

Refaktoring

eLiveShow

Faris Crnčalo

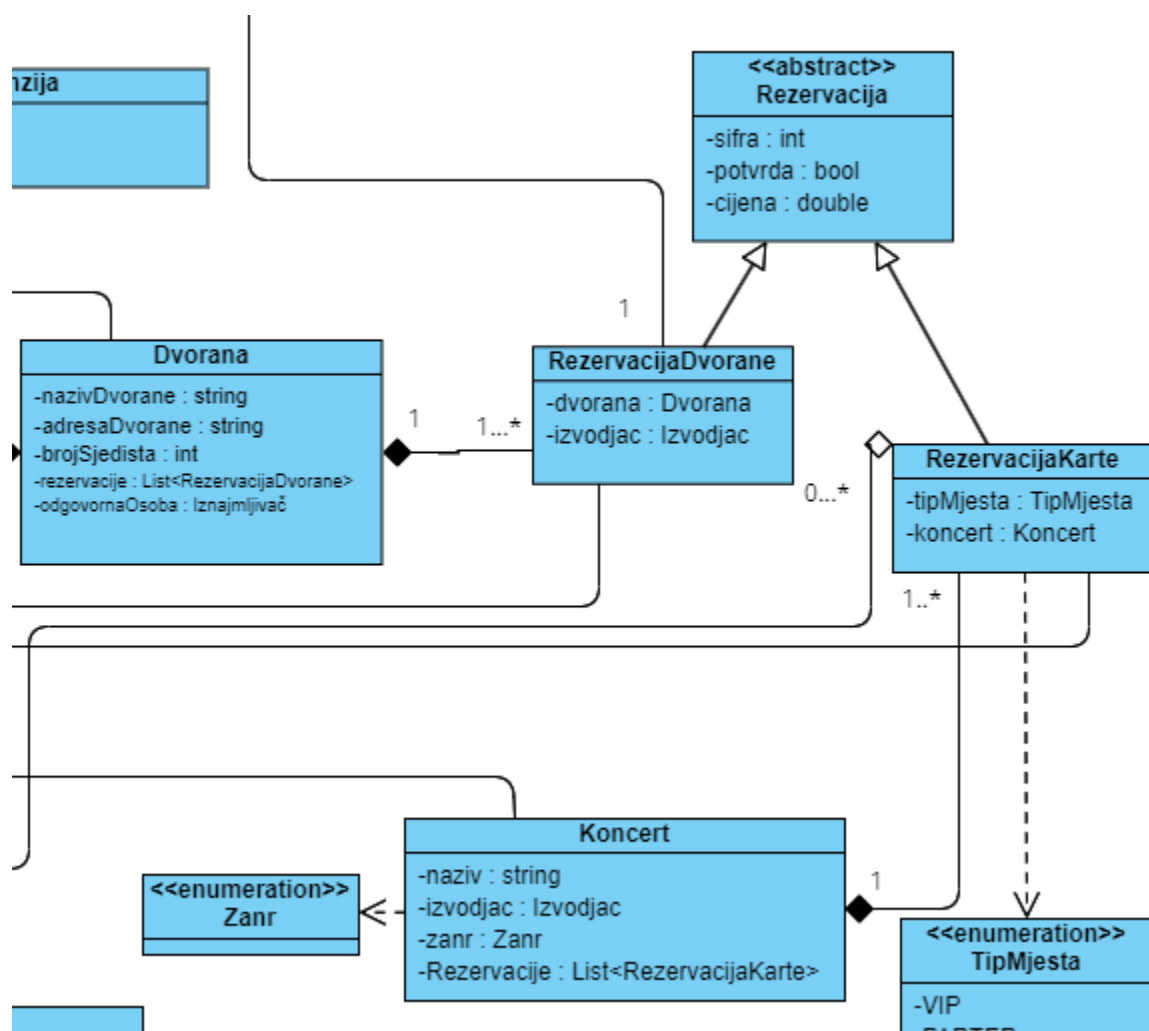
Aldin Islamagić

Anes Ćenanović

Muhamed Mujić

Eldar Babić

Jedan od najranijih problema sa kojim smo se susreli prilikom realizacije našeg projekta je bila dilemma oko veze Koncert-Dvorana. Imamo dvije klase, Dvorana i Koncert, koje su zamišljene na način da jedna dvorana može prihvatiti više koncerata. Uvidjeli smo da će nam biti potrebna neka međuklasa Rezervacija preko koje je moguće uspješno uspostaviti relaciju između pomenutih klasa. Ova međuklasa je kasnije proširena u RezervacijaDvorane i RezervacijaKarte, te sadrži druge atribute koji su nam korisni prilikom implementacije.



Također smo primjetili da smo prilikom realizacije Home Page-a koristili mnogo varijabli koje mogu uzrokovati neke konfuzije, pogotovo ako uzmemo u obzir da je svaki član našeg tima radio na različitim funkcionalnostima. Uzimajući ovo u obzir, obratili smo posebnu pažnju da svaka varijabla svojim imenom i pripadnom funkcijom jasno i koncizno predstavlja svoju funkcionalnost, kao na slici:

```
@inject IDvoranaManager dvoranaManager
@inject IRezervacijaManager rezervacijaManager
@inject UserManager<IdentityUser> UserManager
@{
    ViewData["Title"] = "Home Page";
    var concerts = await koncertManager.GetAll();
    var recommended = koncertManager.GetRecommended();
    var owned = TicketManager.GetOwned(UserManager.GetUserId(User));
    var concertHalls = await dvoranaManager.GetAll();
    var concerHallsThatAreNotTaken = await dvoranaManager.GetUnreserved();
    var reservedConcertHalls = await dvoranaManager.GetReservedByCurrentPerformer();
    var ownedHalls = await dvoranaManager.GetOwnedByCurrentRenter();
    var reservedHalls = await dvoranaManager.GetReservations();
}
```

U svrhu pravilnog pisanja urednog koda, trudili smo se da mijenjamo “magične brojeve” sa konstantnim varijablama, pogotovo u slučajevima kad se ta vrijednost koristi više puta:

```
var recenzija = new Recenzija();
var defaultRating = 5;
recenzija.izvodjac = artist;
recenzija.izvodjacId = artist.Id;
recenzija.rating = defaultRating;
return View(recenzija);
```

Konačno, svugdje gdje smo koristili neke temporary varijable, zamijenili smo ih sa imenima varijabli koje precizno imenuju sta ta varijabla radi, tako da smo umjesto “temp” ili “rez” ovdje koristili “owned” za prikaz dvorana koje trenutni vlasnik posjeduje:

2 references

```
public async Task<IEnumerable<Dvorana>> GetOwnedByCurrentRenter()  
{  
    var owned = await _db.Dvorane.Where(d => d.iznajmljivacId == GetUserId()).ToListAsync();  
    return owned;  
}
```