

Analiza SOLID principa za dijagram klasa

1. Princip pojedinačne odgovornosti

je u slučaju ovog dijagrama klasa ispoštovan. Naime, svaka klasa je zasebna cjelina i posjeduje samo atribute koji su direktno vezani za njen rad. Pošto će se koristiti entity framework, sve metode i interfejsi će biti implementirani kao kontroleri unutar MVC-a. Samim time, nijedna klasa neće ovisiti o implementacijama metoda i izmjenama istih unutar projekta.

2. Otvoreno zatvoren princip

Gledajući strukturu klasa unutar ovog projekta i njihov dijagram vidimo da je i ovaj princip ispoštovan. Naime, sve klase su definirane i kreirane na način da su zasebne cjeline koje ne dozvoljavaju modifikacije, ali su itekako otvoreni za bilo koji oblik nadogradnje, u slučaju potrebe.

3. Liskov princip zamjene

Ne možemo razmatrati u potpunosti ovaj princip po njegovoj definiciji jer unutar ovog projekta nemamo principa nasljeđivanja. No, možemo primjetiti da unutar nekih klasa imamo strukturu tipova (npr. tipovi akaunta, tipovi vozila..), sve ovo će biti realizirano pomoću dodatnih tabela u bazama podataka, tako da možemo reći i da je ovaj princip zadovoljen, time što je svaki podtip zamjenjiv svojim osnovnim tipom (npr. autobus-vozilo..).

4. Princip inverzije ovisnosti

Ponovo, ovaj princip ne možemo posmatrati po njegovoj definiciji jer unutar projekta nema principa nasljeđivanja. No, gledajući na naš projekat, možemo vidjeti da glavne klase neće biti modifikovane, ali da postoji mogućnost dodavanja i modificiranja podtipova naših klasa. Tako da, glavne klase ne ovise o podklasama. Dakle, i ovaj princip je zadovoljen.

5. Princip izoliranja interfejsa

Ovaj princip je zadovoljen, time što će sve metode i interfejsi biti implementirane u kontrolerima, tako da će samo određene vrste korisnika imati određene, za njih specifične kontrolere koje će moći koristiti, samim time neće ovisiti o kontrolerima (metodama) koje neće upotrebljavati.