

**KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ**

## **MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

## **BİLGİSAYAR MÜHENDİSLİĞİ**

**BİL306-Yazılım Mühendisliği Dersi**

**Dr. Öğr. Üyesi \*\*\*\*\*\*\*\*\***

Proje Uygulaması ve Dokümantasyonu

AKILLI ECZANE OTOMASYON SİSTEMİ

**GELİŞTİREN**

**MERVE NİLGÜN YILMAZ**

**MAYIS -\*\*\*\***

**ÖNSÖZ VE TEŞEKKÜR**

Bu proje çalışması süresince her zaman desteklerini hissettiğimiz ve kendilerinden istifade ettiğimiz BİL-306 Yazılım Mühendisliği dersi sorumlu öğretim üyesi **Dr.Öğr.Üyesi Hasan B\*\*** hocamıza teşekkür ederim. Ayrıca hayatım boyunca ve bu çalışma süresince desteklerini esirgemeyen ailem ve arkadaşlarıma teşekkürü bir borç bilirim.

Adı-Soyadı : **Merve Nilgün Yılmaz**

Okul\_no :**19\*\*\*\*\*\***

**İÇİNDEKİLER**

**1. GİRİŞ**

ÖNSÖZ…………………………………………………………………………………………………………………………………01

1.1 Projenin Amacı………………………………………………………………………………………………………………08

1.2 Projenin Kapsamı…………………………………………………………………………………………………………..08

1.3 Projenin İçeriği………………………………………………………………………………………………………………08

1.4 Tanımlamalar ve Kısaltmalar…………………………………………………………………………………………09

**2. PROJE PLANI**

2.1 Giriş……………………………………………………………………………………………………………………….………10

2.2 Projenin Plan Kapsamı……………………………………………………………………………..……………………11

2.3 Proje Zaman-İş Planı………………………………………………………………………………………………………18

2.4 Proje Ekip Yapısı………………………………………………………………………………………………………….…19

2.5 Önerilen Sistemin Teknik Tanımları ………………………………………………………………………………21

2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları………………………………………………………….…21

2.7 Proje Standartları, Yöntem ve Metodolojiler…………………………………………………………………22

2.8 Kalite Sağlama Planı………………………………………………………………………………………………………28

2.9 Konfigürasyon Yönetim Planı…………………………………………………………………………………………30

2.10 Kaynak Yönetim Planı…………………………………………………………………………….……………………30

2.11 Eğitim Planı…………………………………………………………………………………….……………………………32

2.12 Test Planı ……………………………………………………………………………………………………………………32

2.13 Bakım Planı………………………………………………………………………………………………….………………33

**3. SİSTEM ÇÖZÜMLEME**

3.1 Mevcut Sistem İncelemesi

3.1.1 Örgüt Yapısı…………………………………………………………………………………………………..…………..35

3.1.2 İşlevsel Model……………………………………………………………………………………………………………35

3.1.3 Veri Modeli………………………………………………………………………………………………………………..36

3.1.4 Varolan Sistemin Değerlendirilmesi……………………………………………………………………………36

3.1.5 Var Olan Yazılım/Donanım Kaynakları…………………………………………………………………………36

3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş……………………………………………………………………………………………………………………………37

3.2.2 İşlevsel Model…………………………………………………………………………………………….……………..37

3.2.3 Genel Bakış ……………………………………………………………………………………………………………..52

3.2.4 Bilgi Sistemleri/Nesneler……………………………………………………………………………………..…….52

3.2.5 Veri Modeli………………………………………………………………………………………………………………..53

3.2.6 Veri Sözlüğü……………………………………………………………………………………………………………….56

3.2.7 İşlevlerin Sıradüzeni……………………………………………………………………………………………………58

3.2.8 Başarım Gerekleri ……………………………………………………………………………………………………..59

3.3 Arayüz (Modül) Gerekleri

3.3.1 Yazılım Arayüzü………………………………………………………………………………………………………….63

3.3.2 Kullanıcı Arayüzü……………………………………………………………………………………………………….63

3.3.3 İletişim Arayüzü……………………………………………………………………………………….……..…………63

3.3.4 Yönetim Arayüzü……………………………………………………………………………………………….………64

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi…………………………………………………………………………….64

3.4.2 Eğitim Belgeleri………………………………………………………………………………………………………….64

3.4.3 Kullanıcı El Kitapları……………………………………………………………………………………………………64

**4. SİSTEM TASARIMI**

4.1 Genel Tasarım Bilgileri

4.1.1 Genel Sistem Tanımı ………………………………………………………………………………………………..…65

4.1.2 Varsayımlar ve Kısıtlamalar…………………………………………………………………………………….…..67

4.1.3 Sistem Mimarisi…………………………………………………………………………………………………….……68

4.1.4 Dış Arabirimler…………………………………………………………………………………………………………...68

4.1.4.1 Kullanıcı Arabirimleri………………………………………………………………………………………………..68

4.1.4.2 Veri Arabirimleri……………………………………………………………………………………………………...69

4.1.4.3 Diğer Sistemlerle Arabirimler……………………………………………………………………………………69

4.1.5 Veri Modeli…………………………………………………………………………………………………………………69

4.1.6 Testler………………………………………………………………………………………………………………………..70

4.1.7 Performans………………………………………………………………………………………………………………..70

4.2 Veri Tasarımı

4.2.1 Tablo tanımları………………………………………………………………………………………………………70-71

4.2.2 Tablo- İlişki Şemaları………………………………………………………………………………………………...…71

4.2.3 Veri Tanımları……………………………………………………………………………………………………………..72

4.2.4 Değer Kümesi Tanımları………………………………………………………………………………………………72

4.3 Süreç Tasarımı

4.3.1 Genel Tasarım…………………………………………………………………………………………………………….73

4.3.2 Modüller…………………………………………………………………………………………………………………73

4.3.2.1 Yönetici Modülü…………………………………………………………………………………………………..73

4.3.2.1.1 İşlev………………………………………………………………………………………………………………………73

4.3.2.1.2 Kullanıcı Arabirimi…………………………………………………………………………………………………73

4.3.2.1.3 Modül Tanımı………………………………………………………………………………………………………..73

4.3.2.1.4 Modül iç Tasarımı………………………………………………………………………………………………….74

4.3.2.2 Seçmen Modülü

4.3.2.2.1 İşlev…………………………………………………………………………………………………………………..….74

4.3.2.2.2 Kullanıcı Arabirimi…………………………………………………………………………………………………74

4.3.2.2.3 Modül Tanımı……………………………………………………………………………………………………..…74

4.3.2.2.4 Modül iç Tasarımı……………………………………………………………………………………………….…75

4.3.3 Kullanıcı Profilleri………………………………………………………………………………………………………..75

4.3.4 Entegrasyon ve Test Gereksinimleri…………………………………………………………………………….75

4.4 Ortak Alt Sistemlerin Tasarımı

4.4.1 Ortak Alt Sistemler……………………………………………………………………………………………………...75

4.4.2 Modüller Arası Ortak Veriler……………………………………………………………………………………….76

4.4.3 Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri…………………………………………………….76

4.4.4 Güvenlik Alt Sistemler…………………………………………………………………………………………………76

4.4.5 Veri Dağıtım Alt Sistemleri…………………………………………………………………………………………..76

4.4.6 Yedekleme ve Arşivleme İşlemleri……………………………………………………………………………....76

**5. SİSTEM GERÇEKLEŞTİRİMİ**

5.1. Giriş………………………………………………………………………………………………………………………………77

5.2. Yazılım Geliştirme Ortamları……………………………………………………………………………………..…79

5.2.1 Programlama Dilleri………………………….…………………………………………………………………………79

5.2.2 Veri Tabanı Yönetim Sistemleri……………………………………………………………………………………79

5.2.2.1 VTYS Kullanımının Ek Yararları………………………………………………………………………………....80

5.2.2.2 Veri Modelleri……………………………………………………………………………………………………….…80

5.2.2.3 Şemalar…………………………………………………………………………………………………………………...82

5.2.2.4 VTYS Mimarisi………………………………………………………………………………………………………….82

5.2.2.5 Veritabanı Dilleri ve Arabirimleri………………………………………………………………………………83

5.2.2.6 Veri Tabanı Sistem Ortamı………………………………………………………………………………………..83

5.2.2.7 VTYS'nin Sınıflandırılması……………………………………………………………………………………..….84

5.2.2.8 Hazır Program Kütüphane Dosyaları…………………………………………………………………….…..84

5.2.2.9 CASE Araç ve Ortamları…………………………………………………………………………………………….84

5.3. Kodlama Stili…………………………………………………………………………………………………………………84

5.3.1 Açıklama Satırları………………………………………………………………………………………………………..84

5.3.2 Kod Biçimlemesi………………………………………………………………………………………………….……..84

5.3.3 Anlamlı İsimlendirme………………………………………………………………………………………………....84

5.3.4 Yapısal Programlama Yapıları……………………………………………………………………………………...84

5.4. Program Karmaşıklığı…………………………………………………………………………………………………….85

5.4.1 Programın Çizge Biçimine Dönüştürülmesi…………………………………………………………….……85

5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama…………………………………………………………………….…85

5.5. Olağan Dışı Durum Çözümleme……………………………………………………………………………………..86

5.5.1 Olağandışı Durum Tanımları…………………………………………….………………………………………....86

5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları……………………………………………………….86

5.6. Kod Gözden Geçirme…………………………………………………………………………………………………….86

5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi…………………………………………………………………...86

5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular……………………………………………………………86

5.6.2.1 Öbek Arayüzü…………………………………………………………………………………………………………..86

5.6.2.2 Giriş Açıklamaları………………………………………………………………………………………………..……87

5.6.2.3 Veri Kullanımı……………………………………………………………………………………………....………….87

5.6.2.4 Öbeğin Düzenlenişi………………………………………………………………………………………………….87

5.6.2.5 Sunuş……………………………………………………………………………………………………………………….87

**6. DOĞRULAMA VE GEÇERLEME**

6.1. Giriş………………………………………………………………………………………………………………………………89

6.2. Sınama Kavramları……………………………………………………………………………………….……………….89

6.3. Doğrulama ve Geçerleme Yaşam Döngüsü………………………………………………………………….…90

6.4. Sınama Yöntemleri………………………………………………………………………………………………………..90

6.4.1 Beyaz Kutu Sınaması……………………………………………………………………………………………………90

6.4.2 Temel Yollar Sınaması………………………………………………………………………………………………….91

6.5.Sınama ve Bütünleştirme Stratejileri………………………………………………………………………………91

6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme…………………………………………………………………..91

6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme……………………………………………………………….92

6.6. Sınama Planlaması……………………………………………………………………………………………..…………93

6.7. Sınama Belirtimleri ……………………………………………………………………………………………………….94

**7. BAKIM**

7.1 Giriş……………………………………………………………………………………………………………………………….94

7.2 Kurulum…………………………………………………………………………………………………………………...…..94

7.3 Yerinde Destek Organizasyonu…………………………………………………………………………………….…94

7.4 Yazılım Bakımı………………………………………………………………………………………………………………..94

7.4.1 Tanım………………………………………………………………………………………………………………………….95

7.4.2 Bakım Süreç Modeli…………………………………………………………………………………………………….96

**8. SONUÇ** …………………………………………………………………………………………………………………99

**9. KAYNAKLAR** ……………………………………………………………………………………………………100

**1.GİRİŞ**

**1.1.Projenin Amacı:**

Ülkemizdeki eczacılık sektörüne baktığımız zaman yoğunluktan dolayı eczacıların hastalarına gerekli bilgiyi yeterince veremediklerini görmekteyiz. Teknoloji hızla gelişmekte ve eczacılar da teknolojinin getirdiği kolaylıktan kendi nasibine düşeni almaktadır. Akıllı Otomasyon sistemimizin amacı eczacılara kolaylık sağlamak hastalarına ise kullandıkları ilaç hakkında gerekli bilgiyi alabilmelerini sağlamayı hedeflemektedir. Akıllı Eczane otomasyonumuz ile ürün girdi/çıktı, ürün ekleme/çıkarma, ürünün giriş/çıkış tarihi işlemlerini yapabilir bu konuda gerekli kullanıcıların yer almasının sağlanması daha sonra ürünlerle ilgili gerekli işlemlerin yapılmasını sağlayan bir otomasyon sistemi oluşturulabilir.

**1.2.Projenin Kapsamı:**

Eczacıların ve hastaların ilaçların kullanımı hakkında , yan etkilerini, fiyat, ödeme gibi bilgileri otomasyona giriş yaparak bu sistemin üzerinden istedikleri bilgileri daha açık ve kapsamlı bir şekilde verebilmektir. Kullanıcıları eczacılar,hastalar ve isteyen diğer kullanıcılardır. Bazı uygulama alanları ;

* E-sipariş
* Canlı Destek
* İlaç Hakkında Bilgi Alma

**1.3.Projenin İçeriği:**

Eczacılar veya yöneticiler ilaç fiyatlarını , yan etkilerini ve ilaç hakkında bilgi ekleme çıkarma işlemlerini yaparak güncelleme yapabilmelidir. Sitemizin kullanıcıları ise sisteme kullanıcı girişi yaptıktan sonra istedikleri ilaç hakkında bilgiyi site üzerinden kolayca erişebilecek eğer sistemde bilgiler yeterli bulunmadı takdirde canlı destek hizmetine de erişilebilinecektir. Bu projemizden yararlanabilecek olan kısım kısıtlı değil isteyen herkes kolaylıkla yararlanabilecektir.

**1.4 Tanımlamalar ve Kısaltmalar:**

-Yönetici: Sistem Yöneticisi

-VTYS: Veri Tabanı Yönetim Sistemi

-KYS: Kayıt Sistemi

-Kullanıcı: Eczaneci veya diğer kişiler

-İTS: İlaç Takip Sistemi

**2. PROJE PLANI**

**2.1 Giriş**

Sistem kullanıcısı TC kimlik numarası ile kayıt olup otomasyona giriş yaptıktan sonra kullanıcıya hastaneden verilen reçete numarası ile ilaçları görebilir ve bu ilaçlar hakkında bilgi alabilir. Akıllı eczane otomasyonu sayesinde hastalar canlı destek ile detaylı bir şekilde ilaçlar hakkında bilgi alıp sağlıklı bir şekilde ilacı kullanmaya başlayabilirler. Tasarlanacak sistemin şeması aşağıda yer almaktadır.

**Reçete**

**Bilgisi**

**Kullanıcı**

**Girişi**

**İlaç Bilgileri**

**Eczacı veya Yönetici Girişi**

**Kullanım Bilgileri**

**Fatura**

**Yan Etkileri**

ŞEKİL 2.1 Projenin Genel Yapısı

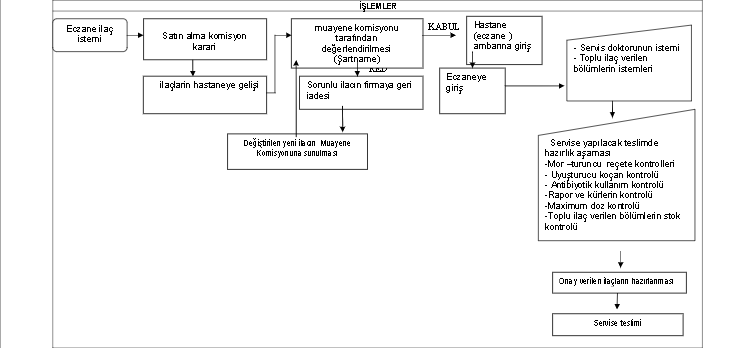
**2.2 Projenin Plan Kapsamı**

Projenin plan kapsamında genel olarak sistemin gerekliliği, mevcut genel olarak eczacıların hasta yoğunluğu nedeni ile hastaları ile detaylı ve kapsamlı bir şekilde ilaç hakkında bilgi verip yan etkilerini söyleyemediklerinden şikayetçi oldukları için böyle bir sistem geliştirdik bu proje sayesinde hastalar daha iyi hizmet alabilecekler

**SİSTEME İSTENEN İLACI SİSTEMDEN**

**GİRİŞ TARAMA BİLGİLENME**

Şekil 2.2 Projenin Genel İşleyişi



Şekil 2.3 Eczane İşlemleri İş Akış Şeması

**Proje Neden Gerekli?**

**-**Müşterilerin eczane işlemlerini istedikleri zamanda rahat bir şekilde yapabilmeleri için

**-**web site üzerinden daha rahat bilgilenmek için

**-**Daha kolay işlem yapabilmek için

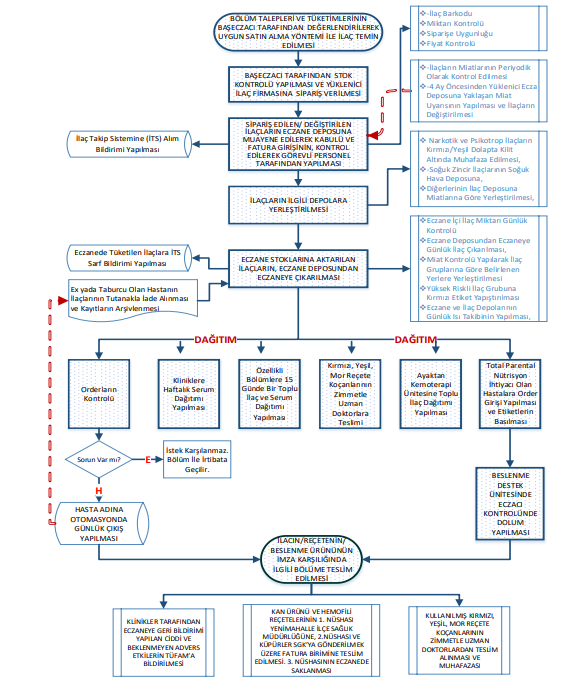
**-**Ekonomik ve zamandan tasarruf sağladığı için

**-**Sistemin kolay kontrol edilebilmesi için

**-**İlaç takibini sağlamak için

**-**Müşteri işlemlerini rahat bir şekilde sağladığı için

**-**Gerektiğinde verilere hızlı erişim sağladığından işlevselliği arttırdığı için

****

Şekil 2.4 Eczane Akış Şeması

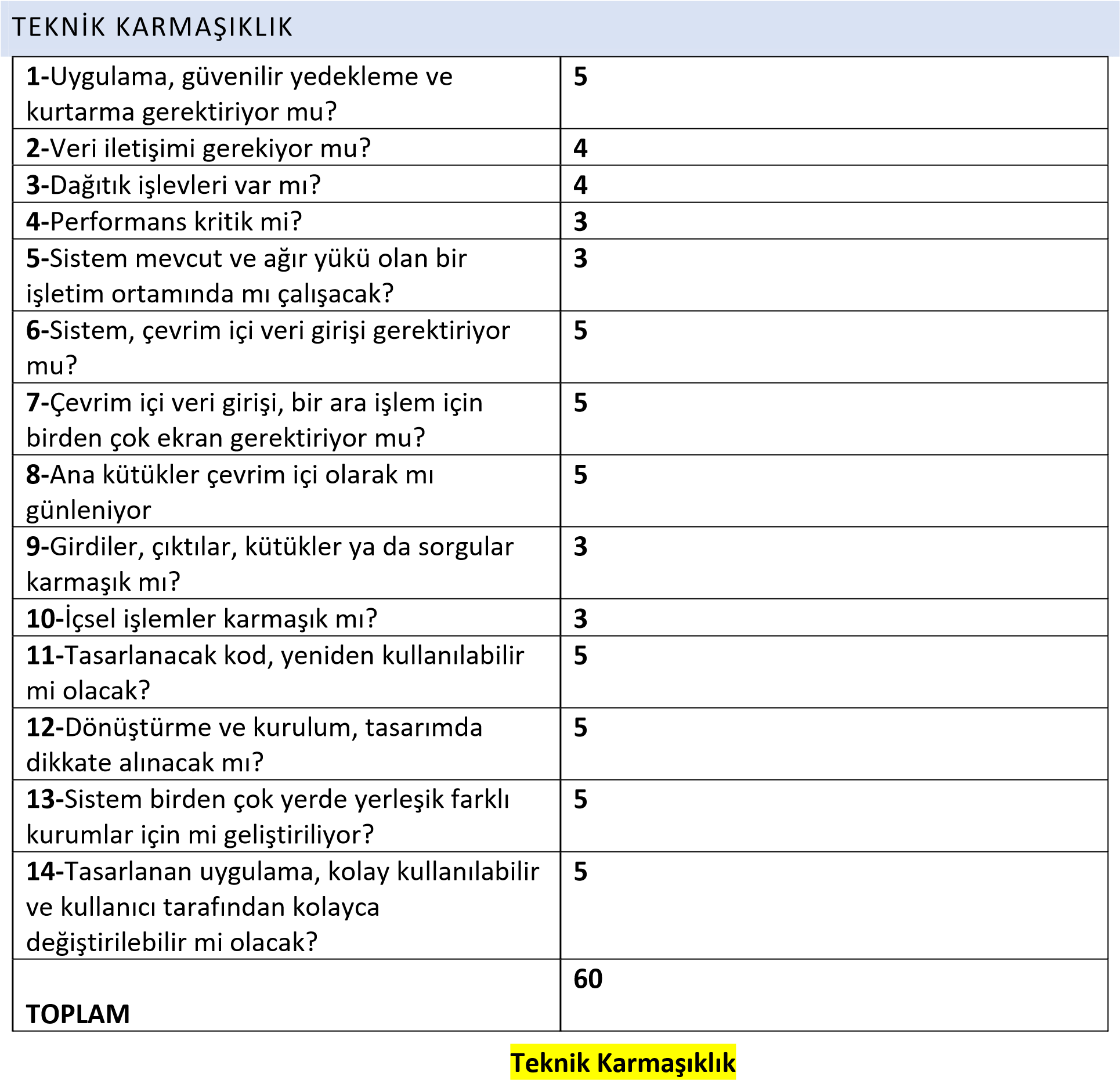
**Sistem Nasıl Çalışacak?**

İlaç hakkında detaylı bilgiye erişmek isteyen müşteriler sisteme girer. Gelen müşteriler sistem tarafından kaydedilir. Sisteme giriş yapan müşteri taradığı ilaç hakkında detaylı bilgiyi görüntüler. Yapmak istediği işlemi seçer. Yaptığı işlemi sisteme kaydeder. Taranan ilaç kontrol edilir ilaç doğru değilse tekrar yazması istenir. Süreç tamamlandıktan sonra bilgi verme adımına geçilir. Müşteri eczzane hizmetlerinden yararlanabilmektedir. Otomasyondan çıkış yapmak istediğinde çıkış yap butonu sayesinde sistemden çıkışı yapılır.

**Maliyet Kestirim Dokümanı:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ölçüm Parametresi** | **Sayı** | **Ağırlık** | **Toplam** |
| Kullanılan Girdi Sayısı | 30 | 5 | 150 |
| Kullanıcı Çıktı Sayısı | 25 | 4 | 100 |
| Kullanıcı Sorgu Sayısı | 5 | 5 | 25 |
| Kütük Sayısı | 20 | 10 | 200 |
| Dışsal Ara yüz Sayısı | 25 | 3 | 75 |
| Ana İşlev Nokta Sayısı |  |  |  |

**Ana İşlev Hesabı Tablosu**



**0: Hiçbir etkisi yok 2: Etkisi var 4: Önemli etkisi var**

**1: Çok az etkisi var 3: Ortalama etkisi var 5: Mutlaka olmalı,kaçınılmaz**

**KOD SATIR SAYISI HESABI:**

İN = AİN \* (0.65 \*0.01 \*TKF ) TKF=60 AİN=550

İN=550\*(0.65\*0.01\*60)

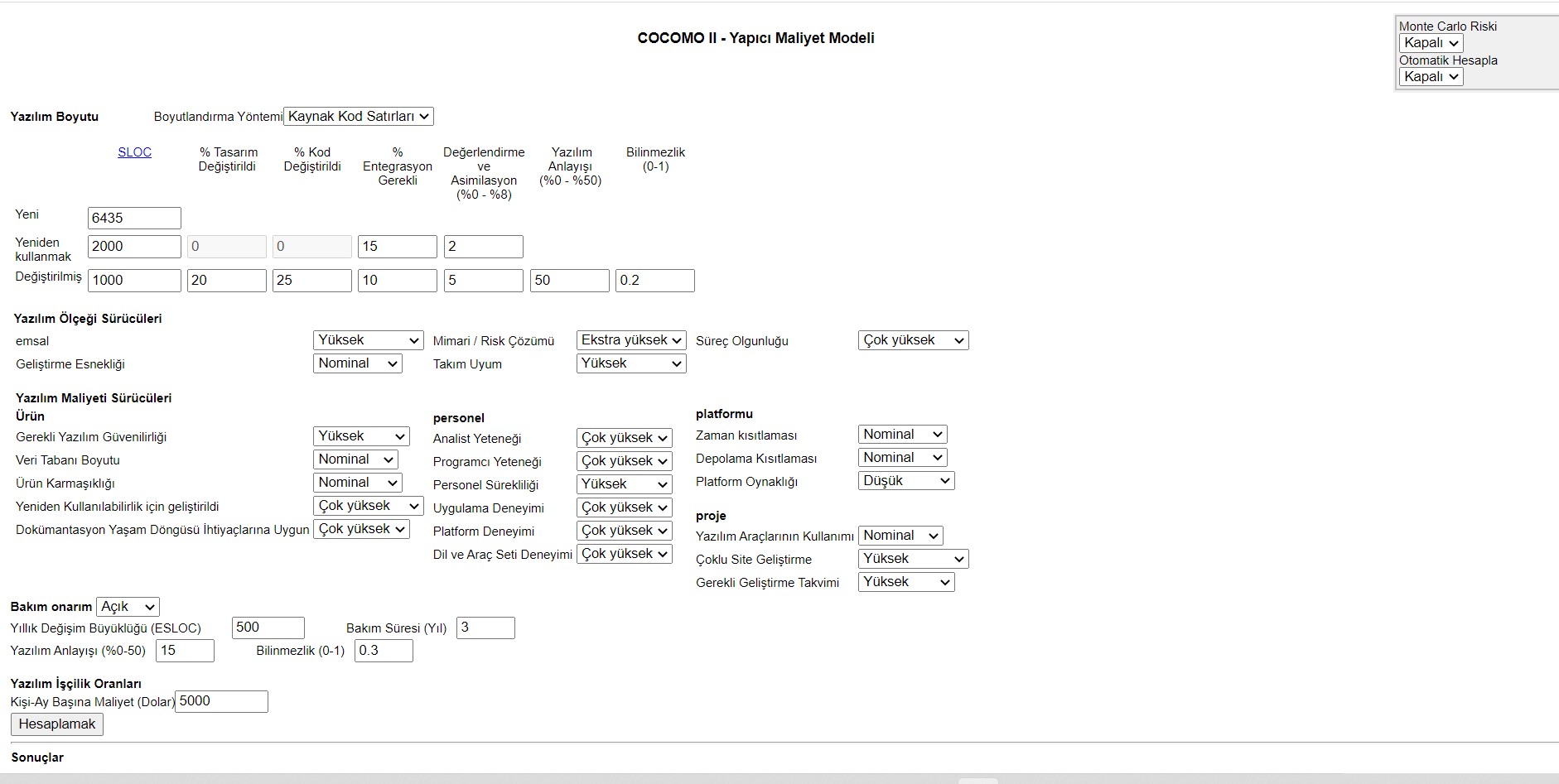
İN=214.5

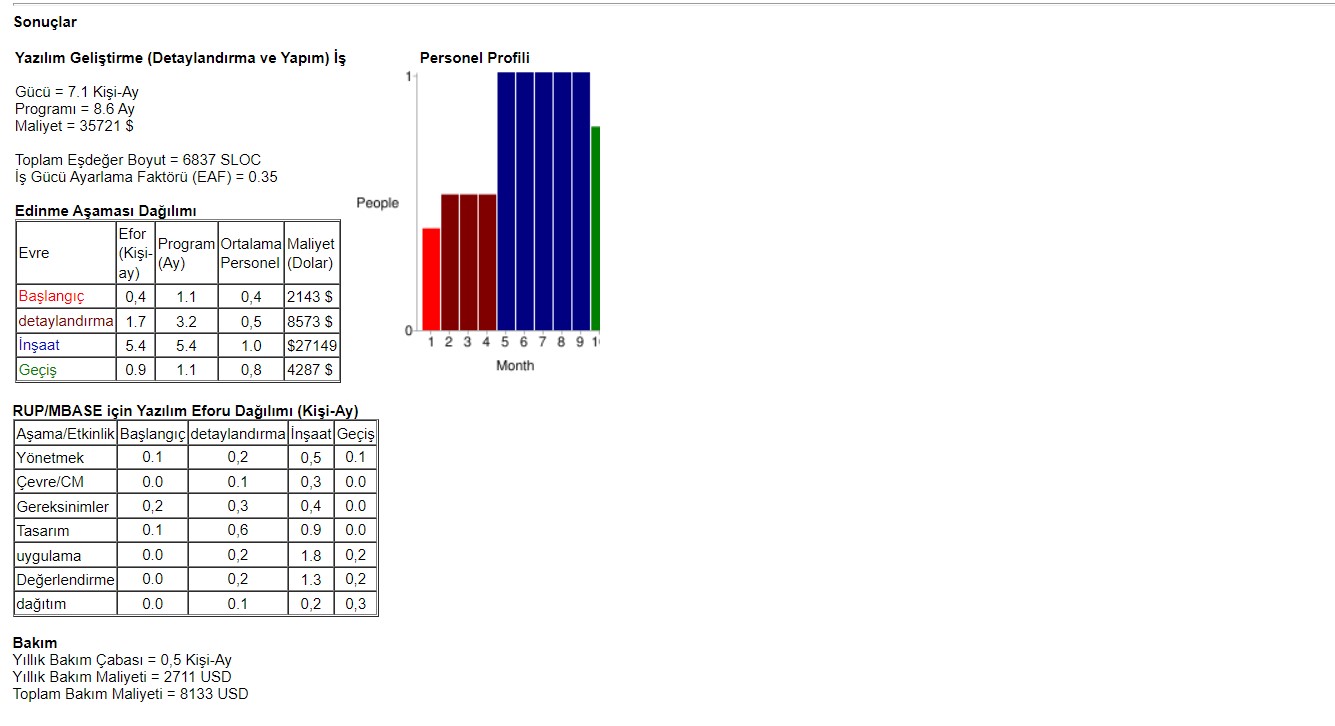
SATIR SAYISI= İN \*30

SATIR SAYISI=214.5\*30=6435

**SATIR SAYISI 6435 SATIR OLARAK BULUNUR.**

Proje Maliyet hesabı için hızlı yöntemlerden biri olan COCOMO—II maliyet modeli kullanılmıştır.





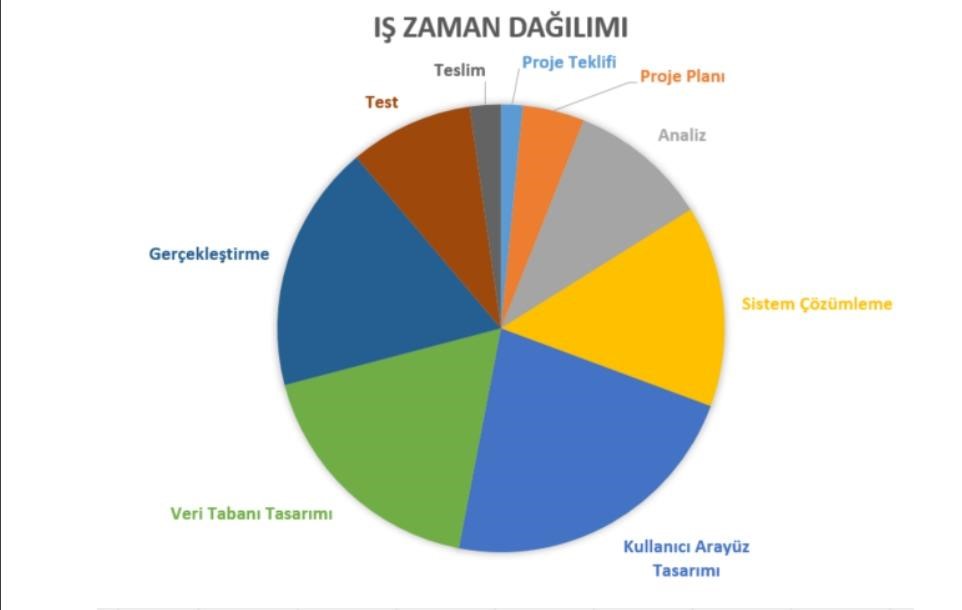
**Cocomo Hesaplamalar**

COCOMO II modelinin öncelikli amacı, modeli sürekli değiştirmek için yetenekleri oluşturmak ve kantitatif analitik yapı, teknikler ve araçlar sağlamaktır. Ayrıca, yazılım teknolojisi iyileştirmelerinin, yazılım geliştirme yaşam döngüsü masrafları üzerindeki etkilerini inceleme yeteneğine de sahipti.

**2.3 Proje Zaman-İş Planı**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Hafta-1** | **Hafta-2** | **Hafta-3** | **Hafta-4** | **Hafta-5** | **Hafta-6** | **Hafta-7** |
| **Proje Planı** |  |  |  |  |  |  |  |
| **Analiz** |  |  |  |  |  |  |  |
| **Sistem Çözümleme** |  |  |  |  |  |  |  |
| **Ara yüz Tasarımı** |  |  |  |  |  |  |  |
| **Gerçekleştirim** |  |  |  |  |  |  |  |
| **Test** |  |  |  |  |  |  |  |
| **Hata Giderme** |  |  |  |  |  |  |  |
| **Sunum** |  |  |  |  |  |  |  |

Şekil 2.3 Proje İş - Zaman Çizelgesi



Şekil 2.4 İş Zaman Dağılımı

**2.4 Proje Ekip Yapısı**

**Proje Yöneticisi**

Merve Nilgün Yılmaz

**Sistem Tasarımcı** Merve Nilgün Yılmaz

**Sistem Yöneticisi** Merve Nilgün Yılmaz

**Sistem Çözümleyici** Merve Nilgün Yılmaz

**Veri Tabanı Yöneticisi** Merve Nilgün Yılmaz

**Yazılım Ekip Lideri** Merve Nilgün Yılmaz

**Web Tasarımcı** Merve Nilgün Yılmaz

**Kalite Uzmanı** Merve Nilgün Yılmaz

Şekil 2.5 Proje Ekibi

**Görev Tanımları**

Projeyi hem teknik hem de iş-model seviyesinde yönetebilecek nitelikte ; görevleri, riskleri ,kaynakları ve kalite standartlarını göz önünde bulundurarak projeyi planlamak, genel takım organizasyonunu sağlamak ve altındaki teknik ekibi yönlendirip, motive etmektir.

**PROJE YÖNETİCİSİ**

Dokümantasyon ve Raporlamanın Hazırlanması

**SİSTEM ÇÖZÜMLEYİCİ**

Bilgisayar sistemlerinin yönetiminden sorumlu olan kişidir. Bilgisayar sistemlerinde ortaya çıkan sorunları gidermek, sorunları kontrol etmek, sistem güvenliğini sağlamak ve düzenlemek gibi birçok çalışmayı yürütür

**SİSTEM YÖNETİCİSİ**

**SİSTEM TASARIMCI**

Sistem çözümleyicinin tanımladığı gereksinimleri mantıksal, ekonomik ve pratik sistem tasarımlarına dönüştürerek ilgili programların yazılabilmesi için

**VERİ TABANI YÖNETİCİSİ**

Veri Tabanı sistemlerinin oluşturulması ve tasarlanması, yazılıma entegrasyonu ile ilgilenir

**KALİTE UZMANI**

Ürünlerin kalitesinin, test prosedürlerinin ve üretim sırasında oluşan kalite sorunlarının takibini yapan, ilgili departmanlara raporlayan ve sorunu çözüme kavuşturmak için çeşitli çalışmalar gerçekleştiren nitelikli kişidir

**WEB TASARIMCI**

Grafik Tasarım Çalışması Web Sayfalarını Kodlaması İçerik Yönetim Sistemi Entegrasyonu Web Sitesinin Test Süreci Web Sitesinin Yayınlanmasından sorumludur.

Yazılım geliştirme ekibine hedefler doğrultusunda liderlik eden, çalışma alanında ekibin motivasyonunu yükselten ve ekibi yönlendiren kişidir. Hiçbir programlama dili ile sınırlı olmaksızın tüm ekibi yazılım tasarım süreçleri doğrultusunda hedefe doğru yönlendirir.

**YAZILIM EKİP LİDERİ**

Şekil 2.6 Görev Tanımları

**2.5 Önerilen Sistemin Teknik Tanımları**

**Kullanılan Teknoloji**

* Sistemdeki bilgileri tutması için veri tabanı kullanılır.
* Kullanıcı kayıt sistemi için şifreleme yöntemleri kullanılır.
* Oluşturulan otomasyon yazılımına web site üzerinden kolayca erişme imkanı sağlanır.
* İlaçlardaki basılı barkodları okuyabilen, barkodun içerdiği verilerin kodunu çözebilen ve verileri bilgisayara gönderebilen optik bir tarayıcı sistemi kullanılır.



Şekil 2.7

**2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları**

**PROGRAMLAMA ARAÇLARI**

**ÇÖZÜMLEME VE DESTEK ARAÇLARI**

\*PHP \*C# \*JAVA

\*REACT \*MYSQL

\*HTML \*BOOTSTRAP

\*Adobe Dreamweaver

\*Adobe Flash CS5

\*Adobe Photoshop CS5

**DESTEK ARAÇLARI**

**SINAMA ARAÇLARI**

\*Microsoft Windows 11

\*ANDROID

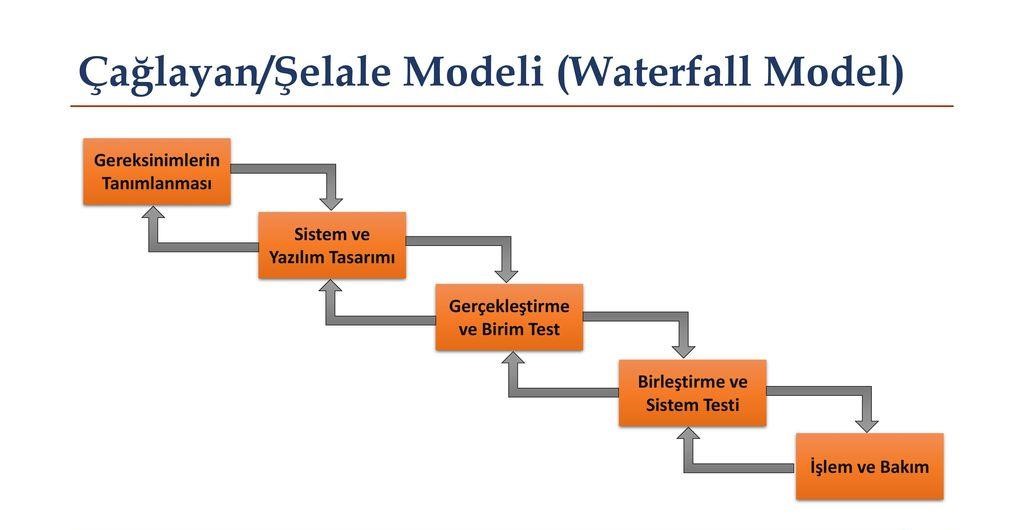
\*GOOGLE CHROME

\*INTERNET EXPLORER

Şekil 2.8

**2.7 Proje Standartları, Yöntem ve Metodolojiler**

Projede gereksinimler değerlendirildiğinde başlangıçta belirlenen gereksinimlerin ileriki aşamalarda değişmesi beklenmektedir. Projemde değişiklik olmamasından ötürü Çağlayan (Waterfall) modeli ile modellenebilir.



Şekil 2.9 Şelale Model

**1.Gereksinimlerin Analizi ve Tanımlanması:** **Sistem servisleri, kısıtlamaları ve hedefleri, kullanıcılarla istişare edilip belirlendikten sonra ayrıntılı olarak tanımlanarak sistem spesifikasyonu oluşturulur.**

**2.Sistem ve Yazılım Tasarımı**: **Sistem tasarım süreci, gereksinimleri donanım ya da yazılım sistemlerine paylaştırılır. Kapsamlı bir sistem mimarisi oluşturulur. Yazılım tasarımı, temel yazılım sistemi soyutlamaları ve bunların ilişkilerinin belirlenerek açıklanmasını içerir.**

**3.Gerçekleştirim ve Birim Sınama:** **Bu aşamada yazılım tasarımı, programlar ya da program birimleri olarak ortaya koyulur. Birim sınama, her birimin spesifikasyonuna uygunluğunun geçerlemesini içerir.**

**4.Bütünleştirme ve Sistem Sınama:** **Her program birimi ya da program, bütün bir sistem olarak bütünleştirilip sınanarak yazılım gereksinimlerinin karşılandığından emin olunur. Sınamadan sonra yazılım sistemi, müşteriye teslim edilir.**

**5.İşletim ve Bakım:** **Normal koşullarda en uzun yaşam döngüsü evresidir. Sistem, bu evrede yüklenerek uygulamaya geçirilir. Bakım, yaşam döngüsünün önceki aşamalarında fark edilmeyen hataların düzeltilmesi, sistem birimlerinin gerçekleştirilmesini içerir**

**Ne İçin Kullanıldığı**

**Çıktı**

**Kulllanılan Yöntem/ Araçlar**

**Aşama**

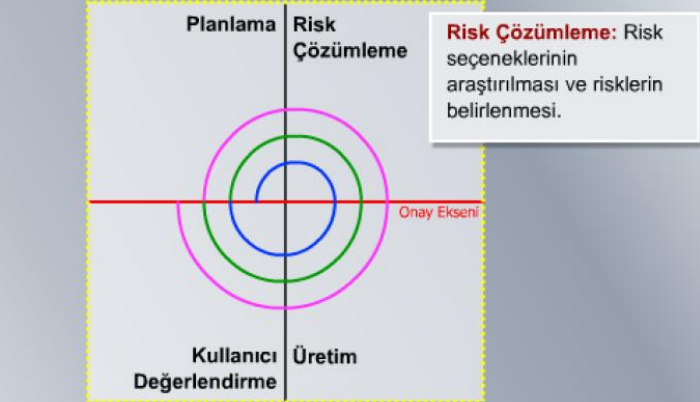
|  |
| --- |
| **• VERİ AKIŞ**  **ŞEMASI**  **• GÖRÜŞME**  **• MALİYET** |
| **• VERİ**  **SÖZLÜGÜ**  **• VERİ AKIŞ**  **ŞEMASı** |
| **• AKIŞA DAYALI**  **ÇÖZÜMLEME**  **• TASARIMA**  **DÖKME** |
| **• VERİ TABANI**  **TABLOLARI**  **• YAPISAL**  **ŞEMALAR** |

|  |
| --- |
| **• PROJE**  **YÖNETİMİ**  **• KAYNAK**  **KESTİRİMİ**  **• SÜREÇ** |
| **• SÜREÇ**  **ÇÖZÜMLEME**  **• VERİ**  **ÇÖZÜMLEME** |
| **• TASARIM**  **• AYRINTILI**  **TASARIM** |
| **• VERİ**  **TASARIMI**  **• GENEL**  **TASARIM** |

|  |
| --- |
| **• PROJE**  **PLANI** |
| **• SİSTEM**  **ÇÖZÜMLEM**  **E RAPORU** |
| **• BAŞLANGIÇ**  **TASARIM**  **RAPORU** |
| **• SİSTEM**  **TASARIM**  **RAPORU** |

|  |
| --- |
| **Planlama** |
| **Çözümleme** |
| **Çözümlemeden Tasarıma Geçiş** |
| **Tasarım** |

Şekil 2.10 Proje Aşamaları



Şekil 2.11 Proje gelişim modeli

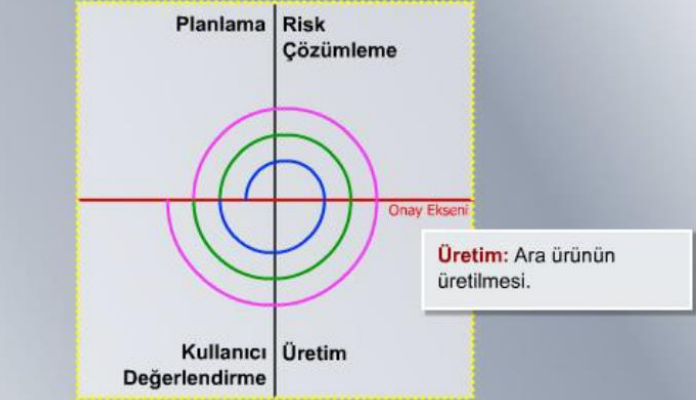


Şekil 2.12 Planlama Aşaması

Bu aşamalar yazılım geliştirmede kullanılmakta olan önemli unsurlardır. Kullandığımız model yazılım geliştirme metodolojilerinden Spiral Model dir. Bu modeli kullanmamızın amacı döngü şeklinde kontrol edip hataları en aza indirmektir.

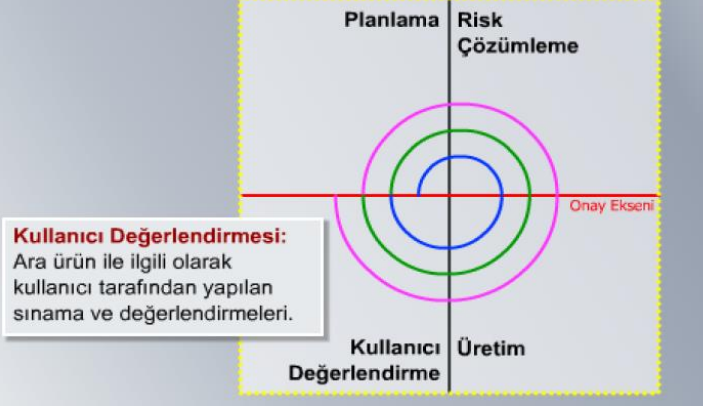


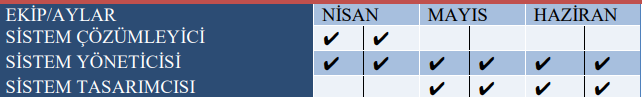
Şekil 2.13 Proje Risk Çözümleme Aşaması



Şekil 2.14 Proje üretim aşaması

Üretim aşaması uzun süren bir aşama olduğu için tasarımı da kapsar ve projenin büyük çoğunluğu üretim aşaması bittiğinde tamamlanmış olur. Bu aşamada ana ürün üretilir ve proje bu aşama sonunda bitmeye yaklaşmıştır.

 Şekil 2.15 Kullanıcı değerlendirme Aşaması

Şekil 2.16 Ekip yapısı zaman planı

**2.8 Kalite Sağlama Planı**

ÜRÜN KALİTESİ

ÜRÜN GELİŞTİRME

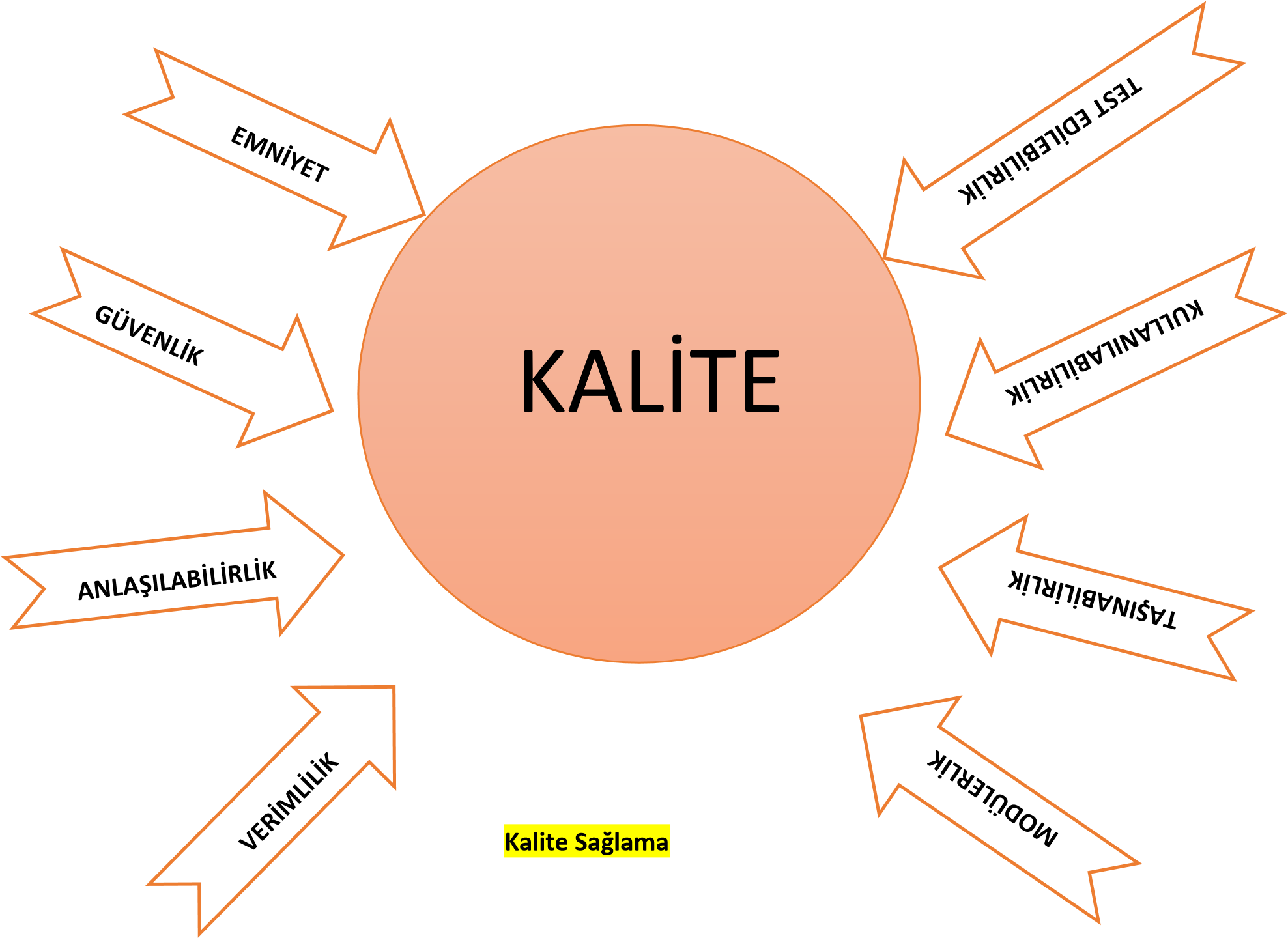
SÜRECİ TANIMLA

STANDARTLAŞTIR

İYİLEŞTİRME

Kalite Tamam mı?

Şekil 2.16 Sürece Dayanan Kalite



**1.Ekonomi**: Ekonomik açıdan hiçbir ekstrası yoktur.

**2.Tamlık**: Projede herhangi bir açık olmamalı ve programda bulunan tüm butonlar, textler vs. çalışır ve tamdır.

**3.Yeniden Kullanılabilirlik**: Otomasyon her koşulda tekrardan düzenlenip kullanılabilecek.

**4.Etkinlik**: Kullanıcı sistemin her alanına hakim olduğu için sistemi etkin bir biçimde kullanacak.

**5.Bütünlük**: Admin sistemin tüm kısımlarına hakim olacak ve program bir bütün halinde çalışacak.

**6.Güvenilirlik**: Otomasyon gerekli güvenlik önlemlerinin alınması yanı sıra şuan devlet bünyesinde bulunan çok yüksek güvenlik önlemli serverlarda saklanacaktır.

**7.Modülerlik**: Modülerlik otomasyonun her seviyesindeki kişinin ayrı ayrı sayfalardan söz sahibi olmasını sağlar.

**8.Belgeleme**: Bu belgeden de anlaşılacağı üzere tam anlamıyla sistemin özeti olacak bu 18 doküman oluşturulmuştur.

**9.Kullanılabilirlik**: Kullanılabilirlik olarak her seviyedeki insana hitap edeceğinden zor renkler karmaşık sistemlerden kaçınılmıştır.

**10.Temizlik**: Temiz kod yazımı ve sadelik

**11.Değiştirilebilirlik**: Admin paneline erişme yetkisi olan ve sistem hakkında bilgisi olan herkes sistemde değişiklik yapabilecek.

**12.Esneklik**: Proje farklı platformlarda ve internet üzerinden çalışacağından gayet esnektir.

**13.Genellik**: Proje Türkiye genelinde kullanılacaktır.

**14.Sınanabilirlik**: Projenin her koşulda çalışabilmesi

**15.Taşınabilirlik**: Sistem internet üzerinden kullanılacağından herhangi bir özel cihaz gerektirmez ve istenilen cihazlarda taşınabilir ve kullanılabilir.

**16.Birlikte Çalışılabilirlik**: Bu projedeki en büyük sıkıntı olacak veri girişi şuanda var olan ve her bireyin bilgilerinin saklayan sistemle birleşik ve eş zamanlı çalışmakta.

**2.9 Konfigürasyon Yönetim Planı**

Sisteme veri ilavesi, rapor formatlarının değişmesi , kullanıcı rollerinde değişiklikler, ilaç bilgisi değişikliği gibi durumların oluşması olası olduğundan konfigürasyon planı hazırlandı.

Proje planlamanın erken safhalarında proje takımı, konfigürasyon yönetiminden sorumlu olan kişi ve proje yöneticisi konfigürasyon kontrolü altındaki bileşenleri tanımlamalıdırlar. Bu konuda Proje Kontrol Listesi yada aktivite listesi kullanılabilir.

Dikkat edilmesi gerekenler genel olarak olarak;

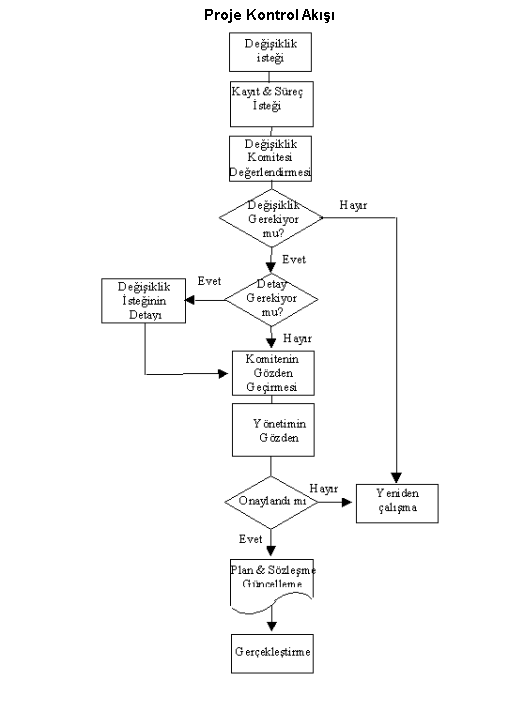
* Donanım konfigürasyonu, sistem mimarisi, iletişim .
* Yazılım, kod, tasarım dokümanları, test planı, yazılım gözden geçirme verileri.
* Proje Yönetimi Planı (programlar, bütçeler, sözleşmeler), destek fonksiyon planları,
* yazışma ve diğer kullanılan dokümanlar

**2.10 Kaynak Yönetim Planı**

Mevcut bir kaynak olmadığı için kaynak olarak elimizde sadece bu proje dokümantasyonu bulunmaktadır.

Kaynak yönetiminde şu hususlar dikkate alınacaktır;

* Yeterli hata bulunuyor mu?
* Kalite beklendiği gibi mi?
* Eğer yeterli sayıda hata bulunmuyorsa veya kalite beklenenden daha iyi görünüyorsa, kalite gerçekten daha iyi olabilir
* Bu projede yeni bir süreç iyileştirmesi kullanılıyor mu?
* Proje elemanları yeni bir eğitim aldı mı?
* Yeni bir araç kullanılıyor mu?
* Yeterli vakit harcanıyor mu?
* İnceleyiciler yeterli hazırlık yapıyor mu?
* İnceleme toplantısında çözüm bulmak için zaman kaybediliyor mu?



Şekil 2.14 Konfigürasyon Yönetimi Süreç Akışı Modeli

**2.11 Eğitim Planı**

Projede Kazanılacak en önemli olaylardan biri de eğitimdir. Projede eğitimin en önemli safhası kullanıcıların ara yüzleri etkin bir şekilde kullanmasının sağlanmasıdır. Veri tabanı ile ilgili bilgiye de sahip olmaları sağlanmalıdır. Eğitimlerin uygulamalı videolar aracılığıyla yapılması kararlaştırılmıştır. Bunun için program ara yüzüne eğitim modülü yerleştirilecek merkez veri tabanından kısa yollar ile kısa videolarla programın kullanımı kullanıcılara aktarılacaktır. Diğer taraftan bakım süresinde online yardım masası kurulacak ve uzaktan canlı destek gerektiğinde verilecektir. Programda yer alacak video içerikleri aşağıda açıklanmıştır. Eğitim için bu sayede bir zaman planlamasına gerek kalmamaktadır. Ancak veri tabanı için gerektiğinde kullanıcı sistemlerine ücreti karşılığında yerinde teknik destek verilecektir Programlama dili ve kullanılan kod yazma editörüne hakım olamaz ise yazılımcı bu proje sağlam bir şekilde neticelenemez proje kapsamı boyunca alınacak olan eğitimler ;

• SQL SERVER

• PHP

• HTML , CSS ,JS,BOOTSTRAP

• REACT

Gerekli Eğitimler bunlardır.

**2.12 Test Planı**

Proje test ekipleri ve görevleri şu şekildedir;

* + Kullanıcı giriş ve kayıt ol sayfaları kontrol edilecek
  + Gerekli bağlantılar sağlanıyor mu bakılacak
  + Veri akışı gerçekleşiyor mu kontrol edilecek

**Test sorumlusu:** Kullanıcılar tarafından çeşitli testlerden geçirerek olası hataların tespiti yapılarak yazılım iyileştirmesi yapılacaktır. Bunun için iş planında belirlenen zaman diliminde testlerin tamamlanması sağlanacaktır.

**Yazılım Ekibi Test Sorumlusu:** Sistemin tasarımında ve birleştirilmesi ve teslimi süreçlerinde yazılımı geliştirme testine tabi tutacaktır. Bu geliştirme testinde projedeki diğer paydaşlarda görev alabilirler.

**Birim Testi:** Program birimleri ve nesne sınıfları test edilecektir.

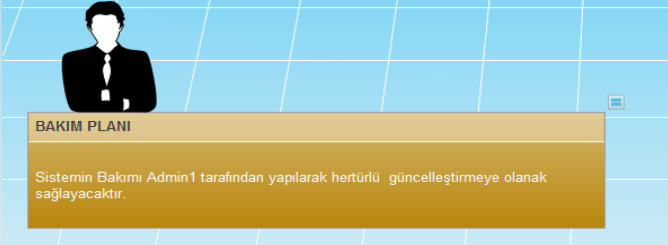
**Bileşen Testi:** Birkaç ayrı birim bir araya gelerek birleştirme yapılacak ve program ara yüzleri test edilecektir.

**Sistem Testi:** Sistem bir bütün halinde test edilecektir.

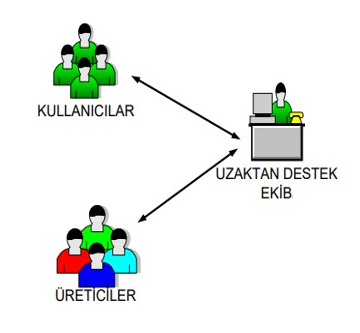
**Sistem Kullanıcıları:** Gereksinimlerin karşılanıp karşılanmadığı vb. konularda tümleştirilen programdan örnek müşteri kaydı, personel kaydı, oda girişleri yapacak ve sistem çıktılarının uygun olup olmadığı yönünde rapor hazırlayacaklardır.

**2.13 Bakım Planı**

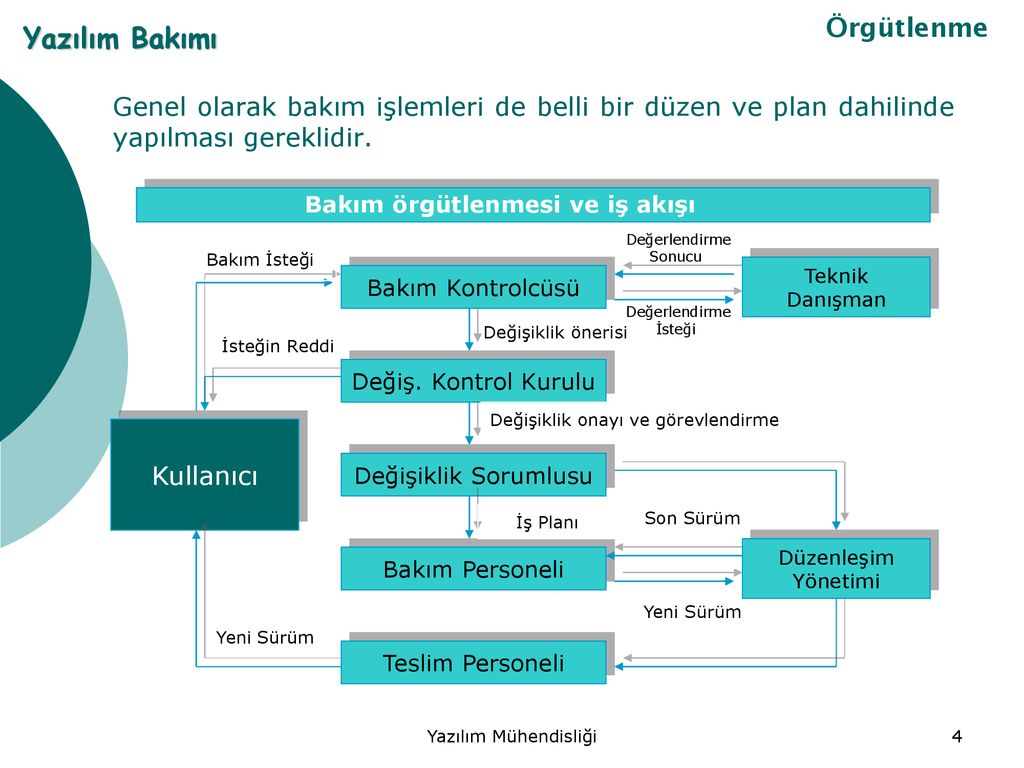
Projenin bakım planında projede olabilecek her türlü sorunlardan ve kilitlenmelerden admin tarafından kontrol edilip düzeltilecek yerdir.

****

Şekil 2.15 Proje Bakım Planı

****

Şekil 2.16 Proje Bakım



Şekil 2.17 Proje Bakım İş Akışı

3. SİSTEM ÇÖZÜMLEME

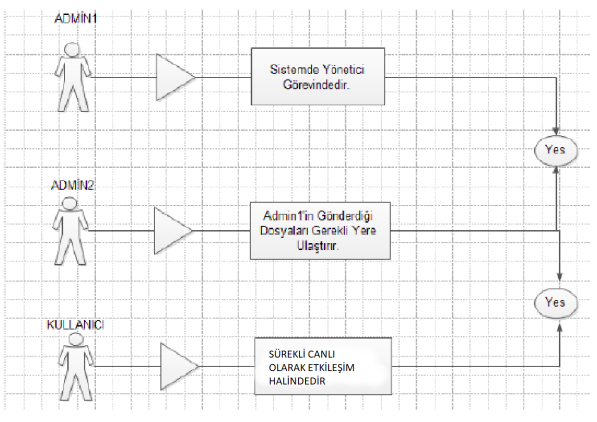
**3.1 Mevcut Sistem İncelemesi**

Mevcut sistemde müşteri talep ettiği ilaçların bilgisini listeler. Eczaneden ilaç almak isteyen müşteriye verilen reçete üzerinden ilaçları temin edilir ilaçların üzerinde bulunan QR kodu ile tüm işlemleri gerçekleştireceklerdir. Otomasyonumuz da çeşitli hizmetler de verilmektedir. Bu hizmetler ilaç sorgulama, İlaç hakkında birebir istediğin eczacıyla görüşme hizmeti ve canlı destek hizmetleridir. Eczaneci ilaç ekleme ,silme ve güncelleme işlemlerini yapabilmektedir. Sistem yöneticisi ise kullanıcı ekleme, çıkarma ve güncelleme işlemi yapabilmektedir.

* + 1. **Örgüt Yapısı**

Eczane bünyesinde çalışan insanlardan oluşan bir örgüt yapısı vardır.

* + 1. **İşlevsel Model**

****

Şekil 3.1 İşlevsel model

* + 1. **Veri Modeli**

Kullanıcı TC ve Reçete Bilgileri

ECZACI NO

CANLI DESTEK ODASI NO

FATURA BİLGİSİ VE İLAÇ YAN ETKİLERİ ÇIKTISI

Şekil 3.2 Veri Modeli

* + 1. **Varolan Sistemin Değerlendirilmesi**

Yukarıda da söylediğimiz gibi TC bilgisi var ve kullanıcılar bu şekilde e devlet ile bağlantılı bir şekilde sisteme giriş yaparak e reçetesi ile birlikte müsait olan eczacı ile görüşme yaparak işlemleri gerçekleştirecek.

REÇETE BİLGİSİ

FATURA VE YAN ETKİLERİ

KULANIC I

**3.1.5 Var Olan Yazılım/Donanım Kaynakları**

Hali hazırda bilgiler bilgisayar aracılığıyla oluşturulan bilgilerden yürütülmektedir.

**3.2 Gereksenen Sistemin Mantıksal Modeli**

**3.2.1 Giriş**

Mevcut sistemler incelendiğinde sonuca giden yolda birçok eksik bulunmaktadır. Bu sistemi Türkiye eczacıların standartlarına uydurmak bize kalıyor. Sistemin işlevsel modeli ile başlamak gerekirse.

**Müşteri**

**Sistem Yöneticisi**

**Eczane Çalışanı**

**ECZANE OTOMASYON SİSTEMİ**

Şekil 3.2 Bağlam Diyagramı

**3.2.2 İşlevsel Model**

****

SİSTEM YÖNETİCİSİNİN GÖNDERDİGİ DOSYALARI GEREKLI YERE YUKLER

SİSTEM YÖNETİCİSİ GÖREVLENDİRİR

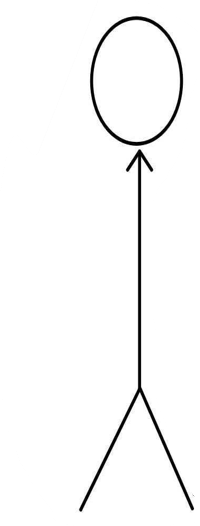
****

KULLANICI TC Sİ İLE GİRİŞ YAPARAK İŞLEMLERİNİ GERÇEKLEŞTİRİR

****

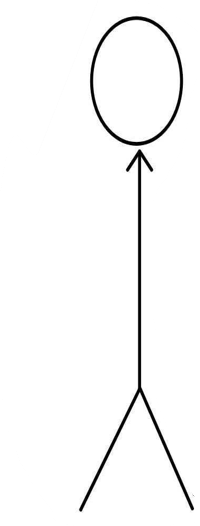
Şekil 3.3 işlevsel model

**Sistemin Kullanıcı ve Aktörleri**



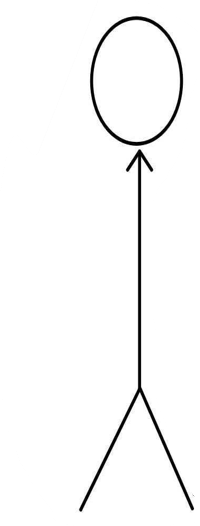
**OTOMASYONU KULLANAN KİŞİ**

**Müşteri**



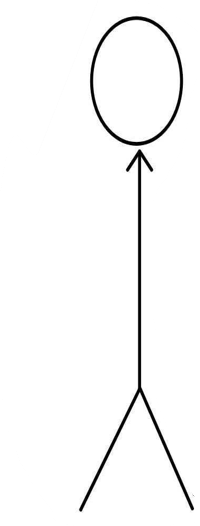
**MÜŞTERİ İŞLEMLERİ YAPAN KİŞİ**

**Eczaneci**



**OTOMASYON İŞLEMLERİ YAPAN KİŞİ**

**Sistem Yöneticisi**



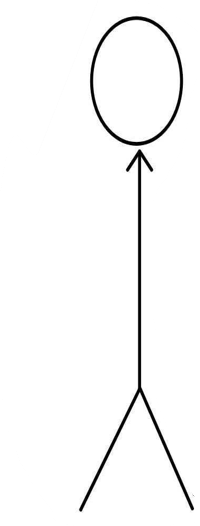
**Müşteri**

**<<uses>>**

İlaç taraması yapar.

Uygun bilgiyi seçer.

Otomasyon sistemlerinden yararlanır.

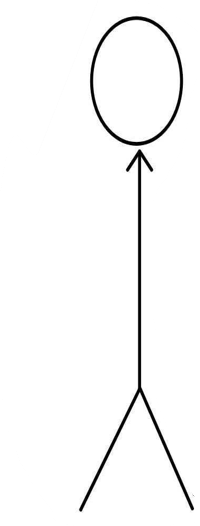


**<<uses>>**

**Eczaneci**

Müşterilerin ilaç bilgilerini alır ve kaydeder.

İlaç bilgilerini alır ve ilaç hakkında tüm bilgiyi sisteme girer.



**Sistem Yöneticisi**

**<<uses>>**

Eczaneci bilgilerini alır ve ekler.

Eczaneci bilgilerini alır ve siler.

Eczaneci bilgilerini alır ve günceller.

**Aktörler ve İşlemler**

**Use-Case 1: İlaç Seçmek**

* Müşteri kendine uygun ilaç bilgisini seçer.

**Use-Case 2: Şifre Belirleme**

* Otomasyon sistemine giriş yaparken kullanıcı şifre belirlemelidir.

**Use-Case 3: İlaç Talep Etme**

* Müşteri otomasyondaki hizmetlerden yararlanmak için sisteme giriş yapmalıdır. İstediği ilacı kendisi belirler. Talep edilen ilaç kısmının boş olmaması gerekmektedir.

**Use-Case 4: Eczacı Talebi**

* Müşteri istediği eczacıyla canlı görüşmeye karar verirse talep işlemi yapmalıdır.

**Use-Case 5: 7\24 Canlı Destek**

* Müşteri canlı destek seçeneği ile isteklerini rahat bir şekilde karşılamaktadır.

**Use-Case 6: Müşteri Ekle**

* Eczane çalışanı tarafından müşteri bilgileri alınarak müşteri ekleme işlemi yapılır.

**Use-Case 7: Müşteri Sil**

* Eczane çalışanı tarafından müşteri bilgileri alınarak müşteri silme işlemi gerçekleştirilir.

**Use-Case 8: Eczacı Kaydı Oluşturms**

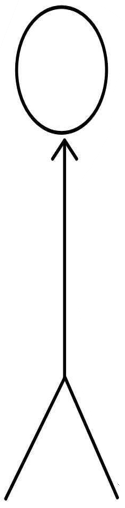
* Sistem yöneticisi eczacı bilgilerini alır ve eczacıyı sisteme kaydeder.

**Use-Case 9: Eczacı Kaydı Silme**

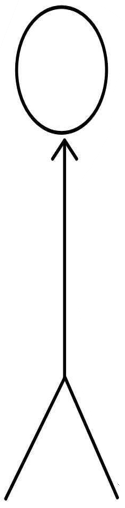
* Sistem yöneticisi eczacı bilgilerini alır ve eczayı sistemden siler.

**Use-Case 10: Eczacı Kaydı Güncelleme**

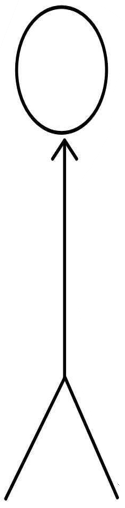
• Sistem yöneticisi eczacı bilgilerini alır ve eczacının bilgilerini günceller.



**Müşteri**



**Müşteri**



**Admin /Sistem Yöneticisi**

## **Activity Diyagram: İlaç seçme**

## C:\Users\90542\Desktop\WhatsApp Image 2022-06-13 at 13.16.55.jpeg

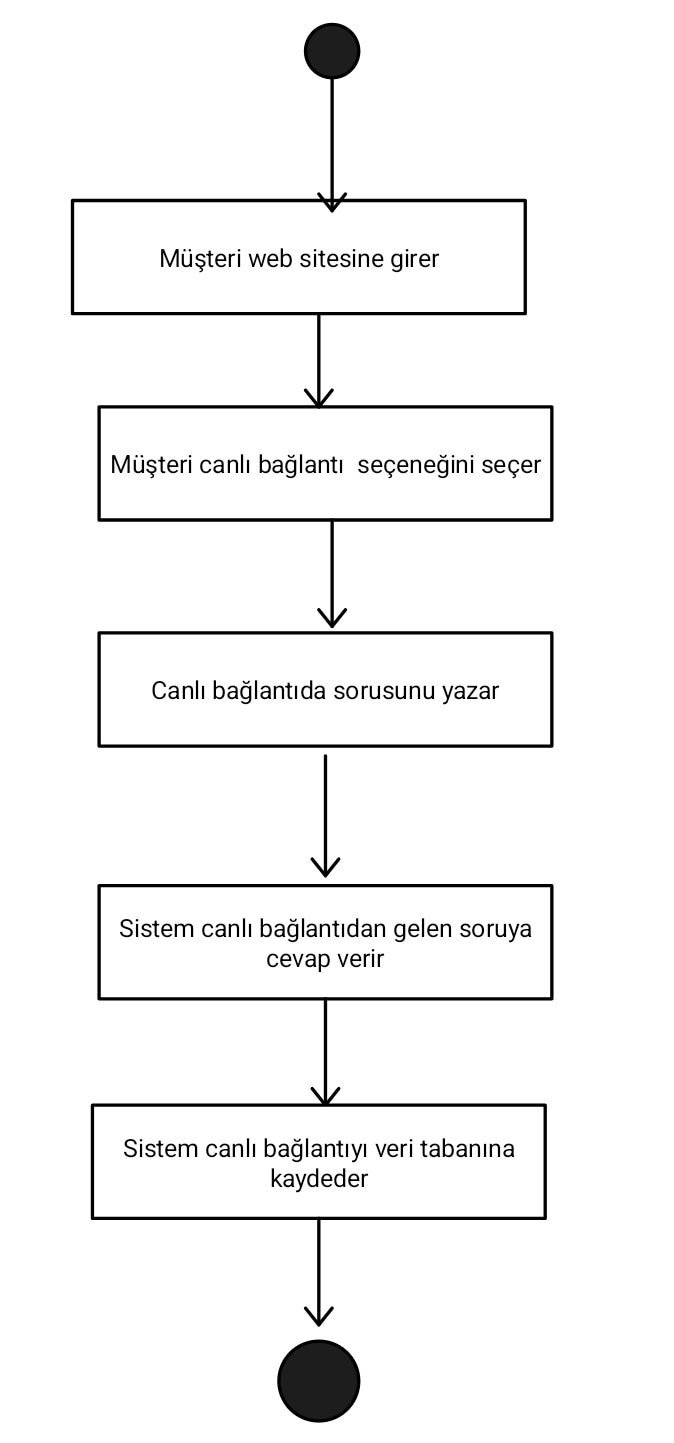
## **Activity Diyagram: Şifre Belirleme**

# 

# **Activity Diagram: İlaç Talep Etme**

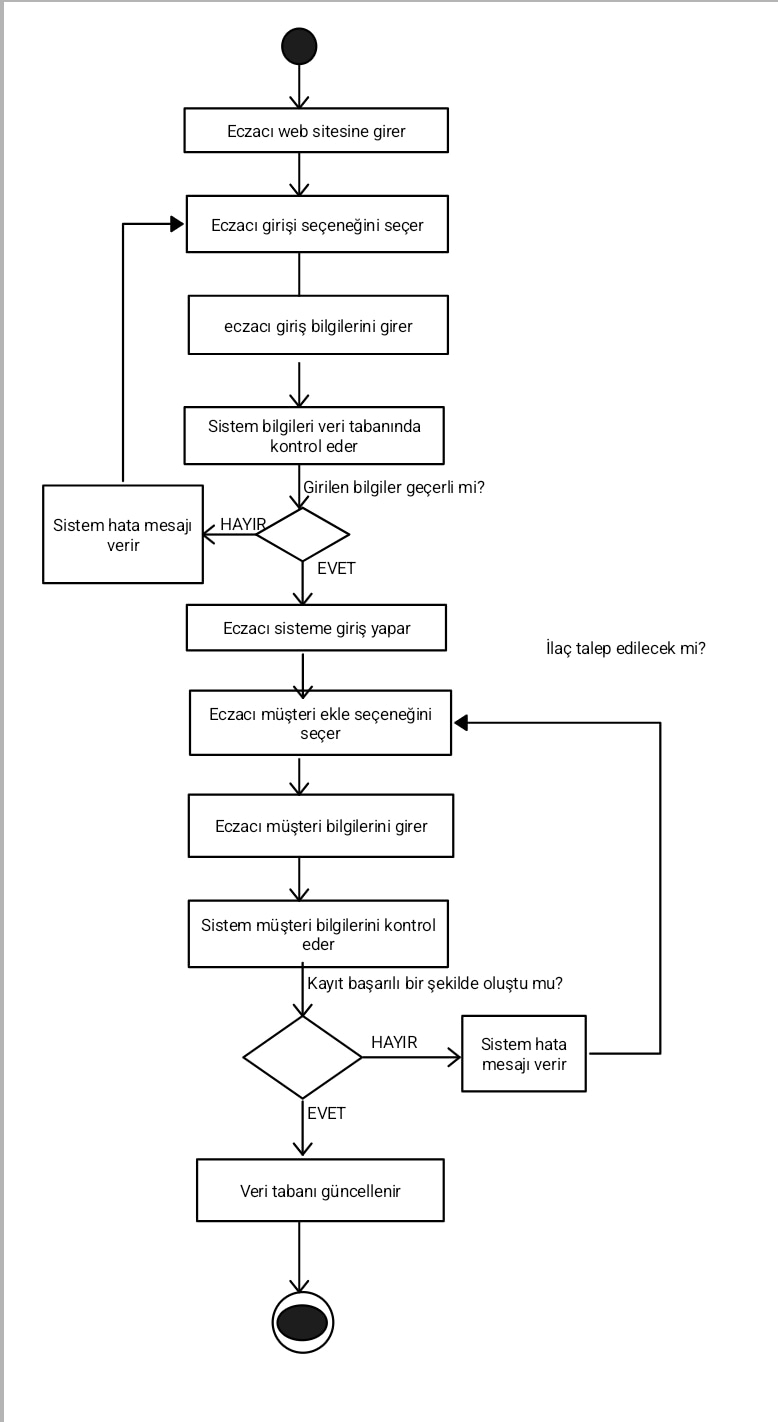
# **C:\Users\90542\Downloads\WhatsApp Image 2022-06-13 at 13.56.48.jpeg**

# **Activity Diagram: Canlı Bağlantı**

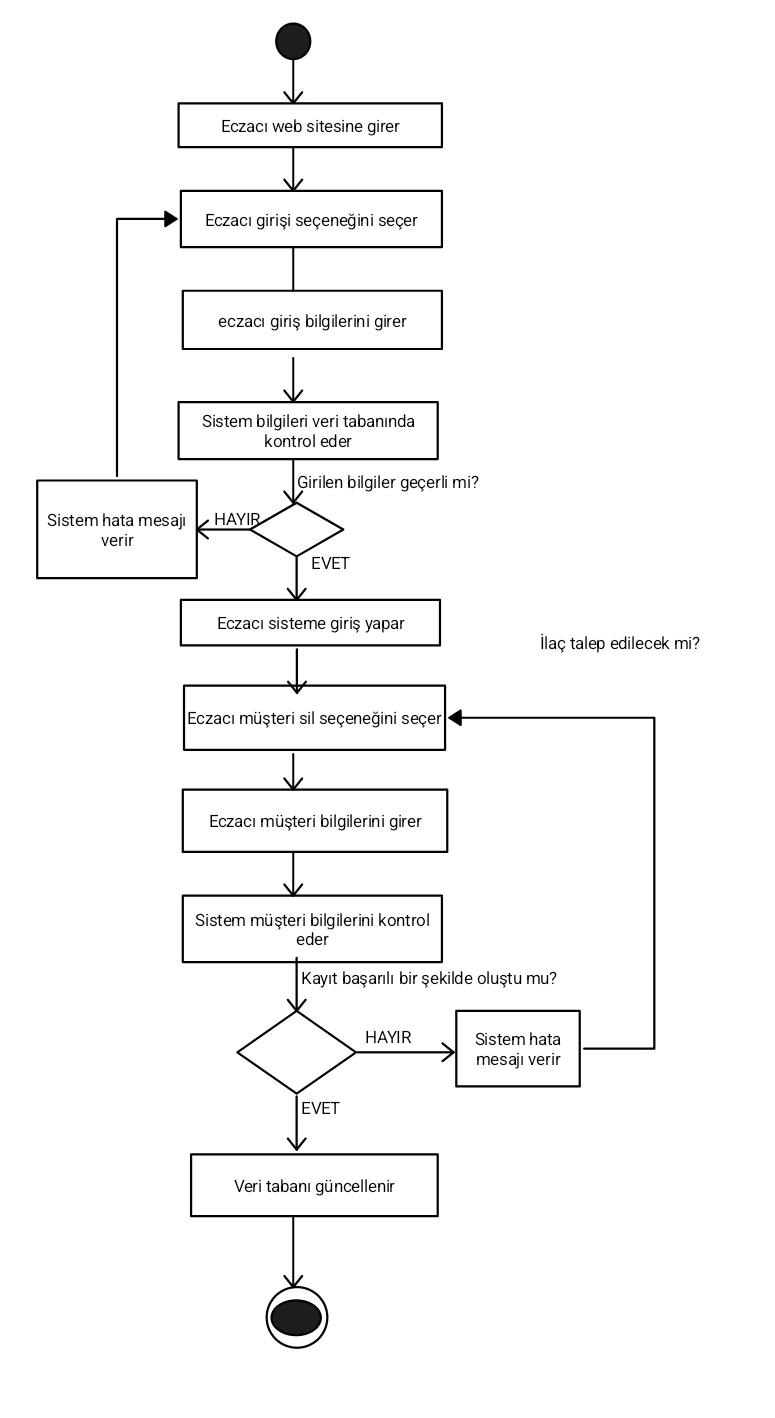


# 

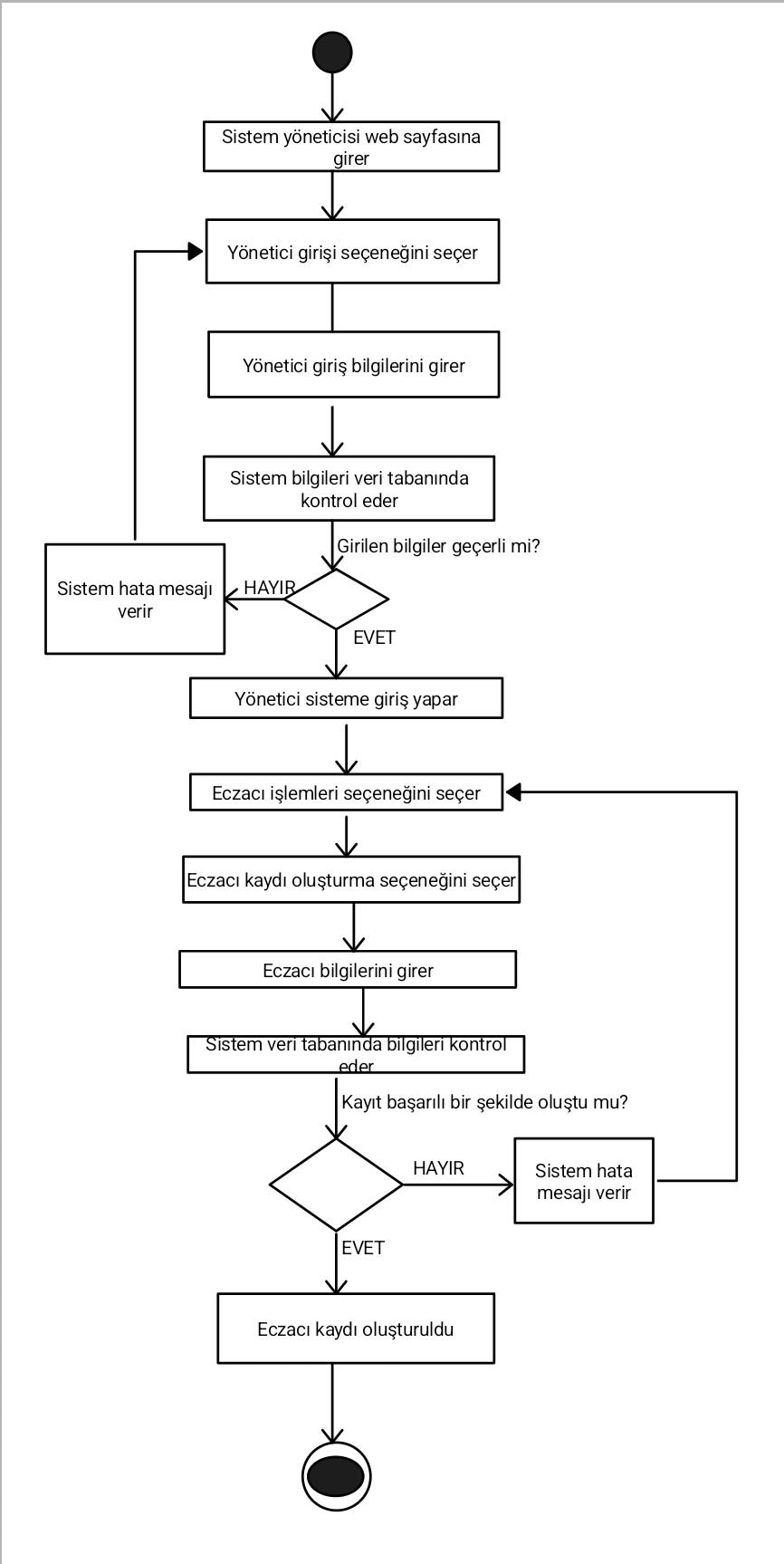
**Activity Diagram: Müşteri Ekleme**



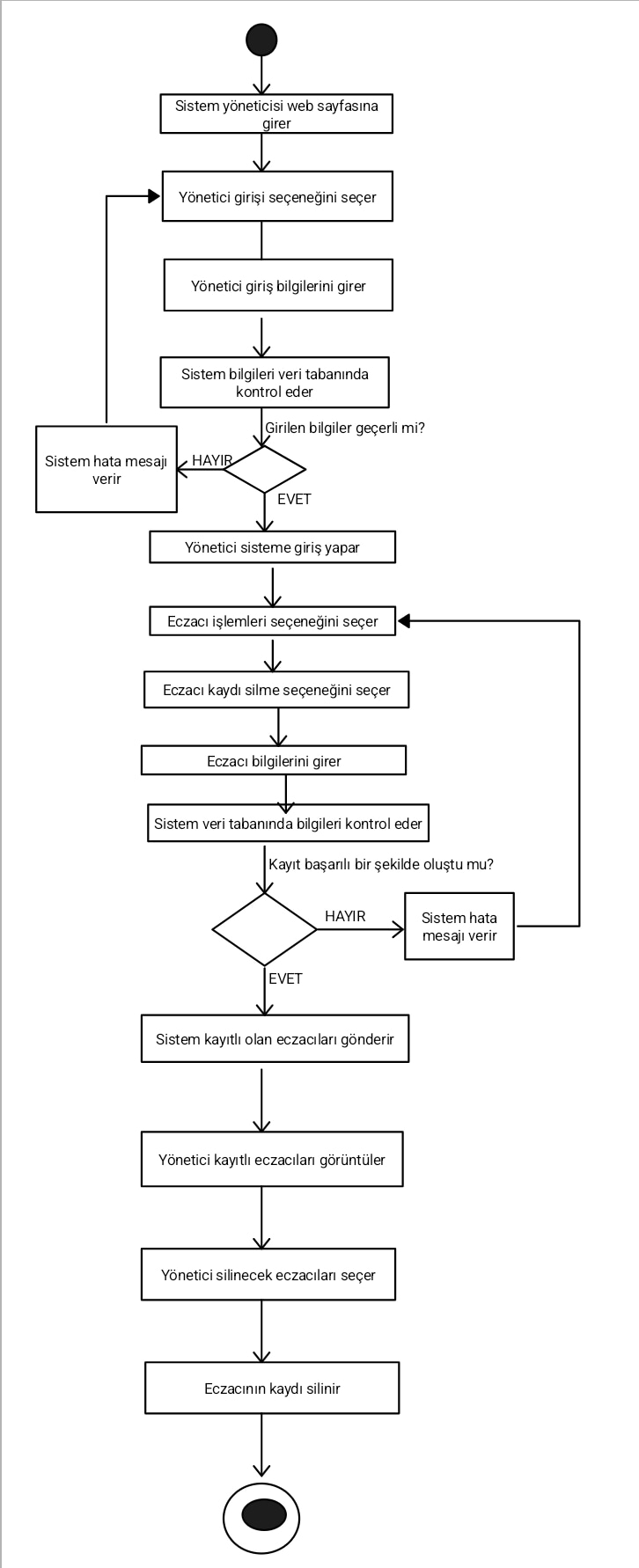
**Activity Diagram: Müşteri Silme**

****

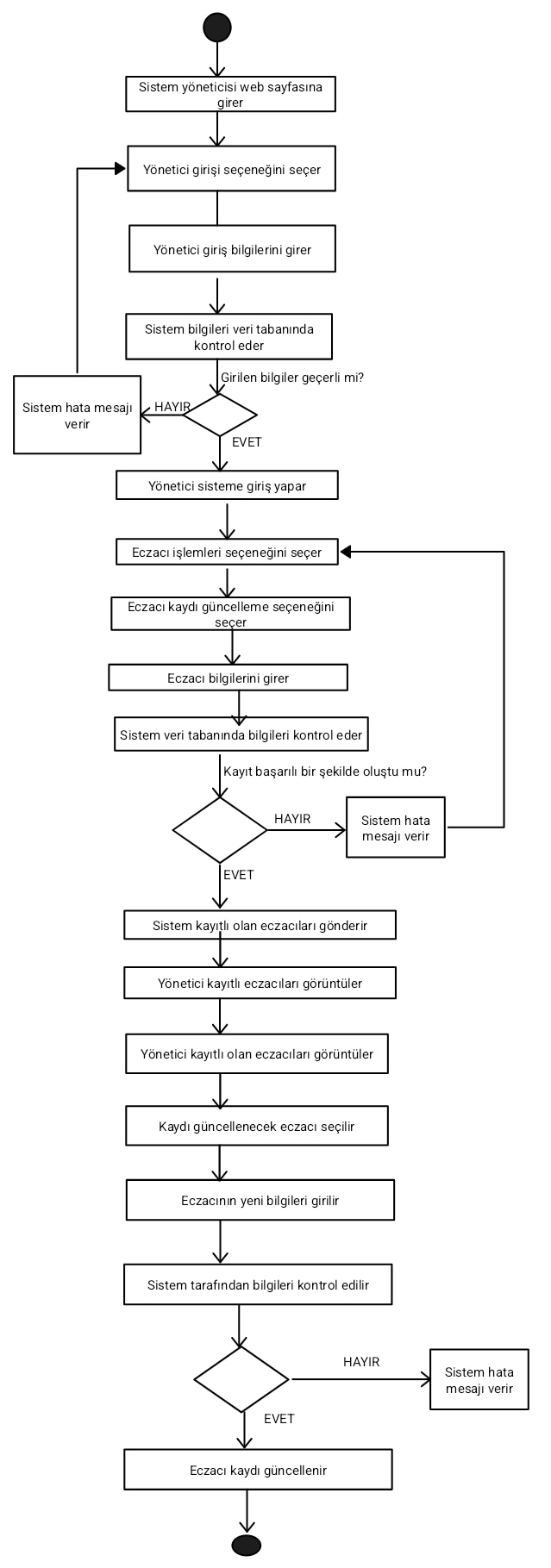
**Activity Diagram: Eczacı Kaydı Oluşturma**

****

**Activity Diagram: Eczacı Kaydı Silme**

****

**Activity Diagram: Eczacı Kaydı Güncelleme**

****

**3.2.3 Genel Bakış**

Genel hatlarıyla sistemi inceleyecek olursak sistem mevzuatında belirtilen tüm detayların iyi araştırılarak eksiksiz yapıldığı söylenebilir. Kurulacak sistem baştan da belirtiğimiz gibi gereksinimleri tam olarak belirlenmiş ve hatayı kabul etmeyen bir yönü vardır. Use-case diyagramlarında net olarak yetkiler tanımlanmış sorumluluklar paylaştırılmıştır. Belgelerin her aşamada berileri eksiksiz olarak giriş sağlanmış ve sonuç raporlarının oluşturulması sağlanmıştır. Son teknolojik motiflerin tasarımda kullanmasıyla modern bir görüntünün yanında güvenilir ürün oluşturma başarıyla sağlanmıştır.

**3.2.4 Bilgi Sistemleri /Nesneler**

**ADMİN1/Yönetici**: Yapının en üstünde bulunan ve en yetkili yapılanmadır. Admin2 ye yetki verir.

**ADMİN2/Eczacı**: Yapının ikinci nesnesidir. Kullanıcının isteğine, gerekli olan dosyaya göre ADMİN1’e gereksinimi bildirir.

**KULLANICI:** Yapının üçüncü nesnesidir . Sisteme giriş yaparak bir çok ilaç taraması yapabilir.

SEÇENEKLER

KULLANICI

KULLANICI

USER

DOSYA

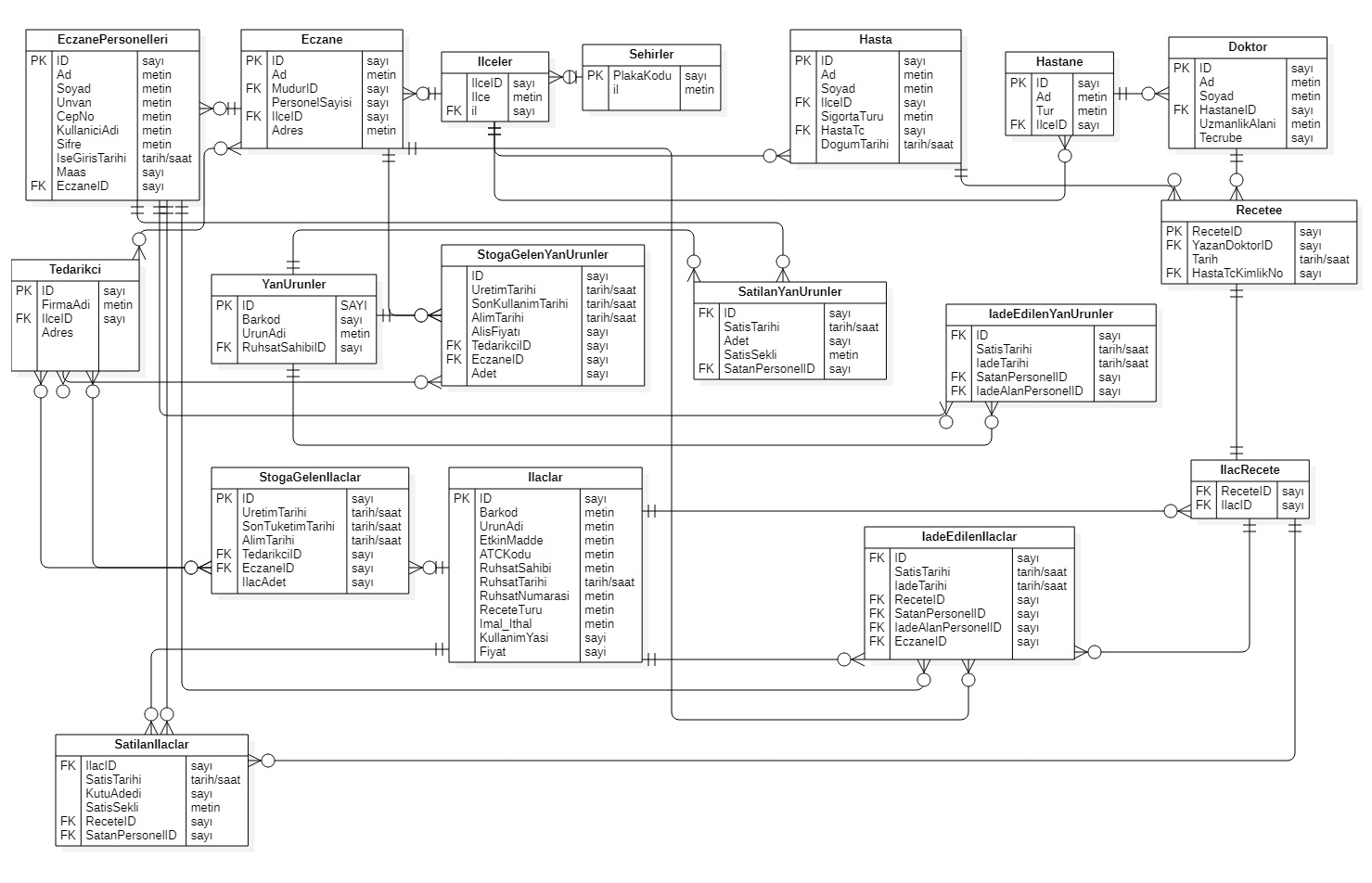
ADMİN

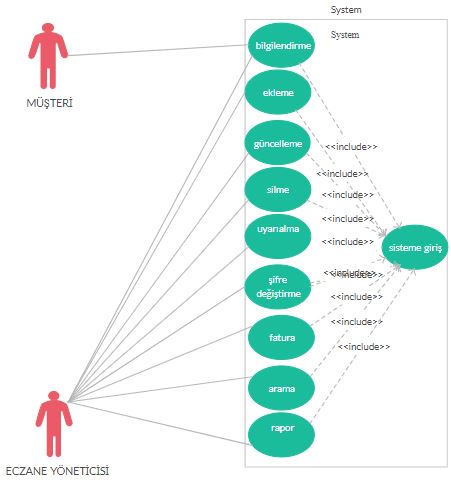
User

Kullanıcı

Adminmm

**3.2.5 Veri Modeli**

****

****

KULLANICI YA DA ADMİN SİSTEME GİRİŞ YAPAR

SİSTEM GİRİŞ EKRANI GÖRÜNTÜLENİR

GİRİŞ BUTONUNA BASILIR

İŞLEMLER GÖRÜLÜR

ONAYLAMA YAPILIR

Şekil 3.4 Veri Modeli

**3.2.6 Veri Sözlüğü**

**Eczane**

**Public int Ad**

**Public int MudurID**

**Private int PersonelSayisi**

**Public void Adres**

**EczanePersonelleri**

**Private int ID**

**Private int Ad**

**Private int Soyad**

**Private int CepNo**

**Private int KullaniciAdi**

**Private int Sifre**

**Private int Maas**

**Private int EczaneID**

**YanUrunler**

**Private int Id**

**Public string Ad**

**Private int Barkod**

**Private int urunAdi**

**Ilaclar**

**Private int ID**

**Private int Barkod**

**Private int UrunAdi**

**Private int EtkinMadde**

**Private int KullaniciAdi**

**Private int ATCKodu**

**Private int RuhsatSahibi**

**Private int RuhsatNumarası**

**Private int ReceteTuru**

**Private int KullanimYasi**

**Private int Fiyat**

**StogaGelenYanUrunler**

**Private int ID**

**Private int Barkod**

**Private int UrunAdi**

**Private int EtkinMadde**

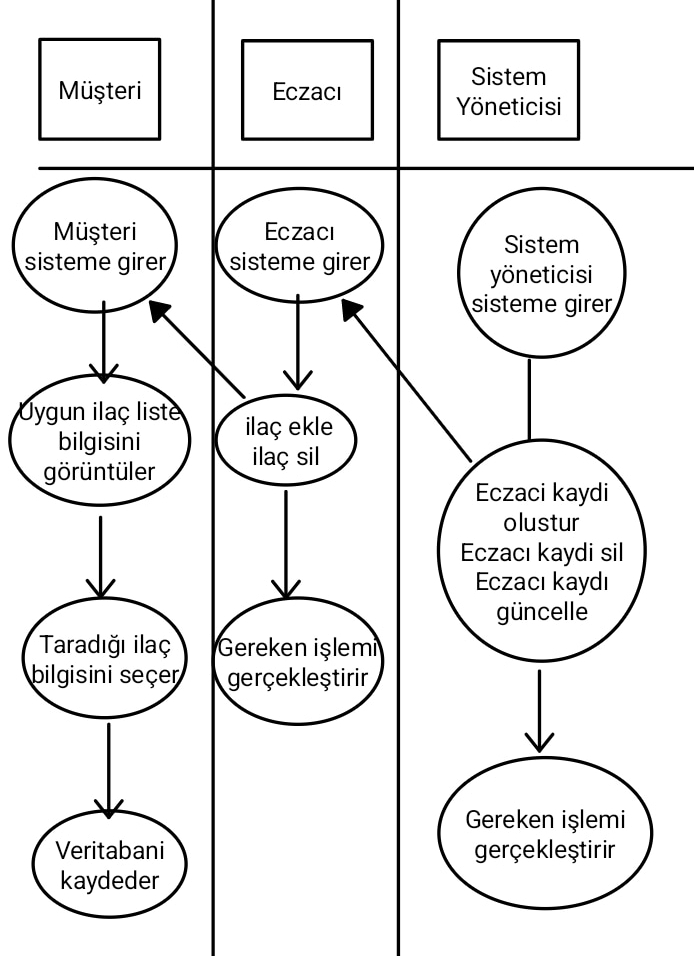
**Private int ATCKodu**

**Private int ReceteTuru**

**Private int KullanimYasi**

**Private int Fiyat**

**3.2.7 İşlevlerin Sıradüzeni**

****

**3.2.8 Başarım Gerekleri**

**Mevcut sistemler incelendi ve mevcut sistemin eksiklerinden yola çıkılarak, sistemin başarımı için;**

• Sistemin sonuç üretim doğrulukları

• Tepki sürelerinin en aza indirilmesi

• Mali külfetin azaltılması

• Hile hata ve yanlışlıkların en aza indirilmesi

• Kullanım kolaylılığı

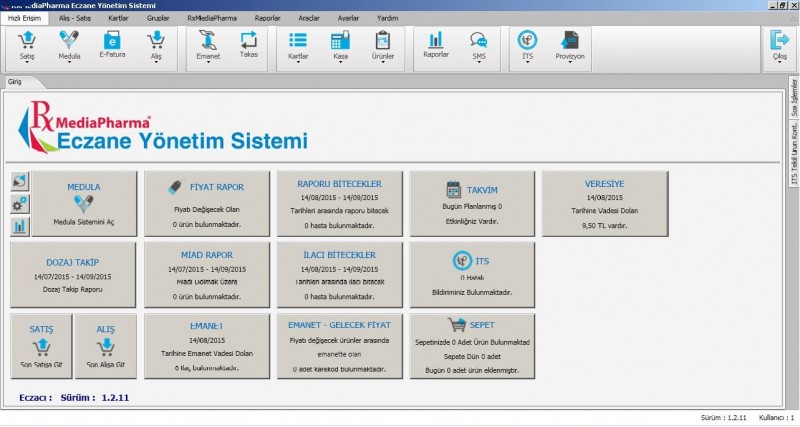
• Anlaşılabilirlik

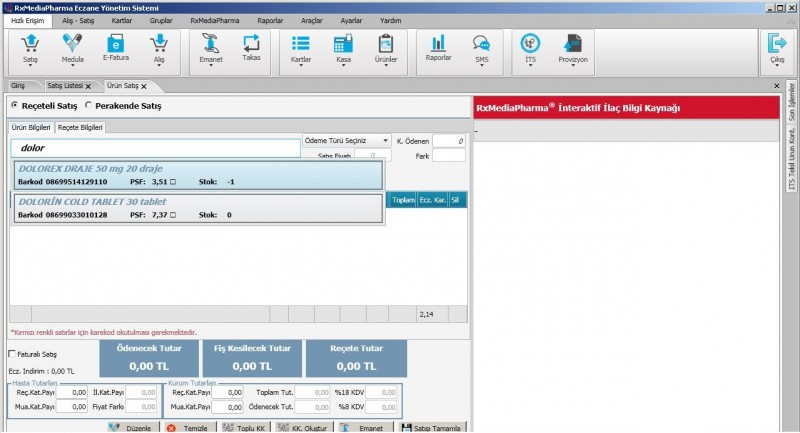
• Tarafsızlık

**Temel gereklilikler olarak tespit edilmiştir.**

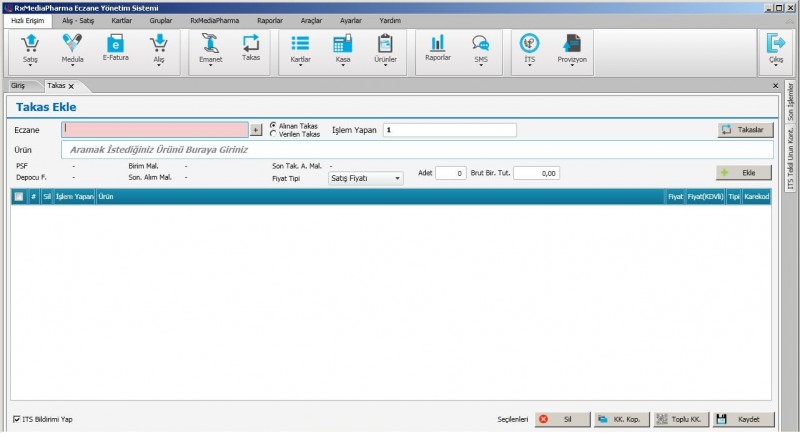
* 1. **Ara yüz (Modül) Gerekleri**

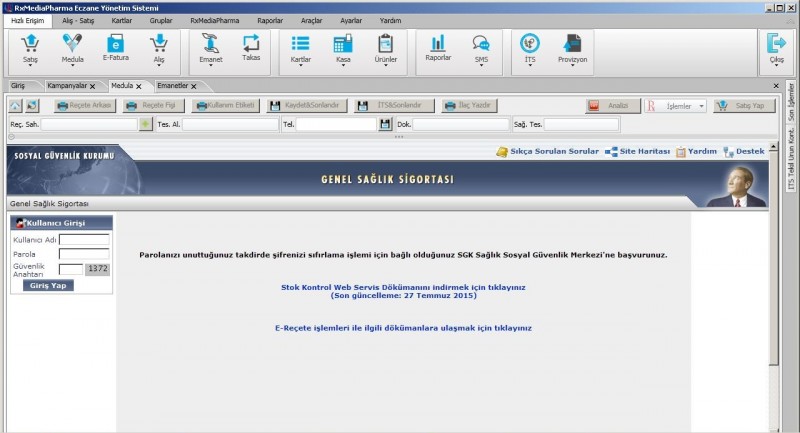
**Bazı Ara yüz Tasarımları**

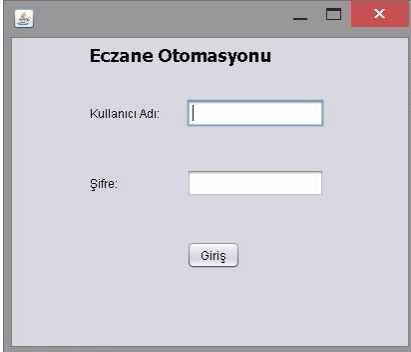
****

****



****

****

****

**3.3.1 Yazılım Arayüzü**

Projenin çalışması esnasında böyle bir açık verilmemesine özen gösterildi. Gerekli olan her türlü değişiklik seurce kodları üzerinden yapılıp tekrar derlenecek.

**3.3.2 Kullanıcı Arayüzü**

Projede kullanıcının arayüzü tasarlanırken herhangi bir şekilde renkler seçilerek tarafsız rahat büyük puntolu yazılı bir arayüz tasarlanacaktır.



**3.3.3 İletişim Arayüzü**

İletişim modülü proje kapsamında yapılmayacaktır.

**3.3.4 Yönetim Arayüzü**

Projenin neredeyse tamamı yönetimsel arayüzden oluştuğu için sadece yetkililerin göreceği ekran yönetimsel arayüzdür. Projenin %90 ı genelde yönetimsel ara yüzlerden oluşacak sadece personelin göreceği ekran kullanıcı ara yüzüne girecek dışındaki her yer yönetimsel ara yüz olacak. Yöneticinin ekrana girdiğinde karşılaşacağı ara yüzdür. İki ara yüz vardır: Biri Kullanıcı Ara yüzü diğeri ise yönetimsel ara yüzdür. Yani bu ara yüz sistem yöneticisi tarafından sisteme girişlerde gerçekleşecektir.

**3.4 Belgeleme Gerekleri**

**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.

**3.4.2 Eğitim Belgeleri**

Mevcut bir belgemiz bulunmamaktadır.

**3.4.3 Kullanıcı El Kitapları**

Proje sonunda rahat ve kolay kullanımdan dolayı bir eğitim semineri ve bir kullanım kitapçığı hazırlanacaktır ve ilaveten kullanım bilgileri web portalı üzerinden videolarla da yapılacaktır.

**4. SİSTEM TASARIMI**

**4.1 Genel Tasarım Bilgileri**

**4.1.1 Genel Sistem Tanımı**

**Gereksinimlerin Tanımlanması**

**Sistem ve Yazılım Tasarımı**

Şekil 4.1

**Gereksinimler**

Gereksinimler kısmında anket düzenlenecek ve ona göre bir tasarım oluşturmaya başlanacaktır. Bu süreçte gerekirse uzman kişiler projeye dahil edilip bu alandaki olumsuzlukların önüne geçilecektir. Sistemin tasarımını yapmak üzere belirlenen ekiple gerektiğinde toplantılar yapılarak sistemin tasarımı yapılacaktır.

**Tasarım**

**İşlevsel Belirtimler**

**Gereksinimler**

Şekil 4.2 Genel Sistem Tanımı

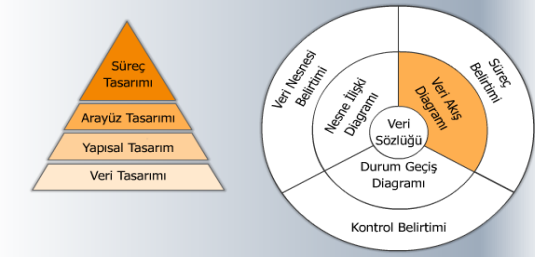
**İşlevsel Belirtimler**

Bu kısımda ilk önce sistemin ne yapacağı sorusuna cevap verelim. Sistem Türkiye deki eczacıların rahat bir şekilde hasatları ile ilgilenmeleri saglayacak. İkinci olarak yazılım ne yapacağı sorusuna cevap vermek gerekirse, yazılım ilk aşamada yönetici ile her fakülte arasındaki iletişimi sağlayacak. Bundan sonra kulancı giriş yaptıktan sonra eczacı ile iletişimini sağlayacak.

**Tasarım**

Tasarım aşamasında neler olacağını grafiksel olarak aktarmak isterim.

1. İlk önce bir Süreç tasarımı olacak ve adımlar aşağıdaki gibi olacak

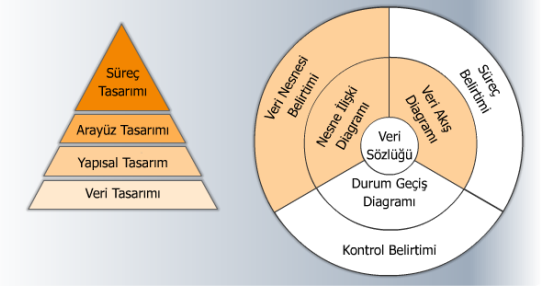




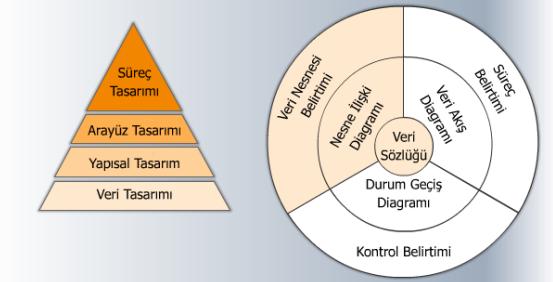
1. Süreç tasarımı bittikten sonra sıra Ara yüz tasarımına geldi ve ara yüz tasarımı aşağıda göründüğü adımlarla gerçekleştirildi.



1. Arayüz tasarımını aştıktan sonra sora yapısal tasarıma geldi. Yapısal tasarımda izlenen yollar aşağıdaki gibi oldu.

****

1. Yapısal tasarımı da yapıp Veri tasarımına geçildi ve şu adımlar izlendi.

****

Şekiller 4.3 Tasarım Aşamaları

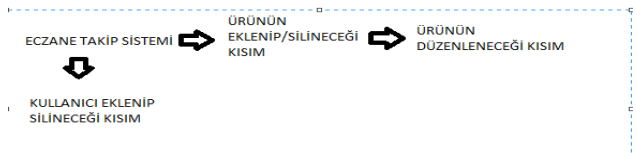
* + 1. **Varsayımlar ve Kısıtlamalar**

Sistemde varsayılan değerler bulunmamakta başlıca kısıtlamalar bulunmaktadır bunlar :

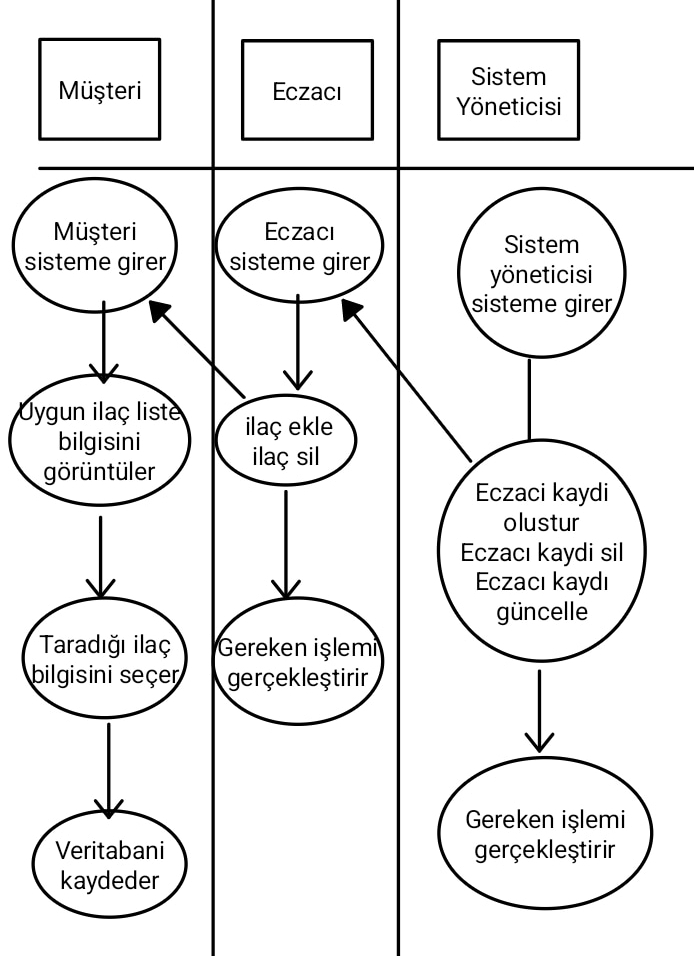
- Yönetim ara yüzüne sadece yetkili kişilerin erişebilirliği

- Otomasyona sistem yöneticileri hariç birisinin erişemeyeceği

**4.1.3 Sistem Mimarisi**

****

Şekil 4.4



**4.1.4 Dış Arabirimler**

**4.1.4.1 Kullanıcı Arabirimleri**

Kullanıcı arabirimlerin ilk başında sistem giriş ekranı bulunacak. Ve her birimin kendine ait ekranları olacak. Kullanıcı girişi bu arabirimde amaçlanmıştır. Kullanıcı, bilgilerini girerek bu sisteme giriş yapmış olacaktır.



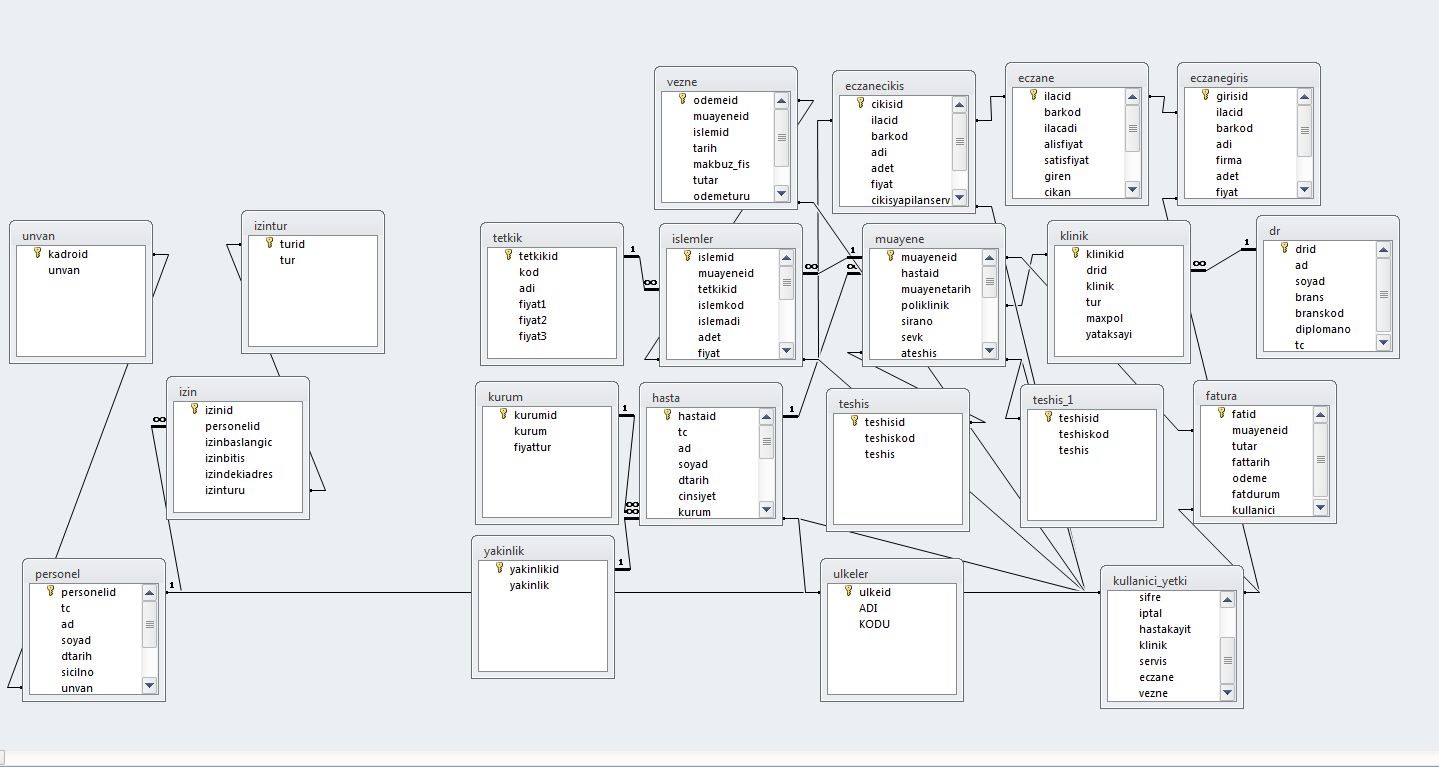
**4.1.4.2 Veri Arabirimleri**

Tasarımda veri tabanı kullanılacak olup bu veri tabanında ürün adı, ürünün grubu, giriş tarihi gibi bilgiler veri tabanından çekilecektir.

**4.1.4.3 Diğer Sistemlerle Arabirimler**

Diğer eczaneler ve hastane ile senkronizasyon için WAN bağlantısı ile bu web sunucularına bağlanılacaktır.

**4.1.5 Veri Modeli**



Şekil 4.5 Veri Modeli

**4.1.6 Testler**

Yazılım Ekibi Test Sorumlusu: Sistemin tasarımında ve birleştirilmesi ve teslimi süreçlerinde yazılımı geliştirme testine tabi tutacaktır. Bu geliştirme testinde projedeki diğer paydaşlarda görev alabilirler.

Birim Testi: Program birimleri ve nesne sınıfları test edilecektir.

Bileşen Testi: Birkaç ayrı birim bir araya gelerek birleştirme yapılacak ve program ara yüzleri test edilecektir.

Sistem Testi: Sistem bir bütün halinde test edilecektir. Sistem Kullanıcıları: Genel hatlarıyla testlerimiz iki aşamada gerçekleştirilecek. Bilinen adıyla pilot bölge uygulaması yapılacak.

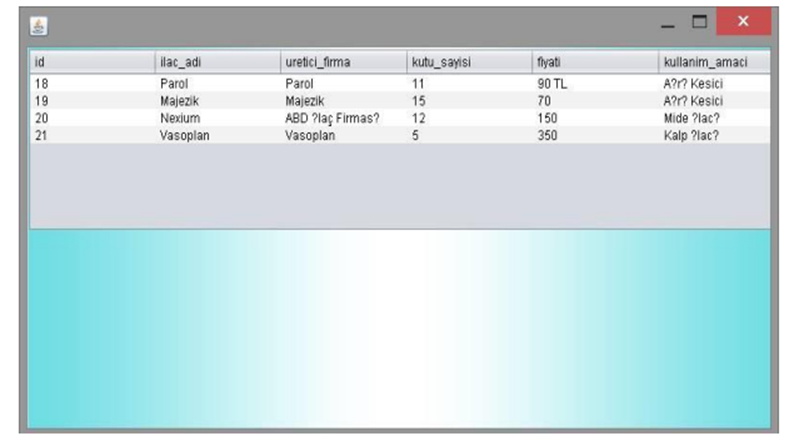
Alfa Aşaması: Sistemin geliştirildiği yerde kullanıcıların gelerek katkıda bulunması sistemi test etmesi ile yapılacak. Beta Aşaması: Kullanıcı, geliştirilen sistemi kendi yerleşkesinde, bir gözetmen eşliğinde yapılacak.

**4.1.7 Performans**

* Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecek Sistemin Tasarıma Uygunluk Performansı;
* Tasarımı yapılan sistemin stabilitesi ve işleyiş performansı değerlendirilecek. Veri Yapısının Sistemle Performansı;
* Veri yapısının sistemle stabilitesi ve çalışma zamanındaki uyumluluk düzeyindeki performansı değerlendirilecek.

**4.2 Veri Tasarımı**

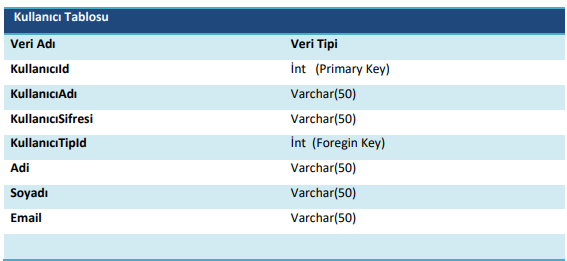
**Tablo Tasarım örnekleri**

****

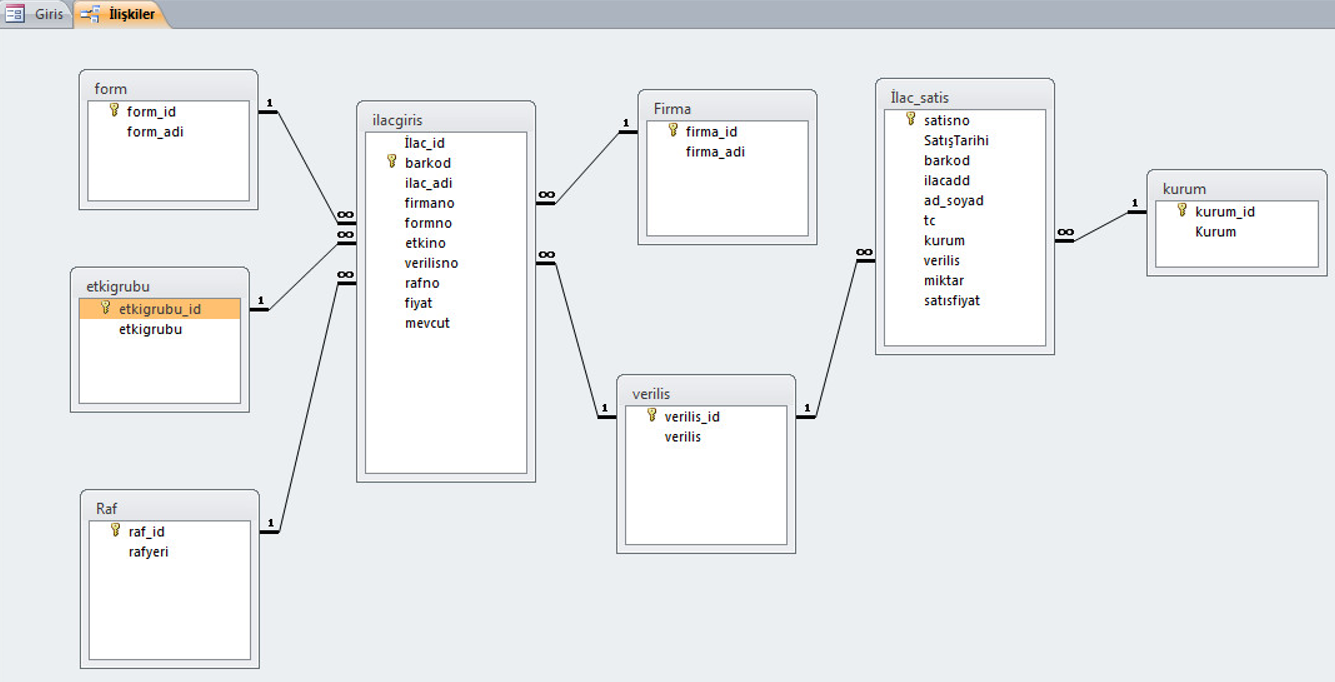
****

**4.2.1 Tablo tanımları**

Tablolar arası ilişki net bir şekilde ortaya konacaktır. Tablolar genişlemeye imkan sağlayacaktır.



* + 1. **Tablo- İlişki Şemaları**



**4.2.3 Veri Tanımları**

Veri tipi olarak integer(int) kullanılmasının amacı sayısal değerleri almaktan ötürüdür. Ayrıca benzersiz kod üretiminde id no olarak kullanılmaktadır. String olarak kullanılan değerler kelime içeren değerleri tutacağından ötürü kullanıldı. Double veriler hesaplamalarda kullanılan değerleri tutacaklardır. Date veri tipindeki veriler tarih ve saat bilgileri için kullanılacaktır.

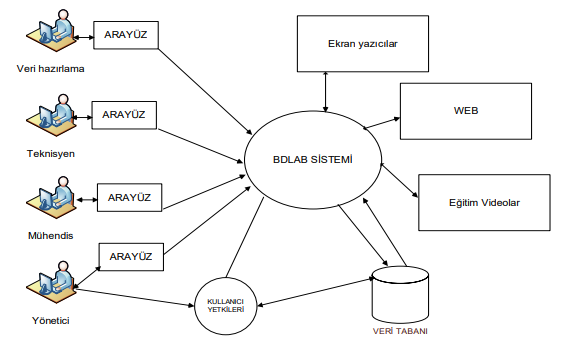
**4.2.4 Değer Kümesi Tanımları**

**Rapor**: Yapılan veri girişleri sonrası yapılan hesaplamalar sonrasında oluşacak bilgidir

**4.3 Süreç Tasarımı**

**4.3.1 Genel Tasarım**

Sistemde öncelikle nesne tasarımı oluşturularak nesneler arası ilişkiler tanımlandı. 3 bölüm halinde ele alınan deney raporları ayrı ayrı tasarlandı. Daha sonra veri tasarımı nesnelerin doğrultusunda oluşturularak tüm tablolar tanımlandı. Giriş ara yüzü ile ilgili daha önceki projelerde kullandığımız ve bu sistem için uyarlanan bir ara yüz kullanılacaktır. Her kullanıcı sisteme aynı ara yüzden bağlanacak ve sistem kullanıcıların yetkisini değerlendirerek ilgili ara yüzüne erişimi sağlayacaktır. Ancak kullanıcı rollerinin atanması ve yetkiler öncelikle yönetici ara yüzünde bulunacak kullanıcı tanımlamaları modüle ile başlangıçta yapılandırma zorunluluğu bulunmaktadır. Web ara yüzü sadece rapor bilgilerinin görüntülenmesi için tasarlanacaktır. Nesne-veri-kullanıcı tasarımından sonra tüm kullanıcıların kendi işlem yaptığı bölümler için ara yüz tasarımları yapılacaktır. Son olarak eğitim videoları ile ilgili her modüle bağlantılar eklenecektir

****

ECZANE

OTOMASYONU

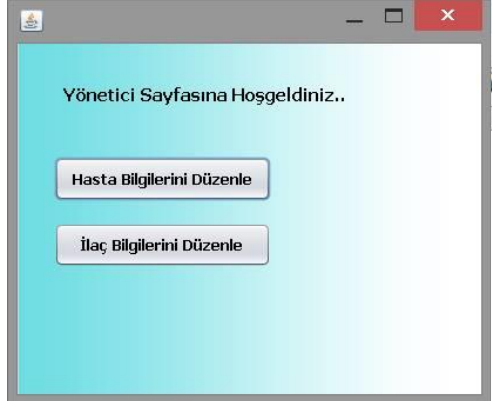
**4.3.2 Modüller**

**4.3.2.1 Yönetici Modülü**

**4.3.2.1.1 İşlev**

Sistem içindeki tüm arabirimler aslına bakacak olunursa yönetim modülüdür. Yönetim modülü içindeki tüm alanlar kullanıcı rolüne göre yetkilendirilmiştir.

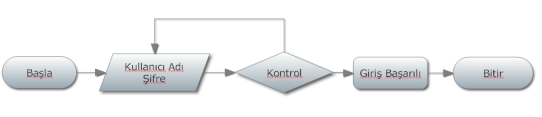
**4.3.2.1.2 Kullanıcı Arabirimi**

****

**4.3.2.1.3 Modül Tanımı**

Yöneticinin kontrolü altında diğer kullanıcılara verilen yetki tanımlamaları ile kullanıcı adı ve şifre ile sisteme giriş yapılacaktır.

**4.3.2.1.4 Modül iç Tasarımı**

****

Şekil 60 İç Tasarım

**4.3.2.2 Seçmen Modülü**

**4.3.2.2.1 İşlev**

Kullanıcının sisteme müdahale edebileceği ekrana erişmesi için aşması gereken bir modüldür

**4.3.2.2.2 Kullanıcı Arabirimi**

****

**4.3.2.2.3 Modül Tanımı**

Yöneticinin kontrolü altında diğer kullanıcılara verilen yetki tanımlamaları ile kullanıcı adı ve şifre ile sisteme giriş yapılacaktır.

**4.3.2.2.4 Modül iç Tasarımı**

kontrol

Kullanıcı adı şifre

**ARAYÜZ**

**Giriş Başarılı**

**4.3.3 Kullanıcı Profilleri**

**ADMİN**: Yapının en üstünde bulunan ve en yetkili yapılanmadır. Admin2’ye ve kullanıcıya yetki verir.

**KULLANICI:** Yapının üçüncü nesnesidir. Sisteme giriş yapıp ilaç taraması gibi birçok istemde bulunur.

**4.3.4 Entegrasyon ve Test Gereksinimleri**

Sistem kendi içinde bir bütündür. Kendi çevrimiyle her veri kontrol edilir.

**4.4 Ortak Alt Sistemlerin Tasarımı**

**4.4.1 Ortak Alt Sistemler**

Kullanılacak tek ortak sistem e- devlet mevcut sistemidir bu sistemle tc numarası bazında kişinin tüm bilgisi alınacaktır

**4.4.2 Modüller Arası Otak Veriler**

**4.4.3 Ortak Veriler İçin Veri Giriş ve Raporlama Modülleri**

Ortak veriler için böyle bir modül kullanılmadı.

**4.4.4 Güvenlik Alt Sistemleri**

Yazılım sistemlerinin güvenilirliğe ilişkin nicelikleri, kullanıcıların gereksinimlerini karşılayacak şekilde ortaya koymak ve güvenilirliğin hesaplanmasına yönelik verileri toplama, istatistiksel tahminleme, ölçütlerin tespiti, yazılıma ait mimari özelliklerin belirlenmesi, tasarım, geliştirme ve bunlara yönelik çalışma ortamının belirlenmesi ve modellenmesini kapsamaktadır. Sistem başlangıçta sıfır hata parolası ile tasarlanacaktır. Çünkü bu sistem ürünleri hata kabul etmez. Diğer taraftan programın tesliminden önce program nesne güvenliği, kullanıcı güvenliği, veri tabanı güvenliği ve WEB sunucu hizmetleri için güvenlik aşamalarından geçirilecek ve gerekli testler test ekibi tarafından yapılacaktır.

**4.4.5 Veri Dağıtım Alt Sistemleri**

Veri dağıtımından ziyade veri alma üzerine bir sistem kuruldu. Dağıtılan sistemlerde zararlı kişilerin müdahalesi kolay olmaktadır bunun yanı sıra bizim sistemimizde terminaller serverlardan veri çekecek serverlar terminallere veri göndermeyecek.

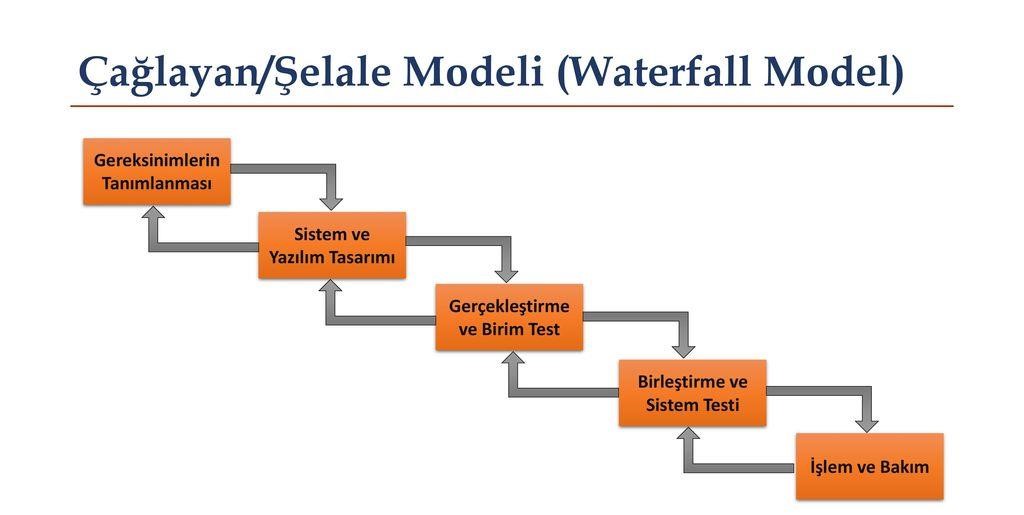
**4.4.6 Yedekleme ve Arşivleme İşlemleri**

Depolanan verilerin, herhangi bir nedenle zarar görmesi, sistemin çalışma süreçlerinde ciddi zararlar oluşturabilir. Yaşanabilecek bir felaket durumu sonrasında, depolanan verilerin geri yüklenememesi, sistemin sağlandığı kullanıcılara veya kurumlara çok ciddi zararlar verebilir. Bu nedenle sistemin çalışma süreçlerine bağlı olarak, yedekleme sistemleri kurulmalı ve yedekleme işlemleri günlük olarak takip edilmelidir. Yedekleme sistemlerinin kurulumu; yedeklenecek veri miktarı, yedekleme sıklığı, yedekleen verinin zaman içerisinde değişme oranı ve maksimum veri kaybı gibi parametrelere bağlıdır. Sistemin birden fazla sunucusunun eş zamanlı yedekleme işlemini yapabilmesi, işletim sistemlerinin kayıt dosyalarını tam ve eş zamanlı olarak yedekleyebilmesi ve işletim sistemleri üzerinde çalışan veri tabanı uygulamasının yedeklerini sistem kapatılmadan alabilmesi gerekmektedir.

**5. SİSTEM GERÇEKLEŞTİRİMİ**

**5.1. Giriş**

Sistemin gerçekleştirimi için fiziksel model oluşturulmuş ama bu modelin yapımı müşteriye aittir. Gerçekleştirme aşamasında mevcut fiziksel modele yazılım entegre edilecektir. Entegrasyon için kurulum eğitimi verilecektir. Yazılımın gerçekleştirilmesi önceki aşamaların tamamlanmasına ve gerekli testlerin yapılmasına bağlıdır.



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.

**FİZİKSEL MODEL**

**GERÇEKLEŞTİRİM**

**VERİTABANI**

**YAZILIM KAYNAK KODLARI**

Sistem Gerçekleştirim Modeli

**5.2. Yazılım Geliştirme Ortamları**

Yazılım geliştirme ortamı, tasarım sonunda üretilen fiziksel modelin, bilgisayar ortamında çalıştırılabilmesi için gerekli olan:

• Programlama Dili

• Veri Tabanı Yönetim Sistemi

• Hazır Program Kitapçıkları

**5.2.1 Programlama Dilleri**

Projemizde kullanılacak olan programlama dillerini belirtmiştik. Kendi içlerinde sınıflandırılan bu diller arasından bizim için en uygun olanı nesneye yönelik programlama dili, olacaktır.. Yine web tasarımı için Java script, PHP,HTML kullanılacaktır. Veri tabanı için MYSQL kullanılması sistem için uygun görülmüştür.

**5.2.2 Veri Tabanı Yönetim Sistemleri**

WEB uygulamasına PHP ile veriye erişecektir. Veri tabanı ise Veri tabanı yönetim sistemi ile kontrol edilecektir.



Veritabanı Yönetim Sistemleri Görevleri:

* Verileri tutmak için veri tabanı oluşturmak
* Varolan veritabanında değişiklik yapmak
* Veritabanının bakımını ve onarımını yapmak
* Verilerin tutulduğu tablolar arası ilişki kurmak
* Kullanıcı yetkilerini belirlemek için oluşturulmuş bir sistemdir.,

**5.2.2.1 VTYS Kullanımının Ek Yararları**

* Veri tekrarını engeller (Varolan tabloda ilişki kur 2 tablo birleştir aynı bilgileri içeren bir daha tabloda oluşturulmaz)
* Veri tutarlılığı (Tablo üzerinde yapılan bir değişiklik tanımlı ilişkiler sayesinde diğer tabloları da etkiler)
* Veri paylaşımı(Farklı kullanıcılar aynı ya da farklı zamanlarda Veritabanına erişim sağlayabilirler.)
* Veri bütünlüğü (Farklı tablodaki ilişkili kayıtlar aynı anda silinir veya güncellenir)
* Veri Güvenliği (Kullanıcıya yazma değil okuma yetkisi vermek vb.)
* Veri bağımsızlığı (Veritabanının fiziksel yapısı kullanıcılardan gizlenir. Arka planda VTYS kullanıcıya ne yaptığını göstermeden kullanıcılar için çeşitli terimlerle veritabanı üzerinde işlem yapılmasını sağlar

**5.2.2.2 Veri Modelleri**

Veri modeli, verileri mantıksal düzeyde düzenlemek için kullanılan yapılar, kavramlar ve işlemler topluluğudur. Her VTYS belirli bir veri modeli kullanır. Veri tabanını tasarlayan kişi, veri modelinin yapılarını ve kavramlarını kullanarak mantıksal düzeydeki düzenlemeleri oluşturur. Daha sonra o veri modelini kullanan bir VTYS üzerinde bu düzenlemelere göre veri tabanı yaratılır. Günümüzde en çok kullanılan veri modeli “ilişkisel veri modeli” dir.

Her VTYS belirli bir veri modeli kullanır. Günümüzde en çok;

**Hiyerarşik VTYS:** Veriler alt-üst ilişkisi şeklinde tutulur. 1960 – 1970’li yıllarda çok kullanılmıştır.

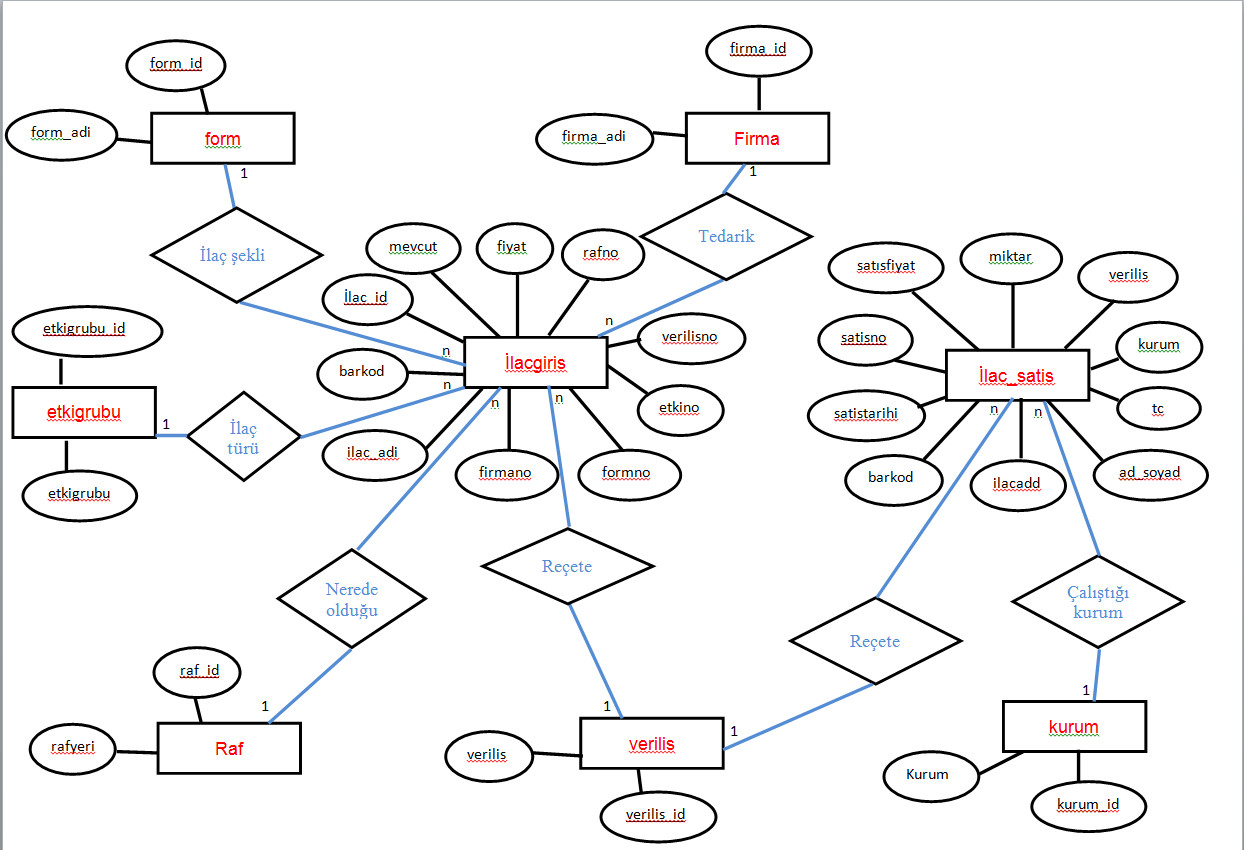
**Ağ VTYS:** Veriler birbirlerine yine alt-üst ilişkisi ile bağlıdır ama her verinin birden fazla üstü bulunabilir. 1970’li yıllarda ve 1980’li yılların ilk yarısında kullanılmıştır.

**İlişkisel VTYS:** Günümüzde en çok kullanılan VTYS türüdür. Veriler tablolar şeklinde gruplanır ve her tablo başka tablolar ile mantıksal olarak ilişkilendirilir. ilk kez 1969 yılında ortaya atılmış, 1970’li yılların sonunda kullanılmaya başlanmış ve 1985 yılından sonra yaygınlaşmış bir yaklaşımdır.

**Nesne Tabanlı VTYS:** Veriler nesneler halinde tutulur. Son on yılı aşkın süredir gündemde olan, günümüzde çok yaygın kullanılmasa bile, kullanımı giderek yaygınlaşan bir yaklaşımdır. Bu proje kapsamında ilişkisel VTYS kullanılarak nesneler tablolar halinde tutulacak ve tablolar arası tanımlanan ilişkiler ile veri aktarımı sağlanacaktır.

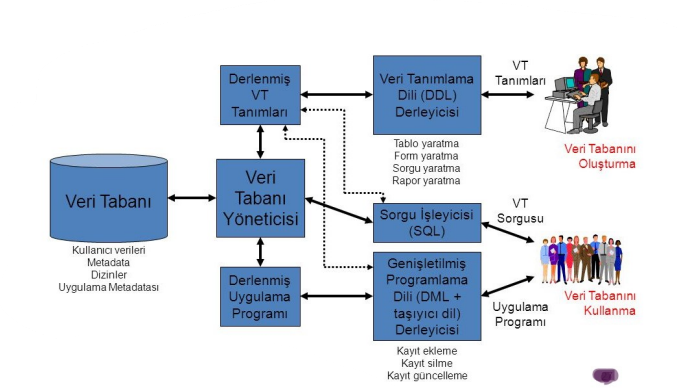
**VERİ MODELİ**

**5.2.2.3 Şemalar**

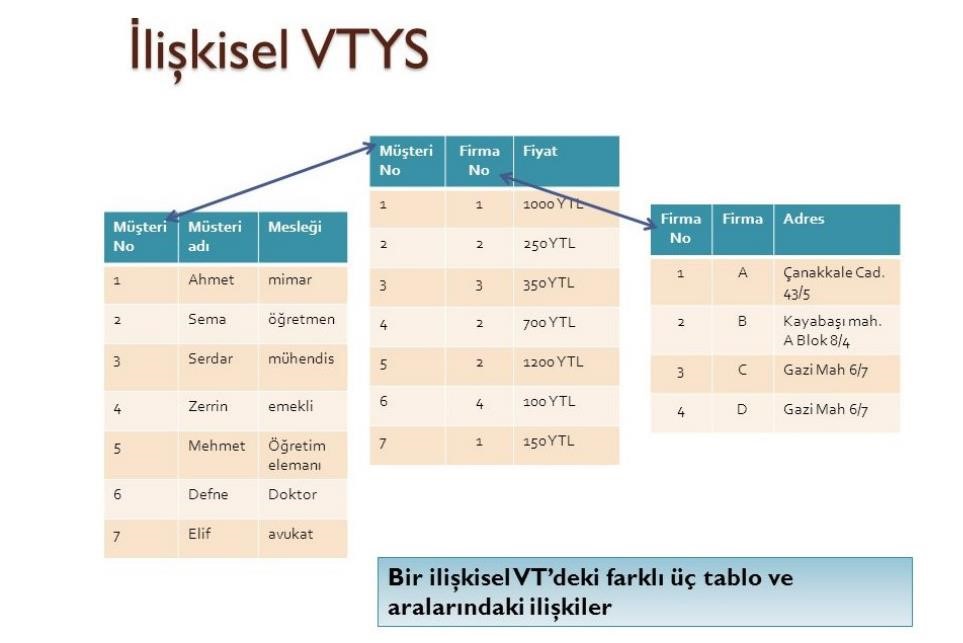
****

Varlık bağıntı şeması

**5.2.2.4 VTYS Mimarisi**

****

VTYS Mimarisi



**İlişkisel Veri Tabanı**

**5.2.2.5 Veritabanı Dilleri ve Arabirimleri**

Sistemimizde veritabanı dili olarak MYSQL kullanıldı. Henüz prototip aşamasında olan sistemimiz için MYSQL arabirimini yeterli gördük.

|  |  |
| --- | --- |
| **Veri Tanımlama Dili (VTD)** | Kavramsal şemaları tanımlamak üzere veri tabanı yönetici ve tasarımcısı tarafından kullanılır. |
| **Saklama Tanımlama Dili (STD)** | İçsel şemayı tanımlamak için kullanılır. |
| **Görüş Tanımlama Dili (GTD)** | Görüş tanımlama dili kullanıcı görüşlerini tanımlamak ve kavramsal şemaya dönüştürmek amacıyla kullanılır. |
| **Veri İşleme Dili (VİD)** | Veri işleme dilidir. Veri tabanı oluşturduktan sonra veri tabanına veri eklemek, değiştirmek, silmek veya eklenmiş veriyi getirmek amacıyla kullanılır. |

**5.2.2.6 Veri Tabanı Sistem Ortamı**

Veri tabanı sistem ortamında php ,MYSQL Workbeanch, kullanılacak. Tüm Yükleme, yedekleme , sıralama, veri sıkıştırma, ve benzeri fonksiyonları yerine getirmek amacıyla bu ortamı kullandık.

**5.2.2.7 VTYS'nin Sınıflandırılması**

En fazla kullanılan veri modelleri ilişkisel, ağ, hiyerarşik, nesne-yönelimli ve kavramsal modellerdir. Bizim Kullandığımız ise ilişkisel veri modelidir

**5.2.2.8 Hazır Program Kütüphane Dosyaları**

Java dilinin bizlere sunduğu oldukça geniş bir kütüphane var bundan yaralandık.

**5.2.2.9 CASE Araç ve Ortamları**

Case araçları olarak ise Microsoft un Visio ve Publisher ürünlerini kullandık.

**5.3. Kodlama Stili**

Daha önceki projelerde kullandığımız bir çok kodlama sitillerinden yararlandık .

**5.3.1 Açıklama Satırları**

Açıklama satırları her satırın devamında yapıldı. Bu şekilde sistemdeki diğer bölümlerle iletişim sorunu ortadan kalktı.

**5.3.2 Kod Biçimlemesi**

Geleneksel java tekniklerine bağlı kalınarak kodlama yapıldı.

**5.3.3 Anlamlı İsimlendirme**

Sistem kodlamasının genel yapısında kullanılan değişkenleri ve nesneleri aynı şekilde veri tabanındada kullandık. Karmaşıklığın önüne geçtik.

**5.3.4 Yapısal Programlama Yapıları**

Genel olarak 3 başlıkta incelenir:

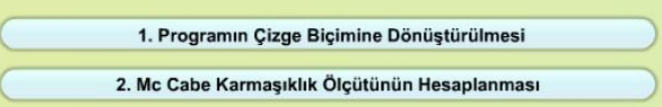
ϖ Ardışık işlem yapıları: Bu yapılarda genellikle fonksiyon , altprogram ve benzeri tekrarlayan yapılar tek seferde çözüldü

ϖ Koşullu işlem yapıları: Bu yapılarda hemen hemen programın tamamında karşılaştırma yapılan her kod da kullanıldı.

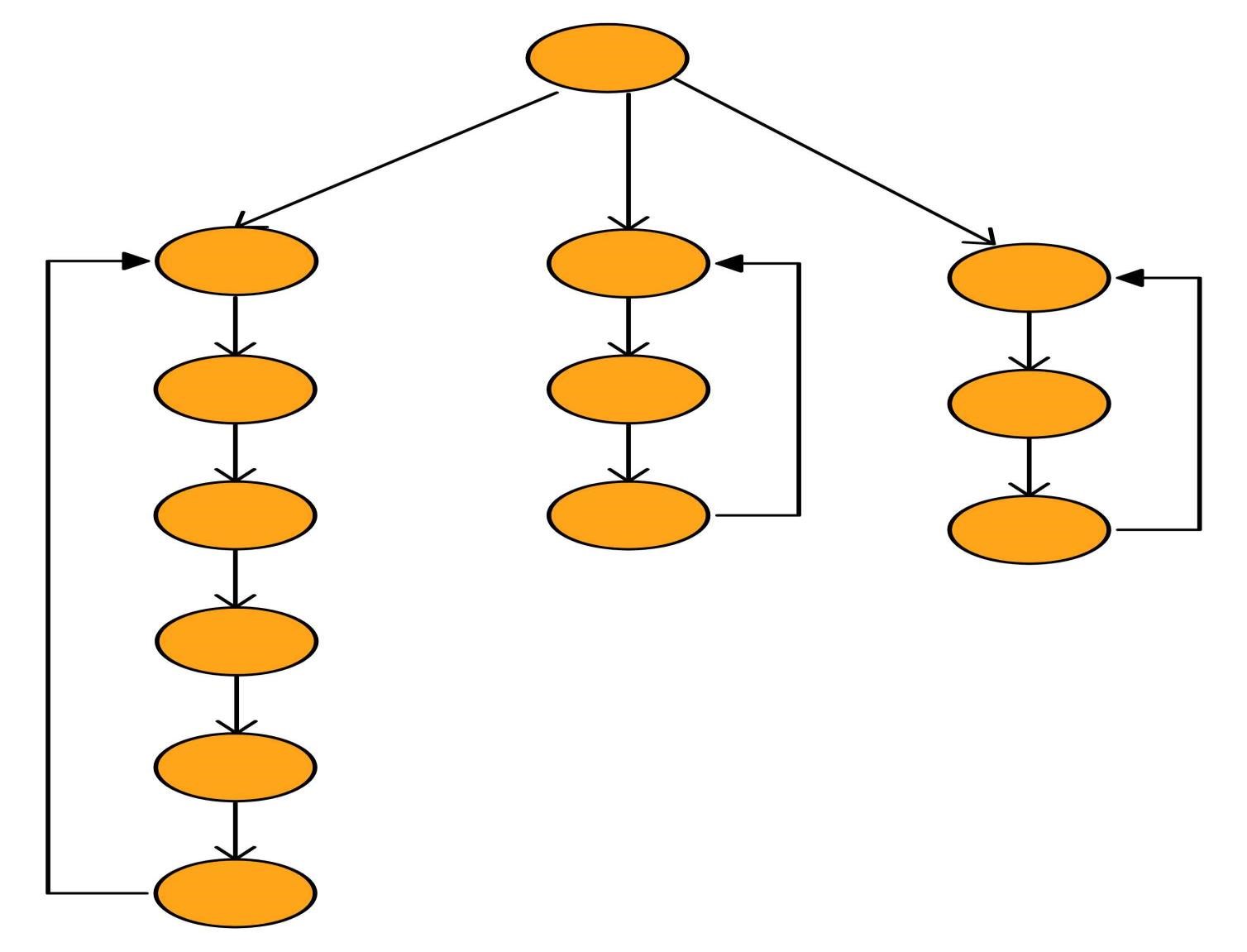
ϖ Döngü yapıları: Bu yapılarda ardışık yapılar gibi arka arkaya yapılacak işlemleri tek bir döngü içerisinde yapıp işlemler kolaylaştırıldı.

**5.4. Program Karmaşıklığı**

Program karmaşıklığının ölçülmesi için bir çok teorik model vardır. Bu modeller içerisinde en eski olanı McCabe karmaşıklık ölçütüdür. Bu ölçüt 1976 yılında McCabe tarafından geliştirilmiş olup bu konuda geliştirilen diğer ölçütlerin çoğu bu ölçütten etkilenmiştir. McCabe ölçütü, programda kullanılan “koşul” deyimlerinin program karmaşıklığını etkileyen en önemli unsur olduğu esasına dayanır ve aşağıda görüldüğü gibi iki aşamada uygulanmaktadır:

****

**5.4.1 Programın Çizge Biçimine Dönüştürülmesi**



**5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama**

K=13 (Kenar sayısı)

D=9 (Düğüm sayısı)

P=1 (Bileşen sayısı)

V(G)=k-d+2p

**5.5. Olağan Dışı Durum Çözümleme**

Olağan dışı durum, bir programın çalışmasının geçersiz, yanlış veri oluşmasından veya başka sebeplerden dolayı istenmeyen bir şekilde sonlanmasına neden olan durumlardır.

**5.5.1 Olağandışı Durum Tanımları**

Olağandışı (beklenmedik bir şekilde) gelişen durumlarda try-catch kod blokları devreye girip program sonlanmadan çalışmaya devam edebilmesi için tasarlanmıştır.

**5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları**

Tüm olağandışı durumlarda program sonlanmadan hata mesajı verilerek programın başına döndürülmek üzere tasarlanmıştır.

**5.6. Kod Gözden Geçirme**

Bir programcı önceki sürümleri incelemeden okunabilir bir program yapamaz. Kod gözden geçirme ile program sınama işlemleri birbirinden farklı durumlardır. Program sınama, programın işletimi sırasında ortaya çıkabilecek yanlış ya da hataları yakalamak için yapılan işlemdir. Kod gözden geçirme ise, programın kaynak kodu üzerinde yapılan inceleme işlemidir. kod gözden geçirme işlemlerinde programın hatalarının %3 ila %5 i yakalanabilmektedir. Programcı programı yazarken programın hemen sonra kod inceleme sürecine gireceğini bilirse program yazdığında daha az hatalı ve okunabilir programlar elde edilebilir

**5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi**

Gözden geçirme sürecinin Temel özellikleri;

• Hataların bulunup düzeltilmesi hedeflenmiştir.

• Olabildiğince az kişi ile yapılmalıdır. En iyi durumda deneyimli bir inceleyici kullanılmasıdır.

• Kalite çalışmalarının bir parçası olarak ele alınmalı ve sonuçlar düzenli ve belirlenen bir biçimde saklanmalıdır.

**5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular**

Bir program incelenirken, programın her bir öbeğinde aşağıdaki soruların cevapları aranmaktadır.

**5.6.2.1 Öbek Arayüzü**

Oluşturduğumuz öbekleri test etmek için yukarıda belirtilen sorular :

• Her öbek tek bir işlevsel amacı yerine getiriyor mu?

• Öbek adı, işlevini açıklayacak biçimde anlamlı mı?

• Öbek tek giriş çıkışlı mı?

• Öbek eğer bi irşlev ise parametrelerin 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 tanıtıyor 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**

* Veriler işlevselliğine göre gruplandırıldı mı?
* Değişken adları taşıdığı verinin özelliklerini anımsatıyor mu?
* Farklı amaçlarla kullanılan değişkenler var mı?
* Aynı işlevi yerine getiren değişkenler mükerrer olarak tanımlanmış mı?
* Verilerde nesneler kullanılmış ise veri soyutlama yapılmış mı?
* Nesneler için kapsülleme 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 bu sorular referans alınmaktadır.

**5.6.2.5 Sunuş**

Son kısımda ise şu sorular bulunmakta:

• Her satır, en fazla bir deyim içeriyor mu?

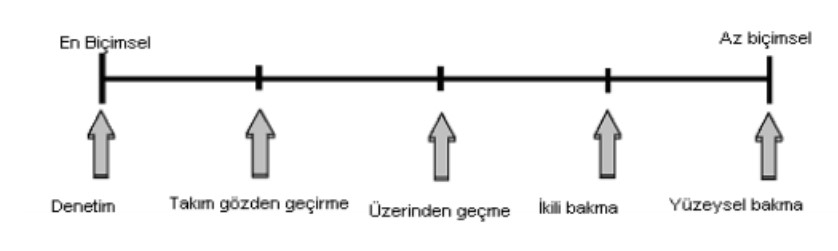
• Bir deyimin birden fazla satıra taşması durumunda, bölünme anlaşılabilirliği kolaylaştıracak biçimde anlamlı mı?

• Koşullu deyimlerde kullanılan mantıksal işlemler yalın halde mi?

• 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?



**Kod Gözden Geçirme**

Denetim şeklinde kod gözden geçirme daha güvenilir sonuç verecektir.

**6. DOĞRULAMA VE GEÇERLEME**

**6.1. Giriş**

Geliştirilecek bilgi sistemi yazılımının doğrulanması ve geçerlemesi, üretim süreci boyunca süren etkinliklerden oluşur. Söz konusu etkinlikler:

• Yazılım belirtimlerinin ve proje yaşam sürecindeki her bir etkinlik sonunda alınan çıktıların, tamam, doğru, açık ve önceki belirtimleri tutarlı olarak betimler durumda olduğunun doğrulanması.

• Proje süresince her bir etkinlik ürününün teknik yeterliliğinin değerlendirilmesi ve uygun çözüm elde edilene kadar aktivitenin tekrarına sebep olması.

• Projenin bir aşaması süresince geliştirilen anahtar belirtimlerin önceki belirtimlerle karşılaştırılması. Yazılım ürünlerinin tüm uygulanabilir gerekleri sağladığının gerçeklenmesi için sınamaların hazırlanıp yürütülmesi biçiminde özetlenebilir

**6.2. Sınama Kavramları**

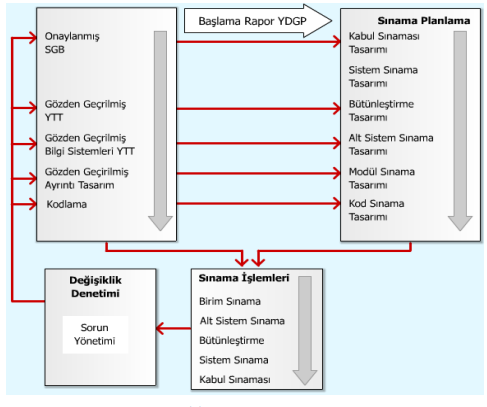
**Birim Sınama:** Sistemin birimleri kendi aralarınsa sistemi sınadılar.

**Alt Sistem Sınama**: Birimlerin birleşmesiyle modüller oluşturulup bunların kendi içinde sınaması yapıldı. Genel olarak arayüzde ki eksiklikler giderildi.

**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 onaylandı.

**6.3. Doğrulama ve Geçerleme Yaşam Döngüsü**

****

Yaşam Döngüsü

**6.4. Sınama Yöntemleri**

Sınama işlemi, geliştirmeyi izleyen bir düzeltme görevi olmak ile sınırlı değildir. Bir "sonra" operasyonu olmaktan çok, geliştirme öncesinde planlanan ve tasarımı yapılması gereken bir çaba türüdür.

**6.4.1 Beyaz Kutu Sınaması**

Beyaz kutu testleri birim, tümleştirme ve sistem test seviyelerinde gerçekleştirilebilir. Birim test seviyesinde gerçekleştirilen beyaz kutu testleri birim tümleştirme öncesinde birimdeki hataları bulmayı amaçlar. Tümleştirme seviyesindeki beyaz kutu testleri ise modüllerin birbiri ile iletişiminde ortaya çıkabilecek olan hataları bulmak hedeflenir. Sistem seviyesinde gerçekleştirilen beyaz kutu testlerinde ise amaç kapsama analizlerinin gerçekleştirilmesidir.

**Sınama arasında:**

* Tüm bağımsız yollarda en az bir kere sınama işleminin yapılması
* Tüm mantıksal karar noktalarında iki farklı karar için sınama yapılması
* Tüm döngülerin sınır değerlerinde sınama yapılması,
* İç veri yapılarının denenmesi

Test ekipleri tarafından en çok kullanılan teknik olan kara kutu test tekniği adından da anlaşılacağı gibi uygulamanın sadece derlenmiş kodu üzerinden test edilmesi olarak bilinir. Bu test tekniğinde, yazılımın programatik yapısı, tasarımı veya kodlama tekniği hakkında herhangi bir bilgi olması gerekli değildir. Yazılımın gereksinimine duyulan şeylere yanıt verip veremediği ve işlevselliği sınanmaktadır. Daha önce sonuçları hesaplanmış bir deney ve raporundaki bilgiler girilerek program çalıştırıldı ve eldeki somut veriler ile program verileri karşılaştırıldı.

**Kara Kutu**

Test ekipleri tarafından en çok kullanılan teknik olan kara kutu test tekniği adından da anlaşılacağı gibi uygulamanın sadece derlenmiş kodu üzerinden test edilmesi olarak bilinir. Bu test tekniğinde, yazılımın programatik yapısı, tasarımı veya kodlama tekniği hakkında herhangi bir bilgi olması gerekli değildir. Yazılımın gereksinimine duyulan şeylere yanıt verip veremediği ve işlevselliği sınanmaktadır. Daha önce sonuçları hesaplanmış bir deney ve raporundaki bilgiler girilerek program çalıştırıldı ve eldeki somut veriler ile program verileri karşılaştırıldı.

**6.4.2 Temel Yollar Sınaması**

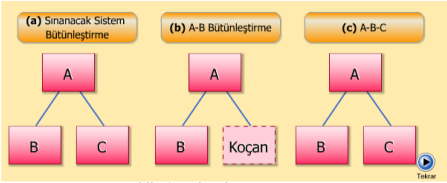
Sistem şartlarının gerekleri gözden geçirilerek yapılır. Kullanım profili, uçdeğer analizi ve rasgele test şeklinde aşamalardan sonra sistemin sınaması bitirilir.

**6.5.Sınama ve Bütünleştirme Stratejileri**

Genellikle sınama stratejisi, bütünleştirme stratejisi ile birlikte değerlendirilir. Ancak bazı sınama stratejileri bütünleştirme dışındaki tasaları hedefleyebilir. Örneğin, yukarıdan aşağı ve aşağıdan yukarı stratejileri bütünleştirme yöntemine bağımlıdır. Ancak işlem yolu ve gerilim sınamaları, sistemin olaylar karşısında değişik işlem sıralandırmaları sonucunda ulaşacağı sonuçların doğruluğunu ve normal şartların üstünde zorlandığında dayanıklılık sınırını ortaya çıkarır.

**6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme**

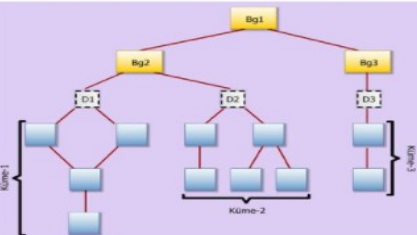
* Yukarıdan aşağı bütünleştirmede , önce sistemin en üst düzeylerinin sınaması ve sonra aşağıya doğru olan düzeyleri, ilgili modüllerin takılarak sınanmaları söz konusudur.
* En üst noktadaki bileşen, bir birim/modül/alt sistem olarak sınandıktan sonra alt alt düzeye geçilir. Ancak bu en üstteki bileşenin tam olarak sınanması için alttaki bileşenlerle olan bağlantılarının da çalışması sağlanmalıdır. Alt bileşenler henüz bu işlem için hazır olmazlar
* Bunların yerine üst bileşenin sınanması için kullanılmak üzere “koçan” programı oluşturulmalıdır. Koçanları bir alt bileşenin, üst bileşen ile ara yüzünü temin eden, fakat işlevsel olarak hiçbir şey yapmayan, boş çerçeve programlardır
* Üst bileşenlerin sınanması bittikten sonra bu koçanlar, içeri doldurularak kendi kodlama ve birim sınama işlemlerini tamamladıktan sonra üst bileşen ile yeniden sınanırlar



## **Yukarıdan Aşağı Sınama**

**6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme**

Aşağıdan yukarı bütünleştirmede ise, önceki yöntemin tersine uygulama yapılır. Önce en alt düzeydeki işçi birimler sınanır ve bu bir üstteki birim ile sınanması gerektiğinde bu üst bileşen, bir ‘sürücü’ ile temsil edilir. Yine amaç, çalışmasa bile arayüz oluşturacak ve alt bileşenin sınanmasını sağlayacak bir birim edinmektir. Bu kez kodlama, bütünleştirme ve sınama aşağı düzeylerden yukarı yüzeylere doğru gelişir ve yukarı düzeylerde önce sürücü olarak yazılan birimler sonra gerçekleriyle yer değiştirerek o düzeyin birimleri /alt sistemleri olurlar.



Aşağıdan Yukarı Sınama

**Şekilde**;

**1**-Belirli bir yazılım alt işlevini gören alt düzey birimler kümeler biçiminde oluşturulurlar

**2**-Denetim amaçlı bir sürücü programı sınama işlemi için girdi ve çıktı oluşturmak amacıyla yazılır Sürücüler aşağıdan yukarı kaldırılır ve gerçek birim ya da birim kümeleriyle değiştirilerek sınama işlemi sürdürülür.

**6.6. Sınama Planlaması**

Bir tablo ile özetlemek gerekirse şu şekilde özetleyebiliriz.

🡪Test raporu hazırlanırken şu özellikler mutlaka planda belirtilmelidir;

* Test planı kimliği: Test planının adı veya belge numarası
* Giriş: Test edilecek yazılımın elemanlarının genel tanıtım özetleri. Ayrıca bu plan kapsamı ve başvurulan belgeler. Kısaltmalar ve terim açıklamaları bu bölümde bildirilmelidir.
* Test edilecek sistem: Sistemde bileşenleri sürüm sayıları olarak sıralar ve sistemin özelliklerini bileşenlerini ve nasıl kullanıldıkları açıklanmalıdır. Ayrıca sistemde test edilmeyecek parçalar belirtilmelidir.
* Test edilecek ana fonksiyonlar: Sistemin test edilecek ana fonksiyonlarının kısa bir tanıtımı yapılmalıdır.
* Test edilmeyecek ana fonksiyonlar: Sistemde test edilmeyecek fonksiyonları ve bunların neden test edilmedikleri açıklanacaktır.
* Geçti/Kaldı Kriterleri: Bir test sonucunda sistemin geçmiş veya kalmış sayılacağını açıklanmalıdır.
* Test dokümanı: Test süresince yapılan işlemleri alınan raporları elde edilen bilgileri rapor içinde sunulmalıdır.
* Sorumluluklar: Hangi kişilerin nelerden sorumlu olduğu ve test takım lideri bilgileri mutlaka raporda belirtilmelidir.
* Riskler ve Önlemler: Test planında varsayılan ve olası yüksek riskli durumları belirtir ve bu durumların olması durumunda, etkilerinin en aza indirilebilmesi için alınması gereken önlemleri açıklar.

**6.7. Sınama Belirtimleri**

Her sınama planı, sınama etkinliklerinin sınırlarını , yaklaşımını, kaynaklarını ve zamanlamasını tutar.Plan neyin sınanacağını neyin sınanmayacağını , sorumlu kişileri ve riskleri açıkça gözler önüne serer.İşte bu sınama planları sınama belirtimlerini içerir. Bu belirtimler bir rapora dönüştürülerek taraflarca imza altına alınır. Bu sınamalardan sorumlu kişi ekip içinde test görevini alan kişidir. Sınamalarda ortaya çıkan hatalar ve açıklamalar bu raporda gösterilir.

**7. BAKIM**

**7.1 Giriş**

Sistemin tasarımı bittikten sonra artık seçimden seçime sistemin bakıma sokulması gerekir daha öncede belirttiğimiz gibi sistem hassas ve hata kabul etmeyecek bir sistemden bahsediyoruz. Bakım için fazla bir personel planlaması yapılmamıştır. Çünkü program %97 başarı ile teslimi hedeflenmiştir. Buradan hareketle pilot olarak seçilen kullanıcıların sistemi testinden sonra yazılım güvenilir konuma gelecek ve kullanım sürecinde çok köklü değişiklikler içermesi beklenmemektedir. Bu nedenle bakım aşaması uzaktan destek ekibi ile yapılacaktır.

**7.2 Kurulum**

Web tarayıcılardan erişilebilir.

**7.3 Yerinde Destek Organizasyonu**

Yerinde destek ekibi planlanmamıştır. Ancak destek gerektiğinde uzaktan bağlantı veya yerinde destek verilecektir. Her destek ayrıca ücrete tabidir.

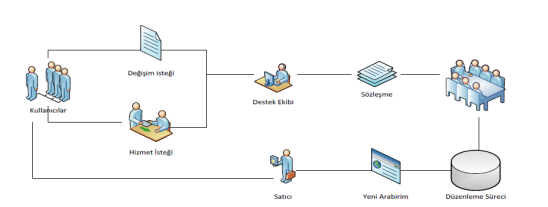
**7.4 Yazılım Bakımı**

Yazılım bakımı sistem teslim edildikten sonra sistemin değişme süreci ile ilgilidir . Yazılımda hata veya gereksinim değişiklikleri için düzeltmeler yapılabilir. Gerektiğinde sisteme yeni bileşenler eklenebilir. Bu kapsamda üç farklı bakım türünden bahsedilebilir.

1-Hataları ve zayıf noktaları düzeltmek için yapılan düzeltmeler.

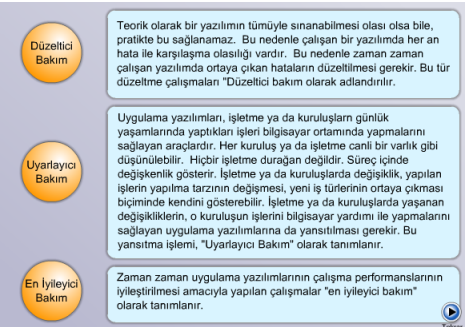
2-Yazılımları yeni platformlara ve ortamlara uyarlamak için yapılan uyarlamalar

3-Yeni özellik veya yeni gereksinimleri karşılamak üzere yapılan düzeltmeler

****

Şekil 77 Yazılım Bakımı Süreci

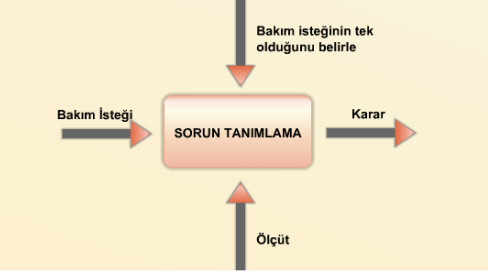
**7.4.1 Tanım**

****

**7.4.2 Bakım Süreç Modeli**

Bakım süreç modeli yukardaki işlemlerin hepsinin baştan yapılmasıdır. Bu işlemleri 7 adımda gösterelim:

1. **Adım: Sorunu Tanımlama Süreci İlk önce bakım ne için yapılıyor sorun ne buna bir bakalım.**

****

1. **Adım: Çözümleme Süreci Sorun tanımlamadan çıkan karar doğrultusunda problemi kâğıt üzerinde çözelim.**



**3. Adım: Tasarım Süreci Çözümlenen sistem sonucunda tasarımı güncelleştirmeye geldi sıra.**

****

**4. Adım: Gerçekleştirim Süreci Tasarımı yapılan sistemin gerçekleştirmesine sıra geldi.**

****

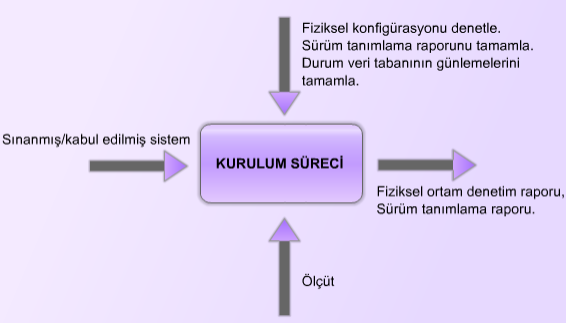
**5. Adım: Sistem Sınama Süreci Artık tekrardan tasarlanan sistemin sınama sürecini tekrar ele almak gerekiyor.**

****

**6. Adım: Kabul Sınaması Süreci Kendi içimizde sınadığımız sistemi birde müşteri karşısında sınıyoruz.**

****

**7. Adım: Kurulum Süreci Kabul sınamasını geçen sistemimiz artık tekrardan kurulum aşamasına geçiyor.**

****

**8. SONUÇ**

Sonuç olarak sistem hayata geçirildiği zaman neler değişeceğini gözler önüne serdik. Bunun yanı sıra basit ama bir o kadarda güvenli olan bu sistemle ek masraflar ortadan kalkacak hataları ortadan kaldırılacak ve hastaların eczacıları ile iyi bir görüşme yapmalarını sağladık .

Hem personel için hem yetkililer için oturdukları yerden işletebilecekleri bu sistem sayesinde artık eskisi kadar yorulmayacak ve herkes bu sisteme minnettar kalacak. Kurumumuz içinde iyi bir referans olacak olan bu sistem sayesinde hem biz hem kullanıcılar için çok güzel olacak. Bu zorlu süreç boyunca bize destek olan herkese bir kez daha teşekkürlerimi iletiyorum.

**9. KAYNAKLAR**

1) Software Engineering, I. Sommerville, Pearson Education Inc., A.B.D., 2011.

2) Report on Software Engineering Conference, NATO Science Committee, P. Naur, B.Randell, Almanya, 1968.

3) Yazılım Proje Yönetimi: Şelale Modeli ve Çevik Yöntemlerin Karşılaştırılması, Cevriye GENCER1, Ali KAYACAN, BİLİŞİM TEKNOLOJİLERİ DERGİSİ, CİLT: 10, SAYI: 3, TEMMUZ 2017.

4) Yazılım Mühendisliği,Yrd.Doç.Dr. Yunus Emre SELÇUK, Şubat 2017. 5) Sistem/Yazılım Geliştirme Sürecinde Doğrulama Faaliyetleri, Mikrodalga ve Sistem Teknolojileri (MST) Grubu, ASELSAN A.Ş., Ankara

5)M.T.Turgay “Yazılımda Kod Gözden Geçirme Sürecinde Kod Kalitesi Ölçümünün Sürece ve Yazılım Kalitesine Etkisinin İncelenmesi “ İstanbul Teknik Üniversitesi Eylül 2008

6)U.Gürtürk,” Proje Uygulamaları ve Dökümantasyonu” Fırat Üniversitesi Mayıs 2013

7)M.Ateş,” Proje Uygulamaları ve Dökümantasyonu” Kahranmaraş Sütçü İmam Üniversitesi Haziran 2021

### İnternet Kaynakları

* Yazılımda Sınama Teknikleri PowerPoint Presentation (firat.edu.tr)
* Yazılım Testi ve Test Süreçleri (itu.edu.tr)
* A.Erdem, “Yazılım Doğrulama ve Geçerleme Süreci ve Test Aşamaları “ https://blog.metu.edu.tr