



FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
Yazılım Mühendisliği



YMT114- Yazılım Mühendisliğinin Temelleri Dersi
Proje Uygulamaları ve Dokümantasyonu

E-Dolap Sistemi

Proje Çalışma Grubu

Tuba AKBAŞ

MAYIS 2020

1. GİRİŞ	
1.1 Projenin Amacı.....	7
1.2 Projenin Kapsamı.....	7
1.3 Tanımlamalar ve Kısaltmalar.....	7
2. PROJE PLANI	
2.1 Giriş.....	8
2.2 Proje Plan Kapsamı.....	8
2.3 Proje Zaman- İş Planı.....	9
2.4 Proje Ekip Yapısı.....	10
2.5 Kullanılan Özel Geliştirme Araçları ve Ortamları.....	11
2.6 Maliyet Kestirim ve Hesaplama.....	11
2.6.1 Ana İşlem Nokta Sayısının Hesaplanması.....	11
2.6.2 Teknik Karmaşıklık Faktörünün Hesaplanması.....	12
2.6.3 Temel Maliyet Modeli – COCOMO	13
2.7 Proje Standartları, Yöntem ve Metodolojiler.....	14
2.8 Kalite Sağlama Planı.....	15
2.9 Kaynak Yönetim Planı.....	15
2.10 Eğitim Planı.....	15
2.11 Bakım Planı.....	15
3. SİSTEM ÇÖZÜMLEME	
3.1 Mevcut Sistem İncelemesi.....	16
3.1.1 İşlevsel Model.....	16
3.1.2 Veri Modeli.....	17
3.1.3 Var Olan Yazılım/Donanım Kaynakları.....	17
3.2 Gereksenen Sistemin Mantıksal Modeli.....	17
3.2.1 Giriş.....	17
3.2.2 İşlevsel Model.....	18

3.2.3 Genel Bakış.....	18
3.2.4 Bilgi Sistemleri/Nesneler.....	19
3.2.5 Veri Modeli.....	19
3.2.6 Veri Sözlüğü.....	20
3.2.7 İşlevlerin Sıradüzeni.....	20
3.2.8 Başarım Gerekleri.....	20
3.3 Ara Yüz (Modül) Gerekleri.....	20
3.3.1 Yazılım Ara Yüzü.....	20
3.3.2 Kullanıcı Ara Yüzü.....	21
3.3.3 İletişim Ara Yüzü.....	21
3.3.4 Yönetim Ara Yüzü.....	21
3.4 Belgeleme Gerekleri.....	21
3.4.1 Geliştirme Sürecinin Belgelenmesi.....	21
3.4.2 Eğitim Belgeleri.....	21
3.4.3 Kullanıcı El Kitapları.....	21
4. SİSTEM TASARIMI	
4.1 Genel Tasarım Bilgileri.....	22
4.1.1 Genel Sistem Tanımı.....	22
4.1.2 Varsayımları ve Kısıtlamalar.....	24
4.1.3 Sistem Mimarisi.....	24
4.1.4 Dış Arabirimler.....	24
4.1.4.1 Kullanıcı Arabirimleri.....	24
4.1.4.2 Veri Arabirimleri.....	24
4.1.4.3 Diğer Sistemlerle Arabirimler.....	25
4.1.5 Veri Modeli.....	25

4.1.6	Testler.....	25
4.1.7	Performans.....	25
4.2	Veri Tasarımı.....	25
4.2.1	Tablo Tanımları.....	25
4.2.2	Tablo-İlişki Şemaları.....	26
4.2.3	Veri Tanımları.....	26
4.3	Süreç Tasarımı.....	26
4.3.1	Genel Tasarım.....	26
4.3.2	Modüller.....	26
4.3.2.1	Giriş Modülü.....	26
4.3.2.1.1	İşlev.....	26
4.3.2.1.2	Kullanıcı Arabirimi.....	26
4.3.2.1.3	Modül Tanımı.....	26
4.3.3	Kullanıcı Profilleri.....	26
4.4	Ortak Alt Sistemlerin Tasarımı.....	27
4.4.1	Ortak Alt Sistemler.....	27
4.4.2	Modüller Arası Ortak Veriler.....	27
4.4.3	Ortak Veriler için Veri Giriş ve Raporlama Modülleri.....	27
4.4.4	Güvenlik Alt Sistemi.....	27
4.4.5	Veri Dağıtım Alt Sistemi.....	27
4.4.6	Yedekleme ve Arşivleme İşlemleri.....	27
5. SİSTEM GERÇEKLEŞTİRİMİ		

5.1 Giriş.....	28
5.2 Yazılım Geliştirme Ortamları.....	28
5.2.1 Programlama Dilleri.....	29
5.2.2 Veri Tabanı Yönetim Sistemleri.....	29
5.2.2.1 VTYS Kullanımının Ek Yararları.....	29
5.2.2.2 Veri Tabanı Dilleri ve Arabirimleri.....	30
5.2.2.3 Veri Tabanı Sistem Ortamı.....	30
5.2.2.4 VTYS'NİN Sınıflandırılması.....	30
5.2.2.5 Hazır Program Kütüphane Dosyaları.....	30
5.2.2.6 CASE Araç ve Ortamları.....	30
5.3 Kodlama Stili.....	30
5.3.1 Açıklama Satırları.....	30
5.3.2 Kod Biçimlemesi.....	30
5.3.3 Anlamlı İsimlendirme.....	30
5.3.4 Yapısal Programlama Yapıları.....	31
5.4 Program Karmaşıklığı.....	31
5.4.1 McCabe Karmaşıklık Ölçütü Hesaplama.....	31
5.5 Olağan Dışı Durum Çözümleme.....	32
5.5.1 Olağan Dışı Durum Tanımları.....	32
5.5.2 Farklı Olağan Dışı Durum Çözümleme Yaklaşımları.....	32
5.6 Kod Gözden Geçirme.....	32
5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi.....	32
5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular.....	32
5.6.2.1 Öbek Ara Yüzü.....	32
5.6.2.2 Giriş Açıklamaları.....	33
5.6.2.3 Veri Kullanımı.....	33
5.6.2.4 Öbeğin Düzenlenişi.....	33
5.6.2.5 Sunuş.....	34

6. DOĞRULAMA VE GEÇERLEME	
6.1 Giriş.....	35
6.2 Sınama Yöntemleri.....	35
6.2.1 Beyaz Kutu Sınaması.....	35
6.3Sınama Planlaması.....	35
6.3.1 Test Planı Kimliği.....	35
6.3.2 Test Edilecek Sistem.....	35
6.3.3 Test Edilecek Ana Fonksiyonlar.....	35
6.3.4 Test Edilmeyecek Ana Fonksiyonlar.....	35
6.3.5 Geçti / Kaldı Kriterleri.....	35
6.3.6 Test Dokümanı.....	36
6.3.7 Sorumluluklar.....	36
6.3.8 Riskler ve Önlemler.....	36
6.4 Sınama Belirtileri.....	36
7. BAKIM	
7.1 Giriş.....	37
7.2 Kurulum.....	37
7.3 Yazılım Bakımı.....	37
7.3.1 Tanım.....	37
7.3.2 Bakım Süreç Modeli.....	38
8. SONUÇ.....	43
9. KAYNAKLAR.....	43

1. GİRİŞ

1.1 Projenin Amacı

Gelişen ve gelişmekte olan dünya var oldukça teknolojiye ilerlemeye ve gelişmeye devam etmektedir. Üretilen elektrikli aletler günümüzde önemli bir yere sahiptir. Şüphesiz bu aletlerden enerji tasarrufu sağlamayı hepimiz isteriz. Peki, bir buzdolabını ele alacak olursak sürekli kapısı açılan buzdolaplarının enerji tasarrufunu nasıl sağlarız?

Buzdolabının kapısını uzun süre açık bırakırsak soğuk hava kaybı yaşanır ve buzdolabı içine sıcak hava girer böylelikle buzluk üzerinde karlanma oluşur. Bu karlanma normalin üstünde olacağı için buzluk üstündeki rezistanslar bu karlanmayı çözemez bazı durumlarda ise buzluk içindeki fan motorunun pervanesine çarpar ve fanın çalışmamasına neden olur. Soğutma tamamen kesilir.

Belirtmiş olduğum problemlerden dolayı E-Buzdolabı projesini düşündüm. Biz tüketiciler haftalık veya aylık alışverişlerimiz için sürekli buzdolabını açıp nelerin eksik ve fazla olduğunu görerek bir alışveriş listesini hazırlarız. Bu işlemi dijital ortamda hiç buzdolabını açmadan halledebiliriz. Yapacağım otomasyonda buzdolabında eksik olan ürünlerin tespitinin daha önceki girilen verilerden tespit edip yeni ürünleri sistemden girmektir. Çıktı olarak alışveriş listesi hazırlanarak markete mail atılır. Böylelikle enerji tasarrufu ve zamandan tasarruf sağlanır.

1.2 Projenin Kapsamı

Projenin herkes tarafından kullanılabilir. Uygulama sahası ise buzdolabına ihtiyaç duyulan her alanda kullanılabilir.

1.3 Tanımlamalar ve Kısaltmalar

VYTS: Veri Tabanı Yönetim Sistemleri

İN: İşlevsel Nokta

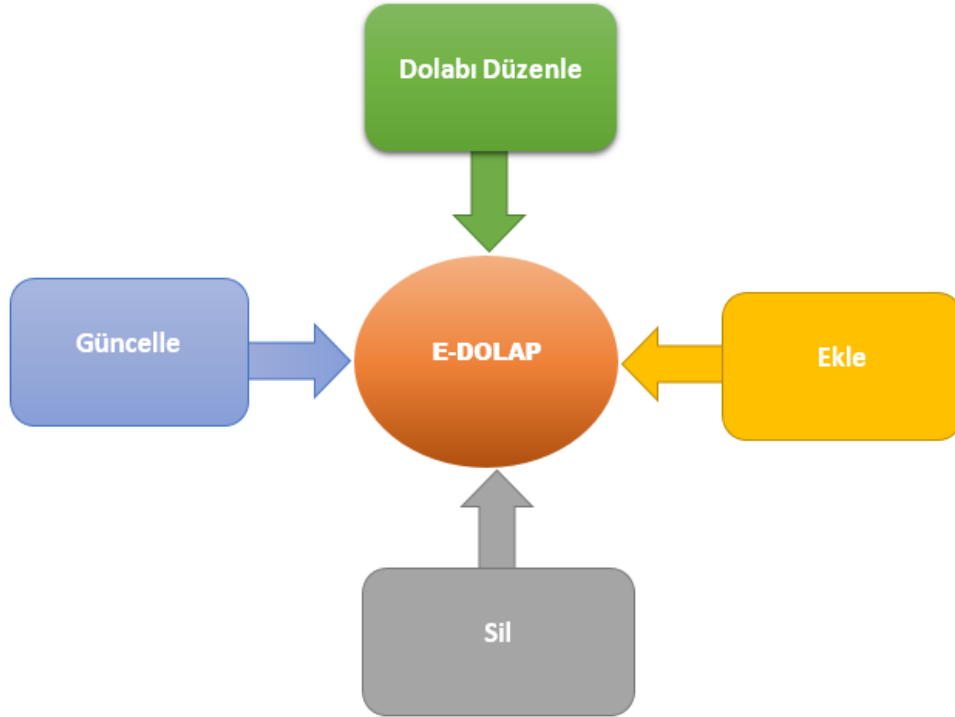
AİN: Ayarlanmamış İşlevsel Nokta

TKF: Teknik Karmaşıklık Faktörü

2. PROJE PLANI

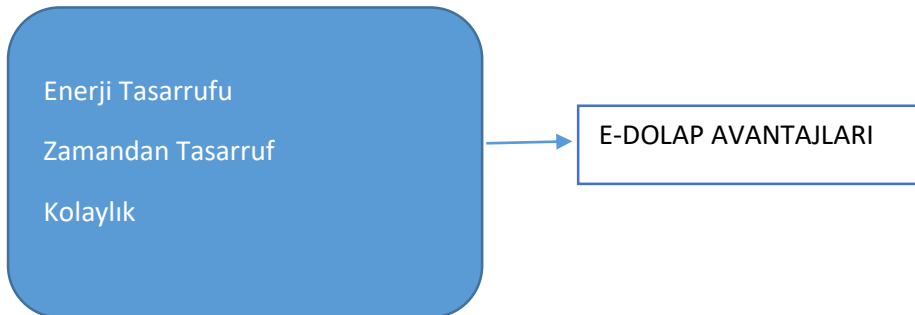
2.1 Giriş

E-Dolap Sistemi kullanıcıların girecekleri veriler doğrultusunda bir alışveriş listesi hazırlar. Kullanıcı istekleri doğrultusunda güncelleme, silme, ekme yapabilmektedir. Kullanımı kolay ve anlaşılırdır.

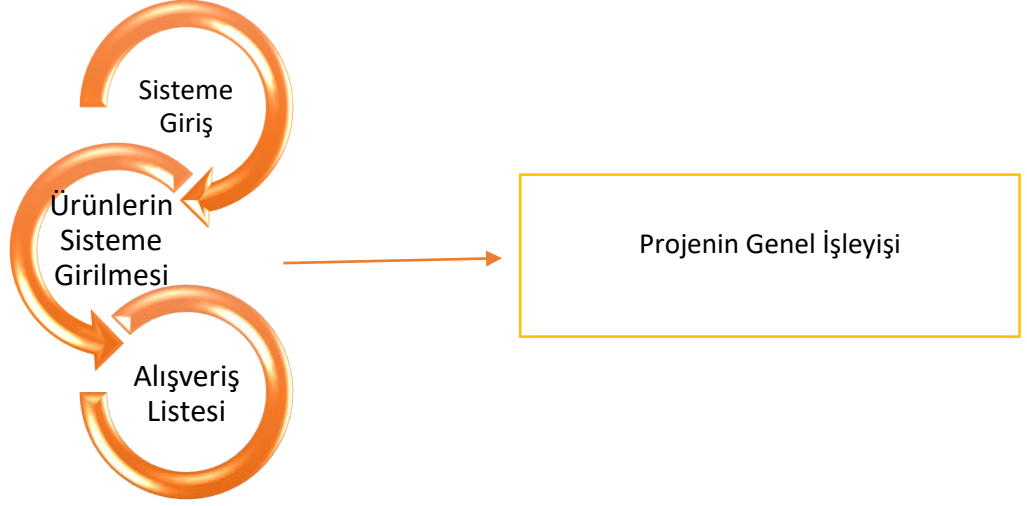


Şekil 2.1 Projenin Genel Yapısı

2.2 Proje Plan Kapsamı



Şekil 2.2 Projenin Avantajları

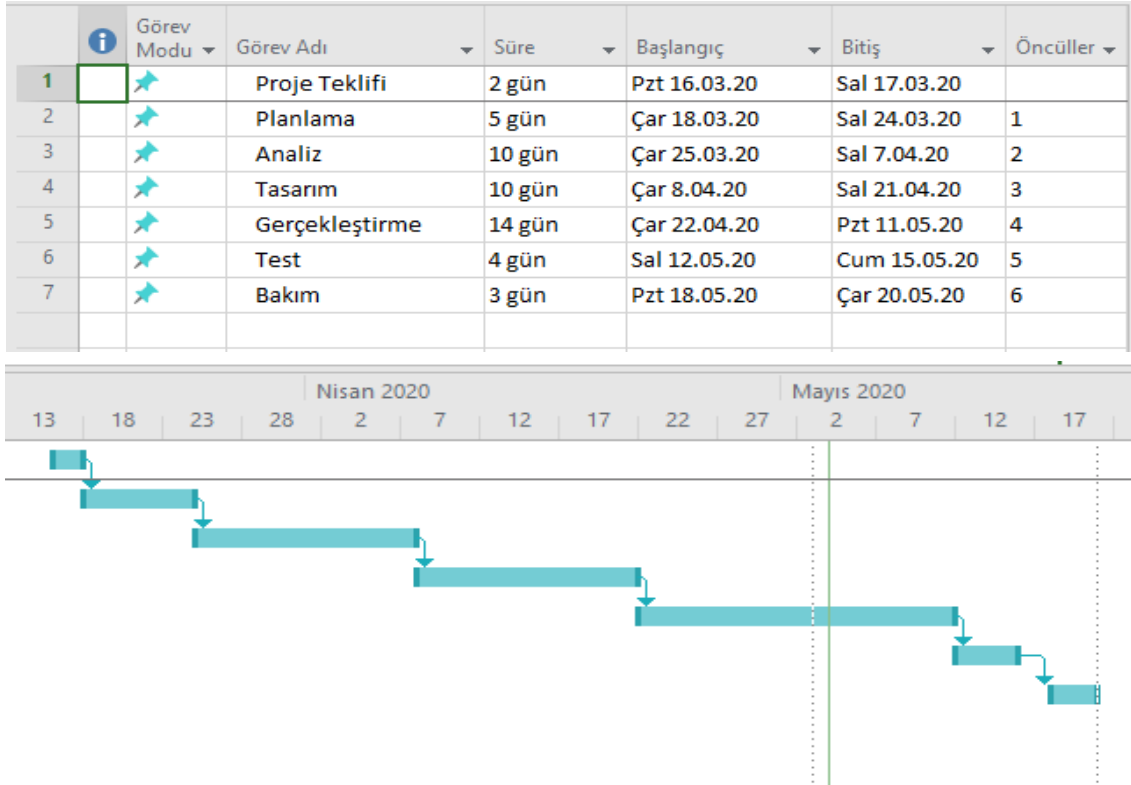


Şekil 2.2 Projenin Genel İşlenişi

E-Dolap Sistemi Nasıl Çalışır?

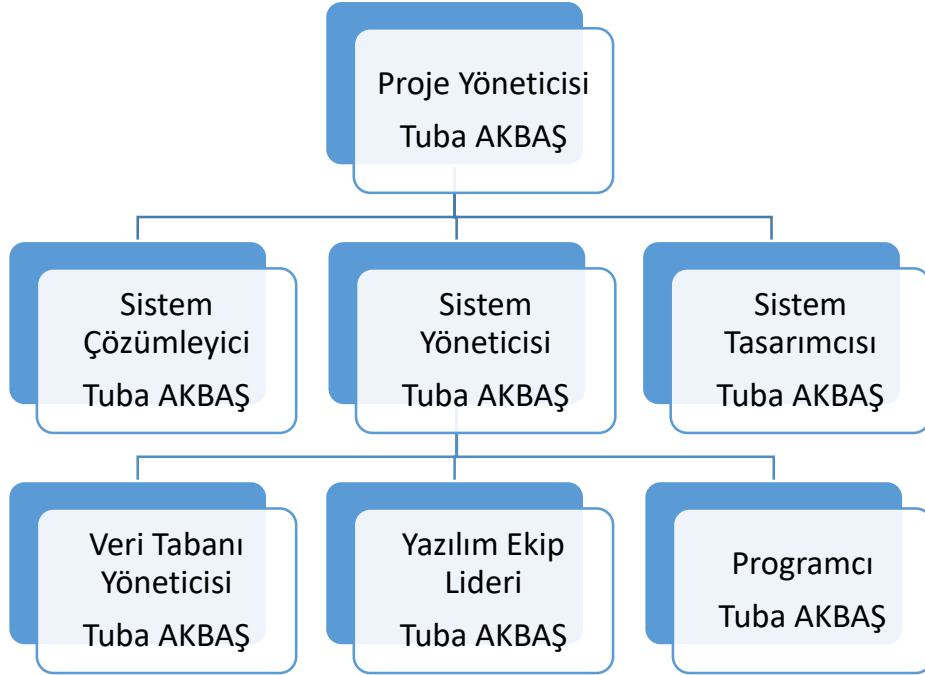
Sisteme giriş yapan müşteri ürünleri sisteme girerek bir alışveriş listesi hazırlar ve çıktı alır.

2.3 Proje Zaman-İş Planı



Gantt/Diyagramı

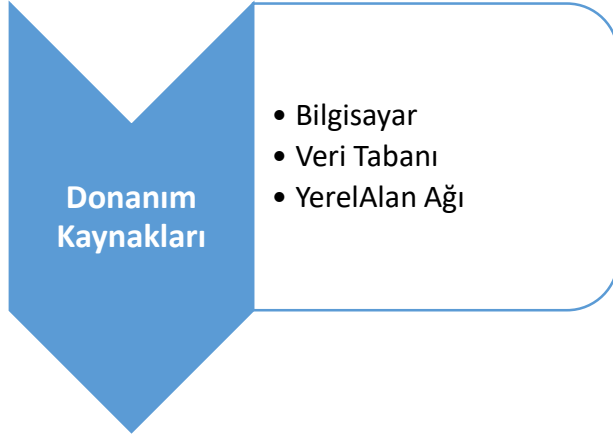
2.4 Proje Ekip Yapısı



Proje Yöneticisi	<ul style="list-style-type: none">•Projenin yönetilmesi•Proje Ekip yapısının oluşturulması•İş planlamasının yapılması
Sistem Çözümleyici	<ul style="list-style-type: none">•Dokümantasyon ve Raporlamanın hazırlanması
Araştırma Ekibi	<ul style="list-style-type: none">•Proje için gerekli kaynakların temin edilmesi•Örnek sistemlerin incelenmesi
Veri Tabanı Ekibi	<ul style="list-style-type: none">•Veri tabanı sistemlerinin oluşturulması•Veri tabanı tasarımının oluşturulması
Programcı	<ul style="list-style-type: none">•Tasarım ve kodlama
Eğitim Ekibi	<ul style="list-style-type: none">•Programcı ekibin eğitimi•Kullanıcı eğitimleri
Test ve Bakım Ekibi	<ul style="list-style-type: none">•Sistem Testlerinin Yapılması•Sistem bakımının planının yapılması

Şekil 2.4 Proje Ekip Görevleri

2.5 Kullanılan Özel Geliştirme Araçları ve Ortamları



Şekil 2.5 Donanım Kaynakları



2.6 Maliyet Kestirim ve Hesaplama

2.6.1 Ana İşlem Nokta Sayısının Hesaplanması

Ölçüm Parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	4	7	28
Kullanıcı Çıktı Sayısı	4	7	28
Kullanıcı Sorgu Sayısı	3	8	24
Kütük Sayısı	2	6	12
Dışsal Ara Yüz Sayısı	0	0	0
Ana İşlem Nokta Sayısı			92

2.6.2 Teknik Karmaşıklık Faktörünün Hesaplanması

Teknik Karmaşıklık Sorusu	Puan
1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?	2
2. Veri İletişimi gerekiyor mu?	4
3. Dağıtık işlem işlevleri var mı?	3
4. Performans kritik mi?	3
5. Sistem mevcut ve ağır yükü olan bir işletim ortamında mı çalışacak?	2
6. Sistem, çevrim içi veri girişi gerektiriyor mu?	4
7. Çevrimiçi veri giriş, ara işlem için birden çok ekran gerektiriyor mu?	3
8. Ana kütükler çevrim içi olarak mı güncelleniyor?	3
9. Girdiler, çıktılar kütük ya da sorgular karmaşık mı?	2
10. İçsel işlemler karmaşık mı?	2
11. Tasarlanacak kod yeniden kullanılabilir mi olacak?	4
12. Dönüştürme ve kurulum, tasarımda dikkate alınacak mı?	4
13. Sistem birden çok yerde farklı kurumlar için mi geliştiriliyor?	1
14. Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından değiştirilebilir mi olacak?	5
TOPLAM	42

Şekil 2.6.2 Teknik Karmaşıklık Faktörünün Hesaplanması

0: Hiçbir Etkisi Yok

1: Çok Az Etkisi Var

2: Etkisi Var

3: Ortama Etkisi var

4: Önemli Etkisi Var

5: Mutlaka Olmalı, Kaçınılmaz

2.6.3 Temel Maliyet Modeli- COCOMO

İşlevsel Nokta Sayısını Hesaplamak için Gerekli Formül

$$\dot{N} = A \dot{N} * (0.65 * 0.01 * TKF)$$

$$\dot{N} = 92 * (0.65 * 0.01 * 42)$$

$$\dot{N} = 25.11$$

$$\text{Satır Sayısı} = \dot{N} * 30$$

$$\text{Satır Sayısı} = 25.11 * 30 = 753$$

Öngörülen toplam satır sayısı 750 satırdır.

Projenin Türü: Ayrık Proje

$$\text{İş Gücü (K)} = 2.4 * S^{1.05}$$

$$\text{İş Gücü (K)} = 2.4 * (0.750)^{1.05} = 1.774$$

$$\text{Zaman (T)} = 2.5 * K^{1.05}$$

$$\text{Zaman (T)} = 2.5 * (1.774)^{1.05} = 4.563$$

S: Bin Türünden Satır Sayısı

2.7 Proje Standartları, Yöntem ve Metodolojileri

Planlama

- **Kullanılan Yöntem/Araçlar**

Veri akış şemaları, süreç belirtimleri, görüşme, maliyet kestirim yöntemleri, proje yönetim araçları

- **Ne için Kullanıldığı**

Süreç inceleme, kaynak kestirimi, proje yönetimi

- **Çıktı**

Proje planı

Çözümleme

- **Kullanılan Yöntem/Araçlar**

Veri akış şemaları, süreç belirtileri, görüşme, nesne ilişki şemaları, veri sözlüğü

- **Ne için Kullanıldığı**

Süreç çözümleme, veri çözümleme

- **Çıktı**

Sistem çözümleme raporu

Çözümlemeden Tasarıma Geçiş

- **Kullanılan Yöntem/Araçlar**

Akışa dayalı çözümleme, Süreç belirtilerinin program tasarım diline dönüştürülmesi, nesne ilişki şemalarının veri tablolarına dönüştürülmesi

- **Ne için Kullanıldığı**

Başlangıç tasarım, ayrıntılı tasarım, başlangıç veri tasarımı

- **Çıktı**

Başlangıç tasarım raporu

Tasarım

- **Kullanılan Yöntem/Araçlar**

Yapısal şemalar, program tasarım dili, veri tabanı tabloları, veri sözlüğü

- **Ne için Kullanıldığı**

Genel Tasarım, Ayrıntılı Tasarım, Veri Tasarımı

- **Çıktı**

Sistem Tasarım Raporu

Proje standartları yukarıda belirtildiği gibidir. Bunun yanı sıra kullanılan süreç modeli ise Artırımsal Geliştirme Süreç Modeli'dir.

Süreç Modeli

Artımlı Geliştirme(Incremental Development)

Eğer bir müşterinin ürünlerinde değişikliğe ihtiyaçları varsa, artımlı model ihtiyaç olan bu değişikliğe ayak uydurur. Artırımsal model bir takvime bağlı olarak yazılım kesim kesim geliştirilip teslim etmeye dayanır. Her bir yeni kesim öncekinin üstüne bazı ek işlevlerin eklenmesini öngörür. Artırımsal model yazılım geliştirmenin kısıtlı sayıda çalışanla işin yapılmasının sağlama gibi bir üstünlüğü vardır. Bu model yinelemelidir. Yeniden kullanılabilir bir ürün, fonksiyonellik sağlamış bir şekilde tüm döngülerin sonunda ortaya çıkar.

Bu modeli kullanmamdaki amacım ise müşterinin isteklerine ihtiyaçlarına göre değişikliğe daha kolay ayak uydurmasıdır.

2.8 Kalite Sağlam Planı

Ekonomi, tamlik, geçerlik, değiştirilebilirlik, yeniden kullanılabilirlik, esneklik, etkinlik, genellik, bütünlük, anlaşılabilirlik gibi kalite unsurları vardır.

2.9 Kaynak Yönetim Planı

Herhangi bir kaynağımız olmadığından kaynak olarak sadece proje dokümantasyonu vardır.

2.10 Eğitim Planı

Bu sistemi kullanacak olan kullanıcılar için kullanım kılavuzu olacaktır ayrıca eğitim semineri verilecektir. Zira verilmemesi halinde kullanıcılar sistemi tam olarak kullanamayınca bazı açıklıklar meydana gelecektir.

2.11 Bakım Planı

Sistemin Bakımı Admin1 tarafından yapılarak her türlü güncelleştirmeye olanak sağlayacaktır.

3. SİSTEM ÇÖZÜMLEME

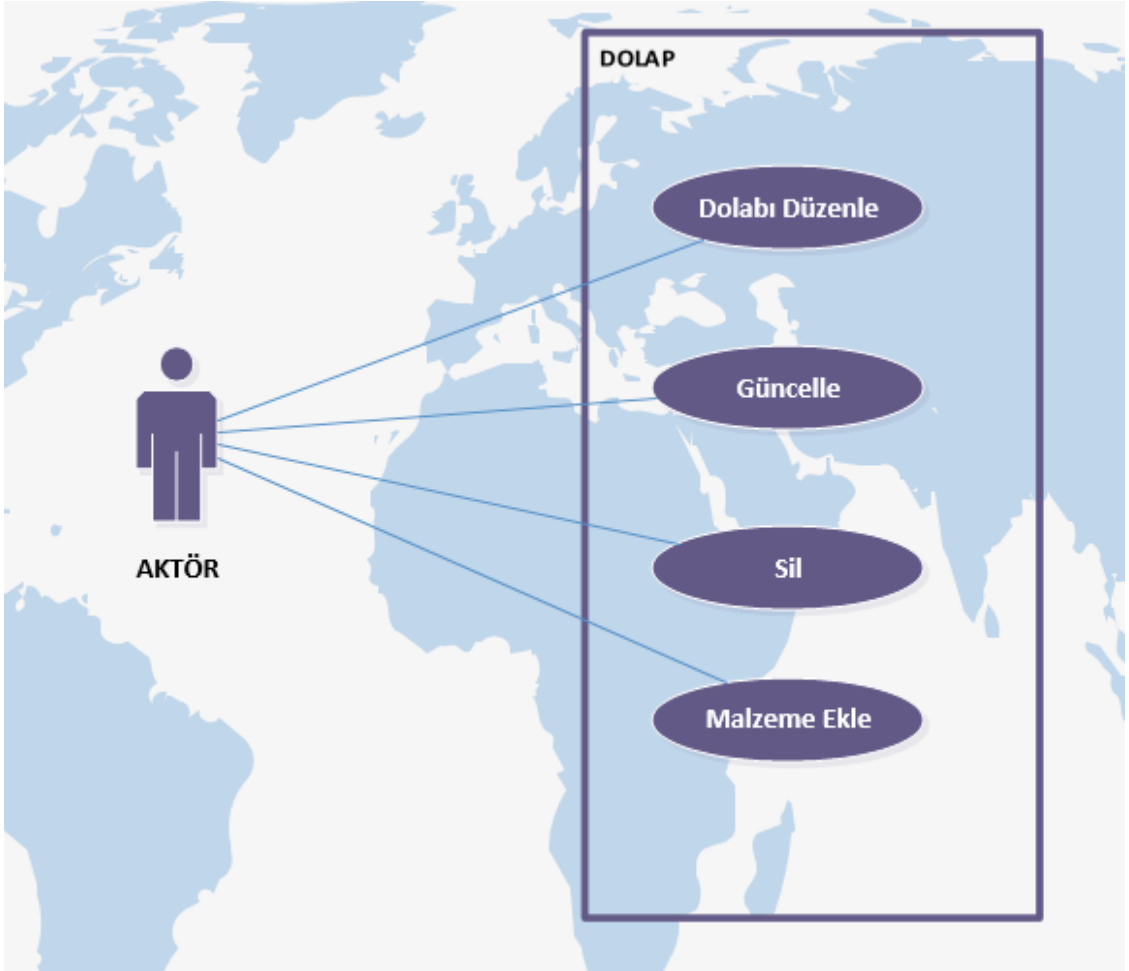
3.1 Mevcut Sistem İncelenmesi

Mevcut sistem incelemesi öncelikle kullanıcılar ilk defa kullanacakları için biraz zorluk çekebilirler ama sistemin anlaşılır basit olması nedeniyle uygulanışı kolay olacaktır.

- **İşlevsel Gereksinimler:** Alışveriş listesinin hazırlanması bunun için girdi olarak ürünlerin sisteme girilmesi gerekmektedir.
- **İşlevsel Olmayan Gereksinimler:**
Yazılım Geliştirme Ortamı: Java, MySQL

3.1.1 İşlevsel Model

Sistemin nasıl kullanıcı tarafından nasıl işlendiğini anlatan use case diyagramı.

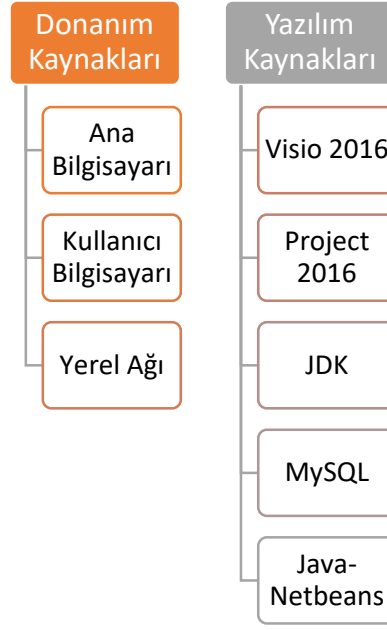


Şekil 3.2.1 İşlevsel Model

3.1.2 Veri Modeli

Bütün tablolar birbirinden bağımsızdır.

3.1.3 Var olan Yazılım/Donanım Kaynakları



3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş

Sistemin amacı kullanıcıya kolaylık sağlayarak alışveriş listesi hazırlamaktır.

Fiziksel Çevre, çevresel kısıtlamalar yoktur.

Ara yüzler, girdiler bir sistemden gelir çıktımız bir tanedir.

Kullanıcı ve İnsan Etmeni: Herkes kullanabilir. Sistemin kullanımı gayet basittir.

İşlevsellik: Sistem girdileri alarak çıktı oluşturur. Sistem bu işlemi anlık olarak gerçekleştirecektir. Çalışma hızı, yanıt süresi ya da çıktı üzerinde kısıtlayıcı etmenler yoktur.

Veri: Veriler için bir veri tabanı olacaktır. Belli bir zaman süresinde veri kaynağında saklanacaktır.

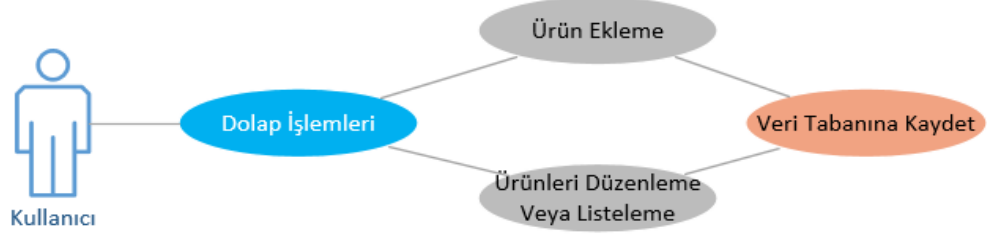
Kaynaklar: personel kaynaklarına ihtiyaç yoktur.

3.2.2 İşlevsel Model



Use-Case Diyagramından da görüleceği gibi sistemin işleyici şu şekildedir:

- ✓ Sisteme Giriş
- ✓ Ürünleri eklemek/silmek/güncellemek
- ✓ Alışveriş Listesini Hazırlamak



Şekil 3.2.2 Temel Kullanım Diyagramı

3.2.3 Genel Bakış

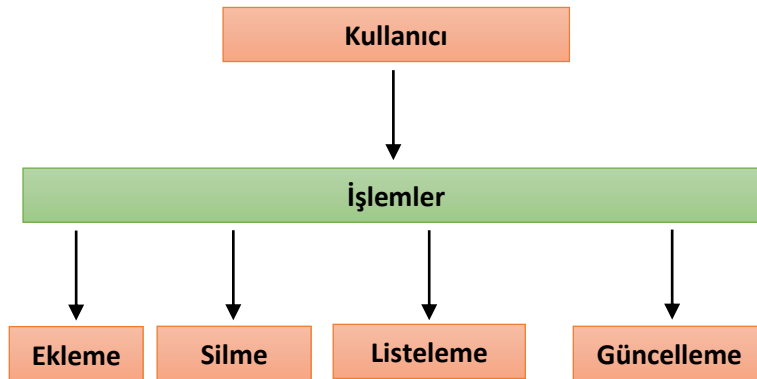
Genel olarak sistemi inceleyecek olursak e-dolap sisteminin ana ekranında bir form açılmaktadır. Buradan dolap işlemleri kısmına girdiğimizde ekleme silme güncelle işlemlerinden istediğimi seçerek çıktı olarak bir alışveriş listesi hazırlarız. Bunun yanı sıra olayın akış şekli USE-CASE diyagramında mevcuttur.

3.2.4 Bilgi Sistemleri/Nesneler



3.2.5 Veri Modeli

Modelin Tanımlanması:



3.2.6 Veri Sözlüğü

Ana Bilgilerin Tanımlanması:

Dosya Adı	Değişken Adı	Veri Tipi	Uzunluk	Açıklama
Dolap	İd	int	4	Ürün Numarası-PK
	urun_adi	varchar	45	Ürün Adı
	urun_kilo	int	4	Ürün Kilosu
	son_kullanim_tarihi	date	-	Ürünün Son Kullanım Tarihi

Şekil 3.2.6 Veri Sözlüğü Tablosu

3.2.7 İşlevlerin Sıradüzeni

İşlevlerin sıradüzeninde ilk olarak kullanıcı yapacağı işlemi belirler eğer ekleme işlemi yapacaksa ürünün adını, kilosunu ve son kullanım tarihini girerek alışveriş listesini hazırlar. Eğer ki güncelleme işlemi yapacaksa tablodaki verilerden değişiklik yapar alışveriş listesini hazırlar. Çıktı olarak işlemi bitirir.

3.2.8 Başarım Gerekleri

Mevcut sistemler incelendi ve eksikliklerden yola çıkılarak, sistemin başarımı için,

- ✓ Kullanım Kolaylığı
- ✓ Anlaşılabilirlik
- ✓ Hata ve yanlışlıkların en aza indirilmesi

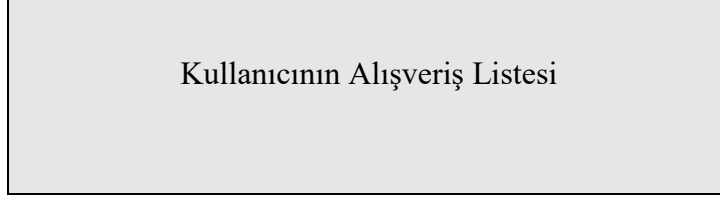
3.3 Ara Yüz (Modül) Gerekleri

3.3.1 Yazılım Ara Yüzü

Gerekli olan her türlü değişiklik kodlar üzerinden yapıp tekrar derlenecektir.

Girdiler sistemden gelir çıktılar bir tanedir.

3.3.2 Kullanıcı Ara yüzü



Dosya İşlemleri

Alışveriş Listesi

Kullanıcı Ara Yüzü şekilde görüldüğü gibidir.

3.3.3 İletişim Ara yüzü

Sistem geliştirilip destek ekipleri oluşturulduğunda müşterinin görüşebileceği iletişim sisteme eklenecektir.

3.3.4 Yönetim Ara yüzü

Sistem geliştirildiğinde yönetici ara yüzü eklenecektir.

3.4 Belgeleme Gereklere

3.4.1 Geliştirme Sürecinin Belgelenmesi

Genel olarak belgelendirilmesi ileride projenin daha da gelişmesine nerelerde eksikler olduğunu gösterme amacıyla yapıldı.

3.4.2 Eğitim Belgeleri

Eğitim Belgesi bulunmamaktadır.

3.4.3 Kullanıcı El Kitapları

Kullanıcının bu sistemi kullanırken rahatlık ve kolaylık sağlaması açısından kullanım kılavuzu hazırlanacaktır.

4. SİSTEM TASARIMI

4.1 Genel Tasarım Bilgileri



4.1.1 Genel Sistem Tanımı

Gereksinimler	Mantıksal Model	Tasarım
<ul style="list-style-type: none">Bu kısımda kullanıcının ne istedi önemlidir.	<ul style="list-style-type: none">Sistemin Ne yapacağı	<ul style="list-style-type: none">Yazılımın gereksinimleri Nasıl Karşılacağı

- Gereksinimler**

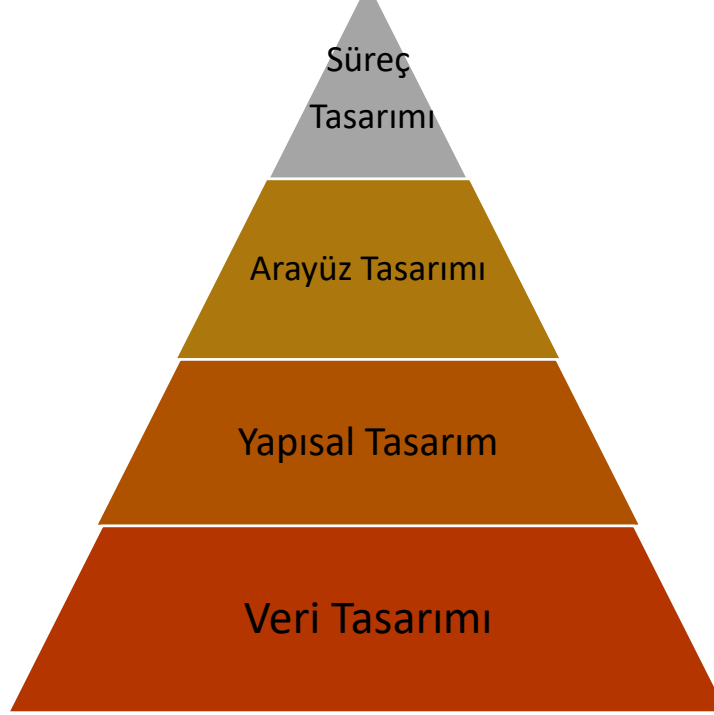
Gereksinimler kısmında müşterinin ne istediği konusu üzerinde durulmuştur. Bunun sonucunda ise müşterinin hayatını kolaylaştıracak aynı zamanda tasarruf sağlayacak yönde olması belirlenmiştir.

- Mantıksal Model**

Bu kısımda ilk olarak sistemin ne yapacağı sorusuna cevap verelim. Sistem müşteri tarafından girilen verileri veri tabanına kaydeder. Kaydedilmiş veriler istendiği zaman tekrardan sisteme girilerek düzenlenebilmektedir. Bu düzenlemeler silme, ekleme, güncelle şeklindedir. Daha sonra sistem geliştirilerek kullanıcı girişi sayesinde kullanıcının adresini sistemde tutarak ilgili marketlere alışveriş listesinin gönderimini sağlar. Bu dokümanımda ise ben sadece müşterinin alışveriş listesini görüntülemesini sağladım.

- **Tasarım**

Tasarım aşamasında neler olacağını grafiksel olarak göstermek gerekirse:



- **Süreç Tasarımı:**

Süreç tasarımında proje sürecinde planlama analiz ve bunların alt basamaklarında ise grafiksel ve tablo şeklinde yaptığım şeyleri anlattım.

- **Ara yüz Tasarımı:**

Kullanıcı ara yüzleri oluşturuldu. Veri akış diyagramı yapıldı.

- **Yapısal Tasarım**

Yapısal tasarımda ise Veri nesnesi belirtimi, nesne ilişki diyagramı ve veri akış diyagramı yapıldı.

- **Veri Tasarımı**

Veri tasarımı kısmında veri sözlüğünü, veri nesne belirtimi ve nesne ilişki diyagramları yapıldı.

4.1.2 Varsayımlar ve Kısıtlamalar

Sistemin kısıtlamaları yoktur.

4.1.3 Sistem Mimarisi



Sistemin mimarisinin akış diyagramı şeklinde verilmesinin temel nedeni sistemin nasıl işlendiğinin mantığının daha iyi anlaşılması sağlamaktır.

4.1.4 Dış Arabirimler

4.1.4.1 Kullanıcı Arabirimleri

İlk açılan sayfa müşterinin seçim yapacağı ekran kullanıcı arabirimidir.

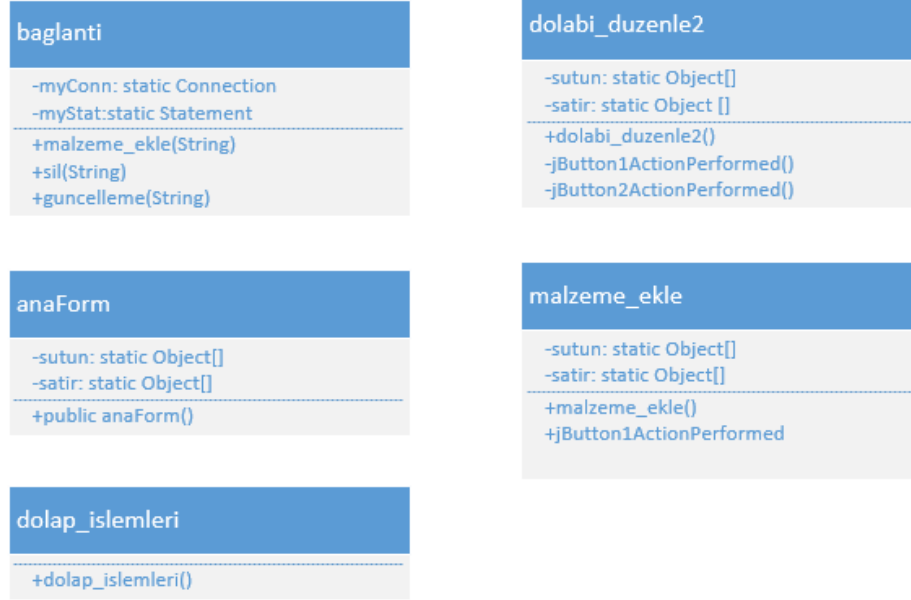
4.1.4.2 Veri Arabirimleri

Kullanıcının sisteme girmiş olduğu verileri ve son şeklini veri tabanı kaydeder.

4.1.4.3 Diğer Sistemlerle Arabirimler

Arabirim kullanılmayacaktır.

4.1.5 Veri Modeli



UML Sınıf Diyagramı

4.1.6 Testler

Kullanıcılar tarafından yapılan geri dönüşler ile sistem test edilecektir ve çözümleri uygulanacaktır.

4.1.7 Performans

Sistemin Tasarıma Uygunluk Performansı, sistemin kararlılığı ve işleniş performansı değerlendirilir.

Veri Yapısının Sistemle Performansı, sistemin kararlılığı ve işleniş performansı değerlendirilecektir.

4.2 Veri Tasarımı

4.2.1 Tablo Tanımları

Sistem 1 tablodan oluşur. Kullanıcı ekstra bir tablo eklemeyecektir.

4.2.2 Tablo-İlişki Şemaları

Tablolar arasında ilişki yoktur.

4.2.3 Veri Tanımları

Veri tipi olarak int kullanılmasının nedeni ürünün kilosunun int olarak girilmesi gerektiğindendir.

String kullanılmasının amacı ise ürünün adının kelime içeren değerleri tutacağından ötürüdür.

Date kullanılmasının amacı da ürünün son kullanım tarihinin tarih formatında tutulmasından ötürü kullanıldı.

4.3 Süreç Tasarımı

4.3.1 Genel Tasarım

Tasarımda ilk olarak veri tabanı modeli oluşturuldu daha sonra kullanıcının ara yüzü oluşturuldu.

4.3.2 Modüller

4.3.2.1 Giriş Modülü

4.3.2.1.1 İşlev

Kullanıcı adı ve şifresi henüz sistemimde mevcut olmadığı için kullanıcı sistemi direkt olarak kullanmaktadır.

4.3.2.1.2 Kullanıcı Arabirimi

Kullanıcı sistemi direkt olarak kullanır.

4.3.2.1.3 Modül Tanımı

Kullanıcılara yönetim tarafından verilecek kullanıcı adı ve şifresi bulunmamaktadır.

4.3.3 Kullanıcı Profilleri

Kullanıcı: Yapının nesnesidir. Kullanıcı istekleri dahilinde sisteme ürünleri ekler ve alışveriş listesini oluşturur.

4.4 Ortak Alt Sistemlerin Tasarımı

4.4.1 Ortak Alt Sistemler

Ortak alt sistemler bulunmamaktadır.

4.4.2 Modüller Arası Ortak Veriler

Modüller arasında ilişkili veri modeli bulunmadığından dolayı ortak veriler yoktur.

4.4.3 Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri

Ortak veriler için bu şekilde bir modül kullanılmadı.

4.4.4 Güvenlik Alt Sistemi

Tahmin edilen hata ve arıza oranları ile gerçekleşen değerleri kıyaslamak.

4.4.5 Veri Dağıtım Alt Sistemi

Ürünler sisteme girilir ve veri tabanına kaydedilir.

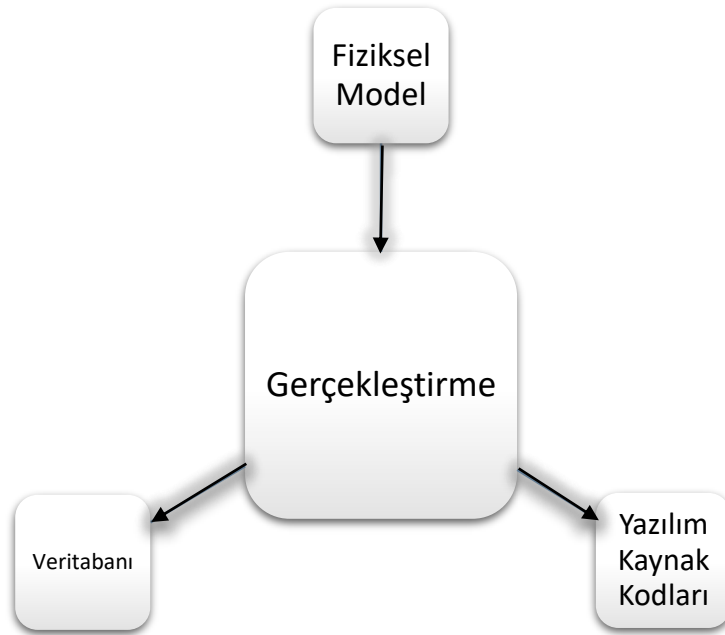
4.4.6 Yedekleme ve Arşivleme İşlemleri

Herhangi bir şekilde yedekleme ve arşivleme işlemi yoktur.

5. SİSTEM GERÇEKLEŞTİRİMİ

5.1 Giriş

Gerçekleştirim çalışması, tasarım sonucu üretilen süreç ve veri tabanının fiziksel yapısının içeren fiziksel modelin bilgisayar ortamında çalışan yazılım biçimine dönüştürülmesi çalışmalarını içerir. Yazılımın geliştirilmesi için her şeyden önce belirli bir yazılım geliştirme ortamının seçilmesi gerekmektedir.



Şekil 5.1 Sistem Gerçekleştirim Modeli

5.2 Yazılım Geliştirme Ortamları



Yazılım Geliştirme Ortamları şemada verilen değerlerden oluşmaktadır.

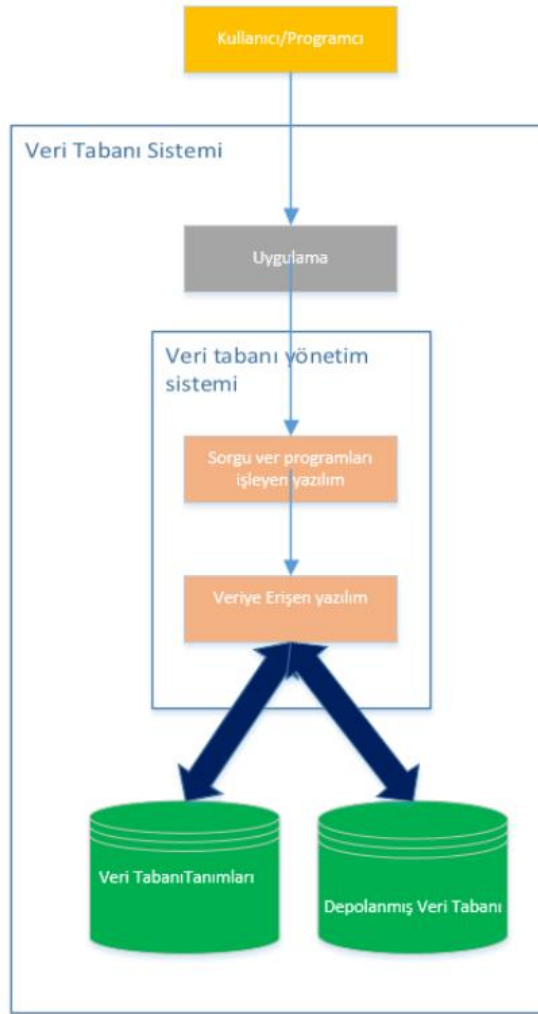
5.2.1 Programlama Dilleri

Birçok programlama dili kullanılmaktadır. Benim tercih ettim ise Java programı.

5.2.2 Veri Tabanı Yönetim Sistemleri

Birbiri ile ilişkili veriler topluluğu veri tabanı olarak tanımlanmaktadır. Veri tabanı herhangi bir boyutta ya da karmaşıklıkta olabilir.

Projede kullanmış olduğum veri tabanında veriler arasında ilişkiler yoktur. Ayrıca bir tablo ve içinde id, urun_adi, urun_kilo, son_kullanim_tarihi bilgilerini tutmaktadır.



Şekil 5.2.2 Veri Tabanı Sistemi

5.2.2.1 VTYS Kullanımının Ek Yararları

Uygulama yazılımı geliştirmenin kolaylaşması, fiziksel yapı ve erişim yöntemi karmaşıklıklarının her kullanıcıya yalnız ilgilendiği verilerin kolay anlaşılır yapılarda sunulması.

5.2.2.2 Veri Tabanı Dilleri ve Arabirimleri

Sistemde veri tabanı olarak SQL kullanıldı. Prototip aşamasında olan sistemimiz için MySQL arabirimi kullanıldı.

5.2.2.3 Veri Tabanı Sistem Ortamı

MySQL Notifer 1.1.8 i kullanıldı.

5.2.2.4 VTYS'nin Sınıflandırılması

İlişkisel veri modeli kullanılmadı.

5.2.2.5 Hazır Program Kütüphane Dosyaları

Kütüphane dosyası olarak jcalendar-1.4.jar, JDK 13 ve mysql-connector-java.8.0.19.jar kullanıldı.

5.2.2.6 CASE Araç ve Ortamları

Case araçları olarak Microsoft'un Visio ve Project ürünlerini kullandık.

5.3 Kodlama Stili

Herhangi bir düzene bağlı kod stiline bağlı kalınmadı.

5.3.1 Açıklama Satırları

Yazılan kod satırında örneğin güncelleme kısmında ilk olarak açıklama satırına bir örnek kullanıldı daha sonra o kod satırının ne işe yaradığı yazıldı.

5.3.2 Kod Biçimlemesi

Alt alta oluşan kodlar şeklinde yazdım ve iç içe biçimde hiyerarşi oluşturdum.

5.3.3 Anlamlı İsimlendirme

Nesne Tabanlı Programlama da isimlendirme kurullarını dikkate alarak sınıf, yöntem, değişken ve sabitleri tanımladım.

Örneğin class isimleri için büyük harf ile başladım. Dolap_islemleri gibi.

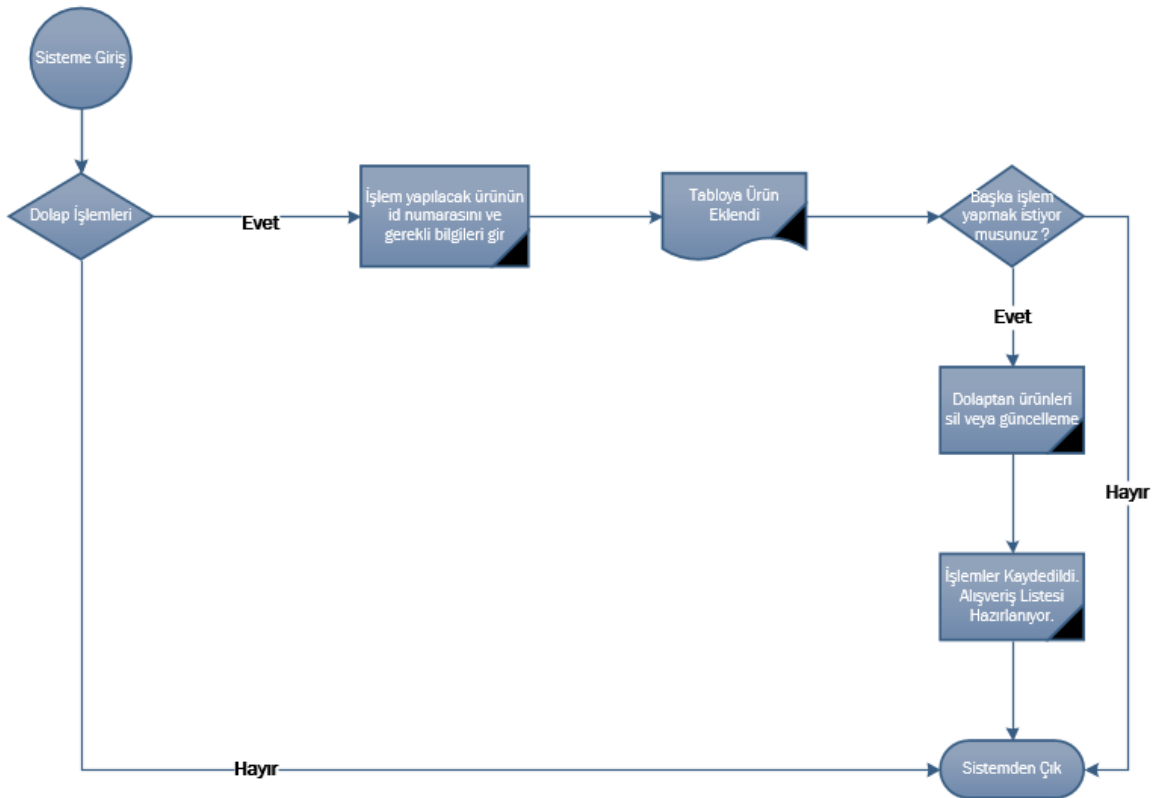
5.3.4 Yapısal Programlama Yapıları

Genel olarak 3 başlıkta incelersek:

- **Ardışık işlem yapıları**, fonksiyon, altprogram ve tekrarlı yapıları tek seferde çözmek için kullandım.
- **Koşullu işlem yapıları**, koşullu ifadelerle çok az yer verdim.
- **Döngü yapıları**, döngü yapılarına çok az yer verdim.

5.4 Program Karmaşıklığı

5.4.1 McCabe Karmaşıklık Ölçütü Hesaplama



$$V(G) = k - d + 2p$$

k: kenar sayısı = 9

d: düğüm sayısı = 9

p: bileşen sayısı = 1

$$V(G) = 9 - 9 + 2 \cdot 1 = 2$$

5.5 Olağan Dışı Durum Çözümleme

Olağan dışı durum program çalışırken yanlış veri oluşumu ya da başka nedenlerle istenmeyen bir biçimde sonlanmasına neden olan durumdur.

5.5.1 Olağan Dışı Durum Tanımları

Olağandışı gelişen durumlarda try-catch blokları devreye girecektir. Program çalışmasına böylelikle devam edecektir.

5.5.2 Farklı Olağan Dışı Durum Çözümleme Yaklaşımları

Tüm olağan dışı durumlar için hata mesajlarıyla tekrar başa dönecektir.

5.6 Kod Gözden Geçirme

Sürümleri gözden geçirildi. İncelendi ve gözden geçirildi.

5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi

Küçük bir ekip tarafından hataların bulunması ve düzeltilmesi yapılmıştır.

5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular

Program incelenirken, her bir öbeği (işlev) aşağıdaki soruların yanıtları aranır.

5.6.2.1 Öbek Ara Yüzü

Sorular:

1. Her öbek tek bir işlevsel amacı yerine getiriyor mu?
2. Öbek adı, işlevini açıklayacak biçimde anlamlı olarak verilmiş mi?
3. Öbek tek giriş ve tek çıkışlı mı?
4. Öbek eğer bir işlev ise, parametrelerinin değerini değiştiriyor mu?

5.6.2.2 Giriş Açıklamaları

Oluşturduğumuz giriş açıklamalarını test etmek için belli sorular sorduk bu sorular:

- Öbek, doğru biçimde giriş açıklama satırları içeriyor mu?
- Giriş açıklama satırları, öbeğin amacını açıklıyor mu?
- Giriş açıklama satırları, çıktıları (parametre, kütük vb.) ve hata iletilerini tanımlıyor mu?
- Giriş açıklama satırları, öbeğin algoritma tanımını içeriyor mu?
- Giriş açıklama satırları, öbekte yapılan değişikliklere ilişkin tanımlamaları içeriyor mu?
- Giriş açıklama satırları, öbekteki olağan dışı durumları tanımlıyor mu?
- Giriş açıklama satırları, Öbeği yazan kişi ve yazıldığı tarih ile ilgili bilgileri içeriyor mu?
- Her paragrafı açıklayan kısa açıklamalar var mı?

5.6.2.3 Veri Kullanımı

Oluşturduğumuz veri kullanımlarını test etmek için belli sorular sorduk bu sorular:

- İşlevsel olarak ilintili bulunan veri elemanları uygun bir mantıksal veri yapısı içinde gruplanmış mı?
- Değişken adları, işlevlerini yansıtacak biçimde anlamlı mı?
- Değişkenlerin kullanımları arasındaki uzaklık anlamlı mı?
- Her değişken tek bir amaçla mı kullanılıyor?
- Dizin değişkenleri kullanıldıkları dizinin sınırları içerisinde mi tanımlanmış?
- Tanımlanan her gösterge değişkeni için bellek ataması yapılmış mı?

5.6.2.4 Öbeğin Düzenlenişi

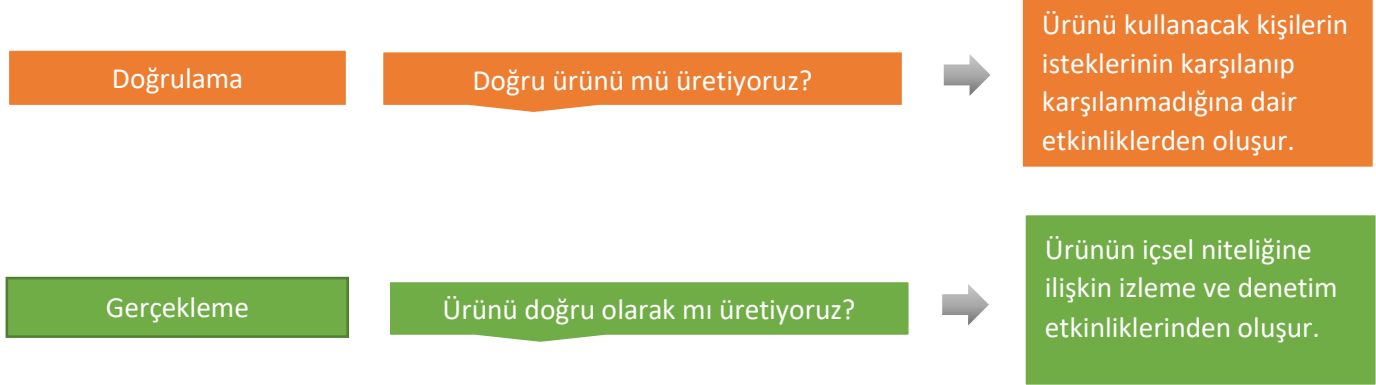
- Modüller birleşimi uyumlu mu?
- Modüller arası veri aktarımları sağlanıyor mu?
- Bütün modüller birleştiğinde sistem çalışıyor mu?

5.6.2.5 Sunuř

- Her satır, en fazla bir deyim ieriyor mu?
- Bir deyimden birden fazla satıra taşması durumunda, bölünme anlaşıla bilirliliđi kolaylařtıracak biçimde anlamlı mı?
- Koşullu deyimlerde kullanılan mantıksal işlemler yalın mı?
- Bütün deyimlerde, karmaşıklılıđı azaltacak şekilde parantezler kullanılmış mı?
- Bütün deyimler, belirlenen program stiline uygun olarak yazılmış mı?
- Öbek yapısı ierisinde akıllı “programlama hileleri” kullanılmış mı?

6. DOĞRULAMA VE GEÇERLEME

6.1 Giriş



6.2 Sınama Yöntemleri

6.2.1 Beyaz Kutu Sınaması

Denetimler arasında:

- Bütün bağımsız yolların en azından bir kere sınandı.
- Bütün mantıksal karar noktalarında iki değişik karar için sınanmalar yapıldı.
- Bütün döngülerin sınır değerlerinde sınanması yapıldı.
- İç veri yapılarının denemsi yapıldı.

6.3 Sınama Planlaması

6.3.1 Test Planı Kimliği

E-Dolap Test

6.3.2 Test Edilecek Sistem

E-Dolap sisteminde bulunan 4 tane form farklı değerler için test edilecektir.

6.3.3 Test Edilecek Ana Fonksiyonlar

Ürün ekleme, silme, güncelleme ve tabloda bunların listelenmesi.

6.3.4 Test Edilmeyecek Ana Fonksiyonlar

Test edilmeyecek ana fonksiyon yoktur.

6.3.5 Geçti/Kaldı Kriterleri

Sistemde herhangi bir hataya rastlanmadıysa geçti olarak sayılacaktır.

6.3.6 Test Dokümanı

Test süresince yapılan işlemleri alınan raporları elde edilen bilgileri rapor içinde sunulmalıdır.

6.3.7 Sorumluluklar

Proje tek kişilik olduğu için sorumluluk sadece tek kişiye aittir.

6.3.8 Riskler ve Önemler

Test planında varsayılan ve olası yüksek riskli durumlar için ekibe üye eklenecektir.

6.4 Sınama Belirtileri

Sınama Türü: Beyaz Kutu

Sınamayı Yapan: Tuba AKBAŞ

Sınama Tarihi: 11.05.2020

Bulunan Hatalar: Sistemde herhangi bir hataya rastlanmadı.

7. BAKIM

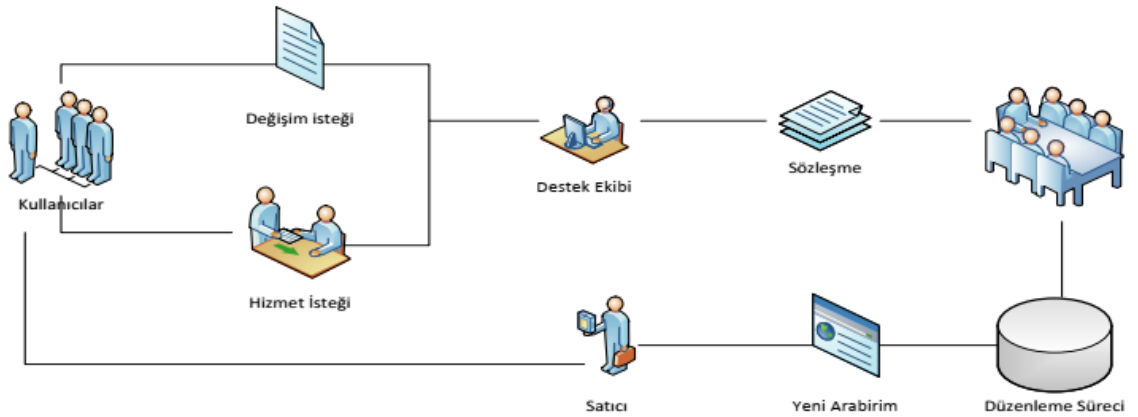
7.1 Giriş

Bakım bölümüne ilişkin yapılan açıklamalarda IEEE 1219-1998 standardı dikkate alınmıştır.

7.2 Kurulum

E-Dolap sistemi bir masaüstü uygulaması olduğu için uygulamanın indirilerek kurulumunun yapılması gerekmektedir.

7.3 Yazılım Bakımı



Şekil 7.2 Bakım Aşaması

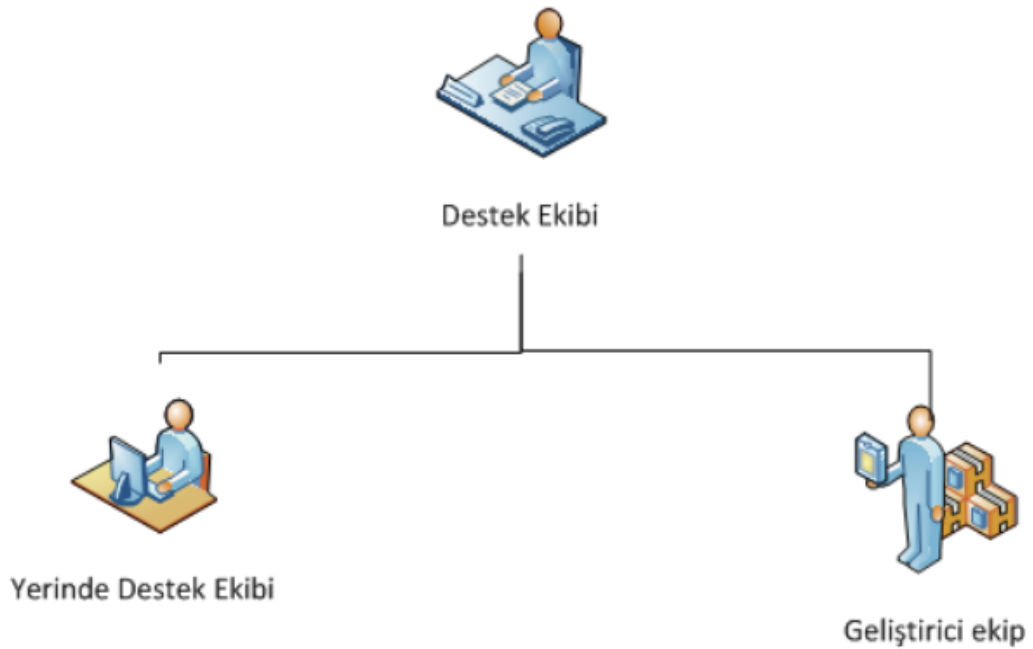
7.3.1 Tanım

Düzeltilici Bakım: Teorik olarak bir yazılımın tümüyle sınanabilmesi olası olsa bile, pratikte bu sağlanamaz. Bu nedenle çalışan bir yazılımda her an hata ile karşılaşma olasılığı vardır. Bu nedenle zaman zaman çalışan yazılımda ortaya çıkan hataların düzeltilmesi gerekir. Bu tür düzeltme çalışmaları Düzeltici Bakım olarak adlandırılır.

Uyarıcı Bakım: Uygulama yazılımları, işletme ya da kuruluşların günlük yaşamlarında yaptıkları işleri bilgisayar ortamında yapmalarını sağlayan araçlardır. Her kuruluş ya da işletme canlı bir varlık gibi düşünülebilir. Hiçbir işletme durağan değildir. Süreç içinde değişkenlik gösterir. İşletme ya da kuruluşlardan değişiklik yapılan işlerin yapılma tarzının değişmesi, yeni iş türlerinin ortaya çıkması biçiminde kendini gösterebilir. İşletme ya da kuruluşlarda yaşanan değişikliklerin o kuruluş işlerini bilgisayar yardımı ile yapmalarını sağlayan uygulama yazılımlarına da yansıtılması gerekir. Bu yansıtma işleminde Uyarıcı Bakım denir.

En İyileyici Bakım: Zaman zaman uygulama yazılımlarını çalışma performanslarını iyileştirilmesi amacıyla yapılan çalışmalar olarak tanımlanır.

7.3.2 Bakım Süreç Modeli



Şekil 7.3.2 Bakım Süreç Modeli

1.Adım: Sorun Tanımlama Süreci

İlk önce bakım ne için yapılıyor sorun ne buna bakılmalıdır.



2.Adım: Çözümleme Süreci

Sorun tanımlamadan çıkan karar doğrultusunda çözümlemelere başvurulur.



3.Adım: Tasarım Süreci

Çözümleme süresi bittikten sonra tasarım süreci başlayacaktır.



4.Adım: Gerçekleştirim Süreci

Tasarım süreci bittikten sonra sistemin gerçekleştirmesine geldi sıra yani kod kısmına.



5.Adım: Sistem Sınama Süreci

Değişikliklerin var olan yazılıma yansıtılmasından sonra elde edilen yeni yazılım sürümünün belirlenen standartlara uygun olarak tümüyle bütünlük sistem üzerinde sınamaların yapılması gerekmektedir.



6.Adım: Kabul Sınaması Süreci

Kullanıcılar ya da kullanıcı temsilcileri tarafından gerçekleştirilen bir süreçtir.



7.Adım: Kurulum Süreci

Kabul sınavasını geçen sistemimiz artık tekrardan kurulum aşamasına geçiyor.



8. SONUÇ

Sonuç olarak E-Dolap sistemi hayata geçirildiği zaman neler değişeceği gözler önüne serildi. Bunun yanı sıra sistemin kullanışı gayet basittir. Müşterilerin zorlanmaması açısından da bir kullanım kılavuzu olacaktır.

9. KAYNAKLARI

<https://www.youtube.com/watch?v=56hINJWKOX8&feature=youtu.be>
<https://www.youtube.com/watch?v=56hINJWKOX8&feature=youtu.be>
<https://www.youtube.com/watch?v=BXBGYOT5Wlw&feature=youtu.be>
<https://www.youtube.com/watch?v=n-OCbJZTPSk&feature=youtu.be>
<https://www.youtube.com/watch?v=EX8nrCcspQE&feature=youtu.be>
<https://www.youtube.com/watch?v=xo6ug5hJ9vE&feature=youtu.be>
<https://www.youtube.com/watch?v=QQmzkCg7skI&feature=youtu.be>
<https://www.youtube.com/watch?v=9-SEHVlrIGk&feature=youtu.be>
https://www.youtube.com/watch?v=Efor_q5Ukk8&feature=youtu.be
<https://www.youtube.com/watch?v=39C6oOBfxtQ&feature=youtu.be>
<https://www.youtube.com/watch?v=zStRO1XFZTU&feature=youtu.be>
<https://www.youtube.com/watch?v=M2ZZvcgDots&feature=youtu.be>
<https://www.youtube.com/watch?v=M7Kxx9qvM0k>
<https://www.youtube.com/watch?v=aXpQW1x6yCs&feature=youtu.be>