



UNEXUS UNITED

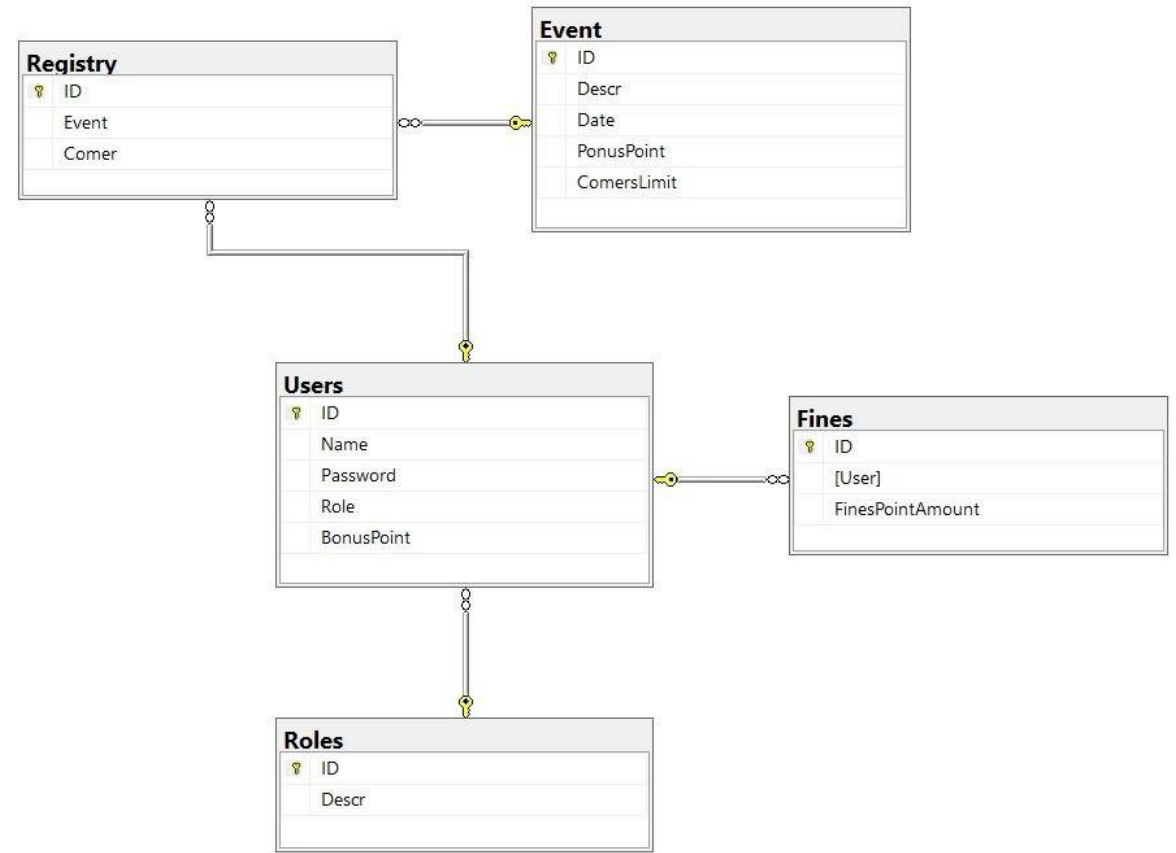
POALELUNGI ION CHIRICIUC ANNA ZACATOV ANDREI EVSTAFIEV NICU GUTU NIKITA



PROBLEM OVERVIEW



HOW CAN OUR APP RESOLVE THIS PROBLEM?



IMPLEMENTATION

```

namespace Unexus.Facade.Classes
{
    0 references
    enum Roles
    {
        Admin,
        Sponsor,
        Comer
    }
}

```

```

namespace Unexus.Facade.Classes
{
    7 references
    public class Event
    {
        1 reference
        public Event() : this(0, string.Empty, DateTime.MinValue, 0, 0) { }

        1 reference
        public Event(int id, string descr, DateTime datetime, int bonusPoint, int comersLimit)
        {
            ID = id;
            Description = descr;
            DateTime = datetime;
            BonusPoint = bonusPoint;
            ComersLimit = comersLimit;
        }

        1 reference
        public int ID { get; set; }
        1 reference
        public string Description { get; set; }
        1 reference
        public DateTime DateTime { get; set; }
        1 reference
        public int BonusPoint { get; set; }
        1 reference
        public int ComersLimit { get; set; }
    }
}

```



```
{
  4 references
  public class Fines
  {
    0 references
    public Fines() : this(0, new User(), 0) { }

    1 reference
    public Fines(int id, User user, int finesPointAmount)
    {
      ID = id;
      User = user;
      FinesPointAmount = finesPointAmount;
    }

    1 reference
    public int ID { get; set; }
    1 reference
    public User User { get; set; }
    1 reference
    public int FinesPointAmount { get; set; }
  }
}
```

```

namespace Unexus.Facade.Classes
{
    4 references
    public class Registry
    {
        0 references
        public Registry() : this(0, new Event(), new User()) { }

        1 reference
        public Registry(int id, Event evt, User comer)
        {
            ID = ID;
            Evt = evt;
            Comer = comer;
        }

        2 references
        public int ID { get; set; }
        1 reference
        public Event Evt { get; set; }
        1 reference
        public User Comer { get; set; }
    }
}

```

```

namespace Unexus.Facade.Classes
{
    7 references
    public class Role
    {
        1 reference
        public Role() : this(0, string.Empty) { }

        1 reference
        public Role(int id, string descr)
        {
            ID = id;
            Descr = descr;
        }

        1 reference
        public int ID { get; set; }
        1 reference
        public string Descr { get; set; }
    }
}

```

[-] namespace Unexus.Facade.Classes

{

10 references

[-] public class User

{

2 references

public User() : this(0, string.Empty, string.Empty, new Role(), 0

1 reference

[-] public User(int id, string name, string password, Role role, int

{

ID = id;

Name = name;

Password = password;

Role = role;

BonusPoint = bonusPoint;

}

1 reference

public int ID { get; set; }

1 reference

public string Name { get; set; }

1 reference

public string Password { get; set; }

1 reference

public Role Role { get; set; }

1 reference

public int BonusPoint { get; set; }

}

}

0 references

```
}

public int CreateEvent(string descr, DateTime datetime, int bonusPoint, int comersLimit)
{
    int eventId = 0;

    using (var connection = new SqlConnection(_connectionString))
    {
        using (var command = new SqlCommand())
        {
            command.Connection = connection;
            command.CommandText = "dbo.UpdateEvent";
            command.CommandType = CommandType.StoredProcedure;

            command.Parameters.Add(new SqlParameter("@Descr", descr));
            command.Parameters.Add(new SqlParameter("@Date", datetime));
            command.Parameters.Add(new SqlParameter("@BonusPoint", bonusPoint));
            command.Parameters.Add(new SqlParameter("@ComersLimit", comersLimit));

            using (var reader = command.ExecuteReader())
            {
                // ...todo fck
            }
        }
    }
}
```

IEnumerable<Event> GetEvents();

```
/// <summary>
/// Create new event
/// </summary>
/// <param name="descr"></param>
/// <param name="datetime"></param>
/// <param name="bonusPoint"></param>
/// <param name="comersLimit"></param>
/// <returns></returns>
```

0 references

```
int CreateEvent(string descr, DateTime datetime, int bonusPoint, int comersLimit);
```

```
/// <summary>
/// Delete event
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
0 references
void DeleteEvent(int id);
```

0 references

```
IEnumerable<Fines> GetFines();
```

```
/// <summary>
/// Выписать штраф
/// </summary>
/// <param name="userId"></param>
/// <param name="fineAmount"></param>
/// <returns></returns>
0 references
void IssueFine(int userId, int fineAmount);
```

0 references

```
IEnumerable<Registry> GetAllScheduledEvents();
```

```
/// <summary>
///
/// </summary>
/// <param name="evtId"></param>
/// <param name="comerId">User ID</param>
/// <returns></returns>
0 references
void RegisterForAScheduledEvent(int evtId, int comerId);
```

```
//-----
```

0 references

```
void CreateNewUser(string name, string password, Role role);
```

0 references


UNEXUS

LocationInterests

Event Name

Event Description

Location



List of members

Join Event !

UNEXUS

LocationInterests

Event NameName

Event NameName

Event NameName

Event NameName

Event NameName

CREATE NEW EVENT

UNEXUS

LocationInterests

UNEXUS

LocationInterests

UX/UI mockup