****

**T.C.**

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**Yazılım Mühendisliği(İ.Ö)**



**YAZILIM MÜHENDİSLİĞİ TEMELLERİ**

**PROJE AMAÇ RAPORU**

**Proje Çalışma Grubu**

**16541055 –Ferhat KORKUTATA**

**GİRİŞ**

**Projenin Adı:** Elektronik Belge Yönetim Sistemi

**1.1. Projenin Amacı**

Projemiz, üniversitemizdeki dosya, rapor ve her türlü belge işleminin web üzerinden daha hızlı ve güvenilir bir şekilde yapılmasını sağlayarak, gerek akademisyenlere gerekse sistem kullanıcılarına daha az maliyetli ve daha hızlı bir çalışma ortamı imkânı sağlamaktadır. Elektronik ortamda belgelerin hangi kuruma ya da kişiye ait olduğunu görerek ya da kişiye gelen ya da giden belgeleri güvenli bir şekilde görüp belgelerin elektronik imza ile onaylanmasını ya da reddedilmesini işlemlerin zaman kaybedilmeden, aynı şekilde güvenilir ve her platformdan erişim imkânı sağlamayı amaçlayan bir projedir. Bu proje sayesinde kullanıcılar, belgelerinin kendilerinin sahip oldukları yetkiler doğrultusunda, sistemde kayıtlı olan onaylı belgelerine de belgelerin sistemde saklanması süresi göz önünde bulundurularak istedikleri zaman erişebilme imkânına sahip olacaktırlar.

**1.2 Projenin Kapsamı**

Üniversitenin ağı üzerinde çalışacak bu sistem akademik yetkisi olan kişilerin birbirleri ile iletişim sağlamalarını kolaylaştıracaktır. Yetkili kişilere gelen belgelerin e- imza ile imzalanarak, kişinin belgesinin aşamalarını sistemde görüp hangi aşamada olduğunu ve bir sonraki aşamanın ne olduğunun görüntüleyerek sistem kullanıcısına belgesini aşama aşama takip etmeyi hedefleyen bir sistemdir. Ayrıca projemiz sunduğu basit ara yüzle sistem hakkında pek fazla bilgisi olmayan kullanıcılara da kolay kullanım rahatlığını sunmaktadır. Sistemimizde e-imza yetkisi ile belgelerin onayının alınmasından dolayı sistemimizde güvenlik kullanılabilirlik ve sistem hızından önce gelmektedir.

**2. PROJE PLANI:**

**2.1 Giriş**

Elektronik belge yönetim sistemi, kâğıttan ve zamandan tasarruf ile kullanıcı dostu olan bir ve üniversitemizde belge ve bilgi alış verişinin elektronik ortama aktarılmasını ve internet üzerinden bu bilgilerin anlık olarak yönetilmesini amaçlayan bir sistemdir. Ayrıca kurum içi ve kurum dışı yazışmalarınıza ait süreçlerinizi kolaylıkla yönetilmesini sağlar. Yazışmalar için harcanan kaynakların minimuma indirilmesi de amaçlanmıştır. Fiziksel ortamdaki yaşanan sorunların ( belgelerin kaybolması, ulaşım süreleri ile ilgili sorunlar, fakslarda yaşanan gecikme ve kopukluklar, personel açığından kaynaklanan sıkıntılar ) önlenmesi veya en aza indirilmesi de sistemin önemli avantajlarındandır.

**2.2 Projenin Plan Kapsamı**

Projenin plan kapsamında genel olarak mevcut sistem, sistemin gerekliliği ve bu sistemin güvenilirliğinden yola çıkıldı. Sistem internet üzerinden gönderilen belgelerin güvenilir ve zamanında işlem görmesini sağlayarak, aynı zamanda bunu yaparken de kâğıt israfını engellemek amacıyla gerçekleştirilen bir projedir. Üniversitemizde bu sistemi kullanarak belge kaybı belgelerin onayı vb. aksaklıkların önüne geçilmek amacıyla bu sistemin yapılmasına gerek duyulmuştur. Ayrıca projenin telefon uygulaması olarak gerçekleştirilecek olması ve uygulamanın bildirim özelliğinin kullanıcıyı sürekli sisteme girip aşama aşama belgesinin adımlarını kontrol etme yükünden de muaf etmiştir. Kullanıcı anlık bilgiler ile kendisine gelen belge isteklerini ya da beklediği belge onaylarını anında zaman kaybı olmadan bildirim olarak görebilecektir.

**2.3.Proje Kaynakları**

**2.3.1. Donanım Kaynakları**

Gerçekleştirim aşamasında olan “Elektronik Belge Yönetim Sistemi” sisteminin de kullanılacak donanım kaynakları belirtilmiştir.

**Donanım**

Telefon veya ilgili bir parçanın (sunucu, telefon ) gözle görülebilen ve elle tutulabilen kısımlarına donanım denir.

**Proje de kullanılacak donanım kaynakları:**

* Kullanıcı Telefonları
* Sunucular (uygulama sunucuları)
* Kablosuz Alan Ağ Yapısı (WLAN)
* Geniş Alan Ağ Yapısı (WAN)

**2.3.1.1.Kullanıcı telefonları**

İnsan ilişkisinin en yoğun olduğu yerdir. Kullanıcı telefonları projemizde gerekli kullanım alanlarına göre yazılacak programların program dilleri seçilerek hayata geçirildiği bölümdür. Bu bölüm iki aşama da ele alınacaktır.

* **Gerçekleştirim Aşamasında Kullanılacak Kullanıcı Bilgisayarları ve Özellikleri**

**Sayı:** 7 adet PC (min.)

**Özellikler: Windows 8 ve üzeri işletim sistemi,**8GB Ram,1 gb ekran kartı, 1TB Sabit Disk, 2 üzeri işlemci hızı, HDMI, SSD

* **Kullanım Ortamında Kullanılacak Kullanıcı telefon ve Özellikleri**

**Sayı:** Birimdeki çalışan kişi sayısı kadar gerekmektedir.

**Özellikler:** 4GB Ram(min.)



**2.3.1.2.Sunucular**

Teknik olarak ele alındığında sunucu; verilerin, bilgisayar ağları üzerinden kullanıcıların erişimine açık olarak barındırıldıkları bilgisayar sistemlerine verilen isim. Daha basit bir ifadeyle sunucu, kullanıcının verilere online olarak ulaşabilmesini sağlayan, bu verileri onların kullanımına sunan sistemlerdir.

Projede kullanılacak sunucular anlık verilerin korunması, paylaşılması, iş akışının hızlı bir şekilde yapılandırılması için zorunlu bir gereksinimdir.

****

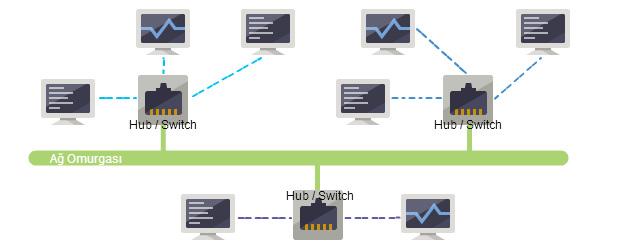
**2.3.1.3.Geniş Alan Ağları (WAN)**

Kurumların farklı coğrafi alanlarda faaliyet göstermesi ve bu lokasyonlar arasında veri haberleşme ihtiyacının doğması sonucu geniş ağ çözümlerine ihtiyaç duyulmuştur. Geniş alan ağı, birden fazla cihazın birbiri ile iletişim kurmasını sağlayan fiziksel veya mantıksal olarak tasarlanan büyük ağdır. Yerel ağların birbirine bağlanmasını sağlayarak çok geniş bir yayılıma ulaşırlar. Geniş alan ağları, bir mekânda bulunan kullanıcı ve bilgisayarların başka bir mekânda bulunan kullanıcı ve bilgisayarlarla iletişim kurabilmesi için LAN’ları ve başkâtip ağları birbirine bağlamakta kullanılır. Birçok WAN tek bir organizasyon için kurulmuştur ve özeldir. İnternet Servis Sağlayıcıları tarafından kurulan diğer WAN’lar ise bir organizasyonun LAN’ından Internet’e bağlantı sağlar.

**Ağ topolojileri**

Ağaç topolojisi yıldız topolojisi ile ortak yol topolojisinin birlikte kullanıldığı topolojidir. Merkezdeki bir ortak yol (omurga) üzerine yerleştirilmiş hub ya da switch’ lere bağlı cihazlarla oluşturulur.

Bu topolojiye ağaç topolojisi denmesinin nedeni omurganın ağaç gövdesini, hub ya da switch üzerinden bağlanan bilgisayarların da ağacın dallarını modellemesidir. Ağaç topolojisi büyük ağların omurgalarını oluşturmak için kullanılır.



**Ağaç topolojisinin avantajları;**

* Farklı üreticilerin donanımları ile uyumlu çalışır.
* Ağın genişletilmesi kolaydır.
* Sorunların tespiti ve giderilmesi kolaydır.
* Ağın yönetimi ve bakımı kolaydır.
* Dallardan birinde oluşacak sorun diğerlerini etkilemez.

**Ağaç topolojisinin dezavantajları;**

* Kablolama işlemi zordur.
* Dallanma arttıkça ağın bakımı ve yönetimi zorlaşır.
* Omurgada yaşanacak bir sorun tüm ağı etkiler.

**2.3.2. İnsan Kaynakları**

**2.3.2. 1.Proje Yöneticisi**

Proje yöneticisi yazılım ekibini bir arada tutan, gerekli toplantıları düzenleyen ve zaman çizelgelerine uyulması için gerekli motivasyonu sağlayan kişidir. Ayrıca yönetim ile proje ekibi arasındaki bilgi alışverişinin de sağlar. Bütçe konularında düzenlemeler ve maliyet analizleri konusunda proje ekibine bilgi ve tavsiye verir.  
( Ferhat KORKUTATA )

**2.3.2. 2.Kalite Uzmanları**

Proje ekibinde kullanılan yazılım ve donanım araçlarını kontrol eden ve herhangi bir problem anında proje ekibine teknik destekte bulunan uzman kişidir. Bizlere proje ekipmanlarının kalite ayrımı konusunda bilgi veren aynı zamanda proje alanındaki gelişmeleri takip eden raporlama ve dokümantasyon becerileri yüksek, planlama ve koordinasyon becerisi gelişmiş kişidir.  
( Ferhat KORKUTATA )

**2.3.2. 3.Yazılım Ekip Lideri**

Sıfırdan gelen bir proje için uygun teknolojiyi ve uygun iş gücünü temin eden. Gerekli elverişsiz durumların ne kadar süreceği ne kadar maliyet tutacağını belirleyen ve analizleri en ince ayrıntısına kadar kontrol eden yetkili kişidir. Bizim projemizde kullanılan toplu proje gelişiminin sağlama aşamasını belli eden iş sistemi planlama aracını (EBYS) proje gelişimi takip edip görev dağılımını yapan yetkili kişidir.

( Ferhat KORKUTATA )

**2.3.2. 4.Donanım Ekip Lideri**

Proje kapsamında gerekli donanımsal kaynakları tespit edip; ekip çalışmasına uygun hale gelebilmesi için ekip içerisinde ki teknik kişilerle konuşup gerekli istişareleri yaptıktan sonra verilerin iletilmesi konusunda destek veren ve kontrol eden yetkili kişidir.

( Ferhat KORKUTATA )

**2.3.2. 5.Donanım Tasarım Mühendisi**

Yazılımsal olarak ekibin aşamaları inceleyip gerekli bilgiye sahip olan projenin hayata geçirilmesi konusunda etkin rol oynayan ve donanım gereksinimleri ile ilgili konularda müşteri ile irtibat kuran. Bizim projemizde de sistemin kurulum aşamasında istihbarat kuruluyla görüşen, donanım destek elemanı ile sürekli iletişimde bulunan ve proje isteklerinin karşılanması konusunda ekibi bilgilendiren mühendistir.

( Ferhat KORKUTATA )

**2.3.2. 6.Proje Sekreteri**

Proje içerinde dökümanların saklanması ve etkin bir şekilde ilerleyişi not eden proje ekip yöneticisinin yardımcısı olan kişidir. ( Ferhat KORKUTATA )

**2.3.2. 7.Bilgisayar Ağ Uzmanı**

Bağlı olan uygulama sunucuların bakımları, bu uygulama sunucuların problemlerinin giderilmesi ve aynı zamanda güvenlik tedbirlerinin alınmasını sağlayan kişidir. İhtiyaç doğrultusunda yeni cihazların tekliflendirilmesi yapan kişidir.

( Ferhat KORKUTATA )

**2.3.2. 8.Sistem Çözümleyicisi**Bilgi işlem sistemlerini kuran ve yeni bilgi toplayan, gereksinimlerin özelliklerini belirleyen sistemlerin kurulmaları ve çalışmaları için gerekli yöntemleri tanımlayan, kurulumlarını yapan, denetleyen ve gelişmeleri için önerilerde bulunan nitelikli kişidir. Bizim projemizde bilgi ağı konusunda gereken mercilerle görüşen ve bu bilgilerle gelişmiş çözümler amaçlayıp ekibe bildiren açıklayan kişidir.

( Ferhat KORKUTATA )

**2.3.2. 9.Yazılım Destek Elemanı**Yazılım destek elemanı, satılmış olan yazılım ürünlerinin kullanılacağı ortamı hazırlar ve veri tabanını kurar, satış yapıldıktan sonra ücretli olarak danışmanlık ve destek hizmetlerini müşteri talebi doğrultusunda yerine getirme görevini üstlenir.

( Ferhat KORKUTATA )

**2.3.2. 10.Sistem Tasarımcısı**Sistem çözümleyicinin tanımladığı gereksinimleri mantıksal, ekonomik(projenin daha hızlı çalışması, sistem karmaşıklığı) ve kullanım açısından pratik sistem tasarımlarına dönüştürerek ilgili programların yazılabilmesi için gerekli ayrıntılı özellikleri hazırlayan kişidir. ( Ferhat KORKUTATA )

**2.3.2. 11.Donanım Destek Elemanı**

Müşteri memnuniyetini göz önünde bulunduran donanım mühendisinin yardımcısı olup ekip ile donanım mühendisi arasındaki iletişim görevini yapan kişidir. Bizim projemizde de istibaharat alanındaki yetkili personellerle iletişim kuran, teknik aksaklıklar da problemlere çözüm bulan ve bu iletişimsel verileri donanım ekibi ile istişare ederek ekibe ileten kişidir. ( Ferhat KORKUTATA )

**2.3.2. 12.Programlama Uzmanı**Yapılan analizlere göre belirlenen teknolojiyi, platformu kullanarak yazılımı kodlayan kişidir. Bizim projemizde gereksinimler doğrultusunda uygun programlama dilini seçerek uygun yazılım araçlarını kullanarak gerçekleştirmeyi sağlayan kişi konumundadır.

( Ferhat KORKUTATA )

**2.3.2. 13.Eğitim Ekip Lideri**

Programı kullanılacak olan müşterilere anlatılması gereken detayları belirleyen ve bunları eğitmenlere aktaran ayrıca verilecek olan kullanım kılavuzları belirleyen ve oluşturan uzman kişidir. ( Ferhat KORKUTATA )

**2.3.2. 14.Sistem Yöneticisi**

Sistem alt yapısını kuran, sistemin gelişme planını hazırlayan tüm bu bilgileri analiz ve kontrol eden kişidir. Aynı zamanda sistem yazılımın, donanımının ve bunların kurulumuyla beraber konfigürasyonunu düzenleyen uzman kişidir. Bizim projemizde de konfigürasyon planında etkin rol oynayan sistem yazılımın ayrıntılarında katkıda bulunan uzman kişidir.

( Ferhat KORKUTATA )

**2.3.2. 15.Eğitmen**

Kullanıcıya proje sisteminde, arayüzünde ve veri tabanında kullanımı sağlamak adına eğitim veren profesyonel kişidir. Bizim projemizde istihbarattaki ve birimlerdeki kullanıcılara anlaşılabilirliği arttırabilmek için gerekli yazılımsal bilgileri ileten kişidir. ( Ferhat KORKUTATA )

**2.3.2. 16.Veri Tabanı Yöneticisi**

Veri tabanı sistemlerinin kurulması, konfigürasyonun yapılması, tasarlanması, sorgulanması ve güvenliğinin sağlanması işlemlerini üstlenmiştir.

( Ferhat KORKUTATA )

**2.3.2. 17.Denetleyici**

Proje aşamasında zaman-maliyet ve iş-zaman alanını denetleyen aksi bir durumda uygun çözümler bulan yetkili kişidir. Bizim projemizin aktif zamanda yapılan iş planını kontrol eden ve gerekli durumlarda çözüm yöntemleri sunan yetki düzeyindeki kişidir.

( Ferhat KORKUTATA )

**2.3.2. 18.Tanıtım ve Medya Uzmanı**

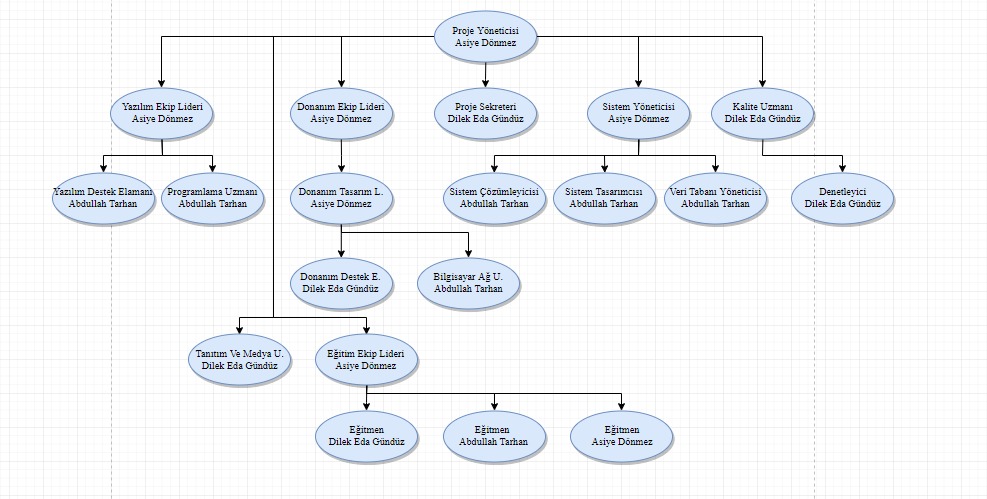
Proje gelişimin güncel reaksiyon şeklinde toplumla paylaşan ve yöneticiye görsel anlamda aktaran yetkili kişidir. Bizim projemizde iletişimizin kuvvetlenmesi için medya anlamında ekip başlıkları kuran ve ekip liderinin görev dağılımının etkin olarak sağlandığını çeşitli fotoğrafları, videoları ortak kullanım uygulaması olan EBYS'da paylaşıp yöneticiye ileten kişidir.

( Ferhat KORKUTATA )

**Proje Ekip Yapısındaki Ek Görevler**

* EBYS sistemin incelenmesi ve gereksinimlerin belirlenmesinde rol alacak kişi: Ferhat KORKUTATA
* EBYS çalışma mantığı anlamadan sorumlu kişi: Ferhat KORKUTATA

**Proje Ekip Yapısı İlişki Gösterimi**

****

**2.3.3.Yazılım Kaynakları**

**İş Sistemleri Planlama Araçları:** İş sistemleri planlama araçları, kurumlardaki iş akış yapısının üst modelinin üretilmesinde kullanılmaktadır. Bilgi akışı, bilgi yapısı, iş birimlerindeki tıkanıklıklar bu araçlar kanalıyla ortaya çıkarılır. Kullanılan araçlar;

* Visio
* draw.io
* Adobe Fireworks cs5
* argoUML

**Proje Yönetim Araçları:** Proje yöneticisi tarafından, projede yapılan işlerin izlenmesi, kaynak ataması, proje iş yapısının üretilmesi, gözlenen değerlerin işlenmesi türündeki işlerin yapılmasını sağlayan araçlardır.

* Trello
* Whatsapp
* Skype
* Teamviewer

**Çözümleme ve Tasarım Araçları:** Sistem yazılım kullanıcı yordamları, metin düzenleyiciler, derleyiciler, hata ayıklayıcılar, nesne kökenli programlama araçları, görsel programlama platformları türündeki programlama araçları yazılım geliştirmede kullanılması kaçınılmaz araçlardır.

* Word
* Excel
* Windows Media Player
* Publisher

**Programlama Araçları:** Doğruluk, bilginin hatasız olması ile özdeştir. Büyük bir veri yığınıyla uğraşıldığında, genelde kayıt ve hesaplama hataları ortaya çıkar. Bu gibi durumlarda, doğruluk özelliği daha fazla önem kazanır.

* Android Studio
* VİSUAL STUDİO

**Sınama Araçları:** Kapsam çözümleyiciler, sınama verisi üreticiler, otomatik sınama yordamları, yazılımın doğrulama ve geçerleme işlemlerinde kullanılmaktadır.

* TESTRAİL
* SELENİUM
* JUNİT

**Prototipleme ve Benzetim Araçları:** Bu araçları temel olarak, geliştirmenin erken aşamalarında kullanıcıya, sonuç ürünün çalışması ile ilgili fikir vermek ve yönlendirmek amacıyla kullanılmaktadır. Bu araçlar sayesinde, kullanıcı sonuçta elde edeceği ürünün davranışı ile ilgili bilgiler edinir ve sonradan ortaya çıkabilecek farklı yorum ve algılamalar önlenmiş olur. Genellikle word metin editörü kullanılmıştır.

* PIDOCO

**Bakım Araçları**: Program bakımını kolaylaştıran, programın anlaşılmasına yönelik olarak kullanılan tersine mühendislik ya da yeniden mühendislik araçları bakım araçlarına örnek olarak verilebilir. Bu araçlar, verilen bir kaynak kodundan, program şemalarının üretilmesi, program veri yapısının ortaya çıkarılması gibi işlevleri yerine getirirler. Bakım aşamasına daha gelinmediğinden daha sonra eklenecektir.

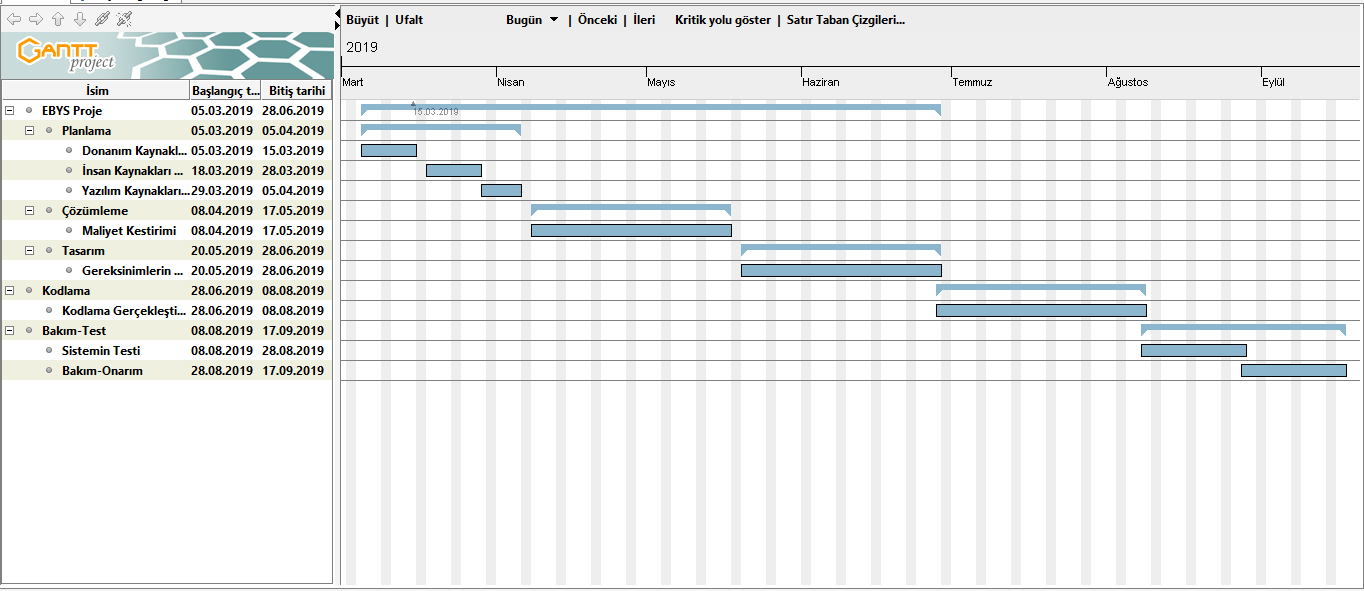
* Atlassian Jira Talep Takip (Issue Tracking) Yazılımı
* Atlassian Confluence (Wiki) Yazılımı
* Atlassian Fisheye + Crucible Yazılımı

**Destek Araçları:** İşletim sistemleri, belge işleme sistemleri, ağ yazılımları, elektronik posta ve ortam yönetim araçları, bu araçlara örnek olarak verilebilir.

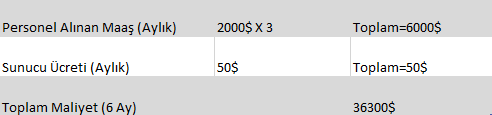
İşletim Sistemleri: Windows 10-8, Linux, Android

Elektronik posta: Gmail

**2.4.İş-Akış Diyagramı**

****

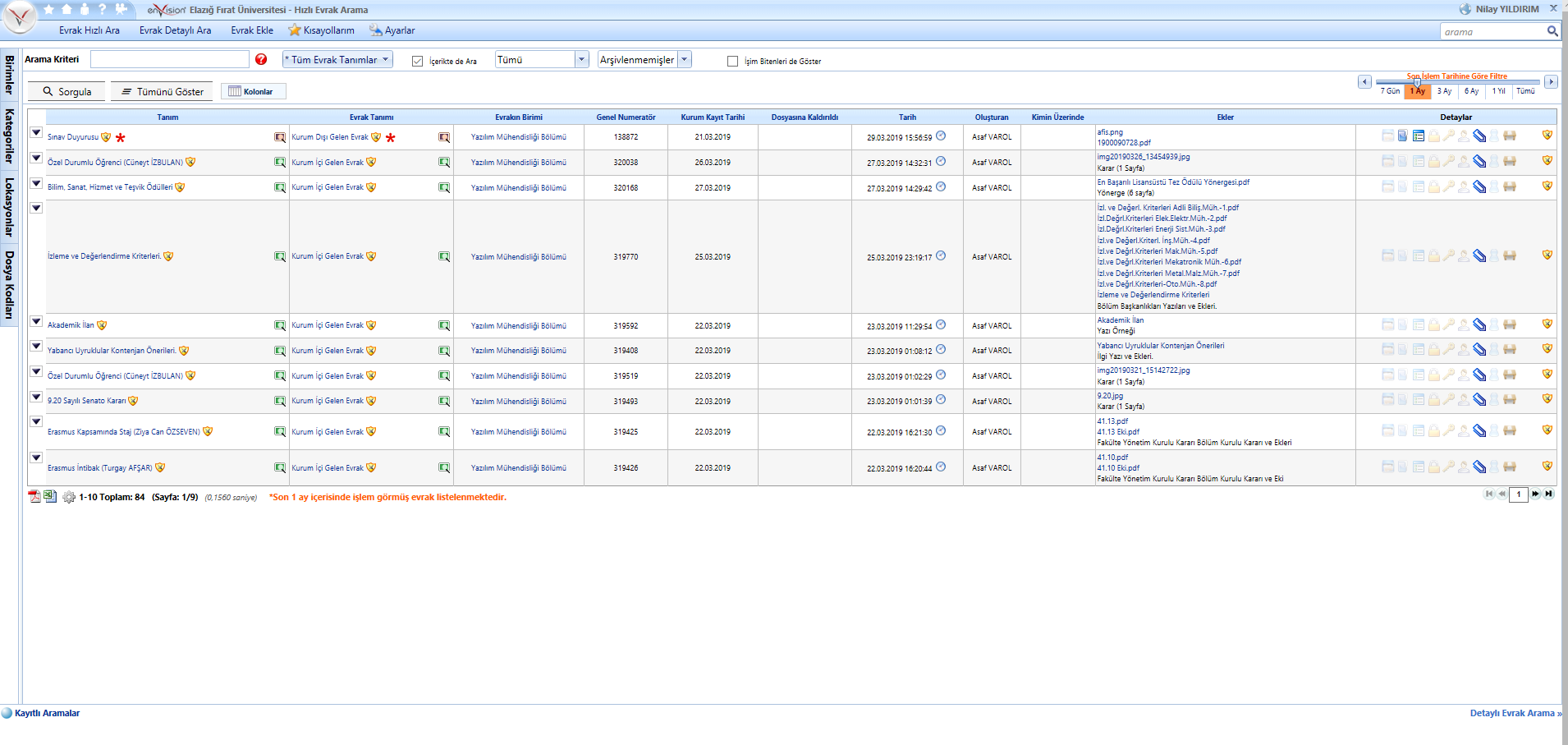
**2.5.Maaliyet Kestirimi**

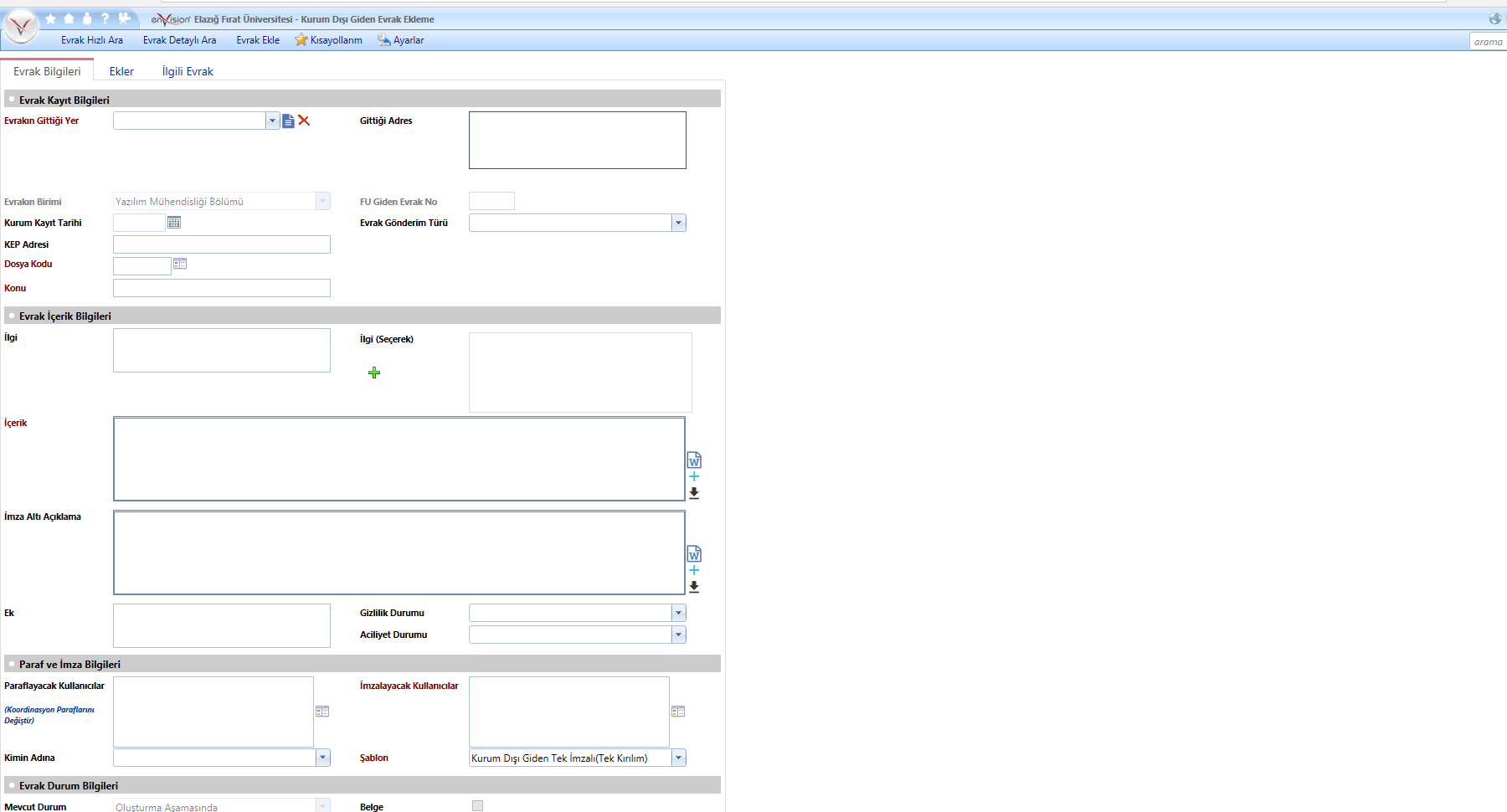
****

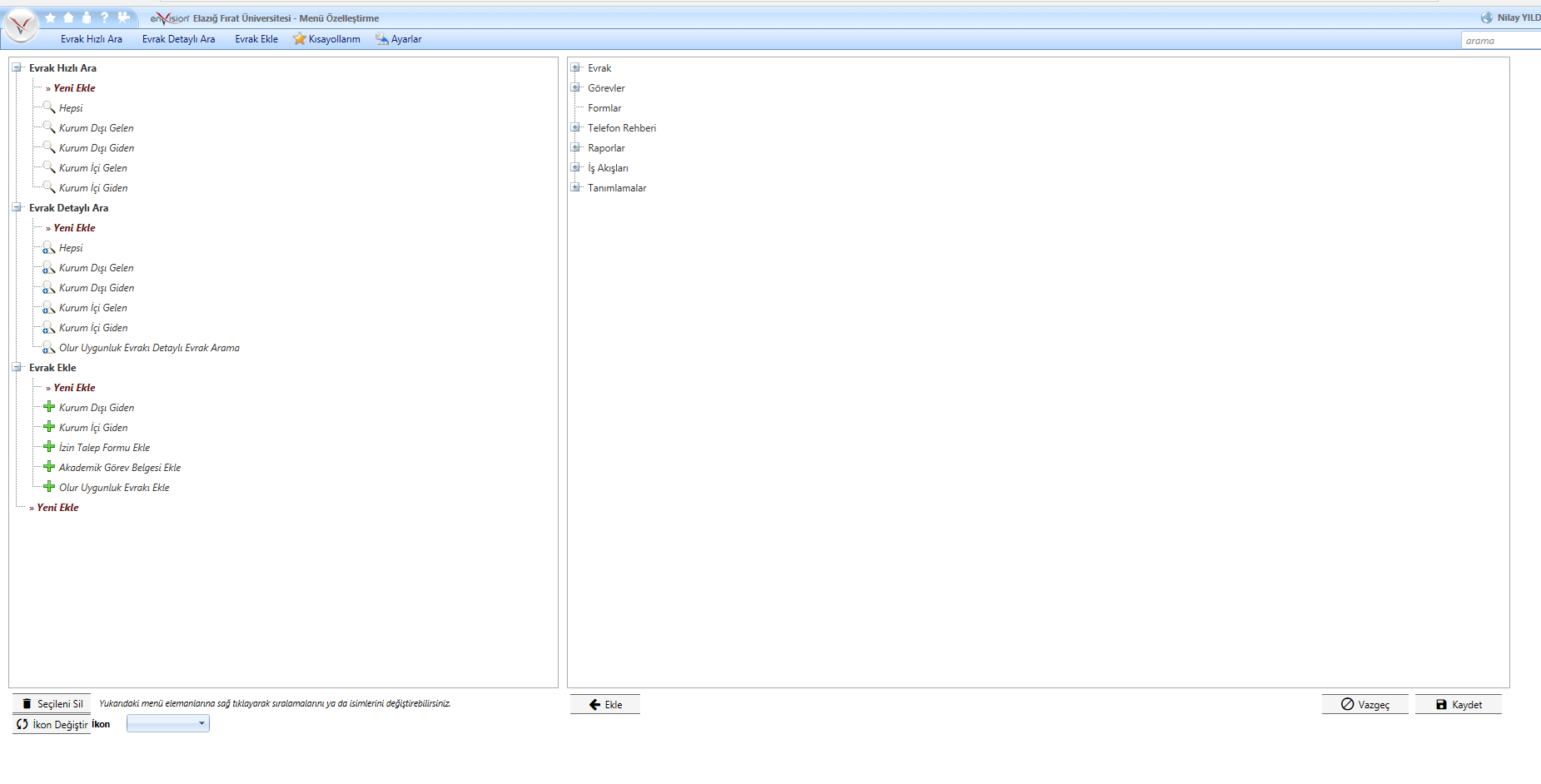
**3. SİSTEM ÇÖZÜMLEME**

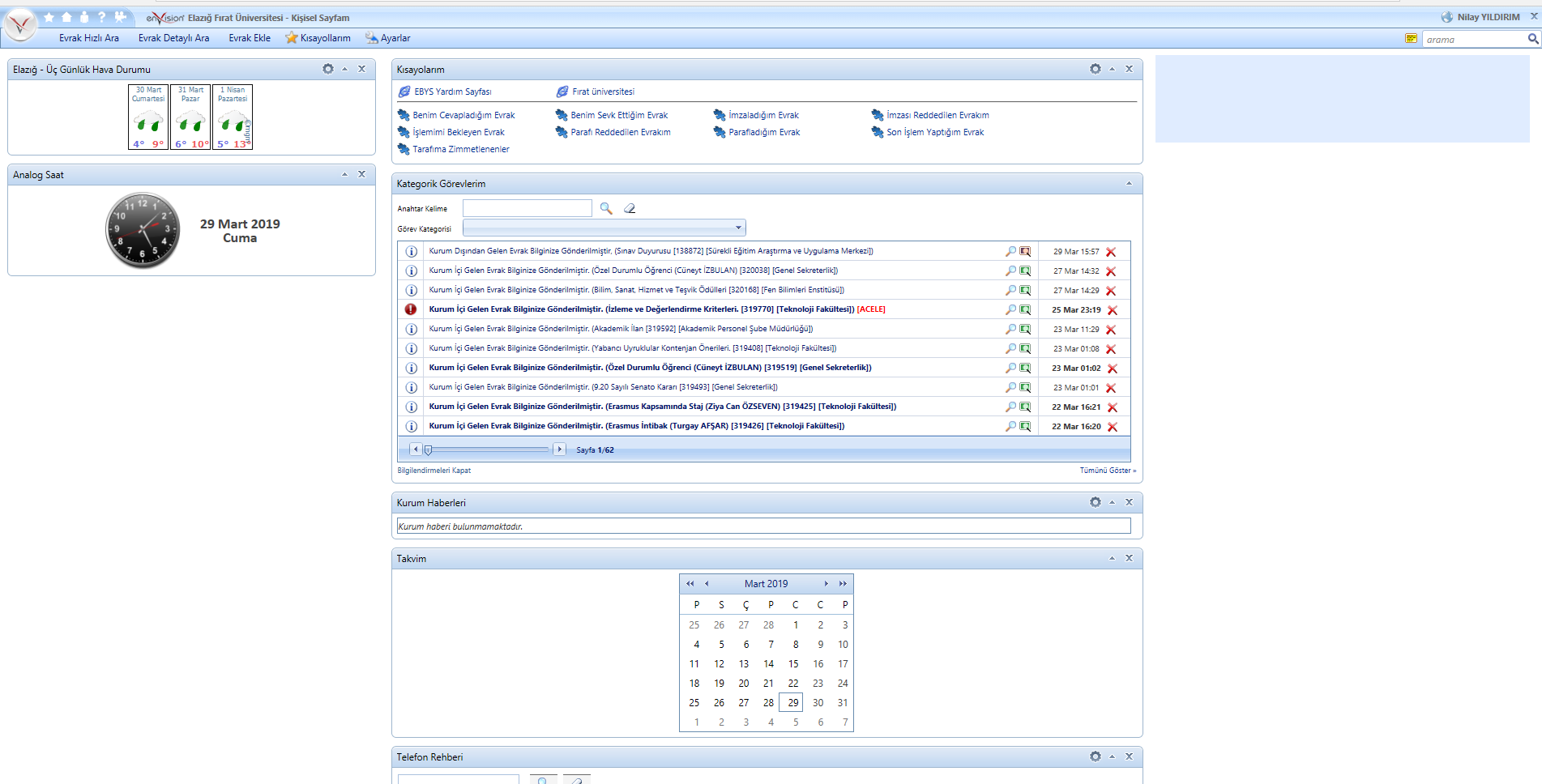
**3.1 Mevcut Sistem İncelemesi**

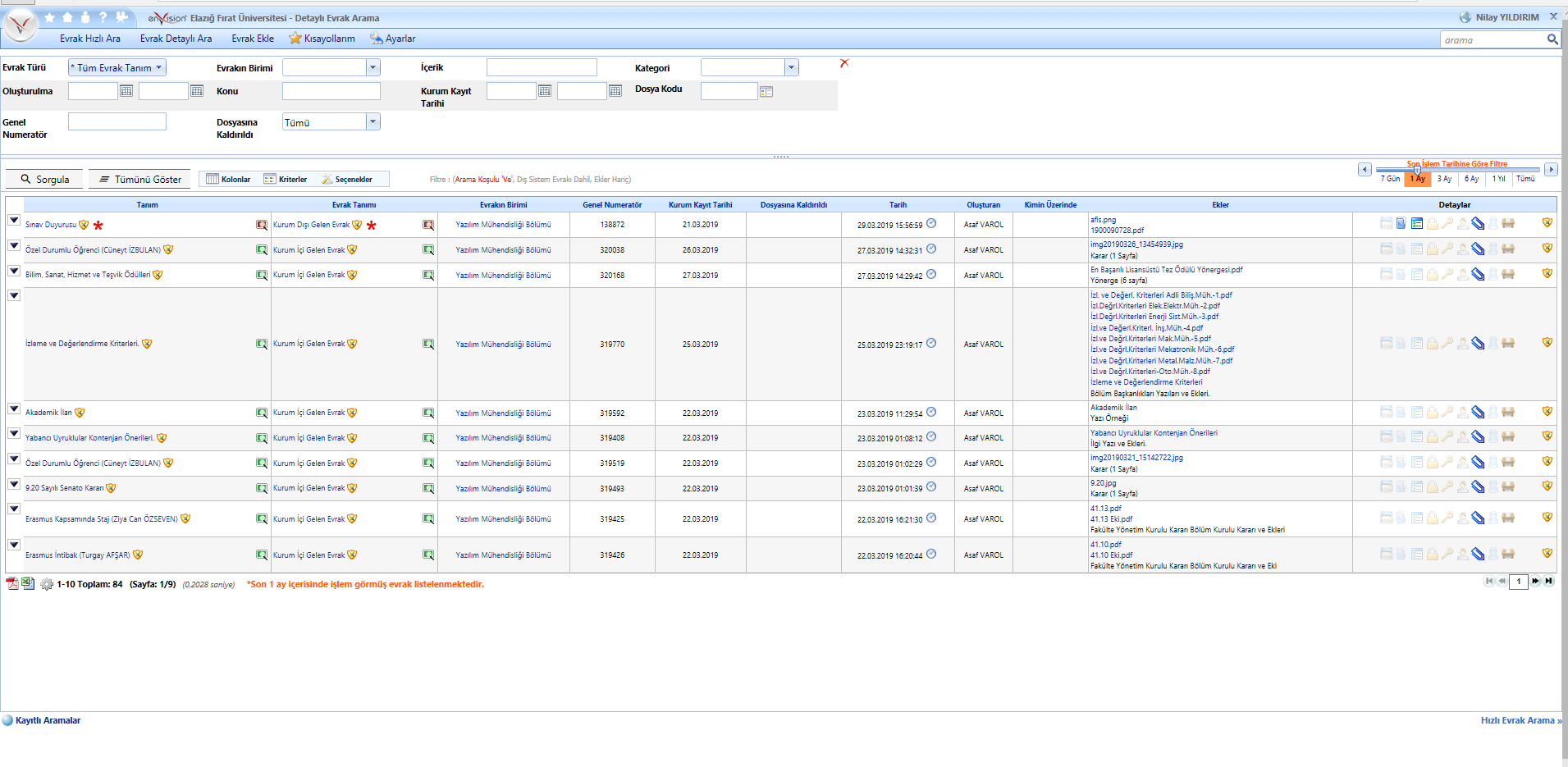
Mevcut sistem üniversitemizde var olan bir sistemdir. Bu sistem bu sisteme kayıtlı olan kişilerin belgelerini yönetebilme imkânı sağlamaktadır. Belgeleri gelen giden ve onay beklenen belgeler olarak sınıflandırarak görüntüleme imkânı sağlamaktadır.

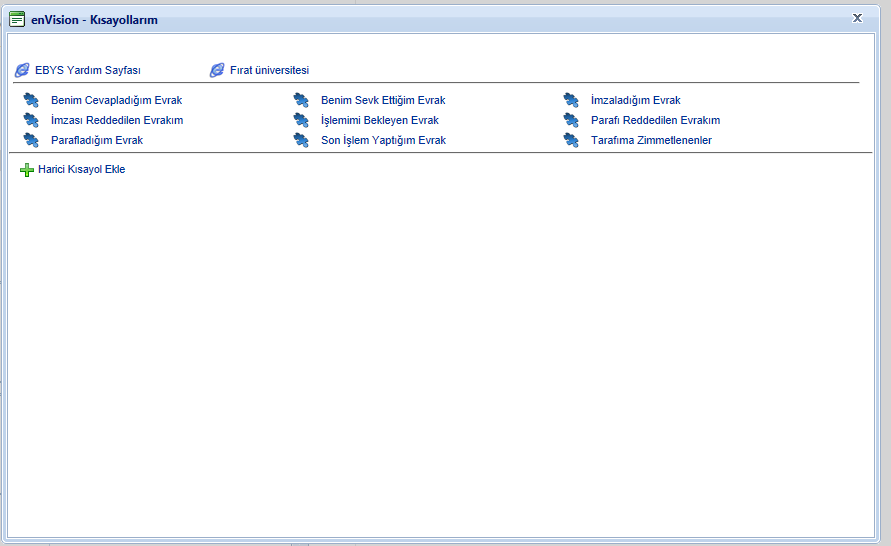










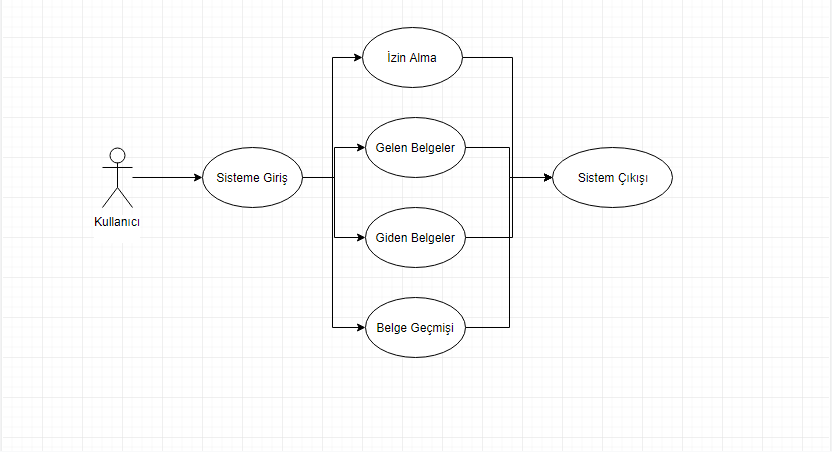


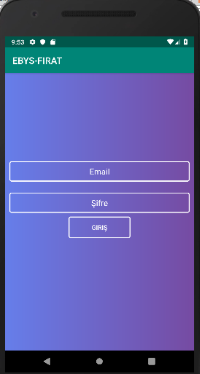
**3.2 Gereksenen Sistemin Mantıksal Modeli**

**3.2.1 Giriş**

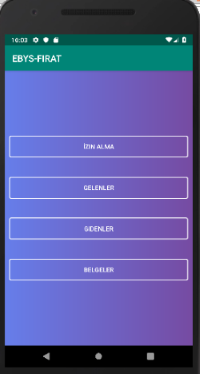
Yapmak istediğimiz elektronik belge yönetim sistemi sadece web tabanlı olan bu sistemi üniversitemizde mobil bir uygulama haline getirilerek daha iyi bir kullanıcı ara yüzü ile gerçekleştirmeyi hedeflemektedir. Yapacağımız mobil uygulama kullanıcıya verdiği anlık bildirimler ile sistemdeki veri transferini de en aza indirmeyi hedeflemektedir.

**3.1.2 İşlevsel Model**





**Giriş Arayüzü**: Sisteme sadece FIRAT üniversitesine ait akademik personeller giriş yapabilmektedir. Mail adresi ve şifre yazıldıktan sonra giriş gerçekleşecektir. Sistemde Şifremi unuttum ya da yeni kayıt gerçekleşmeyecektir. Şifremi unuttum kısmı tamamen bağlı olduğu kurumdan alınıp girişini doğrulayabilecektir.



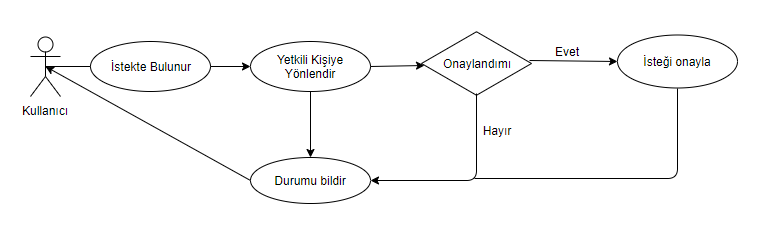
**Ana Menü Ara yüzü:** Ana menü içerisinde izin alma, gelen ve giden mesajlar ve son olarak bütün belge dokümanlarına erişilebilecektir bu kısımda izin alım için izin alma bölümüne giriş yapması gerekmektedir. Gelen izin cevapları ya da mesajları ise gelenler kısmına giriş yapması gerekmektedir. Gönderilen mesajlar ya da kendisine gelen mesaj cevaplarını gidenler kısmında görecektir. Bütün belge dokümanlarını ise belgeler kısmında yer almaktadır.

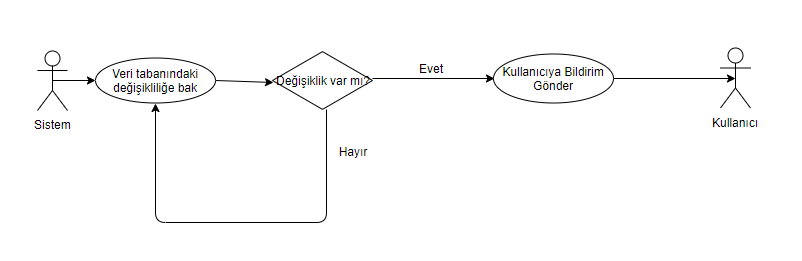


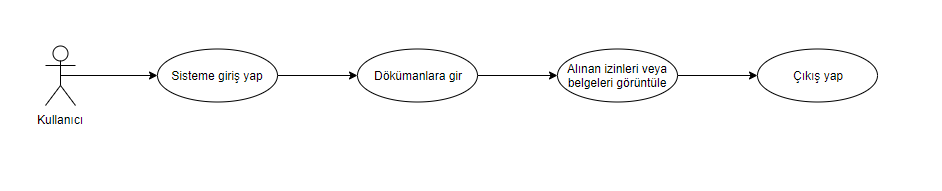
**İzin Alma Arayüzü:** Bu kısımda izin almak isteyen kişi önce nedenini belirtip izin al butonuna tıklaması gerekmektedir.

**3.1.3.Sistem Gereksinimleri (User Story )**

* Belgelerin kaybolmasını engellemek belgelerin yedeğinin alınması ve bu belgelerin güvenilir bir ortamda saklanması için ,
* Sisteme giriş yapan kişiye göre yetki vermesi güvenirlik kavramını üst düzeye taşıdığı için,
* Hangi işlemin ne zaman kimin tarafından yapıldığının görüntülenmesi ile hatayı yapan kişinin bu durumu inkar etme ihtimali olmadığı için ,
* Her türlü işlemin artık kağıt üzerinde değil de bilgisayar üzerinde yapılıyor olması ve devrin teknoloji devri olması ,
* Önemli kavramlardan olan kağıt israfının önüne geçildiği ,
* Akıllı telefonlarda uygulama olarak da kullanılabilmesi erişim kolaylığı olduğundan,
* Sistem tarafından gelen bildirimler sayesinde zaman kaybı olmadan aşama aşama adımların takip edilebiliyor olması.
* Web tabanlı olan sistemin mobil tabanlı (Android) uyarlanması
* Mevcut sistemin karmaşık olmaması.
* Bölüm başkanı ya da bölüm başkan yardımcısının hangi belgenin kim tarafından alındığı işlemini listeleyebilmesi.
* Kullanıcıların izinlerinin bölüm başkanı ya da yardımcısı tarafından görüntülenebilmesi.
* Kullanıcıların mesajlarının bildirim çubuğunda görüntüleyebilmesi.
* Kullanıcıların aldığı izin vb. işlemlerini adım adım nerede kimin tarafından onay beklendiğini görüntüleyebilmesi.
* **3.1.4.Gereksinim Diyagramları**







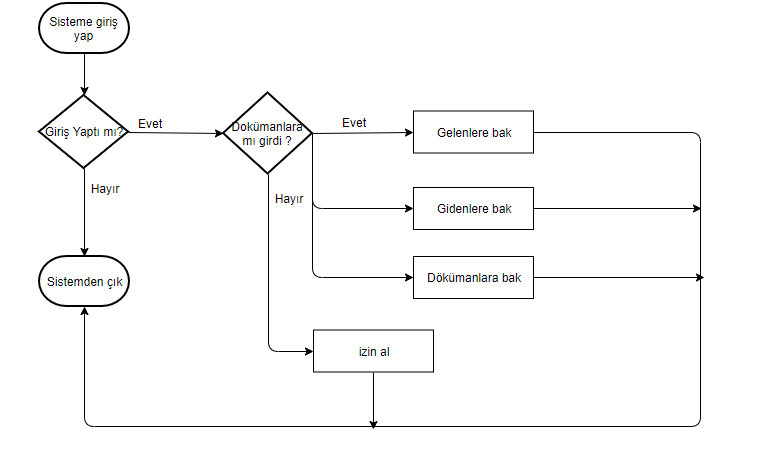
Vb. gibi sebeplerden dolayı elektronik belge yönetim sisteminin Fırat Üniversitesinde kullanılması ihtiyacı ortaya çıkmıştır.

**4. SİSTEM TASARIMI**

**4.1. Sistem Özellikleri**

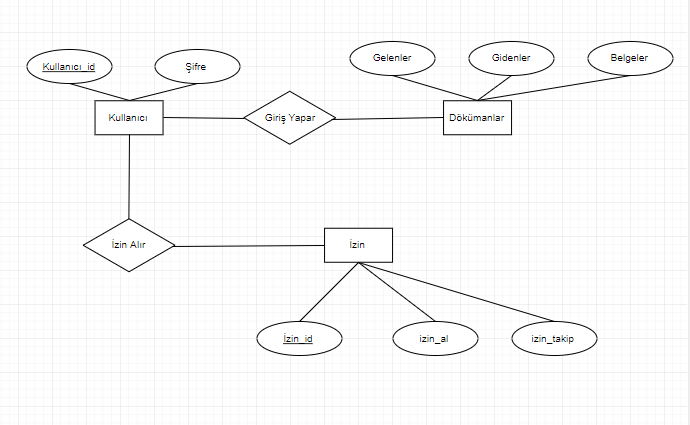
Sisteme giriş işlemi sms doğrulama ile yapılmaktadır. Bu sayede hackleme ve güvenlik konusunda bir adım daha öne geçilmektedir. Kullanıcı sisteme giriş yaptıktan sonra menüler ile işlemini kolaylıkla gerçekleştirebilmektedir.

**4.2. Sistemin Mimarisi**



Sistem üzerinde kullanılacak yollar kap kısmında belirtilmiştir zaten. Sistemdeki test aşamaları her ana maddeler üzerinden yapılacaktır. Ana maddeler şunlardır; sisteme giriş yapıldı mı? , dokümanlara girildi mi? Ve son olarak dokümanlara girilmediyse izin alındı mı? Bu ana maddeler içerisinde test aşamaları gerçekleşecektir. Sistemin giriş kısmında test aşaması sistemde dâhil olmayan kişilerin girişinin yapılmamasının kontrolünü kapsayacaktır. Doküman kısmında ise mesajların doğru iletilip iletilmediği testi gerçekleşecektir.

**4.3. E-R diyagramı**



**4.1.4 Testler**

* **Alfa Sınaması**

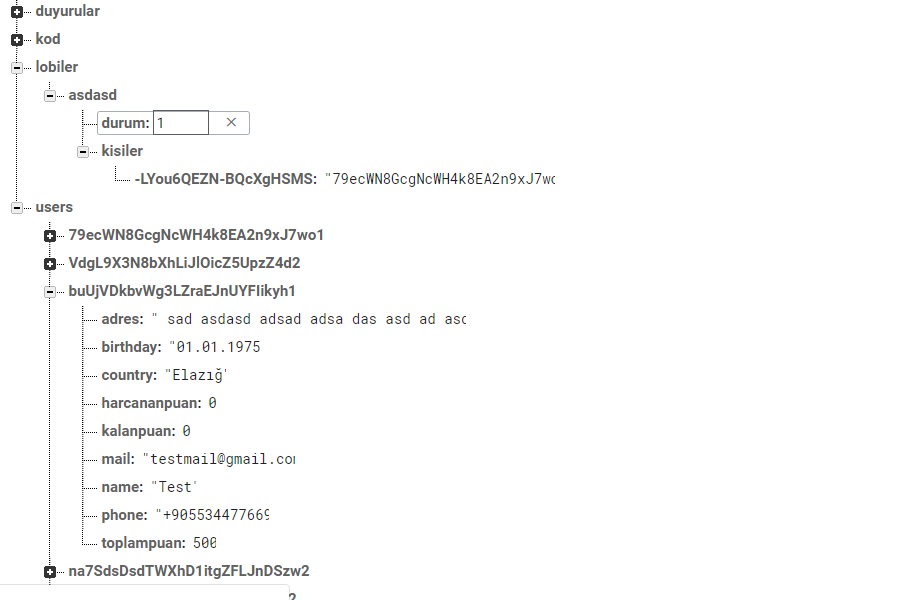
Sistemin geliştirildiği yerde kullanıcıların gelerek katkıda bulunması sistemi test etmesi amaçlanmaktadır.

* **Beta Sınaması**

Kullanıcı, geliştirilen sistemi kendi yerleşkesinde, bir gözetmen eşliğinde yapar.

**4.2 Veri Tasarımı**

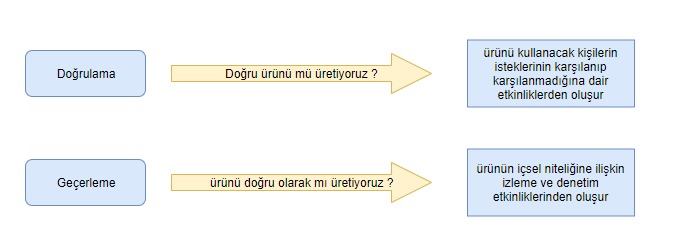
Veritabanı olarak Firebase (NoSql-Json) kullanılmıştır.



**6. DOĞRULAMA VE GEÇERLEME**

|  |
| --- |
|  |
| **6.1.Giriş**  d9b7c5f5-8904-4eee-8d18-13e6e68d087d (1)gbnvn |

Şimdi yazılım geliştirme süreci açısından bu doğrulama ve geçerleme kavramlarına bakalım, geçerleme aktiviteleri, yazılım halen geliştirilirken uygulanırken, doğrulama  aktiviteleri genelde bir faz/modül bittiğinde ya da bütün yazılım tamamlandığında gerçekleştiriliyor. Yazılım doğrulama sürecinin ürün geliştirilirken (analiz, tasarım, geliştirme) devamlı olarak kontrol edilerek, uygulamanın sürekli olarak belirtilen şartlara göre oluşturulduğunu kontrol altına alırken, doğrulama (“Validation“) faaliyeti, doğrulama noktasına nasıl geldiğimiz ile de ilgilenmez (arada kodları gözden geçirdik mi, ara test yaptık mı ya da gereksinimleri gözden geçirdik mi). Sadece doğrulama noktasında mıyız değil miyiz ona bakar. Geçerleme yüksek kalitede bir yazılım elde etmek için kullanılabilecek test, muayene, tasarım analizi, gözden geçirme gibi birçok aktiviteyi içerisinde barındırır. Burada izlenen süreç genel olarak objektiftir ve genel olarak bu süreçteki aktivitelerde de bu amaçlanır.  Bunun ile birlikte doğrulama, geçerlemeye göre oldukça subjektif olabilmektedir. Çünkü burada önemli olan ortaya çıkan yazılım veya ürünün son kullanıcı veya müşterilerin veya ihtiyaç duyulan isterlerin ne kadar sağladığına bakılır. Bu noktada da kullanıcı değerlendirmesi her zaman son kararda etkilidir. Biz kendi projemizde doğrulama standardı olarak 1012-2016 - Sistem, Yazılım ve Donanım Doğrulama ve Doğrulama için IEEE Standardı'nı kullanacağız.



**6.2. Sınama Kavramları**

Sınama ve Bütünleştirme işlemlerinin bir strateji içinde gerçekleştirilmesi, planlanması ve tekniklerinin seçilmesi gerekmektedir.

* Sınama işlemleri dört ana sınıfta incelenebilir:
  + - Birim sınama
    - Alt-sistem sınama
    - Sistem sınama
    - Kabul sınaması

**Birim Sınama**

* Bağlı oldukları diğer sistem unsurlarından tümüyle soyutlanmış olarak birimlerin doğru çalışmalarının belirlenmesi amacıyla yapılır.

**Alt-sistem Sınama**

* Alt-sistemler modüllerin bütünleştirilmeleri ile ortaya çıkarlar.
* Yine bağımsız olarak sınamaları yapılmalıdır.
* Bu aşamada en çok hata ara yüzlerde bulunmaktadır. Bu yüzden ara yüz hatalarına doğru yoğunlaşılmalıdır .

**Sistem Sınaması**

* Üst düzeyde, bileşenlerin sistem ile olan etkileşiminde çıkacak hatalar aranmaktadır.
* Ayrıca, belirtilen ihtiyaçların doğru yorumlandıkları da sınanmalıdır.

**Kabul Sınaması**

* Çalıştırılmadan önce sistemin son sınamasıdır.
* Artık, yapay veriler yerine gerçek veriler kullanılır.
* Bu sınama türü alfa sınaması veya beta sınaması olarak ta bilinir.

**6.3.Sınama Yöntemleri**

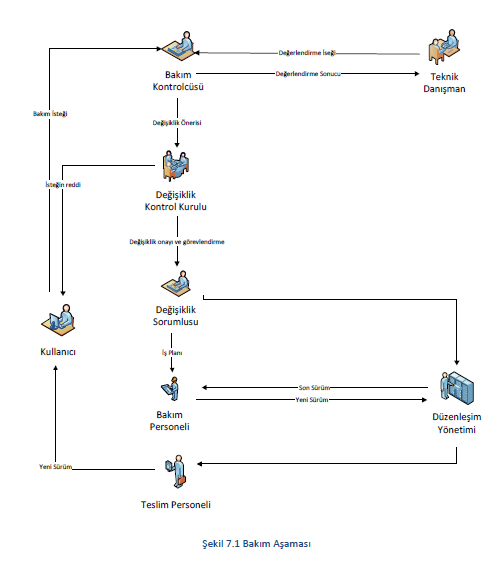
* Her yazılım Mühendisliği ürünü iki yoldan sınanır:
  + **Kara kutu testi (Black-Box testing ):** Sistemin tümüne yönelik işlevlerin doğru yürütüldüğünün testidir. Sistem şartnamesinin gerekleri incelenir.
  + **Beyaz Kutu Testi (White Box testing ):** İç işlemlerin belirtimlere uygun olarak yürütüldüğünün bileşenler tabanında sınanmasıdır.

|  |  |  |
| --- | --- | --- |
| Kriter | Kara Kutu Testleri | Beyaz Kutu Testleri |
| Tanım | Kara kutu testleri yazılımın iç yapısı,tasarım ve gerçekleştirme detayı bilinmeden testlerin tasarlandığı bir yazılım test tasarım tekniğidir. | Beyaz kutu testleri ,yazılımın iç yapısı,tasarım ve gerçekleştirme detayı bilinerekten testlerin tasarlandığı bir yazılım test tekniğidir. |
| Test Seviyesi | Kara kutu testleri, Kullanıcı Kabul Testleri(UAT) ,Sistem Testleri ve Yazılım Testleri seviyelerinde kullanılabilir. | Beyaz kutu testleri ise daha alt seviyelerde olan birim ve entegrasyon test seviyelerinde kullanılmaktadır. |
| Kim Yapar ? | Genellikle yazılım test uzmanları tarafından gerçekleştirilir. | Genellikle, yazılım geliştiriciler tarafından gerçekleştirilir. Gerekli durumlarda testlerin tasarımında ve test yazılımlarının kullanımında test uzmanları yazılım geliştiricilere destek verebilir. |
| Kodlama Bilgisi | Gerekli değildir. | Mutlaka gerekir. |
| Test Tanımı için Gerekenler | Gereksinim belirtimleri | Detaylı tasarım |

**7.BAKIM**

**7.1 Giriş**

Yazılım bakım projeleri kapsamında IEEE 12207 standardına uygun olarak Artırımsal Geliştirme Süreç Modeli yaklaşımı kullanılmaktadır. Artırımsal Geliştirme Süreç Modeli, bakım hizmeti alan müşterinin yazılım bakım sürecinde daha etkin yer almasına olanak vermektedir. Müşteri, sistem gereklerinin yerine getirildiğini ara fazlarla görebilmekte, varsa değişiklik önerisini ve açıklanacak noktaları proje bakım sürecinde bildirebilmektedir. Zaman içerisinde değişim ihtiyacı duyulmayacak bir yazılım sistemi düşünülemez. Kullanıcı ya da müşterilerin ihtiyaçlarındaki değişimlerin sisteme yansıtılması gerekir. Ayrıca, yeni bir donanım ya da yazılım altyapısı nedeniyle sistemin çalışma koşulları değişebilir. Tabi ki testler sırasında fark edilmeyen hatalar tespit edilebilir ve giderilmesi gerekir. Yazılımın dağıtılması ve kullanıma başlanmasından sonra yazılımda yapılacak değişiklikler yazılımın bakımı (software maintenance) olarak adlandırılır. Bu değişiklikler basit kodlama hatalarının düzeltilmesi (bug-fixes) şeklinde olabileceği gibi tasarımdan kaynaklanan hataların giderilmesi gibi daha kapsamlı değişiklikler şeklinde de olabilir.



**7.2 Kurulum**

Sistem kurulumuna değinmek gerekirse devlet güvencesinde verilecek olan serverlara yüklenecek olan sistemimizde FTP arayüzü ile dosyaları servera aktaracağız ve internet explorer olan tüm cihazlarda çalışacak

Android uygulama olan sistemimizde uygulama google play store 'a yüklenecek ve kullanıcılar tarafından buradan ücretsiz olarak indirilebilecektir.

**7.3 Yerinde Destek Organizasyonu**

Bu konuyla ilgili pilot bölgede bizzat desteği ben vereceğim bunun yanında sistem canlandırılıp gerçeğe geçirilirse sistem tanımlaması kurulum için bölgelerde bayilik sistemi gibi alt kuruluşlara yetki verilecek eğer profesyonel destek istenirse yol uçak masrafını karşılamak şartıyla bölgeye yetkili gönderilip orada bir organizasyon yapılacaktır.

**7.4 Yazılım Bakımı**

Sınama işlemleri bitirilen yazılımın kullanıcı ortamına yüklenmesi ve uygulamanın başlatılması gerekmektedir. Kullanıcı ortamı sunucu, desktop veya mobil platformalar olabilir. Yazılım devreye alındıktan sonra, yaşam döngüsünün en önemli ve hiç bitmeyecek aşaması olan **“bakım”** aşaması başlar.

Müşteriye teslim edilmiş ve devreye alınmış çalışmakta olan yazılımın üç tür bakım gereksinimi bulunmaktadır.

**1-Düzeltici bakım**

Bir yazılımın 100% sınanabilmesi teorik olarak mümkün olsa da pratikte pek mümkün değildir. Bu nedenle hata ile karşılaşabilme olasılığı her zaman vardır. Zamanla ortaya çıkan hataların düzeltilmesi “düzeltici bakım” olarak adlandırılır.

**2-Uyarlayıcı bakım**

Uygulamaya alınmış yazılımlar kurumların, şirketleri veya kişilerin günlük hayattaki işlerini bilgisayar ortamında takip etmesini sağlayan araçlardır. Gün geçtikçe iş süreçlerinde yeni gereksinimler veya var olan gereksinimlerin iptali söz konusu olabilmektedir. Örneğin mevzuatların değişmesi, bir kurum yazılımında ilgili bölümün değişmesi anlamına gelmektedir. Bu tür uyarlamalar yazılımda “uyarlayıcı bakım” olarak adlandırılır.

**3-En iyileyici bakım**

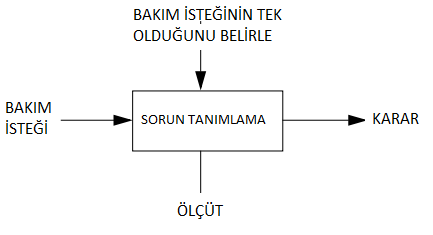
Uygulama yazılımının performansının zamanla arttırılması gerekebilir. Bu tür bakımlar “en iyileyici bakım” olarak adlandırılır.

Yazılım bakımı, uluslararası standart belirleme organizasyonu olan [IEEE](http://standards.ieee.org/)(**i triple e**) tarafından belirli kriterlere ve aşamalara göre gerçekleştirilmektedir. Bakıma ilişkin standart [IEEE 1219-1998](https://standards.ieee.org/findstds/standard/1219-1998.html) baz alınarak gerçekleştirilmektedir. IEEE tarafından sunulan bakım süreci şu adımları içermektedir:

1. Sorun tanımlama süreci
2. Çözümleme süreci
3. Tasarım süreci
4. Gerçekleştirim süreci
5. Sistem test süreci
6. Kabul test süreci
7. Kurulum süreci

Bu adımlar aslında yazılım geliştirme yaşam döngüsünün çekirdek adımlarının tekrarlanması şeklindedir. Ancak tekrarlanan kısım sadece değişiklik isteklerinin mevcut koda aktarılması amacıyla yapılmaktadır.

# **1- Sorun Tanımlama Süreci**



**Girdi**: Sürecinin temel girdisi, “bakım isteği” şeklindedir. Örneğin;

* Sistemde beklenen ve yeni düzenlemelere ilişkin değişiklikler.
* Yeni fonksiyonel talepler.

**İşlem/Süreç:** Bakım isteği oluşturulduğunda yapılması gereken işlemlerdir.

1. Değişiklik isteğine bir tanım numarası atamak.
2. Değişiklik türünü belirlemek.
3. Değişiklik isteğinin kabul edilmesi ya da ayrıntılı incelenmesine karar verilmesi.
4. Değişiklik isteği ile ilgili zaman/boyut/işgücü kestirimi yapılması.
5. Değişiklik isteğinin önceliklendirilmesi.
6. Değişiklik isteğinin diğerleri ile birlikte zaman ve iş planına kaydedilmesi.

Bu adımların uygulanmasına değişiklik talebinde bulunan müşteri, kullanıcı temsilcileri, yazılım mühendisleri ve iş uzmanları ile birlikte çalışılarak karar verilir.

**Denetim**: Sorun tanımlama aşamasında ise değişiklik talebinin daha önceden yapılıp yapılmadığı denetlenir ve tek olduğu belirlenir. Mükerrer iş yapmaktan kaçınmak adına daha önceki değişiklik talepleri taranır.

**Çıktı**: Doğrulanmış, geçerlenmiş ve karar verilmiş “bakım İsteğidir”. Bakım isteğinin ayrıntıları bir veritabanında saklanır ve isteğe ait bilgiler şu şekilde olmalıdır:

1. Sorun ya da yeni gereksinimin tanımı,
2. Sorun ya da gereksinimin değerlendirmesi,
3. Başlangıç önceliği,
4. Geçerleme verisi (Düzeltici bakım için gereklidir),
5. Başlangıç kaynak gereksinimi,
6. Mevcut ve gelecekte kullanıcılar üzerindeki etkileri,
7. Yararlı ve aksak yönleri

**Ölçüt**: Sorun tanımlama sırasında kullanılabilecek ölçütler

* Bakım taleplerinde kabul edilmeyen madde sayısı,
* Gelen bakım istekleri sayısı,
* Sorunun aşılması için harcanan kaynak ve zaman biçimindedir.

Bu adımların tamamlanmasıyla birlikte çözüm süreci başlatılabilir.

# **2- Çözümleme Süreci**

Çözümleme sürecinde, veri tabanında saklanmış ve geçerlenmiş bakım isteği girdi olarak alınır, projeye ilişkin bilgi ve belgeleri kullanarak söz konusu isteğin yerine getirilmesi için gerekli genel plan yapılır.

**Girdi:** Çözümleme sürecinin girdileri:

1. Geçerlenmiş bakım isteği,
2. Başlangıç kaynak gereksinimleri ve diğer veriler ve
3. Mevcut proje yada sistem bilgi ve belgeleri biçimindedir.

**İşlem/Süreç:** Çözümleme süreci temel olarak iki aşamadan oluşur. Bunlar olurluk aşaması ve ayrıntılı çözümleme aşamasıdır.

Olurluk çalışmasında, yapılan değişikliğin etkileri, güvenlik ve emniyet zorunlulukları, insan faktörleri, kısa ve uzun vadeli maliyetler ve yapılacak olan değişikliğin yararları değerlendirilir.

Ayrıntılı çözüm aşamasında, değişiklik isteği için ayrıntılı gereksinim tanımlaması yapılır. Bu çalışmada etkilenen yazılım öğeleri (yazılım tanımları, yazılım gereksinimleri, tasarım, kod, vb)belirlenir. Yazılım bileşenlerinin değişmesi gereken kısımları belirlenir. Bu aşamada en az üç düzeyli test stratejisi (birim testleri, bütünleştirme testleri ve kabul testleri) oluşturulur. Bu aşamada kullanıcıya en az etki yapacak şekilde değişiklik gereksinimlerinin nasıl karşılanacağı bilgilerini içeren **“Başlangıç gerçekleştirim planı”** da hazırlanır.

**Denetim**: Çözümleme çalışmasının denetiminde gerçekleştirilen işlemler şu şekildedir.

1. Gerekli proje yada sistem bilgi/belgelerine erişimin sağlanması.
2. Önerilen değişikliklerin  ve çözümleme çalışmasının teknik ve ekonomik olurluğunun gözden geçirilmesi,
3. Güvenlik ve emniyet konularının tanımlanması,
4. Önerilen değişikliğin, mevcut yazılımla bütünleştirilmesinin dikkate alınması,
5. Proje belgelerinin düzgün olarak günlendiğinin denetimi,
6. Çözümleme belgelerinin düzgün olarak hazırlanmasının sağlanması,
7. Sınama stratejilerinin uygun olarak belirlenmesi.

**Çıktı:** Çözümleme çalışmasının çıktıları:

* Değişiklik isteklerine ilişkin olurluk çalışması,
* Ayrıntılı çözümleme raporu,
* İzlenebilirlik listesini içeren günlenmiş gereksinim tanımları,
* Başlangıç değişiklik listesi,
* Sınama stratejisi,
* Gerçekleştirim planı şeklindedir.

**Ölçüt:** Çözümleme çalışmasında kullanılabilecek ölçütler:

1. Gereksinimlerdeki değişiklik sayısı,
2. Belgeleme hata oranı,
3. Her işlev alanı için gerekecek işgücü,
4. Toplam zaman biçimindedir.

# **3- Tasarım Süreci**

Tasarım aşamasında, değişiklikten etkilenebilecek tüm proje bilgi ve belgeleri üzerinde çalışma yapılıp söz konusu bilgi ve belgeler değişiklikle ilgili olarak güncellenir.

**Girdi:** Tasarım çalışmasının girdileri:

1. Çözümleme çalışması çıktıları,
2. Ayrıntılı çözümleme,
3. Güncellenmiş gereksinim tanımları,
4. Başlangıç değişiklik listesi,
5. Sınama stratejisi,
6. Gerçekleştirim planı,
7. Sistem ve proje belgeleri ve
8. Var olan kaynak kodları ve veritabanları biçimindedir.

**İşlem/Süreç:** Tasarım için gerekli temel işlemler aşağıda belirtilmektedir.

1. Etkilenen yazılım modüllerinin tanımlanması,
2. Yazılım modül belgelerinin değiştirilmesi,
3. Yeni tasarım için, güvenlik ve emniyet konularını da içeren test senaryolarının hazırlanması,
4. İlişki testleri tanımlanması,
5. Kullanıcı belgelerinin güncelleme gereksinimlerinin tanımlanması,
6. Değişiklik listesinin güncellenmesi şeklindedir.

**Denetim:** Tasarımın belirlenen standartlara uygunluğunun denetlenmesi gerekmektedir.

**Çıktı:** Bakım tasarımı çalışmasının çıktıları:

1. Gözden geçirilmiş değişiklik listesi,
2. Güncellenmiş tasarım
3. Güncellenmiş test planları,
4. Güncellenmiş ayrıntılı çözümleme,
5. Güncellenmiş gereksinimler,
6. Gözden geçirilmiş gerçekleştirim planı,
7. Risk ve kısıtlar listesi biçimindedir.

**Ölçüt:** Tasarım çalışması için kullanılabilecek ölçütler aşağıda verilmektedir.

* Yazılım karmaşıklığı,
* Tasarım değişiklikleri
* Her işlev alanı için gerekecek işgücü,
* Toplam zaman,
* Sınama yönerge ve plan değişiklikleri,
* Önceliklendirmedeki hata oranları,
* Varolan kodda, eklenen, çıkarılan ve değiştirilen satır sayısı,
* Uygulama sayısı.

# **4- Gerçekleştirim Süreci**

Gerçekleştirim sürecinde, temel olarak tasarım çıktılarını ve kaynak kodları girdi olarak alınmakta ve değişiklik isteğini gerçekleştiren kod parçaları ile güncellenmiş yazılım kodları üretilmektedir.  Güncellenmiş yazılıma ilişkin test bilgi ve belgelerinin ve eğitim belgelerinin üretimi de bu süreçte yapılmaktadır.

**Girdi:** Gerçekleştirim sürecinin girdileri:

1. Tasarım çalışması sonuçları,
2. Varolan kaynak kodlar, açıklamalar, belgeler ve
3. Proje ve sistem belgeleri biçimindedir.

**İşlem/Süreç:** Gerçekleştirim sürecinin dört ana işlemi vardır:

1. Kodlama ve birim testleri
2. Bütünleştirme
3. Risk çözümleme
4. Sınama hazırlığı gözden geçirme

Kodlama işleminde, değişiklik isteğini karşılayan yazılım kodları, varolan yazılıma eklenmektedir. İşlem sonucunda elde edilen yeni, değişmiş modüllere birim testleri uygulanmaktadır.  Birim test işlemini, bütünleştirme testleri izlemekte, tüm sistem yeniden test edilmektedir. Uygulamadaki riskleri gidermek amacıyla, gerçekleştirim aşamasında sürekli risk çözümleme yapılmaktadır.

**Denetim:** Gerçekleştirim sürecinde oluşturulacak denetim yapısı, aşağıdaki özellikleri sağlamalıdır:

* Belirlenen standartlara uygun olarak kod ve yazılım gözden geçirmeleri yapılması,
* Birim ve bütünleştirme testleri ile ilgili bilgilerin derlenmesi ve kaydedilmesinin sağlanması,
* Test belgelerinin güncellenmesi ve oluşturulmasının sağlanması,
* Test hazırlıklarının gözden geçirilmesi sırasında risk çözümlemenin yapılması,
* Yeni yazılımın, yazılım ortam yönetimi altında kaydedilmesi ve denetlenmesinin sağlanması,
* Teknik ve eğitim belgelerinin güncellenmesi,

**Çıktı:** Gerçekleştirim süreci aşağıdaki çıktıları vermelidir:

1. Güncellenmiş Yazılım,
2. Güncellenmiş tasarım bilgi/belgeleri,
3. Güncellenmiş sınama belgeleri,
4. Güncellenmiş kullanıcı belgeleri,
5. Güncellenmiş eğitim kılavuzları,
6. Riskler ve kullanıcılara etkileri,
7. Sınama hazırlığı gözden geçirme raporu

**Ölçüt:** Gerçekleştirim çalışmasında kullanılabilecek ölçütler kodlama ile ilgili olması gerekmektedir. Bu nedenle teknik ölçütler kullanılmaktadır:

1. Değişiklik oranı
2. Hata oranı

# **5- Sistem Test Süreci**

Değişikliklerin mevcut yazılıma yansıtılmasından sonra elde edilen yeni yazılım sürümünün belirlenen standartlara uygun olarak tümüyle bütünleşik sistem üzerinde testlerin yapılması gerekmektedir.  Sistem testlerinin, kullanıcı ve üretici ekiplerin tanıklığında bağımsız bir yapı tarafından gerçekleştirilmeleri önerilmektedir.

**Girdi:**Sistem test sürecinin girdileri:

1. Test hazırlık raporu
2. Belgeler
3. Sistem test planları,
4. Sistem testleri,
5. Sistem test yönergeleri,
6. Kullanıcı kılavuzları,
7. Tasarım
8. Güncellenmiş sistem biçimindedir.

**İşlem/Süreç:** Sistem testleri, tümüyle bütünleşik bir sistem üzerinde yapılmalıdır. Bu aşamada, işlevsel sistem testi, arayüz testi, regresyon testi ve test hazırlık raporunun gözden geçirilmesi işlemleri yapılır.

**Denetim:** Sistem testleri, üretici ve kullanıcılardan bağımsız bir grup tarafından gerçekleştirilmelidir. Yazılım kodları ve her türlü bilgi belge, yazılım ortam yönetimi tarafından saklanır.

**Çıktı:** Bu aşamanın temel çıktıları:

1. Tümüyle test edilmiş bütünleşik bir sistem,
2. Test raporu,
3. Gözden geçirilmiş test hazırlık raporu şeklindedir.

**Ölçüt:** Bu aşamada kullanılabilecek ölçütler, üretilen ve düzeltilen hata oranlarıdır.

# **6- Kabul Test Süreci**

Kabul test süreci, kullanıcılar ya da kullanıcı temsilcileri tarafından gerçekleştirilen bir süreçtir.  Kullanıcıların, değişiklikleri içeren yeni yazılımı test etmeleri ve kabul etmeleri beklenmektedir.

**Girdi:** Kabul test sürecinin girdileri:

1. Gözden geçirilmiş test hazırlık raporu,
2. Tümüyle bütünleşik sistem,
3. Kabul test planları,
4. Kabul testleri ve
5. Kabul test yönergeleri biçimindedir.

**İşlem/Süreç:** Kabul test işlemleri:

1. İşlevsel kabul testlerinin yapılması,
2. Birlikte çalışabilirlik testleri,
3. Regresyon testleri biçimindedir.

**Denetim:** Kabul testleri sırasında denetimi aşağıdaki işlemleri içermektedir.

1. Kabul testlerinin uygulanması,
2. Sınama sonuçlarının raporlanması,
3. İşlevsel denetim yapılması,
4. Yeni sistemin oluşturulması,
5. Kabul test belgelerinin yazılım konfigürasyonuna yerleştirilmesi.

**Çıktı:** Kabul testlerinin çıktıları, yeni sistem, işlevsel konfigürasyon denetim raporu ve kabul  sınaması raporudur.

**Ölçüt:** Bu aşamada kullanılabilecek ölçütler: üretilen ve düzeltilen hata oranlarıdır.

# **6- Kurulum Süreci**

Kurulum süreci, geliştirilen ya da  değiştirilmiş yeni yazılım sürümünün, uygulama sahasına aktarılma işlemlerini içerir.

**Girdi:** Bu sürecin temel girdisi, tümüyle sınanmış ve kabul edilmiş yeni yazılım sürümüdür.  
İşlem/Süreç: Bu sürecin işlemleri:

* Fiziksel ortam denetiminin yapılması,
* Kullanıcıların bilgilendirilmesi,
* Mevcut sistemin yedeklerinin alınması,
* Kullanıcı tarafında kurulum ve eğitimlerin yapılması şeklindedir.

**Denetim:** Denetim işlemleri:

1. Fiziksel ortam denetiminin yapılması,
2. Sistem ile ilgili bilgi ve belgelerin kullanıcıya ulaştırılması,
3. Sürüm Tanımlama raporunun tamamlanması,
4. Yazılım konfigürasyon ortamına aktarımın sağlanması şeklindedir.

**Çıktı:** Bu sürecin temel çıktıları, fiziksel ortam denetim raporu ve Sürüm tanımlama raporudur.

**Ölçüt:** Bu süreçte kullanılabilecek ölçüt, belgeleme değişiklikleridir.

Tüm süreçlerin tamamlanmasıyla yazılım bakım döngüsü tamamlanmış ve müşteri istekleri eksiksiz yerine getirilmiş olmaktadır. Tekrar belirtmekte fayda var ki bu standartlar dizisi IEEE tarafından belirlenmiştir.

Süreç adımları IEEE 1219-1998 tarafından girdi, çıktı ve kontrol şeklinde belirlenmiştir.

8.SONUÇLAR

Sistem kullanıcısı sisteme kullanıcı adı ve şifre ile sisteme giriş yapmaktadır. Kullanıcı adı ve şifre doğru girildikten sonra telefona gelen doğrulama onay kodu ile sisteme giriş yapılmaktadır. Onay kodu ile giriş yapılması sistemin hacklenmesini önleyerek güvenlik katmanını bir katman daha artırmaktadır. Sisteme giriş yapan kişi yapacağı işlemi kolayca yapılabilir ve yönetilebilmektedir.

9.KAYNAKLAR

[1]. “IEEE Standard Glossary of Software Engineering Terminology”, IEEE Std 610.12-1990, December 1990, pp. 80-81

[2] <https://standards.ieee.org/standard/1012-2012.html>

[3]. <https://www.mobilhanem.com/kara-kutu-test-teknigi-ve-uygulanmasi/>

[4]. <https://www.mobilhanem.com/kara-kutu-test-teknigi-ve-uygulanmasi/>

[5] <http://www.bimetri.com/urunler/yazilim/ozel/yazilim-bakim-metodolojilerimiz/>

[6] <https://www.bayramucuncu.com/yazilimda-bakim-sureci-ve-ieee-1219-standardi/>

[7]. Örnek Proje Dokümantasyonu