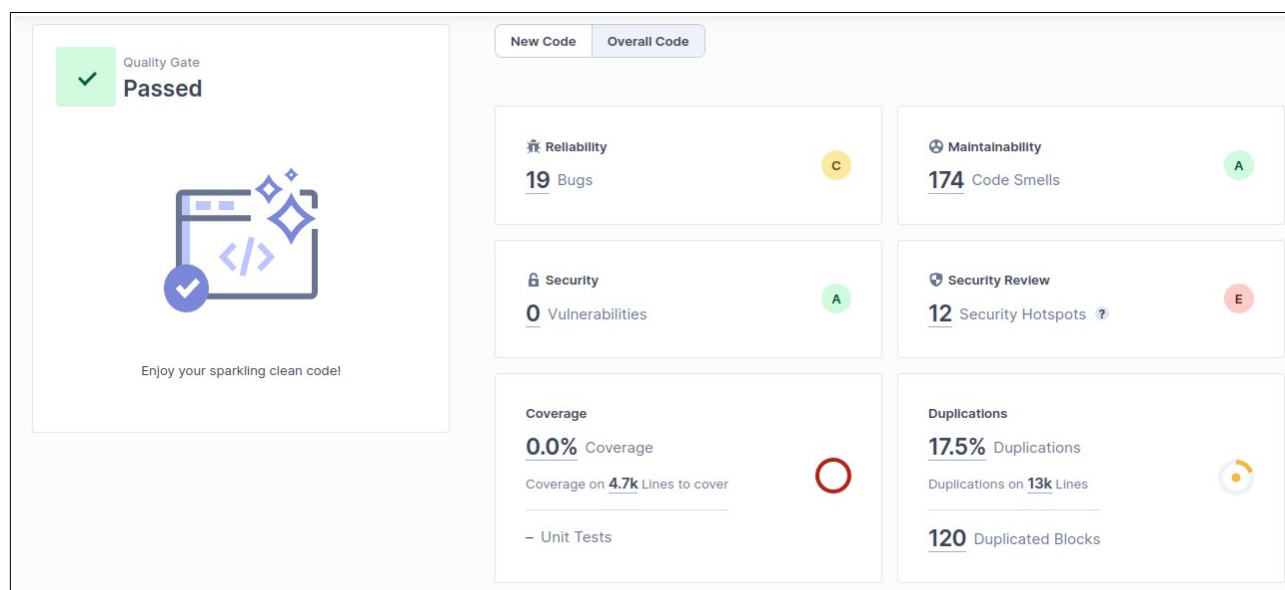


# Izveštaj statičke analize koda StayInn projekta

Korišćeni alati: SonarQube + SonarScanner

Pri prvoj analizi projekta, otkriveno je 19 bug-ova, 174 problema sa code smell, 0 sigurnosnih ranjivosti, 12 security hotspot-ova koje je potrebno proveriti, kao i 120 dupliranih blokova u 13000 linija koda. (Slika 1)

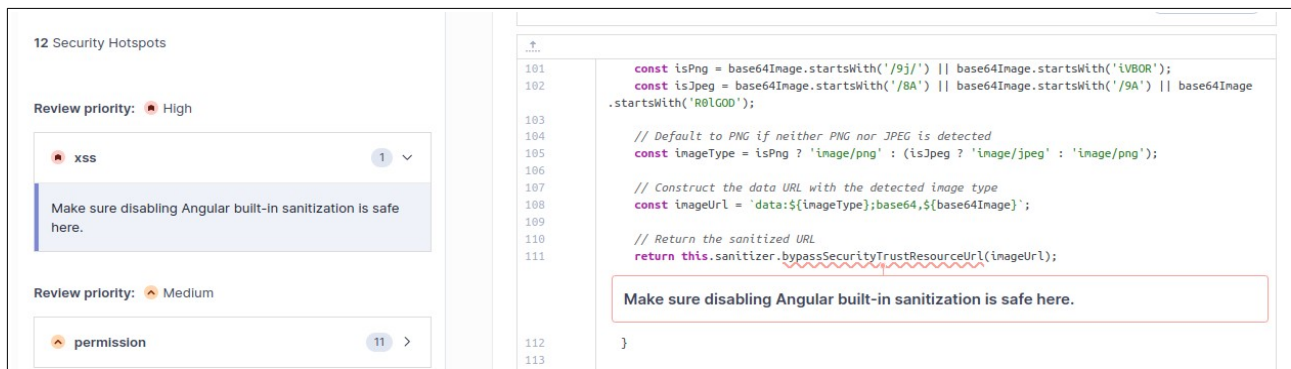


Slika 1: pregled osnovnih informacija o analizi

## **Security Hotspots**

Kod ovog dela analize otkriven je jedan hotspot visokog nivoa i 11 srednjeg nivoa. Hotspot visokog nivoa se ticao konstruisanja saniranog url-a za slike prisutne na frontend-u aplikacije (Slika 2). Prilikom njegovog formiranja, Angular ugrađeno saniranje je bilo onemogućeno, što je predstavljalo opasnost od XSS napada.

Jedanaest hotspot-ova srednjeg nivoa su bili vezani za permisije. Prva vrsta hotspot-ova je bila kod rekurzivnog kopiranja definisanog u Dockerfile-u servisa, gde može doći do dodavanja osetljivih informacija u kontejner (Slika 3). Druga vrsta, isto vezana za Dockerfile, je ta što se Docker slike pokreću sa root korisnikom kao podrazumevanim, što znači da ima permisije da izvrši sve komande kao root (Slika 4).



Slika 2: deo koda u riziku od XSS napada



Slika 3: hotspot vezan za rekurzivno kopiranje



Slika 4: hotspot o pokretanju kontejnera sa root korisnikom

## Security Hotspots – rešenje

Slika 2 – kada se vrši upload slike na frontend aplikaciji, postoji logika za proveru tipa fajla u okviru komponente. Ukoliko korisnik pokuša da prosledi nešto što nije slika, fajl će biti odbijen ili ukoliko promeni HTML input elementa. Ukratko, u vreme kada se koristi pomenuti kod, URL slike je sigurno validan.

Slika 3 – ovo je standard kodiranja i bez rekurzivnog kopiranja postoji rizik da build ne uspe. Ukoliko se rekurzivno kopiranje nebi koristilo, mora se svaki fajl ručno navesti u Dockerfile-u.

Slika 4 – uzimajući u obzir da tok komunikacije ide: frontend aplikacija -> gateway -> kontejner, postoji dovoljno slojeva između korisnika i kontejnera, što smanjuje rizik od pokretanja kontejnera sa root korisnikom, čije su privilegije potrebne za pisanje u log fajlove

## Issues

U issue delu analize, kod atributa clean code-a, otkriveno je 66 problema sa konzistentnošću, 83 problema sa namenom i 44 problema sa adaptibilnošću, bez problema sa odgovornosti. U okviru kvaliteta softvera, otkriveno je 0 problema sa sigurnošću, 19 sa pouzdanosti i 174 sa održavanjem (Slika 5).

Overview	Issues	Security Hotspots
▼ Clean Code Attribute		
Consistency		66
Intentionality		83
Adaptability		44
Responsibility		0
▼ Software Quality		
Security		0
Reliability		19
Maintainability		174

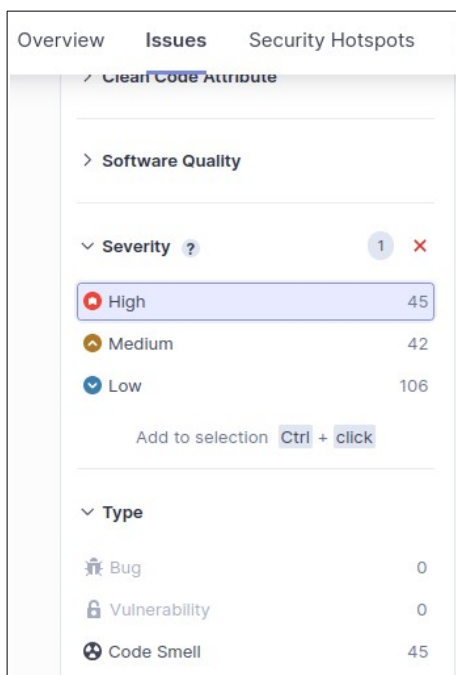
Slika 5: prikaz Issue tab-a

Od navedenih problema, 45 su bili visokog nivoa, 42 srednjeg nivoa i 106 niskog nivoa. Kao kategorizovani tipovi, podeljeni su na 19 bug-ova i 174 code smell-a (Slika 6).

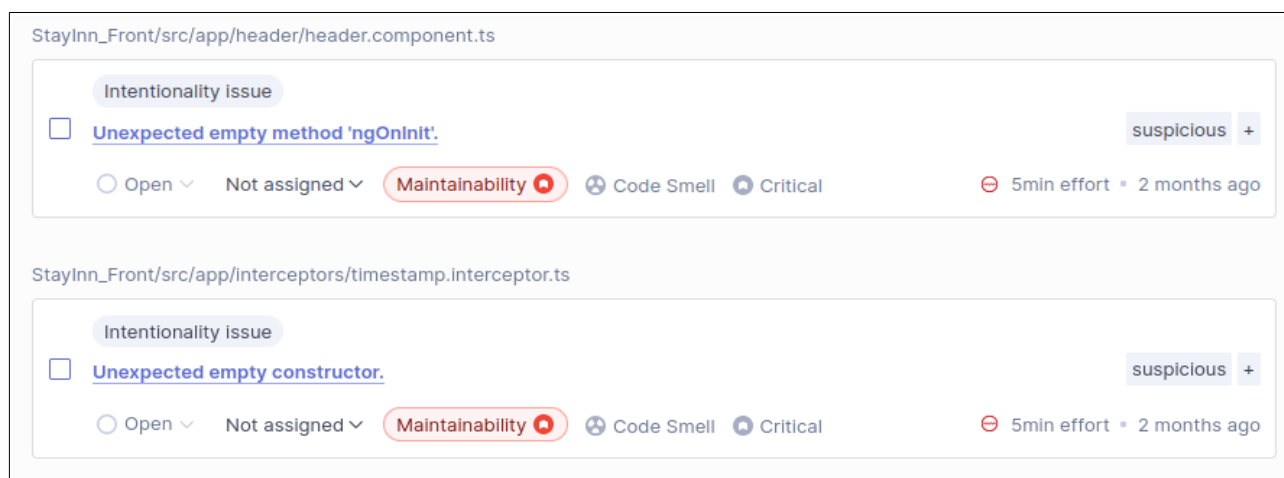
Overview	Issues	Security Hotspots
> Clean Code Attribute		
> Software Quality		
▼ Severity ?		
🔴 High		45
🟡 Medium		42
🟢 Low		106
▼ Type		
🐛 Bug		19
🔒 Vulnerability		0
🕸 Code Smell		174

Slika 6: prikaz nivoa i tipova na Issue tab-u

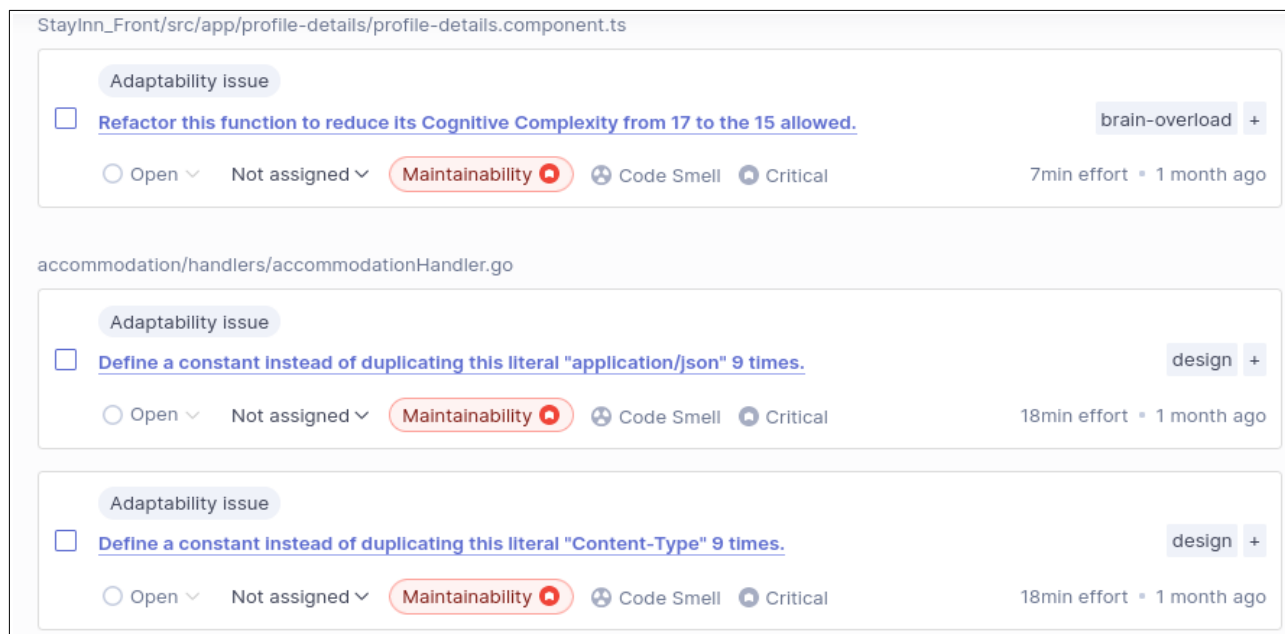
Kada se izdvoje po nivoima, na visokom nivou je postojalo samo 45 code smell problema. Veliki deo problema je bio sa kognitivnom kompleksnošću i dizajnom koda, kao i par sumnjivih delova koda, kao što su prazne metode i konstruktori (Slike 7, 8, 9).



Slika 7: prikaz problema visokog nivoa



Slika 8: sumnjivi kod sa praznim metodama i konstruktorima

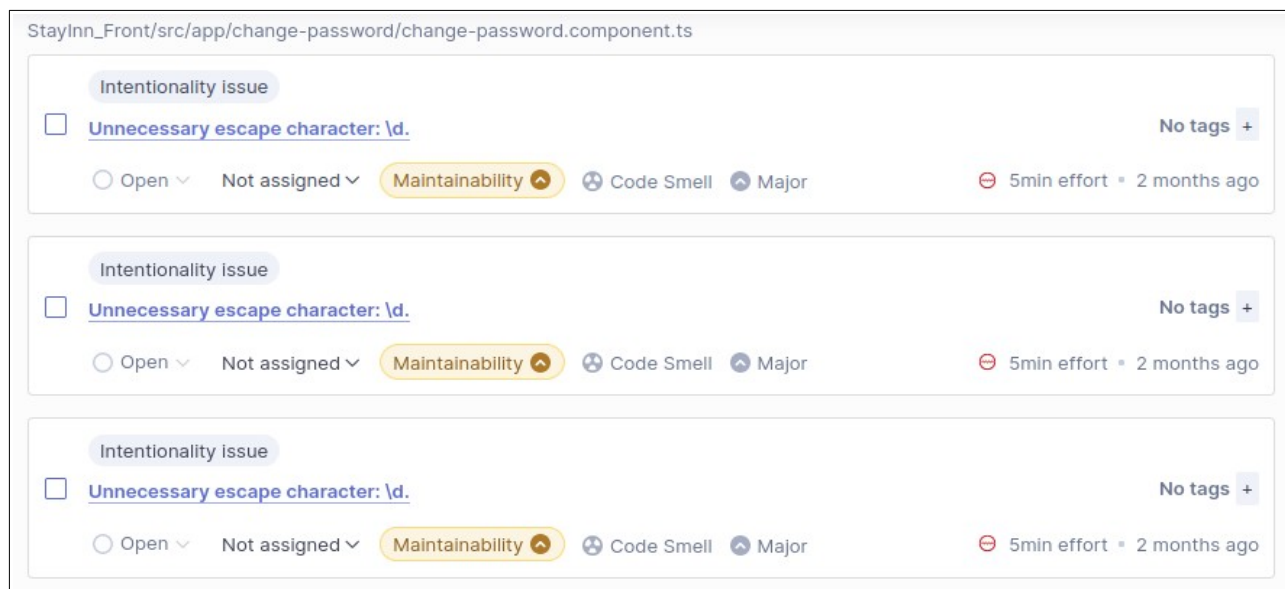


Slika 9: problemi sa kognitivnom kompleksnošću i dizajnom koda

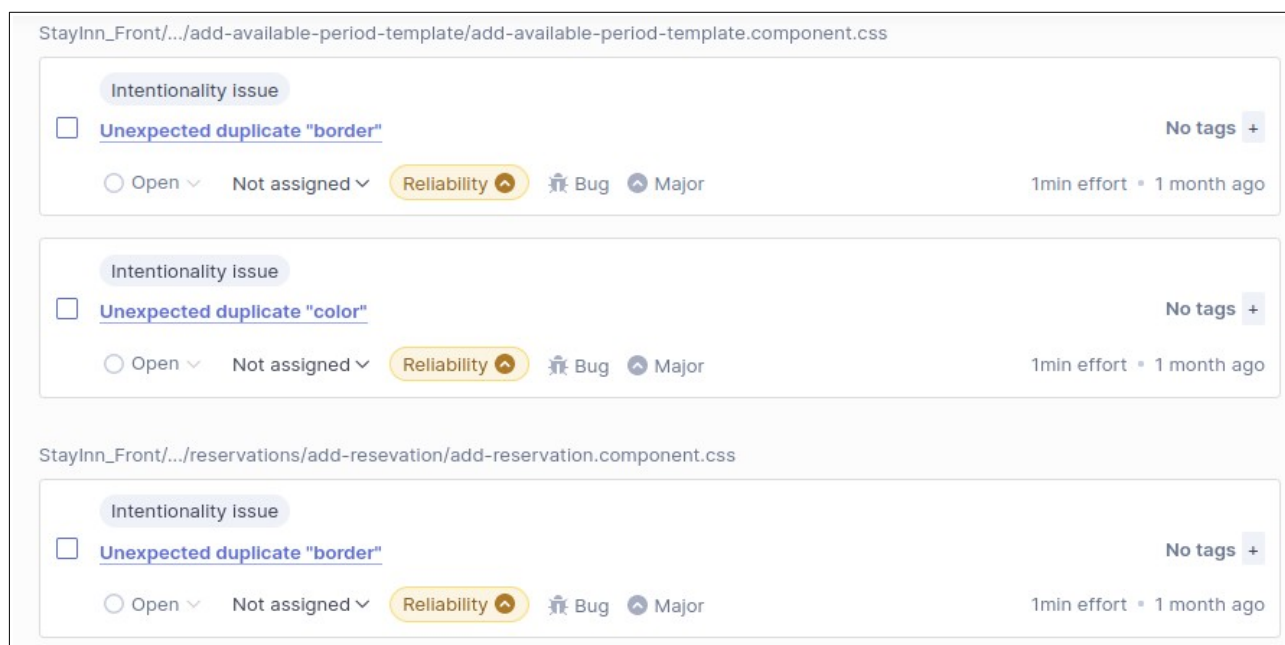
Na srednjem nivou problema, postojalo je 7 bug-ova i 35 code smell problema. Kao prva grupacija je izdvojena nepotrebno korišćenje escape karaktera u regex izrazima. Skoro svi bug-ovi su prouzrokovani duplikatima CSS stilizacije, u kojoj je i dosta problema sa zakomentarisanim kodom. Kao poslednja grupacija na ovom nivou je problem konzistentnosti u Dockerfile, gde po konvenciji "as" se piše velikim slovima i nije naveden verzioni tag za sliku (Slike 10, 11, 12, 13, 14).

Overview	Issues	Security Hotspots
Clear Code Attribute		
> Software Quality		
Severity ?		1 ✕
<input checked="" type="radio"/> High		45
<input checked="" type="radio"/> Medium		42
<input type="radio"/> Low		106
Add to selection <b>Ctrl + click</b>		
Type		
<input checked="" type="radio"/> Bug		7
<input type="radio"/> Vulnerability		0
<input checked="" type="radio"/> Code Smell		35

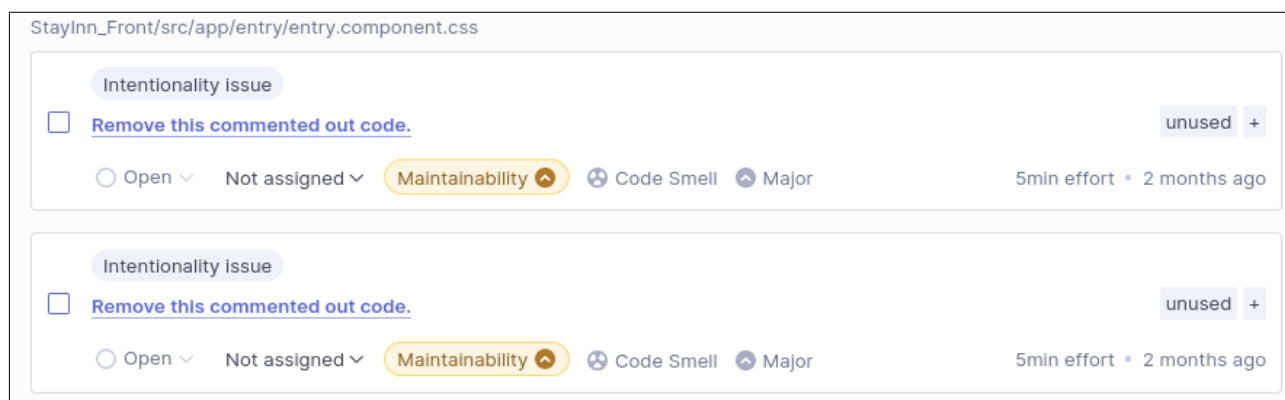
Slika 10: prikaz problema srednjeg nivoa



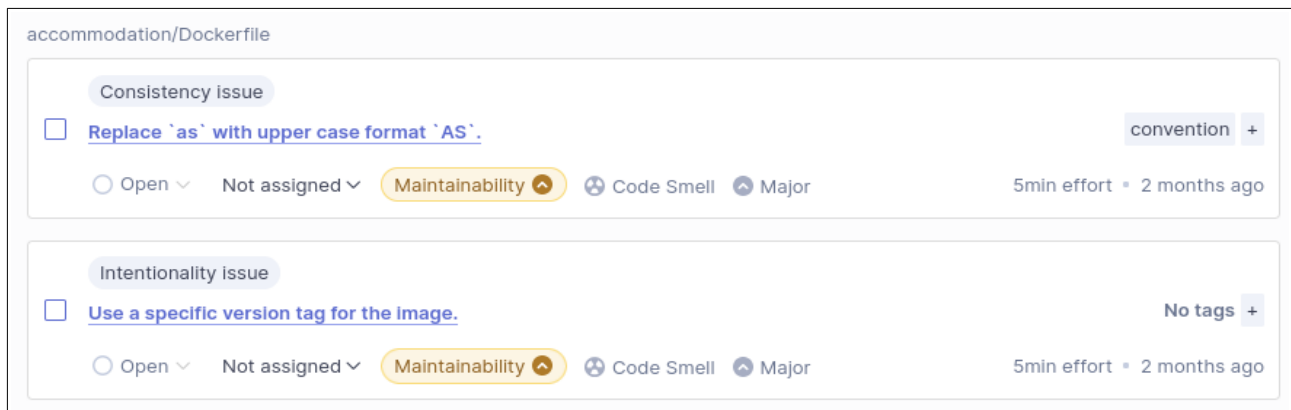
Slika 11: problem sa nepotrebnim escape karakterima



Slika 12: problem sa duplikatima u CSS fajlovima



Slika 13: problem sa zakomentarisanim kodom



Slika 14: problemi u Dockerfile

### **Issues – značaj problema**

Slika 8 – smatra se lošom praksom, dovodi do konfuzije i problema sa čitanjem koda. Prazne metode nemaju funkcionalnost i mogu dovesti da zablude kod drugih osoba da ispunjavaju neki zahtev. Takođe je suprotno typescript:S1186 pravilu. Rešenje je ukloniti takav kod.

Slika 9 – visoka kognitivna kompleksnost otežava praćenje toka kontrole koda. Takav kod je težak za čitanje, razumevanje, održavanje i potencijalne modifikacije. Rešenje je izdvojiti kod na manje funkcionalne celine, nešto slično SOLID principima. Duplirani string-ovi čine proces refaktorisanja podložnim greškama, jer se moraju ažurirati sve instance tog string-a. Takođe je suprotno go:S1192 pravilu. Rešenje je definisati konstantu koja ima vrednost navedenu u string-u.

Slika 11 – ovaj problem je false positive, radi se o regex izrazima, gde "\d" predstavlja sve brojeve, kao i kod drugih escape karaktera, gde su potrebni da bi regex izraz bio validan.

Slika 12 – CSS dozvoljava duple osobine, međutim primenjuje samo poslednji navedeni stil. Ovo može dovesti do konfuzije i suprotno je css:S4656 pravilu. Rešenje je koristiti osobine samo jednom.

Slika 13 – zakomentaran kod odvlači pažnju od koda koji se izvršava i stvara "šum" koji otežava održavanje koda. Često je zakomentaran kod i nevalidan. Rešenje je obrisati zakomentaran kod i povratiti ga ukoliko je potrebno kroz istoriju sistema za verzioniranje.

Slika 14 – instrukcije u Dockerfile treba pisati po konvenciji velikim slovima, inače je suprotno docker:S6476 pravilu. Time se poboljšava čitkost i kolaboracija između članova tima i lakše se uvidi razlika između parametara i instrukcija. Rešenje je pisati sve instrukcije velikim slovima.



Nenavodeći verzioni tag kod slika, koristiće se uvek poslednja verzija te slike, što čini replikaciju build-a nemogućim, jer se ne zna koja je verzija korišćena za build. Dodatno može dovesti do nepredvidivosti i problema sa verzijama, kao i potencijalnih problema sa bezbednošću. Rešenje je koristiti specifičnu verziju slika.

### Issues – rešenje

Slika 8 – obrisane prazne metode i konstruktori

Slika 9 – string literali su zamenjeni sa konstantama koje sadrže njihovu vrednost. Kognitivna kompleksnost funkcija nije menjana.

Slika 11 – false positive

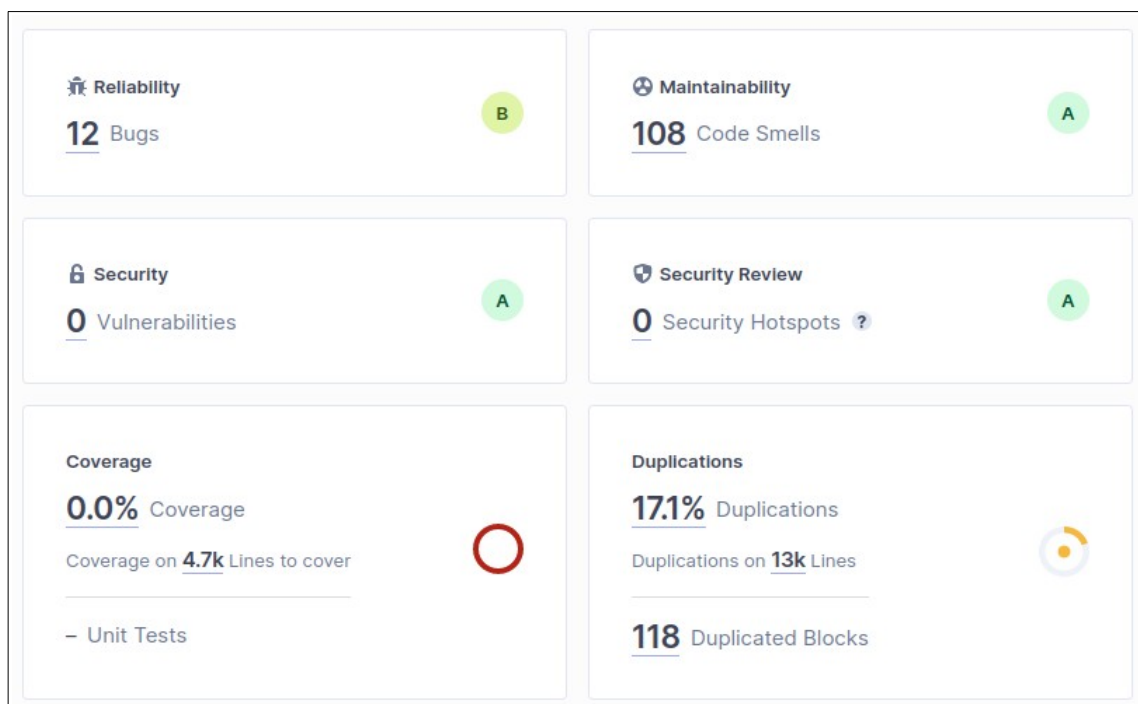
Slika 12 – obrisane duple osobine u stilovima

Slika 13 – obrisani zakomentarisani kod

Slika 14 – instrukcije su napisane velikim slovima i dodate su verzije Docker slika koje se koriste u projektu

### Analiza koda nakon popravki

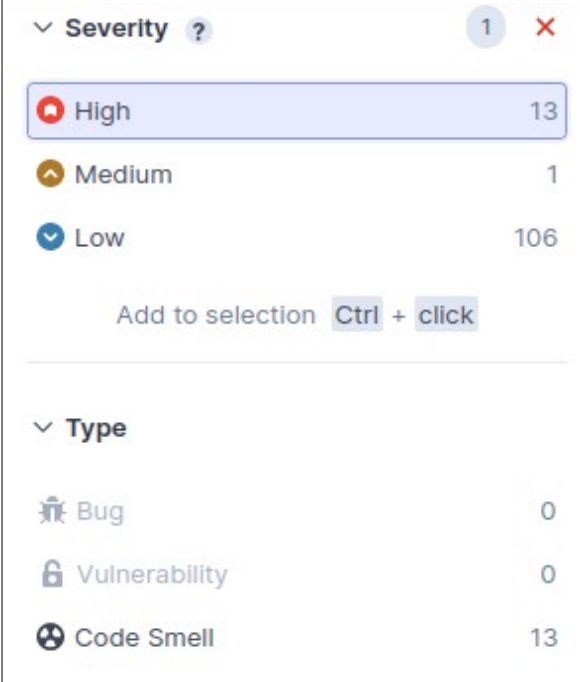
Nakon izvršenih popravki koda prema savetima SonarQube alata, broj bug-ova je smanjen sa 19 na 12, broj problema code smell-a sa 174 na 108 i svih 12 security hotspot-ova je ili obeleženo kao safe ili obeleženo kao acknowledged (Slika 15).



Slika 15: stanje projekta nakon popravki

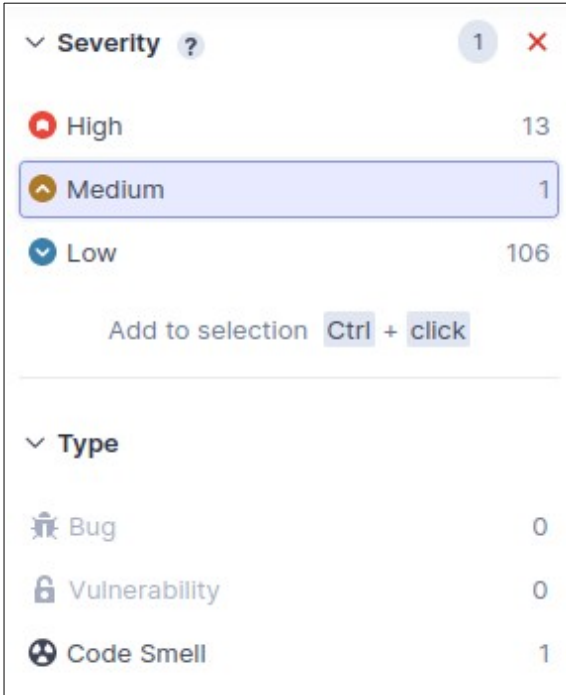


Broj high level problema je smanjen sa 45 na 13, ostali su samo problemi sa kognitivnom kompleksnošću (Slika 16). Problemi srednjeg nivoa su smanjeni sa 42 na 1, jedini problem je korišćenje latest verzionog taga za nginx Docker sliku (Slika 17).



▼ Severity ?		1	×
High	13		
Medium	1		
Low	106		
Add to selection Ctrl + click			
▼ Type			
Bug	0		
Vulnerability	0		
Code Smell	13		

Slika 16: problemi visokog nivoa nakon popravki



▼ Severity ?		1	×
High	13		
Medium	1		
Low	106		
Add to selection Ctrl + click			
▼ Type			
Bug	0		
Vulnerability	0		
Code Smell	1		

Slika 17: problemi srednjeg nivoa nakon popravki

Severity ?	
High	13
Medium	1
Low	106
Type	
Bug	12
Vulnerability	0
Code Smell	108

Slika 18: konačni pregled problema