

**FIRAT ÜNİVERSİTESİ**

TEKNOLOJİ FAKÜLTESİ

Yazılım Mühendisliği Bölümü

**YMH319 – Programlama Dilleri**

**Proje Dokümantasyonu**

**Proje Ekibi**

**İsmet KALAYCI 16541061**

**Şükrü Furkan CEYLAN 16542018**

**PROGRAMLAMA DİLİ ÖĞRETİCİSİ**

**Ocak – 2021**

1

1. GİRİŞ

1.1 Projenin Amacı

1.2 Projenin Kapsamı

1.3 Tanımlamalar ve Kısaltmalar

2. PROJE PLANI

2.1 Giriş

2.2 Projenin Plan Kapsamı

2.3 Proje Zaman-İş Planı

2.4 Proje Ekip Yapısı

2.5 Önerilen Sistemin Teknik Tanımları

2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları

2.7 Proje Standartları, Yöntem ve Metodolojiler

2.8 Kalite Sağlama Planı

2.9 Konfigürasyon Yönetim Planı

2.10 Kaynak Yönetim Planı

2.11 Eğitim Planı

2.12 Test Planı

2.13 Bakım Planı

3. SİSTEM ÇÖZÜMLEME

3.1 Mevcut Sistem İncelemesi

3.1.1 Örgüt Yapısı

3.1.2 İşlevsel Model

3.1.3 Veri Modeli

3.1.4 Varolan Yazılım/Donanım Kaynakları

2

2

3.1.5 Varolan Sistemin Değerlendirilmesi

3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş

3.2.2 İşlevsel Model

3.2.3 Genel Bakış

3.2.4 Bilgi Sistemleri/Nesneler

3.2.5 Veri Modeli

3.2.6 Veri Sözlüğü

3.2.7 İşlevlerin Sıradüzeni

3.2.8 Başarım Gerekleri

3.3 Arayüz (Modül) Gerekleri

3.3.1 Yazılım Arayüzü

3.3.2 Kullanıcı Arayüzü

3.3.3 İletişim Arayüzü

3.3.4 Yönetim Arayüzü

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi

3.4.2 Eğitim Belgeleri

3.4.3 Kullanıcı El Kitapları

4. SİSTEM TASARIMI

**4.1 Genel Tasarım Bilgileri**

**4.1.1 Genel Sistem Tanımı**

**4.1.2 Varsayımlar ve Kısıtlamalar**

**4.1.3 Sistem Mimarisi**

**4.1.4 Dış Arabirimler**

3

**4.1.4.1 Kullanıcı Arabirimleri**

**4.1.4.2 Veri Arabirimleri**

**4.1.4.3 Diğer Sistemlerle Arabirimler**

**4.1.5 Veri Modeli**

**4.1.6 Testler**

**4.1.7 Performans**

**4.2 Veri Tasarımı**

**4.2.1 Tablo tanımları**

**4.2.3 Veri Tanımları**

**4.2.4 Değer Kümesi Tanımları**

**4.3 Süreç Tasarımı**

**4.3.1 Genel Tasarım**

**4.3.2 Modüller**

**4.3.2.1 Yönetici Modülü**

**4.3.2.1.1 İşlev**

**4.3.2.1.2 Kullanıcı Arabirimi**

**4.3.2.1.3 Modül Tanımı**

**4.3.2.1.4 Modül iç Tasarımı**

**4.3.2.2 Seçmen Modülü**

**4.3.2.2.1 İşlev**

**4.3.2.2.2 Kullanıcı Arabirimi**

**4.3.2.2.3 Modül Tanımı**

**4.3.2.2.4 Modül iç Tasarımı**

**4.3.3 Kullanıcı Profilleri**

**4.3.4 Entegrasyon ve Test Gereksinimleri**

4

**4.4 Ortak Alt Sistemlerin Tasarımı**

**4.4.1 Ortak Alt Sistemler**

**4.4.2 Modüller arası Ortak Veriler**

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

**4.4.4 Güvenlik Altsistemi**

**4.4.5 Veri Dağıtım Altsistemi**

**4.4.6 Yedekleme ve Arşivleme İşlemleri**

5. SİSTEM GERÇEKLEŞTİRİMİ

**5.1. Giriş**

**5.2. Yazılım Geliştirme Ortamları**

**5.2.1 Programlama Dilleri**

**5.2.2 Veri Tabanı Yönetim Sistemleri**

**5.2.2.1 VTYS Kullanımının Ek Yararları**

**5.2.2.2 Veri Modelleri**

**5.2.2.3 Şemalar**

**5.2.2.4 VTYS Mimarisi**

**5.2.2.5 Veritabanı Dilleri ve Arabirimleri**

**5.2.2.6 Veri Tabanı Sistem Ortamı**

**5.2.2.7 VTYS'nin Sınıflandırılması**

**5.2.2.8 Hazır Program Kütüphane Dosyaları**

**5.2.2.9 CASE Araç ve Ortamları**

**5.3. Kodlama Stili**

**5.3.1 Açıklama Satırları**

**5.3.2 Kod Biçimlemesi**

5

**5.3.3 Anlamlı İsimlendirme**

**5.3.4 Yapısal Programlama Yapıları**

**5.4. Program Karmaşıklığı**

**5.4.1 Programın Çizge Biçimine Dönüştürülmesi**

**5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama**

**5.5. Olağan Dışı Durum Çözümleme**

**5.5.1 Olağandışı Durum Tanımları**

**5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları**

**5.6. Kod Gözden Geçirme**

**5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi**

**5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular**

**5.6.2.1 Öbek Arayüzü**

**5.6.2.2 Giriş Açıklamaları**

**5.6.2.3 Veri Kullanımı**

**5.6.2.4 Öbeğin Düzenlenişi**

**5.6.2.5 Sunuş**

6. DOĞRULAMA VE GEÇERLEME

**6.1. Giriş**

**6.2. Sınama Kavramları**

**6.3. Doğrulama ve Geçerleme Yaşam Döngüsü**

**6.4. Sınama Yöntemleri**

**6.4.1 Beyaz Kutu Sınaması**

**6.4.2 Temel Yollar Sınaması**

**6.5.Sınama ve Bütünleştirme Stratejileri**

**6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme**

6

**6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme**

**6.6. Sınama Planlaması**

**6.7. Sınama Belirtimleri**

**6.8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri**

7. BAKIM

**7.1 Giriş**

**7.2 Kurulum**

**7.3 Yazılım Bakımı**

**7.3.1 Tanım**

**7.3.2 Bakım Süreç Model**

7. SONUÇ

8. KAYNAKLAR

7

**1.GİRİŞ**

**1.1 Projenin Amacı**

Bu projenin yapılma amacı gelişen ve değişen dünya şartları göz önüne alındığında yazılımın her geçen gün daha da önemli olduğu bilindiğine göre, yazılımın her yaştan insana eğlenceli ve kolay bir şekilde mantığını ve programlama dillerini öğretmektir.

**1.2 Projenin Kapsamı**

Proje kapsamında yazılımın temeli olan algoritma mantığı hakkında hiç fikir sahibi olmayan insandan, yazılım hakkında daha önceden fikir ve tecrübe sahibi olan insana kadar tüm kullanıcılara iyi bir rehber olması amaçlanmaktadır.

**1.3 Tanımlamalar ve Kısaltmalar**

VAD: Veri Akış Diyagramı

8

**2. PROJE PLANI**

**2.1 Giriş**

**2.2 Projenin Plan Kapsamı**

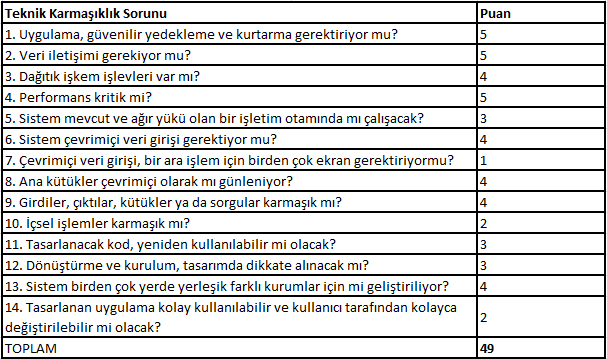
Projenin plan kapsamında mevcut sistemin olumlu ve olumsuz yönleri incelendi. Bu incelemelerden sonra öğrenmeyi arttıran renkler bal rengi ve turkuaz rengi kullanıldı, görsel ve sesli anlatım açısından eksiklikler giderildi. Karşılıklı diyalog ile öğrenme kapasitesi arttırıldı.

**Maliyet Kestirim Dokümanı**



**Şekil 2.1 Maliyet Kestirim Dokümanı 1**

9



**Şekil 2.2 Maliyet Kestirim Dokümanı 2**

0: Hiçbir Etkisi Yok

1: Çok Az etkisi var

2: Etkisi Var

3: Ortalama Etkisi Var

4: Önemli Etkisi Var

5: Mutlaka Olmalı, Kaçınılamaz

10

İN = AİN x (0.65 x 0.01 x TKF)

İN=192\*(0,65\*0,01\*49)

İN=61,152

Satır Sayısı = İN \*30

SATIR SAYISI= 61,152\*30 = 1834,56 SATIR ( YAKLAŞIK 1900 SATIR )

**Etkin Maliyet Modeli – COCOMO**

Organik proje: a=2,4 , b=1,05 ,c=2,5 , d= 0,38

Yarı – Gömülü Projeler İçin: a=3,0 , b=1,12 , c=2,5 , d= 0,35

Gömülü Projeler İçin: a=3,6 , b=1,20 , c=2,5 , d= 0,32

Öncelikle projemizin türünü belirlememiz gerekiyor. Küçük ekip tarafından geliştirildiği için organik projeler arasına giriyor.

Aylık Kişi Başı İş Gücü = E = a x (KSS)b

Geliştirme Süresi (Ay) = D = c x (E)d

Eleman Sayısı = E / D

Formülde verilen değişkenler şöyle:

KSS = Kod Satır Sayısı manasına gelmektedir ve birimi bin satırdır. Projenin tahmini kaç bin satırdan oluşacağını belirtmemizi sağlar.

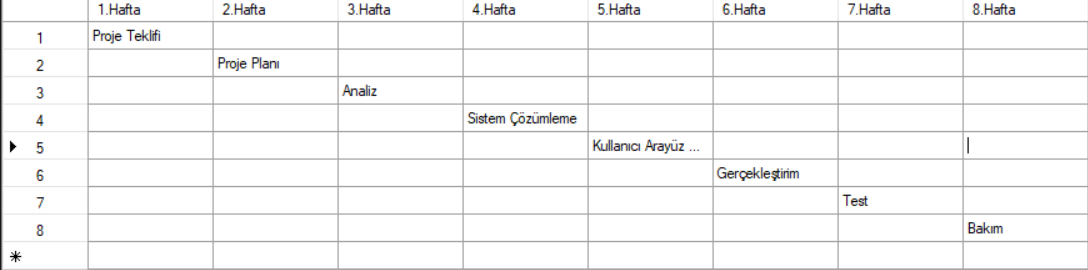
Aylık Kişi Başı İş Gücü = E= 2,4\*19001.05 =6651

Geliştirme Süresi =D= 7x66510.38 = 198

Tahmini Gerekli Personel Sayısı = 6651/ 198= 33 Kişi

11

**2.3 Proje Zaman-İş Planı**

****

**Şekil 2.3 İş Planı**

**2.4 Proje Ekip Yapısı**

Proje Yöneticisi: İsmet KALAYCI

Sistem Yöneticisi: Şükrü Furkan CEYLAN

Sistem Tasarımcısı: İsmet KALAYCI

Sistem Çözümleyicisi: İsmet KALAYCI

Veritabanı Yöneticisi: Şükrü Furkan CEYLAN

Kalite Uzmanı: Şükrü Furkan CEYLAN

12

**2.5 Önerilen Sistemin Teknik Tanımları**

**Kullanılan Teknoloji**

\* Açık Anahtarlı Altyapı Teknolojisi

\*Veritabanından sorular seçilir ve kullanıcıların karşısına çıkar.

**2.6 Kullanılan Özel Geliştirme Araçları ve Ortamları**

**Çözümleme ve Tasarım Araçları:** •Adobe Photoshop CC • Adobe Dreamweaver • Adobe Flash • Adobe After Effects.

**Programlama Araçları:** •PHP • MYSQL • Illustrator • HTML • JAVA • XML • Android Studio.

**Sınama Araçları:** • Microsoft Windows 10 • Iphone OS'X • Android.

**Destek Araçlar:** • Google Chrome • Mozilla Firefox • İnternet Explorer.

**2.7 Proje Standartları, Yöntem ve Metodolojiler**

Spiralin başladığı ilk çeğrek içinde ilk isterler toplanır ve buna göre proje planlaması yapılır. İkinci çeyrekte, ilk tanımlanan isterlere göre risk çözümlemesi yapılır. Üçüncü çeyrekte, risk çözümlemesi sonunda ortaya çıkan isterlerin tanımlanmasındaki belirsizlikleri ortadan kaldırmak için prototipleme yöntemi kullanılır. Gerekirse benzetim(simülasyon) veya diğer modelleme kullanılarak isterlerin daha sağlıklı tanımlanması sağlanır. Dördüncü çeyrekte, müşteri, ortaya çıkan ilk ürünü inceleyerek değerlendirme yapar, önerilerde bulunur. Bu şekilde tenımlanan ilk döngü bir sonraki döngü için bir girdi oluşturur.

13

**2.8 Kalite Sağlama Planı**

1.Ekonomi: Ekonomik açıdan yazılımın maliyeti her ne kadar ilk seferde pahalı olsa da ileriye dönük düşünüldüğünde ve zaman tasarrufundan ötürü gayet uygundur.

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: Admin1 sistemin tüm kısımlarına hakim olacak ve program bir bütün halinde çalışacaktır.

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. Örneğin: Yönetim modülü, Giriş Modülü…

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

14

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

10.Değiştirilebilirlik: Otomasyonun veri tabanını erişme yetkisi olan ve sistem hakkında bilgisi olan herkes sistemde değişiklik yapabilecek.

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

12.Genellik: Proje her üniversitede kullanılabileceğinden geneldir. Ve Türkiye genelinde kullanılacaktır.

13.Sınanabilirlik: Projedeki pilot bölge uygulaması sınana bilirliğinin göstergesidir.

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

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

15

**2.9 Konfigürasyon Yönetim Planı**

Sistemin ilerde kullanıcının yeni istemlerini karşılayamaması veya sistemin yapısındaki bazı bileşenlerin değişmesi sonucu güncelliğini kaybettiğinde olası konfigürasyon planı hazırlandı.

•Gününde düzenli olarak girmeyen kişiler için bir önceki aşamaya geri dönülmesi,

•Sistemde herhangi bir istenmeyen durum halinde,

Durumları için konfigürasyon yönetim planı oluşturuldu.

**2.10 Kaynak Yönetim Planı**

Mevcut bir kaynağımız olmadığından kaynak olarak sadece bu proje dokümantasyonu var.

**2.11 Eğitim Planı**

Projeden kazanılacak en önemli olaylardan biride eğitimdir. Kullanılacak dillerin arayüz editör ve programların kullanımında hakim olunamaması halinde bu program başarıyla neticelendirilemez. Bu yüzden projede bazı eğitimler alınması gereklidir. Proje kapsamında alınacak olan eğitimler;

• Adobe After Effects Kullanım Eğitimi

• Adobe Photoshop Kullanım Eğitim

• PHP Dil Eğitimi

• SQL Dil Eğitimi

Gereken eğitimlerdir.

16

**2.12 Test Planı**

Proje test ekipleri ve görevleri şu şekildedir;

•Site üzerinde bir kısım personel kullanıcı ve admin olarak görev yapacak.

• Gerekli resmi kısımlarda ise resmi olarak ekip yapıları yapılacak.

• Bu ekiplerin test aşamasında yapacağı işler aşağıdaki gibidir;

**2.13 Bakım Planı**

Projenin bakım planına gelecek hergün kullanılacak bu sistem tüm değişim ve bazı durumlarda kullanıcı eklenip çıkarılacak tüm bu sistemsel değişiklikler bakım planında yapılacaktır.

17

3.1 Mevcut Sistemlerin İncelemesi

Projemiz için örnek araştırırken ilk olarak Türkçe ve sade olmalarına dikkat ettik. Ardından Türkçe olan kaynaklar arasında tasarımlara önem verdik. Daha çok görsel açıdan hitap eden görsel olarak hafızada kalmaya yönelik tasarımlara sahip sistemleri inceledik.

İlk olarak incelemeye aldığımız site Busuu. Bu site karşımıza Türkçe içerik ile karşımıza çıkıyor. Kaç dilde eğitim verdiğini, göze güzel hitap eden bir şekilde belirtilmiş. Önemli nokta olarak kullanımı basit bir site tasarlanmış ki bu da herkese uygun olduğunu gösteriyor. Sitede aşağıya doğru indikçe mobil uygulama desteğini ve başarılarını belirtiyor. Öğrenmeye başla seçeneğini tıkladığımızda öğrenmek istediğimiz dil seçeneğini ileri sürüyor. Dilimizi seçtiğimizde ise kayıt ekranı ve daha kolay kaydolmak için Google ve Facebook ile giriş seçenekleri sunulmuş. Ardından ise ücretsiz ve ücretli seçenek sunuyor. Ücretli seçeneğinde bizlere daha çok paket içeriği veriyor. İmkânlarından ücretli yararlanabiliyoruz. Ücretli ücretsiz seçiminden sonra mobil uygulamadan mı yoksa site üzerinden mi devam edeceğimizi soruyor. Bu ekranda seviyemizi belirlememizi istiyor. Seçtiğimiz dile yeni başlıyorsak eğitime yönlendiriyor. İngilizceyi biraz bildiğimizi belirtiyor isek bir teste yönlendiriyor. Dil öğretim sayfasında ise kelimenin seçtiğimiz dildeki karşılığını Türkçe karşılığını ve seçilen dildeki seslendirmeleri ile bize gösteriyor. Ardından bize soru soruyor. Seçilen şıkkın yanlış olup olmadığını bize anında gösteriyor. Eğer yanlış işaretlersek bize ipucu olarak bilgilendirme yapıyor. Yanlış yaptığımız soruyu bir süre sonra tekrar karşımıza çıkarıyor. Bitişinde ise günlük başarıyı ilerleme seviyesini kaç kelime öğrenildiğini bize belirtiyor. Ardından öğrenilen kelimelerin nasıl kullanılacağı hakkında bir teste sokuyor. Duyulan kelime ile dinleme (listening) testi yapmamızı sağlıyor. Günlük hakkımız bitti ise devam etmek için ücretli seçeneğe geçmeyi sunuyor. Son olarak sitede en çok beğendiğimiz nokta ise siteyi nasıl kullanacağımız hakkında bize bir tur düzenlemeleri hoşumuza gitti. Bu da sitenin herkese uygun olmasını sağlıyor.

18

İkinci olarak Duolingo sitesinde kayıt ekranları Busuu ile aynı ilerliyor. Ek olarak günlük kaç dakika çalışacağımızı hedef olarak belirtmemizi istiyor. Sitenin kayıt ekranında ise üye olmadan da öğrenmeye başlayabiliyoruz. İlk test bitince tekrar hesap açma ekranı getiriyor yine hesap açmadan devam edebiliyoruz. Duolingo ücreti sadece reklamları kaldırmak için talep ediyor. Bunun dışında tamamen ücretsiz bir platform sunuyor bize. Duolingo forum sayfaları gibi bize tartışma platformu da sunarak diğer insanlarla iletişime geçmemizi sağlıyor bu sayede aklımıza takılan soruyu rahatça sorup ve cevabını alabilmemizi sağlıyor. Sesli sorularda müsait olmadığımızı belirtirsek bir saat sonra tekrar o soruyu karşımıza çıkarıyor.

Üçüncü sitemiz Memrise kayıt ekranları hepsinde benzer ilerliyor. Üye olmadan eğitime başlamamıza izin vermiyor. Bu sistemde de ücret talep ederek daha çok seçenekten faydalanma sunuyor. Duolingo gibi sesli sorularda erteleme hakkına sahip olabiliyoruz. Memrise bir kelimeyi yazarken altta ipucu harfler veriyor. Memrise akılda kalıcılık için aynı kelimeleri farklı şekillerde karşımıza getiriyor. Eğitim bitince ücretli eğitim fiyatlarını karşımıza çıkarıyor.

Verbling sitesinde videolu dersler veriliyor. Bir sürü öğretmen seçeneği sunuluyor. Bu öğretmenler hakkında bilgiler veriliyor ve o öğretmenden ders satın alıyoruz. Videolu eğitimler veriliyor. Bu sistem hakkında çok fazla araştırma yapamıyoruz. Ödeme bilgileri verilmeden hiçbir işlem yapmamıza izin vermiyor. Fakat ders saatleri öğretmenlerin yaşadığı ülkelerdeki saatlere uygun olduğu için bu bize pek de uygun olacak gibi görünmüyor.

Turkcell Akademi ve Türk Telekom Akademi de videolu eğitim platformuna sahip. Bunlarda üyelik mecburiyetine sahip ve derslerin çoğu ücretli olarak bizlere sunuluyor. Microsoft Akademi baştan sona ücretsiz bir platform fakat kategori seçeneği sadece yazılım alanında veriliyor. Videolu ve test içeriği olarak bir adım öne çıkıyor. Ücretsiz olması da bu platformu çekici kılıyor.

19

3.1.1 Örgüt Yapısı

Örgüt yapısı olarak bünyemizdeki bilgi sistem personellerinin ortak çalışmasıyla oluşan bir örgüt yapısı vardır.

3.1.2 İşlevsel Model

Sistemde ki en güzel özelliklerden birisi tüm öncelikle Elazığ Fırat Üniversitesi öğrencilerinin kullanması için yazılmıştır. Ama sistem daha da geliştirilerek tüm Türkiye geneline hatta tüm dünyaya yayılabilecek olmasıdır. Bu sistemi Fırat Üniversitesi öğrencileri kullanıcı adı ve şifresiyle otomatik olarak giriş yapabilecek. Ama zamanla sistem geliştirilerek lise öğrencileri başka üniversite öğrencileri, dışardan dil öğrenmek isteyen insanların kayıt olarak sistemi kullanmaları sağlanabilir.

3.1.3 Veri Modeli

Veri tabanı ilişkisel veri modelinde veriler tablolar üzerinden kurulan ilişkiye dayanmaktadır.

3.1.4 Varolan Sistemin Değerlendirilmesi

Var olan sistemler içinde en önemlisi Dualingo olduğundan Dualingonun değerlendirmesi yapılacaktır. Dualingo pek çok yönden gelişmiş bir yabancı dil öğreticisi programıdır. Fakat iyileştirilmesi gereken noktalar olduğunu düşünüyoruz. Bunlar; soruların bazen çok sık tekrar etmesi, belli bir seviyeden sonra programın artık kullanıcıyı geliştirmekten çok, rutine girmesi vb. gibi eksik noktalardır.

20

3.2 Gereksenen Sistemin Mantıksal Modeli

3.2.1 Giriş

Mevcut sistemleri incelediğimizde tüm sistemlerde para karşılığında hizmet ya da kısıtlanmış hizmet verildiğini görüyoruz.

3.2.2 İşlevsel Model

Sisteme giriş ----> Seviye Belirleme----->İsteğe göre Ders Dinleme------>Ders Başarımları

Sisteme giriş yapılacak kayıtlı ise giriş ekranı değil ise kayıt ekranına yönlendirecek. Ders seçimi yapılacak. Ardından seviye tespit sınavı yapılacak ardından eğitimlere başlanacak. Ardından ilerleme istatistikleri kullanıcıya gösterilecek.

3.2.3 Genel Bakış

Bu sistemde kullanıcının verilerine ulaşabilen kişiler dersin öğretmeni ve yöneticilerdir. Bu öğrencilerin eksiklerinin giderilmesi için kullanılabileceği düşünülmektedir.

3.2.4 Bilgi Sistemleri/Nesneler

Öğretmen: Öğretmenler ders içerikleri ve sınavları düzenleyen birinci nesnedir.

Kullanıcı: Sistemin ikinci nesnesidir. Sistemin en aktif kullanıcısı ve oluşmasını sağlayan nesnedir.

Arşiv: Dosyaların saklandığı nesnedir.

21

3.2.5 Veri Modeli

3.2.6 Veri Sözlüğü

Tc\_no = \*11 haneli integer değer tipinde bir sayıdan oluşur\*

İsim= \*String tipinde bir değerden oluşur\*

Soyisim= \*String tipinde bir değerden oluşur\*

Görev= \*String tipinde bir değerden oluşur o kişinin görevlerini içeren kelimeye sahiptir\*

Yetki= \*integer veri tipinde bir karakterdir ve kişinin yetkisine göre değişir.\*

Durum= \*Boolean veri tipindedir imza durumunu gösterir \*

Gorevlendirme= \*Kendi altındaki kişiyi görevlendirip görevlendirmediği değerini içeren boolen tipindeki veridir \*

Gorev= \*Kendi üstündeki kişi tarafından görevlendirilip görevlendirilmediği değerini içeren boolean tipindeki veridir\*

Sayım=\*imza atan kişinin imzasının doğru olduğunu gösteren veridir\*

Gorevlendirme= \* Kendi altındaki kişiyi görevlendirip görevlendirmediği değerini içeren boolen tipindeki veridir \*

Gorev= \*Kendi üstündeki kişi tarafından görevlendirilip görevlendirilmediği değerini içeren boolean tipindeki veridir\*

Sayım=\*İmza atan kişinin imzasının doğru olduğunu gösteren veridir\*

3.2.7 İşlevlerin Sıradüzeni

Veri tabanı Admini kullanıcıyı sisteme kayıt eder. Derecelendirme Adminleri kendilerine yapılan geri dönüşlerden çıkarımlar elde ederek bunları sistem yöneticisine iletir. Sistem yöneticisi analiz ve sistem güncelleme ekibine direktifler verir. Analiz ve sistem güncelleme ekibi sistem güncellemesini gerçekleştirir.

22

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 kolaylığı
* Anlaşılabilirlik
* Tarafsızlık
* Ücretsizlik

temel gereklilikler olarak tespit edilmiştir.

23

**3.3 Arayüz (Modül) Gerekleri**

3.3.1 Yazılım Arayüzü

Bu arayüz şu anda tasarlanma aşamasındadır.

3.3.2 Kullanıcı Arayüzü



**Şekil 3.1 Kullanıcı Arayüzü**

24

3.3.3 İletişim Arayüzü

Bu arayüz şu anda tasarlanma aşamasındadır.

3.3.4 Yönetim Arayüzü



Şekil 3.120 Yönetimsel Arayüz

3.4 Belgeleme Gerekleri

3.4.1 Geliştirme Sürecinin Belgelenmesi

Geliştirme sürecinde genel olarak belgelendirilmesi hem ileriye dönük hem de şimdi yapmaya çalıştığımız geliştirme sürecinde projenin sürekli geliştirilip tamamlama yüzdesini ortaya çıkarma ve daha üst seviyelerde kullanılabileceğinin araştırılıp nerelerde eksiklerin olduğunu genel hatlarıyla gösterilmesidir.

25

3.4.2 Eğitim Belgeleri

Aslında sistem eğitim amaçlı olduğundan sistem geliştirildiğinde bir eğitim belgesi alabilir. Ancak sistem de kursları tamamlayan kişilere sanal ortamda verilen bir eğitim başarı belgesi verilmesini düşündük.

26

**4.SİSTEM TASARIMI**

**4.1 Genel Tasarım Bilgileri**

**4.1.1 Genel Sistem Tanımı**

**Gereksinimler**

Programda gereksinim olarak programın kullanıcıya basit bir arayüz ile akıcı ve eğlenceli bir şekilde eğitimin sunulması amaçlanmaktadır. Sistem tasarımı gereksiz ayrıntılardan arındırılmış şekilde tasarlanmalıdır. Kullanıcının ilk oluşturduğu profil kayıt bilgileri hafızada tutularak sonraki girişlerde kullanıcıyı direkt sisteme yönlendirerek kullanıcıya kolaylık sağlamalıdır. Kullanıcıya yöneltilecek olan soru havuzu geniş olmalıdır. Soru tekrarlarından maksimum seviyede kaçınılması amaçlanmaktadır. Görseller ve ses öğeleri ile öğrenim daha akılda kalıcı bir şekilde ilerlemelidir.

**İşlevsel Belirtimler**

İlk olarak bu programın ne iş yapacağına cevap verelim. Neden bu programı yapıyoruz? Öncelikle küresel teknolojik değişim ve bu değişim ile şekillenen insani ihtiyaçlar göz önüne alındığında yazılımın insan hayatındaki büyük, küçük her türlü probleme pratik ve hızlı bir şekilde çözüm bulmasından dolayı yazılımın önemi tartışılmaz bir gerçektir. Hal böyle iken yazılımın gerçekleştirildiği ortam olan programlama dillerini öğrenmenin ve öğretmenin her geçen gün daha da önemi artmaktadır.

İşte burada sistem bu gereksinimleri karşılamak üzere tasarlanmalıdır. Gerçekleştirilecek sistem, programlamanın düşünülenin aksine basit düzey bir matematiksel zekâya sahip herhangi bir bireyin çok kolay bir şekilde öğreneceğini iddia etmektedir.

27

**4.1.2 Varsayımlar ve Kısıtlamalar**

Varsayılan değerler bulunmamaktadır. Kısıtlamalar ise şunlardır;

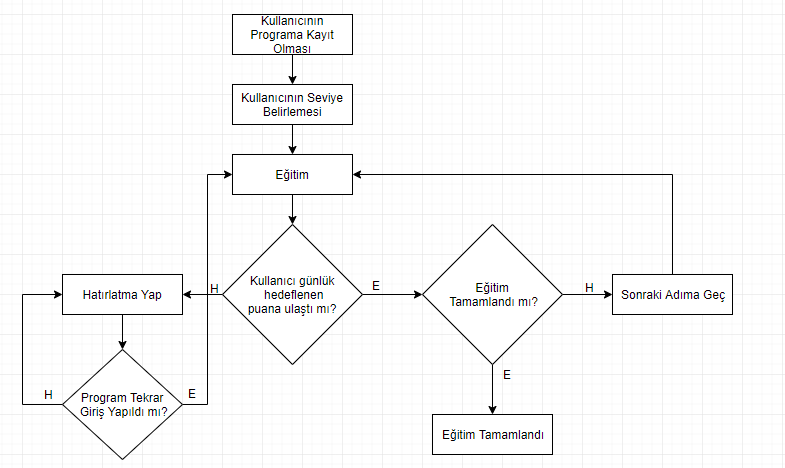
\* Kullanıcının, kullanıcı sözleşmesini kabul etmelidir.

\* Kullanıcının doğrulama kodu için bir telefon numarasına sahip olmalıdır.

\* Kullanıcının internet bağlantısı bulunmalıdır.

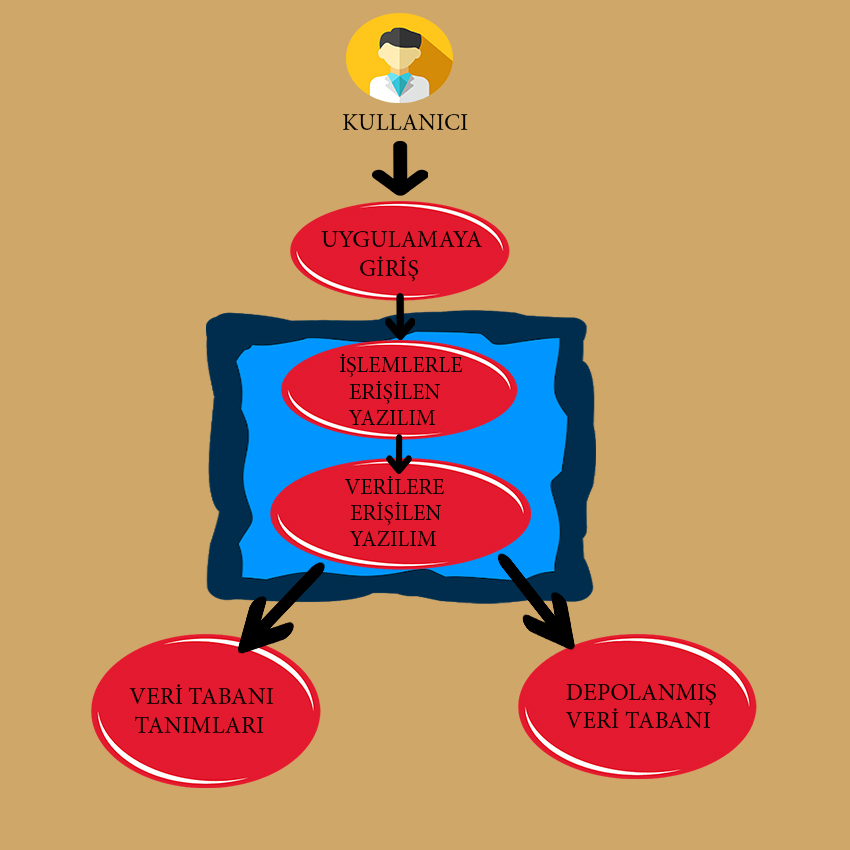
\* Kullanıcı dosya erişimine izin vermelidir.

**4.1.3 Sistem Mimarisi**



**Şekil 4.1 Genel Sistem Mimarisi**

28



**Şekil 4.2 Veri Tabanı Sistemi**

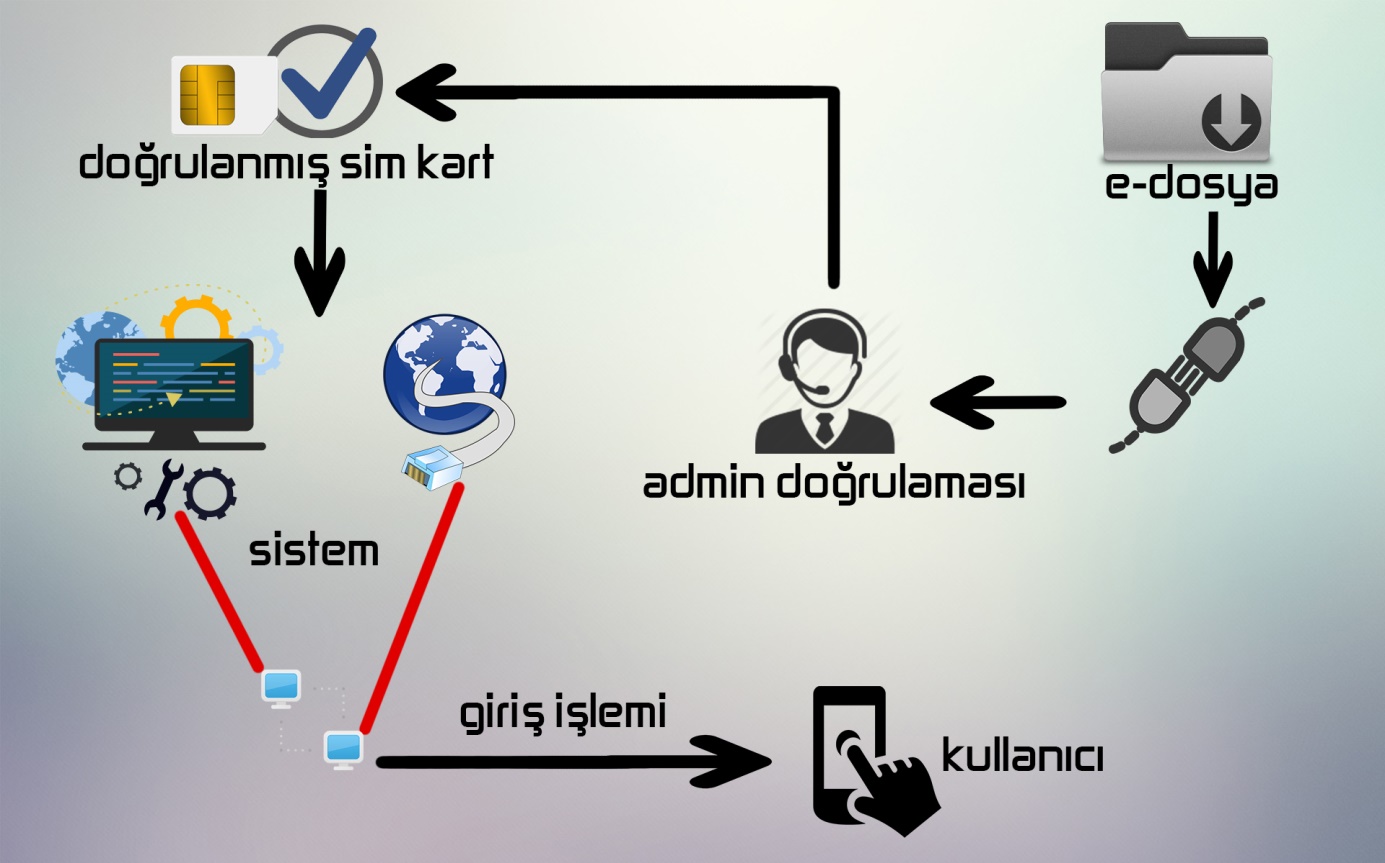
Sistemin mimarisinin akış diyagramı şeklinde verilmesinin temel nedeni sistemin işleyiş mantığının nasıl olduğu ve nasıl bir yol çizileceğinin bilinmesidir. Akış diyagramı sistemin temel mantığı hakkında bize fikir verecektir.

29

**4.1.4 Dış Arabirimler**

**4.1.4.1 Kullanıcı Arabirimleri**

Sisteme ilk girişte kullanıcıyı giriş ve kayıt olma ekranı karşılayacak. Kullanıcı oluşturduğu profil bilgileri sayesinde sisteme giriş yapabilecek.



**Şekil 4.3 Kullanıcı Arabirimi**

**4.1.4.2 Veri Arabirimleri**

Gerçekleştirilecek sistem, kullanıcının sisteme her giriş yaptığında eğitime kaldığı yerden devam edebileceği şekilde tasarlanmalıdır. Bunu sağlamak için SQL altyapısı ile oluşturulmuş veri tabanı dosyası oluşturulmalı ve kullanıcı işlemleri bu veri tabanı üzerinden yürütülmelidir.

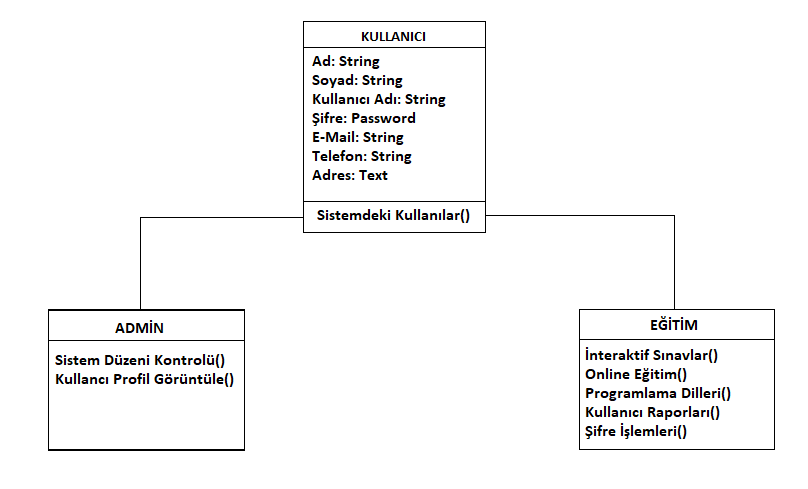
30

**4.1.4.3 Diğer Sistemlerle Arabirimler**

Herkesin bileceği üzere bilgisayarlar, telefonlar, tabletler vb. gibi teknolojik aletler artık hayatımızın her alanında yer almaktadır. Bundan dolayı bu proje gerçekleştirilirken bu unsurlar göz ardı edilmemelidir.

Gerçekleştirilmesi planlanan bu proje hem bilgisayar hem de mobil ortamlar üzerinden tasarlanacak olan sistemlerden oluşacaktır. Bunun sonucunda kullanıcılar dilekleri ortamda eğitimlerine devam etme imkânına sahip olacaklar.

**4.1.5 Veri Modeli**



**Şekil 4.4 Veri Modeli**

31

**4.1.6 Testler**

Bir yazılım projesi üzerinde çalışırken dikkate alınması gereken en önemli unsurlardan birisi gerçekleştirilecek olan yazılımın gerekli test aşamalarından geçmesidir.Çünkütest aşamasında bulunan hatalar ve kullanıcının geri dönüşleri sistem tasarlandıktan sonra oluşabilecek hataları en aza indirmememize yardımcı olur. Bundan dolayı bir yazılımda test aşamasının önemi oldukça fazladır.

Bu sistemde testler iki farklı şekilde gerçekleştirilecektir.

İlk yapılacak olan test, sistemin geliştirildiği yerde kullanıcılardan geri dönüşler alarak yapılmalıdır.

Yapılacak diğer testte ise kullanıcı, geliştirilen sistemi kendi yerleşkesinde, bir gözetmen eşliğinde yapacaktır.

**4.1.7 Performans**

Tasarlanacak sistemlerin kullanıcıya en iyi deneyimi sunması için sistemin performansı önemli bir faktördür. Tasarlanacak sistem gereksiz detaylardan kaçınılarak, abartılı görseller kullanılmadan, basit ayrıca şık görsellere sahip yazı fontları ve temalardan oluşmalıdır. Bu sayede arayüz geçişleri daha optimize bir şekilde çalışacak, sistem daha stabil olacak ve kullanıcıya en iyi deneyimi sunacaktır.

**4.2 Veri Tasarımı**

**4.2.1 Tablo tanımları**

Sistemde kullanıcıların kayıtlarının, eğitim bilgilerinin, programın içerisinde bulunan programlama dillerinin kayıtlarının tutulması gibi pek çok işlem için veri tabanı kullanılacaktır. Bu bilgiler tablolar aracılığı ile veri tabanına aktarılmalıdır.

32

**4.2.2 Veri Tanımları**

Sistemde genel olarak integer, string ve boolen veri tipine yer verilecek.

* İnteger veri tipi sayısal değerleri tutacaktır.
* String veri türü kelime içeren verileri tutacaktır.
* Boolen veri türü ise mantıksal öğeler içeren verileri tutacaktır.

**4.2.3 Değer Kümesi Tanımları**

**Sayım, Sonuç:** Kullanıcıların sisteme kayıtları doğrultusunda oluşacak olan değerdir.

**Yetki:** Sistem yöneticilerine verilen yetkiler, kişilerin akademik kariyer derecelerine ve iş hayatındaki kazanımlara göre değişen bir değerdir.

**4.3 Süreç Tasarımı**

**4.3.1 Genel Tasarım**

Gerçekleştirilecek olan sistemin genel tasarım planı şu şekilde yapılmalıdır.

İlk olarak sistemin en temel yapı taşı olan veri tabanı modeli oluşturulmalıdır. Ardından giriş modülleri ve yönetici modülleri oluşturulmalıdır. En son olarak sistemin belki de en can alıcı noktası olan ve kullanıcıya hitap eden kısım olan sistem arayüzü tasarlanmalıdır.

**4.3.2 Modüller**

**4.3.2.1 Giriş Modülü**

**4.3.2.1.1 İşlev**

Kullanıcının sisteme müdahale edebileceği ekrana erişebilmesi için aşması gereken bir modüldür. Bu modül ayrıca sistem üzerinde yedekleme ve arşivleme çalışmalarının yapılacağı kısım olmalıdır. Kullanıcının sistemi kullanırken yaşayacağı bir aksaklıkta karşılaşabileceği sorunların giderilmesi için bu modülün tasarımına ayrıca önem verilmelidir.

33

**4.3.2.1.2 Kullanıcı Arabirimi**



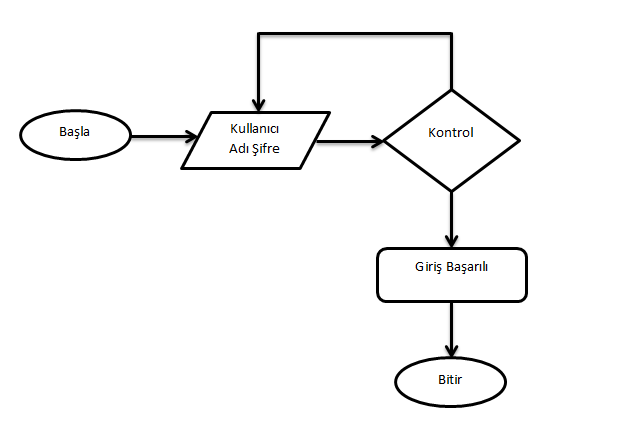
**Şekil 4.5 Kullanıcı Arabirimi**

34

**4.3.2.1.3 Modül Tanımı**

Sistem yöneticileri tarafından kullanıcıların bilgilerinin kontrol edildiği modüldür.

**4.3.2.1.4 Modül İç Tasarımı**



**Şekil 4.6 Modül İç Tasarımı**

**4.3.2.2 İşlev**

Sistem içindeki tüm arabirimler aslına bakacak olunursa yönetim modülüdür. Arayüzler tamamı ile birbirinin aynısı lakin işlevler farklıdır.

35

**4.3.3 Kullanıcı Profilleri**

**ADMİN 1:** Sistem üzerinde en yetkiye sahip olan kişidir. Sistemin bakımı, onarımı ve kontrolünden sorumludur.

**ADİMN2:** Kullanıcılardan sorumlu olan kişidir. Eğitim sürecinin zorluğunu en aza indirmek ve kullanıcılara daha iyi hizmet verebilmek adına sistemi güncel tutan kişi konumundadır.

**ADMİN 3:** Kullanıcı verilerinden sorumlu olan kişidir. Eğitim sürecinde yaşanabilecek olan sorunlara önlem olarak yedekleme ve arşivleme işlemlerini yapar.

**KULLANICI:** Eğitimi alacak olan kişi ve projedeki en önemli unsurdur.

**4.3.4 Entegrasyon ve Test Gereksinimleri**

Sistem tasarlanırken oluşturulacak olan veritabanları ile entegrasyon halinde olmalı ve bu şekilde yaşanacak veri kayıplarının önüne geçilmelidir. Ayrıca test süreçleri boyunca yazılımı tasarlayacak olan yetkililer test sürecine dahil olmalı ve sistem testini gözlemleyip raporlar çıkartmalıdır. Bu raporlar doğrultusunda sitemde yaşanacak sorunların önüne geçilmelidir.

**4.4.1 Ortak Alt Sistemler**

Kullanacağımız ortak sistemler kullanıcının kayıt gerçekleştirmesi sırasında onaylama işlemlerinin tamamlanması için kullanıcının sahip olduğu telefon hattının yada mail adresinin onay sistemi kullanılacaktır.

36

**4.4.2 Modüller arası Ortak Veriler**

Modüller arasında ilişkili veri modelinde olduğu üzere ortak veriler mevcuttur. Bunlar Telefon numarası yada Mail hesabı, Görevlendirme, Görev ve sonuç birbirleriyle ilişkili ortak verilerdir.

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

Ortak veriler için böyle bir modül kullanılmadı

**4.4.4 Güvenlik Altsistemi**

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.

1. Model seçme ve düzenlemeye yönelik faaliyetlerin temelinde uygun hedeflerin tespit edilmesi bulunmaktadır.

2. Hata ve aksaklıkların analiz edilmesi için uygun verilerin tanımlanması gerekmektedir. Örneğin, arıza veya hataları önemine göre sınıflandırmak, hatalar arası ortalama süreyi bulmak, hata nedenlerini araştırmak, hataları bulmaya yönelik test verilerine karar vermek.

3. Belirtilen hedeflere yönelik veriler modellenir.

4. Geçmişe yönelik verilerin zaman bilgilerini de içerecek şekilde elde edilerek yazılım geliştirme sürecine dâhil etmek.

5. Yazılım geliştirme sürecinin modellenmesi, hata ile karşılaşılıp, test sürecine başlamak ve model doğrulama işlemlerine gerçekleştirmek.

6. Güvenilirlik tahminleme modelinin seçilmesini sağlamak.

7. Güvenilirlik modeli tarafından kullanılacak olan parametrelerini tespit etmek.

8. Verilen bir noktayı kullanarak gelecekteki olası hatalar hakkında tahmin yapmak.

9. Tahmin edilen hata ve arıza oranları ile gerçekleşen değerleri kıyaslamak.

37

**4.4.5 Veri Dağıtım Altsistemi**

Bu başlık şu anda incelenme aşamasındadır.

**4.4.4** **Yedekleme ve Arşivleme İşlemleri**

Sistem üzerinde depolanan verilerin, herhangi bir nedenden dolayı zarar görmesi, sistemin çalışma süreçlerinde ciddi sorunlara yol açabilir. Bundan dolayı kullanıcılar çeşitli mağduriyetlerle karşılaşabilirler. Kullanıcı verileri veya sistem verilerinin zarar görmesi durumunda, depolanan verilerin geri yüklenememesi, kullanıcılara ve sistemi tasarlayan kurumlara veya kişilere çok ciddi maddi ve manevi zararlar verebilir.

Bu nedenle sistemin çalışma süreçlerine bağlı olarak, yedekleme sistemleri kurulmalıdır. Yedekleme işlemleri günlük olarak takip edilmelidir.

Ayrıca kullanıcının isteği dışında gerçekleşebilecek sistem kapanmaları vb. durumlara da önlemler alınmalı ve bu durumlarda da yedekleme işlemleri gerçekleştirilmelidir.

38

**5.1. Giriş**

**5.2. Yazılım Geliştirme Ortamları**

**5.2.1 Programlama Dilleri**

Geliştirilen uygulamanın ilk olarak mobil platformda gerçekleştirilmesi planlandığı için mobil sistem tasarımı oluşturmadaki avantajlarından dolayı programlama dili olarak Java, destekleyici sistemlerden erişim yürütücüleri olarak uygunluk ve güçlü yönelerinden dolayı PHP, SQL, XML ara yüz ve panel tasarımları için İllustrator motor olarak ise Android Studio tercih edilmiştir.

**5.2.2 Veri Tabanı Yönetim Sistemleri**

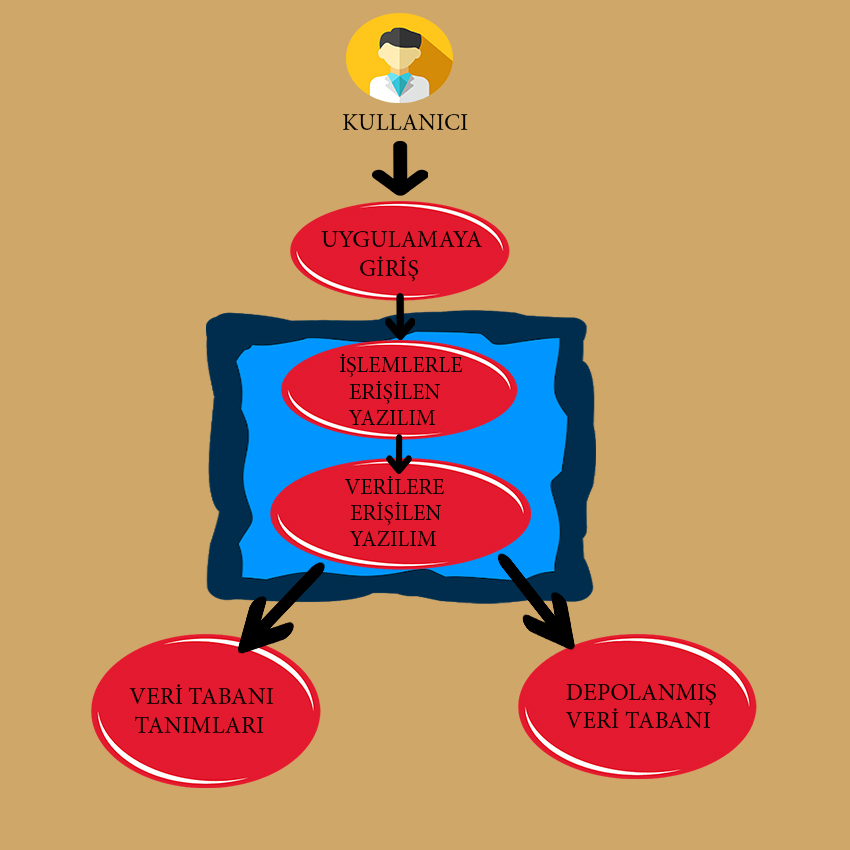
XML’in kullanıcı girişindeki sorgu ve java ilen kodlanan programı işleyen yazılım görevini yürütüyor.

PHP programın içindeki eğitim dokümanları ve soru havuzuna erişen yazılım görevini üstleniyor.

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

Veri tabanı yönetiminde kullandığımız hiyerarşiyi göstermek gerekirse:

39



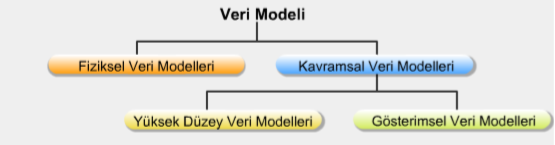
**Şekil 5.1 Veri Tabanı Yönetim Sistemi**

**5.2.2.1 VTYS Kullanımının Ek Yararları**

Genişleme potansiyeli, esneklik, geliştirme zamanının azalması, güncel bilgilerin tüm kullanıcılara aynı anda ulaşması, ölçmede ekonomi, yazılım geliştirmenin kolaylığı

40

**5.2.2.2 Veri Modelleri**

****

**Şekil 5.2 Veri Modelleri**

Fiziksel Veri Modelinde verilerin MySQL de tablolar içindeki alanlarda saklanacağı ve birbirleriyle ilişki içinde olduğunu söyleyebiliriz. Kavramsal Veri Modelini iki ana başlık altında inceleyebiliriz.

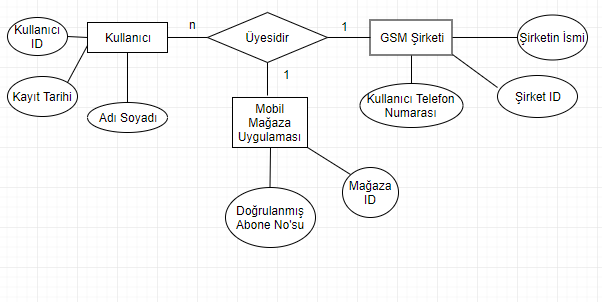
**1. Gereksinimlerin Belirlenmesi**

Programı kullanacak olan kişilerin telefon numaraları ad & soyad ve kullanıcı ID’leri ve şifreleri tutulması için 5 ayrı veri tutacak diziye ihtiyaç vardır. Bu verilerden Şifre kısmı ve telefon numarası private tanımlı olarak erişime görünüme saklı bir şekilde kayıt altında tutulurken diğer veriler başka kullanıcıların görünüme isteğe bağlı olarak açık bir şekilde görüntülenebilecektir. Veri tabanında tutulan telefon numarası ile telefon numarasının ait olduğu GSM şirketi aracılığı ile programa ait bir doğrulama kodu gönderilecektir kullanıcı doğrulama kodunu girdikten sonra tüm sistem hizmetleri kullanıcıya açılacaktır.

**2. Kavramsal Model**

Kullanıcıdan elde edilecek gereksinimler ile ilgili bir analiz çalışmasının yapılması ve birbiriyle bağlantılı verilerin gruplanarak bir düzenleme içinde modellenmesi gerekmektedir. Bu modeli grafiksel olarak varlık bağıntı seçenekleri ile şu şekilde gösterebiliriz:

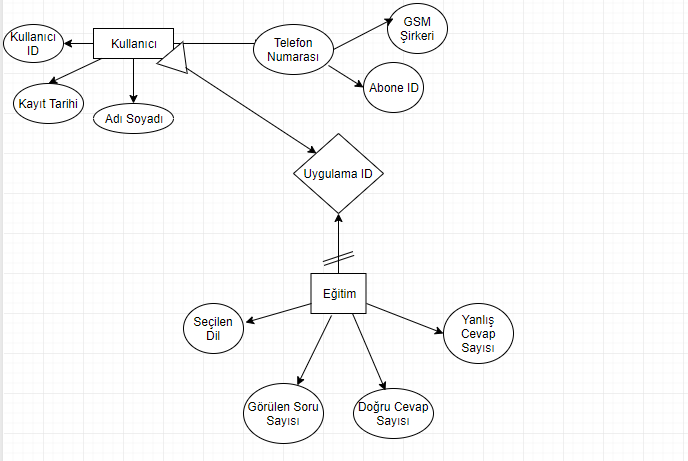
41



**Şekil 5.3 Kavramsal Model**

**3. Mantıksal Model**

Veri tabanı tasarımlarımızın ilişkisel veri tabanı modelinde tablolar ile ifade edilebilmesi için yapılması gereken dönüşümü içerir. Bu bir projenin kullanıcı mantıksal gösterimi açısından çok önemlidir. Mantıksal model ile kullanım süreç aşaması modellenir ve proje kullanım içeriği karmaşıklığı minimum seviyeye indirilir. Bunlar göz önüne alınarak şöyle bir mantıksal model yapısı oluşturulabilir:

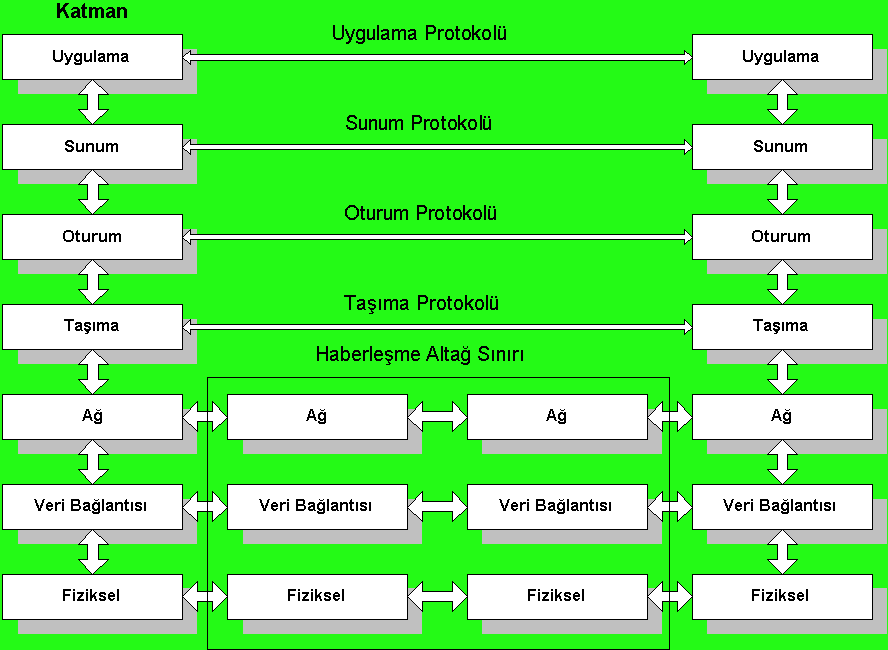


**Şekil 5.4 Mantıksal Model**

42

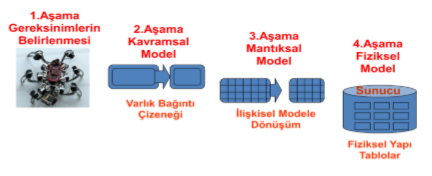
**4. Fiziksel Model**

Fiziksel olarak sistemin kurulması sağlanır. Kullanılacak VTYS ile ilgili ilk temas burada kurulur. Fiziksel model tasarımı şu şekilde olabilir:



**Şekil 5.5 Fiziksel Model**

**5.2.2.3 Şemalar**



**Şekil 5.6 Şema Tasarımı**

43

Üçlü şema mimarisinde görülen yapıların, sistemi kullanacak kişilerin gereksinimleri göz önüne alınarak aşamalı bir şekilde fiziksel olarak gerçekleştirilmesidir.

**5.2.2.4 VTYS Mimarisi**



**5.2.2.5 Veritabanı Dilleri ve Arabirimleri**

Veri tabanı dili olarak SQL kullanılmalı. Henüz prototip aşamasında olan sistemimiz için MYSQL arabirimi tercih edilmiştir.

**5.2.2.6 Veri Tabanı Sistem Ortamı**

Yükleme, yedekleme, performans ölçme, sıralama, veri sıkıştırma, ve benzeri fonksiyonları yerine getirmek amacıyla phpMyAdmin kullanılması planlanmıştır.

44

**5.2.2.7 VTYS'nin Sınıflandırılması**

ilişkisel veri modeli tercih edilmiştir.

**5.2.2.8 Hazır Program Kütüphane Dosyaları**

Ekstra bir kütüphane dosyasına ihtiyaç duyulmamaktadır.

**5.2.2.9 CASE Araç ve Ortamları**

Microsoft un Visio Project ürünlerini kullanmalıdır.

**5.3. Kodlama Stili**

Herhangi bir hazır düzene bağlı kalmadan modüler olarak kodlanacaktır.

**5.3.1 Açıklama Satırları**

Açıklama satırları karmaşık her satırın sonunda yapılacaktır.

**5.3.2 Kod Biçimlemesi**

Kod biçimlemesine değinmek gerekirse alt alta oluşan kodlarda tabi indexleri kullanılmalı ve iç içe bir biçimde hiyerarşi oluşturulmalı

45

**5.3.3 Anlamlı İsimlendirme**

Sistem kodlamasının genel yapısında kullanılan değişkenlerin veri tabanında karşılığı varsa önce “tabloadı\_islevadı\_sayısı” seklinde bir anlamlı isimlendirme yaptık. Sonradan kod okumasında daha kolay anlaşılabilmesi için değişkenlere sınıflara ve fonksiyonlara uzun açıklayıcı isimler verilmiştir.

**5.3.4 Yapısal Programlama Yapıları**

Genel olarak 3 başlıkta incelersek:

**Ardışık işlem yapıları:** Bu tür yapılarda genellikle fonksiyon, altprogram ve buna benzer tekrarlı yapıları tek bir seferde çözümlenmesi planlandı

**Koşullu işlem yapıları:** Yazılacak olan programda çok fazla sayıda koşul ifadesi olacağından dolayı birçok yerde koşullu işlemlere yer verilmesi gerekmektedir

**Döngü yapıları:** Tıpkı ardışık işlemler gibi alt alta birkaç satır yazacağımıza tek bir döngüyle bu sorunların üstesinden gelmesi planlanmalı ve uzun döngüler kullanma zorunluluğu olan kod bölümlerinde recursive fonksiyonlar tercih edilmelidir.

**5.4. Program Karmaşıklığı**

Bu başlık şu anda incelenme aşamasındadır.

**5.4.1 Programın Çizge Biçimine Dönüştürülmesi**

Bu başlık şu anda incelenme aşamasındadır.

**5.4.2 McCabe Karmaşıklık Ölçütü Hesaplama**

Bu başlık şu anda incelenme aşamasındadır.

46

**5.5. Olağan Dışı Durum Çözümleme**

Bekleyen bir durumda program uzun süre cevap vermezse kullanıcıya hatayı bildirmek isteyip istemediğini sorar ve tasarlanan try-cath kodlaması ile program çalışmasına en başa, Kullanıcı işlemler paneline dönüp devam eder.

**5.5.1 Olağandışı Durum Tanımları**

Olağandışı gelişen durumlarda try-catch blokları devreye girecek ve program kırılmadan çalışmasına devam edebilecek şekilde tasarladık.

**5.5.2 Farklı Olağandışı Durum Çözümleme Yaklaşımları**

Tüm olağan dışı durumlarda program kırılmadan hata mesajlarıyla tekrar başa dönecek şekilde tasarladık.



**Şekil 5.7 Olağandışı Durum Çözüm Modellemesi**

47

**5.6. Kod Gözden Geçirme**

Hiç kimse, önceki sürümlerini gözden geçirmeden ve incelemeden okunabilir bir program yazamaz. Hiçbir yazı editörün onayını almadan basılamayacağı gibi hiçbir program da incelenmeden, gözden geçirilmeden işletime alınmamalıdır. Kod gözden geçirme ile program sınama işlemlerini birbirinden ayırmak gerekir.

Program sınama, programın işletimi sırasında ortaya çıkabilecek yanlış ya da hataları yakalamak amacıyla yapılır. Kod gözden geçirme işlemi ise, programın kaynak kodu üzerinde yapılan bir incelemedir. Kod gözden geçirmelerinde program hatalarının %3-5 oranındaki kesimi yakalanabilmektedir. Eğer programı yazan kişi, yazdığı programın hemen sonra bir "kod inceleme" sürecine girdi olacağını bilerek program yazdığında daha etkin, az hatalı ve okunabilir programlar elde edilebilmektedir.

**5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi**

Bu başlık şu anda incelenme aşamasındadır.

**5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular**

Gözden geçirme sürecinin temel özellikleri;

 Hataların bulunması, ancak düzeltilmemesi hedeflenir,

 Olabildiğince küçük bir grup tarafından yapılmalıdır. En iyi durum deneyimli bir inceleyici kullanılmasıdır. Birden fazla kişi gerektiğinde, bu kişilerin, ileride program bakımı yapacak ekipten seçilmesinde yarar vardır.

 Kalite çalışmalarının bir parçası olarak ele alınmalı ve sonuçlar düzenli ve belirlenen bir biçimde saklanmalıdır. biçiminde özetlenebilir. Burada yanıtı aranan temel soru, programın yazıldığı gibi çalışıp çalışmayacağının belirlenmesidir.

48

**5.6.2.1 Öbek Arayüzü**

Oluşturduğumuz öbekleri test etmek için belli sorular sorduk bu sorular:

 Her öbek tek bir işlevsel amacı yerine getiriyor mu?

 Öbek adı, işlevini açıklayacak biçimde anlamlı olarak verilmiş mi?

 Öbek tek giriş ve tek çıkışlı mı?

 Öbek eğer bir işlev ise, parametrelerinin değerini değiştiriyor mu?

.

**5.6.2.2 Giriş Açıklamaları**

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

 Öbek, doğru biçimde giriş açıklama satırları içeriyor mu?

 Giriş açıklama satırları, öbeğin amacını açıklıyor mu?

 Giriş açıklama satırları, parametreleri, küresel değişkenleri içeren girdileri ve kütükleri 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?

Şeklinde oldu.

**5.6.2.3 Veri Kullanımı**

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

 İşlevsel olarak ilintili bulunan veri elemanları uygun bir mantıksal veri yapısı içinde gruplanmış mı?

 Değişken adları,işlevlerini yansıtacak biçimde anlamlı mı?

 Değişkenlerin kullanımları arasındaki uzaklık anlamlı mı?

 Her değişken tek bir amaçla mı kullanılıyor?

 Dizin değişkenleri kullanıldıkları dizinin sınırları içerisinde mi tanımlanmış?

 Tanımlanan her gösterge değişkeni için bellek ataması yapılmış mı?

49

**5.6.2.4 Öbeğin Düzenlenişi**

Modüller birleşimi uyumlumu?

 Modüller arası veri aktarımları sağlanıyor mu?

 Bütün modüller birleştiğinde sistem çalışıyor mu?

Gözden geçirme sırasında referans alınacak sorular olacaktır.

**5.6.2.5 Sunuş**

Artık son kısma gelindiğinde ise şu sorular soruldu:

 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 mı?

 Bütün deyimlerde, karmaşıklığı azaltacak şekilde parantezler kullanılmış mı?

 Bütün deyimler, belirlenen program stiline uygun olarak yazılmış mı?

 Öbek yapısı içerisinde akıllı "programlama hileleri" kullanılmış mı?

50

**6. DOĞRULAMA VE GEÇERLEME**

**6.1. Giriş**

Bu bölüm yazılım yaşam döngüsünde ki her bir ürününün teknik yeterliliğinin değerlendirildiği ve uygun çözümler bulunana kadar aktivitelerin tekrarlandığı yazılım yaşam döngüsü aşamalarından birisidir.

Bu bölümün temel hedefi ise doğru yazılımın üretilip üretilmediğini sınamaktır. Aynı zamanda yazılımın doğru bir şekilde üretilip üretilmediği de bu bölümde sınanmaktadır. Biz de bu sistemi tasarlarken kendimize bu soruları referans alarak projeyi yürütmeliyiz.

**6.2. Sınama Kavramları**

**Birim Sınama:** Bu bölümde sistemin birimleri sınandı ve sonuçlar çıkarılmalıdır.

**Alt Sistem Sınama:** Bu bölümde arayüzlere yönelik sınanmalar yapılmıştır. Hatalar bulunup düzeltilmiş ve sistem eksiksiz bir şekilde onaylanmalıdır.

**Sitem Sınama:** Sistem bütün olarak baştan sona sınandı, hatalar bulunup düzenlemeler yapıldı ve sistemin eksiksiz olduğu onaylanmalıdır.

**Kabul Sınama:** Sistem aktif hale getirilmeden önce son kez sınandı ve bir kez daha eksiksiz olduğu onaylanmalıdır.

**6.3. Doğrulama ve Geçerleme Yaşam Döngüsü**

Bu başlık şu anda incelenme aşamasındadır.

51

**6.4. Sınama Yöntemleri**

Sınama işlemi geliştirme aşamalarını izleyen ve hatalar bulunması sonucunda düzenlemelerin yapıldığı bölümdür. Ayrıca geliştirme öncesinde planlanan ve tasarımı yapılması gereken bir aşamadır.

**6.4.1 Beyaz Kutu Sınaması**

Beyaz kutu sınamasında, sistemin süreç belirtiminden yararlanılır.

-Bütün bağımsız yollar en az bir kere sınandı.

-Bütün mantıksal karar noktaları sınandı.

-Döngüler sınandı.

**6.4.2 Temel Yollar Sınaması**

Sistemdeki bağımsız yollar sınandıktan sonra, bu kadar sayıda sınama yaparak programın her birimini sınamalara dâhil etmiş olduk. Bunu gerçekleştirmek için iş akış şemaları diyagramlarından yararlanılmalıdır.

**6.5.Sınama ve Bütünleştirme Stratejileri**

Sistemde genel olarak sınama stratejileri ve bütünleştirme stratejileri birlikte değerlendirilmelidir.

Yukarıdan aşağıya ve aşağıdan yukarıya sınama stratejilerinin ikisi de sistem sınamasında uygulanmalıdır.

52

**6.5.1 Yukarıdan Aşağı Sınama ve Bütünleştirme**

Yukarıdan aşağıya sınama ve bütünleştirme de adından da anlaşılacağı üzere önce sistemin en üst düzeylerinin sınaması yapılmalıdır. Daha sonra aşağıya doğru olan düzeyleri sınanmalıdır.

**6.5.2 Aşağıdan Yukarıya Sınama ve Bütünleştirme**

Aşağıdan yukarıya sınama ve bütünleştirme aşamasında ise bir önceki aşamadaki işlemelerin tersi yönünde bir sınama işlemi yürütülmelidir.

**6.6. Sınama Planlaması**

Sınama işlemi çok kapsamlı ve detaylı yapılması gereken bir işlemdir. Bu nedenle bu sistemde sınama işlemleri yapılırken belli bir planlama çerçevesinde hareket edilmelidir. Ayrıca her bir yazılım döngü aşamasında farklı ayrıntılara odaklanılarak birden fazla sınama planı hazırlanmalıdır.

Yapılan sınama planlarında örnekler vermek gerekirse;

- Birim (Modül) Sınama Planı

- Alt Sistem Sınama Planları

- Bütünleştirme Sınama Planları

- Kabul Sınama Planları

-Sistem Sınama Planları

şeklinde örnekler verilebilir.

53

**6.7. Sınama Belirtimleri**

Sınama belirtimleri sistemin sınanma işleminin nasıl yapılacağına dair ayrıntıları içeren aşamadır. Bu ayrıntılar temel olarak sınama verilerini, sınama senaryolarını, sınama türünü, sınama stratejisini vb. gibi bilgileri içerir.

Bu sistemde de bu sınama belirtimlerinden faydalanılmalıdır. Ayrıca sınama işleminin bitiminden sonra sınama raporları da oluşturulmuştur.

**6.8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri**

Bilindiği üzere yazılım yaşama döngüsü bir yazılımı oluştururken dikkat edilmesi gereken en önemli unsurlardan birisidir. Bu sistem tasarlanırken yazılım yaşam döngüsü boyunca;

* Planlama
* Çözümleme
* Tasarım
* Gerçekleştirim
* Kurumlum
* Bakım

gibi temel aşamalar üzerinde durularak tasarlanmalıdır ve bu aşamaların her birinde sistem sınanmaları yapılmalıdır.

Sınama sırasında bulunan her hata için, değişiklik kontrol sistemine (DKS), "Yazılım Değişiklik İsteği" türünde bir kayıt girilir. Hatalar, DKS kayıtlarında aşağıdaki gibi gruplara ayrılabilir:

**• Onulmaz Hatalar:** BT projesinin gidişini bir ya da birden fazla aşama gerileten ya da düzeltilmesi mümkün olmayan hatalardır.

**• Büyük Hatalar:** Projenin kritik yolunu etkileyen ve önemli düzeltme gerektiren hatalardır.

**• Küçük Hatalar:** Projeyi engellemeyen ve giderilmesi az çaba gerektiren hatalardır.

**• Şekilsel Hatalar:** Heceleme hatası gibi önemsiz hatalardır.

54

**7.BAKIM**

**7.1 Giriş**

**7.2 Kurulum**

Sistem kurulumuna değinmek gerekirse, kendi serverlarımıza yüklenecek olan sistemimizde FTP arayüzü ile dosyalar servera aktarılmalı ve internet bağlantısı olan tüm ortamlarda çalışmalıdır.

**7.3 Yazılım Bakımı**

Kullanıcılar

Değişim isteği

Hizmet İsteği

Destek Ekibi

Sözleşme

Satıcı

Yeni Arabirim

Düzenleme Süreci

**7.3.1 Tanım**

Düzeltici Bakım

Uyarlayıcı Bakım

En İyileyici Bakım

55

**7.3.2 Bakım Süreç Modeli**

1. Adım: Sorunu Tanımlama Süreci

2. Adım: Çözümleme Süreci

3. Adım: Tasarım Süreci

4. Adım: Gerçekleştirim Süreci

5. Adım: Sistem Sınama Süreci

6. Adım: Kabul Sınaması Süreci

7. Adım: Kurulum Süreci

56