Projektni zadatak - Konzistencija

Ognjen Čavić E2 161/2024

November 25, 2025

Sadržaj

1 Uvod

Potrebno je napraviti sistem u kojem 10 senzora na nasumičnim vremenskim itervalima generišu nasumične vrednosti, dok ”menadžer” vrši njihovo poravnjanje, tj. bira jednu od tih vrednosti koja će biti proglašena glavnom. Ovaj sistem se može opisati kao skup od 3 glavne komponente:

- **Simulator** - Pokreće po jednu nit za svaki senzor. Svaka nit na vremenskom intervalu između 1 i 10 sekundi generiše novu vrednost i upisuje je u sopstvenu tabelu u bazi podataka pomoću *Entity Framework*-a.
- **Menadžer**- Na svakih 60 sekundi vrši poravnanje, tako što bira čitanje senzora koje je najpovoljnije i upisuje u tabele svih senzora. Kriterijum za izbor najbolje vrednosti je najskorije očitavanje u prihvatljivom opsegu $\pm 5^oC$. Ukoliko takvo ne postoji uzima se najskorije.
- **WCF servis** - Definiše način komunikacije između menadžera i senzora.

2 Organizacija rešenja

Rešenje se sastoji od 4 manja projekta, gde svaki predstavlja jednu od prethodno navedenih celina, sa tim da dodati četvrti projekat opisuje način struktuiranja podataka za upis u bazu:

- ConsistencyService - WCF servis
- ConsistencyManager - menadžer koji vrši poravnanje
- Sensors - pokretanje niti koje simuliraju senzore
- SensorsDatabaseContext - kontekst baze podataka

2.1 ConsistencyService

2.1.1 ConsistencyService/ConsistencyService.cs

Klasa **TemperatureInfo** opisuje oblik podataka koji se razmenjuju između korisnika servisa i formiraju ga 4 polja:

1. **sensor_id** - Identifikacioni broj senzora, broj od 1 do 10
2. **temperature** - Vrednost temperature
3. **timestamp** - vremenski podatak
4. **source** - izvor podatka, tj. ko ga upisuje u bazu, senzor ili menadžer tokom poravnanja

U ovom fajlu su samo još definisani interfejsi servisa. **IPublisher** definiše metod za upis u bazu, dok interfejs **ISubscriber** definiše metod za slanje upita.

2.1.2 ConsistencyService/ConsistencyService.svc.cs

Implementacija **IPublisher** i **ISubscriber** interfejsa unutar **ConsistencyService** klase. Bitno je primetiti da klasa nasleđuje oba interfejsa. Metoda **publishTemp** upisuje zadati **TemperatureInfo**, što se koristi da i menadžer i senzori upisuju u bazu, tako što **source** polje postave na odgovarajuću vrednost. Poslednje očitavanje senzora se dobija pozivom **querySensor** i prosleđivanjem odgovarajućeg rednog broja.

2.2 ConsistencyManager

2.2.1 ConsistencyManager/Program.cs

Sadrži dve funkcije, **Main** i **findBest**. **Main** funkcija na svakih 60 sekundi šalje upit bazi podataka za poslednji podatak iz svake tabele, potom pomoću funkcije **findBest** određuje koje od čitanja senzora je bilo najbolje. Nakon toga se izmenjuju polja **source** i **timestamp** tako da se označi da je podatak upisan kao rezultat poravnanja i vreme kada je on upisan, takav podatak se upisuje u svaku od tabela. Potrebno je napomenuti da menadžer instancira *subscribera* i *publishera*, koje koristi za upite i upise u bazu, respektivno.