



Stok Takip Otomasyonu

**ATA BERKAY
KARAKUS**

“KARAKUSLAR MARKET”



- **Giriş**

Günümüz iş dünyasında bulunan market sahiplerinin kullanabileceği minimalist bir “stok takip otomasyonu” yapmaya odaklanılmıştır.

- **Çözümleme**

Şirketlerin iş süreçleri, stok yönetimi uygulamaları ve mevcut sistemleri hakkında kısa bir araştırma yapıp bu şirketler hakkında bilgiler toplandı. Toplanan bilgiler ile ilgili birimler bir araya gelip toplantılar yapıldı.

Karakuslar Market’in Personeli ile iletişime geçildi ve ihtiyaçları belirlemeye yönelik toplantılar yapıldı.

Stok takip programlarının açıkları ve hataları bulunup düzeltmeler yapıldı.

Karakuslar Market’in stok takip programında neleri beklediği konuşuldu verilen bilgiler doğrultusunda düzeltmeler yapıldı.

Kullanıcıların optimize ve entegre bir şekilde uygulamayı kullanabilmesi için görüşmeler, anketler, toplantılar ve yorumları bir araya getirilerek analiz edildi.

Proje yapılırken oluşabilecek riskler belirlenilmeye çalışıldı.

Risklerin oluşturabilecek mali zararları belirlendi.

- **Tasarım**

Projeye başlamadan önce tasarım için hangi işlevleri yerine getireceği, hangi verileri saklayacağını ve kimlere hizmet verileceği analiz edilip belirlendi.

Akıllardaki soru işaretlerini giderebilmek için geçmiş yıllarda yapılmış olan örnek tasarımlar incelendi ve bu tasarımlar dahilinde yapılmış olan yanlışlar ve hatalar belirlendi.

Belirlenen yanlışlar ve hatalar dahilinde prototipler oluşturup kâğıt üzerinde çizimler ve prototipler yapılmaya çalışıldı.

Kendi içimde müzakereler yapıp yapılan prototiplerin müşteri tarafından istenilen gereksinimler karşılayıp karşılamadığı ve tasarımın uygulanabilir olup olmadığı konuşuldu.

- **Gerçekleşme**

Yapmış olduğumuz tasarımı yazılım alanında kullanabilmek için JAVA dili seçildi ve bu programlama dili ile tasarıma uygun, entegre ve optimize olmuş bir şekilde yapıldı.

Nesne Tabanlı Programlama standartlarına uygun olarak kodlama süreçleri izlenilip yapıldı.

- **Sınama**

Yapılan yazılımın hatalarını azaltıp maliyeti düşürüldü.

Yazılımın doğru bir şekilde çalışıp, istenilen işlemleri yerine getirip getirilmediği kontrol edildi.

Yazılım performans testine tabii tutuldu.

Performans Testi ile yazılımın belirli bir yük altında nasıl bir performans gösterdiğini değerlendirildi.

- **Bakım**

Teslim ettiğimiz projenin uzun yıllar boyunca kullanılabilmesi için düzenli olarak müşteri dönütleri ciddiye alındı ve bu dönütlere göre güncellemeler yapıp programın optimize çalışması sağlandı.

Performans iyileştirmeleri yapıldı.

Müşteri memnuniyeti artırılmaya çalışıldı.

Kullanıcı dönütlerini ciddiye alarak projede yapılan işlevleri güncelledik ve yenilerini ekledik

- **Bitiş**

Stok takip otomasyonu bütün işlevleri yerine getirdiği kontrol edildi ve müşterimiz “Karakuslar Market” sahiplerine teslim edildi.

Şelale Modeli

Projemizi gerçekleştiren “Şelale Modeli” seçtim çünkü programı yaparken kullanacağımız adımları iyi bir şekilde kontrol edip ilerlememi sağlıyor, bir adımın tamamlanmasından sonra diğerine geçilmesini ve bu şekilde bir adımın doğru bir şekilde bittiğinden emin oluyoruz.

Bu modelde katı bir disiplin ile çalıştık çünkü bir adımdan başka bir adıma geçtikten sonra geri o adıma dönmek çok büyük maliyetler ortaya çıkaracaktır.

Her adım tamamlandıktan sonra diğer adımlara geçilmesi sağlandı.

- **Sistem Gereksinimleri**

Müşterimiz olan Karakuşlar Market ile projenin hedeflerini ve gereksinimlerini belirledik.

Projemizin bir stok takip otomasyonu olduğu için hangi işlevleri yerine getirmesi gerektiği konuşuldu.

- **Yazılım Gereksinimleri**

Sistem gereksinimlerinden yola çıkılarak Stok takip otomasyonunda kullanılacak olan işlevlerin performans beklentilerini ve sistem ile nasıl bir etkileşim içerisinde olacaklarını belirledik.

Çeşitli kağıtlar üzerinde çalışarak doğru yola ulaşmaya çalışıldı.

- **Analiz**

Yazılım gereksinim belgesi temel alınarak projenin detaylı analizini yapmaya çalışıldı.

Stok takip otomasyonu içerisinde kullanılacak olan Kullanıcı arayüzü ve Veri tabanı mimarisi gibi detaylar araştırılıp analiz edildi.

Yazılımın yapısı belirlendi.

- **Tasarım**

Analiz aşamasında belirlenen gereksinimler ele alınarak, yazılımın tasarımı yapıldı.

Belirlenen arayüz tasarımı yapıldı.

- **Kodlama**

Tasarımı yapılan yazılım, hangi dilde çalışacağı belirlendi.

Kodun değişkenleri ve fonksiyonları oluşturuldu.

Yapılan yazılımın çalışıp çalışmadığı test edildi.

- **Test**

Yazılımın gerekli işlevleri yerine getirip getirmediği kontrol edildi.

Projemizin kodlama aşamasında yapılan yazılım sistemi performans ve yük testlerinde nasıl çalışacağı test edildi.

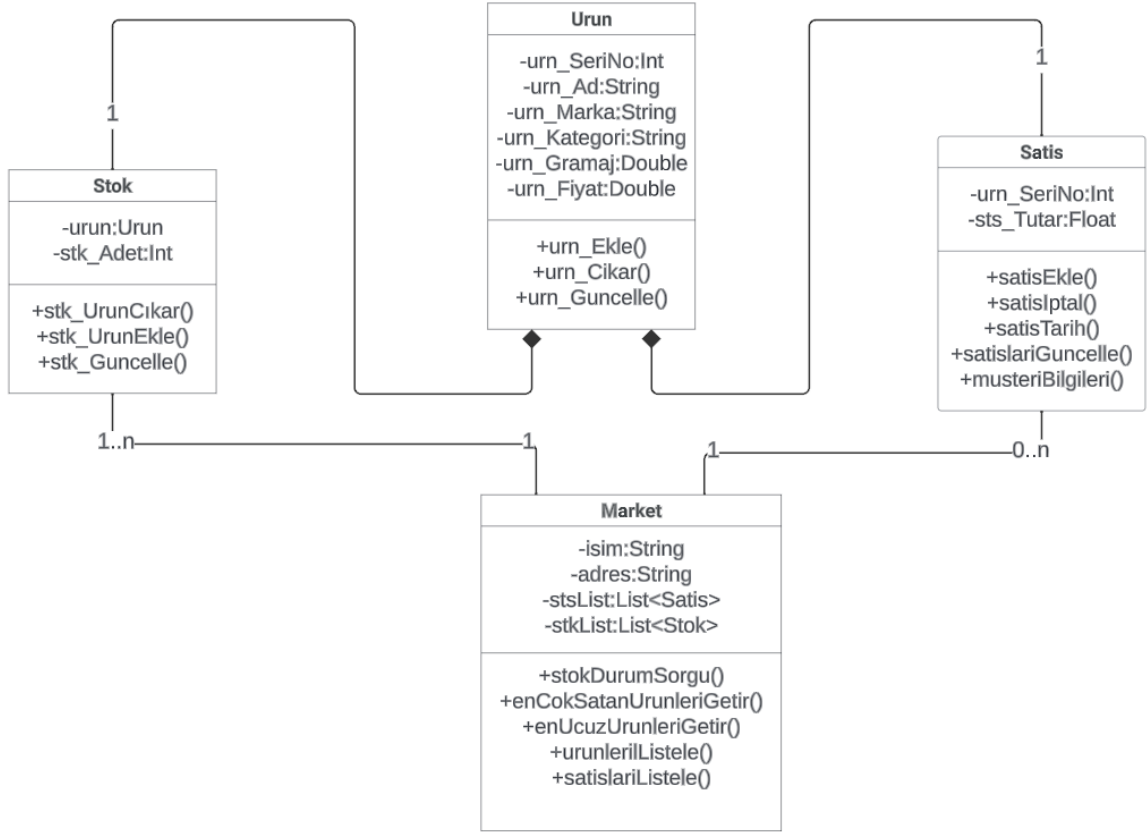
- **Bakım**

Müşterimiz olan “Karakuslar Market” ürünümüzü kullanmaya başladıktan sonra oluşabilecek hataları bizlere bildirdi.

Yazılımın nasıl bir performans verdiğini bu süreçte izlemiş olduk.

Gelen geri dönüşler ile geliştirmeye ve yenilemeye yönelik özelleştirmeler yapıldı.

Yeni özellikler eklendi.



UML Class Diyagramı

Urun ---- Stok ilişkisi: Urun ile stok arasında tek taraflı bir ilişki olduğunu gösteriyor.

Urun bir stoka aittir ve urun sınıfı yok olduğunda stok sınıfında yok olacaktır.

Bir ürün bir kere stoklanabilir.

Urun ---- Satış ilişkisi: Urun ile satış arasında tek taraflı bir ilişki olduğunu görüyoruz.

Daha doğru açıklayacak olursam bir ürün, bir kez satılabilir ve ürün sınıfı yok olduğunda satış sınıfı da yok olur.

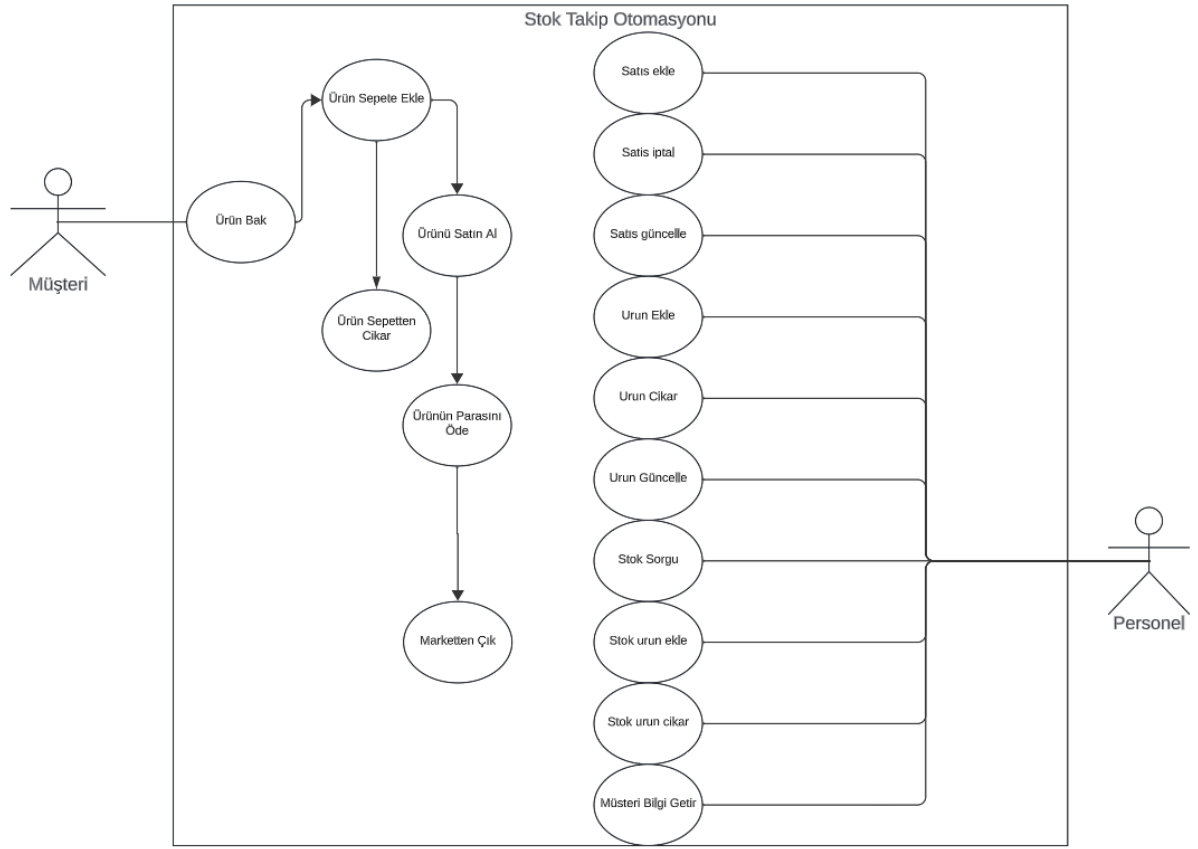
Bir ürün bir kere satılabilir.

Stok ---- Market ilişkisi: Stok ile Market arasında bir "1..n" ve "1" ilişkisi vardır. Market stoğa birden fazla kez gidebilir stok markete 1 kere gider.

Marketin bir veya daha fazla stoğu olabilir ama stoğun bir marketi olabilir

Satis ---- Market ilişkisi: Satis ile Market arasında "0..n" ve "1" ilişkisi vardır. Market 0 veya daha fazla kez satış gidebilir ama Satis markete bir kere gidebilir.

Market sıfır veya daha fazla satış yapabilir ama satis bir markette olabilir.

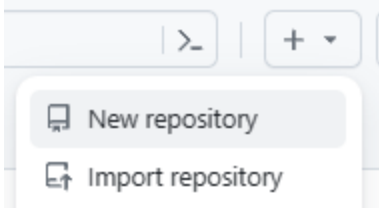


Use Case Diyagramı

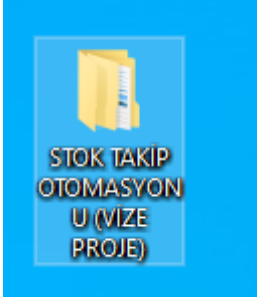
Aktör Müşteri : Müşteri Sisteme girdiğinde ürünlere bakıp ürünleri sepete ekleyebilir ekledikten sonra ürünü sepetten çıkarabilir veya ürünü satın alabilir ürünü Satın Al'a bastıktan sonra ürünün parasını öder ardından marketten çıkar.

Aktör Personel : Personel sisteme satış ekle , satış iptal ,satış güncelle , ürün ekle , ürün çıkar , ürün güncelle , stok sorgu , stok urun ekle ,stok urun çıkar ve müşteri bilgileri getir gibi işlevler ekleyebilir.

GİTHUB'A PROJEYİ YÜKLEME



1. Github'ta bir repository oluşturmak için hesabıma girdim ve new repository yaparak "stokTakipOtomasyon_VP" adında bir repository oluşturdum.



2. Anasayfama gelip bir dosya oluşturdum dosyanın içerisine girip sağ tık yapıp git bash'ı açtım

```
berka@DESKTOP-1BTGRPP MINGW64 ~/OneDrive/Masaüstü/STOK TAKIP OTOMASYONU (VİZE PR  
OJE)  
$ git clone https://github.com/wayolapre/stokTakipOtomasyon_VP.git  
Cloning into 'stokTakipOtomasyon_VP'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (3/3), done.
```

3. Git clone komudu ile oluşturduğum repository'i dosyama ekledim

```
berka@DESKTOP-1BTGRPP MINGW64 ~/OneDrive/Masaüstü/STOK TAKIP OTOMASYONU (VİZE PR  
OJE)  
$ cd stokTakipOtomasyon_VP/
```

4. Cd komudu ile git clone komuduyla dosyamın içerisine eklediğim repository'inin içerisine girdim

```
berka@DESKTOP-1BTGRPP MINGW64 ~/OneDrive/Masaüstü/STOK TAKIP OTOMASYONU (VİZE PR  
OJE)/stokTakipOtomasyon_VP (main)  
$ git add .
```

5. "Git add ." komuduyla dosyamın içerisine eklediğim raporumu yerel repository'e ekledim

```
berka@DESKTOP-1BTGRPP MINGW64 ~/OneDrive/Masaüstü/STOK TAKIP OTOMASYONU (VİZE PROJ  
E)/stokTakipOtomasyon_VP (main)  
$ git commit -m "proje eklendi"  
[main e97357b] proje eklendi  
1 file changed, 0 insertions(+), 0 deletions(-)  
rename "Yaz\304\261l\304\261m5\303\274recAnalizRapor.pdf" => "Yaz\304\261l\304\261m Geli\305\237tirme Rapor.pdf" (100%)
```

6. Git commit -m komutu "git add" komutu ile eklediğiniz tüm değişiklikleri kayıt etmek için kullandım

```
berka@DESKTOP-1BTGRPP MINGW64 ~/OneDrive/Masaüstü/STOK TAKIP OTOMASYONU (VİZE PROJ  
E)/stokTakipOtomasyon_VP (main)  
$ git push  
Enumerating objects: 6, done.  
Counting objects: 100% (6/6), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (5/5), 450.65 KiB | 26.51 MiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/wayolapre/stokTakipOtomasyon_VP.git  
2c738c2..e97357b main -> main
```

7. Git push komutuyla birlikte oluşturmuş olduğum projeyi Github'taki kaynağa gönderdim