Naloga 1: Načrtovanje projekta

Informacijski sistem za pregled podatkov triatlona

Marko Šimunović, 2021

Analiza problema

V projektu je potrebno narediti informacijski sistem za pregled in urejanje podatkov tekmovanj v triatlonu. Končni izdelek mora omogočati pregled in urejanje podatkov, dostop do podatkov preko REST API. Prav tako potrebujemo dve aplikaciji eno namizno, katere namen je manipulacija podatkov in uporaba s strani administratorjev. Druga je pa spletna aplikacija in je namenjena navadnim uporabnikom saj omogoča le pregled podatkov.

Zahteve informacijskega sistema

Namizna aplikacija

- Uvoz podatkov
    - JSON
    - CSV
- Urejanje rezultatov
- Dodajanje rezultatov
- Brisanje rezultatov
- Dodajanje tekmovalcev
- Dodajanje prihajajočih tekmovanj
- Brisanje rezultatov
- Dodajanje uporabnikov
- Brisanje uporabnikov

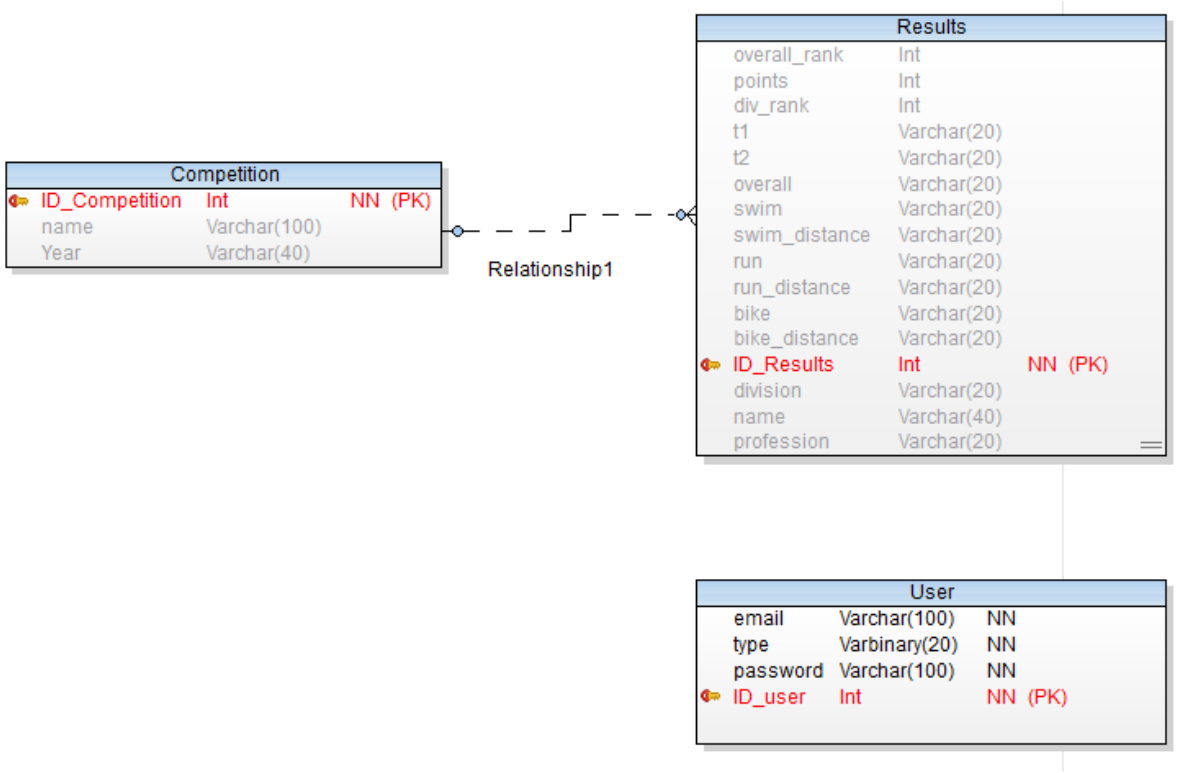
Spletna aplikacija

- Prikaz rezultatov
- Prikaz statistike po tekmovalcu
- Prikaz tekmovalcev
- Prikaz najboljših
- O organizacijah
- Prihajajoča tekmovanja
- Registracija
- Prijava
- Izbris svojega računa

Podatkovna baza

Podatkovna baza je bila načrtovana na podlagi zahtev, ki smo jih pridobili s strani naročnika oz. navodil. Pri načrtovanju sem ugotovil, da so za delovanje informacijskega sistema dovolj tri tabele.

- Results
  - Tabela za shranjevanje rezultatov
- Competition
  - Tabela za shranjevanje tekmovanj
- User
  - Tabela za hranjenje uporabnikov



Realizacija podatkovne baze

Podatkovna baza SQL je bila narejena s pomočjo ogrodja Entity Framework na podlagi objektnega modela narejenega v programskem jeziku C#.

Razred *Results*

```
0 references
public Results()
{
    this.Competitions = new HashSet<Competition>();
}

1 reference
public Results(string[] arr)...



string[] values;


[Key]
//[DataMember]
0 references
public int ID { get; set; }


3 references
public virtual ICollection<Competition> Competitions { get; set; }
// [DataMember]
1 reference
public string Name { get; set; }
//[DataMember]

1 reference
public string GenderRank { get; set; }
//[DataMember]

1 reference
public string DivRank { get; set; }
// [DataMember]
1 reference
public string OverallRank { get; set; }
//[DataMember]
1 reference
public string Bib { get; set; }
```

```
1 reference
public string Division { get; set; }
//[DataMember]
1 reference
public string Age { get; set; }
// [DataMember]
1 reference
public string State { get; set; }
//[DataMember]
1 reference
public string Country { get; set; }
//[DataMember]
1 reference
public string Profession { get; set; }
//[DataMember]
1 reference
public string Points { get; set; }
// [DataMember]
1 reference
public string Swim { get; set; }
//[DataMember]
1 reference
public string SwimDistance { get; set; }
// [DataMember]
1 reference
public string T1 { get; set; }
// [DataMember]
1 reference
public string Bike { get; set; }
// [DataMember]
1 reference
public string BikeDistance { get; set; }
// [DataMember]
1 reference
public string T2 { get; set; }
// [DataMember]
1 reference
public string Run { get; set; }
//[DataMember]
1 reference
public string RunDistance { get; set; }
```

Razred *Competition*

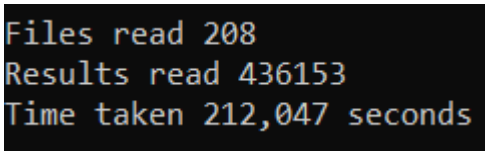
```
namespace Naloga1_DotNET
{
    12 references
    public class Competition
    {

        0 references
        public Competition()
        {
        }
        1 reference
        public Competition(string name, string year)
        {
            Name = name;
            Year = year;
        }
        [Key]
        0 references
        public int ID { get; set; }
        1 reference
        public string Name { get; set; }
        1 reference
        public string Year { get; set; }

    }
}
```

Razred *User*

```
namespace Naloga1_DotNET
{
    1 reference
    public class User
    {
        [Key]
        0 references
        public int ID { get; set; }
        0 references
        public string Email { get; set; }
        0 references
        public string password { get; set; }
    }
}
```

Polnjenje podatkovne baze

Polnjenje je bilo narejeno preko skripte napisane v jeziku C#. Program pridobi podatke iz CSV datoteke, ki jo doda uporabnik. Pri izvajanju se je pojavilo ozko grlo »Prenos podatkov v PB«, kar je izdatno povečalo izvajanje podatkovne baze. Prav tako je prihajalo do težave, da je kontekst postajal prevelik. Zato je bilo potrebno enkraten prenos omejiti na približno 10.000 podatkov hkrati.



Slika 1: Statistika polnjenja podatkovne baze