

Interfaccia Web di un campionato di calcio

Progetto realizzato da
Mattia Desiderio e
Giorgia Gibellini



Specifiche tecniche



- Framework : Django
- Query Language : SQLite3
- Linguaggio di programmazione : Python
- Database : locale



django



Funzionalità



USER

- Visualizzazione di tutte le squadre
- Visualizzazione dei risultati e delle statistiche delle partite per ogni squadra
- Visualizzazione di tutte le partite esistenti
- Visualizzazione della classifica
- ...

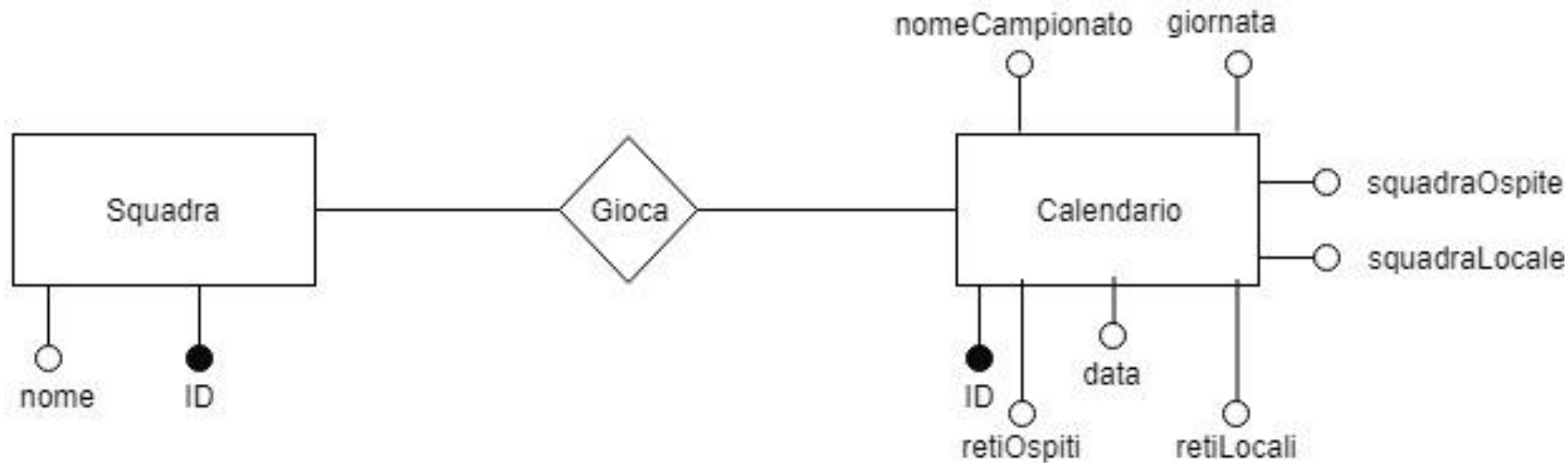


ADMIN

- Creazione nuova squadra e relative partite
- Eliminazione squadra e relative partite
- Modifica squadra
- Modifica partite
- ...



Modello ER



Principali funzioni create

- **Statistics**, presente nel file view.py nella cartella squadre.

```
# Statistiche di una squadra
def statistics(request, squadra_id):
    partite_squadra = Calendario.objects.filter(Q(squadraLocale=squadra_id)|Q(squadraOspite=squadra_id))

    def calcola_goal_punti():
        vinte=0
        pareggiate=0
        perse=0
        for conta in partite_squadra:
            if (conta.squadraLocale.id == squadra_id and conta.retiLocali > conta.retiOspiti) or (conta.squadraOspite.id ==
                squadra_id and conta.retiOspiti > conta.retiLocali):
                vinte=vinte+1
            if (conta.squadraLocale.id == squadra_id and conta.retiLocali < conta.retiOspiti) or (conta.squadraOspite.id ==
                squadra_id and conta.retiOspiti < conta.retiLocali):
                perse=perse+1
            if conta.retiLocali == conta.retiOspiti:
                pareggiate=pareggiate+1
        return vinte, perse, pareggiate, vinte*3+pareggiate

    goal_fatti_casa = Calendario.objects.filter(squadraLocale=squadra_id).aggregate(Sum('retiLocali'))
    goal_fatti_trasferta = Calendario.objects.filter(squadraOspite=squadra_id).aggregate(Sum('retiOspiti'))
    goal_subiti_casa = Calendario.objects.filter(squadraLocale=squadra_id).aggregate(Sum('retiOspiti'))
    goal_subiti_trasferta = Calendario.objects.filter(squadraOspite=squadra_id).aggregate(Sum('retiLocali'))

    squadra = get_object_or_404(Squadra, pk=squadra_id)
```

Principali funzioni create

- **Classifica**, presente nel file view.py nella cartella squadre.

```
# Classifica finale
def classifica(request):

    def calcola_classifica():
        lista_squadre = Squadra.objects.order_by('nome')
        classifica = {}
        for x in lista_squadre:
            partite_squadra = Calendario.objects.filter(Q(squadraLocale=x.id)|Q(squadraOspite=x.id))
            vinte=0
            pareggiate=0
            for conta in partite_squadra:
                if (conta.squadraLocale.id == x.id and conta.retiLocali > conta.retiOspiti) or (conta.squadraOspite.id == x.id
                    and conta.retiOspiti > conta.retiLocali):
                    vinte=vinte+1
                if conta.retiLocali == conta.retiOspiti:
                    pareggiate=pareggiate+1
            classifica[x.nome] = str(vinte*3+pareggiate)
        return list(sorted(classifica.items(), key=itemgetter(1), reverse=True))

    return render(request, 'squadre/classifica.html', {'calcola_classifica': calcola_classifica()})
```