

İsim: Murat

Soyisim: Saak

Proje İsmi: FootballManagerApp

PROJE ÖZETİ

Bir futbol ligi sistemi oluşturulmaya alışıldı. Lig içindeki oyuncular,takımlar ve bu takımların birbirleriyle oynadığı maçlardan oluşan fikstür oluşturuldu. Oynanan maçlar sonucunda oluşan puan durumu, takım istatistikleri ve maç istatistikleri de kayıt altına alındı.

PACKAGE YAPILARI

1.entity

Toplamda 19 adet entity tanımlandı. Her bir sınıf ve field ilişkilerini tanımlamak için Hibernate anotasyonları kullanıldı. Constructor,getter,setter ve toString metodları otomatik oluşturulması için Lombok bağımlılığı kullanıldı.

1.1.BaseEntity: Diğer bütün entitylerin tek bir ana sınıftan miras alması için oluşturuldu. State, createAt ve updateAt gibi tüm sınıflarda bulunacak ortak fieldlar burada tanımlandı. Bu sınıf ana sınıf olarak oluşturulduğu için @MappedSuperclass anotasyonu ile işaretlendi. Bu sınıftaki 3 adet field alt sınıflarda nesne oluşturulurken otomatik olarak atanması için default değerler atanıp @Builder.Default lombok anotasyonu kullanıldı.

1.2.Person: Ara sınıf olarak düşünüldü. Manager ve Player sınıfları bu sınıftan miras alınarak oluşturuldu.

Ortak alanlar şu şekilde;

String name: İsim ve soyisim tutar

Integer age: Yaş tutar

1.3.Season: Bu sınıf sezonluk takvimi oluşturulmak için tanımlandı. Entitylerin bu sınıfla olan ilişkileri diğer entitylerde tanımlandı. Sınıf içinde tanımlanan fieldlar;

Long id: Nesnelere has id tutar.

String title: Sezon başlığının tanımlandığı alan.

LocalDate BEGINNING OF SEASON: Sezon başlangı tarihini belirler. Bu alan final olarak tanımlanıp kullanıcı tarafından bir daha değıştirilmemesi sağlanabilirdi.

LocalDate currentDate: Program başladıktan sonra kullanıcı maçları manuel ya da otomatik olarak oynattığında bu alan otomatik olarak maç tarihinin oynandığı güne eşit olacak şekilde değıştirilir. Ayrıca menüde şuan ki Tarih takibi yapılabilmesi için sürekli gösterilir.

1.4.League: Ligi tanımlamak için oluşturuldu.

Long id: Nesnelere has id tutar.

String leagueName: Ligin ismini tanımlar.

ERegion region: Ligin bulunduğu bölge Enum değeri olarak tanımlanır.

Edivision divison: Bir lige ait alt lig üst lig kavramları olabileceği için, o ligin o bölgedeki hangi lig olarak oluşturulduğunun kolaylıkla anlaşılması için Enum değerler olarak oluşturuldu.

Season season: Lig içinde manyToOne anotasyonu ile tanımlandı. Bundaki sebep bir lige ait birden fazla sezonun olabileceği düşüncesi idi. (2024-2025,2025-2026 sezonu vs. gibi)

Fixture fixture: OneToOne anotasyonu ile tanımlandı. Bundaki sebep her sezona ait her bir ligin kendine ait tek bir fikstürü olması düşüncesi idi.

1.5.Stadium: Takımlara ait stadyumları tanımlamak için oluşturuldu. Takımlarla olan ilişkiler Team entitisinde tanımlandı.

Long id: Bu sınıftan oluşturulan nesnelere has id oluşturur. (Buradan sonra artık id fieldları tanıtılmayacak)

String name: Stadyum ismi

String location: Stadyumun bulunduğu şehir

Integer capacity: Stadyum taraftar kapasitesi

1.6.Team: Takımları tanımlamak için oluşturuldu.

String teamName: Takımın ismi

Long budget: Takım bütçesi burada tutulur. Takımlar transfer yaptıkça ya da var olan futbolcularını sattıkça bu field güncellenir.

Stadium stadium: Takımlara ait bir adet stadyum olacağı düşünüldüğü için OneToOne olarak tanımlandı.

League league: Bir takım birden fazla ligde bulunacağı(Süperlig,Şampiyonlar Ligi,Uefa Ligi vs.) düşünüldüğü için ManyToOne olarak tanımlandı.

List<Player> players: Takımlara ait kadrolar ekstra sorguya gerek kalmadan getirilebilmesi için o oyuncular Team entitisinde liste olarak tutuldu. OneToMany anotasyonu kullanıldı.

1.7.Player: Oyuncuya ait özellikler tanımlandı.

Eposition position: Oyuncunun pozisyonu (kaleci,defans vb.) Enum olarak tanımlandı.

Integer skillLevel: Oyuna ait yetenek seviyesi 1-100 arasında tanımlandı. Oyuncunun yetenek seviyesine bağlı olarak maç skorlarına etki etmesi planlandı.

Long value: Futbolcunun piyasadaki bonservis değeri.

Long salary: Futbolcunun güncel takımından aldığı maaş.

Team team: Futbolcunun takımı manyToOne olarak tanımlandı.

1.8.Manager: Takım yöneticilerini tanımlamak için açıldı bu entity.

String username: Kullanıcı girişinde kullanıcı adı olarak tanımlandı. Bu alanın eşsiz olarak yaratıldığından emin olmak için @Column(nullable = false) anotasyonu ile işaretlendi.

String password: Kullanıcı giriş şifresi.

Integer experience: 1-10 arasında menajer yetenek ve deneyimini tanımlamak için düşünüldü.

Team team: Menajerin takımı @OneToOne anotasyonu ile işaretlendi.

1.9.Fixture: Lige ait maçları tanımlamak için kullanılan sınıf.

String description: Fikstür açıklaması.

League league: Her lige ait bir fikstür olacağı düşüncesiyle OneToOne anotasyonu ile tanımlandı.

1.10.Match: Oynanan maçların kaydını tutmak için oluşturulan entity.

Long homeTeamId: Ev sahibi takımın Team id'si. Burda kolaylık olsun diye id yerine direk Team tutulabilirdi.

Long awayTeamId: Deplasman takımı idsi

int homeTeamScore: Ev sahibi maç skoru.

int awayTeamScore: Deplasman takımı maç skoru.

LocalDate matchDate: Maçın oynandığı tarih. Maç oynandıktan currentDate bu tarihe eşitlenip güncel tarihin takibi sağlanır.

boolean isPlayed: Maç oynanıp oynanmadığını kontrol etmek için oluşturuldu. Burada boolean primitive olarak tanıtıldı, bunun sebebi default olarak isPlayed false değerinin atanmak istenmesi. Maçlar oynandıktan sonra isPlayed = True olarak set ediliyor.

Fixture fixture: @ManyToMany anotasyonu ile tanımlandı. Amaç bu fikstüre ait maçların belirlenip listelenebilmesi.

1.11.Stats: Takımlara ait oynanan maç sayısı, atılan gol, yenilen gol, averaj, toplam puan vb. gibi istatistiklerin tutulmasına yardımcı olan entity. Her maç oynandıktan sonra otomatik olarak güncellenicek şekilde ayarlandı. Hangi takıma ait istatistik olduğunun belirli olması için @OneToOne anotasyonu ile Team team olarak ayrı bir field tanımlandı.

1.12.MatchStats: Oynanan maça ait pas yüzdeleri, toplam şut sayısı, toplam pas sayısı vb. istatistiklerin tutulduğu sınıftır. Team team @OneToOne anotasyonu ile tanımlandı. Maç fikstürü oluşturulurken maçlarla birlikte MatchStats nesneleri de oluşturuldu. Her maç oynandıktan sonra o maça ait istatistik otomatik olarak update edilir.

1.13.TransferOffer: Menajer ile giriş yapan kullanıcının başka takımdan transfer oluşturmak isterse toplam 3 aşamadan oluşan transfer sürecinin ilk aşaması için oluşturulan entity. Burada tanımlanan fieldlar sadece bir takımdan diğerine sunulacak bonservis teklifi ve teklif alan takımın cevabını içerir.

String message: Transfer etmek isteyen takımın karşı takıma mesajı.

Long biddingMoney: Bonservis için teklif edilen ücret.

OfferResponse: Transfer teklifine verilen cevap. Enum olarak tanımlanmıştır. (ACCEPTED, REJECTED, ON_WAIT)

Team proposer: Transfer teklifi yapan klüp.

Team receiver: Teklif alan klüp.

Player player: Transferde adı geçen söz konusu oyuncu. Bu ve yukarıdaki 2 fieldda @ManyToOne anotasyonu ile tanımlandı.

1.14.ContractOffer: Eğer bir klübün diğer takıma yapmış olduğu transfer teklifi kabul edilirse, oyuncu ile sözleşme görüşmeleri yapılabilmesi için bu entitiy oluşturuldu.

Long offeredSalary: Oyuncuya teklif edilen maaş.

EOfferResponse response: Oyuncunun teklif edilen maaşa cevabı. ACCEPTED ya da REJECTED enum olarak tanımlanabilir. Programda oyuncunun cevabı teklif edilen maaşa bağlı olarak otomatik olarak cevaplanıyor, fieldin yeni değeri de bu cevaba bağlı olarak set ediliyor.

TransferOffer: Kabul edilen bonservis teklifindeki fieldlara ulaşabilmek için bu entity içinde tutuldu.

1.15.Transfer: Bir takım hem oyuncu hem de onun takımıyla transfer konusunda anlaştıysa, artık bu transferin tamamlanıp kayıt altına alınması için oluşturuldu.

İçinde id'ye ek olarak sadece ContractOffer tanımlıdır. Bunun sebebi de transfere ait tüm alanlara rahatça ulaşılacak istenmesi. (Alıcı,satıcı,oyuncu,maaş,bonservis bilgileri vb.).

Not: Bu entitylere ek olarak oynanmayan maçlar üzerine bahis oynatabilmek için 4 adet entity daha tanımlanmıştır. Fakat hala yapım aşamasında olduğu için bu dökümanda yer verilmeyecektir.

2.Repository

Bu package'ta Database ile ilgili işlemleri yazılacak sorgular yazıldı. Oluşturulan her bir entity için Repository sınıfı oluşturuldu. 5 Temel Crud işlemi her Repository'de olunması istendiğinden ve tekrar tekrar yazmaktan kaçınmak için RepositoryManager adlı ana bir sınıf oluşturulup CRUD metodları burada tanımlandı. Diğer repository sınıfları bu sınıftan miras aldı, eğer o sınıfın repositorisine ait özel bir sorgu yazılması gerekirse bu sorgu sınıfa ait repository'de yazıldı. Repository'lere erişim sadece o sınıfın Service sınıfından sağlanacak şekilde yapı oluşturuldu.

3.Service

Repository sınıfı gibi burda da ServiceManager ana sınıfı oluşturularak diğer Service sınıflarının bu sınıftan miras alınması sağlandı. Service sınıfında yapılması istenen işler repository üzerinden databaseden çekilip (ya da kaydedilip) yapılması sağlandı. Servicelerde ayrıca if else gibi kontroller de yapıldı.

4.Controller

Servicelerle iletişime geçen sınıflar burada oluşturuldu. Fakat ekstra olarak içeriklerine bu aşama bir şey eklenemedi. Şuanki fonksiyonu sadece çağırdığı service metodunu guide çalıştırmak olarak kaldı.

5.Gui

Main,League, Manager ve Match menüleri burada oluşturuldu.

Main menüsünde kullanıcının giriş yapıp yapmadığına bağlı olarak 2 farklı menü çıkıyor. Eğer menajer olarak giriş yaparsa Manajer Menüsüne ulaşımı da buradan sağlanıyor. Aynı şekilde maç menüsü için de Manager girişi yapılması zorunlu hale getirildi.

Lig menüsünde lige ait puan durumu, maç sonuçları, genel fikstür gibi seçenekler görüntülenebiliyor.

Manager menüsünde manajerin takımına ait bilgiler, transfer tekliflerini görüntüleme veya başka bir takıma transfer teklifi yapma, Manajerin kendi profilini görüntülemesi gibi seçenekler eklendi.

Maç menüsünde lig menüsüne benzer olarak kolaylık açısından puan durumu, maç fikstürü gibi seçenekler yine eklendi. Bu menüye özel olarak maç oynatma veyahut belirlenen ileri bir tarihe kadar olan bütün maçları otomatik oynatma seçeneği eklendi. Bu maçlar oynandıktan sonra kullanıcının geçmiş maç sonuçlarını ve güncel puan durumunu görüntüleyebilmesi sağlandı.

6.Utility

Burada yardımcı olabilecek package'ları aşağıda listeleyip açıklayacağım.

Data

DemoData ve FixtureGenerator olarak iki ayrı sınıf oluşturuldu. Bu iki sınıfın da amacı program sıfırdan başlatılmadan önce veritabanında Player,Team,League ve bunlara ait fikstür,maçlar,istatistikler gibi verilerin otomatik oluşturulmak istenmesi. Program başlamadan önce eğer bu veriler sıfırdan yüklenilmek istenirse Main'den metod olarak çağırabilecek şekilde düzenlendi.

Engine

Maç simülasyonları için oluşturuldu. Takımların oyuncularını baz alınarak hesaplanan güçlerine göre skorlara değer atanması burada oluşturulan MatchEngine yardımıyla yapıldı. Aynı sınıf default ve NoComment şeklinde oluşturuldu. Bundaki sebep kullanıcı tek bir maç oynatmak istediğinde maç yorumlarıyla birlikte maçı canlı izleme havası yaratmak veya birden fazla maç yapmak istendiğinde tüm maçların yine aynı şekilde simüle edilip fakat maç yorumlarının bulunmadığı bir seçenek oluşturulmak istenmesi idi. Maç simülasyonunu sağlayan algoritma ileride daha da geliştirilip gerçeğe daha yakın olması sağlanabilir.

Enums

Bu package'ta entity fieldlarında tanımlanan Enum sınıfları oluşturulmuştur. İçerikleri entity bölümünde ayrıyetten açıklanmıştır.

ConsoleTextilUtils (Class)

Bu sınıf menü yaratmak, çıktı vermek, kullanıcıdan input almak gibi işlemleri kolaylaştırmak için oluşturulan yardımcı bir sınıftır. İlk defa bu projede kullanılıp proje boyunca çokca kez kullanılıp faydalı olduğu tespit edilmiştir.

HibernateConnection (Class)

Hibernate üzerinden veritabanına bağlantıyı kolayca sağlamak, açma kapama işlerini otomatik hale getirmek için oluşturuldu. Global erişim sağlanıp buradaki fieldlar ve metodlar statik olarak istenilen yerlerde çağırıldı.

KAPANIŞ

Özet olarak baştan sona uğraştırıcı, zevk veren ve öğreten bir proje oldu kendi açımdan. Katmanlı mimari yapısını ilk defa amacına uygun olarak uygulamaya çalıştığımız bu projede grubum ile önümüze çıkan sayısız sorunları çözmeye çalışırken bir çok yararlı yeni bilgi öğrendim. Çok ileride dahi bu projede edindiğim bilgi ve tecrübelerin işime mutlaka yarayacağını düşünmekteyim.