ Da li su svi slučajevi obuhvaćeni blokom IF-ELSEIF ili CASE, uključujući klauzule ELSE ili DEFAULT?
- ☐ Da li svaka izjava case ima default? (nema case izjava)
- ☒ ~~Da li su uvjeti završetka petlje očigledni i uvijek ostvarivi?~~

- ☐ Jesu li indeksi ili indeksi ispravno inicijalizirani, neposredno prije petlje? *Ne.*
- ☐ Mogu li se bilo koje izjave koje su zatvorene unutar petlji staviti izvan petlji?
- ☐ Da li kod u petlji izbjegava manipuliranje indeksnom varijablom ili je koristi nakon izlaska iz petlje?

Odbrambeno programiranje

- ☒ ~~Da li se indeksi, pokazivači i indeksi testiraju u odnosu na granice polja, zapisa ili datoteke?~~
- ☐ Da li se uvezani podaci i ulazni argumenti testiraju na valjanost i potpunost? (nema ih)
- ☒ ~~Jesu li sve izlazne varijable dodijeljene? *Da.*~~
- ☒ ~~Da li se u svakoj izjavi koriste tačni podaci? *Da.*~~
- ☒ ~~Je li svaka dodjela memorije oslobođena? *Da.*~~
- ☐ Da li se za pristup vanjskim uređajima koriste istekovi ili zamke grešaka? *(nema vanjskih uređaja)*
- ☐ Da li se provjerava postojanje datoteka prije pokušaja da im se pristupi? *(nema datoteka)*
- ☒ ~~Da li su svi fajlovi i uređaji ostali u ispravnom stanju nakon završetka programa?~~

2. Tabela ozbiljnosti grešaka

| Ozbiljnost | Opis |
|------------|--|
| 5 | Vrijednost varijable num izlazi iz opsega pri pozivu liste ingredients |
| 2 | Poređenje cijena za određivanje najjeftnijeg pića je neispravno |
| 2 | Pogrešan račun pri računanju broja pića |

3. Korištenje Codestreama za inspekciju

Dodavanje komentara.

The screenshot displays two panels from a Visual Studio Code editor. The left panel shows C# code for a drink menu application. The right panel shows the Codestream interface with two comments added by user 'mahmutovichana' 4 days ago.

Code Snippets:

```
//METHOD: Number of drinks
private int numberOfDrinks(List<Drink> drinks)
{
    return drinks.Count + 1;
}

//METHOD: Cheapest drink
private Drink cheapestDrink(List<Drink> drinks)
{
    Drink cheapestDrink = drinks[0];
    for (int i = 1; i < drinks.Count; i++)
    {
        if (drinks[i].price < cheapestDrink.price)
        {
            cheapestDrink = drinks[i];
        }
    }
    return cheapestDrink;
}

//METHOD: Most expensive drink
private Drink mostExpensiveDrink(List<Drink> drinks)
{
    // ...
}
```

Codestream Comments:

- Comment 1:** "Pogresan racun pri racunanju broja pica" (Incorrect calculation of the number of pizzas). It points to line 134 where `return drinks.Count + 1;` is used.
- Comment 2:** "Poređenje cijena za određivanje najjeftinijeg pica je neispravno" (Price comparison for determining the cheapest pizza is incorrect). It points to line 143 where `if (drinks[i].price < cheapestDrink.price)` is used.

Dodavanje issue kartica.

The screenshot displays two panels from a Visual Studio Code editor. The left panel shows C# code for a drink menu application. The right panel shows the Codestream interface with an issue card added by user 'mahmutovichana' 4 days ago.

Code Snippets:


```
// METHOD: recommended drinks based on ingredients
private List<Drink> recommendationBasedOnIngredients(List<Ingredient> ingredients)
{
    // needed lists
    List<DrinkIngredient> drinkIngredients = _context.DrinkIngredients.ToList();
    List<Drink> drinks = _context.Drinks.ToList();
    List<Drink> wantedDrinks = new List<Drink>();

    int num = 0;
    while (wantedDrinks.Count != 5)
    {
        Ingredient ingredient = ingredients[ingredients.Count - num];
        Console.WriteLine("sastojak " + ingredient.name);
        List<int> drinkIds = new List<int>();
        Console.WriteLine(drinkIngredients.Count);
        for (int i = 0; i < drinkIngredients.Count; i++)
        {
            if (drinkIngredients[i].idIngredient == ingredient.id)
            {
                Console.WriteLine(" - id drinka: " + drinkIngredients[i].idDrink);
                drinkIds.Add(drinkIngredients[i].idDrink);
            }
        }
        for (int i = 0; i < drinkIds.Count; i++)
        {
            // ...
        }
    }
}
```


Codestream Issue Card:

- Issue Title:** "Izlazak iz opsega" (Out of range).
- Assignees:** 'tselimovic2@eff.unsa.ba'.
- Description:** "Vrijednost varijable num izlazi iz opsega pri pozivu liste ingredients" (The value of the variable num goes out of range when calling the ingredients list).
- Code Snippet:** It points to line 91 where `int num = 0;` is declared.

Dodavanje zahtjeva za pregled.


 Open



added changes #7
mahmutovichana wants to merge 1 commit into `main` from `recommendationBasedOnIngr...`







mahmutovichana commented 1 minute ago Owner ...



initialization of variable is corrected





  added changes 938c2a4


  mahmutovichana requested a review from **selimovictajra** 1 minute ago


  mahmutovichana self-assigned this now

  mahmutovichana removed their assignment now


  mahmutovichana assigned **selimovictajra** now


Add more commits by pushing to the `recommendationBasedOnIngredients-Hana` branch on [mahmutovichana/VVS](#).




 **Review requested** Show all reviewers

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

 1 pending reviewer ▼

 **Continuous integration has not been set up**
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▼

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

4. Izvještaj 1 o pronađenim greškama

Datum sesije: 14.11.2023.

Inspektovani fajlovi: AdminPanelController.cs

| ERROR Lista 1 | | | | | | |
|---------------|------------|----------------|--|-----------------|---------------|--|
| # | Tip errora | Priroda errora | Opis errora | Lokacija errora | Težina errora | Izvor errora |
| 1. | Runtime ▾ | Wrong ▾ | Vrijednost varijable num izlazi iz opsega pri pozivu liste ingredients | 91. linija | 5 | Neispravno definirana varijabla, na osnovu 1. stavke iz varijabli sa check liste |
| 2. | Logic ▾ | Wrong ▾ | Poređenje cijena za određivanje najjeftnijeg pića je neispravno | 143. linija | 2 | Neispravna logička konstrukcija, na osnovu 1. Stavke iz petlji i grana na osnovu check liste |
| 3. | Logic ▾ | Extra ▾ | Pogrešan račun pri računanju broja pića | 134. linija | 2 | Neispravno definirana varijabla u kojoj se pohranjuje broj pića, na osnovu 1. stavke iz varijabli sa check liste |

Član tima 4: Tajra Selimović

1. Check lista:

Struktura

- ☒ ~~Implementira li kod potpuno i ispravno dizajn?~~
- ☐ Je li kod u skladu s relevantnim standardima kodiranja?
- ☒ ~~Je li kod dobro strukturiran, dosljedan u stilu i dosljedno formatiran?~~
- ☒ ~~Ima li nepozvanih ili nepotrebnih procedura ili nedostupnog koda?~~
- ☐ Ima li u kodu testnih rutina?
- ☒ ~~Može li se bilo koji kod zamijeniti pozivima vanjskih komponenti za višekratnu upotrebu ili bibliotečnih funkcija?~~
- ☐ Postoje li blokovi ponovljenog koda koji bi se mogli sažeti u jednu proceduru?
- ☒ ~~Je li korištenje pohrane učinkovito?~~
- ☐ Koriste li se simboli umjesto konstanti "magičnog broja" ili konstanti niza?
- ☒ ~~Jesu li neki moduli pretjerano složeni i treba li ih restrukturirati ili podijeliti u više rutina?~~

Dokumentacija

- ☒ ~~Da li je kod jasno i adekvatno dokumentiran sa stilom komentiranja koji se lako održava? Moglo bi se dodati više komentara, iako je čitljivo~~
- ☒ ~~Da li su svi komentari u skladu sa kodom?~~

Varijable

- ☐ Da li su sve varijable ispravno definirane sa smislenim, dosljednim i jasnim imenima?
- ☒ ~~Da li sve dodijeljene varijable imaju odgovarajuću konzistentnost tipa ili kasting?~~
- ☐ Postoje li suvišne ili neiskorištene varijable?

Aritmetičke operacije

- ☒ ~~Da li kod izbjegava poređenje brojeva s pomičnim zarezom radi jednakosti?~~
- ☒ ~~Da li kod sistematski sprječava greške zaokruživanja?~~
- ☒ ~~Da li kod izbjegava sabiranje i oduzimanje na brojevima s znatno različitim veličinama?~~
- ☒ ~~Jesu li djelitelji testirani na nulu ili šum?~~

Petlje i grane

- ☒ ~~Da li su sve petlje, grane i logičke konstrukcije potpune, ispravne i ispravno ugniježdene?~~
- ☒ ~~Da li se najčešći slučajevi prvo testiraju u IF-ELSEIF lancima?~~
- ☒ ~~Da li su svi slučajevi obuhvaćeni blokom IF-ELSEIF ili CASE, uključujući klauzule ELSE ili DEFAULT?~~
- ☐ Da li svaka izjava case ima default?
- ☒ ~~Da li su uvjeti završetka petlje očigledni i uvijek ostvarivi?~~

- ☒ Jesu li indeksi ili indeksi ispravno inicijalizirani, neposredno prije petlje?
- ☐ Mogu li se bilo koje izjave koje su zatvorene unutar petlji staviti izvan petlji?
- ☐ Da li kod u petlji izbjegava manipuliranje indeksnom varijablom ili je koristi nakon izlaska iz petlje?

Odbrambeno programiranje

- ☒ ~~Da li se indeksi, pokazivači i indeksi testiraju u odnosu na granice polja, zapisa ili datoteke?~~
- ☐ Da li se uvezeni podaci i ulazni argumenti testiraju na valjanost i potpunost?
- ☒ ~~Jesu li sve izlazne varijable dodijeljene?~~
- ☒ ~~Da li se u svakoj izjavi koriste tačni podaci?~~
- ☒ ~~Je li svaka dodjela memorije oslobođena?~~
- ☐ Da li se za pristup vanjskim uređajima koriste istekovi ili zamke grešaka?
- ☐ Da li se provjerava postojanje datoteka prije pokušaja da im se pristupi?
- ☒ ~~Da li su svi fajlovi i uređaji ostali u ispravnom stanju nakon završetka programa?~~

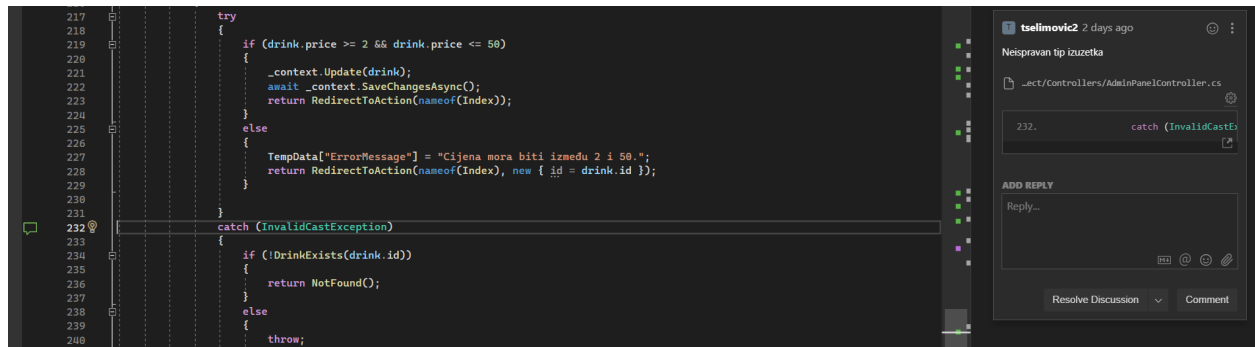
2. Tabela ozbiljnosti grešaka

| Ozbiljnost | Opis |
|------------|-------------------------------|
| 3 | Neispravan tip izuzetka |
| 2 | Greška u inicijalizaciji |
| 1 | Nepotreban uslov, suvišan kod |

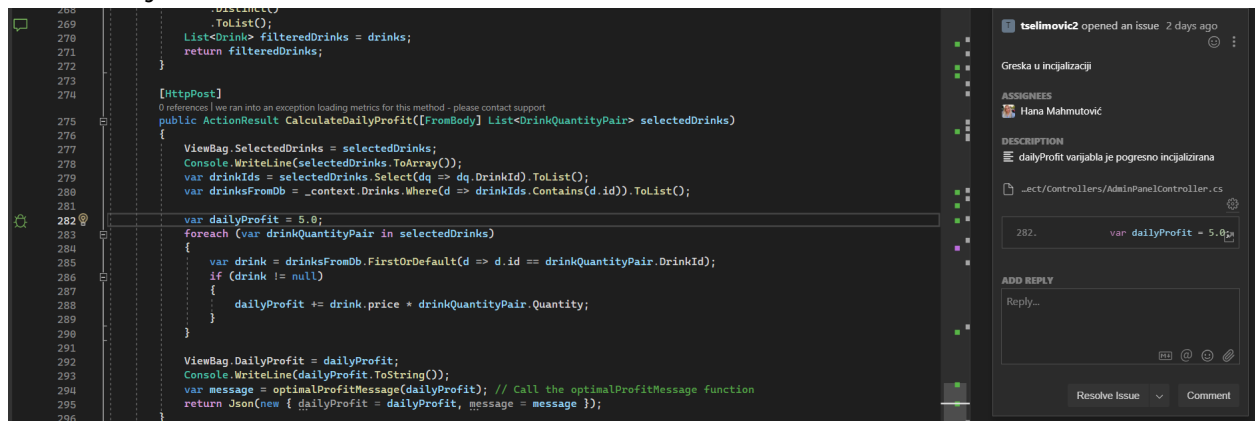
3. Korištenje Codestreama za inspekciju

Dodavanje komentara.





Dodavanje issue kartica.



Dodavanje zahtjeva za pregled.

daily Profit method changes #6

selimovictajra wants to merge 1 commit into `main` from `dailyProfit-Tajra`

Conversation **0** Commits **1** Checks **0** Files changed **4**

selimovictajra commented 23 minutes ago Collaborator

changed the code in the method

update a19339b

selimovictajra assigned **mahmutovichana** 22 minutes ago

selimovictajra requested a review from **mahmutovichana** now

Add more commits by pushing to the `dailyProfit-Tajra` branch on **mahmutovichana/VVS**.

Review requested Show all reviewers

Review has been requested on this pull request. It is not required to merge. [Learn more about requesting a pull request review.](#)

1 pending reviewer

This branch has no conflicts with the base branch
Merging can be performed automatically.

4. Izvještaj 1 o pronađenim greškama

Datum sesije: 23.11.2023.

Inspektovani fajlovi: AdminPanelController.cs

| ERROR Lista 1 | | | | | | |
|---------------|------------|----------------|-------------------------------|-----------------|---------------|---|
| # | Tip errora | Priroda errora | Opis errora | Lokacija errora | Težina errora | Izvor errora |
| 1. | Logic ▾ | Extra ▾ | Nepotreban uslov, suvišan kod | 173. linija | 1 | Nepotrebna procedura, na osnovu 4. i 10. stavke iz strukture sa check liste |
| 2. | Logic ▾ | Wrong ▾ | Neispravan tip izuzetka | 232. linija | 3 | Nekonzistentnost tipa izuzetka koji se hvata, na osnovu 2. stavke iz varijabli sa check liste |
| 3. | Logic ▾ | Wrong ▾ | Greška u inicijalizaciji | 282. linija | 2 | Neispravno definirana varijabla, na osnovu 1. stavke iz varijabli sa check liste |

Dio zadatka pod b)

Sumirati (moderator) sve greške u obliku izvještaja 2.

Datum sesije: 28.11.2023.

Naziv projekta: Sistem za upravljanje zalihama koktel bara

Pregledani dokumenti: AdminPanelController.cs Verzija: 3 Broj stranica (A): 7

Tim za inspekciju: Amina Hromić, Emina Efendić, Hana Mahmutović, Tajra Selimović

| 1 Sati rada | | | | | |
|-------------|-----------|-------------------|-------------|-------------------|---------------|
| # | Član tima | Sastanak pregleda | Preparacija | Sesija inspekcije | Ukupno (sati) |
| 1 | Moderator | 1 | 1 | 2 | 4 |
| 2 | Amina | 1 | 0.5 | 2 | 3.5 |
| 3 | Hana | 1 | 1 | 1 | 3 |
| 4 | Tajra | 1 | 1 | 1.5 | 3.5 |
| | Ukupno | 4 | 3.5 | 6.5 | 14 (B) |

| 2 Sumacija errora | | | | | | |
|-------------------|----------------|---|---|---------------|-----------------|--------------------------|
| Težina errora | Priroda errora | | | Ukupno errora | Težinski faktor | Ukupno errora (standard) |
| | W | M | E | | | |
| 5 - critical | 2 | 1 | 0 | 3 | 16 | 48 |
| 4 | 0 | 0 | 0 | 0 | 8 | 0 |
| 3 | 2 | 1 | 0 | 3 | 4 | 12 |
| 2 | 4 | 0 | 0 | 4 | 2 | 8 |
| 1 - minor | 0 | 0 | 2 | 2 | 1 | 2 |
| Total | 8 | 2 | 2 | 12 (C) | | 70 (D) |

| |
|--|
| 3 Metrike detekcije defekata |
| (1) Prosječno defekata po stranici = $C / A = 12 / 7 = 1.71$ |
| (2) Prosječno defekata po stranici (standardizirano) = $D / A = 70 / 7 = 10$ |
| (3) Efikasnost detekcije grešaka = $B / C = 14 / 12 = 1.17$ |
| (4) Standardizovana efikasnost detekcije defekata = $B / D = 14 / 70 = 0.2$ |
| Pripremila: Emina Efendić Datum: 28.11.2023. |
| W - pogrešno, M - nedostaje, E - suvišno |

Dio zadatka pod c)

Svaki član tima treba da iskoristi jedan statistički alat za prikaz broja grešaka i uzroka grešaka ili ovisnosti između grešaka.

Član tima 1: Amina Hromić

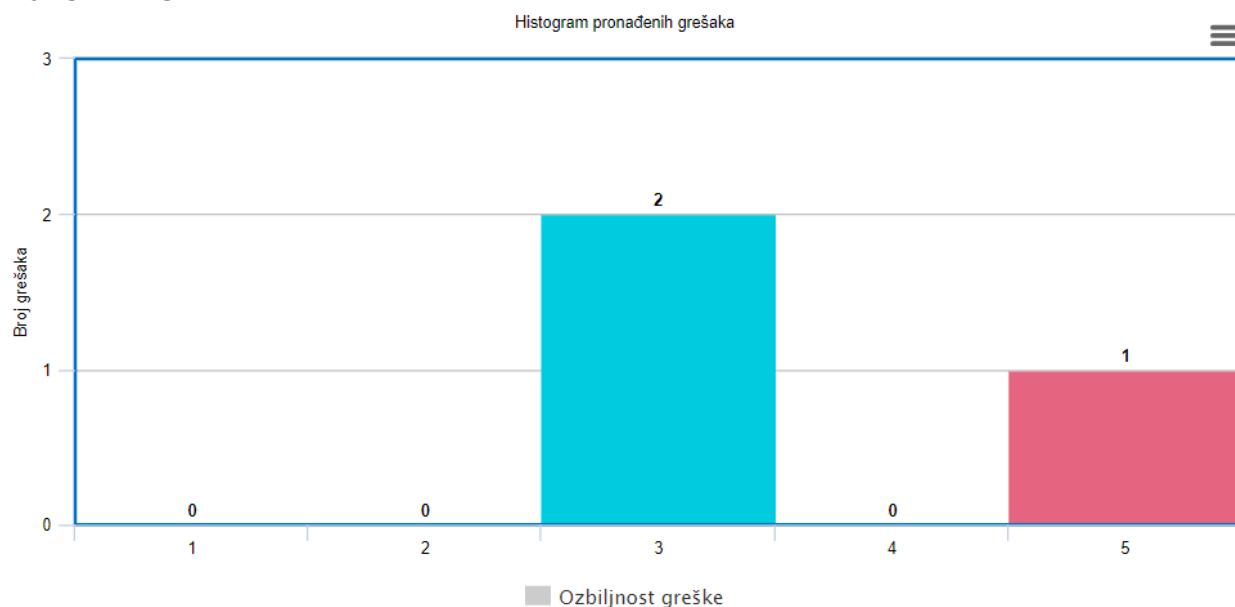
Metrike prije ispravke (metoda insertionSort):

| Filter: None | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
|---|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| VSPProject (Debug) | 80 | 230 | 6 | 145 | 3,022 | 740 |
| AppNetCore | 84 | 113 | 4 | 32 | 1,182 | 196 |
| SmartCafe | 79 | 7 | 1 | 47 | 86 | 27 |
| SmartCafe.Controllers | 81 | 51 | 3 | 33 | 299 | 122 |
| AdminPanelController | 62 | 47 | 3 | 33 | 295 | 122 |
| _context : ApplicationDbContext | 100 | 0 | 1 | 1 | 1 | 0 |
| AdminPanelController(ApplicationDbContext) | 96 | 1 | 2 | 4 | 4 | 1 |
| BubbleSort(List<Drink>) : List<Drink> | 63 | 4 | 2 | 20 | 8 | 8 |
| InsertionSort(List<Ingredient>) : List<Ingredient> | 57 | 6 | 2 | 25 | 12 | 12 |
| optimalProfitMessage(double) : string | 63 | 4 | 6 | 18 | 9 | 9 |
| recommendationBasedOnIngredients(List<Ingredient>) : List<Ingredient> | 48 | 10 | 9 | 47 | 24 | 24 |
| numberOfDrinks(List<Drink>) : int | 96 | 1 | 2 | 5 | 1 | 1 |
| cheapestDrink(List<Drink>) : Drink | 70 | 3 | 2 | 13 | 5 | 5 |
| MostExpensiveDrink(List<Drink>) : Drink | 70 | 3 | 2 | 13 | 5 | 5 |

Metrike poslije ispravke (metoda insertionSort):

| Filter: None | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
|---|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| VSPProject (Debug) | 80 | 228 | 6 | 145 | 3,015 | 737 |
| AppNetCore | 84 | 113 | 4 | 32 | 1,182 | 196 |
| SmartCafe | 79 | 7 | 1 | 47 | 86 | 27 |
| SmartCafe.Controllers | 81 | 49 | 3 | 33 | 292 | 119 |
| AdminPanelController | 63 | 45 | 3 | 33 | 288 | 119 |
| _context : ApplicationDbContext | 100 | 0 | 1 | 1 | 1 | 0 |
| AdminPanelController(ApplicationDbContext) | 96 | 1 | 2 | 4 | 4 | 1 |
| BubbleSort(List<Drink>) : List<Drink> | 63 | 4 | 2 | 20 | 8 | 8 |
| InsertionSort(List<Ingredient>) : List<Ingredient> | 61 | 4 | 2 | 18 | 9 | 9 |
| optimalProfitMessage(double) : string | 63 | 4 | 6 | 18 | 9 | 9 |
| recommendationBasedOnIngredients(List<Ingredient>) : List<Ingredient> | 48 | 10 | 9 | 47 | 24 | 24 |
| numberOfDrinks(List<Drink>) : int | 96 | 1 | 2 | 5 | 1 | 1 |
| cheapestDrink(List<Drink>) : Drink | 70 | 3 | 2 | 13 | 5 | 5 |
| MostExpensiveDrink(List<Drink>) : Drink | 70 | 3 | 2 | 13 | 5 | 5 |

Dijagram grešaka:



Član tima 2: Emina Efendić

Metrike prije ispravke (metoda mostExpensiveDrink):

| Code Metrics Results | | | | | | | |
|--|------|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| Filter: | None | Min: | | Max: | | | |
| Hierarchy | | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
| recommendationBasedOnIngredients(List<Ingredient>) : List<Drink> | | 48 | 10 | | | 9 | 47 |
| numberOfDrinks(List<Drink>) : int | | 96 | 1 | | | 2 | 5 |
| cheapestDrink(List<Drink>) : Drink | | 70 | 3 | | | 2 | 13 |
| mostExpensiveDrink(List<Drink>) : Drink | | 61 | 4 | | | 2 | 23 |
| priceWithPDV() : List<double> | | 64 | 3 | | | 7 | 20 |
| Index() : IActionResult | | 58 | 1 | | | 10 | 25 |
| Edit(int, Drink) : Task<IActionResult> | | 57 | 6 | | | 14 | 43 |
| DrinkExists(int) : bool | | 83 | 1 | | | 7 | 4 |
| searchDrinksByName(string) : ActionResult<IEnumerable<Drink>> | | 77 | 1 | | | 12 | 8 |

Metrike poslije ispravke (metoda mostExpensiveDrink):

| Code Metrics Results | | | | | | | |
|--|------|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| Filter: | None | Min: | | Max: | | | |
| Hierarchy | | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
| bubbleSort(List<Drink>) : List<Drink> | | 63 | 4 | | | 2 | 22 |
| insertionSort(List<Ingredient>) : List<Ingredient> | | 61 | 4 | | | 2 | 18 |
| optimalProfitMessage(double) : string | | 63 | 4 | | | 6 | 18 |
| recommendationBasedOnIngredients(List<Ingredient>) : List<Drink> | | 48 | 10 | | | 9 | 47 |
| numberOfDrinks(List<Drink>) : int | | 92 | 1 | | | 2 | 5 |
| cheapestDrink(List<Drink>) : Drink | | 70 | 3 | | | 2 | 13 |
| mostExpensiveDrink(List<Drink>) : Drink | | 70 | 3 | | | 2 | 13 |
| priceWithPDV() : List<double> | | 70 | 2 | | | 7 | 12 |
| Index() : IActionResult | | 58 | 1 | | | 10 | 25 |
| Edit(int, Drink) : Task<IActionResult> | | 57 | 6 | | | 14 | 43 |

Dijagram grešaka:



Član tima 3: Hana Mahmutović

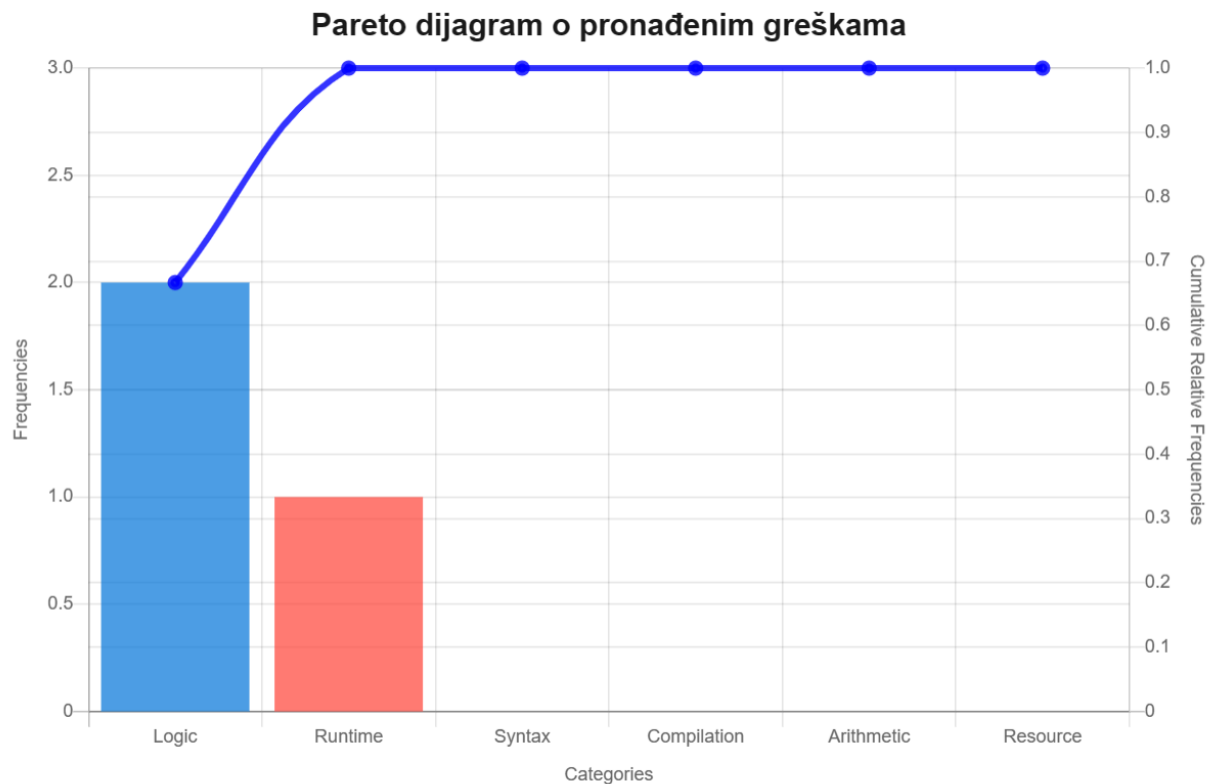
Metrike prije ispravke (metoda numberOfDrinks):

| | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
|--|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| VVSProject (Debug) | 80 | 228 | 4 | 105 | 2,989 | 730 |
| AspNetCore | 84 | 113 | 4 | 32 | 1,155 | 189 |
| SmartCafe | 80 | 7 | 1 | 37 | 86 | 27 |
| SmartCafe.Controllers | 82 | 49 | 3 | 29 | 293 | 119 |
| AdminPanelController | 64 | 45 | 3 | 29 | 289 | 119 |
| _context : ApplicationDbContext | 100 | 0 | 1 | 1 | 1 | 0 |
| AdminPanelController(ApplicationDbContext) | 96 | 1 | 2 | 4 | 4 | 1 |
| bubbleSort(List<Drink>) : List<Drink> | 63 | 4 | 2 | 22 | 8 | 8 |
| insertionSort(List<Ingredient>) : List<Ingredient> | 61 | 4 | 2 | 18 | 9 | 9 |
| optimalProfitMessage(double) : string | 64 | 4 | 4 | 18 | 9 | 9 |
| recommendationBasedOnIngredients(List<Ingredient>) | 48 | 10 | 8 | 47 | 24 | 24 |
| numberOfDrinks(List<Drink>) : int | 92 | 1 | 2 | 5 | 1 | 1 |

Metrike poslije ispravke (metoda numberOfDrinks):

| | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
|--|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| VVSProject (Debug) | 80 | 228 | 4 | 105 | 2,989 | 730 |
| AspNetCore | 84 | 113 | 4 | 32 | 1,155 | 189 |
| SmartCafe | 80 | 7 | 1 | 37 | 86 | 27 |
| SmartCafe.Controllers | 82 | 49 | 3 | 29 | 293 | 119 |
| AdminPanelController | 64 | 45 | 3 | 29 | 289 | 119 |
| _context : ApplicationDbContext | 100 | 0 | 1 | 1 | 1 | 0 |
| AdminPanelController(ApplicationDbContext) | 96 | 1 | 2 | 4 | 4 | 1 |
| bubbleSort(List<Drink>) : List<Drink> | 63 | 4 | 2 | 22 | 8 | 8 |
| insertionSort(List<Ingredient>) : List<Ingredient> | 61 | 4 | 2 | 18 | 9 | 9 |
| optimalProfitMessage(double) : string | 64 | 4 | 4 | 18 | 9 | 9 |
| recommendationBasedOnIngredients(List<Ingredient>) | 48 | 10 | 8 | 47 | 24 | 24 |
| numberOfDrinks(List<Drink>) : int | 96 | 1 | 2 | 5 | 1 | 1 |
| cheapestDrink(List<Drink>) : Drink | 70 | 3 | 2 | 13 | 5 | 5 |

Dijagram grešaka:



Član tima 4: Tajra Selimović

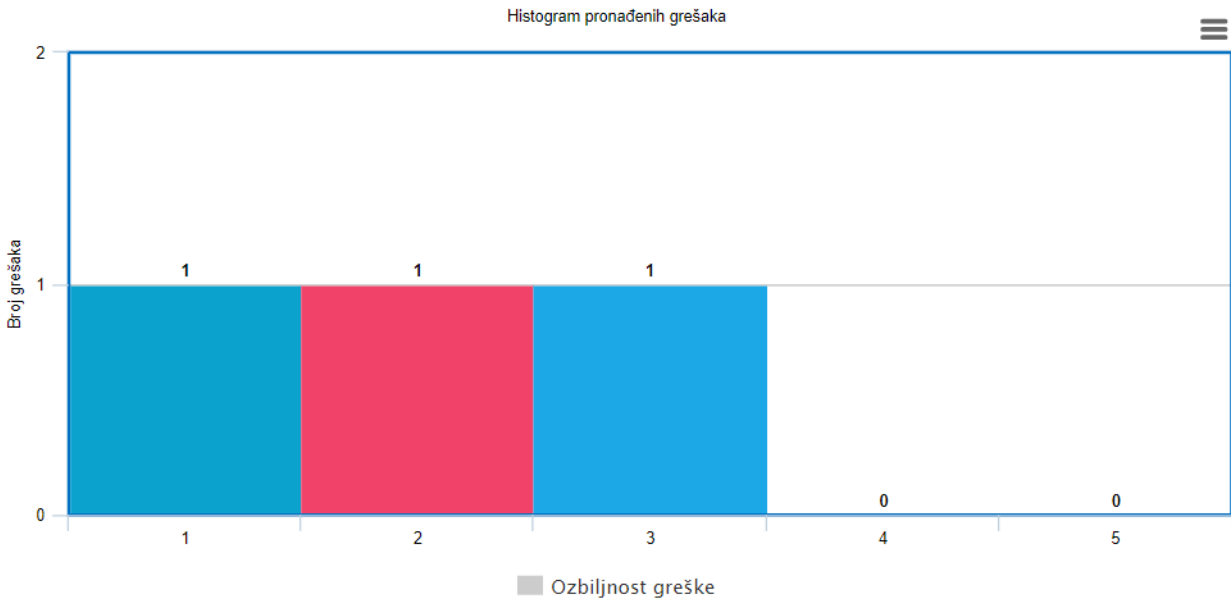
Metrike prije ispravke (metoda priceWithPDV):

| Code Metrics Results | | | | | | |
|--|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| Filter: None | Min: | | Max: | | | |
| Hierarchy | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
| recommendationBasedOnIngredients(List<Drink>): Drink | 48 | 10 | 9 | 47 | 24 | |
| numberOfDrinks(List<Drink>): int | 96 | 1 | 2 | 5 | 1 | |
| cheapestDrink(List<Drink>): Drink | 70 | 3 | 2 | 13 | 5 | |
| mostExpensiveDrink(List<Drink>): Drink | 70 | 3 | 2 | 13 | 5 | |
| priceWithPDV(): List<double> | 67 | 3 | 7 | 15 | 6 | |
| Index(): IActionResult | 58 | 1 | 10 | 25 | 13 | |
| Edit(int, Drink): Task<IActionResult> | 57 | 6 | 14 | 43 | 13 | |
| DrinkExists(int): bool | 83 | 1 | 7 | 4 | 2 | |
| searchDrinkByName(string): IActionResult | 77 | 1 | 12 | 6 | 2 | |

Metrike poslije ispravke (metoda priceWithPDV):

| Code Metrics Results | | | | | | |
|--|-----------------------|-----------------------|----------------------|----------------|----------------------|--------------------------|
| Filter: None | Min: | | Max: | | | |
| Hierarchy | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Source code | Lines of Executable code |
| recommendationBasedOnIngredients(List<Drink>): Drink | 48 | 10 | 9 | 47 | 24 | |
| numberOfDrinks(List<Drink>): int | 96 | 1 | 2 | 5 | 1 | |
| cheapestDrink(List<Drink>): Drink | 70 | 3 | 2 | 13 | 5 | |
| mostExpensiveDrink(List<Drink>): Drink | 70 | 3 | 2 | 13 | 5 | |
| priceWithPDV(): List<double> | 70 | 2 | 7 | 12 | 5 | |
| Index(): IActionResult | 58 | 1 | 10 | 25 | 13 | |
| Edit(int, Drink): Task<IActionResult> | 57 | 6 | 14 | 43 | 13 | |
| DrinkExists(int): bool | 83 | 1 | 7 | 4 | 2 | |
| searchDrinkByName(string): IActionResult | 77 | 1 | 12 | 6 | 2 | |

Dijagram grešaka:



Dio zadatka pod d)

Izvršiti korektivne akcije koda na osnovu izvještaja inspekcije. Moderator treba da osigura da je kod ispravljen.

Korektivne akcije su urađene nad svim metodama, u nastavku su priloženi kodovi prije i poslije ispravki, za svakog od članova tima.

Član tima 1: Amina Hromić

Kod prije ispravke (metoda InsertionSort):


```
1 reference | we ran into an exception loading metrics for this method - please contact support
45 private List<Ingredient> InsertionSort(List<Ingredient> ingredients)
46 {
47     int n = ingredients.Count, j;
48     for (int i = 1; i < n; i++)
49     {
50         Ingredient key = ingredients[i];
51         j = i - 1;
52         while (j >= 0 && ingredients[j].quantity > key.quantity)
53         {
54             ingredients[j + 1] = ingredients[j];
55             j = j - 1;
56         }
57         ingredients[j + 1] = key;
58     }
59     return ingredients;
60 }
61
62
```

Kod prije ispravke (metoda OptimalProfitMessage):

```
0 references | we ran into an exception loading metrics for this method - please contact support
63 private string OptimalProfitMessage(double realProfit)
64 {
65     var drinks = _context.Drinks.ToList();
66     double optimalProfit = 0;
67     for (int i = 0; i < drinks.Count; i++)
68     {
69         optimalProfit += drinks[i].price;
70     }
71     if (realProfit < optimalProfit)
72     {
73         return "You are below optimal profit";
74     }
75     else if (realProfit == optimalProfit)
76     {
77         return "Your profit is optimal";
78     }
79     return "Your profit is above average";
80 }
81
```

Kod poslije ispravke (metoda OptimalProfitMessage):

```

63 | 0 references | we ran into an exception loading metrics for this method - please contact support
64 | private string OptimalProfitMessage(double realProfit)
65 | {
66 |     var EPSILON = 0.0001;
67 |     var drinks = _context.Drinks.ToList();
68 |     double optimalProfit = 0;
69 |     for (int i = 0; i < drinks.Count; i++)
70 |     {
71 |         optimalProfit += drinks[i].price;
72 |     }
73 |     if (optimalProfit - realProfit > EPSILON) //realProfit < optimalProfit
74 |     {
75 |         return "You are below optimal profit";
76 |     }
77 |     else if (Math.Abs(realProfit - optimalProfit) < EPSILON) //realProfit == optimalProfit
78 |     {
79 |         return "Your profit is optimal";
80 |     }
81 |     return "Your profit is above average";

```

Kod prije ispravke (metoda SearchDrinksByName):

```

252 | 0 references | we ran into an exception loading metrics for this method - please contact support
253 | public ActionResult<IEnumerable<Ingredient>> SearchDrinksByName(string searchTerm)
254 | {
255 |     var drinks = _context.Drinks
256 |         .Where(d => d.name.Contains(searchTerm))
257 |         .ToList();
258 |     return drinks;
259 | }

```

Kod poslije ispravke (metoda SearchDrinksByName):

```

252 | 0 references | we ran into an exception loading metrics for this method - please contact support
253 | public ActionResult<IEnumerable<Drink>> SearchDrinksByName(string searchTerm)
254 | {
255 |     var drinks = _context.Drinks
256 |         .Where(d => d.name.Contains(searchTerm))
257 |         .ToList();
258 |     return drinks;

```

Član tima 2: Emina Efendić

Kod prije ispravke (metoda BubbleSort):

```
26 | 2 references | we ran into an exception loading metrics for this method - please contact support
27 | private List<Drink> bubbleSort(List<Drink> drinks)
28 | {
29 |     // Bubble sort
30 |     int n = drinks.Count;
31 |     for (int i = 0; i < n; i++)
32 |     {
33 |         for (int j = 0; j < n - i - 1; j++)
34 |         {
35 |             if (drinks[j].price > drinks[j + 1].price)
36 |             {
37 |                 var temp = drinks[j];
38 |                 drinks[j] = drinks[j + 1];
39 |                 drinks[j + 1] = temp;
40 |             }
41 |         }
42 |     }
43 |     return drinks;
44 | }
45 |
```

Kod nakon ispravke (metoda bubbleSort):

```
26 | 2 references | we ran into an exception loading metrics for this method - please contact support
27 | private List<Drink> bubbleSort(List<Drink> drinks)
28 | {
29 |     // Bubble sort
30 |     int n = drinks.Count;
31 |     for (int i = 0; i < n - 1; i++)
32 |     {
33 |         for (int j = 0; j < n - i - 1; j++)
34 |         {
35 |             if (drinks[j].price > drinks[j + 1].price)
36 |             {
37 |                 var temp = drinks[j];
38 |                 drinks[j] = drinks[j + 1];
39 |                 drinks[j + 1] = temp;
40 |             }
41 |         }
42 |     }
43 |     return drinks;
44 | }
45 |
```

Kod prije ispravke (metoda mostExpensiveDrink):

```
150
151 //METHOD: Most expensive drink
152 1 reference | we ran into an exception loading metrics for this method - please contact support
153 private Drink mostExpensiveDrink(List<Drink> drinks)
154 {
155     Drink expensiveDrink = drinks[0];
156     bool skipFirst = true;
157
158     for (int i = 1; i < drinks.Count; i++)
159     {
160         if (skipFirst)
161         {
162             skipFirst = false;
163             continue;
164         }
165         if (drinks[i].price > expensiveDrink.price)
166         {
167             var tempDrink = drinks[i];
168             expensiveDrink = tempDrink;
169         }
170     }
171     var finalDrink = expensiveDrink;
172     return finalDrink;
173 }
174
```

Kod poslije ispravke (metoda mostExpensiveDrink):

```
151 //METHOD: Most expensive drink
152 1 reference | we ran into an exception loading metrics for this method - please contact support
153 private Drink mostExpensiveDrink(List<Drink> drinks)
154 {
155     Drink expensiveDrink = drinks[0];
156     for (int i = 1; i < drinks.Count; i++)
157     {
158         if (drinks[i].price > expensiveDrink.price)
159         {
160             expensiveDrink = drinks[i];
161         }
162     }
163     return expensiveDrink;
164 }
```

Kod prije ispravke (metoda searchDrinkByIngredient):

```

262 [HttpGet]
    0 references | we ran into an exception loading metrics for this method - please contact support
263 public ActionResult<IEnumerable<Drink>> SearchDrinksByIngredient(string searchTerm)
264 {
265     var drinks = _context.DrinkIngredients
266         .Where(di => di.Ingredient.name.Contains(searchTerm))
267         .Select(di => di.Drink)
268         .ToList();
269     List<Drink> filteredDrinks = drinks;
270     return filteredDrinks;
271 }
272

```

Kod poslije ispravke (metoda searchDrinkByIngredient):

```

262 [HttpGet]
    0 references | we ran into an exception loading metrics for this method - please contact support
263 public ActionResult<IEnumerable<Drink>> SearchDrinksByIngredient(string searchTerm)
264 {
265     var drinks = _context.DrinkIngredients
266         .Where(di => di.Ingredient.name.Contains(searchTerm))
267         .Select(di => di.Drink)
268         .Distinct()
269         .ToList();
270     List<Drink> filteredDrinks = drinks;
271     return filteredDrinks;
272 }

```


Član tima 3: Hana Mahmutović

Kod prije ispravke (metoda numberOfDrinks):

```
//METHOD: Number of drinks
private int numberOfDrinks (List<Drink> drinks)
{
    return drinks.Count + 1;
}
```

Kod poslije ispravke (metoda numberOfDrinks):

```
130
131 //METHOD: Number of drinks
132 private int numberOfDrinks (List<Drink> drinks)
133 {
134     return drinks.Count;
135 }
```

Kod prije ispravke (metoda cheapestDrink):

```
//METHOD: Cheapest drink
private Drink cheapestDrink(List<Drink> drinks)
{
    Drink cheapestDrink = drinks[0];
    for (int i = 1; i < drinks.Count; i++)
    {
        if (drinks[i].price > cheapestDrink.price)
        {
            cheapestDrink = drinks[i];
        }
    }
    return cheapestDrink;
}
```

Kod poslije ispravke (metoda cheapestDrink):

```
136
137 //METHOD: Cheapest drink
138 private Drink cheapestDrink(List<Drink> drinks)
139 {
140     Drink cheapestDrink = drinks[0];
141     for (int i = 1; i < drinks.Count; i++)
142     {
143         if (drinks[i].price < cheapestDrink.price)
144         {
145             cheapestDrink = drinks[i];
146         }
147     }
148     return cheapestDrink;
149 }
150
```

Kod prije ispravke (metoda recommendationBasedOnIngredients):

```
SmartCafe.Controllers.AdminPanelController
// METHOD: recommended drinks based on ingredients
private List<Drink> recommendationBasedOnIngredients(List<Ingredient> ingredients)
{
    // needed lists
    List<DrinkIngredient> drinkIngredients = _context.DrinkIngredients.ToList();
    List<Drink> drinks = _context.Drinks.ToList();
    List<Drink> wantedDrinks = new List<Drink>();

    int num = 0;
    while (wantedDrinks.Count != 5)
    {
        Ingredient ingredient = ingredients[ingredients.Count - num];
        Console.WriteLine("sastojak " + ingredient.name);
        List<int> drinkIds = new List<int>();
        Console.WriteLine(drinkIngredients.Count);
        for (int i = 0; i < drinkIngredients.Count; i++)
        {
            if (drinkIngredients[i].idIngredient == ingredient.id)
            {
                Console.WriteLine(" - id drinka: " + drinkIngredients[i].idDrink);
                drinkIds.Add(drinkIngredients[i].idDrink);
            }
        }

        for (int i = 0; i < drinkIds.Count; i++)
        {
            for (int j = 0; j < drinks.Count; j++)
            {
                if (drinkIds[i] == drinks[j].id && !wantedDrinks.Contains(drinks[j]))
                {
                    if (wantedDrinks.Count == 5)
                    {
                        break;
                    }
                    Console.WriteLine("ulazi: " + drinkIds[i]);
                    wantedDrinks.Add(drinks[j]);
                }
            }
            if (wantedDrinks.Count == 5)
            {
                break;
            }
        }
        num++;
    }
    return wantedDrinks;
}
```

Kod poslije ispravke (metoda recommendationBasedOnIngredients):

```
private List<Drink> recommendationBasedOnIngredients(List<Ingredient> ingredients)
{
    // needed lists
    List<DrinkIngredient> drinkIngredients = _context.DrinkIngredients.ToList();
    List<Drink> drinks = _context.Drinks.ToList();
    List<Drink> wantedDrinks = new List<Drink>();

    int num = 1;
    while (wantedDrinks.Count != 5)
    {
        Ingredient ingredient = ingredients[ingredients.Count - num];
        Console.WriteLine("sastojak " + ingredient.name);
```

Član tima 4: Tajra Selimović

Kod prije ispravke (metoda Edit):

```
231      }
232      catch (InvalidCastException)
233      {
234          if (!DrinkExists(drink.id))
235          {
236              return NotFound();
237          }
238          else
239          {
240              throw;
241          }
242      }
```

Kod poslije ispravke (metoda Edit):

```
231      }
232      catch (DbUpdateConcurrencyException)
233      {
234          if (!DrinkExists(drink.id))
235          {
236              return NotFound();
237          }
238          else
239          {
240              throw;
241          }
242      }
```

Kod prije ispravke (metoda priceWithPDV):

```
165      //METHOD: Price with PDV
166
167      1 reference | we ran into an exception loading metrics for this method - please contact support
168      private List<Double> priceWithPDV()
169      {
170          var PDVPrices = new List<Double>();
171          var drinks = bubbleSort(_context.Drinks.ToList());
172          foreach(var drink in drinks)
173          {
174              if (drink.price > 0)
175              {
176                  PDVPrices.Add(Math.Round(drink.price + 0.17 * drink.price, 2));
177              }
178          }
179          return PDVPrices;
180      }
```

Kod poslije ispravke (metoda priceWithPDV):

```
165 //METHOD: Price with PDV
166
167 1 reference | we ran into an exception loading metrics for this method - please contact support
168 private List<Double> priceWithPDV()
169 {
170     var PDVPrices = new List<Double>();
171     var drinks = bubbleSort(_context.Drinks.ToList());
172     foreach(var drink in drinks)
173     {
174         PDVPrices.Add(Math.Round(drink.price + 0.17 * drink.price, 2));
175     }
176     return PDVPrices;
177 }
```

Kod prije ispravke (metoda calculateDailyProfit):

```
282 var dailyProfit = 5.0;
283 foreach (var drinkQuantityPair in selectedDrinks)
284 {
285     var drink = drinksFromDb.FirstOrDefault(d => d.id == drinkQuantityPair.DrinkId);
286     if (drink != null)
287     {
288         dailyProfit += drink.price * drinkQuantityPair.Quantity;
289     }
290 }
291
292 ViewBag.DailyProfit = dailyProfit;
293 Console.WriteLine(dailyProfit.ToString());
294 var message = optimalProfitMessage(dailyProfit); // Call the optimalProfitMessage function
295 return Json(new { dailyProfit = dailyProfit, message = message });
296 }
```

Kod poslije ispravke (metoda calculateDailyProfit):

```
292
293 public class DrinkQuantityPair
294 {
295     public int DrinkId { get; set; }
296     public int Quantity { get; set; }
297 }
298
299 public Tuple<double, string> CalculateDailyProfit(List<DrinkQuantityPair> selectedDrinks)
300 {
301     var drinkIds = selectedDrinks.Select(dq => dq.DrinkId).ToList();
302     var drinksFromDb = _context.Drinks.Where(d => drinkIds.Contains(d.id)).ToList();
303
304     var dailyProfit = 0.0;
305     foreach (var drinkQuantityPair in selectedDrinks)
306     {
307         var drink = drinksFromDb.FirstOrDefault(d => d.id == drinkQuantityPair.DrinkId);
308         if (drink != null)
309         {
310             dailyProfit += drink.price * drinkQuantityPair.Quantity;
311         }
312     }
313
314     var message = optimalProfitMessage(dailyProfit);
315
316     return Tuple.Create(dailyProfit, message);
317 }
318
319
320 }
```