

**KOCAELİ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**YAZILIM LAB. I- 3.Proje**  
**PROJE TESLİM TARİHİ: 28.12.2024**

**Eş Zamanlı Sipariş ve Stok Yönetimi Uygulaması**

---



**Amaç:** Bu proje, eş zamanlı sipariş yönetimi ve stok güncellemelerini gerçekleştiren bir sistem tasarlamayı amaçlar. Amaç, multithreading ve senkronizasyon mekanizmaları kullanarak aynı kaynağa eş zamanlı erişim problemlerini çözmek ve process, thread, mutex, semafor ve process önceliği gibi kavramların kullanımını kavramaktır.

**Programlama Dilleri:** Java, JavaScript, Python, C, C#.

**Veritabanları:** MSSQL, MySQL, PostgreSQL, MongoDB, Firebase Realtime Database.

**Not:** Masaüstü uygulama yapabilirsiniz. Web uygulama yapmak isterseniz client/server mimarisinin tasarımı iyi kurgulanmış olmalıdır.

---

## PROJE GEREKSİNİMLERİ

### Müşteriler ve Türleri:

- ✓ Müşteri bilgileri: *CustomerID, CustomerName, Budget, CustomerType (Premium/Standard), TotalSpent.*
- ✓ Müşteriler başlangıçta random sayıda atanmalıdır, bütçeler de benzer şekilde random atanmalıdır.

- Müşteri sayısı 5-10 arasında değişmelidir.
- Bütçe miktarı 500-3000 TL arasında değişmelidir.
- Başlangıçta en az 2 adet premium müşteri bulunmalıdır.
- İki tür müşteri vardır:
  - **Premium Müşteriler:** Bu müşteriler yüksek öncelikli olup işlemleri normal müşterilerden önce gerçekleştirilir.
  - **Normal (Standard) Müşteriler:** İşlemleri, Premium müşterilerin işlemleri tamamlandıktan sonra işlem sırasına girer.
- Aynı anda birden fazla müşteri aynı ürünü almak istediğinde, Premium ve Normal müşteriler arasında uygun bir işlem sırası sağlanmalıdır.
- ✓ Her müşteri, belirli bir veya birden fazla ürün çeşidini satın almak için başvurabilir.
- ✓ Bir müşteri, her üründen **en fazla 5 adet** satın alabilir.

#### Admin:

- ✓ Admin sistemi yönetir:
  - Ürün ekleme, silme ve stok güncelleme işlemlerini yapar.
- Admin işlemleri, müşteri işlemleri ile paralel olarak gerçekleştirilir.
- Admin, ürün ve stok bilgilerine erişirken diğer işlemleri beklemeye alır.

#### Stok Yönetimi:

- ✓ Her ürün başlangıçta sistem tarafından sabit bir stok miktarı ile tanımlanır.
- ✓ Mağazada başlangıçta 5 farklı ürün bulunmaktadır ve ürünlere ait bilgiler aşağıdaki tabloda verilmiştir:

<i>ProductID</i>	<i>ProductName</i>	<i>Stock</i>	<i>Price (TL)</i>
1	Product1	500	100
2	Product2	10	50
3	Product3	200	45
4	Product4	75	75
5	Product5	0	500

- ✓ Admin yeni ürünler ekleyebilir veya mevcut ürünleri silebilir.
- ✓ Admin, stok miktarlarını artırabilir veya azaltabilir.
- ✓ Ürün stokları satın almadan dolayı eş zamanlı olarak birden fazla müşteri tarafından güncellenmek istenebilir.
- ✓ Eğer bir ürünün stoğu yeterli değilse, işlem reddedilir.
- ✓ Ürün satın alındığında stoklar hemen güncellenmelidir.

### Bütçe (Budget) Yönetimi:

- ✓ Müşteriler, bir bakiye hesabına sahip olur ve ödeme işlemleri bu bakiyeden yapılır.
- ✓ Müşteri bakiyesi yetersiz olduğunda işlem reddedilir.
- ✓ Bütçeler random ( 500-3000 TL arası) olarak atanır.

### Dinamik Öncelik Sistemi:

- ✓ Müşteri işlemleri sırasında, işlem süresine göre öncelik dinamik olarak değişmelidir.
- Bekleme süresi ve müşteri türü gibi kriterlere dayalı olarak öncelik sıralaması sürekli olarak güncellenmelidir.
- ✓ Premium müşteriler varsayılan olarak yüksek önceliğe sahiptir. Normal müşteriler için bekleme süresi arttıkça öncelik sıralaması yükseltilir.
- Eğer bir Standard müşteri Premium müşteri olduysa, müşterinin mevcut işlem sırasına dokunulmamalıdır. Müşteri türü değişikliği bir sonraki işlemde etkili olmalıdır.

### Dinamik Öncelik Hesaplama:

- Her müşteriye bir öncelik skoru atanır:

$$\text{ÖncelikSkoru} = \text{TemelÖncelikSkoru} + (\text{BeklemeSüresi} \times \text{BeklemeSüresiAğırlığı})$$

- **Temel Öncelik Skoru:** Premium müşteriler için 15, Normal müşteriler için 10 olarak tanımlanmalıdır.
- **Bekleme Süresi:** Saniye cinsinden müşterinin sipariş ver butonuna bastıktan sonra admin tarafından siparişin onaylanmasına kadar geçen süreyi ifade eder.
- **Bekleme Süresi Ağırlığı:** Bekleme süresinin etkisini belirler. Her bir saniye bekleme süresinin ağırlığı 0.5 puandır.
- Her işlem sırasında sıralama, dinamik olarak yeniden hesaplanmalıdır.

### Veritabanı:

- Veritabanında olması beklenen minimum tablolar ve atributeleri aşağıda verilmiştir:
  - Customers: *CustomerID, CustomerName, Budget, CustomerType, TotalSpent*
  - Products: *ProductID, ProductName, Stock, Price*
  - Orders: *OrderID, CustomerID, ProductID, Quantity, TotalPrice, OrderDate, OrderStatus*
  - Logs: *LogID, CustomerID, OrderID, LogDate, LogType, LogDetails.*

### Loglama ve İzleme:

- Her işlem başladığında, işlem hakkında bir log kaydedilir.
- Log kaydı hem müşteri hem de admin işlemleri için tutulmalıdır.
- Loglar, işlem sırasına göre oluşturulmalıdır.
- Her log kaydı aşağıdaki bilgileri içermelidir:
  - Log ID

- Müşteri ID
- Log türü: “Hata”, “Uyarı”, “Bilgilendirme”
- Müşteri türü: Premium veya Standard
- Satın alınan ürün ve miktar
- İşlem (satın alma) zaman bilgisi
- İşlem (satın alma) sonucu: İşlem başarıyla tamamlanmışsa işlem sonucu olarak "Satın alma başarılı" kaydedilir. Eğer işlem başarısız olursa aşağıdaki hata mesajlarından uygun olan mesaj kaydedilir.

#### Bazı Hata Mesajı Türleri:

- ✓ ○ “Ürün stoğu yetersiz”: Bir müşteri satın alma işlemine başvurduğunda, ürün stoğunun yeterli olmaması durumunda tetiklenir.
  - “Zaman aşımı”: İşlem belirlenen maksimum sürede tamamlanamazsa tetiklenir.
  - ✓ ○ “Müşteri bakiyesi yetersiz”: Müşterinin bütçesi, seçilen ürün(ler)in toplam fiyatını karşılamıyorsa tetiklenir.
  - “Veritabanı Hatası”: Bağlantı sorunları veya işlem sırasında oluşan deadlock'lar gibi durumlarda tetiklenir.
- Log Kaydı Örneği:
    - **Log ID:** 4
    - **Müşteri ID:** 1
    - **Log Türü:** Hata
    - **Müşteri Türü:** Premium
    - **Ürün:** Product5
    - **Satın Alınan Miktar:** 5
    - **İşlem Zamanı:** 2024-11-28 14:32
    - **İşlem Sonucu:** Ürün stoğu yetersiz

#### UI Entegrasyonu:

- **Müşteri Paneli**

- ✓ ○ **Müşteri Listesi:**

- Tabloda **CustomerID, Ad, Tür (Premium/Normal), Bütçe, Bekleme Süresi, Öncelik Skoru** gibi bilgileri gösterin.

- ✓ ○ **Sipariş Oluşturma Formu:**

- Müşterinin **Ürün Seçimi, Adet Girişi** ve **Sipariş Ver** butonu ile işlem yapmasına izin verin.

- **Bekleme Durumu:**

- Her müşterinin sipariş sırasındaki durumu (bekliyor, işleniyor, tamamlandı) renklendirme ile görselleştirilebilir.

- ✓ ● **Ürün Stok Durumu Paneli**

- **Ürün Tablosu:**

- **Ürün Adı, Stok Miktarı, Fiyat** gibi bilgiler tablo halinde gösterilir.

- Stok miktarları her işlem sonrası güncellenir.
- **Grafik Temsili:**
  - Bar veya dairesel grafik ile stok durumu görselleştirilebilir.
  - Stok kritik seviyeye geldiğinde renk değiştiren bir grafik ile görsel uyarı eklenebilir.
- **Log Paneli**
  - Gerçek zamanlı loglama ile her işlemin sonucunu listeler:
    - Örneğin: "Müşteri 1, Product3'ten 2 adet aldı. İşlem Başarılı."
    - Hatalar için: "Müşteri 2, Product5'ten 3 adet almak istedi. Yetersiz Stok."
  - Loglar sırayla kayar bir liste şeklinde UT'da işlemler gerçekleşirken eş zamanlı gösterilmelidir. Aşağıda bir örnek verilmiştir ancak bu örnek herhangi bir senaryoya ait değildir.
  - **Örneğin:**
    - Müşteri 1 (premium) product1'den 10 adet sipariş verdi.
    - Müşteri 2 (normal) product2'den 50 adet, product5'ten 30 adet sipariş verdi.
    - Müşteri 3 (premium) product3'ten 10 adet, product4'ten 100 adet sipariş verdi.
    - Müşteri 1 (premium) siparişi işleme alındı. (Bu işlem müşteri sipariş ver butonuna bastığında gerçekleşir.)
    - Müşterinin siparişi onaylandı. (Bu işlem admin siparişleri onayladığında gerçekleşir.)
    - ...
    - Müşteri 4'ün siparişi stok yetersizliğinden iptal edildi.
- **Dinamik Öncelik ve Bekleme Paneli**
  - **Bekleme Süresi ve Öncelik Skoru** bir tabloda gösterilir ve her işlemde güncellenir.
  - Müşteri sırasını animasyonla gösterebilirsiniz:
    - Örneğin, sipariş sırası değiştiğinde listeler yukarı/aşağı hareket eder.
- **Sipariş İşleme Animasyonu**
  - İşlemden olan siparişler için bir animasyon göstergesi:
    - Örneğin: "Müşteri 1'in siparişi işleniyor" şeklinde bir yükleme çubuğu veya hareketli bir simge.

---

### **Proje Teslimi:**

- Rapor IEEE formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır. Veritabanındaki tablolar arasında kurulan ilişki raporda **ER diyagramı** ile belirtilmelidir.
- Dersin takibi projenin teslimi dahil edestek2.kocaeli.edu.tr sistemi üzerinden

yapılacaktır. edestek2.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.

- Proje ile ilgili sorular yalnızca **edestek2.kocaeli.edu.tr** sitesindeki forum üzerinden **Arş. Gör. Kübra ERAT** veya **Arş. Gör. Gamze KORKMAZ ERDEM'e** sorulabilir.
- Demo sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamamız istenebilir.
- Proje grupları, her öğretim kendi içinde olmak üzere **en fazla iki kişiden** oluşturulmalıdır.