



**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

**Mühendislik-Mimarlık Fakültesi**

**Bilgisayar Mühendisliği**

**VERİ TABANI YÖNETİM SİSTEMLERİ B**

**2023-2024 Güz Dönemi**

**OYUN YÖNETİM SİSTEMİ**

**GRUP NO: 3**

*Buse ERCAN	152120201064
Yakub ALKIŞ	152120201082
İrem KAVAKPINAR	152120201069
Okan SEVİNÇ	152120201100

**Firma Yetkilisi**



Serhat ERSEL 0544 251 07 43 [info@avkaryazilim.com.tr](mailto:info@avkaryazilim.com.tr)

Aralık 2023

## İçindekiler

İçindekiler .....	2
1. Giriş.....	5
1.1 Amaç .....	5
1.2. Projeyi Seçme Amacımız ve Sebebimiz .....	5
1.3. Projede Kullanılacak Teknolojiler .....	5
1.4. Projenin Hedefleri ve Başarı Kriterleri .....	5
1.5. Benzer Ürünler .....	5
1.5.1. Benzerlikler .....	5
1.5.2. Farklılıklar .....	5
1.6. Fizibilite Çalışmaları .....	5
1.6.1. Teknik Fizibilite .....	5
1.6.2. Ekonomik Fizibilite.....	6
1.6.3. Operasyonel Fizibilite .....	6
1.7. Kullanılacak Teknolojiler .....	6
1.7.1 Java Spring Boot .....	6
1.7.2 React JS.....	6
1.7.3 Postgre SQL.....	6
1.8 Yapılan Çalışmalar.....	6
1.9. Geline Nokta.....	6
2. Sistem Gereksinimleri.....	7
2.1. Fonksiyonel İsterler .....	7
2.2. Fonksiyonel Olmayan İsterler .....	7
2.2. Yazılımsal İsterler .....	7
2.3. Arayüz İsterleri .....	7
2.4. Veri İsterleri .....	8
2.4.1. Kullanıcı İsterleri .....	8
2.4.2. Satın Alım İsterleri .....	8
2.4.3. Seviye/ Rütbe İsterleri .....	8
2.4.4 Oyun İçi Eşya (İtem) İsterleri .....	9
2.4.5. Çanta İsterleri .....	9
2.5. Genel Olarak Neler Eklendi.....	9
3. Modül Açıklamaları.....	9
3.1 Oyun Kullanıcısı .....	9
3.2 Oyun İçi Eşyalar .....	9
3.3 Oyun Çantası.....	9

3.4 Loglama .....	10
3.5 Para Yükleme .....	10
3.6 Hesap Yönetimi.....	10
4. Vtys Tasarımı İçin Olan Gereksinimler.....	10
4.1. Oyuncu.....	10
4.2. Rütbe .....	11
4.3. Seviye.....	11
4.4. Cüzdan .....	12
4.5. Kayıt (Loglama) .....	12
4.6. Satın_Alım.....	12
4.7. Eşya.....	13
4.8. Canta.....	13
4.9. Kategori .....	13
4.10. Özellikler .....	13
4.11. Fiyat_Tarihi .....	14
5. Sistem Modelleri.....	14
5.1. Aktörler.....	14
5.2. Use-Case Diyagramı .....	15
5.3. Uygulama Senaryosu .....	16
6.Sorgular (Karmaşık Sorgu) .....	17
7.Vtys Araç Kullanımı.....	18
7.1. Tetikleyici (Trigger) .....	18
7.2. Error Handling .....	18
8. Proje Kısıtları.....	19
8.1. Kaynak kısıtlaması .....	19
8.2. Bütçe Kısıtlaması.....	19
8.3. Risk Kısıtlaması .....	19
8.4. Düşünülmeyen Kısıtlar.....	19
9. Sunuma Sonradan Eklenenler .....	19
11. Kullanıcı Arayüzü Görüntüleri.....	19
12. Diğer Entegrasyonlar .....	21
12.1. Sosyal medya ile login .....	21
12.2. E-mail/GSM msg entegrasyonu .....	21
12.3. XLS Import .....	21
13. Toplantılar .....	21
13. Proje ekibi değerlendirmesi.....	22

13.1. Grup bilgisi.....	22
13.2. Toplam Çalışma süreleri .....	23

## 1.Giriş

### 1.1 Amaç

Amacımız şirket için kullanıcıların bir oyunda satın alımlarını, oyun etkinlik bilgilerini, başarımlar, oyun içi satın alınabilir ve harcanabilir nesneleri veri tabanı ile ilişki kurarak ve bunları sistematik şekilde ilişkilendirip kaydeden bir backend servisi tasarlamak. Aynı zamanda oyundaki satın alımları loglayıp (kayıtlayıp) şirket tarafında analiz ve istatistik için kullanılacak yapıyı sağlamak. Kısacası bir oyunun bütün backend servisini yazacağız.

### 1.2. Projeyi Seçme Amacımız ve Sebebimiz

Bu projeyi seçme amacımız gerekli teknolojileri de kullanarak hem grup olarak deneyim kazanmak hem de veri tabanı yönetim sistemi olarak firmaya katkı sağlamak. Seçme sebebimiz ise ekip üyelerimizin bildiği teknolojileri kullanma yönünde en optimum proje bu olduğu için bunu seçtik.

### 1.3. Projede Kullanılacak Teknolojiler

Verileri tutmak için PostgreSQL, backend servisini yazmak için Spring Boot kullanacağız. Kullanıcıların aksiyonlarını sağlıklı şekilde test edebileceğimiz bir ara yüz yazacağız. Bu ara yüz için CSS ve React kullanacağız.

### 1.4. Projenin Hedefleri ve Başarı Kriterleri

Kullanıcının oyun içindeki transactional (veri tabanında karşılık bulan işlem) aksiyonlarının başarılı bir şekilde yapılabilip ve bunun hem veri tabanında yine başarılı bir şekilde implemente (işlenmesi / uygulanması) edilmesi hem de servise gelecek HTTP isteklerinin başarılı bir şekilde handle (kontrol) edilmesi hedefimiz ve başarı kriterimizdir.

### 1.5. Benzer Ürünler

Araştırdığımız World of Warcraft, Farm Ville, ThiefMaster gibi oyunlarda ilişkili veri tabanı kullanımı, kullanıcı hesapları ve profil yönetimi, oyun içi eşyalar vb. gibi yapıların ortak kullanıldığını fark ettik.

#### 1.5.1. Benzerlikler

- İlişkili veri tabanı kullanımı (PostgreSQL)
- Kullanıcı hesapları (Kullanıcıların eşsiz bir kullanıcı adı ile hesap oluşturması)
- Oyun içi eşyalar (Kullanıcının oyun içi eşya alımı yapması)
- Sisteme kayıt ve giriş yönlendirmeleri olması
- Seviye ve rank sisteminin olması

#### 1.5.2. Farklılıklar

- Şirketimizin isteri doğrultusunda profesyonel bir ödeme sistemi olmaması
- Yine ister doğrultusunda profil yönetimi olmaması

### 1.6. Fizibilite Çalışmaları

#### 1.6.1. Teknik Fizibilite

Projemiz client-server tabanlı uygulamaya dayalı web tabanlı bir uygulamadır. Client tarafı aslında bir proje isteri değildir sadece server tarafındaki REST API'deki fonksiyonel işlemlerin doğru çalışıp çalışmadığını test etmek için tasarlanan bir arayüzden ibaret olacaktır. Ana amaç bir oyun servisi yani bir server yazmaktır.

### 1.6.2. Ekonomik Fizibilite

Tahmini maliyetler, personel, donanım, yazılım, pazarlama ve operasyonel giderleri içerir. Personel maliyetleri, proje yönetimi, donanım-yazılım gereksinimleri ve tanıtım için harcanacak bütçe göz önüne alındığında, toplam tahmini maliyet belirlenmiştir. Kar marjı ve beklenen geri dönüş süresi analiziyle, projenin ekonomik başarısı vurgulanmıştır. Ayrıca, risk analizi yapılarak olası risk faktörleri göz önüne alınmış ve düzenli finansal izleme önerilmiştir. Bu ekonomik fizibilite analizi, proje sahiplerine bilinçli kararlar almalarına yardımcı olacak bir çerçeve sunmaktadır.

### 1.6.3. Operasyonel Fizibilite

Operasyonel Fizibilite Analizi, Oyun Servisi Yazma Projesi'nin günlük işleyişini değerlendirir. Projenin operasyonel başarısını sağlamak adına, yazılım geliştirme ekibiyle işbirliği içinde süreçlerin düzenli bir şekilde yönetilmesi önemlidir. Geliştirme ortamının uygunluğu, yazılımın düzenli güncellenmesi operasyonel başarının temel unsurlarıdır. Ayrıca, kullanıcı verilerinin güvenliği ve teknik sorunlara hızlı müdahale kapasitesi, projenin günlük operasyonlarını etkileyen kritik faktörler arasındadır. Bu bağlamda, operasyonel süreçlerin düzenli gözden geçirilmesi ve iyileştirilmesi, projenin uzun vadeli sürdürülebilirliği için önemli bir strateji olacaktır.

## 1.7. Kullanılacak Teknolojiler

### 1.7.1 Java Spring Boot

Java dilinde güçlü MicroService yazılmasına olanak tanıyan bir framework kullanılacaktır. Springboot sayesinde backend uygulama geliştirirken veri tabanı işlemlerimizi Java ile Language Integrated olarak yazılabilmesine olanak sağlar. Aynı zamanda client tarafındaki uygulamalar içinse servis uçları sağlayarak veri girdi-çıkı oluşturulabilir.

### 1.7.2 React JS

Kullanıcı ara yüzleri oluşturmak için açık, verimli ve esnek bir JavaScript kütüphanesidir.

### 1.7.3 Postgre SQL

PostgreSQL, özgür ve açık kaynak kodlu, SQL destekli bir ilişkisel veri tabanı yönetim sistemidir.

## 1.8 Yapılan Çalışmalar

- Backend tarafında ER tablomuzaya uygun olarak gerekli varlıklar (entities) oluşturulup aralarındaki ilişkiler kuruldu.
- Backend tarafında veri alışverişi için gerekli servisler için operasyonlar (controller sınıfları) yazıldı.
- Backend tarafında bir varlığın bütün verisini göndermek yerine veri transfer objeleri (dto) oluşturduk. (Örn: PurchaseDto.java)
- Client tarafında belirlediğimiz normal kullanıcı ve admin kullanıcısının aksiyonlarını oluşturduk. Temel düzeyde tasarım sağladık.

## 1.9. Geline Nkta

- Belirlediğimiz normal kullanıcı ve admin kullanıcısına tanımladığımız aksiyonları sağlıklı bir şekilde tamamladık ve sağlıklı çalıştığını test ettik.
- Client tarafından server tarafımıza istekleri şekilde geldiğini doğruladık.
- Oyun içi tanımladığımız aksiyonları sağlıklı bir şekilde çalıştığını doğruladık.

## 2. Sistem Gereksinimleri

### 2.1. Fonksiyonel İsterler

- Kullanıcı oyun ara yüzüne sisteme kayıt olduğu şifre ve kullanıcı adı ile giriş yapabilecek.
- Kullanıcı; kullanıcı adı, email, telefon numarası ve şifre ile kayıt olabilecek.
- Kullanıcının parolası hash ( şifrelenerek ) işlemi yapılarak veri tabanına kaydedilecek.
- Kullanıcı sisteme giriş yaptığında veya üye olduğunda mail / sms bildirimi alacak. ( Ek ister )
- Yönetici tüm kullanıcıların bilgilerini görüntüleyebilecek.
- Kullanıcı oyun içi eşyalar satın alabilip ve bu eşyaların kullanıcının çantasına ekleme işlemleri yapılacaktır.
- Kullanıcının para birimi oyun içerisinde geçerli olacak. Oynadığı oyundan ödül olarak para kazanabilecek ve bu işlem bizim sorumluluğumuzda olup bu kazandığı parayı kullanıcının mevcut bakiyesine ekleyip kayıt edeceğiz.
- Kullanıcı Github ve Google hesapları ile giriş yapabilecek. ( Ek ister )

### 2.2. Fonksiyonel Olmayan İsterler

- Gizlilik  
Kullanıcı parolaları şifrelenerek veri tabanına kayıt edilecek ve decryption algoritması uygulaması ulaşılmaz olacak.  
Kullanıcı bilgilerine yalnızca kendisine ait şifre ile ulaşabilecektir.  
Farklı kullanıcılar birbirlerinin bilgilerini göremeyecektir.  
Admin kullanıcılara ait tüm bilgileri görebilecektir.
- Güvenilirlik  
Bir kullanıcı başka bir kullanıcı bilgilerini göremeyecek.
- Desteklenebilirlik  
Sistem ileride isterlerin değişimine göre modüler olarak programlanacak olup, yeni fonksiyon ekleme veya eskisini güncelleme gibi durumlarda kolaylık sağlayacaktır.  
Geliştirmeler sayesinde sistem geleceğe dönük olacak, zaman içerisinde yeni modüller eklenebilecek ve güncelleştirilebilir olacaktır.

### 2.2. Yazılımsal İsterler

- Sistem web uygulaması olarak çalışacaktır.
- Modüler olarak programlanması amaçlanarak ileride yapılan değişiklikler rahatlıkla yapılabilir.
- Veri tabanı ile entegre şekilde çalışacak bir sistem oluşturulacaktır.
- Firmanın isterlerinin bir kısmına cevap verebilecek olup, ileride firmanın isterlerini tamamı ile karşılayabilecek bir sistem amaçlanmaktadır.

### 2.3. Arayüz İsterleri

- Biz ve şirket olarak yazdığımız sistemin sağlıklı olarak çalışıp çalışmadığını test edebileceğimiz bir arayüz sağlayacağız.

- Adminlerin yetkileri doğrultusunda, tüm kullanıcıların bilgilerini görüntüleyebileceği bir arayüz sağlanacaktır.
- Custom Login (Giriş) ve Registration (Kayıt) sayfası oluşturulacak. Bu sayfalarda kullanıcı formları doldurarak giriş yapabilir veya kayıt olabilecek.

## 2.4. Veri İsterleri

Veriler PostgreSQL veri tabanında tutulacaktır.

### 2.4.1. Kullanıcı İsterleri

- Kullanıcı sisteme giriş yaptığında veya kayıt olduğunda kullanıcıların logları (kayıtları) tutulacaktır. Loglanma (kayıt) türü “auth” olacaktır.
- Kullanıcının parolası veri tabanında hash edilmiş (şifrelenmiş) şekilde tutulacaktır.
- Admin bir kullanıcının hesabını silebilmelidir.
- Kullanıcının e-mail adresi ve kullanıcı adı eşsiz olmalıdır. Kayıt işleminde bunun kontrolü sağlanmalıdır.
- Kullanıcının telefon numarası “+9005xxxxxxx” formatında tutulacaktır.

### 2.4.2. Satın Alım İsterleri

- Kullanıcı oyun içerisinde eşya satın alımlarının logları (kayıtları) tutulacak.(Kayıt tablosu)
- Loglanma (kayıt) türü “satın\_alım” olacaktır. (Kayıt tablosundaki tür özelliği)
- Eşyaların farklı tarihlerde farklı fiyatlara sahip olacak. Yani veri tabanında bir eşya ile ilgili farklı fiyat ve tarih bilgileri (Fiyat\_Tarihi tablosu) olacaktır.
- Kullanıcının bir cüzdanı olacaktır ve bu cüzdan veri tabanında Cüzdan ismindeki tablo ile tutulacaktır. Cüzdandaki oyun parası hem oyundan kazanılan para ödülleri ile hem de kullanıcın kredi kartı ile marketten oyun parası satın alması ile artabilecektir. Aynı zamanda, eşya satın alım sonrası azalacaktır.
- Eşyanın fiyat tarihi bilgilerinde bulunan fiyat birimi oyun içi para ile alınabilip ya da TL ile mi alınabileceğini belirler. Örneğin; eşyanın fiyat birimi oyun içi para ise kullanıcının cüzdanındaki oyun içi para bakiyesi yeterli olma durumunda alınabilecek, yeterli olmadığı durumunda da kullanıcı oyun içi para satın alımına yönlendirilecektir.
- Markette oyun içi para satın alınabilecek ve bu da eşya olarak değerlendirildiğinden dolayı fiyat birimini TL olarak tutacağız. Bu kısım kullanıcıyı ödeme sistemine yönlendirir ve bu bizim sorumluluğumuzda olmayacak. ( Şirket geliştiricileri bu kısmı ele alacak biz bu kısım için sadece demo aksiyon oluşturacağız. )

### 2.4.3. Seviye/ Rütbe İsterleri

- Kullanıcı oyun oynadıkça oyunun ön yüzünden gelen isteğe göre seviye atlayacaktır. Kullanıcı oyunda kazanıp kaybettiğinde oyunun ön yüzünden gelen isteğe göre rütbe değiştirilecektir. Burada bizim sorumluluğumuz sadece rütbe ve seviyenin değiştirilip veri tabanına kaydedilmesi olacaktır. (Kullanıcının performans işlemleri oyunun ön yüz geliştiricilerinin sorumluluğundadır.)
- Hem seviye için hem de rütbe için kullanıcı oyun oynadığında ve bunun kaydedilmesi gerektiği için client tarafından istek atılarak rütbenin ve seviyenin değiştirilmesi sağlanacak. Kısacası, kullanıcı ne kadar oyun kazanırsa rütbesi o şiddette artacak veya ne kadar oyun kaybederse rütbesi o şiddette azalacak.
- Seviye ve rütbe değerlerinin bir minimum puan değeri olduğu için oyuncunun mevcut puanını bu değer ile kıyaslayarak seviye veya rütbe değiştireceğiz.



#### 2.4.4 Oyun İçi Eşya (İtem) İsterleri

- Servisimizde birden çok kategori olacaktır ve bu kategoriler altında birbiri ile uyumlu eşyalar tutulacaktır.
- Her eşyanın özellikleri olacaktır ve yine bu da özellikler altında birbiriyle uyumlu eşyalar arasında tutulacaktır.
- Her özelliğin spesifik olarak bir güç miktarı olacaktır. Örneğin vuruş gücünün büyüklüğü veya hız büyüklüğü olabilir amaç sadece mevcut olan yeteneğin miktarını tutmaktır.
- Her eşya, kategori ve özelliğin bir sembolü olacaktır. Servis tarafında sadece bu sembollerin dosya isimlerini tutacağız. Client tarafı bu dosya isimlerini kendi proje dizinindeki resim dosyasıyla uyuşan resmi yükleyip arayüzde görüntüleyecek.

#### 2.4.5. Çanta İsterleri

- Çantadaki eşyalar client tarafından gelen isteğe göre veri tabanından silinebilecektir.
- Her kullanıcının kendisine ait bir çantası bulunacaktır. Bu çantada satın aldığı veya kazandığı oyun içi eşyalar (item) mevcut olacaktır.
- Kullanıcı isterse çantasından eşya silebilecektir.

### 2.5. Genel Olarak Neler Eklendi

- Başta satın alım işlemi için amacından sapan bir karışık bir yapısı vardı. Biz sadece bir tabloya indirgeyip basitleştirdik. Budget\_preview ve Purchsed\_History tablosunu kaldırıp Cüzdan ve Satın\_alım altında topladık.
- Rütbe ve seviye ekledik.
- Kullanıcı özelliklerinde değişiklikler yaptık.
- Kayıt (Log) tablosu ekledik.
- Özelliklere güç miktarı eklendi.
- Oyuncu tablosu üzerinde genel olarak özellik değişiklikleri yapıldı.
- İstemci (Client) uygulaması tasarlandı.
- Oyun servisi (Server) oluşturuldu.

## 3.Modül Açıklamaları

### 3.1 Oyun Kullanıcısı

Oyun kullanıcısı modülünde oyuncunun kişisel bilgileri, oyundaki kullanıcı ismi, ona ait olan oyun cüzdanı ve eşyaları gibi ayrıntıları içermesi gerekir. Oyuncunun oyundaki rütbe ve seviyesi gibi verilerin de tutulması gerekir.

### 3.2 Oyun İçi Eşyalar

Her bir eşyanın ait olduğu bir kategori, kendisine ait özellikleri, simgesi ve kendisine ait değerleri bulunmalıdır. Aynı zamanda şirket isterleri doğrultusunda bu eşyaların fiyatları tarihlere göre değişebilir bir şekilde tutulmalıdır. İleri tarihler içinde fiyatlandırma yapılabilmelidir.

### 3.3 Oyun Çantası

Kullanıcının sahip olduğu oyun içi eşyalarını görüntüleyebilmeli ve düzenleyebilmeli.

### 3.4 Loglama

Şirketin istatistik amaçlı veri analizi için kullanıcının oyun içinde para harcama ve yükleme, oyun içi eşya satın almak gibi bütün transactional işlemlerin kaydını tutan bir loglama sistemi bulunmalıdır. Her işlemin bir gerçekleşme zamanı ve işlem ile ilgili bilgiler bulunmalıdır.

### 3.5 Para Yükleme

Şirketin isterleri doğrultusunda kullanıcı oyun cüzdanına para yükleyebilmelidir. Bu paranın oyunda önceden belirli tarihlerde belli bir birimle altın veya oyun içi eşya alabilmelidir. Aynı zamanda yapılan satın alımlarının tarihleri, ne kadar para yüklendiği, ne alındığı gibi detayları içermelidir.

### 3.6 Hesap Yönetimi

Kullanıcı oyundaki username ve şifresi ile hazırladığımız client uygulamamıza giriş yapabilmeli ve oyun içi bilgilerini görüntüleyebilmelidir.

## 4. Vtys Tasarımı İçin Olan Gereksinimler

### 4.1. Oyuncu

Bu tabloda oyun servisini kullanacak olan oyuncuların bilgileri tutulacaktır. Oyuncu tablosunun diğer tablolarla ilişkisi şu şekildedir:

Cüzdan tablosu ile 1-1 ilişkisi bulunmaktadır.

Kayıt tablosu ile 1-N ilişkisi bulunmaktadır.

Satın-Alım tablosu ile 1-N ilişkisi bulunmaktadır.

Çanta tablosu ile 1-1 ilişkisi bulunmaktadır.

Rütbe tablosu ile N-1 ilişkisi bulunmaktadır.

Seviye tablosu ile N-1 ilişkisi bulunmaktadır.

Oyuncu tablosunun diğer tablolarla ilişkisi tabloların açıklamalarında belirtilmiştir.

- 1) Oyun servisimize kayıt olan her kullanıcıyı diğer kullanıcılardan ayıran bir değere ihtiyaç vardır bu değer bigint(19) tipinde **ID** adında primary key şeklinde tutulacaktır.
- 2) Oyun servisimize bizim kayıt formumuz üzerinden kayıt olan her kullanıcının farklı bir kullanıcı adı bulunması gerekmektedir. Bundan dolayı kullanıcı adları varchar(25) tipinde **kullaniciAdi** adında Unique(benzersiz) olarak tutulacaktır.
- 3) Oyun servisimize bizim kayıt formumuz üzerinden kayıt olan her kullanıcının bir şifresi bulunması gerekmektedir. Bundan dolayı şifreler varchar(255) (Hash edilerek tuttuğumuz için 225) tipinde **sifre** adında tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).
- 4) Oyun servisimize kayıt olan her kullanıcının e-mail ile bilgilendirme alabilmesi için farklı bir e-mail adresi bulunması gerekmektedir. Bundan dolayı e-mail adresleri varchar(25) tipinde **eposta** adında Unique (benzersiz) olarak tutulacaktır.
- 5) Oyun Servisimize kayıt olan her kullanıcının telefon numarası ile bilgilendirme alabilmesi için bir telefon numarası bulunması gerekmektedir. Bundan dolayı telefon numaraları varchar(25) tipinde **telefonNo** adında tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).
- 6) Oyun servisimizde kullanıcımızın nereden kayıt olduğu kayıt edilmelidir. (Local-bizim servisimiz-, Github, Google) Bundan dolayı bu değer varchar(15) tipinde **saglayici** adında tutulacaktır.
- 7) Şirket isteri doğrultusunda her kayıt olan oyuncunun ne zaman kayıt olduğunun saklanması istenmektedir. Bundan dolayı oyuncu kayıt olduğunda kayıt olduğu tarihi tutması için

varchar(15) tipinde **kayitTarihi** adında tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).

- 8) Oyun servisimize kayıt olmuş kullanıcılarımızın sembolü olabilmektedir. Bu semboller client üzerinden gelmektedir fakat bizim de bu sembollere erişebilmek için adlarını tutmamız gerekmektedir. Bundan dolayı bu sembollerin isimlerine ulaşmak için varchar(25) tipinde **sembol** adında tutulacaktır. (Eğer kullanıcı profil fotoğrafı eklemese null değer alır. Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).
- 9) Oyun servisimize kayıt olmuş kişilerin admin mi yoksa oyuncu mu olduğunu ayırmamız gerekmektedir. Bundan dolayı bu ayrımı yapabilmek için varchar(15) tipinde **rol** adında tutulacaktır.
- 10) Oyun servisimizde her oyuncunun bir seviyesi bulunmaktadır. Oyuncunun hangi seviyede olduğunu belirlemek adına oyuncu oyun oynadıkça değişen bir değer tutmamız gerekmektedir. Bundan dolayı bu değer integer(10) tipinde **seviye** adında tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).
- 11) Oyun servisimizde her oyuncunun bir rütbesi bulunmaktadır. Oyuncunun hangi rütbede olduğunu belirlemek adına oyuncu oyun oynadıkça değişen bir değer tutmamız gerekmektedir. Bundan dolayı bu değer integer(10) tipinde **rütbe** adında tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).
- 12) Oyun servisimizde her oyuncunun seviyesini belirlemek için seviye tablosundan bulunduğu seviyeye ait ID bilgisi foreign key tipinde alınır. Yani bu tabloda bigint(19) tipinde **SeviyeID** adında foreign key şeklinde tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).
- 13) Oyun servisimizde her oyuncunun rütbesini belirlemek için rütbe tablosundan bulunduğu rütbeye ait ID bilgisi foreign key tipinde alınır. Yani bu tabloda bigint(19) tipinde **RütbeID** adında foreign key şeklinde tutulacaktır. (Kullanıcı eğer github ve google üzerinden girerse bu değer null olacaktır).

#### 4.2. Rütbe

Bu tabloda oyun servisini kullanacak kullanıcıların rütbe bilgileri tutulacaktır. Oyuncu tablosu ile 1-N ilişkisi bulunmaktadır. Bu bilgiler aşağıda sıralanmıştır:

- 1) Oyun servisine kayıt olan her kullanıcının rütbesi olacaktır. Rütbe, bigint(19) tipinde **ID** adında primary key şeklinde tutulacaktır.
- 2) Rütbe, varchar(15) tipinde **rutbeAdi** adında bronz, altın gibi rütbe isimleriyle tutulacaktır.
- 3) Bu tabloda, integer(10) tipinde **minimumPuan** adında bir puan mevcuttur. Bu değer bu rütbeye ulaşmak için minimum değeri gösterir.

#### 4.3. Seviye

Bu tabloda oyun servisini kullanacak kullanıcıların seviye bilgileri tutulacaktır. Oyuncu tablosu ile 1-N ilişkisi bulunmaktadır. Bu bilgiler aşağıda sıralanmıştır:

- 1) Oyun servisine kayıt olan her kullanıcının seviyesi olacaktır. Seviye, bigint(19) tipinde **ID** adında primary key şeklinde tutulacaktır.
- 2) Seviye, varchar(15) tipinde **seviyeAdi** adında 1,2,3 gibi seviye isimleriyle tutulacaktır.
- 3) Bu tabloda, integer(10) tipinde **minimumPuan** adında bir puan tutulur. Bu değer bu seviyeye ulaşmak için gereken minimum değeri gösterir.

#### 4.4. Cüzdan

Bu tabloda oyun servisini kullanacak kullanıcıların cüzdan bilgileri tutulacaktır. Oyuncu tablosu ile 1-1 ilişkisi bulunmaktadır. Bu bilgiler aşağıda sıralanmıştır:

- 1) Oyun servisine kayıt olan her kullanıcının bir cüzdanı olacaktır. Cüzdan, integer(10) tipinde **suankiOyunParası** adında bir özellik içermektedir. Bu özellik kullanıcının hem oyundan kazanılan para ödülleri ile hem de kullanıcın kredi kartı ile marketten oyun parası satın alması ile artabilecektir. Aynı zamanda, eşya satın alım sonrası azalacaktır.
- 2) Cüzdana, hangi oyuncunun cüzdanının olduğunun ayırt edilebilmesi için Oyuncu tablosundan o oyuncuya ait ID bilgisi foreign key tipinde alınır. Yani bu bilgi bu tabloda, bigint(19) tipinde **OyuncuID** adında foreign key şeklinde tutulacaktır.

#### 4.5. Kayıt (Loglama)

Bu tablo oyun servisini kullanacak kullanıcıların log (kayıt) bilgileri tutulacaktır. Bu tablo kullanıcının eşya satın alım ve giriş çıkışlarının kayıtlarını tutmayı sağlar. Oyuncu tablosu ile N-1 ilişkisi bulunmaktadır. Bu bilgiler aşağıda sıralanmıştır:

- 1) Kayıt tablosu, bigint(19) tipinde **ID** adında primary key şeklinde tutulacaktır.
- 2) Log(kayıt), varchar(25) tipinde **tür** adında bir özellik içermektedir. Bu özellik kullanıcının eşya satın alım ve giriş çıkışlar gibi log(kayıt) türlerini tutmayı sağlar.
- 3) Kayıt tablosu, varchar(255) tipinde **mesaj** adında bir özellik içermektedir. Bu özellik kullanıcının oluşturulan kayıtları ile ilgili ayrıntıları içerir.
- 4) Kayıt tablosuna, hangi oyuncunun logunun(kaydının) olduğunun ayırt edilebilmesi için Oyuncu tablosundan o oyuncuya ait ID bilgisi foreign key tipinde alınır. Yani bu bilgi bu tabloda, bigint(19) tipinde **OyuncuID** adında foreign key şeklinde tutulacaktır.

#### 4.6. Satın\_Alım

Bu tablo oyun servisini kullanacak kullanıcıların satın alım bilgileri tutulacaktır. Bu tablo oyundaki eşyaların satın alındıktan sonraki satın alım bilgilerini içerir. Oyuncu tablosu ile N-1 ilişkisi bulunmaktadır. Bu sayede hangi oyuncunun satın alım bilgisinin olduğu anlaşılır. Satın alım tablosunda kullanıcının hangi eşyayı aldığı tutulmalıdır. Bu yüzden Satın\_Alım tablosu Eşya tablosu ile Satın\_Alım\_Eşya adında ilişki tablosu ile N-M ilişkisi vardır.

Bu bilgiler aşağıda sıralanmıştır:

- 1) Satın\_alım tablosu, bigint(19) tipinde **ID** adında primary key şeklinde tutulacaktır.
- 2) Bu tablo, integer(10) tipinde **satınAlımFiyatı** adında bir özellik içermektedir. Bu özellik kullanıcının aldığı eşyanın fiyat türüne göre(oyun içi para veya TL) satın alınan fiyatını içerir.
- 3) Bu tablo, varchar(15) tipinde **satınAlımTarihi** adında bir özellik içermektedir. Bu özellik kullanıcının aldığı eşyanın satın alım tarihini içerir.
- 4) Satın\_Alım tablosu integer(10) tipinde **adet** adında bir özellik içermektedir. Bu özellik, kullanıcının satın aldığı eşyanın adedini gösterir.
- 5) Bu tablo, varchar(25) tipinde **fiyatTürü** adında bir özellik içermektedir. Bu özellik, kullanıcının satın aldığı eşyayı oyun içi para mı yoksa TL mi olarak satın alındığını ifade eder.
- 6) Satın\_Alım tablosuna, hangi oyuncunun satın alım geçmişinin olduğunun ayırt edilebilmesi için Oyuncu tablosundan o oyuncuya ait ID bilgisi foreign key tipinde alınır. Yani bu bilgi bu tabloda, bigint(19) tipinde **OyuncuID** adında foreign key şeklinde tutulacaktır.

#### 4.7. Eşya

Bu tabloda oyun servisimizdeki eşyaların bilgileri tutulacaktır. Eşya tablosunun diğer tablolarla ilişkisi şu şekildedir:

Eşya tablosu, Satın\_Alım tablosu ile Satın\_Alım\_Eşya adında ilişki tablosu ile N-M ilişkisi vardır. Çünkü bir eşya birçok satın alımda bulunabileceği gibi satın alımda da birçok eşya bulunabilir.

Eşya tablosu, Canta tablosu ile Canta\_Eşya adında ilişki tablosu ile N-M ilişkisi vardır. Çünkü bir eşya birden fazla çantada bulunabileceği gibi bir çantada da birden fazla eşya olabilir.

Eşya tablosu, Özellikler tablosu ile Özellikler\_Eşya adında ilişki tablosu ile N-M ilişkisi vardır. Çünkü bir eşya birden fazla özelliğe sahip olabileceği gibi bir özellik birden fazla eşyada bulunabilir.

FiyatTarihi tablosu ile 1-N ilişkisi bulunmaktadır.

Kategori tablosu ile N-1 ilişkisi bulunmaktadır.

Eşya tablosunun diğer tablolarla ilişkisi tabloların açıklamalarında belirtilmiştir.

- 1) Oyun servisimizdeki her eşyanın birbirinden ayırt edilebilmesi için birincil anahtar olarak kullanılması amacıyla bir değere ihtiyacı vardır. Bundan dolayı bu çanta verisi bigint(19) tipinde **ID** adında birincil anahtar olarak tutulacaktır.
- 2) Oyun servisimizde ki her eşyanın bir adı bulunmaktadır. Bu eşya adı varchar(25) tipinde **esyaAdı** olarak tutulacaktır.
- 3) Oyun servisimizde ki her eşyanın bir sembolü bulunmaktadır. Bu sembol veri tabanında ismi ile tutulacağı için varchar(25) tipinde **sembol** olarak tutulacaktır.
- 4) Oyun servisimizdeki her eşyanın bir kategorisi olması gerekmektedir. Bundan dolayı bu bilgi bu tabloda, bigint(19) tipinde **KategoriID** adında foreign key şeklinde tutulacaktır.

#### 4.8. Canta

Bu tabloda oyuncuların çanta bilgisi tutulacaktır. Oyuncu tablosu ile 1-1 ilişkisi, Eşya tablosu ile Canta\_esya adında ilişki tablosu ile N-M ilişkisi vardır. Kullanıcının çantasındaki eşyaları çekebilmek için Eşya tablosu ile bu yüzden ilişki kurduk.

- 1) Oyun servisimize kayıt olan her oyuncunun bir çantası olması gerekmektedir. Bundan dolayı bu çanta verisi bigint(19) tipinde **ID** adında birincil anahtar olarak tutulacaktır.
- 2) Çantaya, hangi oyuncunun çantası olduğunu ayırt edilebilmesi için Oyuncu tablosundan o oyuncuya ait ID bilgisi foreign key tipinde alınır. Yani bu bilgi bu tabloda, bigint(19) tipinde **OyuncuID** adında foreign key şeklinde tutulacaktır.

#### 4.9. Kategori

Bu tabloda eşyaların kategorileri tutulacaktır eşya tablosu ile 1-N ilişkisi bulunmaktadır.

- 1) Oyun servisimizde ki her eşyanın bir kategorisi olması gerekmektedir. Bundan dolayı bu kategori verisi bigint(19) tipinde **ID** adında birincil anahtar olarak tutulacaktır.
- 2) Oyun servisimizde ki her kategorinin bir adı bulunmaktadır. Bu kategori adı varchar(25) tipinde **kategoriAdı** olarak tutulacaktır.
- 3) Oyun servisimizde ki her kategorinin bir sembolü bulunmaktadır. Bu sembol veri tabanında ismi ile tutulacağı için varchar(25) tipinde **sembol** olarak tutulacaktır.

#### 4.10. Özellikler

Bu tabloda oyundaki eşyaların özelliklerinin bilgisi tutulacaktır. Eşya tablosu ile Özellikler\_esya adında ilişki tablosu ile N-M ilişkisi vardır.

- 1) Oyun servisimizde ki her özelliği birbirinden ayırt etmek için bir birincil anahtar gerekmektedir. Bundan dolayı bu veri bigint(19) tipinde **ID** adında birincil anahtar olarak tutulacaktır.
- 2) Oyun servisimizde ki her özelliğin bir adı bulunmaktadır. Bu özellik adı, varchar(25) tipinde **adı** olarak tutulacaktır.
- 3) Oyun servisimizde ki her özelliğin kullanıcıyı bilgilendirmesi için bir açıklaması olmalıdır. Bundan dolayı bu açıklama varchar(100) tipinde **tanımı** olarak tutulacaktır.
- 4) Oyun servisimizde ki her özelliğin bir sembolü bulunmaktadır. Bu sembol veri tabanında ismi ile tutulacağı için varchar(25) tipinde **sembol** olarak tutulacaktır.
- 5) Oyun servisimizde ki her özelliğin şiddetini belirlemek için bir skaler büyüklüğe ihtiyaç vardır.(Bazı durumlarda gerekmebilir) Bundan dolayı bu değer integer(10) (boş bırakılabilir) tipinde **gucMiktarı** adında tutulacaktır.

#### 4.11. Fiyat\_Tarihi

Bu tabloda satın alınabilen eşyaların ileri tarihlerdeki fiyatlarını önceden belirleyebilmek için bilgiler tutulacaktır. Bir eşyanın birden fazla ileri tarihli fiyatı olabileceği için eşya tablosu ile N-1 ilişkisi vardır.

- 1) Her eşyanın ileri tarihli fiyatlarını birbirinden ayırabilmek için bir birincil anahtar gerekmektedir. Bundan dolayı bu veri bigint(19) tipinde **ID** adında birincil anahtar olarak tutulacaktır.
- 2) Tabloda bu fiyatların skaler büyüklüklerini tutmak amacıyla integer(10) türünde **fiyat** adıyla tutulacaktır.
- 3) Tabloda tablodaki fiyatın ne zaman geçerli olacağını belirlemek amacıyla varchar(15) tipinde **fiyatTarihi** adında tutulacaktır.
- 4) Tabloda ileri tarihli fiyatını belirlediği eşyanın TL ile mi yoksa oyun içi para ile alınıp alınmayacağını belirlemek için bir değere ihtiyaç vardır. Bundan dolayı bu değer varchar(25) tipinde **fiyatBirimi** adında tutulacaktır.
- 5) Bu tabloda hangi eşyanın ileri tarihli fiyatını belirlediğimizi tutmak amacıyla belirttiğimiz eşyanın birincil anahtarına ihtiyacımız vardır. Bundan dolayı bu veri bigint(19) tipinde **EşyaID** adında eşya tablosundan gelen foreign key tutulacaktır.

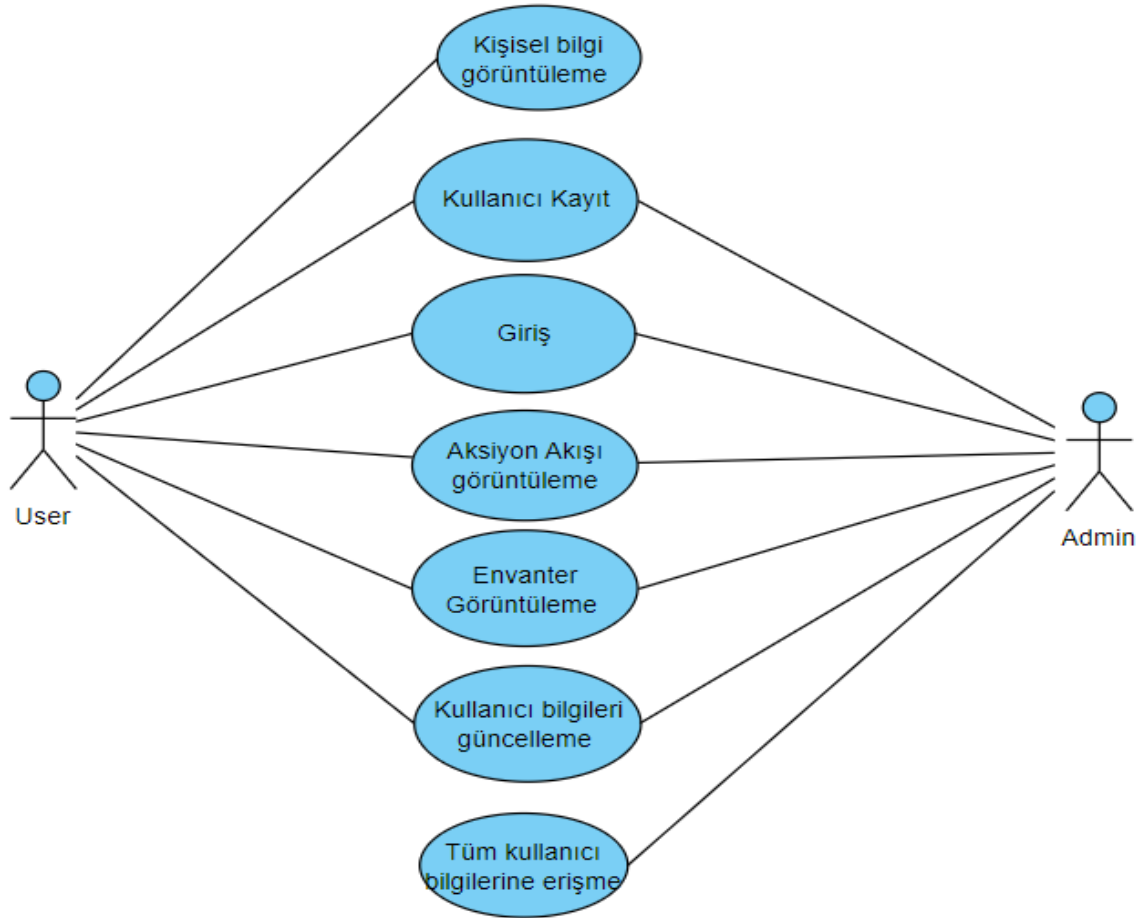
## 5. Sistem Modelleri

### 5.1. Aktörler

AKTÖR	TANIMI/ GÖREVİ
Kullanıcı	<ul style="list-style-type: none"> <li>➤ Custom (özel) kayıt form ekranımız ile kayıt olabilir ve giriş yapabilir.</li> <li>➤ Github veya Google hesabıyla kayıt olabilir ve giriş yapabilir.</li> <li>➤ Item (oyun eşyası) satın alabilir ve görüntüleyebilir.</li> <li>➤ Kendi çantasındaki eşyaları detaylı bilgileri ile görüntüleyebilir.</li> </ul>

	<ul style="list-style-type: none"> <li>➤ Oyun içi parasını görüntüleyebilir.</li> <li>➤ Hesap bilgilerini görüntüleyebilir (örneğin: rütbe, seviye bilgisi , ).</li> </ul>
Admin	<ul style="list-style-type: none"> <li>➤ Normal bir kullanıcının sahip olmadığı yetkilere sahiptir.</li> <li>➤ Satın alımları görüntüleyebilir.</li> <li>➤ Tüm kullanıcıların bilgilerini görüntüleyebilir.</li> <li>➤ Belirli bir eşyanın farklı tarihler için farklı fiyatlar ekleyebilir.</li> <li>➤ Kategori ekleyip çıkartabilir.</li> <li>➤ İtem ekleyip çıkartabilir.</li> </ul>

## 5.2. Use-Case Diyagramı



### 5.3. Uygulama Senaryosu

Senaryoya katılan varlık	Senaryo adı	Olay akışı
Yusuf Bey: Kullanıcı	Kayıt olma / Giriş yapma	1)Kullanıcı Uygulamayı açar. 2)Kayıt olma ekranı açılır. 3)Kullanıcı nereden kayıt olacağını seçer veya giriş yapar. (Github, Google, Custom kayıt formu/ Custom giriş formu) 4)Bilgilerini girer ve kayıt ol butonuna tıklar. 5)Eğer veri tabanında aynı mail adresi ve kullanıcı adına sahip başka bir kullanıcı yok ise kullanıcının kayıt olma işlemi tamamlanır.
Yusuf Bey: Kullanıcı	Çanta Görüntüleme	1)Kullanıcı giriş yapar. 2)Kullanıcı çanta görüntüle butonundan çantasına geçiş yapar. 3)Kullanıcı çantasındaki eşyalarını görüntüleyebilir.
Yusuf Bey: Kullanıcı	Profil Bilgilerini Görüntüleme	1)Kullanıcı giriş yapar. 2) Kullanıcı ana sayfadaki profil görüntüleme butonuna basıp profilini görüntüleyebilir.
Yusuf Bey: Kullanıcı	Marketteki Eşyaları Görüntüleme	1)Kullanıcı giriş yapar. 2)Kullanıcı market butonuna tıklayıp oyun içi eşyaları görüntüler.
Yusuf Bey: Kullanıcı	Eşya Satın Alım İşlemi	1) Kullanıcı markete girer. 2) Kullanıcı marketteki bir eşyanın üzerindeki satın al butonuna tıklar ve eşya kullanıcının çantasına eklenir.
Yusuf Bey: Kullanıcı	Satın Alımları Görüntüleme	1)Kullanıcı satın alımlarını görüntüle butonuna tıklar. 2) Kullanıcı kendi satın alım geçmişini görüntüler.
Serhat Bey: Admin	Kullanıcıları Görüntüleme	1)Serhat Bey giriş yapıp yönetici sayfasını açar 2)Kullanıcıları görüntüleme butonundan kullanıcıları listeleyebilir. 3)Ayrı ayrı kullanıcıların bilgilerini görüntüleyebilir.



Serhat Bey: Admin	Kategori Ekleme	<ol style="list-style-type: none"> <li>1) Serhat Bey giriş yapıp yönetici sayfasını açar.</li> <li>2) Kategori ekle butonuna tıklayıp kategori ekleme formu ile ekleme yapar.</li> <li>3) İsteddiği kategoriyi veri tabanına ekleyebilir.</li> <li>4)</li> </ol>
Serhat Bey: Admin	Eşya Ekleme	<ol style="list-style-type: none"> <li>1) Serhat Bey giriş yapıp yönetici sayfasını açar.</li> <li>2) Eşya ekle butonundan ekleme formu ile ekleme yapar.</li> </ol>
Serhat Bey: Admin	Kayıtları Görüntüleme	<ol style="list-style-type: none"> <li>1) Admin kayıtları görüntüle butonuna tıklar.</li> <li>2) Veri tabanındaki tüm kayıtları (logları) görüntüler.</li> </ol>
Serhat Bey: Admin	Satın Alımları Görüntüleme	<ol style="list-style-type: none"> <li>1) Admin satın alma ve görüntüleme butonuna tıklar.</li> <li>2) Tüm satın alımları görüntüler.</li> </ol>

## 6.Sorgular (Karmaşık Sorgu)

Bir eşyanın ilk olarak eşya ismini, bu eşyaya sahip olan kullanıcı sayısını, bu eşyanın dahil olduğu envanter sayısını ve bu eşyanın toplam satış sayısını gösterir. Bütün sorgularımız postgreSQL ile Spring Boot üzerinden yazılmıştır.

```
@Query("SELECT u.username, p.purchaseDate, SUM(p.amount), i.itemName " +
"FROM User u " +
"JOIN u.purchases p " +
"JOIN p.items i " +
"WHERE u.id = :userId " +
"GROUP BY u.username, p.purchaseDate, i.itemName " +
"ORDER BY SUM(p.amount) DESC")
List<Object[]> findTotalPurchaseDetailsByUserAndDate(@Param("userId") Long userId);
```

Bir kullanıcının hangi alışveriş tarihinde hangi üründen kaç adet satın aldığını gösterir. Bütün sorgularımız PostgreSQL ile Spring Boot üzerinden yazılmıştır.

```
// T0-DD: buraya itemin bütün price datelerini getiren method yazılacak
//her bir itemin ilk olarak item ismini, bu eşyaya sahip olan kullanıcı sayısını, bu eşyanın dahil olduğu envanter sayısını ve bu eşyanın toplam satış sayısını gö
1 usage 1 iremkvknr
@Query("SELECT i.itemName, COUNT(DISTINCT u.id) AS userCount, COUNT(DISTINCT inv.id) AS inventoryCount, COUNT(DISTINCT p.id) AS totalSales " +
"FROM Item i " +
"LEFT JOIN i.inventories inv " +
"LEFT JOIN inv.user u " +
"LEFT JOIN i.purchases p " +
"GROUP BY i.itemName")
List<Object[]> getInventoryUserCountsGroupByItem();
}
```

## 7.Vtys Araç Kullanımı

### 7.1. Tetikleyici (Trigger)

Yeni kullanıcının seviye ve sıralama puanlarını 'seviye' ve 'rütbe' tablolarında tanımlanan minimum puana ayarlar.

```
-- FUNCTION: public.update_user_level_rank()

-- DROP FUNCTION IF EXISTS public.update_user_level_rank();

CREATE OR REPLACE FUNCTION public.update_user_level_rank()
  RETURNS trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
  NEW.level_point_of_user := (SELECT min_point FROM level ORDER BY min_point ASC LIMIT 1);
  NEW.rank_point_of_user := (SELECT min_point FROM rank ORDER BY min_point ASC LIMIT 1);

  NEW.level_id := (SELECT id FROM level ORDER BY min_point ASC LIMIT 1);
  NEW.rank_id := (SELECT id FROM rank ORDER BY min_point ASC LIMIT 1);

  RETURN NEW;
END;
$BODY$;

ALTER FUNCTION public.update_user_level_rank()
  OWNER TO postgres;
```

255)	level_point_of_user integer	rank_point_of_user integer	symbol character varying (255)	budget_id bigint	inventory_id bigint	level_id bigint	rank_id bigint
	[null]	[null]	[null]	[null]	[null]	[null]	[null]
	[null]	[null]	[null]	[null]	[null]	[null]	[null]
	100	100	[null]	25	25	1	1
	100	100	[null]	26	26	1	1
	100	100	[null]	27	27	1	1

### 7.2. Error Handling

- Eğer yeni kayıt olmaya çalışan kişi istenen bilgileri boş bırakıp direk kayıt ol butonuna tıklarsa sistem girişini reddeder ve uyarı mesajı verir.
- Eğer yeni bir kullanıcı önceden sisteme kayıtlı bir kullanıcı adı ve mail adresi ile kayıt olmaya çalışırsa sistem kaydını reddeder ve uyarı mesajı verir.
- Kullanıcı yanlış kullanıcı adı veya şifre ile giriş yapmaya çalışırsa butona tıkladığında oturum açılması reddedilir ve hata mesajı alır.
- Kullanıcı, kullanıcı adı ve şifre bölümünü boş bırakarak giriş yapmaya çalışırsa hata mesajı alır.
- Kullanıcı kayıt olma esnasında ülke kodu ile uygun bir telefon numarası (+90505xxxxxxx) girmez ise hata mesajı alır.
- Servise giriş yaptıktan sonra servis tarafından aldığımız JWT'yi ( JSON Web Token ) kullanmayarak HTTP isteği atarsa sistem reddeder ve client tarafına hata mesajı gönderir.
- Client tarafından server tarafına atılan HTTP isteklerinin başarılı olmaması durumunda client tarafında isteği hata durumunda yakalayacağız ve sayfanın crash ( bozulma ) olmasını engelleyeceğiz.
- Sistemimizde oyun parası ile fiyat tipi TL olan eşyaları satın alamaz çünkü bu kısımda sunucumuzun client (istemci) tarafına hata mesajı göndermesi istendiği için böyle bir

implementasyon yaptık. Bu aşamada şirketin oyun geliştiricileri istemci tarafından bu mesajı alıp kullanıcıyı markete (play store, app store) yönlendirecekler.

## 8. Proje Kısıtları

### 8.1. Kaynak kısıtlaması

- Projede, oyun motorunun ve tasarımının bitmiş olmamasından kaynaklı bizim sınırlı veri ve isterle ilerliyor oluşumuz.
- Big dataya sahip olmamak.
- Cloud platformlarda bütçeye sahip olmama (Resim ve dosyalar gibi veri parçalarının cloud tarafıyla değil de client tarafında dosyalarda tutulmasının tercih edilmesi.)

### 8.2. Bütçe Kısıtlaması

- Veri tabanının boyutu her geçen gün artması öngörülmektedir, bu nedenle veri tabanı yedekleme ve veri bakım maliyetini arttırmaktadır.
- Personel eksikliği
- Kısıtlı bir bütçeye sahip olmak

### 8.3. Risk Kısıtlaması

- Proje revize süresinin uzun olması
- Benzer projelerin yatırım alması
- Ürünün arzı karşılayamaması
- Yeterli reklam çalışmasının yapılamaması

### 8.4. Düşünülmeyen Kısıtlar

- Rekabet ortamının düşünülenden daha fazla olması
- Bütçe planlamasının öngörülenden fazla olması


## 9. Sunuma Sonradan Eklenenler

- Giriş kısmında projeyi seçme amacımız ve hedefleri başlıkları altında bazı güncellemeler yaptık.
- Sistem gereksinimleri kısmında fonksiyonel olan, fonksiyonel olmayan ve veri isterleri başlıkları altında eklemeler yaptık.
- Error handling başlığı altındaki içeriği düzenledik.
- Modül açıklamalarını değişen isterler doğrultusunda yeniledik.
- Sistem modelleri başlığında aktörler tablomuzu yeniledik.
- Vtys Tasarımı İçin Olan Gereksinimler başlığı ekledik.
- ER diyagramını hem şirket isterleri doğrultusunda hem de ders hocasının geri bildirimine göre tekrar düzenledik.
- Ara yüzümüze eklemeler yaptık.

## 11. Kullanıcı Arayüzü Görüntüleri

Kullanıcılar servise girmek için kayıt olurken ve giriş yaparken görecekları sayfalar bunlardır.

Game Service Client App



### Hello, Again

We are happy to have you back.


Username:

Password:

[Don't have an account? Sign Up.](#)

[Log in with Google](#) [Log in with Github](#)

Login



### Hello, New User

We are excited to welcome you!

Username

Email

Phone Number

Password

Choose your role:  
☐ Normal User ☐ Admin

[Already registered? Login](#)

Register

Bu sayfada normal kullanıcı giriş yaptığında bu aksiyonları görecektir.

Game Service Client App

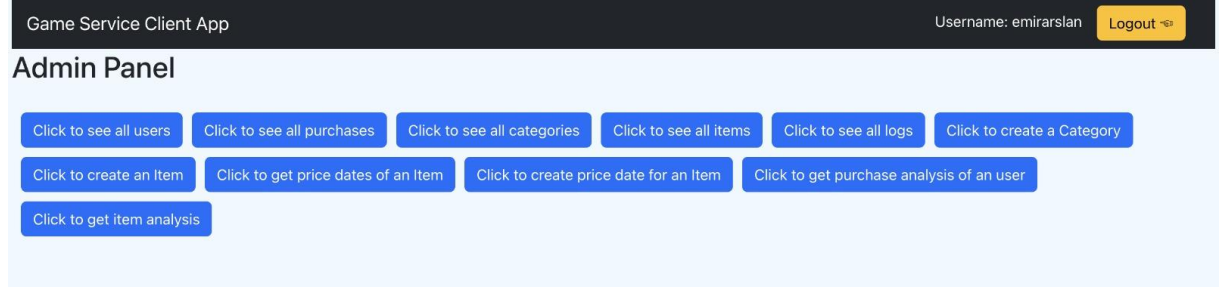
Username: yakub.alkis Logout

### User Panel

[Click to see your personal details](#) [Click to add/sub game money to your budget](#) [Click to see all items](#) [Click to buy an item](#) [Click to see your inventory](#)

[Click to see your purchases](#)

Bu sayfada admin giriş yaptığında bu aksiyonları görecektir.



## 12. Diğer Entegrasyonlar

### 12.1. Sosyal medya ile login

Sosyal medya bağlantısı olarak google ve github butonlarına tıkladığında kayıt olup girişleri yapılıyor.

### 12.2. E-mail/GSM msg entegrasyonu

Servisimize kayıt olduğunda ve her giriş yapıldığında veri tabanında bulunan kullanıcı e-mail ve telefon numarasına bildirim gitmektedir.

### 12.3. XLS Import

Burada kullanıcı veya yönetici game servise girdiğinde önceden yüklü olması gereken verileri (eşya, kategori, özellik, fiyat tarihleri) excel dosyası üzerinden import ediyoruz.

## 13. Toplantılar

TARİH	KATILIMCILAR	TOPLANTI KONUSU	TOPLANTI SONRASI ALINAN KARARLAR
21.10.2023	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Tanışma, proje konusu araştırması	Herkesin proje konusu araştırılmasına karar verildi.
23.10.2022	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Proje için firmalarla görüşülmesi ve gerçek bir projeye imza atılması adına şirketlerle iletişime geçilmesi	Firmalarla iletişime geçildi.
26.10.2022	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Firma tarafından belirtilen taskların gözden geçirilmesi, yapılan araştırmaların değerlendirilmesi.	Görev dağılımı yapıldı. Firma yetkilisi tarafından neler yapılacağı detaylı bir şekilde açıklandı.
11.11.2023	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Rapor görüşmesi gerçekleştirilmesi.	Proje raporu oluşturuldu. Proje hakkındaki fikirler grup üyeleri arasında konuşuldu.
24.11.2023	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR	Projeye başlangıcı.	Projenin kodlanması için gerekli görev dağılımlar yapıldı.

	Okan SEVİNÇ		
10.12.2023	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Rapor hazırlığı	Yapılan işlemler değerlendirilip rapor hazırlandı.
16.12.2023	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Proje ilerleyişi değerlendirme	Hazırlanan kodların toplanması ve değerlendirilmesi.
22.12.2023	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	3. Rapor hazırlığı	Neler yapılacağı hakkında değerlendirme yaptık.
2.01.2024	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Görev dağılımı	Kimin ne yapacağı hakkında konuşuldu.
5.01.2024	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Proje eksiklerinin gözden geçirilmesi	Eklenmesi gereken kısımlar tartışıldı.
6.01.2024	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Proje eksikleri için güncellemelerin yapılması	Eksiklerin tamamlanması
7.01.2024	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Projenin tamamlanması.	Projenin tamamlanıp kontrollerin yapılması
10.01.2024	Buse ERCAN Yakub ALKIŞ İrem KAVAKPINAR Okan SEVİNÇ	Final Rapor Hazırlığı	Son rapor için eksikler tamamlandı.

## 13. Proje ekibi değerlendirmesi

### 13.1. Grup bilgisi

Grup Koordinatörü: Buse Ercan

Görev Paylaşımı:

Buse Ercan: o Veri tabanı tasarımı

o Kayıt veya giriş yapan kullanıcıya telefon numarası ile bilgilendirme

o Front-end geliştirme.

o Back-end API yazma

o Rapor ve sunum hazırlanması

Yakub Alkış: o Veri tabanı tasarımı

o Front-end hazırlama.

o Back-end API yazma.

o REST-API 'nin güvenlik kısmını JWT(JSON Web Token) ve oauth2 için güvenlik configürasyonları

o Rapor ve sunum hazırlanması

İrem Kavakpınar: o Veri tabanı tasarımı

o Back-end API yazma.

o Front-end geliştirme

o Kayıt veya giriş yapan kullanıcıya e-mail ile bilgilendirme

o Rapor ve sunum hazırlanması

Okan Sevinç: o Veri tabanı tasarımı

o Front-end hazırlama.

o Back-end API yazma

o Data Import/Export işlemleri

o Rapor ve sunum hazırlanması

## 13.2. Toplam Çalışma süreleri

- Buse Ercan: 86 Adam/ Saat
- Yakub Alkış: 90 Adam/Saat
- İrem Kavakpınar: 87 Adam/Saat
- Okan Sevinç: 85 Adam/Saat