

Yazılım Lab.-1

3. Proje

Anıl Engin Keretli
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
Kocaeli, Türkiye
230201128

Ahmet Burak Karkaç
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
Kocaeli, Türkiye
220201173

I. GİRİŞ

A. Eş Zamanlı Sipariş ve Stok Yönetimi Platformu

Dijital dönüşüm çağında, stok ve sipariş yönetimi gibi kritik süreçlerin otomasyonu ve optimizasyonu, işletmelerin verimliliğini artıran önemli bir faktör haline gelmiştir. Geleneksel yöntemlerle sipariş ve stok yönetimi, genellikle zaman alıcı, hatalara açık ve sınırlı verimlilik sunarken, modern dijital platformlar, eş zamanlı veri işleme ve kullanıcı dostu çözümler sunarak bu alanda devrim yaratmaktadır.

Bu bağlamda, Eş Zamanlı Sipariş ve Stok Yönetimi Platformu, işletmelerin stoklarını etkin bir şekilde yönetmelerini ve müşterilerin sipariş süreçlerini hızlı ve güvenilir bir şekilde tamamlamalarını sağlayan yenilikçi bir çözüm olarak tasarlanmıştır. Proje, multithreading, senkronizasyon mekanizmaları ve dinamik öncelik algoritmaları gibi modern yazılım prensiplerini entegre ederek kullanıcı deneyimini en üst düzeye çıkarmayı hedeflemektedir.

B. Temel Özellikler ve Yenilikçi Çözümler:

1) *Müşteri ve Sipariş Yönetimi:* Platform, müşteri türüne (Premium ve Normal) göre dinamik öncelik sıralaması yaparak eş zamanlı sipariş işlemlerini optimize etmektedir. Müşteriler, rastgele belirlenen bütçe ve özelliklerle sisteme eklenmekte, işlemler sırasında bütçe ve stok yeterliliği gibi kriterler otomatik olarak kontrol edilmektedir. Dinamik Öncelik Sistemi sayesinde, bekleme süresi ve müşteri türüne göre işlem önceliği sürekli güncellenmekte, bu da müşteri memnuniyetini artırmaktadır.

2) *Stok Yönetimi:* Stok miktarlarının anlık olarak güncellenmesi, aynı ürüne eş zamanlı erişim taleplerini yöneten güçlü senkronizasyon mekanizmalarıyla desteklenmiştir. Kritik stok seviyelerinde görsel ve metinsel uyarılar sunulurken etkin stok takibi sağlanmaktadır.

3) *Admin Paneli:* Ürün ekleme, silme ve stok düzenleme işlemleri, müşteri işlemleriyle paralel bir şekilde gerçekleştirilebilmektedir. Admin, sistemin genel işleyişini gerçek zamanlı loglama sistemiyle takip ederek aksiyon alabilir.

4) *Loglama ve İzleme:* Sistem, tüm işlemleri kayıt altına alarak hata, uyarı ve bilgi mesajlarını detaylı bir şekilde raporlamaktadır. Örneğin: "Ürün stoğu yetersiz" veya "Müşteri bakiyesi yetersiz" gibi hata mesajları. "Müşteri X, Y ürününden Z adet aldı. İşlem başarılı." gibi bilgi mesajları.

5) *Kullanıcı Arayüzü:* Müşteri paneli, kullanıcıların işlem durumlarını renklendirme ve animasyonlarla görselleştiren bir yapıya sahiptir. Stok durumları bar veya dairesel grafiklerle takip edilebilir. Gerçek zamanlı loglar ve dinamik müşteri sırası ile işlemler eş zamanlı izlenebilmektedir. Sonuç ve Hedefler: Bu proje, stok ve sipariş yönetiminde eş zamanlılık, dinamik öncelik ve güvenilirlik sağlayarak işletmelerin operasyonel süreçlerini daha düzenli ve verimli hale getirmektedir. Ayrıca, hem bireysel kullanıcılar hem de sistem yöneticileri için kapsamlı bir çözüm sunmaktadır. Multithreading, senkronizasyon, veritabanı tasarımı ve kullanıcı dostu arayüz gibi unsurların bir arada ele alındığı bu platform, hem teknik bilgi hem de uygulama becerileri açısından değerli bir deneyim sunmaktadır.

II. ÖZET

Bu projede, işletmelerin sipariş ve stok yönetim süreçlerini optimize etmek amacıyla **Eş Zamanlı Sipariş ve Stok Yönetimi Platformu** geliştirilmiştir. Platform, müşterilerin sipariş işlemlerini gerçekleştirmesine, stok durumunu kontrol etmesine ve yöneticilerin ürün yönetimi süreçlerini kolaylaştırmasına olanak tanıyan kapsamlı özelliklere sahiptir.

Proje, kullanıcı dostu bir arayüz ve güçlü bir teknik altyapıyla, modern yazılım teknolojilerini bir araya getirmiştir. Dinamik öncelik sıralaması, eş zamanlı veri işleme mekanizmaları, stok çakışma kontrol algoritmaları ve gerçek zamanlı loglama özellikleri platformun verimliliğini artırmaktadır. Ayrıca, müşterilerin işlemlerini öncelik sırasına göre düzenleyen akıllı algoritmalar ve görselleştirme teknikleri ile kullanıcı deneyimi iyileştirilmiştir.

Yöneticiler için ise admin paneli üzerinden ürün ekleme, stok yönetimi ve işlem takibi süreçleri kolaylaştırılmıştır. Platform, aynı zamanda kullanıcı işlemleriyle paralel çalışan güvenilir senkronizasyon mekanizmaları ile desteklenmiştir. Sistem, hem bireysel kullanıcılar hem de yöneticiler için etkili ve güçlü bir çözüm sunmayı başarmıştır.

Sonuç olarak, bu proje, stok ve sipariş yönetimini daha düzenli, verimli ve güvenilir bir hale getirerek yenilikçi bir dijital çözüm sunmayı hedeflemiş ve bu doğrultuda başarılı bir sistem geliştirilmiştir. Tasarlanan platform, işletmelerin operasyonel süreçlerini optimize eden ve kullanıcı memnuniyetini artıran bir teknoloji ürünü olarak öne çıkmaktadır.

III. PROJE TANIMI

Bu proje, işletmelerin sipariş ve stok yönetim süreçlerini daha verimli, düzenli ve güvenilir bir şekilde yönetmelerini sağlamak için Eş Zamanlı Sipariş ve Stok Yönetimi Platformu geliştirmeyi amaçlamaktadır. Platform, müşterilerin sipariş vermesini, stokların güncellenmesini ve yöneticilerin ürün yönetimini kolaylaştıracak şekilde tasarlanmış, masaüstü veya web tabanlı bir sistemdir.

Proje, müşteri türlerine göre (Premium ve Normal) dinamik öncelik sıralaması, eş zamanlı veri işleme ve çakışma kontrolü gibi modern teknolojik çözümleri bir araya getirmektedir. Kullanıcılar, sipariş işlemleri sırasında bütçe ve stok yeterliliği gibi kriterlerin otomatik olarak kontrol edildiği bir sistemde, hızlı ve güvenli işlemler gerçekleştirebilir. Ayrıca, stok durumunun grafiklerle görselleştirilmesi, kritik seviyeler için uyarılar ve işlem loglarının gerçek zamanlı olarak kaydedilmesi gibi özelliklerle kullanıcı deneyimi zenginleştirilmiştir.

Platform, sadece müşteriler için değil, yöneticiler için de kapsamlı araçlar sunmaktadır. Admin paneli, ürün ekleme, silme, stok düzenleme ve işlem yönetimini mümkün kılarken, loglama ve izleme sistemleriyle sistemin işleyişi üzerinde tam kontrol sağlar. Güçlü bir teknik altyapı ve eş zamanlı veri yönetimi sayesinde, işlemler güvenilir bir şekilde senkronize edilmekte ve işletmelerin operasyonel süreçleri optimize edilmektedir.

Bu proje, eş zamanlı sipariş ve stok yönetimi süreçlerinde modern teknolojilerin sunduğu yeniliklerden yararlanmayı, müşteri memnuniyetini artırmayı ve işletmelerin operasyonel verimliliğini desteklemeyi hedeflemektedir. Teknik altyapısı ve kullanıcı dostu tasarımıyla, hem bireysel kullanıcılar hem de yöneticiler için güçlü bir çözüm sunmayı başarmaktadır.

ARAŞTIRMALAR VE YÖNTEM

Proje kapsamında eş zamanlı sipariş ve stok yönetimi süreçlerinde modern yazılım teknolojilerinin sunduğu avantajlar araştırılmış ve etkin bir platform geliştirmek için kapsamlı bir tasarım süreci yürütülmüştür.

İlk olarak, mevcut stok ve sipariş yönetim sistemleri (örneğin, SAP, Oracle NetSuite) analiz edilerek güçlü ve zayıf yönler belirlenmiştir. Bu analiz, işletmelerin ihtiyaçlarını anlamak, kullanıcı deneyimindeki zorlukları tespit etmek ve projenin temel hedeflerini netleştirmek amacıyla yapılmıştır. Kullanıcı beklentilerine dair yapılan anket ve görüşmelerle, sistemden beklenen fonksiyonlar, öncelikli ihtiyaçlar ve zorluklar ortaya çıkarılmıştır.

Teknolojik araştırmalar, multithreading, senkronizasyon, dinamik öncelik algoritmaları ve veritabanı tasarımı gibi temel konulara odaklanmıştır. Java, Python ve C gibi programlama dillerinin yanı sıra MSSQL, MySQL ve MongoDB gibi veritabanı yönetim sistemlerinin performans ve

ölçeklenebilirlik açısından değerlendirilmesi yapılmıştır. Dinamik öncelik sıralaması ve çakışma kontrolü algoritmalarının uygulanabilirliği detaylı bir şekilde incelenmiştir.

Yöntem: Projenin geliştirilmesinde Agile metodolojisi benimsenmiştir. Geliştirme süreci, küçük ve yönetilebilir sprintlere ayrılmış, her sprint sonunda belirli bir modül test edilerek geri bildirimler doğrultusunda iyileştirmeler yapılmıştır.

1) *Veritabanı Tasarımı*:: Kullanıcılar, ürünler, siparişler ve loglar gibi temel verileri içeren ilişkisel bir veritabanı yapısı oluşturulmuştur. Veritabanı tasarımında normalizasyon kurallarına uyularak verilerin tutarlılığı sağlanmıştır. Tablo ilişkileri, siparişlerin ve stok yönetiminin bütüncül bir yapıda ele alınmasına olanak tanıyacak şekilde modellenmiştir.

2) *Multithreading ve Dinamik Öncelik Sistemi*:: Müşterilerden gelen siparişler, premium ve normal müşteri ayrımı ile dinamik öncelik sıralaması kullanılarak işleme alınmıştır. Öncelik sırası, müşteri türü, bekleme süresi ve işlem süresine göre belirlenmiş ve sürekli güncellenmiştir. Bu işlemler, multithreading ve senkronizasyon mekanizmalarıyla desteklenerek çakışmaların önüne geçilmiştir.

3) *Algoritma Geliştirme*:: Stok çakışma kontrol algoritmaları, aynı ürünü satın almak isteyen birden fazla müşterinin işlemlerini yönetmek için tasarlanmıştır. Stok durumu ve müşteri bütçesi kontrolü işlemlerin temelini oluşturmuştur. Kritik seviyedeki stoklar için uyarılar içeren görselleştirme araçları geliştirilmiştir.

4) *Kullanıcı Arayüzü ve Loglama*:: Müşteri ve admin panelleri, kullanıcı dostu bir deneyim sağlamak için detaylı bir şekilde tasarlanmıştır. Gerçek zamanlı loglama sistemi ile işlem sonuçları ve hata mesajları kaydedilmiş ve görsel bir formatta kullanıcıya sunulmuştur.

5) *Test Süreçleri*:: Platformun işlevselliğini ve güvenilirliğini sağlamak için manuel ve otomatik test araçları kullanılmıştır. Multithreading mekanizmalarının düzgün çalıştığı, algoritmaların çakışmaları doğru bir şekilde yönettiği ve kullanıcı arayüzünün beklentilere uygun olduğu test edilmiştir.

Bu araştırmalar ve yöntemler, sistemin hem teknik açıdan güçlü hem de kullanıcı dostu bir yapıya sahip olmasını sağlamıştır. Böylece işletmelerin operasyonel süreçlerini optimize edecek ve müşteri memnuniyetini artıracak bir platform ortaya çıkarılmıştır.

IV. GELİŞTİRME ORTAMI

Bu proje java programlama dili ile yazılmıştır
Veri tabanı olarak mongodb kullanıldı
İntellij idea idesi kullanıldı

V. DENEYSEL SONUÇLAR

Proje kapsamında geliştirilen Eş Zamanlı Sipariş ve Stok Yönetimi Platformunun performansını ve işlevselliğini değerlendirmek amacıyla çeşitli testler ve deneyler gerçekleştirilmiştir. Bu testler, sistemin teknik altyapısını, kullanıcı gereksinimlerini karşılama düzeyini ve işlem

süreçlerindeki verimliliği analiz etmeyi hedeflemiştir. Deneyisel sonuçlar şu şekilde özetlenmiştir:

1) *Kullanıcı Deneyimi Testleri::* Platformun kullanıcı arayüzü, gerçek kullanıcılar tarafından test edilmiş ve genel olarak olumlu geri bildirimler alınmıştır. Kullanıcılar, müşteri panelindeki sipariş oluşturma ve stok görüntüleme özelliklerini anlaşılır ve kullanımı kolay bulmuştur. Bekleme sırasını ve işlem durumunu görselleştiren animasyonlar, kullanıcı deneyimini zenginleştiren unsurlar arasında öne çıkmıştır.

2) *Dinamik Öncelik Sistemi Performansı::* Dinamik öncelik sıralama algoritması, müşterilerin türü (Premium/Normal), bekleme süresi ve işlem süresi gibi kriterlere dayalı olarak doğru bir şekilde çalışmıştır. Algoritma, özellikle Premium müşterilere öncelik sağlarken, Normal müşteriler için bekleme süresi arttıkça öncelik seviyesinin doğru bir şekilde yükseltildiği gözlemlenmiştir. Bu durum, işlemlerin adil ve verimli bir şekilde gerçekleştirilmesine olanak tanımıştır.

3) *Stok Çakışma Algoritması::* Stok çakışma kontrol algoritması, aynı ürüne birden fazla müşteri tarafından eş zamanlı talep olduğunda, işlemleri başarıyla yönetmiştir. Algoritmanın stok yetersizliği durumunda ilgili işlemi doğru bir şekilde reddettiği ve kullanıcıya uyarı mesajı verdiği testlerle doğrulanmıştır. Kritik stok seviyeleri için grafik temelli görsel uyarılar da sorunsuz çalışmıştır.

4) *Veritabanı Performansı::* Veritabanı işlemleri yüksek hızda gerçekleştirilmiş ve testlerde herhangi bir performans problemiyle karşılaşılması. İlişkisel veritabanı yapısı, tüm işlemlerin doğru ve güvenilir bir şekilde kaydedilmesini sağlamıştır. Verilerin bütünlüğü ve sistemin ölçeklenebilirliği test edilerek güvenilir bir yapı oluşturulmuştur.

5) *Loglama ve İzleme::* Gerçek zamanlı loglama sistemi, hem müşteri hem de admin işlemlerine ait detaylı kayıtlar sağlamıştır. Hatalar, uyarılar ve bilgi mesajları test edilerek sistemin izlenebilirliği doğrulanmıştır. Örneğin, "Ürün stoğu yetersiz" veya "Müşteri bakiyesi yetersiz" gibi hata mesajlarının doğru bir şekilde oluşturulduğu gözlemlenmiştir.

6) *Multithreading ve Senkronizasyon Testleri::* Multithreading mekanizmaları, aynı anda birçok işlemi güvenilir bir şekilde yönetmiştir. Admin ve müşteri işlemleri arasında senkronizasyon sağlanarak işlem çakışmalarının önüne geçilmiştir. Mutex ve semaforlar kullanılarak kaynaklara eş zamanlı erişim sorunsuz bir şekilde kontrol edilmiştir.

VI. SONUÇ

Bu proje kapsamında geliştirilen Eş Zamanlı Sipariş ve Stok Yönetimi Platformu, işletmelerin sipariş ve stok süreçlerini verimli bir şekilde yönetmelerine olanak sağlayan kapsamlı bir sistem olarak tasarlanmıştır. Proje, dinamik öncelik sıralaması, multithreading ve senkronizasyon mekanizmaları, stok çakışma kontrol algoritmaları ve kullanıcı dostu bir arayüz gibi özelliklerle hem kullanıcı deneyimini hem de sistem performansını başarıyla optimize etmiştir.

Proje sürecinde, müşteri ve stok yönetimi için geliştirilen algoritmalar, ilişkisel veritabanı tasarımı ve kullanıcı arayüzü oluşturma konularında detaylı çalışmalar gerçekleştirilmiştir

ve bu alanlarda önemli kazanımlar elde edilmiştir. Veritabanı tasarımında, verilerin doğru bir şekilde ilişkilendirilmesi ve veri çağırma süreçlerinin hızlandırılması sağlanmıştır. Özellikle dinamik öncelik algoritması, müşterilerin türüne ve işlem süresine göre adil ve etkin bir işlem sırası sunarak sistemin işlevselliğini artırmıştır.

Platformun kullanıcı dostu yapısı ve sunduğu özellikler, test kullanıcıları tarafından olumlu geri bildirimlerle değerlendirilmiştir. Kullanıcılar, gerçek zamanlı loglama, işlem sırası görselleştirmesi ve stok durumu takibi gibi özelliklerin sistemi daha işlevsel ve kolay kullanılabilir hale getirdiğini belirtmiştir. Geri bildirimler doğrultusunda yapılan iyileştirmeler, sistemin daha sağlam bir teknik altyapıya ve kullanıcı memnuniyetini artıran bir yapıya kavuşmasını sağlamıştır.

Sonuç olarak, **Eş Zamanlı Sipariş ve Stok Yönetimi Platformu**, belirlenen hedeflere ulaşmış ve işletmelerin operasyonel süreçlerini optimize etmek için güçlü bir dijital çözüm sunmayı başarmıştır. Bu proje, gelecekte yapılacak geliştirmelerle daha geniş bir kullanıcı kitlesine hitap edebilecek ve benzer sistemler için referans teşkil edebilecek bir altyapı sunmaktadır. Proje, yazılım geliştirme süreçlerinde kazanılan deneyimlerle birlikte, teknolojinin işletme süreçlerine entegre edilmesinin başarılı bir örneğini temsil etmektedir.

VII. YALANCI KOD

A. Müşteri Kayıt Süreci

- 1) *Kullanıcı, "Kayıt Ol" ekranını açar:*
- 2) *Kayıt formuna müşteri adı, bütçe ve müşteri türü (Premium/Normal) gibi bilgileri girer:*
- 3) *Sistem bilgileri doğrular::* Zorunlu alanlar eksiksiz mi? Premium müşterilerin sayısı minimum 2 mi?
- 4) *Doğrulama başarılıysa::* Müşteri bilgileri veritabanına kaydedilir. Kullanıcıya "Kayıt başarılı" mesajı gösterilir.
- 5) *Doğrulama başarısızsa::* Kullanıcıya hata mesajı gösterilir. Kullanıcı bilgilerini düzenleyip tekrar dener.

VIII. SİPARİŞ OLUŞTURMA SÜRECİ

- 1) *Kullanıcı, sipariş formuna müşteri ID, ürün adı ve miktar bilgilerini girer:*
- 2) *Sistem bilgileri doğrular::* Müşteri bakiyesi yeterli mi? Ürün stok durumu yeterli mi?
- 3) *Doğrulama başarılıysa::* Sipariş, veritabanına kaydedilir. Stok miktarı güncellenir. Kullanıcıya "Sipariş başarılı" mesajı gösterilir.
- 4) *Doğrulama başarısızsa::* Kullanıcıya hata mesajı gösterilir (örneğin, "Stok yetersiz" veya "Bakiye yetersiz").

A. Dinamik Öncelik Sıralaması

- 1) *Sistem, tüm aktif siparişleri sıraya koyar:*

2) Her müşterinin öncelik skoru hesaplanır::
 $\text{ÖncelikSkoru} = \text{TemelÖncelikSkoru} + (\text{BeklemeSüresi} \times \text{BeklemeSüresiAğırlığı})$
 Premium müşteriler için $\text{TemelÖncelikSkoru} = 15$, Normal müşteriler için $= 10$.
 Bekleme Süresi Ağırlığı $= 0.5$.

3) Siparişler, öncelik skoruna göre işlenir.: —

B. Stok Güncelleme Süreci

1) Admin, stok yönetim panelini açar.:

2) Güncellemek istediği ürünü seçer ve yeni stok miktarını girer.:

3) Sistem bilgileri doğrular.:: Yeni stok miktarı geçerli bir değer mi?

4) Doğrulama başarılıysa.:: Stok bilgileri veritabanında güncellenir. Admin'e "Stok başarıyla güncellendi" mesajı gösterilir.

5) Doğrulama başarısızsa.:: Admin'e hata mesajı gösterilir.

C. Zaman Çakışması Kontrolü

1) Bir müşteri yeni bir sipariş oluşturduğunda, sistem diğer müşterilerin aktif işlemlerini kontrol eder.:

2) Aynı ürüne yönelik eş zamanlı talepler karşılaştırılır.:

3) Eğer çakışma varsa.:: Daha yüksek öncelikli müşteri işlemi gerçekleştirilir. Diğer işlemler bekleme sırasına alınır.

4) Çakışma yoksa, işlem devam eder.: —

D. Loglama Süreci

1) Her işlem başladığında, sistem bir log kaydı oluşturur.:

2) Log, şu bilgileri içerir.:: İşlem türü (sipariş, stok güncelleme vb.). Müşteri ID. Siparişin başarı durumu (başarılı/başarısız).

3) Log kaydı veritabanına eklenir.:

4) Gerçek zamanlı log panelinde kullanıcıya gösterilir.:

E. Admin Yönetim Süreci

1) Admin, platformun yönetim paneline giriş yapar.:

2) Admin şu işlemleri yapabilir.:: Yeni ürün ekleme, mevcut ürünleri silme veya düzenleme. Kullanıcıları silme veya düzenleme.

3) İşlem tamamlandığında sistem log kaydı oluşturur ve admin'e onay mesajı gösterir.:

F. Raporlama ve Geri Bildirim Süreci

1) Admin, logları ve işlem geçmişini analiz eder.:

2) Kullanıcı geri bildirimlerini toplar.:

3) Veriler, iyileştirme ve güncellemeler için kullanılır.:

IX. GÖRSELLER

X. REFERANSLAR

REFERENCES

- [1] <https://www.udemy.com/course/c-sharp-programlama-dili-her-seviyeye-hitap-eden-egitim-seti/learn/lecture/24327176overviewC#> Programlama Dili Eğitim Seti, Udemey
- [2]
- [3] <https://www.udemy.com/course/sifirdan-ileri-seviye-csharp-programlama/learn/lecture/8388470?start=120overviewIleriC#> Programlama, Udemey

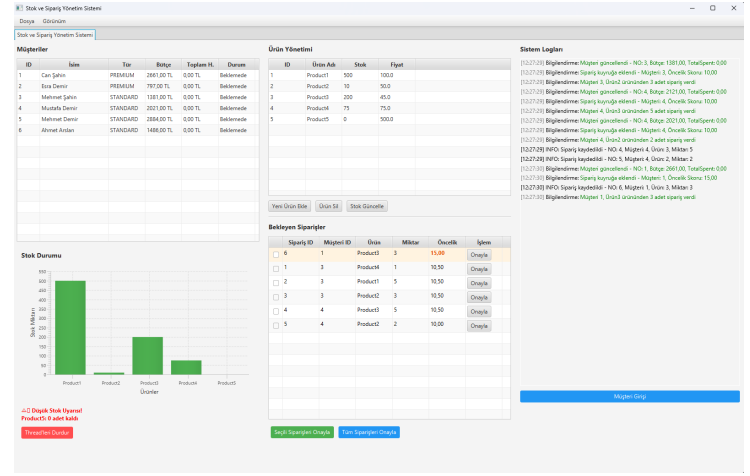


Fig 1. Arayüz

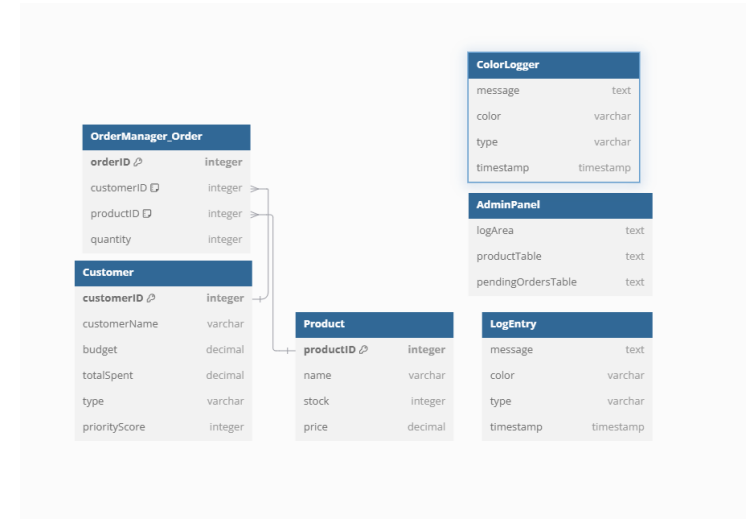


Fig. 2. ER diyagramı

_id: 1 name: "Product1" stock: 500 price: 100
_id: 2 name: "Product2" stock: 10 price: 50
_id: 3 name: "Product3" stock: 200 price: 45
_id: 4 name: "Product4" stock: 75 price: 75
_id: 5 name: "Product5" stock: 0 price: 500

Fig. 3. Loglar