




**FIRAT ÜNİVERSİTESİ**  
**TEKNOLOJİ FAKÜLTESİ**  
Yazılım Mühendisliği

**YMT114 - Yazılım Mühendisliğinin Temelleri Dersi**  
Proje Uygulamaları ve Dokümantasyonu

**Projenin Adı**

E-vim 

**Proje Çalışma Grubu**

Hüseyin Yasin DÖNMEZ

Proje Yürütücüleri

Hüseyin Yasin DÖNMEZ

MAYIS –2020

# İÇİNDEKİLER

<b>1. GİRİŞ</b>	
Önsöz.....	07
1.1 Projenin Amacı.....	08
1.2 Projenin Kapsamı.....	08-09
1.3 Tanımlamalar Ve Kısaltmalar.....	09
<b>2. PROJE PLANI</b>	
2.1 Giriş.....	10
2.2 Projenin Plan Kapsamı.....	11-12
2.3 Proje Kaynakları.....	13-14
2.4 Proje Maliyet Kestirimi.....	15-17
2.5 Proje İş Zaman Planı.....	17
2.6 Proje Standartları, Yöntem Ve Metodolojiler .....	18
2.7 Proje Kalite Sağlama Planı .....	19
2.8 Kaynak Yönetim Planı.....	20
2.9 Eğitim Planı.....	20
2.10 Test Planı.....	21
2.11 Bakım Planı.....	21

<b>3. SİSTEM ÇÖZÜMLEME</b>	
<b>3.1</b> Mevcut Sistem Çözülmesi.....	22
3.1.1 Örgüt Yapısı.....	22
3.1.2 İşlevsel Model.....	22
3.1.3 Veri Modeli.....	22
3.1.4 Gereksinim Toplanması.....	23
3.1.4 Var olan Yazılım/Donanım Kaynakları.....	24
3.1.5 Var olan Sistemin Değerlendirilmesi.....	24
<b>3.2</b> Gereksenen Sistemin Mantıksal Modeli.....	24
3.2.1 Giriş.....	24
3.2.2 İşlevsel Model.....	25
3.2.3 Temel kullanım diyagramının açıklanması.....	26
3.2.4 Önerilen Yazılım Donanım Kaynakları.....	26
3.2.5 4 Önerilen Sistemin Değerlendirilmesi Sisteme Genel Bakış.....	26
3.2.6 Veri Modeli.....	26
3.2.7 Veri Sözlüğü .....	27
3.2.8 İşlevlerin Sıra Düzeni.....	28
3.2.9 Başarım Gerekleri .....	29
<b>3.3</b> Ara Yüz Modül Gerekleri.....	29
3.3.1 Yazılım Ara yüzü.....	29
3.3.2 Kullanıcı Ara yüzü.....	29
3.3.3 İletişim Ara yüzü.....	30
3.3.4 Yönetim Ar yüzü.....	30
<b>3.4</b> Belgeleme Gerekleri.....	30
3.4.1 Geliştirme Sürecinin Belgelenmesi.....	30
3.4.2 Eğitim Belgeleri.....	30
3.4.3 Kullanıcı El Kitapları.....	30

<b>4. SİSTEM TASARIMI</b>	
<b>4.1 Genel Tasarım Bilgileri</b>	31
4.1.1 Genel Sistem Tanımı	31-32
4.1.2 Varsayımlar ve Kısıtlamalar	32
4.1.3 Sistem Mimarisi	32
4.1.4 Sistemin Uml Diyagramı	33
4.1.5 Dış Arabirimler	34
4.1.5.1 Kullanıcı Arabirimleri	34
4.1.5.2 Veri Arabirimleri	34
4.1.5.3 Diğer Sistemlerle Arabirimler	34
4.1.6 Veri Modeli	34
4.1.7 Testler	34
4.1.8 Performans	34
<b>4.2 Veri Tasarımı</b>	35
4.2.1 Tablo tanımları	35
4.2.2 Tablo- İlişki Şemaları	36
4.2.3 Veri Tanımları	36
<b>4.3 Süreç Tasarımı</b>	36
4.3.1 Genel Tasarım	36
4.3.2 Modüller	37
4.3.2.1.1 Kullanıcı Arabirimi	37
4.3.3 Kullanıcı Profilleri	37
4.3.4 Entegrasyon ve Test Gereksinimleri	37
<b>4.4 Ortak Alt Sistemlerin Tasarımı</b>	37

## 5. SİSTEM GERÇEKLEŞTİRİMİ

<b>5.1.Giriş.....</b>	<b>38</b>
<b>5.2 Yazılım Geliştirme Ortamları.....</b>	<b>38</b>
5.2.1 Programlama Dilleri.....	38
5.2.2 Veri Tabanı Yönetim Sistemleri.....	39
5.2.3 Veri Tabanı Dilleri Ve Arabirimleri.....	39
5.2.4 Veri Tabanı Sistem Ortamı .....	39
5.2.5 Case Araç Ve Ortamları.....	40
5.2.6 Açıklama Satırları.....	40
5.2.7 Kod Biçimlemesi.....	40
5.2.8 Anlamlı İsimlendirme.....	40
5.2.9 Yapısal Programlama Yapıları.....	40
5.2.10 Hazır program kütüphane dosyaları .....	40
<b>5.3 McCabe Karmaşıklık Ölçütü Hesaplama.....</b>	<b>41</b>
5.3.1 McCabe Formülü.....	41
<b>5.4 Olağan Dışı durum Çözümleme.....</b>	<b>42</b>
5.4 .1 Olağandışı Durum Tanımları.....	42
5.4.2Farklı Olağandışı Durum Çözümleme Yaklaşımları.....	42
<b>5.5 Kod Gözden Geçirme.....</b>	<b>42</b>
5.5.1 Gözden Geçirme Sürecinin Düzenlenmesi.....	42
5.5.2 Gözden Geçirme Sırasında Kullanılacak Sorular.....	43
5.5.2.1 Öbek Ara yüzü.....	43
5.5.2.2 Giriş Açıklamaları.....	43
5.5.2.3 Veri Kullanımı.....	43
5.5.2.4 Öbeğin Düzenlenişi.....	44
5.5.2.5 Sunuş.....	44

<b>6. DOĞRULAMA VE GEÇERLEME</b>	
6.1 Sınama Planlaması.....	45
6.1.2 Test Edilecek Sistem.....	45
6.1.3 Test Edilecek Ana Fonksiyonlar.....	45
6.1.4 Test Edilmeyecek Ana Fonksiyonlar.....	45
6.2 Geçti Kaldı Kriterlerinin Belirlenmesi .....	45
6.3 Test Dokümanı.....	46
6.4 Sorumluluklar .....	46
6.5 Riskler ve Önlemler .....	46
6.6 Test İşlemleri .....	46
6.7 Sınama İşlemi Hakkında Bilgiler.....	47
<b>7. BAKIM</b>	
7.1 Bakım Standartı.....	48
7.2 Kurulum.....	49
7.3 Yerinde Destek Organizasyonu.....	49
7.4 Yazılım Bakımı.....	49
7.4.1 Bakım Süreç Modeli.....	50-53
<b>8. SONUÇ</b>	
8.1 Sonuç.....	54
<b>9. KAYNAKLAR</b>	
9.1 Yararlanılan Kitaplar.....	54
9.2 Yararlanılan İnternet Siteleri.....	54

## ÖN SÖZ

Bu çalışma Fırat Üniversitesi yazılım mühendisliği ödevi için gerçekleştirilmiş olan E-vimin yazılımın dokümanını içermektedir. Dokümanın amacı proje ekibin eksiksiz bir şekilde projede ilerlemesini ve kullanıcı ihtiyaçlarının tam olarak karşılanması için gerekli olan yazılım yaşam döngüsü fazlarının detaylı bir şekilde belirlenerek yönergeler halinde gerçekleştirmeyi ve bunun sonucunda tam anlamıyla başarılı bir ürün sunmak için gerekli dokümanların bir arada tutulmasını hedeflemektedir.

Çalışma boyunca 9.Bölümde yararlanılan kaynaklar kısmındaki belirtilen kaynakların sahiplerine sonsuz teşekkürlerimi iletiyorum.

# 1. GİRİŞ

## 1.1 Projenin Amacı

Gelişen teknolojiyle birlikte insan ihtiyaçları artmış, bu artışla insan ihtiyacının karşılanması ve yaşamının kolaylaşması hedeflenmiş yeni ürünler hizmetler ortaya çıkmıştır ve çıkmaya devam etmektedir. Örneğim: ürün bazında eski zamanlarda bir yazıcı, Mouse, bilgisayar; Çamaşır makinesi, bulaşık makinesi; Giyilebilir teknolojiler gibi teknolojilerden söz edilemiyordu veya bir koltuk takımı bir abajur bir masa lambası gibi ürünlerden bahsetmek imkânsızdı. Hizmet anlamında ise uzaktan fatura ödemeleri, uzaktan kart dolum gibi sanal işlemler söz etmek mümkün değildi fakat teknolojinin gelişmesi ve bunun akabinde bireylerin bu teknolojileri ulaşmasıyla birçok farklı sorun ortaya çıkmaya başladığı gözlemlenmektedir

Bu sorunlar: Alınan ürünlerin faturalarının kaybı, Ödenen makbuzların kaybedilmesi veya alınan ürünlerin garanti belgelerinin kaybedilmesi, ürünlerinin garantilerinin bitiği düşünülerek, bozulan ürünlerin atılarak bir teknoloji çöplüğüne dönüştürülmesi veya bu ürünlerin ücretli olarak tamir ettirilmesi gibi durumlar söz konusu olmaktadır bunun sonucunda doğa kirliliği ve ekonomik kayıp meydana gelmektedir. Aynı zamanda internet üzerinden yapılan ödemelerde ve alışverişlerde bir takım sorunlar ortaya çıkmaktadır sonuçta sanal olan bir şeyin yüzde yüz korunması ve saklanma ihtimali yoktur sanal sistemlerde gerek güvenlik açıklarından dolayı gerekse donanım bileşenleri bakımından birçok istenmeyen durum karşılaşılabilmekte ve bunun sonucun birçok kurum veya kişisel baz da maddi zararlar meydana gelmektedir.

Yukarıda bahsedilen sorunlarla yaşanmaması için çalışmalara başladım ve E-vim adlı projemi geliştirmeyi hedeflemekteyim.



## 1.2 Kapsam

E-vim adlı projede bireysel kullanıcılarının ödemiş oldukları faturalar , almış olunan ürünlerin faturalarıyla aynı zamanda alınan ürün ve hizmetlerin garanti belgelerine dair bilgileri bir arada tutarak ödenmiş faturaların kurum kaynaklı ödenmedi görünmesi ve bunun sonucunda kullanıcının zarara uğraması. Kullanıcıların henüz garantisi bitmemiş ürünlerinin bozulması durumunda garanti belgesine ulaşamadığından veya aramakla uğraşmak istemediğinden garantisinin bitmiş olduğunu düşünüp ücretiz olarak tamir ettirememesi, telefonlarındaki şifrelerin unuttuklarında garanti belgelerinin kayıp olmasından dolayı bir daha telefonlarını kullanamaması veya garantisi bitmemiş ürünün bozulduğunda garantisi bitmiş olduğu düşünülerek teknoloji çöplüğüne göndererek hem maddi hem de doğal çevreye zarar vermesi durumlarına çözüm olmak amacıyla bu belgelerin ve faturaları bilgilerinin hem metinsel veri olarak bir arada tutulması hem de fatura ve garanti belgelerin görsellerinin bir arada tutularak kullanıcıların bu gibi durumlarda karşılaşacağı zorlukları minimize etme doğaya ve dünya ekonomisine olan zararı engellemektir. Projemiz sadece bireysel ev kullanıcılara hitap etmekte olduğu gibi Kurum ve Kuruluşların kullanmasında da bir engel yoktur.

## 1.3 Tanımlamalar ve Kısaltmalar

**E-vim:** Hazırlanan Projenin Adı

**VTYS:** Veri tabanı yönetim sistemi

**MYSQL:** projede kullanılan veri tabanı yönetim sistemi

**Use-case:** Kullanıcı durum diyagramı

**İN:** İşlevsel Nokta

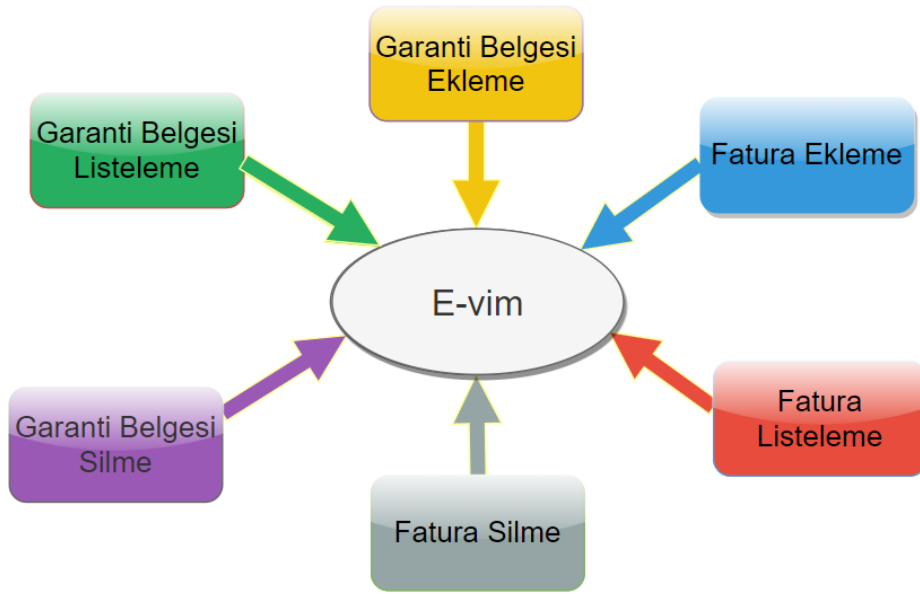
**TKF:** Teknik Karmaşıklık Faktörü

**AİN:** Ayarlanmamış işlevsel nokta

## 2. PROJE PLANI

### 2.1 Giriş

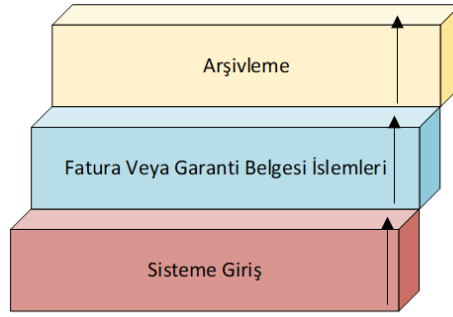
E-vim Kullanıcıların fatura ve garanti belgelerinin bilgilerini, fatura dekontlarını ve garanti belgelerini görsel bir içeriğini sistemde muhafaza altına almayı garanti eden kullanıcıların gerektiğinde bu belge ve bilgilere daha kolay ve hızlı bir şekilde ulaşmasını sağlayan bir program olarak planlanmaktadır



Şekil 2.1 Projenin genel yapısı

## 2.2 Projenin Plan Kapsamı

Projenin plan kapsamında genel olarak mevcut sistem, sistemin gerekliliği ve hali hazırda benzer projelerinin yetersizliği göz önünde bulundurularak yola çıkıldı. E-vim uygulaması her türlü fatura ve garanti belgesi dokümanını bir elektronik ortamda saklanması sağlar. E-vim ile faturalar, garanti belgeleri elektronik sistemde tutulduğu için bu belgelere erişmek istendiği zaman daha hızlı ve kolay bir şekilde, vakit kaybı olmadan erişilebilecektir.



Şekil 2.2 Sistemin genel işleyişi

### E-vim Neden Gerekli?

Birçok ev kullanıcısı aldıklarını ürünlerinin garanti belgelerini, ödedikleri faturaların makbuzlarını evlerinde saklamaktadır. Bu saklama sırasında kullanıcılar aldıkları ürünlerin garanti belgelerini, faturaların makbuzlarını nereye koyduklarına pek dikkat etmezler bunun sonucunda ilerleyen zamanlarda bir takım problemlerle karşıya karşıya gelme ihtimali artmaktadır. Bu problemler genellikle kurumlardan kaynaklı hatalar olmaktadır. E-vim, kullanıcıların bu problemlerinin önüne geçmek ve önlemesini sağlamaktadır

## Kullanıcıların Yaşadıkları Sıkıntılara Dair Görseller

### İPhone Telefon Kilitlenmesi Sorunu!

 **Hakan** · 2 hafta önce · Şu anda 49 kişi, seninle birlikte iPhone şikayetlerini okuyor.

İPhone 6 telefonumun iCloud şifresini unuttuk mail kurtarım için sorulan sorulara da doğru cevap veremiyoruz. Fatura istediler faturası kayboldu yeni fatura çıkartmak için telefonu aldığımız Turkcell bayiine gittim 2 yıl önce yangın çıkmış hiçbir şekilde fatura veremiyorlarmış kayıtlarının hepsi yok olmuş ama Apple şirketinden telefonu açmak için ne istiyorlarsa resmi bir şekilde yazı hazırlayıp onların tarafına gönderebileceklerini söylediler. Müşteri hizmetlerini aradım aynı şekilde beyan ettiğim halde hala daha bana faturasını bulmadan telefonu açmayacaklarını söylüyorlar. Telefonumun çalıntı olmadığını kullanan kişinin de her türlü telefonun bize ait olduğunu ispat edebilmemize rağmen sırf yeni telefon almamız için şirketin böyle bir şey yapması mümkün değildir diye ıvır zıvır söylüyorlar. Bu geçirdiğimiz zor günlerde bu şekilde mağdur kalmak gerçekten üzücü.

### Arzum Garanti Belgesi Olmadan İşlem Yapmıyor

 **Uğur** · 04 Mart 2015 08:50 · Şu anda 55 kişi, seninle birlikte Arzum şikayetlerini okuyor.

Ar-346 Çay Makinesi aldım. Alalı henüz 6 ay oldu; fakat garanti belgesi ile faturasını kaybettim. Ürünün açma kapama düğmesi arızalandı. Servise götürdüm ve servis garanti kapsamına girdiğini söyledi. Ancak garanti belgesi olmadığı için ücretsiz tamir yapamayacakmış. Müşteri hizmetlerini aradığım zaman 3 yıl garanti süresi olan ve daha 1 yıldır belki satışta olan ürünün garantisinin olup olmadığını teyit edemeyecekleri şeklinde saçma bir cevap aldım ve bana herhangi bir yardımda bulunmadılar. Ürünü aldığım yer ile irtibat kurmam gerektiğini söylediler, bende tüm uğraşlarıma rağmen şu anki ikametimden 300 km uzaktaki mağazaya ulaşamadım. Ürün arızalı bir şekilde elimde kaldı, sorun çözülmedi

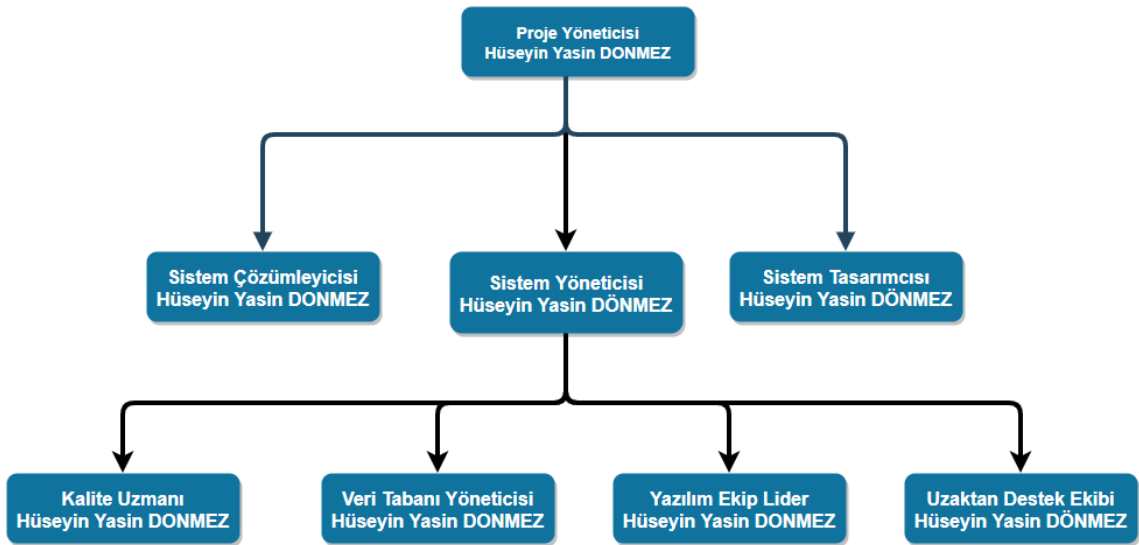
3.601 Görüntüleme

### E-vim Nasıl Çalışır

E-vimin çalışma mantığı gayet basittir. Kullanıcıdan, aldığı bilgileri bir veri tabanına kayıt eder veya kullanıcın aradığı veriyi kullanıcıya bir sorgu ile sunar.

## 2.3 Proje Kaynakları

### 2.3.1 İnsan Kaynakları



Şekil 2.3 İnsan Kaynakları Şablonu

### 2.3.2 Donanım Kaynakları:

- 1) Ana bilgisayarlar
- 2) Kullanıcı Bilgisayarları
- 3) Yerel Alan Ağı (LAN) Alt Yapısı
- 4) Veri tabanı

### 2.3.3 Yazılım Kaynakları

Çözümleme Ve Tasarım Araçları
Microsoft Visio Microsoft Project Adobe Photoshop diagrams.net

Destek Araçları
Google Chrome Mozilla Firefox İnternet Explore

Sınama Araçları
Windows 7 Windows 8.1 Windows 10

Programlama Araçları
Netbeans Mysql Jdk Sdk

## 2.4 Maliyet Kestirimi Ve Hesaplanması

### 2.4.1 Ana İşlem Nokta Sayısının Hesaplanması

Ölçüm parametresi	Sayı	Ağırlık	Toplam
Kullanıcı Girdi Sayısı	3	8	24
Kullanıcı Çıktı Sayısı	3	8	24
Kullanıcı Sorgu Sayısı	3	7	21
Kütük Sayısı	2	7	14
Dışsal Ara yüz Sayısı	0	0	0
Ana İşlem Nokta Sayısı			83

**Şekil 2.4 İşlem Nokta Sayısını Hesaplamak için kullanılan Tablo**

## 2.4.2 Teknik Karmaşıklık Faktörünün Hesaplanması

Teknik Karmaşıklık Sorusu	Puan
1.Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu?	3
2.Veri iletişimi gerekiyor mu?	3
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?	5
7.Çevrim içi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu?	3
8.Ana kütükler çevrim-içi olarak mı güncelleniyor?	3
9.Girdiler, çıktılar, kütükler ya da sorgular karmaşık mı?	3
10.İçsel işlemler karmaşık mı?	2
11.Tasarlanacak kod, yeniden kullanılabilir mi olacak?	3
12.Dönüştürme ve kurulum, tasarımda dikkate alınacak mı?	4
13.Sistem birden çok yerde yerleşik farklı kurumlariçin mi geliştiriliyor?	0
14.Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak?	4
TOPLAM	42

Şekil 2.5 teknik karmaşıklık faktörünün hesaplanması için kullanılan tablo

**0:**Hiçbir Etkisi Yok

**1:**Çok Az etkisi var

**2:**Etkisi Var

**3:**Ortalama Etkisi Var

**4:**Önemli Etkisi Var

**5:**Mutlaka Olmalı, Kaçınılamaz



## 2.4.3 Temel Maliyet Modeli – COCOMO

$İN = AİN \times (0.65 \times 0.01 \times TKF)$  İşlevsel nokta sayısını hesaplamak için gerekli formül

$$İN = 83 \times (0.65 \times 0.01 \times 42)$$

$$İN = 22,66$$

**Satır Sayısı** =  $İN \times 30$  (nesne tabanlı Programlama yapılacağı için standart gereği 30 ile çarpılmıştır)

$$\text{Satır Sayısı} = 22,66 \times 30 = 678$$

Yazılımda kullanılması öngörülen **toplam satır sayısı yaklaşık 680 satirdir**

**Projenin turu:** Ayrık Proje

**Ayrık projelerde hesaplama**

$$\text{İs gücü } K = 2.4 \times S^{1,05}$$

$$\text{İs gücü} = 2.4 \times (0,680)^{1,05} = 1.672$$

$$\text{Zaman } T = 2.5 \times (K)^{1,05}$$

$$\text{Zaman} = 2.5 \times (1,672)^{0,38} = 3,03 \text{ ay}$$

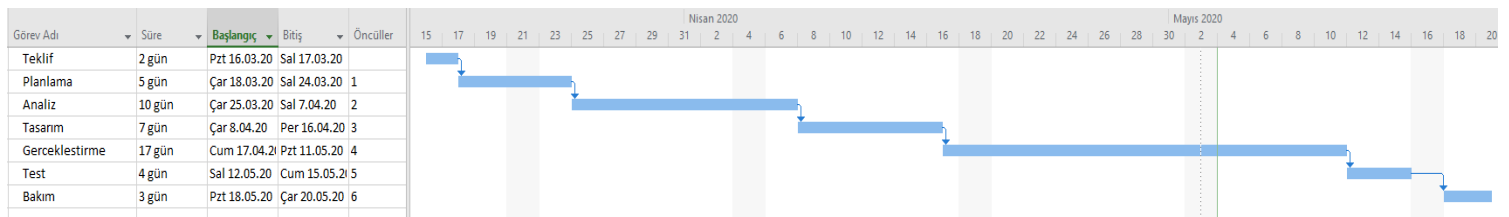
**S:bin türünden satır sayısı**

$$\text{Tahmini Gerekli Personel} = 3.03 / 1,672 = 2$$

## 2.5 Proje İş Zaman Planı

İş - Zaman Çizelgesi										
Zaman \ İş	16.03.2020-23.03.2020	23.03.2020-30.03.2020	30.03.2020-06.04.2020	6.04.2020-13.04.2020	13.04.2020-20.04.2020	20.04.2020-27.04.2020	27.04.2020-04.05.2020	04.05.2020-11.05.2020 Hafta	11.05.2020-18.05.2020	18.05.2020-20.05.2020
Proje Teklifi	✓									
Proje Planı	✓	✓								
Proje Analizi		✓	✓	✓						
Sistem Tasarımı				✓	✓					
Gerçekleştirme					✓	✓	✓	✓		
Test							✓		✓	
Bakım									✓	✓
Sunum										✓

Şekil 2.3 Proje İş Planı



Şekil 2.4 Proje Gantt diyagramı

## 2.6 Proje Standartları, Yöntem ve Metodolojiler

### 2.6.1 Projede Kullanılan Süreç Model: Spiral Model(Helezonik Model)

Spiralin başladığı ilk çeyrek içinde ilk isterler toplanır ve buna göre proje planlaması yapılır. İkinci çeyrekte, ilk tanımlanan isterler göre risk çözümlemesi yapılır. Üçüncü çeyrekte, risk çözümlemesi sonunda ortaya çıkan isterlerin tanımlanmasındaki belirsizlikleri ortadan kaldırmak için prototipleşme yöntemi kullanılır. Gerekirse benzetim(simülasyon) veya diğer modelleme kullanılarak isterlerin daha sağlıklı tanımlanması sağlanır. Dördüncü çeyrekte, müşteri, ortaya çıkan ilk ürünü inceleyerek değerlendirme yapar, önerilerde bulunur. Bu şekilde tanımlanan ilk döngü bir sonraki döngü için bir girdi oluşturur. Hataların en minimize edilmesini sağlayan bu yöntem nedeniyle bu projede spiral model kullanılmıştır

Aşama	Kullanılan Yöntem/Araçlar	Ne İçin Kullanıldığı	Çıktı
Planlama	<ul style="list-style-type: none"><li>- Veri Akış Şemaları,</li><li>- Süreç Belirtileri,</li><li>- Görüşme,</li><li>- Maliyet Kestirim Yöntemleri</li><li>- Proje Yönetim Araçları</li></ul>	<ul style="list-style-type: none"><li>- Süreç İnceleme</li><li>- Kaynak Kestirimi</li><li>- Proje Yönetimi</li></ul>	Proje Planı
Çözümleme	<ul style="list-style-type: none"><li>- Süreç Belirtileri,</li><li>- Veri Akış Şemaları,</li><li>- Görüşme,</li><li>- Nesne İlişki Şemaları,</li><li>- Veri Sözlüğü</li></ul>	<ul style="list-style-type: none"><li>- Süreç Çözümleme</li><li>- Veri Çözümleme</li></ul>	Sistem Çözümleme Raporu
Çözümlemeden Tasarıma Geçiş	<ul style="list-style-type: none"><li>- Akışa Dayalı Çözümleme,</li><li>- Süreç Belirtilerinin Program Tasarım Diline Dönüştürülmesi</li><li>- Nesne İlişki Şemalarının Veri Tablolarına Dönüştürülmesi</li></ul>	<ul style="list-style-type: none"><li>- Başlangıç Tasarım</li><li>- Ayrıntılı Tasarım</li><li>- Başlangıç Veri Tasarımı</li></ul>	Başlangıç Tasarım Raporu
Tasarım	<ul style="list-style-type: none"><li>- Yapısal Şemalar</li><li>- Program Tasarım Dili</li><li>- Veritabanı Tabloları</li><li>- Veri Sözlüğü</li></ul>	<ul style="list-style-type: none"><li>- Genel Tasarım</li><li>- Ayrıntılı Tasarım</li><li>- Veri Tasarımı</li></ul>	Sistem Tasarım Raporu

Şekil 2.5 proje aşamaları

## 2.7 Kalite Saęlama Planı

**1)Ekonomi:** Ekonomik açıdan yazılım maliyeti yok denecek kadar aza indirgenmelidir kullanılan veri tabanı ve yazılım ortamı ücretsiz olması sağlanacaktır

**2)Tamlık:** Projede herhangi bir açık ve eksiklik olmamasına özen gösterilecektir

**3)Yeniden Kullanılabilirlik:** Yazılan yazılım her koşulda yeniden kullanılmaya elverişli olarak düzenlenecektir

**4)Etkinlik:** Kullanıcının kullanımının kolay olması amaçlanacaktır.

**5)Bütünlük:** Kullanıcı Sistemi tam anlamıyla aktif bir şekilde kullanabilmelidir.

**6)Güvenilirlik:** Yazılan yazılım ve entegre edilen sistem kullanıcıya tavsiyeler verilerek ve açıkları kapatarak tamamıyla güvenilir hale getirilmeli belli yedekleme işlemleri yapılmalıdır.

**7)Belgeleme:** Hem kullanıcı için hem de yazılım ekibi için gerekli belgeler temin edilecektir.

**8)Kullanılabilirlik:** Kullanıcıyı rahatsız edecek renklerden ve kullanımı zor kullanışlı bir ara yüzden kaçınılmalıdır

**9)Temizlik:** Kullanıcının gerektiğinde istenmeyen verilerden kurtarılması imkanı sağlanacaktır.

**10)Deęiştirilebilirlik:** Kullanıcıya rahatça istedięi verilerde deęişiklik yapabilme imkânı sağlanacaktır

**11)Esneklik:** Proje Kullanılan programlama dili platformdan bağımsız olduęu için farklı platformlarda çalışabilmelidir

**12)Genellik:** Genellikle Bireysel ev kullanıcılarına hitap etmesini öngörülmektedir

**13)Sınana bilirlik:** Projedeki test kısmı içermeli ve birkaç prototipi kullanıcılara dağıtılıp geri dönüşe göre programın aksak yönleri düzeltilmelidir

**14)Taşınabilirlik:** Sistem dosyasının çok küçük olması ve Bir internet sitesi üzerinden kullanıcılara dağıtılmasından dolayı taşınabilirliği kolay olması amaçlanmalıdır.

## 2.8 Kaynak yönetim planı:

Bir öğrenci projesi olduğu için herhangi bir kaynağımız bulunmamaktadır

## 2.9 Eğitim Planı

Eğitim planlaması önemi büyük bir önem arz etmektedir program kullanımı kolay olmasından dolayı eğitim içeriği fazla sürmeyecektir fakat kullanıcıların programı nasıl

Kullanacağını bilmemesinden dolayı kullanıcının programı kullanmayı reddetme gibi bir olası senaryo ile karşılaşılmasında için bir program kullanım kitapçığı aynı zamanda programın kullanımının videolu bir şekilde anlatılması planlanmıştır. Ekip bazında ise hali hazırda ekibin yeterli derecede eğitime sahip olmasından dolayı ekip bazında bir eğitim planlanmamıştır.

Eğitimin İçeriği	Süre
Programın Ara Yüzünün Tanıtılması	5 Dakika
Veri Eklerken Dikkat Edilmesi Gereken Unsurların Kullanıcıya aktarılması	5 Dakika
Veri Silerken Dikkat Edilmesi Gereken Unsurların Kullanıcıya aktarılması	5 Dakika
Veri Sorgularken Dikkat Edilmesi Gereken Unsurların Kullanıcıya aktarılması	5 Dakika
Verilere Ait Resimlerin Kayıt Edilirken Dikkat Edilmesi Gereken Unsurların Kullanıcıya aktarılması	5 Dakika
Toplam Süre	25 Dakika

**Şekil 2.6 Eğitim içeriğinin planı**

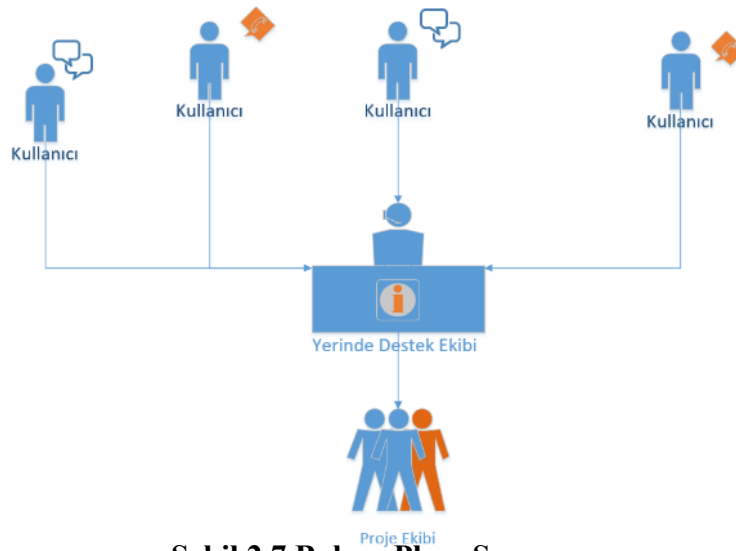
## 2.10 Test Planı:

**Proje Test Planı Şu Şekildedir.**

- Her bir kayıt alanı farklı değerler için denenmelidir
- Her bir kayıt alanının belirli değerler haricinde girişine izin verilmemelidir
- Her bir dosya uzantısı kutucuğunun dosya dizinin yeri belirtilmelidir
- E-vime her bir tablo için 10 veri kayıt edilip ara yüzdeki değişimlerin olup olmadığını kontrol edilmelidir. Bu 10 deneme verisi üzerinden projenin kullanıcıya sunduğu hizmetlerin doğru çalışıp çalışmadığı denetlenmelidir.
- Ekleme, silme, güncelleme gibi durumlarda bir hata olup olmadığı kontrol edilmelidir
- Veri tabanındaki alanları değerleriyle kayıt için gereken değerlerin türlerinin uyuşup uyuşmadığı kontrol edilmelidir.

## 2.11 Bakım planı

Geri dönüş ekibine kullanıcılardan gelen geri dönüş ve istekler doğrultusunda geri dönüş ekibinin yazılım ekibiyle iletişime geçmesi sağlanıp bakım planları Kullanıcıdan gelen uygun geri dönüşler ve isteklerin olması durumunda bildirim süresinden itibaren 30 iş gününü geçmemek şartıyla projenin bakım çalışmaları yapılacaktır.



**Şekil 2.7 Bakım Planı Şeması**

## 3.Sistem Çözümleme

### 3.1 Mevcut Sistemin Çözülmesi

Mevcut bankaların mobil bankacılık aracıyla ödenen faturaları Sistemlerinde tutukları bilinmektedir fakat Sistemde tutulan faturaların ne kadar süreyle ve nasıl tutulduğu hakkında bir bilgi yoktur. Piyasa araştırılması yapıldığında Evim adlı projenin çok benzeri olmamakta birlikte var olan sistemlerde mobil sistemlerdir ve bir internet bağlantısı ve hesap oluşturulması gerekmektedir. Bu yüzden kullanıcılar tarafından pek tercih edilmediği gözlemlenmiştir

#### 3.1.1 Mevcut Sistemin Örgüt yapısı

Mevcut sistemlerin örgüt yapısı tam anlamıyla public bir yapı olmadığı için yapı incelenememiştir

#### 3.1.2 Mevcut İşlevsel Yapısı

Mevcut sistemlerin yapıları tam anlamıyla bilinmediği için sadece bir varsayım üzerinden çıkarımlar yapılacaktır. Mevcut sistemler kullanıcıların verilerini internet aracılığıyla uzaktaki bir sunucuda sakladıkları ve tablo yapısı olarak kullanıcıların bulunduğu aynı zamanda bir yönetici panelinin bulunduğu bilinmektedir.

#### 3.1.3 Mevcut Sistemin Veri Modeli

Herkese açık bir mevcut sistem yazılımı bulunamadığı için veri model yapısı incelenememiştir.

### 3.1.4 Gereksinim Toplanması

Kullanıcılara aşağıdaki sorular yönetilerek ilgili gereksinimler toparlanmalıdır.

Kategori	Sorular
Donanım	Mevcut Donanımın geliştirilmesi veya yenilenmesini düşünüyor musunuz?
Donanım	Mevcut yazılım ortamından hangi işletim sistemleri kullanılacaksınız?
Donanım	Mevcut sisteminizde hangi anti virüs programını kullanıyorsunuz?
Donanım	Mevcut sisteminizde herhangi bir donanımsal problem mevcut mu?
Yazılım	Şuan sistemsizde hangi uygulamaları kullanmaktasınız?
Yazılım	Daha önce bu uygulamaya benzer uygulamalar kullandınız mı?
Yazılım	Uygulamanın Çalışmasını İstedığınız Platformlar neler?
Yazılım	Uygulamada olmasını istediğiniz kısımlar neler?
Yazılım	Sizin için uygulamanın güvende olması ne kadar önemli?
Yazılım	Uygulama için olmazsa olmaz dediğiniz kısımlar var mı?
Yazılım	Uygulamadaki verilerin yedeklenmesine ihtiyaç duyacak mısınız?
Yazılım	Uygulamayı Hangi Koşullarda kullanacaksınız?
Yazılım	Uygulamanız için hız önemli mi?
Yazılım	Uygulamadaki verilerinizi gerektiğinde yazıcıdan çıktı almak ister misiniz?
Ara yüz	Ekran tasarımının nasıl olmasını istersiniz(Rengi, düzeni) ?
Ara yüz	Ekranın boyutları değiştirilmeye açık olmalı mı?
Ara yüz	Uygulamada kullanılmasını istediğiniz logonuz var mı?
Ara yüz	Uygulamanızın Görünümü sizin için ne kadar önemli?
İnternet	Sistemin internet üzerinden güncellenebilir olmasını ister misiniz?
İnternet	Uygulamanızdaki verilere internet üzerinden erişebilmeyi istiyor musunuz?
İnternet	Uygulamanızı kullanırken internet bağlantısı gereksinimi olmalı mı?

**Tablo 3.0 Gereksinim Analizi İçin İlgili Sorular**

### **3.1.5 Var Olan Yazılımlar Ve Donanım Kaynakları**

Mevcut Sistemde Elektronik Sertifika Sağlayıcılar, uzak bir sunucu ve kullanıcılar ile sunucuların haberleşmesini sağlayan çeşitli apiler bulunduğu varsayılmaktadır.

### **3.1.6 Var Olan Yazılım Değerlendirilmesi**

Daha Öncede yukarıda belirtildiği gibi var olan sistem bir internet bağlantısı gerektirmekte ve aynı zaman sistem kullanıcıların sisteme bir kullanıcı adı ve parolayla girmelerini istemektedir ve buda bireysel ev kullanıcıları için sorunlar teşkil etmektedir. Var olan sistemin Verileri kullanıcılarının kendi cihazlarında saklamadığı gözlemlenmiş ve buda kullanıcılar tarafından olumsuz bir yön olarak görülmektedir.

## **3.2 Gereksenen Önerilen Sistemin Mantıksal Modeli**

### **3.2.1 Giriş**

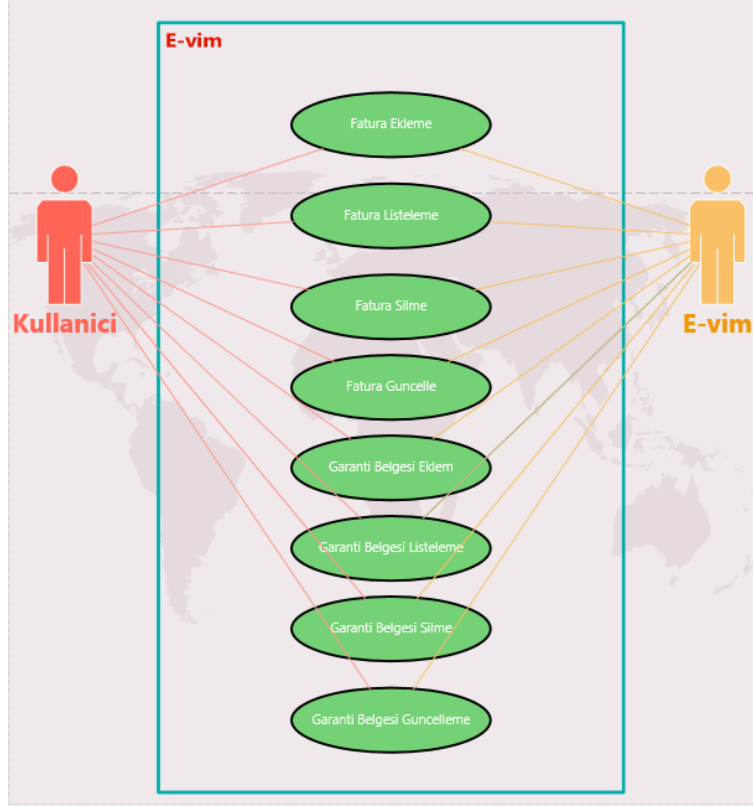
Mevcut sistemler incelenmesi sonucunda mevcut sistemlerin birçok karmaşık yapıda olduğu ve kullanıcıların ara yüzleri tam yeterli bir şekilde kullanamadığı görülmüştür. Mevcut sistemler kullanıcının verilerini uzak bir sunucuya kayıt etmektedir buda kullanıcılar tarafından rahatsızlığa sebep olmaktadır. Bir başka sorunda mevcut sistemleri kullanabilmek için kullanıcının bir internet trafiğine sahip olması gerekmesidir. Ayrıca mevcut sistemlerin sunucuları ve sahipleri 3.şahıs kişiler olduğu 3.şahıs kişilerin olası yazılımı kullanımdan kaldırması durumunda birçok verisi sisteme kayıtlı olan kullanıcının mağdur olacağı izlemi oluşmuştur. E-vim çevrim dışı olarakta çalışabilme imkanının olması , kolay bir kullanım sunması için bir admin paneli oluşturulmaması aynı kullanıcıların güvenliği için kullanıcı verilerinin kendi cihazlarında saklanması önerilmektedir.

### **3.2.1 Örgüt yapısı**

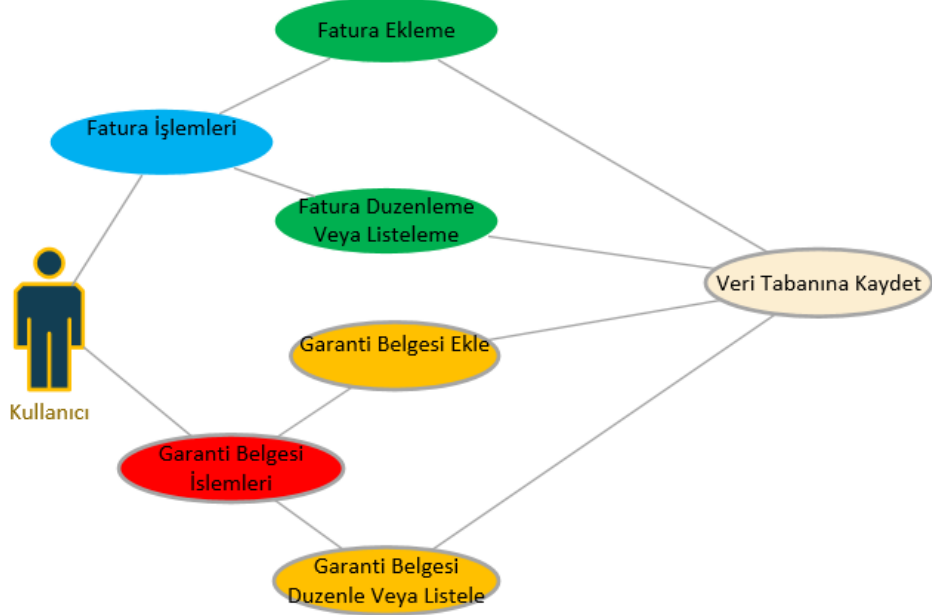
Örgüt yapısı olarak Ayyıldız yazılım bünyesindeki personellerin ortak çalışmasıyla oluşan bir yapıdır



### 3.2.2 Sistemin İşlevsel yapısı



Şeki 3.1 Kullanıcı diyagramı



Şekil 3.2 Temel Kullanım Diyagramı

### 3.2.3 Temel Kullanım Diyagramının Açıklanması

1)Kullanıcı E-vimi çalıştırır Fatura İşlemleri veya Garanti Belgesi İşlemleri seçeneklerinden birini seçer

2)Eğer seçilen seçenek fatura işlemleri ise kullanıcıya iki kısım sunulur fatura ekleme ve fatura düzenleme veya listeleme kullanıcı işlem yapmak istediği seçeneği ekler. Açılan pencerede E-vime gerekli bilgileri girdikten sonra yapılan işlem veri tabanına kayıt edilir

3) Eğer seçilen seçenek garanti belgesi işlemleri ise kullanıcıya iki kısım sunulur Garanti Belgesi ekleme ve Garanti belgesi düzenleme veya listeleme kullanıcı işlem yapmak istediği seçeneği ekler. Açılan pencerede E-vime gerekli bilgileri girdikten sonra yapılan işlem veri tabanına kayıt edilir

### 3.2.4 Önerilen Yazılım Donanım Kaynakları

- MS Word
- MS Excel
- MS Visio
- Mysql
- Netbeans
- Kullanıcı Bilgisayarları

### 3.2.5 Önerilen Sistemin Değerlendirilmesi Sisteme Genel Bakış

Sistem çevrimdışı zamanlıdır herhangi bir internet bağlantısı gerekmez. Sistemin veri tabanı kullanıcı bilgisayarındadır. Sistemin çevrimdışı zamanlı olmasından dolayı herhangi bir kullanıcı ve şifre tablosu yoktur. Bu durum bireysel ev kullanıcıları için bir güvenlik tehdidi oluşturmadığı düşünülmüştür

### 3.2.6 Veri Modeli

E-vimde herhangi bir ilişkisel veri tabanı bulunmamaktadır bunun sebebi program bireysel ev kullanıcıları hitap etmektedir. Birçok kullanıcı şifrelerini unuttuğu için herhangi bir kullanıcı ve şifre tablosu oluşturulmamıştır. Faturalar ve garanti belgesi tabloları da birbirinden bağımsızdır.

### 3.2.7 Veri Sözlüğü

E-vime ait veri sözlüğü aşağıdaki tabloda gösterilmiştir. E-vim iki adet tabloya sahiptir ve bu tablolar birbirinden bağımsızdır.

Kayıt no kısımları primary key olduğundan boş geçilemez olarak belirlenmiş diğer kısımlarda herhangi bir kısıtlama yapılmamıştır

Fatura Dekontunu ve Urunun faturası kısımlarındaki uzunluklar, bu alanların kullanıcıların fatura veya garanti belgelerinin yollarını göstereceği için 100 olarak ayarlanmıştır

Tarih kısımları için herhangi bir uzunluk belirtilemediği için bir uzunluk ifadesi girilememiştir

Tarih kısımlarının genel formatı yıl-ay-gün şeklindedir örneğin: 2020.05.04

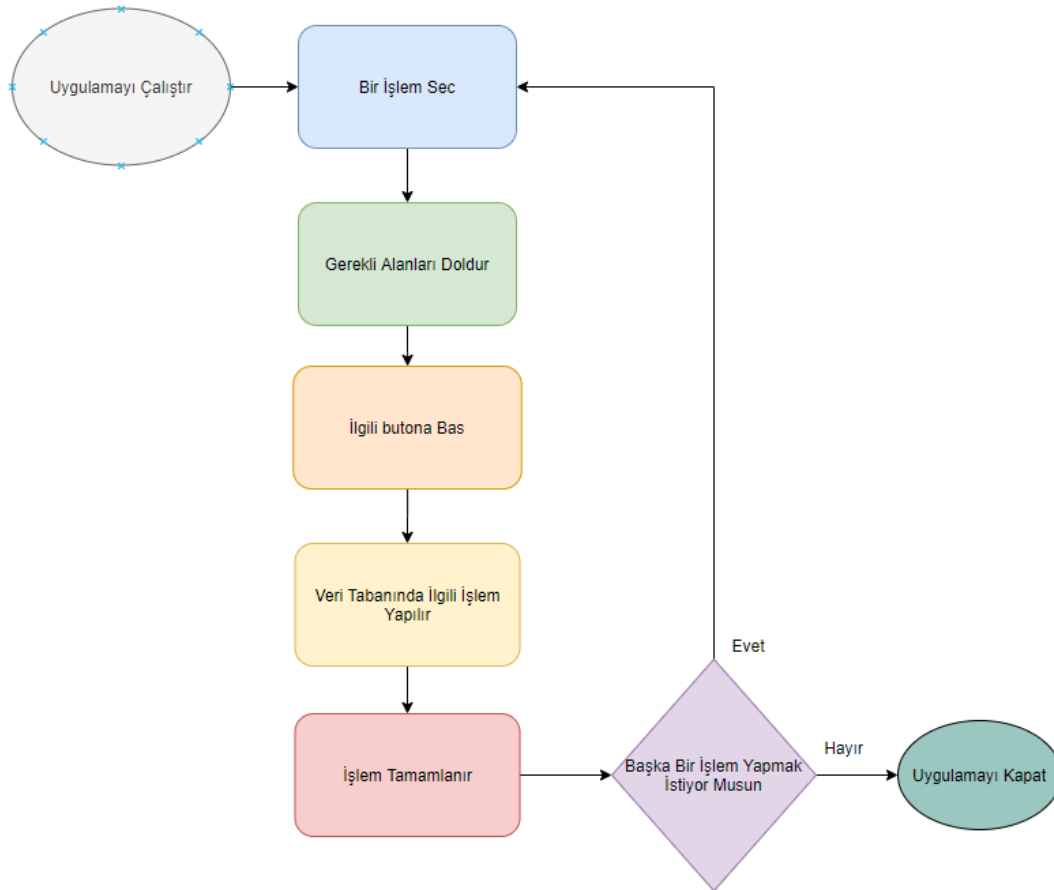
Evim Projesi Veri Sözlüğü				
Dosya Adı	Degisken Adı	Veri Tipi	Uzunluk	Aciklama
Faturan_Tbl	Fatura_Kayitno	İnt	4	Faturanin Sisteme Kayıt Edilme No(Pk)
	Fatura_Turu	Varchar	15	Faturanin Turunue Ait Bilgi
	Fatura_Tarihi	Date	-	Faturanin Tarihine Ait Bilgi
	Fatura_Odemetarihi	Date	-	Faturanin Odendigi Tarihe Ait Bilgi
	Fatura_Miktari	İnt	6	Faturanın Miktarı Ait Bilgi
	Fatura_Dekontu	Varchar	100	Fatura Dekontunun Kayıtlı Oldugu Dizinin Yolu
Garantibelge_Tbl	Garantibelge_Kayitno	İnt	4	Garanti Belgesinin Sisteme Kayıt Edilme No(Pk)
	Garantibelge_Urununturu	Varchar	15	Urunun Turune Ait Bilgi
	Garantibelge_Urununmarkasi	Varchar	15	Urunun Markasına Ait Bilgi
	Garantibelge_Urununmodeli	Varchar	15	Urunun Modeline Ait Bilgi
	Garantibelge_Urununadi	Varchar	20	Urunun Adına Ait Bilgi
	Garantibelge_Urununserino	Varchar	15	Urunun Seri Numarasına Ait Bilgi
	Garantibelge_Urunungarantibitissuresi	Date	-	Urunun Garanti Bitis Tarihini Ait Bilgi
	Garantibelge_Urununfaturasi	Varchar	100	Urunun Faturasinin Kayıtlı Oldugu Dizinin Yolu

Şekil 3.3 Veri Sözlüğü

### 3.2.8 İşlevlerin Sıradüzeni

Kullanıcı E-vimi çalıştırır yapmak istediği işlemin kategorisine ait butona tıklar ve açılan pencereden yapmak istediği işlemin türüne ait butona tıklayarak gerekli forma yönlendirilir burada kullanıcı eğer bir ekleme işlemi yapacaksa ilgili alanları doldurarak ekleme butonuna basarak veri tabanına ilgili verisini kayıt eder.

Eğer işlem bir listeleme, silme veya güncelleme işlemi içeriyorsa işlem yapacağı evrağın kayıt numarasını biliyorsa kayıt numarasını girer eğer kayıt numarası bilinmiyorsa yandaki tablodan evrağın kayıt numarasını bularak ilgili kutucuğa kayıt numarasını girer ve listele butonuna basarak ilgili veriye ulaşır ve bu adımdan sonra ilgili veriyi silebilir veya güncelleyebilir.



Şekil 3.4 Aktivite Diyagramı

### 3.2.9 Başarım Gerekləri

Mevcut sistemler incelendi ve mevcut sistemin eksiklerinden yola çıkılarak sistemin başarımı için

- Uygulamanın veri tabanı uzak bir sunucu üzerinden olmamasına karar verildi
- Uygulama için herhangi bir giriş ekranı olmamasına karar verildi
- Uygulamanın Ara yüzünün kolay ve anlaşılabilir olması istendi
- Uygulamadaki açıkların kapatılması öngörüldü
- Uygulamayı tepki Süresinin en hızlı olacak şekilde tasarlanması hedeflendi

### 3.3 Ara Yüz (Modül) Gerekləri

#### 3.3.1 Yazılım Ara Yüzü

Projenin çalışması esnasında herhangi bir açık verilmemesi özen gösterilerek. Ekibe konuyla ilgili direktifler verilmesine kararlaştırıldı

**3.3.2 Kullanıcı Ara Yüzü:** Kullanıcı ara yüz için yalın anlaşılabilir bir ara yüz tasarlanması gereklidir projenin ara yüzünde kullanan yazı stili ile yazının punto aralığı kullanıcıyı rahatsız etmemesi gereklidir. Bu kararlar göze alınarak bir yazılım tasarlanacaktır

The screenshot displays a user interface with several forms and tables. The top section shows a 'Faturalarınız' (Your Invoices) table with columns for invoice number, type, date, due date, amount, and payment status. Below this is a 'Garanti Belgeleriniz' (Your Warranty Documents) table with columns for document number, product type, brand, model, name, serial number, warranty end date, and invoice number. The middle section contains a 'Fatura Düzenleme Formu' (Invoice Editing Form) with fields for invoice number, type, date, due date, amount, and payment status, along with a 'Mevcut Faturalar (Azalan Sıralama)' (Current Invoices (Decreasing Sort)) table. The bottom section shows a 'Garanti Belgesi Düzenleme Formu' (Warranty Document Editing Form) with fields for document number, product type, brand, model, name, serial number, warranty end date, and invoice number, along with a 'Mevcut Olan Belgeleriniz (Garanti Süresi Azalan Ürünler Basta)' (Current Documents (Warranty Period Decreasing Products First)) table. The right side of the interface features a 'Garanti Belgesi Ekleme Formu' (Warranty Document Adding Form) with fields for document number, product type, brand, model, name, serial number, warranty end date, and invoice number, and a 'Fatura Ekleme Formu' (Invoice Adding Form) with fields for invoice number, type, date, due date, amount, and payment status. The bottom right corner shows a 'Fatura İşlemleri Formu' (Invoice Operations Form) with buttons for 'Yeni Fatura Ekle' (Add New Invoice), 'Fatura Düzenle' (Edit Invoice), 'Garanti Belgesi Ekleme' (Add Warranty Document), and 'Garanti Belgesi Düzenleme' (Edit Warranty Document).

Şekil 3.5 Kullanıcı Ara yüzleri İlk Örnek

### **3.3.2 İletişim Ara Yüzü**

İletişim ara yüzü olarak yapımcı firmanın bilgilerinin bulunduğu bir ara yüz tasarlanacaktır

### **3.3.4 Yönetim Ara Yüzü**

Programda herhangi bir aktör farklılığı bulunmadığı için her bireysel ev kullanıcısı kendi sisteminin hem yöneticisi hem de kullanıcısıdır Bu yüzden yönetim ara yüzü düşünülmemiştir

## **3.4 Belgeleme Gereklere**

### **3.4.1 Geliştirme Sürecinin Belgelenmesi**

Geliştirme sürecinde genel olarak belgelendirilmesi hem ileriye dönük hem de şimdiki geliştirme sürecinde projenin tamamlanma yüzdesini nerede kalınıp nerelerde eksikler olduğunu genel hatlarıyla göstermesi amacıyla yapıldı. Bunun yanı sıra projeye yeni dahil olan personellerin olaya hakimiyeti açısından bu yönteme başvuruldu

### **3.4.2 Eğitim Belgeleri**

Bu belge Uygulamayla birlikte Hem görsel anlatım hem de bir kullanım kılavuzu ile birlikte verilecektir

### **3.4.3 Kullanıcı El Kitapları**

Bu belge programın kullanıcıya teslim edilmeden önce düzenlenerek uygulamanın yanında ek dosya olarak verilecektir

## 4. SİSTEM TASARIMI

### 4.1 Genel Tasarım Bilgileri

#### 4.1.1 Genel Sistem Tanımı



Şekil 4.1 Genel Sistem Tasarımı

#### A) Gereksinimler

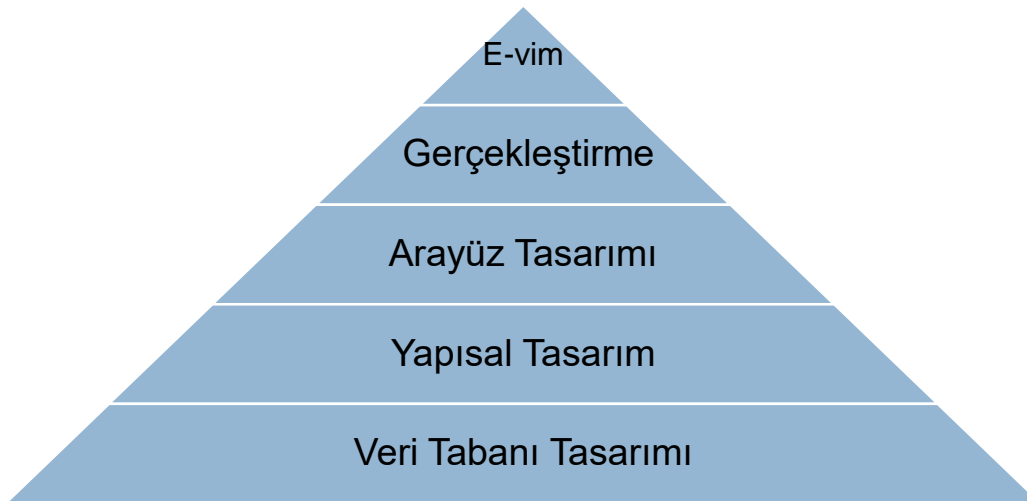
Gereksinimler kısmında çeşitli sosyal medya hesaplarından farklı kullanıcılarla iletişime geçilip kullanıcıların görüş ve önerilerine bakılarak bir çıkarım yapılacaktır

#### B) İşlevsel Belirtilimler

E-vim kullanıcıların faturalarını ve garanti belgelerini elektronik ortama aktararak bir arada tutmayı ve bunların hızlı ve kolay bir şekilde erişilmesini hedeflenmektedir denebilir.

#### C) Tasarım

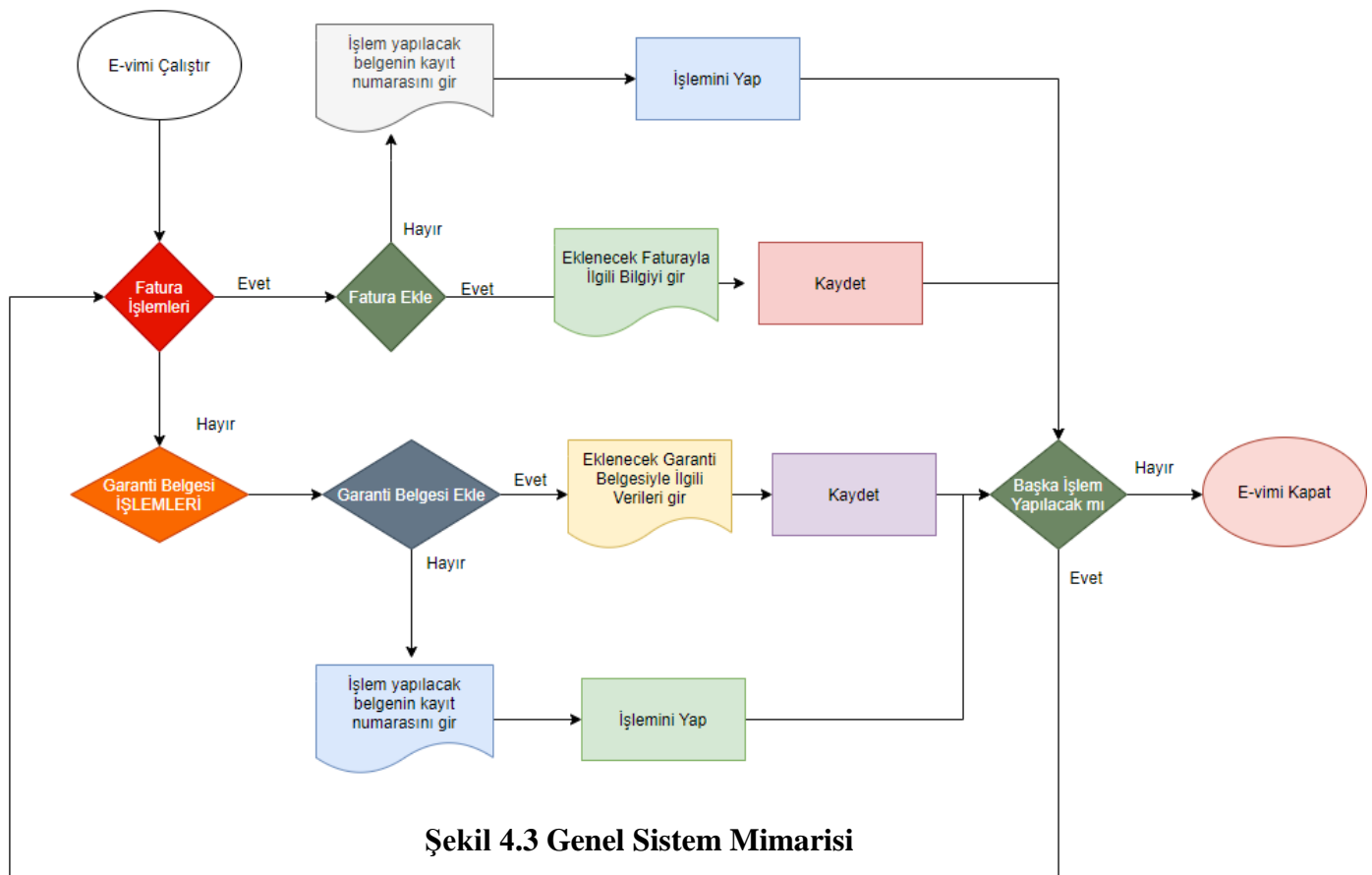
İlk olarak ihtiyaç duyduğumuz veriler belirlenerek bir veri tabanı tasarımı ortaya konulur veri tabanı ortaya konulacak. Veri tabanı tasarımından yola çıkarak yapısal tasarım için gerekli işlemler gerçekleştirilecek. Yapısal tasarım aşaması bittikten sonra oluşan yapısal modele göre bir ara yüz tasarımı sağlanacaktır. Ara yüz tasarımından sonra gerçekleştirme aşamasına geçilerek gerekli kodlamalar yapılacak ve ürünün yazılım ekibi tarafından son hali ortaya koyulacaktır. Ürün ortaya çıktıktan sonra test ekibi tarafından test edilip eksik kısımlar veya hatalı kısımlar varsa yazılım ekibine iletilmesi sağlanacak ve ürün yazılım ekibine iade edilerek yazılımdaki eksiklik ve hataların önüne geçilecektir. Eksikleri ve hataları giderilen yazılım tekrar test ekibine yönlendirilerek son kontrollerinin sağlanması istenilecektir. Eğer son kontrollerde bir sıkıntı çıkmadıysa ürün son kullanıcıya teslim edilecektir



## 4.2 Sistem Kısıtlamaları

Sistemde evrak kayıt numaralarının en son kaydın numarasına göre otomatik olarak belirlemektedir. Sistemde ki alanlara sadece istenilen türdeki verilerin girilmesine yönelik kısıtlamalar getirilmiştir

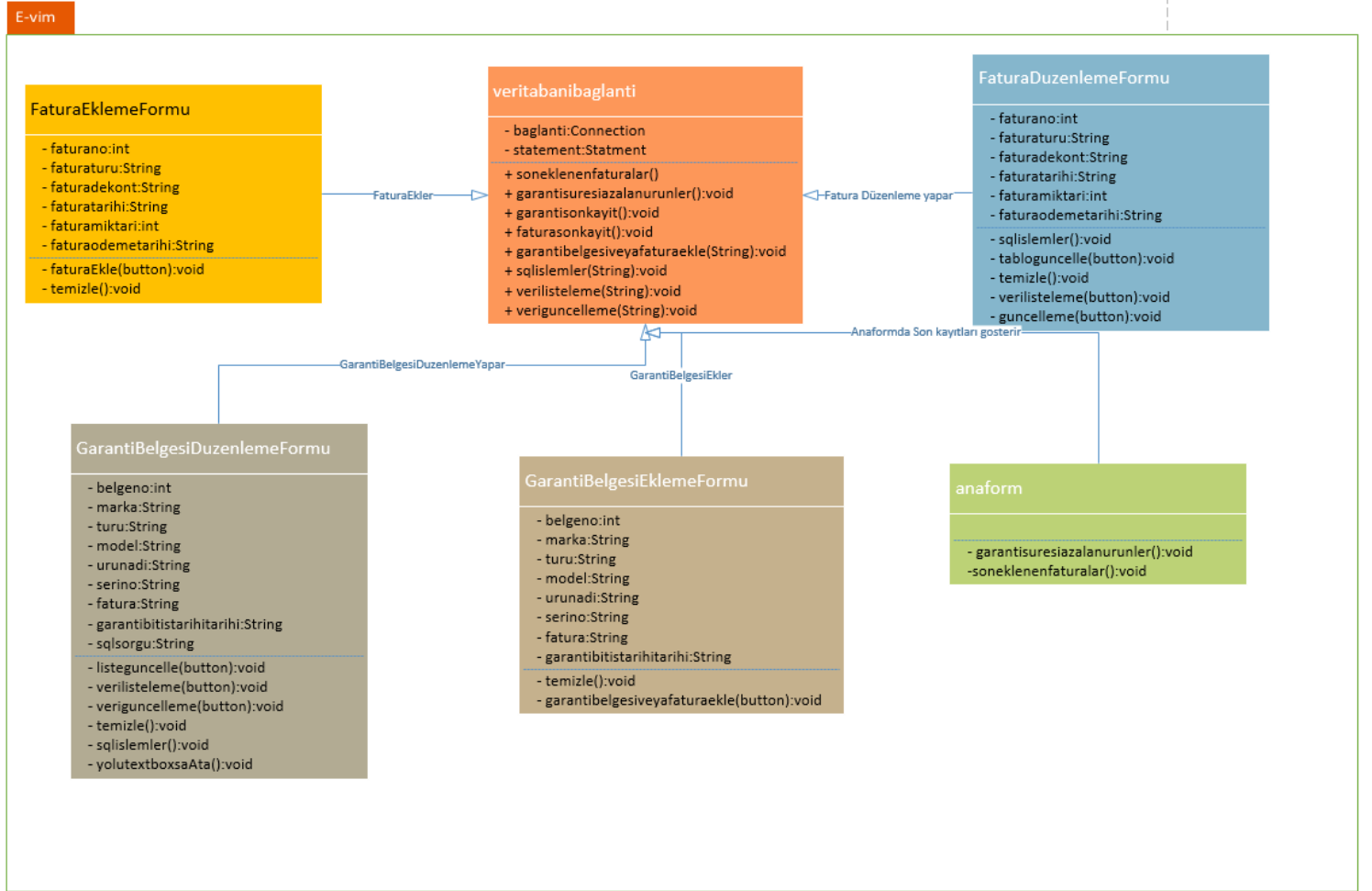
### 4.3 Sistem Mimarisi





**Not:** Sistemin mimarisinin akış diyagramı şeklinde verilmesinin temel sebebi sistemin sistemin mimarisin akış diyagramıyla kolay bir şekilde anlaşılabilir olmasındır.

#### 4.3.1 Sistemin Uml Diyagramını



Şekil 4.3.1 Uml Diyagramı

## **4.4 Dış Arabirimler**

### **4.4.1 Kullanıcı Arabirimleri**

E-vim çalıştırıldığında kullanıcıyı karşılayan form ana form olarak adlandırılmıştır bu sayfadan çeşitli formlara iletişim sağlayarak kullanıcının istediği işlemleri yapılması sağlanacaktır.

### **4.4.2 Veri Arabirimleri**

veri tabanı yönetim sistemi olarak mysql ara birimini kullanıyoruz

### **4.4.3 Diğer Sistemlerle Arabirimler**

E-vimin diğer sistemlerle bir arabirimi bulunmamaktadır

### **4.4.5 Veri Modeli**

E-vimde birbiriyle ilişkili veriler bulunmadığı için herhangi bir veri modeli oluşturulmamıştır

### **4.4.6 Testler**

Testler yazılım ekibiyle ve bir grup kullanıcı tarafından yapılacaktır kullanıcılar tarafından geri dönüşler çağrı merkezi tarafından yazılım ekibine iletilecek ve incelenecektir

### **4.4.7 Performans**

Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecek

#### **A))Sistemin Tasarıma Uygunluk Performansı;**

Tasarımı yapılan sistemin stabilizesi ve işleyiş performansı değerlendirilecek.

#### **B)Veri Yapısının Sistemle Performansı;**

Veri yapısının sistemle stabilizesi ve çalışma zamanındaki uyumluluk düzeyindeki performansı değerlendirilecek.

## 4.2 Veri Tasarımı

### 4.2.1 Tablo Tanımları

Sistem 2 tablodan oluşmaktadır. Bunun yanı sıra kullanıcının ekstra istekleri doğrultusunda ek tablolara açık bir sistem tasarlanmaktadır. Sistemde ilişkili tablolar bulunmamakta ve herhangi bir kullanıcı tablosu bulunmamaktadır.

Dosya Adı	Degisken Adı	Veri Tipi	Uzunluk	Aciklama
Faturan_Tbl	Fatura_Kayitno	İnt	4	Faturanin Sisteme Kayıt Edilme No(Pk) Boş Geçilemez
	Fatura_Turu	Varchar	15	Faturanin Turunue Ait Bilgi
	Fatura_Tarihi	Date	-	Faturanin Tarihine Ait Bilgi
	Fatura_Odemetarihi	Date	-	Faturanin Odendigi Tarihe Ait Bilgi
	Fatura_Miktari	İnt	6	Faturanın Miktarı Ait Bilgi
	Fatura_Dekontu	Varchar	100	Fatura Dekontunun Kayıtlı Oldugu Dizinın Yolu

Şekil 4.5 Fatura Tablosunu Göstermektedir

Garantibelge_Tbl	Garantibelge_Kayitno	İnt	4	Garanti Belgesinin Sisteme Kayıt Edilme No(Pk) Boş Geçilemez
	Garantibelge_Urununturu	Varchar	15	Urunun Turune Ait Bilgi
	Garantibelge_Urununmarkasi	Varchar	15	Urunun Markasına Ait Bilgi
	Garantibelge_Urununmodeli	Varchar	15	Urunun Modeline Ait Bilgi
	Garantibelge_Urununadi	Varchar	20	Urunun Adına Ait Bilgi
	Garantibelge_Urununserino	Varchar	15	Urunun Seri Numarasına Ait Bilgi
	Garantibelge_Urunungarantibitissuresi	Date	-	Urunun Garanti Bitis Tarihini Ait Bilgi
	Garantibelge_Urununfaturasi	Varchar	100	Urunun Faturasinin Kayıtlı Oldugu Dizinın Yolu

Şekil 4.6 Garanti Tablosunu Göstermektedir

## 4.2.2 TABLO- İLİŞKİ ŞEMALARI

2 Tabloda birbirinden bağımsız tablodur ve tabloları ilişkilendirilmeleri yoktur

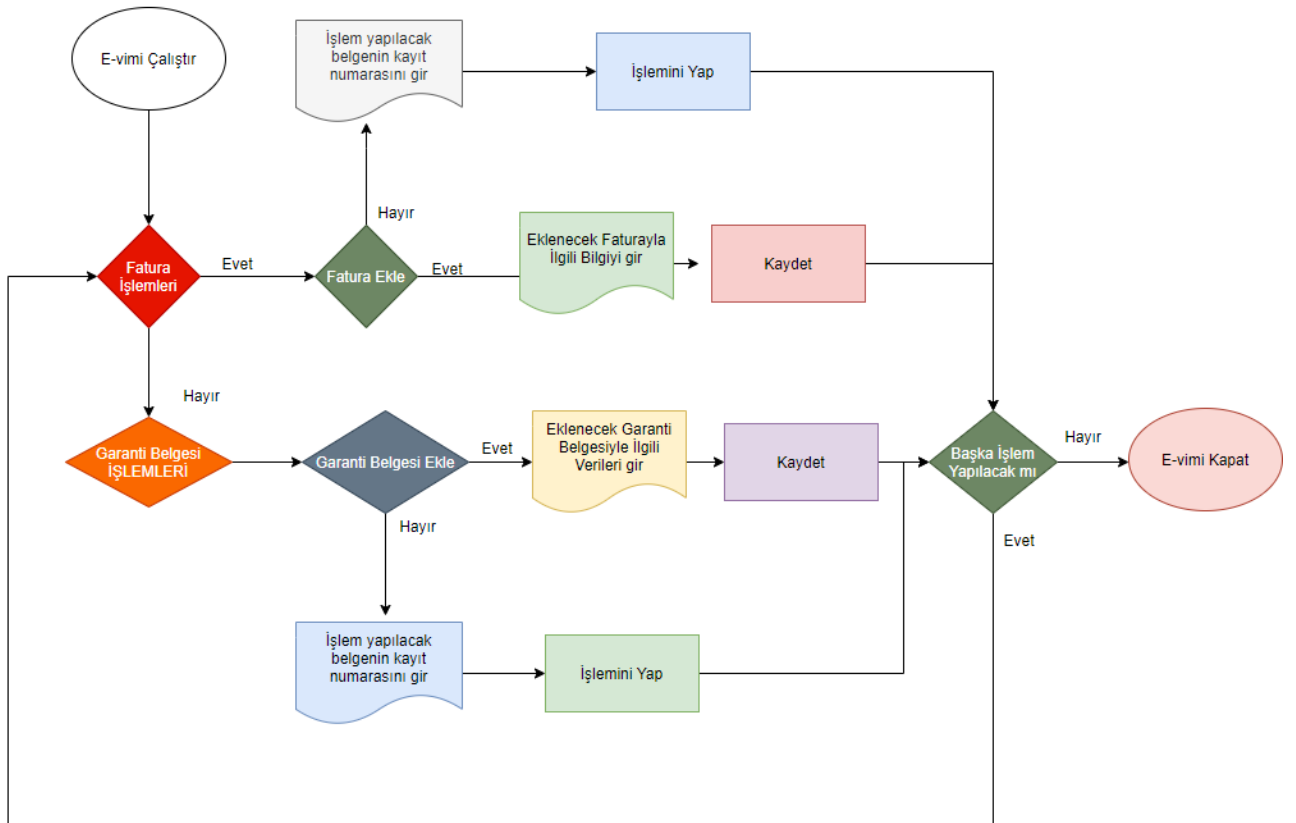
## 4.2.3 Veri Tanımları

Veri tipi olarak integer(int) kullanılmasının amacı sayısal ilgili verinin sayısal bir veri türü olmasıdır. Varchar veri tiplerinde string ifadedi veriler için kullanılmaktadır. Date veri tipi ise tarih olarak bir veri girileceği zaman en uygun veri tipinin bu olmasında dolaydır

## 4.3 Süreç Tasarımı

### 4.3.1 Genel Tasarım

Genel olarak tasarımı ilk önce veri tabanı modeli oluşturuldu. Ardından sistemin tasarımı yapıldı daha sonra ise sistemin ara yüzü oluşturuldu.



Şekil 4.7 Sistemin Genel İşleyiş Tasarımı

## **4.3.2 Modüller**

### **4.3.2.1 Kullanıcı Arabirimi**

Arabirim olarak Mysql vtys programı kullanılmaktadır

### **4.3.3 Kullanıcı Profilleri**

Kullanıcı: E-vim üzerinde her türlü işlemi yapan kullanıcı profilidir ve yazılımda tek kullanıcı profilidir

### **4.3.4 Entegrasyon Ve Test Gereksinimleri**

Sistem herhangi bir entegrasyon işlemine gerek duymamaktadır yükle kullan tarzında bir yazılımdır.

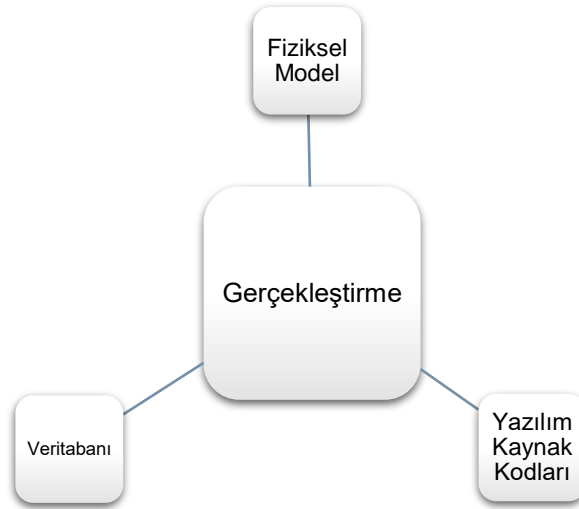
## **4.4 Ortak Alt Sistemlerin Tasarım**

Sistemin herhangi bir alt sistemi yoktur.

## 5.Sistem Gerçekleştirimi

### 5.1 Giriş

Gerçekleştirim çalışması, tasarım sonucu üretilen süreç ve veri tabanının fiziksel yapısını 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 ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan:

- Programlama dili (Java)
- Veri tabanı yönetim sistemi(Mysql)
- Sistem Geliştirme ortamı(Netbeans)
- Case araçları(MS Visio )

Olarak belirlendi ve yazılım geliştirme ortamı hazırlandı

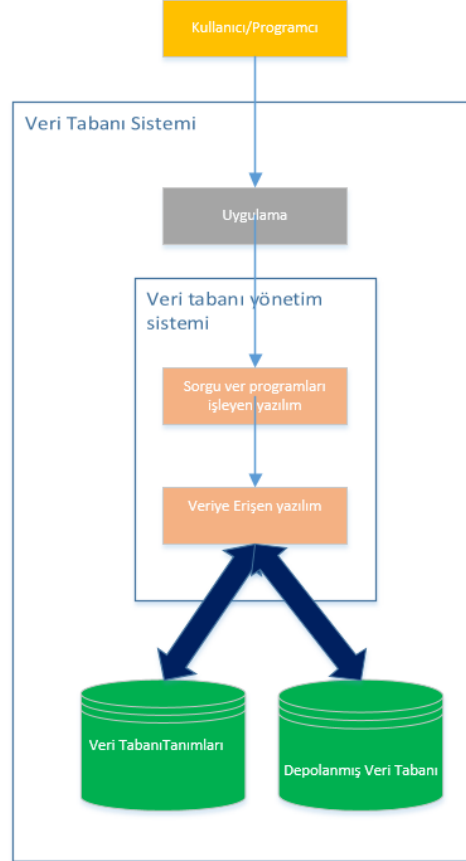
#### 5.2.1 Programlama Dilleri

Sistem Programlama Dili olarak güvenilirlik açısından Java seçilmiştir. Sistem sorgulama dili olarak SQL kullanılmaktadır. IDE ortamı olarak daha gelişmiş bir ara yüze sahip olduğu için netbeans kullanılmıştır

### 5.2.2 Veri Tabanı Yönetim Sistemleri

Mysql sorgu ve veri tabanı arasında bağlantıyı sağlıyor

Veri tabanı tanımları ve Depolanmış veri tabanı ise SQL ile çözümleniyor.



Şekil 5.2 Veri Tabanı Sistemi

### 5.2.3 Veri Tabanı Dilleri Ve Arabirimleri

Sistemimizde veri tabanı dili olarak SQL kullanıldı. Henüz prototip aşamasında olan sistemimiz için MySQL arabirimini yeterli olacağı düşünüldü

### 5.2.4 Veri Tabanı Sistem Ortam

Veri tabanı sistem ortamı olarak MySQL yazılımının MySQL Notifer (1.1.8)i kullandık

### **5.2.4 CASE Araç Ve Ortamları**

CASE araç kullanılmasına özen gösterilmelidir case ortam aracı olarak MS Visio programı kullanılmalıdır gerekli durumlarda başka programlardan yardım alınabilir

### **5.2.5 Açıklama Satırları**

Açıklanma satırlarının kullanılmasına özen gösterilmelidir alışagelmışin dışında olan kod parçacıklarının yanlarına açıklama yapılmalıdır. Her bir kod parçacığının nasıl çalıştığı ne işe yaradığı açıklanmalıdır.

### **5.2.6 Kod Biçimlemesi**

Kod biçimlemesi olarak alışagelmışin dışında ve karmaşık görünen kısımlara açıklama satırları yazılmalıdır. Parantezlerin kullanım şekilleri netbeans tanımladığı şekilde yan yana yazılmalıdır ve karmaşıklıktan olabildiğince kaçınılmalıdır. Metotlar oluşturulurken her bir metot oluşturulmadan önce bir açıklama satırıyla metodun ne iş yaptığı açıklanmalıdır

### **5.2.7 Anlamlı İsimlendirme**

Genel olarak kullanılan deve sırtı modelini kullandık ve OOP'nin gerektirdiği gibi sınıfların isimleri büyük harfle başlandı. Tablolar oluştururken tabloadı\_tbl şeklinde tablodaki alanların isimleri de tabloadı\_alanadi şeklinde oluşturuldu

### **5.2.8 Yapısal Programlama Yapıları**

Yapısal programlama yapısı olarak ekibin sıralı, döngülü ve koşullu yapılar kullanılmalı goto deyiminden kaçınılması gerekmektedir.

### **5.2.9 Hazır Program Kütüphane Dosyaları**

Hazır kütüphane dosyası olarak mysql-connector-java-8.0.19 kullanıldı ayrıca date componenti eklendi Java SQL Kütüphanesi kullanıldı



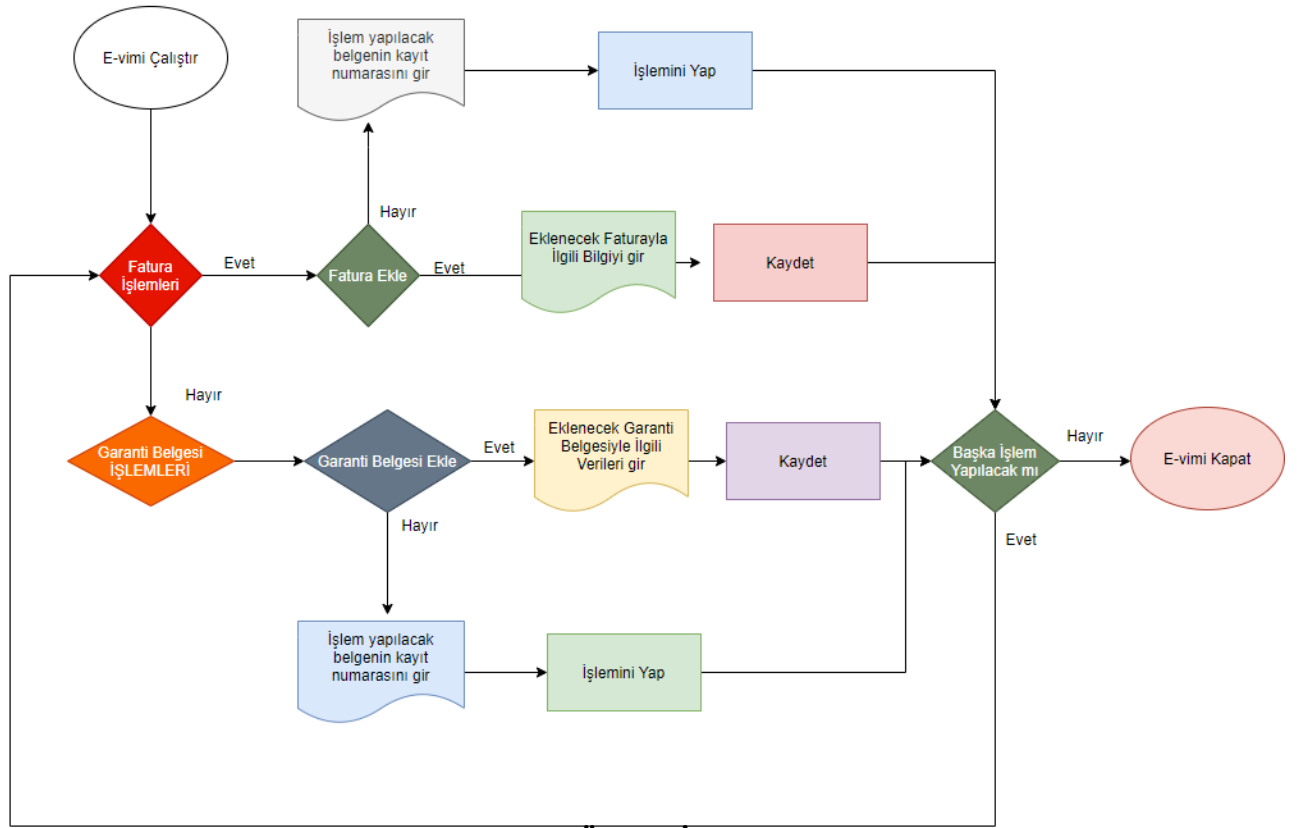
## 5.3 McCabe Ölçütü Hesaplanması

### 5.3.1 Mccabe Formülü: $V(G)=K-D+2p$

**K:** kenar sayısı

**D:** düğüm sayısı

**P:** bileşen sayısı



Şekil 5.3 Mccabe Karmaşıklık Ölçütü İçin Kullanılan Diyagram

**Kenar sayısı:17**

**Düğüm sayısı 10**

**Bileşen sayısı=1**

**$P=17-10+2.1=9$**

**Karmaşıklığımız 9 çıkmıştır standartlar 1-10 arasında ise fazla riskli olmayan basit bir modül olarak tanımlanmaktadır sonuç olarak gerçekleştirmemiz geçerlidir**

## **5.4 Olağın Dışı Durum Çözümleme**

Olağın dışı bir durumda program için gerekli hata ayıklama blokları tanımlanmış ve bu hatanın içeriğinin kullanıcıya gösterilmesi hedeflenmiştir

### **5.4.1 Olağın Dışı Durum Tanımları**

Olağın dışı gelişen durumlarda try-catch blokları devreye girecek ve program kırılmadan çalışmasına devam edebilecek şekilde tasarlanmıştır.

### **5.4.2 Farklı Olağın Dışı Durum Çözümleme Yaklaşımları**

Tüm olağın dışı durumlarda program kırılmadan hata mesajlarıyla tekrar başa dönecek şekilde tasarladık. Olası olağın dışı durum saptandığı kullanıcıya durumun konusuyla ilgili bir mesaj göstererek kullanıcının bizlerle iletişime geçip hata kodunun paylaşılması istenecektir.

## **5.5 Kod Gözden Geçirme**

Yazılım ekibi ve yazılım test mühendisliği ekibindeki arkadaşlarımızdan oluşan bir ekip kurularak kodlar gözden geçirilmiştir.

### **5.5.1 Gözden Geçirme Sürecinin Düzenlenmesi**

Bu kısımda hataların bulunması hedeflenmiş fakat hataların düzeltilmesi hedeflenmemiştir hatalar not alınarak geliştirici ekibe iletilmesi amaçlanmıştır

## 5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular

### 5.6.2.1 Öbek Ara Yüzü

Oluşturduğumuz öbekleri test etmek için belli sorular sorduk bu sorular:

- Her öbek tek bir işlevsel amacı yerine getiriyor mu?
- Öbek adı, işlevini açıklayacak biçimde anlamlı olarak verilmiş mi?
- Öbek tek giriş ve tek çıkışlı mı?
- 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, parametreleri, küresel değişkenleri içeren girdileri ve kütükleri tanıtlıyor 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 uyumlumu?
- Modüller arası veri aktarımları sağlanıyor mu?
- Bütün modüller birleştiğinde sistem çalışıyor mu? Gözden geçirme sırasında referans alınacak sorular olacaktır.

#### **5.6.2.5 Sunuş**

- Her satır, en fazla bir deyim içeriyor mu?
- Bir deyim birden fazla satıra taşması durumunda, bölünme anlaşılabilirliği kolaylaştıracak biçimde anlamlı mı?
- Koşullu deyimlerde kullanılan mantıksal işlemler yalın mı?
- Bütün deyimlerde, karmaşıklığı azaltacak şekilde parantezler kullanılmış mı?
- Bütün deyimler, belirlenen program stiline uygun olarak yazılmış mı?
- Öbek yapısı içerisinde akıllı "programlama hileleri" kullanılmış mı?

## 6)DOĞRULAMA VE GEÇERLEME

### 6.1 Sınama Planlaması

**Test planı kimliği:** E-vim Test

**Sınama Standarttı:** IEEE 1028-2008 Standartlarına göre sınanmıştır

**Sınama Etkinlik Tarihi:** 12.05.2020-15.05.2020

**Sınama Konusu:** Genel Sınama

#### 6.1.2Test Edilecek Sistem:

Test edilecek sistem bireysel ev kullanıcıları için oluşturulmuş bir sistemdir bu sistemde 2 adet veri tablosu bulunmaktadır. Veri tabloları birbiriyle ilişkili değildir. Sistemde 1 ana form olmak üzere 2 yönlendirme 4 adette işlem formu bulunmaktadır. Sistemde bulunan tablolar ve her bir veri girişi alanları farklı değerler ile sınanmalıdır.

**6.1.3 Test Edilecek Ana Fonksiyonlar:** Sistemin test edilecek fonksiyonları fatura ekleme, güncelleme, silme, listeleme; Garanti belgesi ekleme, silme, güncelleme ve listeleme fonksiyonları olacaktır

**6.1. 3 Test Edilmeyecek Ana Fonksiyonlar:** Sistemde test edilmeyecek fonksiyonlar temizle fonksiyonu sistemde test edilmeyecektir herhangi bir olası hatanın olmadığı gözlenmiş ve önemli bir fonksiyon olmadığı saptanmıştır.

**6.2 Geçti/Kaldı Kriterleri:** Eğer verilerin ekleme silme güncelleme veya listeleme kısımlarında bir hata rastlanırsa sistem kaldı olarak değerlendirilecektir Bir sorun görülmemesi durumunda sistem testten geçmiş sayılacaktır

**6.3 Test Dokümanı:** Test süresince yapılan işlemler raporlanarak ana dokümana eklenecektir

**6.4 Sorumluluklar:** Yazılım test ekipleri fonksiyonların denetlenmesinde sorumludur veri tabanı ekibi eklenen verilerin veri tabanındaki alanlarla uygunluğu değerlendirme ve boyutlarının belirlenmesinde sorumludur.

**6.5 Riskler ve Önlemler:** Sistem bir veri tabanı ile birlikte çalışacağından veri tabanındaki alana gönderilen verinin veri tabanındaki veri türüyle uyuşmaması durumunda sistemde hatalar meydana geldiğinden çok iyi bir şekilde denetlenmesi konusunda yazılım test ve veri tabanı ekibimiz uyarılmalıdır.

## 6.6 Test İşlemleri

### 6.6.1 Testin Türü

#### Beyaz Kutu Sınaması

Sistem için en uygun sınama beyaz kutusu sınaması olarak görüldü ve geçerli sınamalar aşağıdaki şartlar dâhilinde yazılım yaşam döngüsünün her fazında sınıandı

- Bütün bağımsız yolların en azından bir kere sınıandı
- Bütün mantıksal karar noktalarında iki değişik karar için sınamaları yapıldı
- Bütün döngülerin sınır değerlerinde için denemeler yapıldı
- İç veri yapılarının denenmesi yapıldı.
- Sistem için otomatik sınama verisi üreten programlardan yararlanarak gerekli sınamalar yapıldı

## 6.7 Sınama İşlemi Hakkında Bilgileri

**Sistem Sınama:** Sistemin bütün olarak sınanması yapıldı ve programın eksiksiz olduğu onaylandı.

**Kabul Sınama:** Sistem prototipten çıkartılıp gerçek veriler girildi ve sorunsuz olduğu bir kez daha yazılım test ekibi tarafında onaylandı

**Sınama tarihi** 12.05.2020-15.05.2020

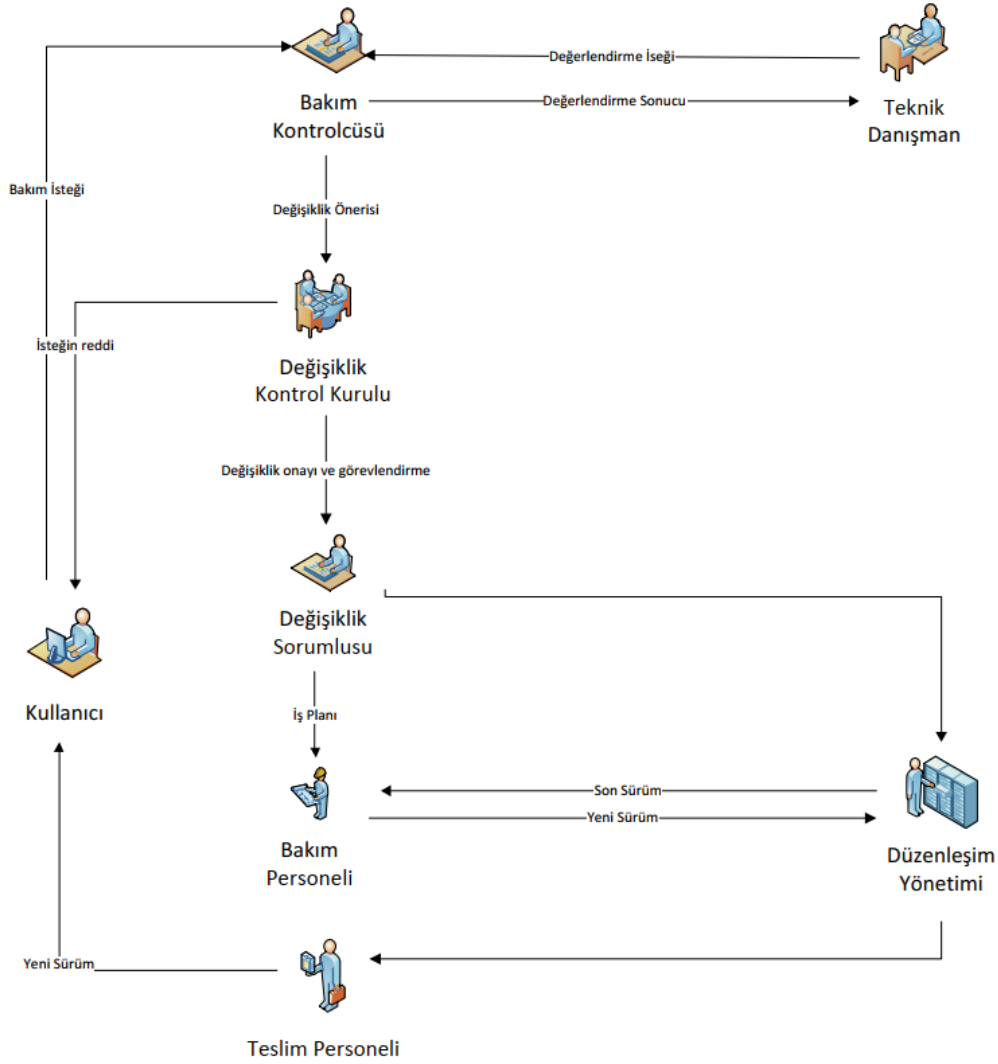
**Sınamayı Yapan:** Hüseyin Yasin DÖNMEZ, Veri Tabanı Ekibi ve Test Ekibi

**Sonuç:** Sistemde yazılım test mühendisleri tarafından yapılan sınamalar sonucunda herhangi bir hataya rastlanmadı.

## 7.Bakım

### 7.1 Bakım Standarttı

**Bakım Standarttı:** IEEE 1219-1998 standardı uygulanmıştır ve uygulanacaktır.



**Şekil 7.1 Bakım aşaması**



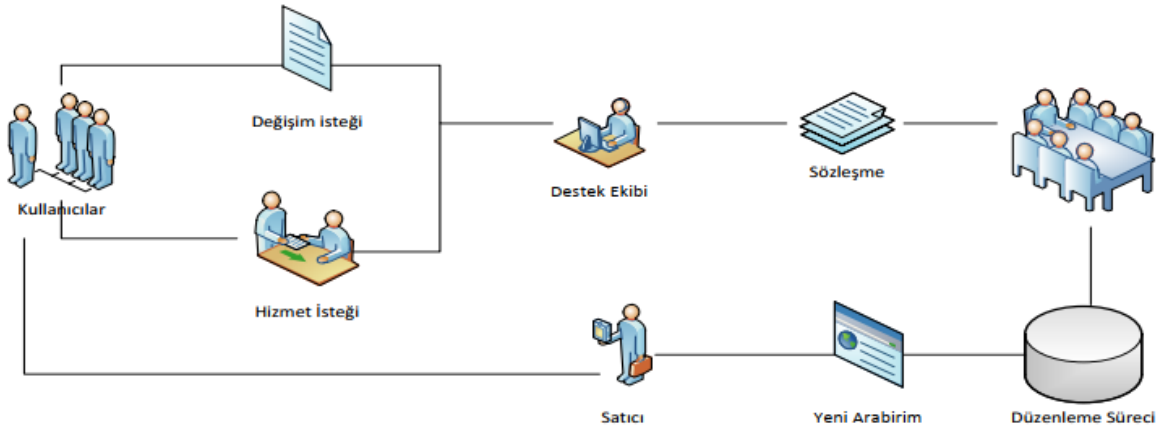
## 7.2 Kurulum

Sistem kurulumu internet üzerinden kurulum dosyasının indirilerek kurulmasıyla gerçekleştirilecektir. Dileyen müşterilere yardımcı programlar tarafından bağlanarak uzaktan kurulum yapılacaktır.

## 7.3 Yerinde Destek Organizasyonu

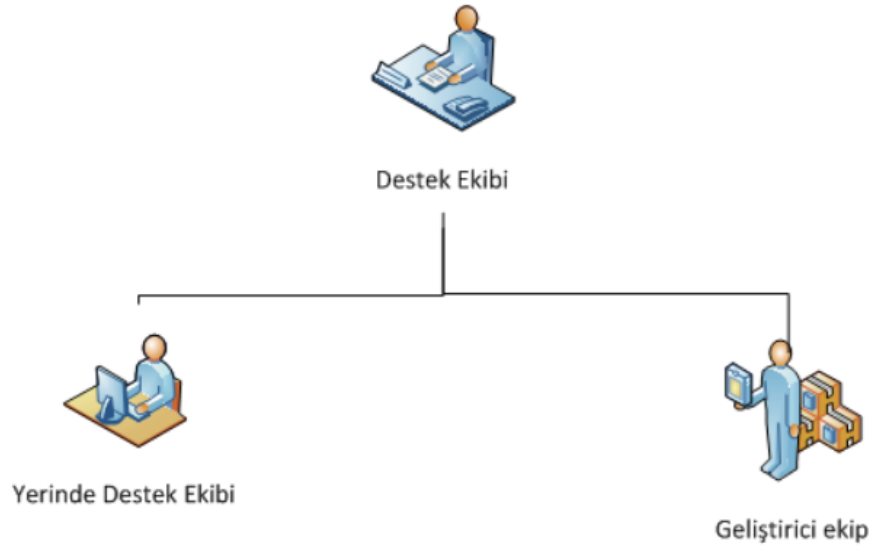
Yerinde destek ekibi kullanıcılardan gelen destek ihtiyaçlarına karşılık vermek amacıyla hafta içi 8.00 ile 17.00 arasında aktif olacaktır. Hafta sonları ise 12.00 ile 17.00 arasında sadece elektronik posta yoluyla destek verilecektir. Aynı zamanda yerinde destek ekibi kullanıcılar tarafından gelen istekleri ve hataları destek ekibine iletmekle yükümlüdür.

## 7.4 Yazılım Bakımı



Şekil 7.2 Yazılım Bakım Süreci Modellemesi

### 7.4.1 Bakım Süreç Modeli



Şekil 7.3 Bakım Süreç Modeli

#### 1. Adım: Sorunu Tanımlama Süreci

Gelen bakım isteği sonrası istek incelenmek üzere yazılım bakım ekibine iletilecek. Ve sorunun tanımlanması amaçlanacaktır.



Şekil 7.4 Yazılım Sorun Tanımlama Sürecinin Gösterimi

## 2. Adım: Çözümleme Süreci

Sorun tanımlamadan çıkan karar sonucunda çözüm süreci başlatılacaktır.



Şekil 7.5 Çözümleme Sürecinin Gösterimi

## 3. Adım: Tasarım Süreci

Çözümlenen sistem sonucunda tasarımı adımına geçilerek mevcut çözüme göre tasarım yapılacaktır.



Şekil 7.6 Tasarım Sürecinin Gösterimi

#### 4. Adım: Gerçekleştirim Süreci

Mevcut tasarım çıktılarına göre sistem kodlanmaya başlanacaktır.



Şekil 7.7 Gerçekleştirim Sürecinin Gösterimi

#### 5. Adım Sistem Sınama Süreci

Oluşan yeni ürün test ekipleri tarafından test edilerek diğer aşamaya hazırlanacaktır.



Şekil 7.8 Sistem Sınama Sürecinin Gösterimi

## 6. Adım: Kabul Sınaması Süreci

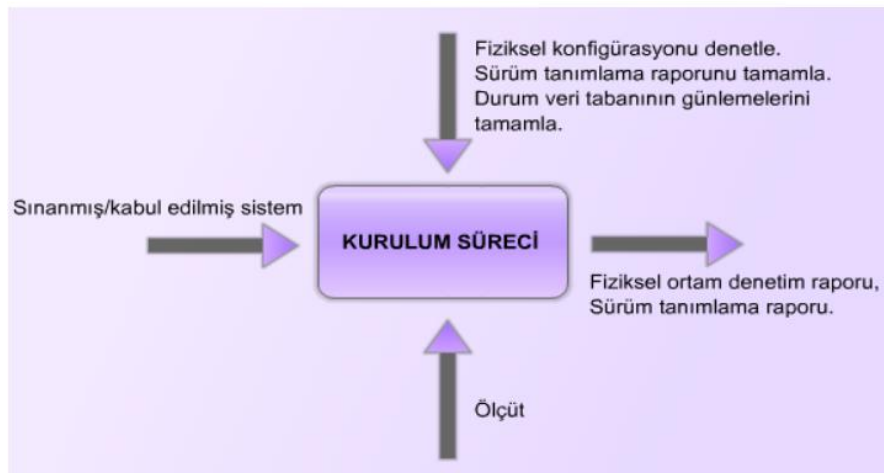
Sınanan Sistem kullanıcılar tarafından ve tekrar yazılım ekibi tarafından sınanacaktır. Sınanan sistem kullanıcılar tarafından kabul edilirse kurulum aşamasına geçilecektir.



Şekil 7.9 Kabul Sınaması Sürecinin Gösterimi

## 7. Adım: Kurulum Süreci

Kabul sınamasını geçen sistemimiz kurulum aşamasına alınarak kullanıcıların cihazlarına kurulacaktır ve bunun sonucundan fiziksel ortamda denetim raporu ve sürüm raporu oluşturulacaktır.



Şekil 7.9 Kurulum Sürecinin Gösterimi

## 8.Sonuç

### 8.1 Sonuç

Sonuç olarak E-vim dokümantasyonu hazırlanmış bir ürün olarak ortaya çıkarılarak kullanıcıların kullanımına sunuldu. Yazılım yaşam döngüsünün sonsuz olan kısımları arasında yerini almayı başardı. Artık mevcut sistemlerin aksine kullanıcılar kendilerini güvende hissettikleri ve daha kullanılabilir bir ara yüzü E-vime kavuşmuş oldular. E-vim kullanıcıların faturalarını ve garanti belgelerini bir arada tutma hedefine ulaştı. Mevcut sistem her uzaktan erişim özelliği olmaması bir dezavantajdır. İleriki aşamalarda sistemimize farklı doğrulama sistemleriyle uzaktan erişim desteğini de tanımlanması planlanmaktadır

## 9.Kaynaklar

### 9.1 Yararlanılan Kitaplar:

K: Dr. SARIDOĞAN, Erhan *Yazılım Mühendisliği Temelleri*: Papatya Bilim, Eylül 2016

Genel anlamda Dokümantasyon oluştururken kullanılmıştır.

K: SOMMERVILLE, Ian *Yazılım mühendisliği*: Nobel akademik yayıncılık, Şubat, 2018

Genel anlamda Dokümantasyon oluştururken kullanılmıştır.

### 9.2 Yararlanılan İnternet Siteleri

1) <http://web.firat.edu.tr/mbaykara/ORNEKPROJEDOKUMANI.pdf>

Genel anlamda Dokümantasyon oluştururken kullanılmıştır.

2) <http://muhammetbaykara.com/> Genel anlamda Dokümantasyon oluştururken kullanılmıştır.

3) <http://www.yazilimcildunyasi.com/2017/01/cocomo-constructive-costing-model.html>

2.bölüm maliyet kestirimi ve hesaplanması cocomo model [bölüm 2.4.3]

4) [https://fatihkabakci.com/article-McCabe\\_CYCLOMATIC\\_COMPLEXITY](https://fatihkabakci.com/article-McCabe_CYCLOMATIC_COMPLEXITY)

5.bölüm projenin karmaşıklığını ölçmek için kullanılmıştır bölüm [5.3]