

# YAZILIM LABORATUVARI-1

## PROJE-3

1. Zehra KARABEKTAŞ

*Bilgisayar Mühendisliği*

*Kocaeli Üniversitesi*

Kocaeli, Türkiye

zhrakarbektas679@gmail.com

**Özetçe—**Bu proje, kullanıcıların eş zamanlı sipariş verebildiği, admin tarafından yönetilen ve stok güncellemelerini gerçekleştiren bir web sitesi projesidir. Multithreading ve senkronizasyon mekanizmaları kullanılarak aynı kaynağa eş zamanlı erişimler yönetilmiş olup process, thread, mutex, semafor ve process önceliği gibi senkronizasyon yapıları kullanılmıştır.

**Anahtar Kelimeler —** Asp.Net Core Mvc; Asp.Net Core Api Veritabanı Yönetimi; C# ; MSSQL ; Dapper; Senkronizasyon; Mutex; Semafor; Thread .

### I. ÖZET

Bu proje, müşterilerin eş zamanlı sipariş verebildikleri, adminin yönetimi ile siparişlerin onaylandığı ve senkronizasyon yöntemi ile bütçe ve stok güncellemelerini gerçekleştiren bir web sitesi tasarlanmıştır. ASP.NET Core MVC ve ASP.NET Core API kullanılarak, kullanıcılar için dinamik ve hızlı bir arayüz sunulmuş, arka planda güçlü bir API ile veri işlemleri yapılmıştır. SQL sorguları Dapper kütüphanesi ile optimize edilerek gerçekleştirilmiş, giriş işlemleri JWT token ile güvence altına alınmıştır. Kullanıcı deneyimini artırmak amacıyla animasyonlar ve fetch ile sayfa yenileme işlemleri yapılmıştır.

### II. GİRİŞ

Bu projede, eş zamanlı çalışan bir sipariş ve stok yönetimi sistemi geliştirmek amaçlanmış olup web sitesi olarak gerçekleştirilmeye çalışılmıştır. Proje oluşturulurken, modern teknolojiler kullanılarak, kullanıcıların hızlı ve güvenli bir şekilde işlem yapabilmesini sağlayacak şekilde tasarlanmıştır. Projenin temel yapısı ASP.NET Core MVC ve ASP.NET Core API üzerine inşa edilmiş olup veritabanı işlemleri için Dapper kütüphanesi kullanılarak verimli, işlemleri daha hızlı ve hafif bir sorgu yapısı ile MSSQL veritabanı ile ASP.NET Core API arasında bağlantı kurulmuştur. Veri güvenliği ve kimlik doğrulama işlemleri JWT token yöntemiyle güvence altına alınmıştır. Kullanıcı deneyimini artırmak ve sistemi daha akıcı hale getirmek için fetch API ile sayfa yenileme olmadan veri alışverişi gerçekleştirilmiş, animasyonlar ile görsel zenginlik ve işlem akışını daha kolay anlaşılır hale getirme sağlanmıştır.

Projenin temel amacı, müşteri ve stok yönetimi işlemlerini eş zamanlı olarak yönetirken veri bütünlüğünü korumak ve kaynakların güvenli bir şekilde kullanılmasını sağlamaktır. Bu doğrultuda, sistem müşterilerin sipariş işlemlerini yönetirken

aynı anda admin işlemleri gibi kritik süreçlerin de paralel olarak çalışmasına olanak tanımaktadır. Müşteri işlemleri sırasında dinamik öncelik sistemi devreye girerek Premium ve Normal müşteriler arasında işlem sırası belirlenmiş, müşteri durumu kullanılarak bekleme süresi ile adminin butona bastığı anda öncelik skorları hesaplanarak gibi müşteri sipariş sırası dinamik olarak güncellenmiştir.

Stok yönetimi tarafında, her ürün başlangıçta sabit stok miktarlarıyla tanımlanmış ve satın alma işlemleri sırasında bu stok miktarları güncellenmiştir. Stok yetersizliği durumunda ilgili işlemler reddedilmiş ve bu durum detaylı bir loglama sistemi ile kaydedilmiştir. Sistem, müşterilerin bütçelerini yönetmek için ayrı bir katman barındırmış ve müşterinin bütçesi yetersiz olduğunda sipariş işlemleri durdurularak sipariş iptal edilmiştir.

Kullanıcı arayüzü ASP.NET Core MVC ile geliştirilmiş ve sistemin işleyişini kullanıcıya kolayca anlamalarını sağlayacak şekilde tasarlanmıştır. Kullanıcılar, müşteri paneli aracılığıyla sipariş oluşturabilir, stok durumlarını görebilir ve geçmiş işlemleri izleyebilir. Admin paneli, ürün ekleme, silme ve stok güncellemeleri gibi kritik yönetim işlemleri için kullanılmıştır. Loglama paneli, gerçekleşen işlemlerin detaylarını gerçek zamanlı olarak sunmuş ve sistemin izlenebilirliğini arttırmıştır.

Bu proje, eş zamanlı işlemlerde senkronizasyon problemlerinin nasıl çözülebileceğini ve çok katmanlı bir mimari ile güvenli bir sistemin nasıl oluşturulacağını göstermeyi hedeflenmiş olup hem kullanıcı dostu bir arayüz hem de güçlü bir altyapı sunarak eş zamanlı süreçlerin etkin bir şekilde yönetimini sağlamak amaçlanmıştır.

### III. YÖNTEM

Bu projede C# ,veritabanı için MsSql ,veritabanı api arasındaki bağlantıyı kurmak için Dapper, kullanıcı arayüzü oluştururken Asp.Net Core Mvc ve Razor View, giriş işlemlerinde jwt token modeli kullanılarak tasarlanmıştır.

#### A. Veritabanı Tasarımı

Projede veritabanı tasarımı MsSql veritabanı kullanılarak yapılmıştır. Veritabanı tasarımı, müşteri, ürün, sipariş ve yönetim süreçlerini kapsayan bir yapıdadır. Bu tasarımda, sistemdeki temel varlıklar olan müşteriler, ürünler, siparişler ve

yönetim işlemleri farklı tablolar aracılığıyla organize edilmiştir. Tablolar arasındaki ilişkiler veri bütünlüğünü sağlamak ve işlemleri optimize etmek amacıyla detaylı bir şekilde tasarlanmıştır.

Customers tablosu, müşterilere ait bilgileri tutar. Bu tabloda her müşteri için benzersiz bir kimlik (CustomerID) tanımlanmıştır. Müşteri adı, e-posta adresi, şifre, adres, bütçe, müşteri türü, toplam harcama ve hesap aktiflik durumu gibi bilgiler burada saklanır. Customers tablosu, Orders ve Logs tabloları ile bire-çok (1:N) ilişkisine sahiptir. Bir müşteri birden fazla sipariş verebilir ve birden fazla işlem kaydına sahip olabilir. Bu ilişkiler, Orders ve Logs tablolarındaki CustomerID sütunlarının, Customers tablosundaki CustomerID sütununa yabancı anahtar olarak bağlanmasıyla sağlanmıştır.

Products tablosu, sistemde bulunan ürünlere dair bilgileri içerir. Bu tabloda her ürün için benzersiz bir kimlik (ProductID) tanımlanmıştır. Ürün adı, stok durumu, fiyat ve ürün görsel yolu gibi bilgiler burada yer alır. Products tablosu, Orders ve Sepetim tabloları ile bire-çok (1:N) ilişkisine sahiptir. Bir ürün birden fazla siparişte veya birden fazla müşterinin sepetinde yer alabilir. Bu ilişkiler, Orders ve Sepetim tablolarındaki ProductID sütunlarının, Products tablosundaki ProductID sütununa yabancı anahtar olarak bağlanmasıyla sağlanmıştır.

Orders tablosu, verilen siparişlere ait bilgileri tutar. Her sipariş benzersiz bir kimlik (OrderID) ile tanımlanmıştır. Ayrıca siparişin hangi müşteriye ait olduğu (CustomerID), hangi ürünün sipariş edildiği (ProductID), sipariş miktarı, toplam fiyatı, sipariş tarihi, durumu, bekleme süresi ve öncelik skoru gibi detaylar bu tabloda yer alır. Orders tablosu, hem Customers hem de Products tablolarıyla ilişkilidir.

Logs tablosu, sistemde gerçekleşen işlem kayıtlarını saklar. Bu tabloda her log için benzersiz bir kimlik (LogID) tanımlanmıştır. Log kaydının hangi müşteriye ait olduğu, ilgili sipariş bilgisi, log tarihi, türü ve detayları bu tabloda yer alır. Logs tablosu, Customers tablosu ile bire-çok (1:N) ilişkisindedir.

Sepetim tablosu, müşterilerin sepetlerine ekledikleri ürünleri takip eder. Bu tabloda her kayıt benzersiz bir kimlik (Id) ile tanımlanmıştır. Sepete eklenen ürün miktarı, toplam fiyatı ve oluşturulma tarihi gibi bilgiler burada saklanır. Sepetim tablosu, hem Customers hem de Products tablolarıyla bire-çok (1:N) ilişkisine sahiptir. Bir müşteri birden fazla ürünü sepetine ekleyebilir ve bir ürün birden fazla müşterinin sepetinde yer alabilir.

Admin tablosu, sistem yöneticilerine ait bilgileri saklar. Bu tabloda her yönetici benzersiz bir kimlik (AdminID) ile tanımlanmıştır. Yöneticinin adı, e-posta adresi ve şifre bilgileri burada tutulur. Admin tablosunun diğer tablolarla doğrudan bir ilişkisi bulunmamaktadır; bu tablo daha çok yönetimsel görevler için bağımsız bir yapı olarak tasarlanmıştır.

Müşterilerin siparişlerini ve ürünleri yönetecek kişiler bu tabloda tutulmuştur.

Bu tasarım sayesinde müşteriler, ürünler, siparişler ve loglar arasında tutarlılık sağlanmış; sistemdeki işlemlerin etkin bir şekilde izlenmesi, yönetilmesi ve analiz edilmesi mümkün hale getirilmiştir. Tablolar arasındaki ilişkiler, yabancı anahtarlar aracılığıyla kurulmuş olup veri bütünlüğü garanti altına alınmıştır.

## B. Kullanıcı Arayüzü (GUI) Tasarımı

Uygulamada kullanıcı arayüzü, Admin ve Customer girişlerinden oluşmaktadır ve Razor View teknolojisi kullanılarak geliştirilmiştir. Admin paneli, log yönetimi, sipariş yönetimi, ürün yönetimi ve müşteri bilgileri üzerinde işlem yapma gibi işlevleri içermektedir. Log ve sipariş panelleri, fetch API ile sürekli güncellenmekte ve dinamik animasyonlarla kullanıcı dostu bir deneyim sunmaktadır. Sipariş panelinde, siparişler gerçek zamanlı olarak sıralanır ve işlenen siparişler animasyonlarla vurgulanmıştır. Log paneli ise yeni log kayıtları geldikçe kayan bir animasyonla sayfa yenilemeden ekranda gösterilmiştir. Ürün ve müşteri panellerinde ekleme, güncelleme ve silme işlemleri kolayca yapılabilmektedir. Customer paneli, müşterilerin sipariş oluşturmaya, mevcut siparişlerini gerçek zamanlı olarak takip etmesine ve hesap bilgilerini yönetmesine olanak tanır. Ayrıca müşteri panelinde Sepetim özelliği bulunmaktadır. Bu özellik sayesinde müşteriler birden fazla ürünü aynı anda sepetlerine ekleyebilir ve toplu olarak satın alabilirler. Sepet yönetimi, Razor View ile dinamik olarak tasarlanmış ve kullanıcıların ürün miktarlarını belirleyerek toplam fiyatı kolayca görmesine imkân sağlanmıştır. Fetch API ile dinamik veri akışı sağlanarak, sepet ve sipariş bilgilerinin sürekli güncel kalması mümkün kılınmıştır. Razor View ve MVC mimarisi sayesinde sistem, hem kullanıcı dostu bir arayüz hem de güçlü bir altyapı sunmaktadır. Sepet yönetimi, dinamik sipariş güncellemeleri ve kullanıcı odaklı tasarımı ile bu yapı, hem admin hem de müşteri kullanıcılarına etkin ve modern bir deneyim sağlamaktadır. UI ile Api arasındaki bağlantı controller sınıflarında client/server mimarisi ile yapıldı.

## C. Fonksiyonel Özellikler

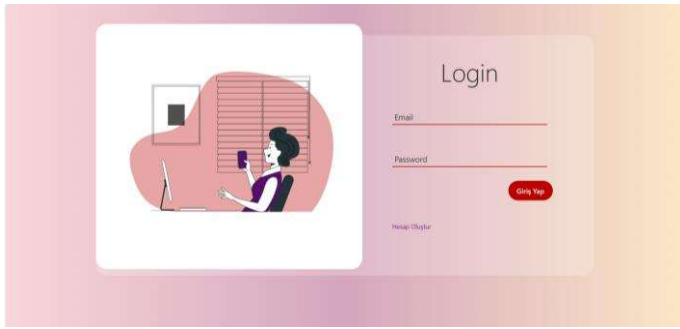
Bu proje, eş zamanlı sipariş yönetimi ve stok kontrolü için multithreading, mutex ve semaphore gibi senkronizasyon mekanizmalarını kullanan bir sistem geliştirmeyi amaçlamaktadır. Sistem, hem müşteri hem de admin işlemlerini eş zamanlı bir şekilde ele alırken, kaynakların güvenli ve tutarlı bir şekilde yönetilmesini sağlar. Admin tarafından başlatılan sipariş onaylama işlemi, bekleyen siparişlerin toplanarak müşteri türü (Premium veya Standart) ve öncelik skoruna göre sıralanmasını içerir. Premium müşterilere ait siparişler daha yüksek önceliğe sahiptir ve öncelikli olarak işlenir. Bu işlem sırasında semaphore mekanizması kullanılarak admin işlemleri sırasında diğer işlemler beklemeye alınır. Her sipariş için ayrı bir iş parçacığı oluşturulur ve bu parçacıklar, bekleme süresi ile öncelik skorunu hesaplayarak sipariş durumunu günceller. Admin

işlemi tamamlandığında semaphore serbest bırakılır, böylece diğer işlemler devam edebilir. Onaylanmış siparişler, ürünlere göre gruplandırılır ve her ürün üzerinde işlem yapmak için bir mutex mekanizması kullanılır. Mutex, aynı ürün üzerinde aynı anda birden fazla iş parçacığının işlem yapmasını engeller. Benzer şekilde, müşteri verileri üzerinde yapılan işlemler de mutex ile korunur, böylece bir müşterinin verileri eş zamanlı olarak birden fazla iş parçacığı tarafından değiştirilemez. Sipariş işleme sürecinde stok ve bütçe kontrolleri yapılır. Yetersiz stok veya bütçe durumunda, sipariş iptal edilir ve bu durum loglanır. İşlemi başarılı olan siparişlerde, stok miktarı düşürülür, müşteri bütçesi güncellenir ve toplam harcama bilgisi artırılır. Ayrıca, toplam harcaması 2000 TL'yi geçen standart müşteriler, premium müşteri türüne yükseltir. Sistem, her sipariş sonrası stok miktarını ve müşteri bütçesini güncelleyerek verilerin tutarlılığını sağlar. Yetersiz stok veya bütçe gibi durumlar anında tespit edilip loglanır. Admin tarafından gerçekleştirilen stok güncelleme ve ürün silme işlemleri sırasında diğer sipariş işlemleri bekletilir ve admin işlemi tamamlandıktan sonra devam ettirilir. Sistem, her işlem için detaylı bir log kaydı oluşturur. Bu loglar, işlemin türünü, müşteri bilgilerini, ürün detaylarını ve işlem sonucunu içerir. Başarılı işlemler "Bilgilendirme" olarak loglanırken, yetersiz stok, yetersiz bütçe veya admin müdahalesi gibi durumlar "Uyarı" logları olarak kaydedilir. Bu sistem, eş zamanlı iş parçacıklarının kaynaklara güvenli bir şekilde erişimini sağlamak, stok ve müşteri bütçesi üzerinde tutarlı işlemler gerçekleştirmek ve admin müdahalelerini etkin bir şekilde yönetmek için tasarlanmıştır. Proje, multithreading ve senkronizasyon tekniklerinin başarılı bir uygulamasını sunmaktadır.

#### IV. DENEYSEL SONUÇLAR

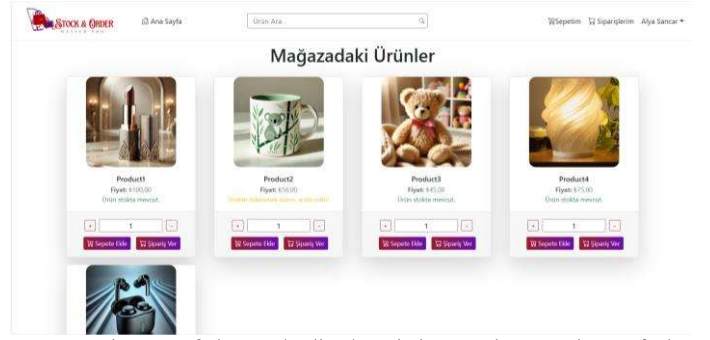
Bu bölümde, projenin genel görünümü ve işlevleri hakkında bilgi verilmektedir. Projenin yapısı ve gerçekleştirilen işlevler detaylı bir şekilde sunulmuştur.

##### A. Müşteri Giriş Sayfası



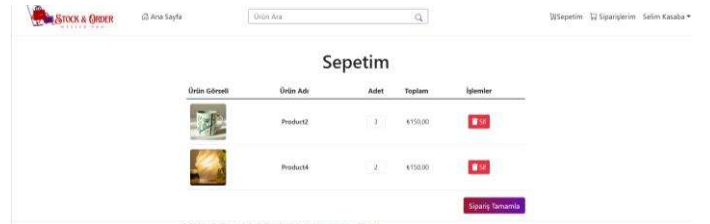
Müşteri Giriş Sayfası

##### B. Customer Ana Sayfa



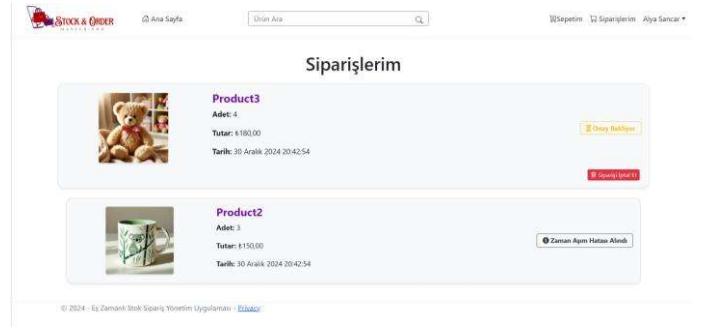
Müşteri ana sayfa ürünler listelenmiştir. Buradan sepetine en fazla 5 adet olacak şekilde ürün sepete ekleyip sipariş oluşturabilir.

##### C. Customer Sepetim Paneli



Müşteri sepete eklediği ürünleri burada görüntüleyebilir, sepetten çıkartabilir veya siparişini oluşturabilir.

##### D. Customer Siparişlerim Paneli



Müşteri tüm verdiği siparişleri bu panelde görüntüleyebilmektedir. Tamamlanan veya onay beklemede olan siparişlerini silebilmektedir.

## E. Admin Müşteri Paneli

| Kayıtlı Müşteriler |                |                         |          |         |              |               |
|--------------------|----------------|-------------------------|----------|---------|--------------|---------------|
| Customer ID        | Müşteri Adı    | Email                   | Adress   | Bütçe   | Müşteri Türü | Total Harcama |
| 3073               | Aliya Sancar   | aliyasancar@gmail.com   | Trabzon  | 2429.00 | Premium      | \$2.429.00    |
| 3074               | Aren Durak     | arendurak@gmail.com     | İstanbul | 1105.00 | Premium      | \$3.380.00    |
| 3075               | Ela Durak      | eladurak@gmail.com      | Rize     | 1980.00 | Standart     | \$67.00       |
| 3076               | Selma Kasaba   | selmakasaba@gmail.com   | Sakarya  | 1278.00 | Standart     | \$1.657.00    |
| 3077               | Selma Sancar   | selmasancar@gmail.com   | İstanbul | 2808.00 | Standart     | \$519.00      |
| 3078               | Arda Kurt      | ardakurt@gmail.com      | Rize     | 1362.00 | Standart     | \$1.330.00    |
| 3079               | Arda Albayrak  | ardasalbayrak@gmail.com | İstanbul | 922.00  | Premium      | \$3.471.00    |
| 3080               | Ela Turan      | elaturan@gmail.com      | Trabzon  | 1825.00 | Premium      | \$2.438.00    |
| 3081               | Arda Kasaba    | ardakasaba@gmail.com    | Ankara   | 2181.00 | Standart     | \$20.00       |
| 3082               | Zehra Albayrak | zehraalbayrak@gmail.com | Rize     | 932.00  | Standart     | \$428.00      |

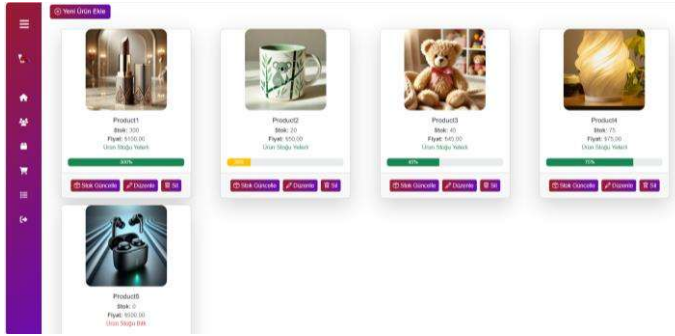
Admin müşteri panelinde müşterilerin bilgilerini görüntüleyebilir.

## KAYNAKLAR

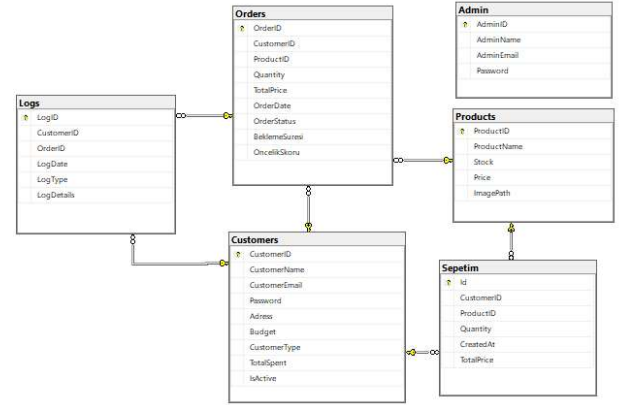
- [1] Murat Yücedağ, Asp.Net Core Kampı
- [2] Youtube
- [3] JWT.io. (n.d.). JSON Web Token (JWT) Introduction. Erişim adresi: <https://jwt.io/>
- [4] Atasoy Web. (n.d.). Çok Parçacıklılık (Multithreading), Mutex'ler ve Semaforlar. Erişim adresi: <https://www.atasoyweb.net/Cok-Parcaciklilik-Multithreading-Mutexler-Ve-Semaforlar>
- [5] Stack Overflow

## ER Diyagramları

## F. Admin Ürün Listesi, Ekleme, Güncelleme Paneli



Admin bu panelde ürünleri görüntüleyebilir, güncelleyebilir, silebilir ve siteye yeni ürün ekleyebilmektedir.



## G. Admin Log Paneli

| Log Paneli                 |               |  |
|----------------------------|---------------|--|
| Tarih                      | Tür           | Detay  |
| 30.12.2024<br>21:51:08.627 | Bilgilendirme | Müşteri Arda Kurt (Premium) Product3'ten 1 adet sipariş tamamladı.   |
| 30.12.2024<br>21:51:08.623 | Bilgilendirme | Müşteri Arda Kurt (Premium) Product3'ten 1 adet sipariş sonrası Toplam harcama: 2180.00 Bütçe:512.00.                                  |
| 30.12.2024<br>21:51:08.620 | Bilgilendirme | Müşteri Arda Kurt (Premium) Product3'ten 1 adet sipariş sonrası stok güncellendi. Yeni stok:44   |
| 30.12.2024<br>21:51:08.613 | Bilgilendirme | Müşteri Arda Kurt (Premium) Product3'ten 1 adet sipariş işleme alındı.   |
| 30.12.2024<br>21:51:07.597 | Uyarı         | Müşteri Aren Durak (Premium) Product3'ten 1 adet sipariş yetersiz stok nedeniyle iptal edildi. Stok adedi: 0.                          |
| 30.12.2024<br>21:51:07.597 | Bilgilendirme | Müşteri Selma Kasaba (Premium) Product3'ten 2 adet sipariş yetersiz bütçe nedeniyle iptal edildi. Toplam Tutar: 880.00, Bütçe: 528.00. |
| 30.12.2024<br>21:51:07.583 | Bilgilendirme | Müşteri Aren Durak (Premium) Product3'ten 1 adet sipariş işleme alındı.  |
| 30.12.2024<br>21:51:07.570 | Bilgilendirme | Müşteri Selma Kasaba (Premium) Product3'ten 2 adet sipariş işleme alındı.  |
| 30.12.2024<br>21:51:06.377 | Bilgilendirme | Müşteri Aren Durak (Premium) Product3'ten 1 adet sipariş sonrası Toplam harcama: 4360.00 Bütçe:125.00.                                 |
| 30.12.2024<br>21:51:06.377 | Bilgilendirme | Müşteri Arda Kurt (Premium) Product3'ten 1 adet sipariş sonrası Toplam harcama: 2135.00 Bütçe:557.00.                                  |

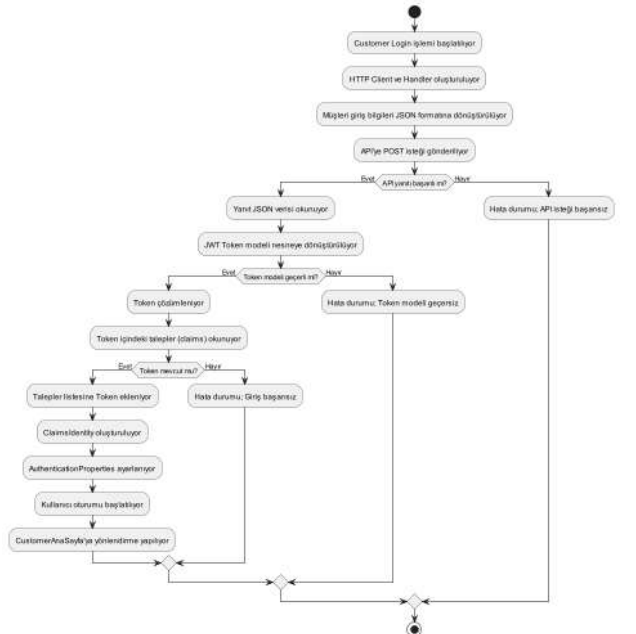
Adminin görüntülediği sipariş log paneli. Fetch Api ile belli aralıklarla veri güncelleme işlemi yaparak animasyon kurgulanmıştır.

## SONUÇ

Bu proje, eş zamanlı sipariş ve stok yönetimi sistemi olarak tasarlanmış ve uygulanmıştır. Müşteri ve admin işlemleri multithreading, mutex ve semafor mekanizmaları kullanılarak eş zamanlı bir şekilde gerçekleştirmek amaçlanmış olup gerçekleştirilmeye çalışılmıştır. Premium müşteriler öncelikli olarak işlenmiş, standart müşteriler ise bekleme sürelerine bağlı olarak sıralamada yükselmiştir. Stok ve bütçe yönetimi dinamik olarak kontrol edilmiş, yetersiz durumlarda işlemler iptal edilerek loglanmıştır. Sistem, eş zamanlı işlem yönetiminde etkin bir çözüm sunmaya olanak tanımaktadır.

Şekil.1  
Projenin ER Diyagramı 6 adet tablodan oluşmaktadır.  
(Admin, Orders, Products, Logs, Sepetim, Customers)

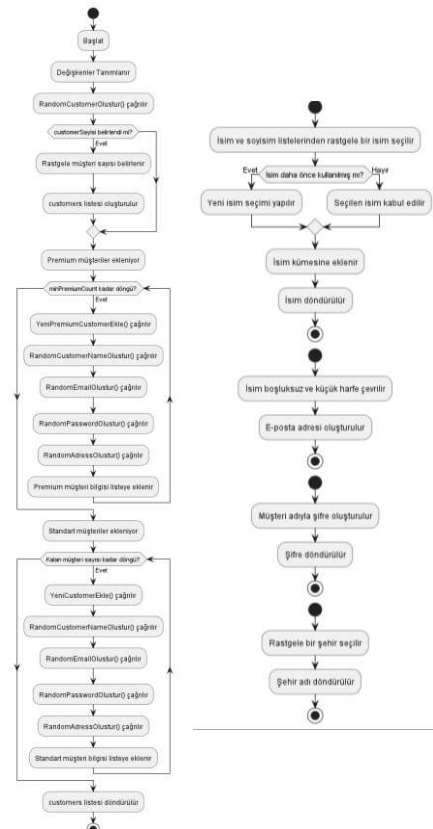
## Akış Diyagramları



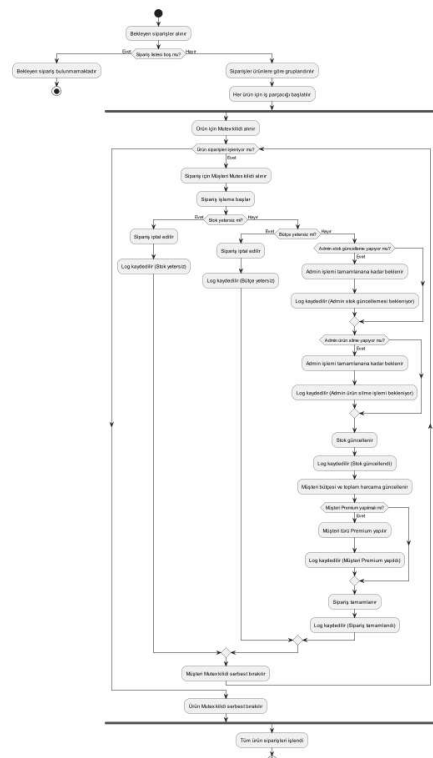
Şekil.2  
Customer Giriş Akış Diyagramı



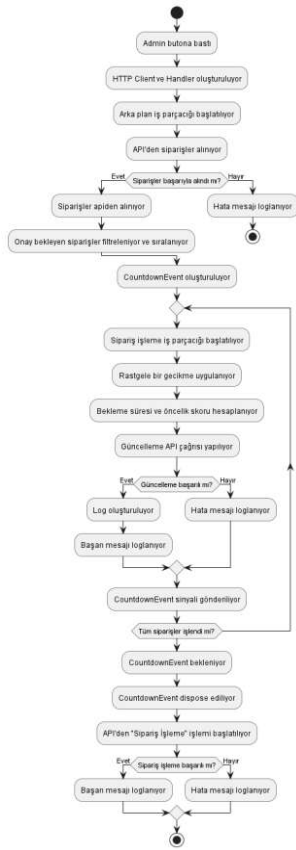
Şekil.3  
Customer sipariş oluşturma diyagramı



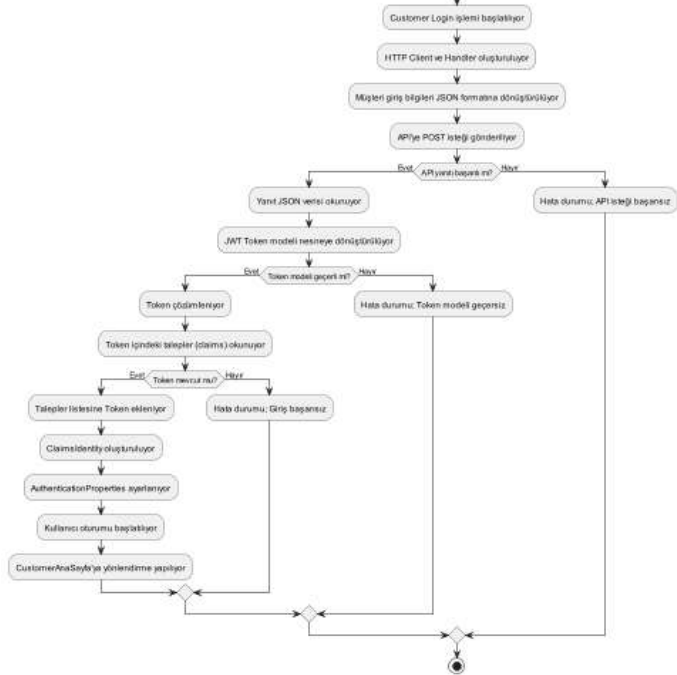
Şekil.4  
Random Customer Oluşturma



Şekil.5  
Müşteri Sipariş İşleme



Şekil.5  
Admin Onaylama İşlemi



Şekil.6  
Giriş Algoritma Akışı