

## Code refactoring

- Uspjele smo implementirati sve funkcionalnosti koje su planirane.
- Poštujući SOLID principe i implementirajući paterne, postigle smo:
  - čitljiv i razumljiv kod
  - koristile smo automatsko generisanje koda gdje je bilo moguće
  - razdvojile smo kontrolere od pogleda
  - klase koje služe samo jednoj svrsi
  - nemamo ponavljanje koda
  - nemamo metode koje ne rade ništa
  - nazivi klasa i atributa prate sintaksna pravila
- Svi dijagrami i specifikacije prikazuju posljednje stanje sistema tako da nemamo proturječnosti.
- ERD je identičan dijagramu klasa jer smo izgenerisale ERD nakon posljednje migracije
- Tokom same implementacija projekta smo shvatile da nam, u suštini, nije potreban veliki broj kontrolera kao u prvobitnoj verziji iz razloga što su određeni pogledi informacionog sadržaja i bilo je sasvim dovoljno da npr HomeController upravlja nad njima. Sve potrebne funkcionalnosti kreiranja novih instanci ili algoritama koji vrše obradu podataka smo uspjeli implementirati u nekoliko metoda jednog kontrolera.
- Kreirali smo pomoćnu tabelu *Dtorequest* za lakše bindanje podataka
- Najvažnija promjena je bila to što smo potpuno izbacile apstraktnu klasu Person te naslijeđene klase Employee i Customer jer smo shvatile da sve odnose i funkcije možemo riješiti preko ASPNetUser klase i rola.

- Dodali smo i neke dodatne atribute u već postojećim tabelama kako bismo mogli vršiti efikasniju obradu podataka za poboljšanje funkcionalnosti sveukupnog sistema.
- Tokom implementacije smo shvatili da nam je potrebna i tabela Cart koja nam je olakšala interakciju korisnika sa svojim shopping cart proizvodima u smislu brisanja i dodavanja novih prije samog checkout-a u kojem bi on izvršio plaćanje i napravio finalnu narudžbu.
- Na kraju smo ipak implementirale Singleton pattern umjesto Prototype patterna, zbog nedostatka vremena ali i zato što nam je Singleton bio neophodan pri realizaciji asinhronne funkcionalnosti.