T.C YILDIZ TEKNİK ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM MÜHENDİSLİĞİ DERSİ FİNAL PROJESİ

Beş Duyu Medya Takip Sistemi Programı

Hazırlayan: Mehmet Furkan KUNDURACI

Civan KORKMAZ

Deniz

Dersin Sorumlusu: Prof. Dr. Oya KAPUSUZ

İSTANBUL, 2019

İçindekiler

[1) Giriş 4](#_Toc388638073)

[1.1) Projenin Amacı 4](#_Toc388638074)

[1.2) Projenin Kapsamı 4](#_Toc388638075)

[2. PROJE PLANI 5](#_Toc388638076)

[2.1 Giriş **Error! Bookmark not defined.**](#_Toc388638077)

[2.2 Projenin Plan Kapsamı 5](#_Toc388638078)

[2.3 Proje Ekip Yapısı 7](#_Toc388638079)

[2.4 Proje Zaman-İş Planı 7](#_Toc388638080)

[2.5 Önerilen Sistemin Teknik Tanımları 8](#_Toc388638081)

[2.6) Kullanılan Özel Geliştirme Araçları ve Ortamları 8](#_Toc388638082)

[2.7) Proje Standartları, Yöntem ve Metodojiler 8](#_Toc388638083)

[2.8 Kalite Sağlama Planı 9](#_Toc388638084)

[2.9 Konfigürasyon Yönetim Planı 9](#_Toc388638085)

[2.10 Kaynak Yönetim Planı 9](#_Toc388638086)

[2.11) Eğitim Planı 10](#_Toc388638087)

[2.12) Test Planı 10](#_Toc388638088)

[2.13 Bakım Planı 10](#_Toc388638089)

[3) Sistem Çözümleme 11](#_Toc388638090)

[3.1) Mevcut Sistemin incelenmesi 11](#_Toc388638091)

[3.2) Gereksenen Sistemin Mantıksal Modeli 12](#_Toc388638092)

[3.3 Arayüz (Modül) Gerekleri 18](#_Toc388638093)

[3.3.2 Kullanıcı Arayüzü 19](#_Toc388638094)

[3.4 Belgeleme Gerekleri 19](#_Toc388638095)

[4) Sistem Tasarımı 20](#_Toc388638096)

[4.1) Genel Tasarım Bilgileri 20](#_Toc388638097)

[4.2 Veri Tasarımı 24](#_Toc388638098)

[4.3 Süreç Tasarımı 25](#_Toc388638099)

[4.3.4 Entegrasyon ve Test Gereksinimleri 26](#_Toc388638100)

[4.4 Ortak Alt Sistemlerin Tasarımı 26](#_Toc388638101)

[5. Sistem Gerçekleştirimi 27](#_Toc388638102)

[5.1. Giriş 27](#_Toc388638103)

[5.2. Yazılım Geliştirme Ortamları 27](#_Toc388638104)

[5.3. Kodlama Stili 30](#_Toc388638105)

[5.4. Program Karmaşıklığı 30](#_Toc388638106)

[5.5. Olağan Dışı Durum Çözümleme 30](#_Toc388638107)

[5.6. Kod Gözden Geçirme 30](#_Toc388638108)

[6. Doğrulama Ve Geçerleme 33](#_Toc388638109)

[6.1) Giriş 33](#_Toc388638110)

[6.2) Sınama Kavramları 33](#_Toc388638111)

[6.3. Doğrulama ve Geçerleme Yaşam Döngüsü 33](#_Toc388638112)

[6.4. Sınama Yöntemleri 33](#_Toc388638113)

[6.5. Sınama Planlaması 34](#_Toc388638114)

[6.6. Sınama ve Bütünleştirme Stratejileri 34](#_Toc388638115)

[6.7. Sınama Belirtimleri 34](#_Toc388638116)

[6.8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri 35](#_Toc388638117)

[7) Bakım 36](#_Toc388638118)

[7.1 Giriş 36](#_Toc388638119)

[7.2 Kurulum 36](#_Toc388638120)

[7.3 Yerinde Destek Organizasyonu 36](#_Toc388638121)

[7.4 Yazılım Bakımı 36](#_Toc388638122)

[8. SONUÇ 37](#_Toc388638123)

## Giriş

## Projenin Amacı

Haberleşme, haber alma insan varoluşundan bu yana hep bir ihtiyaç olmuştur. Geçmişten günümüze haber alım araçları değişip gelişmiştir. Günümüzde bir çok farklı kaynaktan haber alımı ve medya takibi yapılmaktadır. Fakat medya sektörü gün geçtikçe gelişmekte her geçen gün bir yayın organı faaliyete geçmektedir. İnternet siteleri, Gazeteler, Dergiler, Sosyal Medya, Radyo gibi alanlarda binlerce yayın organı faaliyet göstermektedir. Bu sebeple bir çok şirket, kişi gerek bazı alanlara yönelik haberleri, gerekse kendilerinin yer aldığı haberleri takip etmekte zorlanmaktadır. Bu sebeple bu takibi yapan şirketlere veya yazılımlara ihtiyaç duymamaktadırlar. Bu projenin amacı istenilen alanlarda veya konularda yazılı, görsel veya duyusal medyada çıkan haberlerin takibi yaparak talepte bulunan müşteriye raporlama yapmaktır. Müşterileri istediği konular hakkında raporlama alabilir.

## Projenin Kapsamı

Sistemi kullanmak için Windows 7 ve üzeri işletim sistemine sahip olmak gerekir.

* Çeşitli tipte medya takibi yapılır.
* Her tipte değişik sayıda yayın takip edilir.
* Şirket müşterileri için standart olarak günlük, haftalık, aylık raporlar hazırlamaktadır
* Müşteri raporunu özel olarak değişik periyotlarla da (örneğin saatlik rapor) isteyebilir.
* Rapor fiyatları belirlidir değişmez.

## 1.3 Tanımlamalar ve Kısaltmalar

|  |  |
| --- | --- |
| Şirket Yöneticisi | Programı kullanan ve yönetimini yapan kişidir. |
| Şirket Müşterisi | Belli haklara sahip olan rapor alan kişidir. |
|  |  |
|  |  |

## 2. PROJE PLANI

### 2.1) Proje Kaynaklarının Belirlenmesi

### 2.1.1.) İnsan Kaynakları

//// burayı bilemedim///

Oluşturulacak sistemin geliştirme safhası süresince yazılım geliştiriciler ve yazılım yönlendiriciler olmak üzere iki alt grup sürekli dahil edilecektir.

##### Yazılım Geliştiriciler

* + **Tasarım Uzmanı ;** Web arayüzününün tasarımını gerçekleştirecek.
  + **Web Programcısı ;** PHP ya da ASP alanında uzmanlaşmış bir mühendis
  + **Veri Tabanı Uzmanı ;** MYSQL ve SQLite alanında uzmanlaşmış bir veri tabanı uzmanı olacak.
  + **Proje Yöneticisi ;** Projenin her aşamasında uzmanlara müdahale eden onları yönlendiren kişidir.

##### Yazılım Yönlendiriciler

* + **Fitness Antrenörü ;** Gerekli egzersizlerin oluşturulması için gerekli direktifleri veren yönlendirici kişi.

///////////////////////////////////////

### 2.1.2.) Donanım Kaynakları

Gerçekleştirceğimiz sistem hem mobil tabanlı hem de web tabanlı bir sistem olduğundan en az bir adet Android tabanlı telefona ihtiyaç vardır. Bununla birlikte web arayüzün nasıl çalıştığını görmemiz açısından bir adet istemci bilgisayarın olmasında yarar vardır.

### 2.1.3.) Yazılım Kaynakları

* **Programlama Dili ;** C# Programlama Dili kullanılacak
* **Veritabanı olarak ;** MYSQL ve SQLite Veri tabanı Kullanılacak
* **Arayüz için ;** Visual Studio

### 

### 2.2 Projenin Plan Kapsamı

|  |  |  |  |
| --- | --- | --- | --- |
| **Ölçüm Parametresi** | **Sayı** | **Ağırlık** | **Toplam** |
| Kullanıcı Girdi Sayısı | 6 | 10 | 60 |
| Kullanıcı Çıktı Sayısı | 4 | 10 | 40 |
| Kullanıcı Sorgu Sayısı | 2 | 4 | 8 |
| Kütük Sayısı | 6 | 9 | 54 |
| Dışsal Arayüz Sayısı | 3 | 5 | 10 |
| Ana İşlev Nokta Sayısı |  |  | 172 |
| **Teknik Karmaşıklık Sorusu** | | | | | **Puan** |
| 1. Uygulama, güvenilir yedekleme ve kurtarma gerektiriyor mu? | | | | | 4 |
| 2. Veri iletişimi gerekiyor mu? | | | | | 5 |
| 3. Dağıtık işlem işlevleri var mı? | | | | | 1 |
| 4. Performans kritik mi? | | | | | 3 |
| 5. Sistem mevcut ve ağır yükü olan bir işletim ortamında mı çalışacak? | | | | | 3 |
| 6. Sistem, çevrim içi veri girişi gerektiriyor mu? | | | | | 4 |
| 7. Çevrim içi veri girişi, bir ara işlem için birden çok ekran gerektiriyor mu? | | | | | 3 |
| 8. Ana kütükler çevrim-içi olarak mı günleniyor? | | | | | 4 |
| 9. Girdiler, çıktılar, kütükler ya da sorgular karmaşık mı? | | | | | 2 |
| 10. İçsel işlemler karmaşık mı? | | | | | 3 |
| 11. Tasarlanacak kod, yeniden kullanılabilir mi olacak? | | | | | 4 |
| 12. Dönüştürme ve kurulum, tasarımda dikkate alınacak mı? | | | | | 3 |
| 13. Sistem birden çok yerde yerleşik farklı kurumlar için mi geliştiriliyor? | | | | | 1 |
| 14. Tasarlanan uygulama, kolay kullanılabilir ve kullanıcı tarafından kolayca değiştirilebilir mi olacak? | | | | | 5 |
| Toplam | | | | | 45 |
|  | | | | |  |

////// puanlamayı birlikte yapalım..

0: Hiçbir Etkisi Yok

1: Çok Az etkisi var

2: Etkisi Var

3: Ortalama Etkisi Var

4: Önemli Etkisi Var

5: Mutlaka Olmalı, Kaçınılamaz

İN=97x(0.65 x 0.01 x 43)

İN=50,31

Satır Sayısı = 50,31 \*30

Satır Sayısı =1510

**Ayrık Projeler :**

İş gücü(K)=2,4\*S^1,05=2,4\*2^1,05=~4

Zaman(T)=2,5\*K^0,38=2,5\*5^0,38=~4

Üretkenlik=İN/Kişi-Ay => 50/4=12,5

### 2.3 Proje Ekip Yapısı

##### Yazılım Gerçekleştirme Aşamasında görev Alanlar

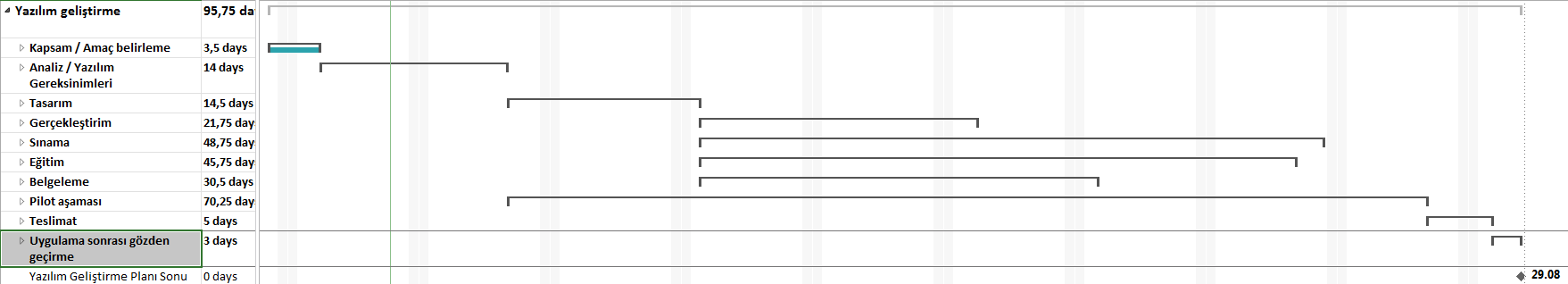
Proje kapsamında 5 eleman görev almaktadır. Mobil Arayüzün ve Web arayüzünün kodlanması için 2 kişi. Mobil ve Web arayüzlerinin veritabanı yapısının oluşturulması için bir Veritabanı uzmanı , Mobil ve Web Arayüzlerinin tasarlanması için bir adet Tasarımcı ve Projenin doğru şekilde yürütülmesini sağlması için bir adet Proje yöneticisi.

##### Yazlım Bilgilendirme Aşamasında Görev Alanlar

Proje kapsamında yazılım bilgilendirme aşamasında Şirket yöneticisi görev almaktadır. Yönetici raporlamanın nasıl yapıldığını vs gibi bilgileri vererek yönlendirme sağlar.

### 2.4 Proje Zaman-İş Planı

* Projenin zaman planı MS Project üzerinden takip edilecektir.
* Zaman Planı her hafta sonu gelen veriler ışığında güncellenir.
* Zaman planındaki sapmalar sebepleriyle Zaman çizelgesi ve teslim süreci tekrar raporlanır.



### 2.5 Önerilen Sistemin Teknik Tanımları

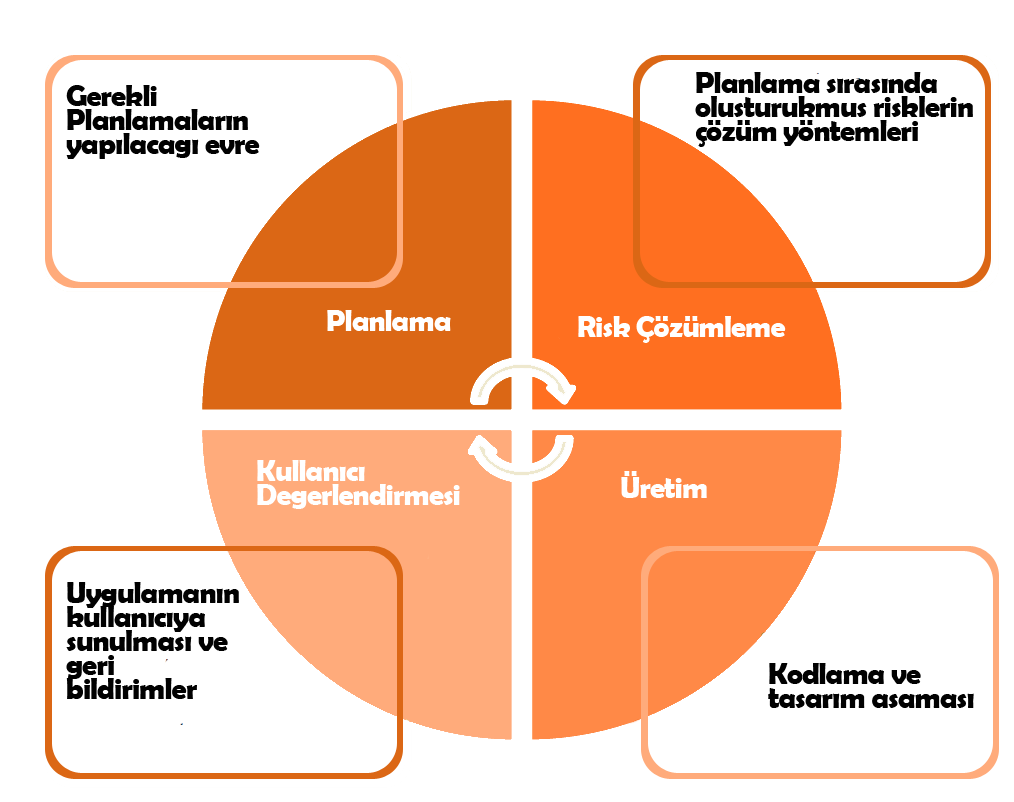
Sistem esnek bir yapıya sahip programlama dili olan Java patformu tabanlı Windows Form olarak geliştirilmektedir. Son verilere göre Dünyada halen geniş bir kesim Windows İşletim sistemi kullanmaktadır. Özellikle kurumsal firmaların tamamına yakınında Windows işletim sistemi kullanılmaktadır. Bu yüzden Sistem bu kitleyi hedef alarak geliştirmiş olup uygulama gerekli görülürse IOS, Android gibi popüler platformlara port edilecektir.

### 2.6) Kullanılan Özel Geliştirme Araçları ve Ortamları

* Sistem C# platformunda geliştirileceği için IDE olarak Visual Studio seçilmiştir.
* Veri tabanı olarak C# ile uyumlu çalışan mySQL seçilmiştir.
* Form arayüzün tasarlanması için Visual Studio seçilmiştir.
* Projenin Dökümantasyonu ve raporlama için MS WORD araçları kullanılmıştır.
* Proje Zaman ve iş planlaması için MS Project 2013 kullanılmıştır.

### 2.7) Proje Standartları, Yöntem ve Metodojiler

Projenin gelişim sürecinde Helezonik model tercih edilecektir. Bu modelin tercih sebeplerinden en önemlisi Kullanıcı Değerlendirmesidir. Mobil uygulamalar kullanıcı geri bildirimleriyle gelişir ve zenginleşir. Helezonik model yapı itibariyle bu sisteme uygundur.



### 2.8 Kalite Sağlama Planı

SPICE(ISO 15504) Modeli referans alınacaktır. Bu Modelin kapsamı;

Yazılım satın alma.

Yazılım geliştirme.

İşletim.  
Bakım ve destek süreçleri için

### 2.9 Konfigürasyon Yönetim Planı

Ara Çözünürlük değerine sahip telefonlarda Layout’lar istenilen şekillerde görüntülenmeyebilir.

### 2.10 Kaynak Yönetim Planı

Proje ekiplerinin, risk yönetimi de dahil olmak üzere projenin her bölümünü gözlemleyen ve ölçülendiren bir mekanizmaya gereksinim duymalarından dolayı yönetim kontrol sistemleri oldukça önemlidir.

**-** Risk yönetimi içim temel hedeflerin belirlenmesi

**-** Risk yönetiminin ilgi alanının belirlenmesi(hangi risk alanları üzerinde çalışılacak ve çalışmalar hangi seviyede detaylandırılacak)

- Risk yönetimi yetkisinin belirlenmesi uygun bütçenin tanımlanması

- Kabul edilmiş risklerin belirlenmesi

- Risk yönetimi işlemleri, yöntemleri veya tekniklerinin tanımlanması

- Proje ilgileri ve önceliklerinin tanımlanması.

### 2.11) Eğitim Planı

Proje kapsamında herhangi bir eğitim planı uygulanmayacaktır.

### 2.12) Test Planı

Uygulama çeşitli işlem gücüne ve donanım özelliklerine sahip Windows işletim sistemi kurulu olan masaüstü ve taşınabilir bilgisayarlarda test edilip , aksaklıkları giderilecektir.

### 2.13 Bakım Planı

Sürümlerde ortaya çıkan herhangi bir sorunun kullanıcıların şikayeti üzerine bir sonraki sürümde düzeltilerek ve istenilen başka özellikler geliştirilerek tekrar kullanıcılara yeni sürüm sunulur.

## Sistem Çözümleme

### 3.1) Mevcut Sistemin incelenmesi

#### 3.1.1) Örgüt Yapısı

* Kullanıcı kas gruplarına göre egzersizleri görüntüleyebilir.
* Kullanıcı aramak istediği egzersizi arama sayfasında aratır.
* Sonuçlanan cevaplardan birini seçer.
* Egzersiz hakkında bilgi alır
* Kullanıcı Fitness Programı oluşturabilir.
* Kullanıcı oluşturduğu Fitness programını web sitesinde ya da Sosyal Medyada paylaşabilir.

#### 3.1.2) İşlevsel Model

* Kullancılar egzersizleri aratır ve sorguları inceler.
* Kullanıcıların egzersizler hakkında doğru ve yeterli bilgiye ulaşmaları zordur.
* Egzersiz programı oluşturmak zahmetlidir.
* Kullanıcılar sosyal medyada paylaşmak istediği veriye ulaşamayabilir.

#### 3.1.3) Veri Modeli

#### 3.1.4) Var olan Yazılım / Donanım Kaynakları

Jefit ve birçok uygulama mevcuttur.

#### 3.1.5) Var olan Sistemin Değerlendirilmesi

* Jefit Android Application mutlak Internet bağlantısı gerektirir. Kullanıcılar Spor yaparken internet bağlantısına sahip olmayabilir.
* Jefit Android Application sadece İngilizce dil desteğine sahiptir. Kullancılar İngilizce bilmeyebilir.

### 3.2) Gereksenen Sistemin Mantıksal Modeli

#### 3.2.1) Giriş

*Önerilen sistemde Kullanıcı cep telefonu üzerinden aşağıda sıralanan sorunları kolaylıkla çözmüş olacak;*

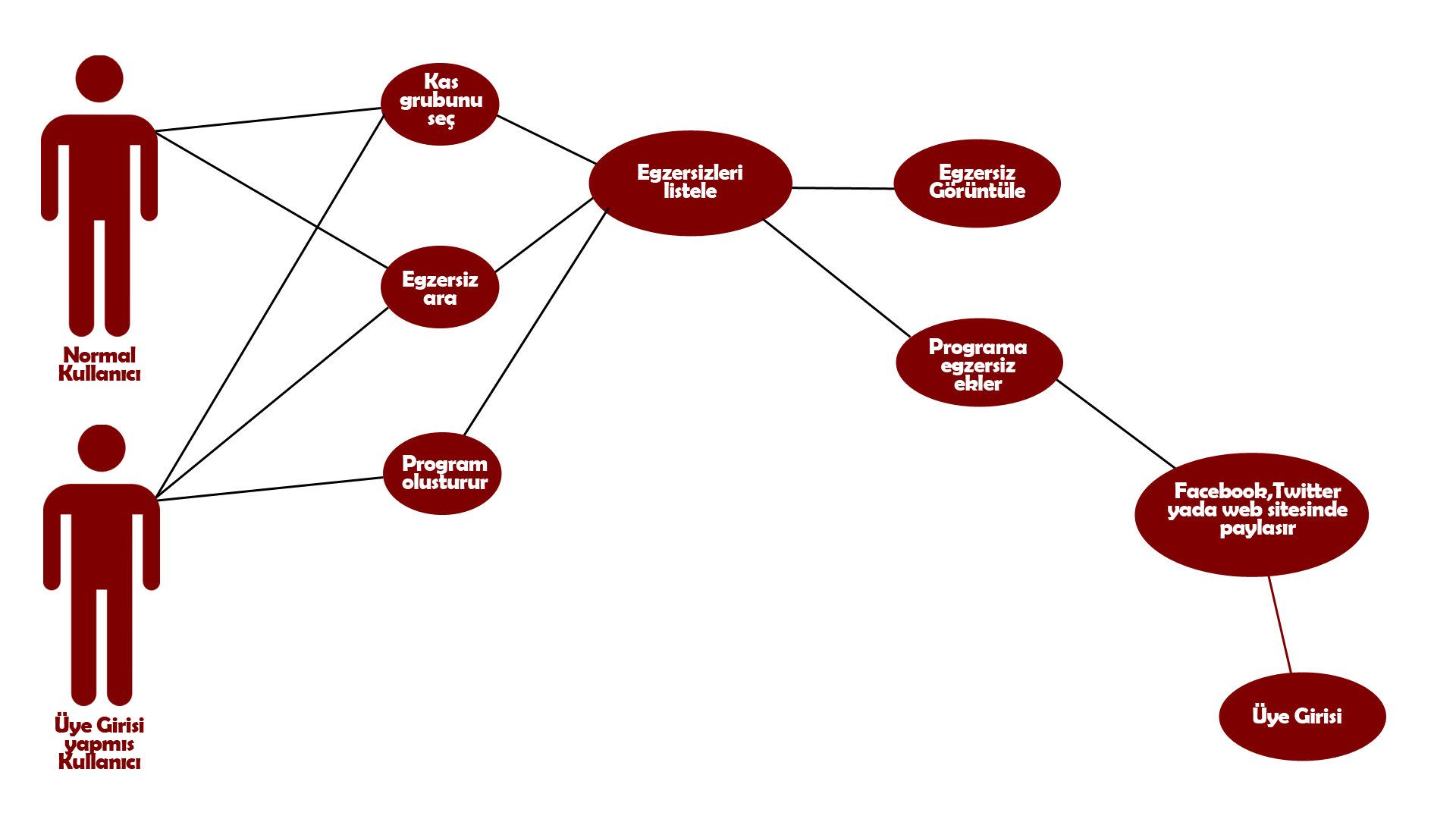
* Çeşitli medya organlarını takip edebilecek
* Her tipte değişik sayıda yayın takip edebilecek
* Günlük, Haftalık ve Aylık rapor alabilecek
* İstediği takdirde Özel rapor alabilecek
* İtibari, Bilinirliği vb. Konularda anket isteyebilecek
* Anket sonuçlarını görüntüleyebilecektir.

#### 3.2.2) Genel Bakış

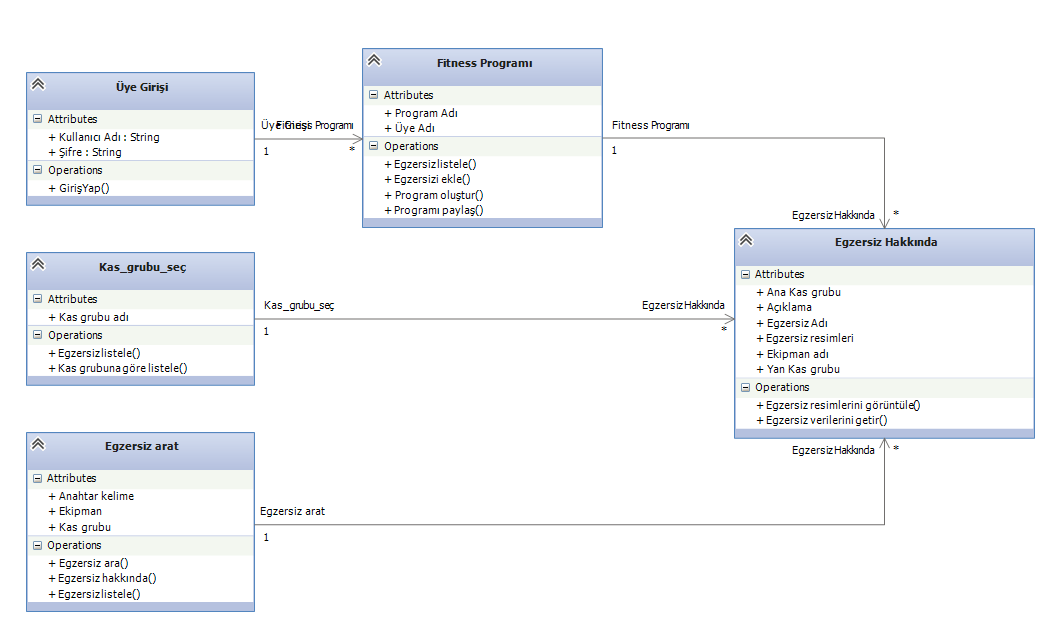
Egzersiz hakkında bilgiye ulaşması için, kas grubu ekranından istediği kas grubunu seçip egzersizleri listeledikten sonra istediği egzersize tıklamalı ya da egzersiz aratma ekranından çeşitli parametrelere göre egzersizi aratıp çıkan listeden istediği egzersize tıklamalı.

#### 3.2.3) Bilgi Sistemleri/Nesneler Veri Modeli

#### 3.2.4) İşlevsel Model



Use-Case Diyagramı yukarıdaki gibidir.

  
*Uygulamanın Class(Sınıf) Diyagramı yukarıdaki gibidir.*

#### 3.2.5) Veri Modeli

#### 3.2.6 Veri Sözlüğü

##### EGZERSİZ ARAT

**Egzersiz Ara():**Egzersiz arama layout’unda egzersiz aratılınca gerçekleştirilen fonksiyon.   
**Egzersiz Listele():**Bulunan egzersiz sonuçlarını listeler.   
**Egzersiz Hakkında():**Egzersiz Hakkında detaylı bilgi ekranına ulaşmasını sağlayan fonksiyon.

##### EGZERSİZ HAKKINDA

**Egzersiz bilgilerini getir():** Veritabanından egzersiz bilgilerini getiren fonksiyon.  
**Egzersiz resimlerini görüntüle():** Egzersiz resimlerini Id’ye göre bir düzen içinde anime eder.

##### KAS GRUBU GORUNTULE

**Kas\_gruplarini\_getir():** Kas gruplarini yerlerini gostererek getirir.  
**Egzersiz Listele():** Bulunan egzersiz sonuçlarını listeler.   
**Egzersiz Hakkında():** Egzersiz Hakkında detaylı bilgi ekranına ulaşmasını sağlayan fonksiyon.

##### FİTNESS\_PROGAMI()

**Program oluştur():** Bir isimle Fitness Programı oluşturur.

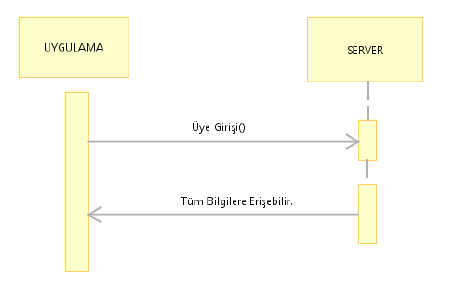
**Egzersiz Listele():** Egzersizleri listeler.

**Programa egzersiz ekle():** Oluşturulan fitness programına listeden seçilen egzersizleri ekler.

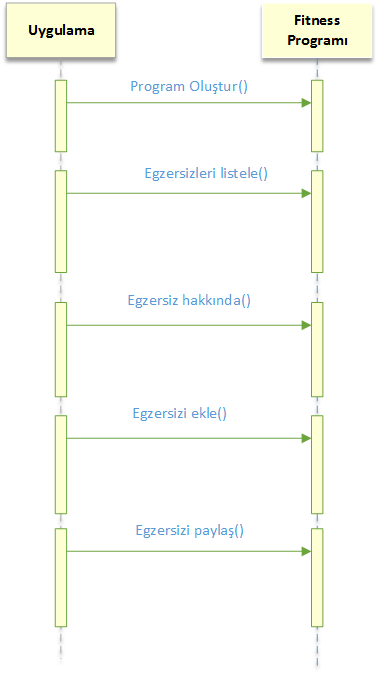
**Programı paylaş():** Oluşturulan programı sosyal medyada yada web sitesinde paylaşır.

**Programı sil/duzenle():** Oluşturulan programın silinmesini ya da düzenlenmesini sağlar.

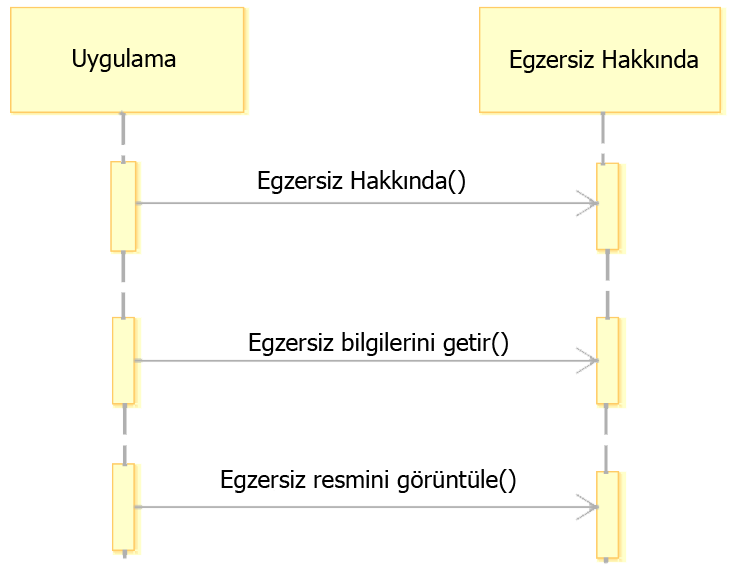
#### 3.2.7 İşlevlerin Sıra düzeni



*Şekil-3.2.1:Üye Girişi sequance(sıra) diyagramı.*



*Şekil-3.1:Fitness Programı sequance(sıra) diyagramı.*



*Şekil-3.2:Egzersiz Hakkında sequance(sıra) diyagramı.*

#### 3.2.8 Başarım Gerekleri

Uygulamanın bir kaç büyük şirket tarafından kullanılması İlk öncelikli hedeftir. Hedeflenen kitleyle geri bildirimlerle uygulamanın yeni versiyonunda daha çok şirket ve kullanıcıya hitap etmesi öngörülmektedir.

### 3.3 Arayüz (Modül) Gerekleri

#### 3.3.1 Yazılım Arayüzü

* Kullanıcı Ana Menüden seçim yapar. Egzersiz butonuna tıklarsa Kas grubu görüntüle sayfası, arama butonuna tıklarsa egzersiz arama sayfası, hakkında butonuna tıkarsa program hakkında sayfası ekrana gelir.
* Kullanıcı Egzersiz arattığında anahtar sözcük, ekipman adı ve kas grubuna göre veritabanında arama yapılır. Bulunan sonuçlar listelenir.
* Kullanıcı Egzersiz listesinde herhangi bir egzersize tıkladığında o egzersiz hakkındaki bilgiler veritabanından çekilir.
* Kullanıcı Egzersiz listesinde herhangi bir egzersize tıkladığında o egzersizin resimleri görüntülenir.
* Kullanıcı Fitness programı oluşturmak istediğinde Program butonuna tıklar. Kullanıcının karşısına programın ismini gireceği sayfa gelir. Ardından egzersizler listelenir.Kullanıcı listeden istediği egzersizleri seçer ve programa ekler.
* Kullanıcını Üye girişi yaparsa alınan veriler facebook for Android sdk sayesinde facebook serverlarına iletilir. Kullanıcı bilgileri doğrulandıktan sonra kullanıcı üye girişi yapmış olacaktır.

### 3.3.2 Kullanıcı Arayüzü

* Ana Menü
* Kas Gruplarını görüntüle
* Egzersiz listele
* Egzersiz hakkında
* Egzersiz ara
* Program hakkında
* Fitness Programı
* Üye Girişi
* Web Arayüzü

#### 3.3.3 İletişim Arayüzü

Facebook pop-up’la facebook üye girişi sağlanacaktır. Twitter pop-up’la twitter üye girişi sağlanacaktır. Ayrıca web sitesiyle üye girişi sağlanacaktır.

#### 3.3.4 Yönetim Arayüzü

Uygulama herhangi bir yönetim ara yüzüne sahip değildir.

### 3.4 Belgeleme Gerekleri

#### 3.4.1 Geliştirme Sürecinin Belgelenmesi

Geliştirme Süreci gerekli kriterler doğrultusunda belgelendirilmiştir.

#### 3.4.2 Eğitim Belgeleri

Kullanıcı için hazırlanılmış eğitim dokümanları aşağıdaki gibidir;

*-Görsel olarak hazırlanmış web sayfası.*

*-Video olarak hazırlanmış klip.*

#### 3.4.3 Kullanıcı El Kitapları

Uygulama mobil bazlı olduğundan kullanıcı için geliştirilmiş El Kitap’ı yoktur. Sanal El-Kitap aşağıdaki adresten ulaşabilir.

## Sistem Tasarımı

### 4.1) Genel Tasarım Bilgileri

#### 4.1.1) Genel Sistemin Tanımı

Hedef, yapı diyagramını oluşturmaktır. Veri akış diyagramından yola çıkarak işlemlerin tanımlanmasıdır.

#### 4.1.2) Dış Arabirimler

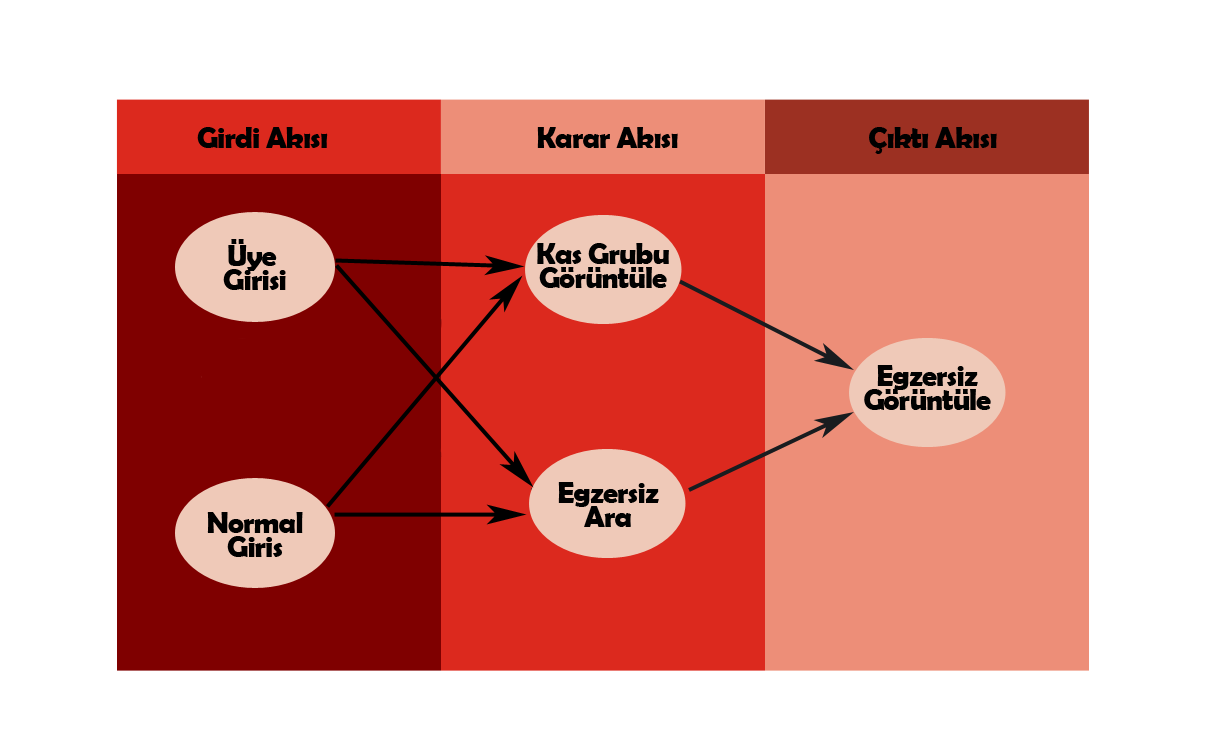
##### 4.1.2.1 Kullanıcı Arabirimleri

Sistem sadece kullanıcıları kapsadığından tüm arabirimler kullanıcı arabirimidir. Bunlar;

Egzersiz Hakkında, Kas Grubu Görüntüleme , Egzersiz Listeleme, Egzersiz Arama,Fitness Programı, Üye girişi ’ dır.

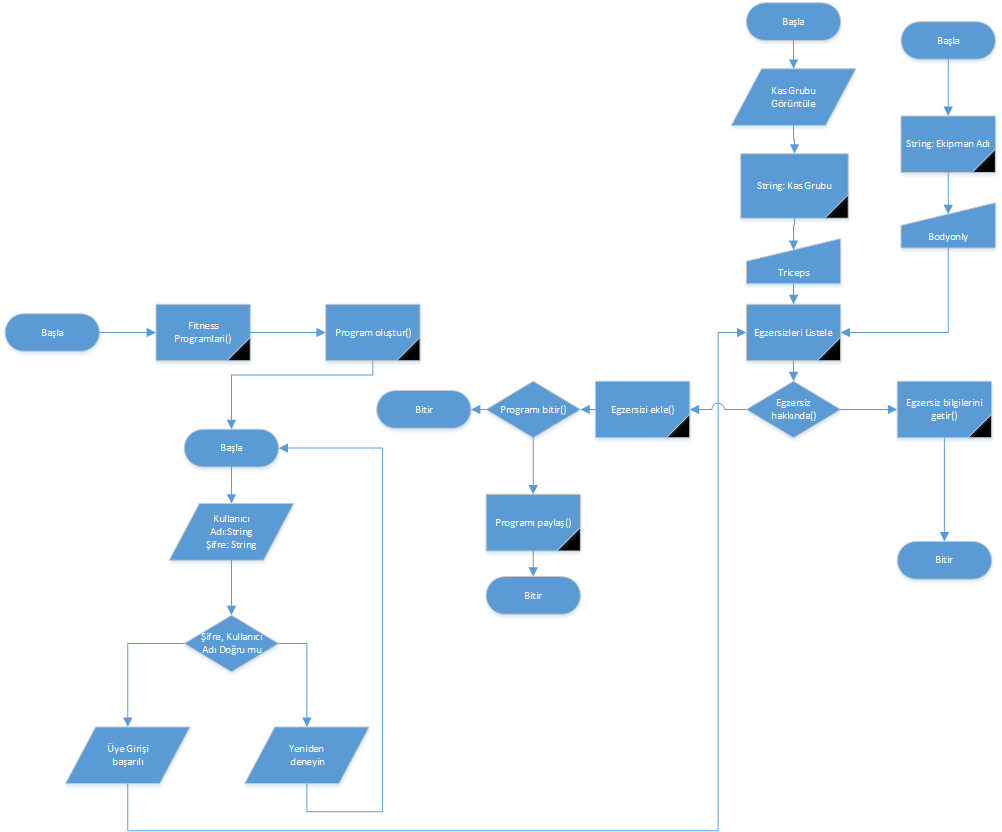
##### 4.1.2.2 Veri Arabirimleri

Veri akışlarını üç parçaya ayırmak mümkündür. Girdi akışı, karar akışı ve çıkış akışı.



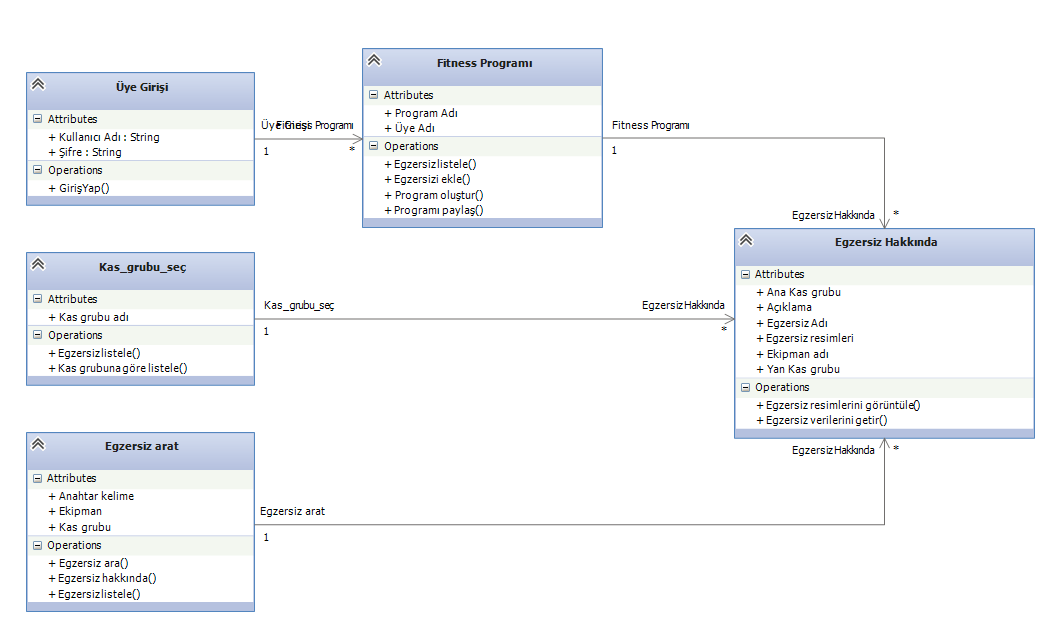
Ara İşlem Akışı

#### 4.1.3) Sistem Mimarisi



Şekil 4.1 Sistemin Akış diyagramı

#### 4.1.4 Veri Modeli



Tasarlanan activitylerden

Egzersiz hakkında ve Egzersiz Arat arasında “bire çok ilişki” bulunmaktadır.

Egzersiz hakkında ve Kas Grubu Görüntüle arasında “bire çok ilişki” bulunmaktadır.

Egzersiz hakkında ve Fitness Programı arasında “bire çok ilişki” bulunmaktadır.

Üye Girişi ve Fitness Programı arasında “bire çok ilişki” bulunmaktadır.

* Nesne tabanlı bir veri tabanı olduğundan ER diyagramları ile veri tabanı tasarımı gerçekleştirildi.
* Veri tabanı olarak web arayüzü için MYSQL, mobil arayüz için SQLite kullanıldı.
* Sistem çeşitli modüllerden oluşmaktadır.

##### 4.1.4.1) Veri Tabanında Bulunan Tablolar ve Özellikleri

* Egzersizler
* Fitness Programları
* Üyeler
* Kas grubu

Projenin gerekleri kapsamında sistemde olması gereken özellikler şu şekildedir.

1. **Egzersiz Bilgi sistemi**

Kullanıcılar aradığı egzersizleri kolayca bulmasını sağlayan ve egzersizin nasıl yapılması gerektiği hakkında bilgi veren sistem.

1. **Fitness programı yönetim sistemi**

Kullanıcıların beğendikleri egzersiz hareketlerinden bir egzersiz programı oluşturmalarını sağlayan sistem. Kullanıcılar bu programları oluşturabilir yada mevcut programlarını düzenleyebilir veya silebilir.

1. **Üyelik Sistemi**

Kullancıların Facebook ve ya Twitter bağlantısıyla ya da herhangi bir bağlantıya gerek kalmaksızın sisteme üye olmalarını sağlayan sistem.

#### 4.1.5 Testler

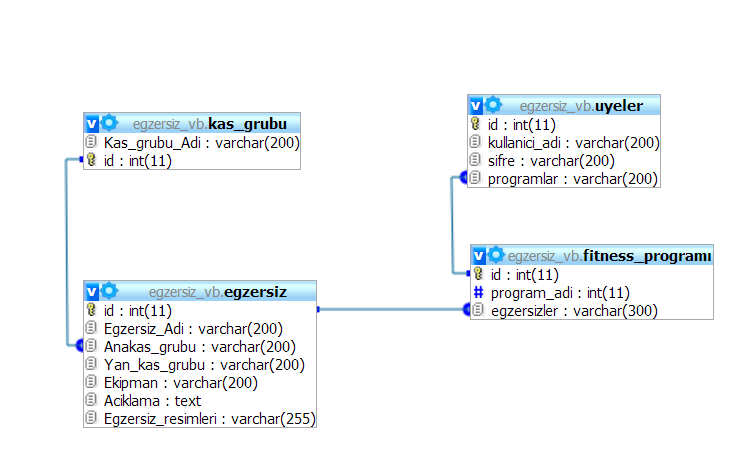
Veri modelleri yüksek egzersiz kaydı ve buna bağlı performans kayıtlarıyla test edilecektir. Optimizasyon kriterleri gerekli uygulama için önem arz etmemektedir.

#### 4.1.6 Performans

Yazılımın modüler olarak yönetilmesi, gerekli dokümantasyonun sağlanması ve plana bağlı kalarak uygulamanın geliştirmesi performans değerlerini öncülük etmektedir. Uygulamanın performans sonuçları Test konusunda detaylı aktarılmıştır.

### 4.2 Veri Tasarımı

#### 4.2.1 Tablo tanımları



#### 4.2.2 Tablo- İlişki Şemaları

Egzersiz hakkında ve Kas Grubu arasında “bire çok ilişki” bulunmaktadır.

Egzersiz hakkında ve Fitness Programı arasında “bire çok ilişki” bulunmaktadır.

Üyeler ve Fitness Programı arasında “bire çok ilişki” bulunmaktadır.

#### 4.2.3 Veri Tanımları

Egzersiz ismi: Egzersizin ismini tutar.

Ana Kas Grubu: Egzersizin çalıştırdığı ana kas grubunu tutar.

Yan Kas Grubu: Egzersizin çalıştırdığı yan kas grubunu tutar.

Ekipman: Egzersiz için gerekli ekipman(lar)ı tutar.

Aciklama: Egzersizin uygulama aşamalarını açıklayan metni tutar.

Program adi: Program adını tutar.

Kullanıcı Adı: Kullanıcı adını tutar.

Kullanıcı şifresi: Kullanıcı şifresini tutar.

### 4.3 Süreç Tasarımı

#### 4.3.1 Genel Tasarım

Süreç tasarımında algoritmik detayların belirtilmiştir.

#### 4.3.2 Modüller

##### 4.3.2.1 Veritabanı Modülü

###### 4.3.2.1.1 İşlev

Verilerin veritabanına kaydedilmesini sağlar.

###### 4.3.2.1.2 Kullanıcı Arabirimi

Herhangi bir arabirime sahip değildir.

###### 4.3.2.1.3 Modül Tanımı

Uygulamada egzersiz bilgilerini tutmamız için veritabanına ihtiyaç vardır. Bunu sağlamak için Veritabanı oluşturup bilgileri veritabanına eklememiz gerekir. Bu modül uygulama ilk çalıştırıldığında verileri veritabanına kaydeder. Sonraki açılışlarda bu verileri kullanır.

#### 4.3.3 Kullanıcı Profilleri

2 tip kullanıcı profili mevcuttur. “Normal kullanıcı” ve “Üye kullanıcı” profillerinin yapabilecekleri aşağıda belirtilmiştir.

### 4.3.4 Entegrasyon ve Test Gereksinimleri

Uygulama Sony Xpreia Walkman Live , GM Discovery , Samsung Galaxy S4, Google Nexus S model telefonlarda test edilip , aksaklıkları giderilecektir.

### 4.4 Ortak Alt Sistemlerin Tasarımı

#### 4.4.1 Ortak Alt Sistemler

Bir bilgi sistemi tasarlarken bilgi sistemlerinde ortak olarak bulunan bazı alt sistemler bulunmaktadır. Söz konusu alt sistemler;

• Güvenlik Alt sistemi

• Veri Dağıtım Alt sistemi

• Yedekleme ve Arşivleme İşlemleri

#### 4.4.2 Güvenlik Alt sistemi

Bu sistem bilgi sisteminde yapılan işlerin ve yapan kullanıcıların izlerinin saklanması ve gereken durumlarda sunulmasıyla ilgilidir. Bu bağlamda tutulan “log” kayıtları kritik durumlarda kullanıcıya veya yetkiliye sunulması amaçlanmaktadır.

#### 4.4.3 Veri Dağıtım Alt sistemi

Veri dağıtımı çevrim dışı olarak yapılmaktadır. Uygulamada veriler ilk açılışta otomatik olarak veritabanına yüklenir. Ardından kullanıcı ulaşmak istediği veriye menülerü ve fonksiyonları kullanarak ulaşır.

#### 4.4.4 Yedekleme ve Arşivleme İşlemleri

Kullanıcı herhangi bir nedenden dolayı programın verilerini silerse , program, bu verileri silmeden hemen sonraki ilk açılışta tekrar yükler.

## 5. Sistem Gerçekleştirimi

5.1. Giriş

Tasarım sonucu üretilen verilerin bilgisayar ortamında çalışan yazılıma dönüştürülme çalışmalarını içerir. Gerçekleştirim sırasında yazılım dokümantasyonun önemini gerek kodlayıcı gerekse uygulama tasarlayıcısı fark edecektir. Sistem Gerçekleştirimi aşamasındaki tüm hatalar Test ve Sınamada bakım altına alınacaktır.

### 5.2. Yazılım Geliştirme Ortamları

#### 5.2.1 Programlama Dilleri

Android mobil uygulamaları Java dilinde yazılmaktadır bu bağlamda sistemde Java programlama dili tercih edilmiştir Ayrıca diğer telefonlara port edilmesini kolaylaştırmaktadır.

Web Arayüzü için programla dili olarak PHP edilmiştir.

#### 5.2.2 Veri Tabanı Yönetim Sistemleri

##### 5.2.2.1 VTYS Kullanımının Ek Yararları

VTYS kullanımının ek yararları.

##### 5.2.2.2 Veri Modelleri

Gösterimsel Veri modelinde bulunana “Ağ Modeli” kullanılacaktır.

##### 5.2.2.3 VTYS Mimarisi

VTYS Mimarisi İçsel Yüzey, Kavramsal Yüzey, Dışsal Yüzey olarak ayrılır. Kısaca bu yüzeylerde hedeflenen amaçlar; **İçsel Yüzey:** Fiziksel veri yolunu detaylarıyla belirtir. **Kavramsal Düzey:** Kavramsal şema içerir ve kullanıcılar için veri tabanın yapısına açıklar. **Dışsal Yüzey:** Bu yüzey kullanıcı görüşlerini içerir. Her şema bir veri tabanının bir bölümünü açıklar.

##### 5.2.2.4 Veri tabanı Dilleri ve Arabirimleri

SQLite: Tamamen C/C++ programlama dilleriyle geliştirilmiş sunucu yazılımı ve konfigürasyon gereksinimi olmayan, işlevsel ve ilişkisel bir SQL veri tabanı motorudur.

MySQL: PHP ile uyumlu bir veritabanı sistemidir.

##### 5.2.2.5 Veri Tabanı Sistem Ortamı

Veri Tabanı Sistem Ortamın ’da yükleme, yedekleme, ölçme, sıralama gibi fonksiyonları yerine getirebilmek için yardımcı yazılımlar bulunur.

##### 5.2.2.6 VTYS'nin Sınıflandırılması

VTYS’nin Sınıflandırılmasında VTYS’nin dayandığı veri modeli yer alır. Bu modeller ilişkisel, ağ, hiyerarşik, nesne-yönelimli ve kavramsal modellerdir. Bu modellerden görsel ara yüze sahip “Ağ Modeli” tercih edilecektir.

5.2.2.7 Hazır Program Kütüphane Dosyaları

Facebook Api ve Twitter Api kütüphane dosyalarında mobil yazılımda kullanılmıştır.

##### 5.2.2.8 Case Araç ve Ortamları

### 5.3. Kodlama Stili

Yazılım kodlama stili konusunda herhangi bir standart bulunmamaktadır. İsimlendirilen değişken ve nesneler test uzmanı veya uzman kişiler tarafından incelendiğinde okunurluğu en yüksek düzeyde geliştirilecektir.

#### 5.3.1 Açıklama Satırları

Her bir modülün işlevlerini yazan kişi gibi bilgileri içerir. Örnek açıklama satırı aşağıdaki gibidir. Ayrıca uygulamanın kritik noktalarında açıklama satırıyla gerekli görülen notlar belirtilir.

#### 5.3.2 Kod Biçimlemesi

Kod biçimlenmesi açıklama satırlarına olan ihtiyacı azaltır. Kod biçimlemesinde önemli olan az satır değil kodun okunabilirliğidir. Bu uygulama bu kriterler göze almaktır.

#### 5.3.3 Anlamlı İsimlendirme

Anlamlı isimlendirme de kullanılan önemli teknik hangi değişkenlere hangi modülleriyle ilgili olduklarının belirtilerek adlandırılmasıdır.

#### 5.3.4 Yapısal Programlama Yapıları

Yapısal Programlama Yapıları okunabilirlik ve anlaşabilirdik bakımından önemlidir. Üç dala ayrılan Ardışık İşlem Yapıları, Koşullu İşlem Yapıları, Döngü Yapıları uygulamanın kodlanması sırasında sık sık kullanılacaktır.

### 5.4. Program Karmaşıklığı

#### 5.4.3 Programın Çizge Biçimine Dönüştürülmesi

Uygulamanın sonlandırılmasının ardından program karmaşıklığını çizge biçimine dönüştürerek aktarabiliriz.

#### 5.4.4 McCabe Karmaşıklık Ölçütü Hesaplama

McCabe Karmaşıklık Ölçütünün Hesaplanması için alt program dalları ve metotların oluşturulması gerekmektedir. Uygulamanın kodlanmasına başlamadığından dolayı McCabe Karmaşıklık Ölçütü hesaplanmamıştır.

### 5.5. Olağan Dışı Durum Çözümleme

Olağan dışı durumlar gerek kod yazım sürecinde gerekse testler sırasında gerçekleşebilir. Verilen hata kodlarının gözden geçirilip kısa sürede gözden geçirilmesi gerekir. Modüler olarak kodlanan programda bir modülde gerçekleşen hata diğerini etkilese bile çözümü kolay olacaktır

### 5.6. Kod Gözden Geçirme

5.6.1 Gözden Geçirme Sürecinin Düzenlenmesi

Gözden geçirme sürecinde aşağıdaki sıra uygulanacaktır;

* + Hataların bulunması, ancak düzeltilmemesi hedeflenir,
  + Olabildiğince küçük bir grup tarafından yapılmalıdır. En iyi durum deneyimli bir inceleyici kullanılmasıdır.
  + Kalite çalışmalarının bir parçası olarak ele alınmalı ve sonuçlar düzenli ve belirlenen bir biçimde saklanmalıdır.

#### 5.6.2 Gözden Geçirme Sırasında Kullanılacak Sorular

* Her bir modül kendi içinde doğru mu?
* Fonksiyonlar anlaşılabilir kodlanmış mı?
* Kod fazlalığı var mı(Algoritma güçlü mü)?

Gibi sorular ele alınacaktır.

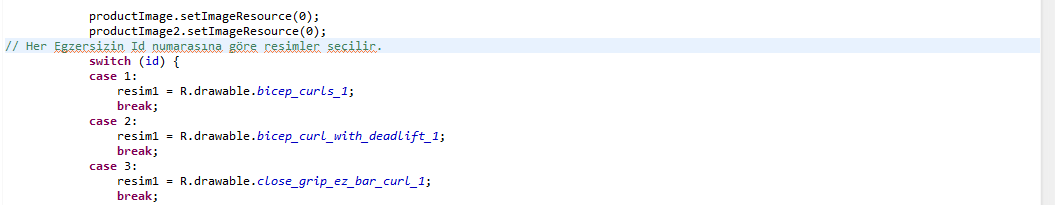
##### 5.6.2.1 Öbek Arayüzü

Öbek ara yüzüne örnek bir “Egzersiz Hakkında” modülünün kod öbeği aşağıda verilmiştir. 

Şekil 5.1 - Örnek kod öbeği.(Gerçek uygulamada değişkenlik olabilir)

##### 5.6.4.2 Giriş Açıklamaları

Giriş açıklamaları aşağıdaki şekildeki gibi yapılır.



Şekil 5.2 - Giriş Açıklamaları

##### 5.6.4.3 Veri Kullanımı

Şekil 5.2 ‘de “id” veri kullanımına örnektir. Egzersizlerin id numarasına göre resimleri seçiyoruz.

##### 5.6.4.4 Öbeğin Düzenlenişi

Şekil 5.2 ‘de de görüldüğü üzere Üst kısımda koşullu işlem yapısı tercih edilmiştir. (switch-case).

##### 5.6.4.5 Sunuş

Şekil 5.1 ve Şekil 5.2 ‘de de görüldüğü gibi düzenli kod tutumu ve yalın değişkenlere bağlı modüler programala kodlamanın okunabilirliğini kolaylaştırmaktadır.

## 6. Doğrulama Ve Geçerleme

### 6.1) Giriş

Doğrulama ve Geçerleme ’de uygulama kullanıcısının isteklerine yakınlıkla ölçülebilir.

### 6.2) Sınama Kavramları

Sınama kavramları aşağıdaki gibidir kısaca ele alınacak konular;

Birim sınama: Bağlı olduğu diğer sistemlerden soyutlanarak her bir birim kendi içinde ele alınır.

Alt Sistem Sınama: Modüllerin birleşmesiyle ortaya çıkar.

Sistem Sınama: Taranan modüllerden sonra üst düzeydeki sistem ile olan etkileşimleri ele alınmaktadır.

Kabul Sınama: Uygulamanın çalıştırılmadan önceki son sınamasıdır.

### 6.3. Doğrulama ve Geçerleme Yaşam Döngüsü

Doğrulama ve geçerleme yaşam döngüsü yazılım üretimini ve Gerçekleştirim evresine kadar olan tüm süreçleri ele alır. Tüm sistemin planlaması yapılır. Planlama genel olarak birim, alt birimler, bütünleştirme sistem ve kabul sınamalarının tasarımını içerir. Uygulama sonrası elde edilen veriler Yazılım doğrulama ve Geçerleme biçiminde sürekli raporlanır.

### 6.4. Sınama Yöntemleri

Sınama yöntemlerinde tercih edilen yol Beyaz Kutu Sınamasıdır.

Yapılabilecek denetimler arasında:

* Bütün bağımsız yolların en azından bir kere sınanması,
* Bütün mantıksal karar noktalarında iki değişik karar için sınamaların yapılması,
* Bütün döngülerin sınır değerlerinde sınanması,
* İç veri yapılarının denenmesi

Bulunur.

Mobil uygulamalarda modülerlik fazla buna bağlı olarak karmaşıklık azdır. Beyaz kutu Sınanmasının birincil tercih sebebi budur. Ayrıca Uygulamanın tek kişi tarafından gerçekleştirilmesi sınamaya olan sürenin kısıtlı olmasına sebeptir. Beyaz Kutu Yöntemi kısa sürede sınama yapmamıza olanak tanır.

### 6.5. Sınama Planlaması

### 6.6. Sınama ve Bütünleştirme Stratejileri

Sınama ve Bütünleştirme Stratejilerinde Yukarıdan Aşağı sınama gerçekleştirilecektir. Yukarıdan aşağı bütünleştirmede önce sistemin en üst düzeyi daha sonra aşağı doğru olan düzeyler sınanır. Öncelikle A Modülü incelenir eğer doğru sonuçlar veriyorsa B ve C modüllerine geçilir.

### 6.7. Sınama Belirtimleri

Sınama işleminin nasıl yapılacağına ilişkin ayrıntılar aşağıdaki gibidir.

Sınanacak Modüller

* Egzersiz Hakkında
* Kas Grubu görüntüleme
* Egzersiz Listeleme
* Egzersiz Arama
* Program oluşturma
* Program paylaşma

Sınama Türü

Beyaz Kutu yöntemi

Sınama Senaryoları

1.Teknik Sınama,

2.Biçimsel Sınama,

3.İşletimsel Sınama,

4.Senaryo Sınaması,

5.Kullanıcı Sınama

### 6.8. Yaşam Döngüsü Boyunca Sınama Etkinlikleri

**Planlama** Sistem Sınama Planı

**Çözümleme** Alt Sistem Sınama Planı

**Tasarım** Modül Sınama Planı Sınama Belirtimleri Sınama Eğitim Kılavuzları

**Gerçekleştirim** Modül sınama Bütünleştirme Sınama Sınayıcı Eğitim

**Kurulum** Kullanıcı Sınama Sınama Raporları

## 7) Bakım

### 7.1 Giriş

Sınama aşaması bittikten sonra yapılan kurulumla yazılımların bakım serüveni başlamış olur. Kullanıcı analizleriyle birlikte yapılan uygulamanın daha performanslı ve etkileşimli hali yayınlanır.

7.2 Kurulum

Kurulum işlemi sırasında kullanıcı uygulama hakkında bilgi sunulacak, sistem gereksinimleri belirtilecektir.

### 7.3 Yerinde Destek Organizasyonu

Uygulama için destek play.google sayfasında ve uygulama için oluşturulacak blog’dan sağlanacaktır.

### 7.4 Yazılım Bakımı

#### 7.4.1 Tanım

Bakım sırasında düzeltici bakım olarak adlandırılan yöntem izlenecektir. Kullanıcı dönüşleri ve pratikteki test sonunda elde edilen süreçler bu yöntemle işlenir. Ayrıca bakım sırasında sadece uygulama eksikleri giderilmeyecek zamana bağlı ihtiyaçların tedariklerinde yapılacaktır.

#### 7.4.2 Bakım Süreç Modeli

* Sorun Tanımlama/sınıflandırma
* Çözümleme,
* Tasarım,
* Gerçekleştirim,
* Sistem Sınama,
* Kabul Sınama
* Kurulum

Bakım süreç Modelinin süreçleridir.

## 8. SONUÇ

Mobil pazara yönelen teknoloji uygulama geliştiricilerini de etkilemiştir. Kullanıcı dostu akıllı telefonlara geliştirilmek üzere tasarlanan uygulamada spor yapmayı seven ve amaç edinmiş kullanıcılara yönelik olsa da egzersizlerin listelenmesi, egzersizlerin görüntülenmesi gibi çok fonksiyonlu seçeneklere sahip olması her Android kullanıcısı tarafından tercih edilebilir. Uygulama gerekli kitleye ulaşması halinde diğer Mobil işletim sistemlerine port edilecektir. Bu işletim sistemlerindeki değişiklikler uygulamanın ana yapısını değiştirmeden kullanıcıyı amaç alarak geliştirilecektir. Uygulamanın planlama, kodlama, tasarım ve gerçekleştirim aşamaları detaylı şekilde incelenmiştir. Bu aşamaları gerçekleştirirken oluşabilecek sorunlarda hangi yöntemlere başvurulacağı detaylı şekilde ele alınmıştır. Uygulama geliştirme sırasında bu dokümantasyon kodlama ve tasarım görevlerinde yer alacak personele yardımcı kaynak oluşturacaktır.

Uygulama haklarını satın almak isteyen şirket ya da kişiler uygulama hakkında detaylı bilgiye bu dokümantasyon sayesinde ulaşacak. Bu süreçte uygulamanın nasıl ele alındığını inceleme fırsatına sahip olacaktır.