

# Proiect PSSC

Studenti: Betegh Denis

Budea Sebastian

Coste Corina

An: 4

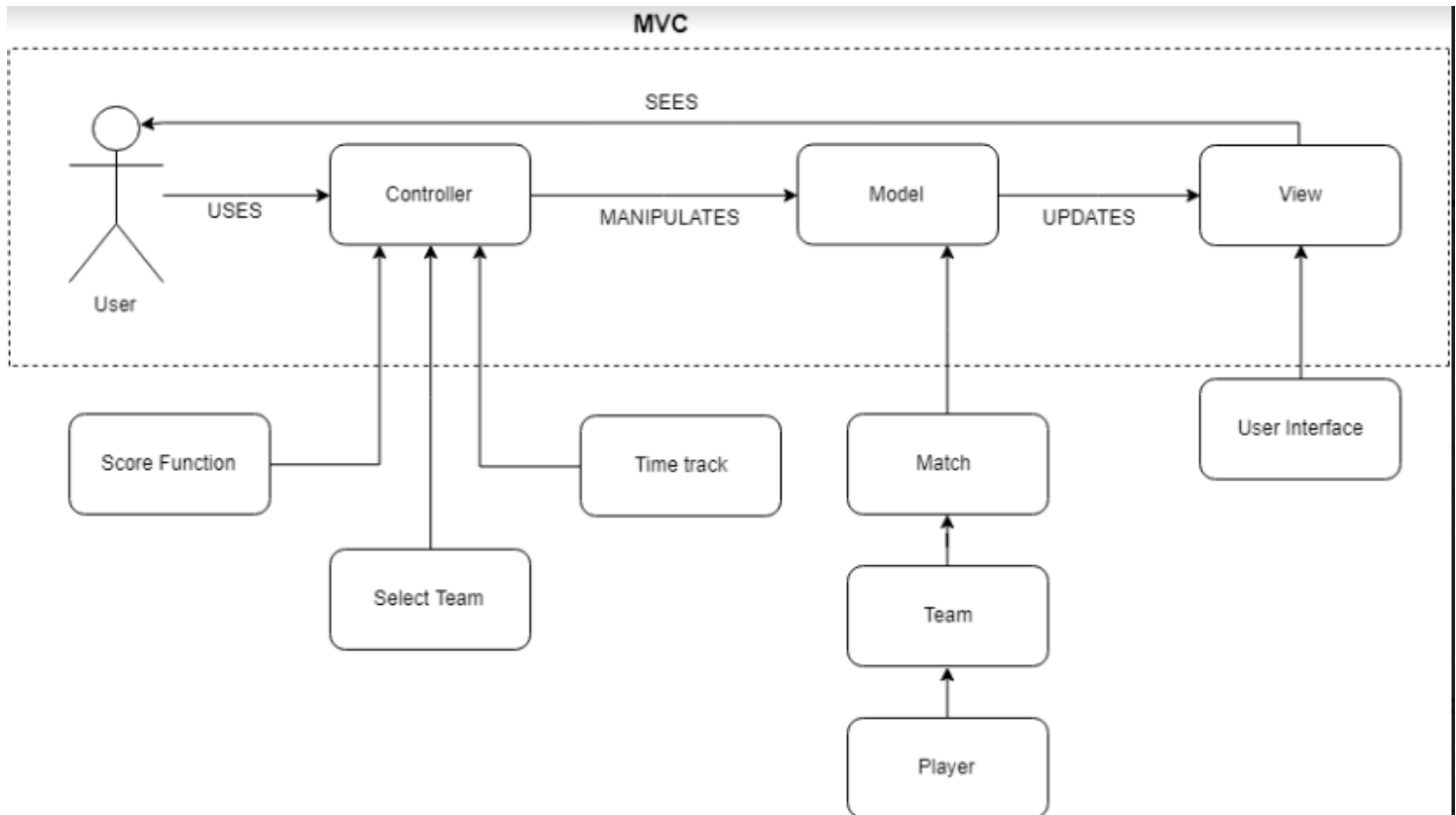
Secția: Ingineria Sistemelor

Facultate: Automatică și Calculatoare

# Tema proiectului

Am creat o aplicație care ține evidența unui meci de baschet.

## Arhitectura sistemului



În controller avem următoarele funcții:

- Score Function, care are rolul de a ține scorul
- Select Team Function, cu ajutorul căruia putem alege echipele dintr-o bază de date
- Time track Function, care afișează timpul rămas

Modelele sunt structurate astfel:

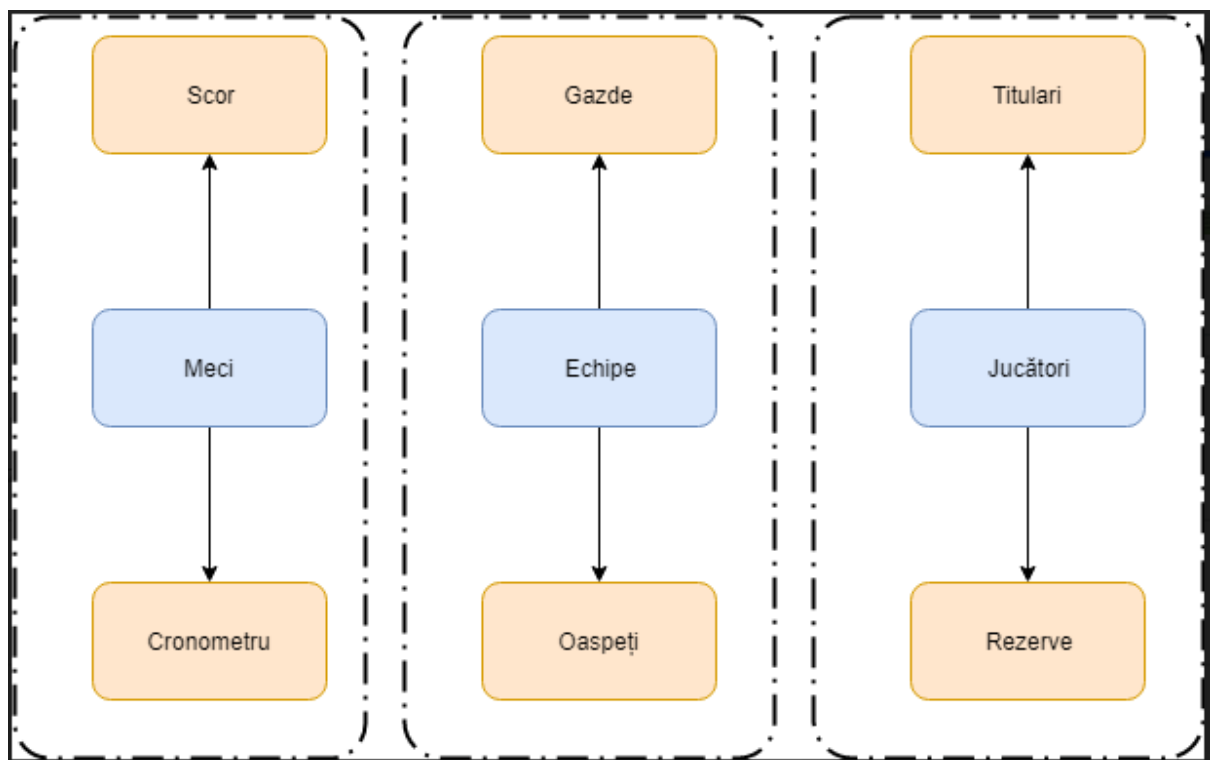
- Modelul meciului are relația “one to many” cu modelul echipe
- Modelul echipe are relația “one to many” cu modelul jucător

În view userul poate să facă următoarele lucruri:

- Adaugare puncte pe tabela de scor
- Pornirea cronometrului
- Resetarea tablei de marcaj
- Modificarea scorului în caz de eroare

## Modelul DDD

### Aggregation Roots



### Repository

Pentru fiecare aggregation root ilustrat mai sus, avem câte un repository conținând cod specific pentru fiecare tip de persistență.

Pentru Meci:

```
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace Repository.Meci
7 {
8     public class RMeci
9     {
10         private static List<Modele.Meci.Meci> _meci = new List<Modele.Meci.Meci>();
11
12         public void ADD_Meci(Modele.Meci.Meci meci)
13         {
14             var new_meci = _meci.FirstOrDefault(nm => nm.Equals(meci));
15             if (new_meci != null) throw new DuplicateWaitObjectException();
16
17             _meci.Add(meci);
18             Console.WriteLine("New match added");
19         }
20
21         public void DELETE_Meci(Modele.Meci.Meci nume_meci)
22         {
23             var del_meci = _meci.FirstOrDefault(dm => dm.Equals(nume_meci));
24             _meci.Remove(del_meci);
25             Console.WriteLine("Match deleted");
26         }
27
28         public void UPDATE_Meci()
29         {
30             /*Baza de Date*/
31             Console.WriteLine("Update Succesfull");
32         }
33     }
34 }
35
```

Pentru Echipă:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace Repository.Echipa
7 {
8     public class REchipa
9     {
10         private static List<Modele.Echipa.Echipa> _echipa = new List<Modele.Echipa.Echipa>();
11
12         public void ADD_Echipa(Modele.Echipa.Echipa echipa)
13         {
14             var new_echipa = _echipa.FirstOrDefault(ne => ne.Equals(echipa));
15             if (new_echipa != null) throw new DuplicateWaitObjectException();
16             _echipa.Add(new_echipa);
17             Console.WriteLine("New team added");
18         }
19
20         public void DELETE_Echipa(Modele.Echipa.Echipa echipa)
21         {
22             var del_echipa = _echipa.FirstOrDefault(de => de.Equals(echipa));
23             _echipa.Remove(del_echipa);
24             Console.WriteLine("Team deleted");
25         }
26
27         public void UPDATE_Echipa()
28         {
29             /*Baza de Date*/
30             Console.WriteLine("Update Succesfull");
31         }
32     }
33 }
34
```

Pentru Jucător:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace Repository.Jucator
7  {
8      public class RJucator
9      {
10         private static List<Modele.Jucator.Jucator> _jucator = new List<Modele.Jucator.Jucator>();
11
12         public void ADD_Jucator(Modele.Jucator.Jucator jucator)
13         {
14             var new_jucator = _jucator.FirstOrDefault(nj => nj.Equals(jucator));
15             if (new_jucator != null) throw new DuplicateWaitObjectException();
16
17             _jucator.Add(new_jucator);
18             Console.WriteLine("New match added");
19         }
20
21         public void DELETE_Jucator(Modele.Jucator.Jucator nume_jucator)
22         {
23             var del_jucator = _jucator.FirstOrDefault(dj => dj.Equals(nume_jucator));
24             _jucator.Remove(del_jucator);
25             Console.WriteLine("Player deleted");
26         }
27
28         public void UPDATE_Jucator()
29         {
30             /*Baza de Date*/
31             Console.WriteLine("Update Succesfull");
32         }
33     }
34 }
35
```

Entity

Pot fi indentificate printr-un identificator unic și anume prin nume.