Borsa Yatırım Fantezi Ligi

Rapor-2

Yazılım Mühendisliği

[Proje Linki](http://www.celalkutluer.com.tr/)

[Proje Repository](https://github.com/celalkutluer/Borsa_Projesi)

[Proje Raporları](https://github.com/celalkutluer/Borsa_Projesi_Raporlari/)

Proje Ekibi:

4. Grup Üyeleri

Celal KUTLUER

Beyzanur TAŞKÖPRÜ

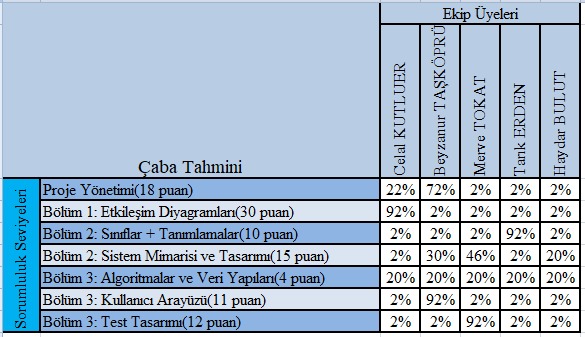
Merve TOKAT

Tarık ERDEN

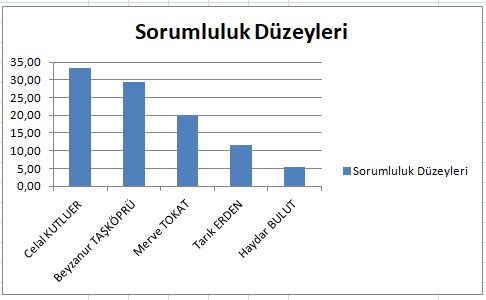
Haydar BULUT

12/04/2020

### **4.1. Çaba Tahmini**



### **4.2 Sorumluluk Düzeyi**



Bağlantı Linkleri:

[Proje Linki](http://www.celalkutluer.com.tr/)

[Proje Repository](https://github.com/celalkutluer/Borsa_Projesi)

[Proje Raporları](https://github.com/celalkutluer/Borsa_Projesi_Raporlari/)

Revizyon Geçmişi

|  |  |
| --- | --- |
| Revizyon No | Revizyon Tarihi |
| v1.0 | 29/03/2020 |
| V1.1 | 12/04/2020 |
|  |  |
|  |  |
|  |  |

**İçindekiler**

[**BÖLÜM 1** 5](#_Toc37598843)

[**SİSTEM ETKİLEŞİM DİYAGRAMLARI** 5](#_Toc37598844)

[**1. Giriş** 5](#_Toc37598845)

[**2. Diyagramlar** 5](#_Toc37598846)

[**3. Alternatif Çözüm Şeması** 11](#_Toc37598847)

[**4. Tasarım Desenleri** 15](#_Toc37598848)

[**BÖLÜM 2** 16](#_Toc37598849)

[**SINIF DİYAGRAMI VE ÖZELLİKLERİ** 16](#_Toc37598850)

[**1.Sınıf Diyagramı Ve Tanımlamalar** 16](#_Toc37598851)

[**1.1 Sınıf Diyagramı** 16](#_Toc37598852)

[**2. Veri Türleri Ve İşlem İşaretleri** 17](#_Toc37598853)

[**2.1 Tanımlamalar** 17](#_Toc37598854)

[**2.2 Varlıklar** 17](#_Toc37598855)

[**2.3 Kullanıcılar** 18](#_Toc37598856)

[**2.4 Loglar** 18](#_Toc37598857)

[**2.5 Ligler** 18](#_Toc37598858)

[**3. İZLENEBİLİRLİK MATRİSİ** 19](#_Toc37598859)

[**SİSTEM MİMARİSİ VE SİSTEM TASARIMI** 19](#_Toc37598860)

[**1. Mimari Tarzlar** 19](#_Toc37598861)

[**1.1 SQL** 20](#_Toc37598862)

[**2. Alt Sistemlerin Tanımlanması** 20](#_Toc37598863)

[**3. Alt sistemleri Donanıma Eşleme** 21](#_Toc37598864)

[**4. Kalıcı Veri Depolama** 21](#_Toc37598865)

[**5. Ağ Protokolü** 22](#_Toc37598866)

[**6.Global Kontrol Akışı** 22](#_Toc37598867)

[**6.1 Yürütme Düzeni** 22](#_Toc37598868)

[**6.2 Zamana Bağlılık** 23](#_Toc37598869)

[**7. Donanım Gereksinimleri** 23](#_Toc37598870)

[**7.1 İnternet Bağlantısı** 23](#_Toc37598871)

[**7.2 Disk Alanı** 24](#_Toc37598872)

[**7.3 Sistem Belleği** 24](#_Toc37598873)

[**7.4 İstemci Tarafı Donanım Gereksinimleri** 25](#_Toc37598874)

[**BÖLÜM 3** 26](#_Toc37598875)

[**KULLANICI ARAYÜZÜ TASARIMI VE UYGULAMASI** 26](#_Toc37598876)

[**1.** **Giriş Sayfası** 26](#_Toc37598877)

[**2.** **Ana sayfa** 27](#_Toc37598878)

[**2.1 Genel Üstbilgi** 28](#_Toc37598879)

[**3.** **Alım - Satım İşlem Ekranı** 29](#_Toc37598880)

[**4.** **Lig Ekranı** 29](#_Toc37598881)

[**5.** **Liderlik Gösterim Ekranı** 30](#_Toc37598882)

[**6.** **Kullanıcı Profil Ekranı** 31](#_Toc37598883)

[**BÖLÜM 4** 33](#_Toc37598884)

[**TEST TASARIMI** 33](#_Toc37598885)

[**1.Test Kutuları** 33](#_Toc37598886)

[**2. Ünite Testleri** 34](#_Toc37598887)

[**2.1 Veritabanı Yöneticisi** 34](#_Toc37598888)

[**2.2 Sipariş Yöneticisi** 35](#_Toc37598889)

[**3. Lig Teknik Direktörü** 37](#_Toc37598890)

[**4. Hesap Denetleyici** 40](#_Toc37598891)

[**5.Finans Bağdaştırıcısı** 42](#_Toc37598892)

[**6.Test Kapsamı** 43](#_Toc37598893)

[**7. Entegrasyon Testi** 43](#_Toc37598894)

[**BÖLÜM 5** 44](#_Toc37598895)

[**PROJE YÖNETİMİ** 44](#_Toc37598896)

[**1.Ekip Üyelerinin Bireysel Katkıları** 44](#_Toc37598897)

[**1.1 Beyzanur TAŞKÖPRÜ** 44](#_Toc37598898)

[**1.2. Celal KUTLUER** 44](#_Toc37598899)

[**1.3 Tarık ERDEN** 44](#_Toc37598900)

[**1.4 Merve TOKAT** 45](#_Toc37598901)

[**1.5 Haydar BULUT** 45](#_Toc37598902)

[**2. Proje Koordinasyon** 46](#_Toc37598903)

[**3.Çalışma Planı** 46](#_Toc37598904)

[**4. Sorumlulukların Dağılımı** 47](#_Toc37598905)

[**4.1. Çaba Tahmini** 47](#_Toc37598906)

[**4.2 Sorumluluk Düzeyi** 48](#_Toc37598907)

[**BÖLÜM 6** 48](#_Toc37598908)

[**KAYNAKÇA** 48](#_Toc37598909)

# **BÖLÜM 1**

# **SİSTEM ETKİLEŞİM DİYAGRAMLARI**

## **1. Giriş**

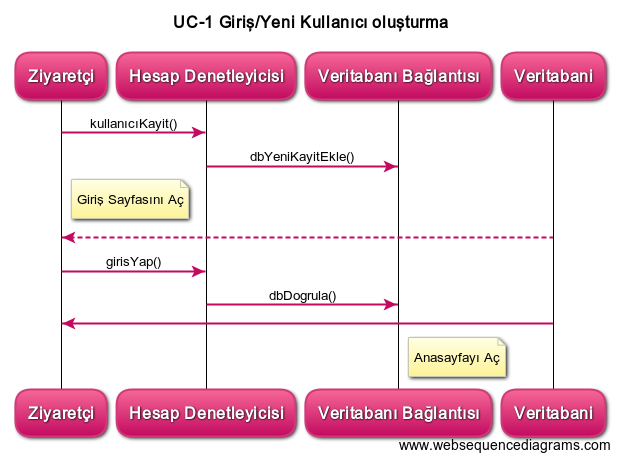
Aşağıdaki bölümdeki etkileşim diyagramları, yazılımımızın en önemli bölümlerindeki sistem etkileşimlerini açıklayacaktır. Her özel kullanım durumu için, sistemler ve veri tabanları arasındaki etkileşimleri özetleyeceğiz. Ayrıca, sistemlerin farklı senaryoları ele alacağı birçok durumu analiz edeceğiz. Yani, sistemin hem başarısızlık hem de başarı koşullarını nasıl ele aldığını gösterecektir. Bu web tabanlı ve veri tabanlı bir uygulama olduğundan, veri tabanı ve denetleyici hemen hemen her durumda kullanılır.

## **2. Diyagramlar**

**Use Case 1:**

UC-1 için dizi diyagramında gösterilen Ziyaretçinin sisteme dahil olabilmesi için iki seçenekle vardır. Giriş yapmak ya da kayıt olmak. Bir kullanıcı yeni bir hesap kaydetmeye çalışırsa, Hesap Denetleyicisi ile kullanıcı bilgileri arasında bağlantı kurulur. Daha sonra, veri tabanı bağlantı modülü aracılığıyla veri tabanında yinelenen oturum açma bilgilerinin bulunmadığından emin olmaya çalışır(burada e-posta adresini kontrol eder) ve eğer değilse yeni kullanıcı bilgilerini veri tabanına kaydeder. Bu gerçekleştikten sonra kullanıcıya bir onay e-postası gönderilir. Ardından Hesap Denetleyicisi Giriş Sayfasına gider.

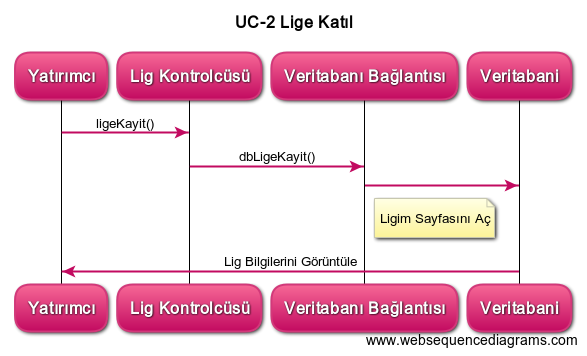
Bir kullanıcı oturum açmaya çalışırsa, Hesap Denetleyicisi DB Bağlantısı modülü aracılığıyla veri tabanında bulunan ayrıntılarla oturum açma ayrıntılarını doğrulamaya çalışır. Ayrıntılar doğru bir şekilde eşleşirse, Hesap Denetleyicisi misafiri kullanıcı(yatırımcı) moduyla Ana sayfaya gönderir.



Şekil 1: UC-1

**Use Case 2:**

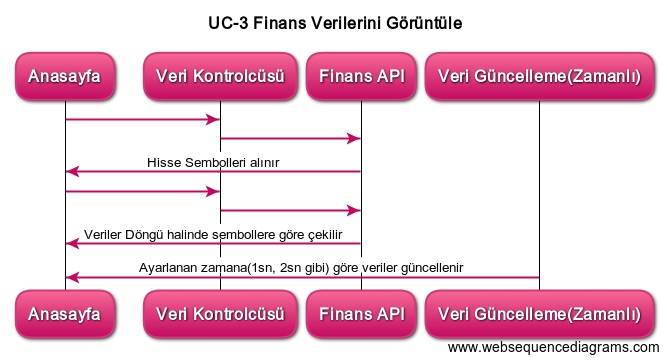
UC-2 dizi diyagramında gösterilen, bir yatırım liginin nasıl oluşturulacağının akışıdır. Bir yatırımcı Ligler sayfasından bir lig oluşturmayı seçtiğinde Lig Kontrolcüsü ile temasa geçilecektir. Lig Kontrolcüsü Ligler arasında bağlantı kurulur. Bundan sonra, ligi yaratacak ve DB Bağlantısı üzerinden veri tabanında işleyecektir



Şekil 2: UC-2

**Use Case 3:**

Piyasa verileri ana sayfa da görüntülenir. Veriler veya Hisselere ait semboller ile arama yapılabilir. Burada sayfa açıldığında aracılığı ile Finans APİ’ sine bağlanılır ve veriler alınır.

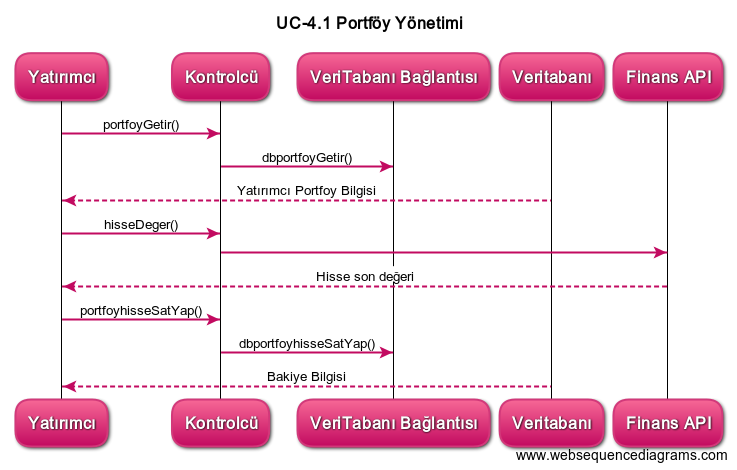


Şekil 3: UC-3

**Use Case 4:**

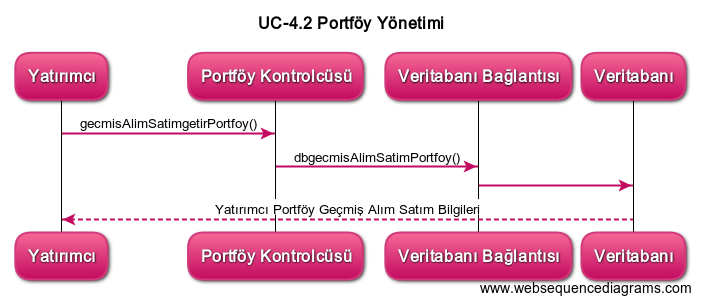
Yatırımcı, Portföyünü görüntüleyebilmeli ve üzerinde değişiklik yapabilmelidir. Portföy yönetimi 2 şekilde sağlanır. Bunlar;

* Kullanıcı portföyünü görüntülemek için Aktif Varlıklarıma tıkladığında, Portföy Denetleyicisi Veri tabanı Bağlantısı modülü aracılığıyla yatırımcı portföyü stoklarını veri tabanından alır. Satım yapılmak istenirse Portföy Denetleyicisi Veri tabanı Bağlantısı modülü aracılığıyla veri tabanını günceller.



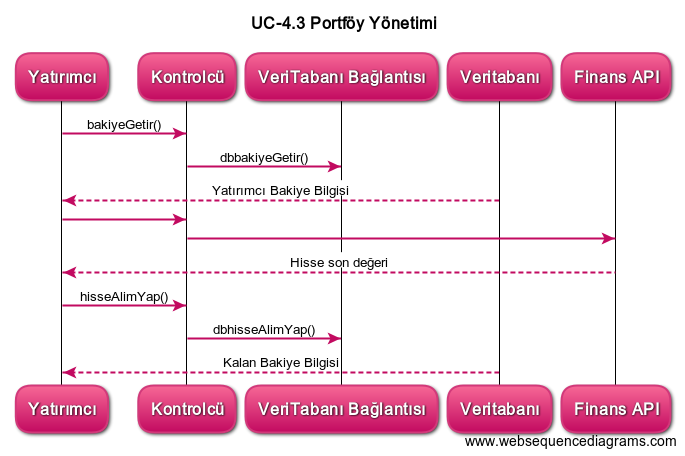
Şekil 4: UC-4.1

* Yatırımcının daha önce alıp sattığı portföy verilerinin görüntülenmesi için yatırımcı Geçmiş Alım-Satımlar a tıkladığında Portföy Denetleyicisi Veri tabanı Bağlantısı modülü aracılığıyla bilgileri veri tabanından alır.



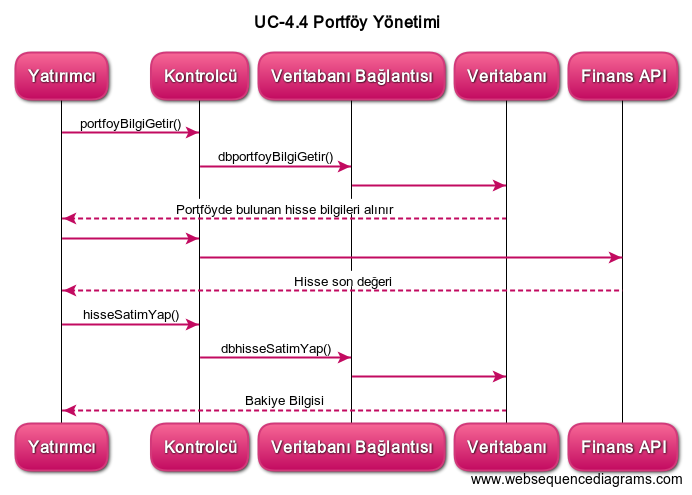
Şekil 5: UC-4.2

* Ana sayfa üzerinden portföy alım satımları yapılabilir Kontrolcü Veri tabanı Bağlantısı aracılığı ile Veri tabanından bakiye bilgisini alır. Hisse alımı sırasında Kontrolcü Veri tabanı Bağlantısı aracılığı ile Veri tabanına alınan hisse bilgilerini kaydeder ve kalan bakiye bilgisini gönderir.



Şekil 6: UC-4.3

* Bu alanda satım işlemi de yapılabilir Kontrolcü Veri tabanı Bağlantısı aracılığı ile Veri tabanından portföyde bulunan hisse bilgilerini alır. Hisse satımı sırasında Kontrolcü Veri tabanı Bağlantısı aracılığı ile Veri tabanına alınan hisse bilgilerini düşer ve bakiye bilgisini gönderir.

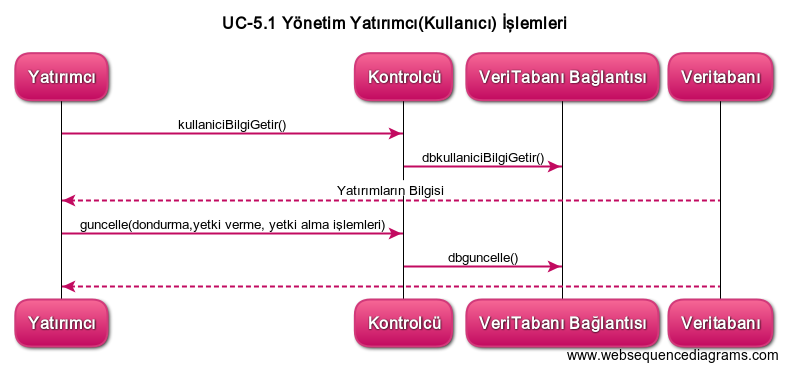


Şekil 7: UC-4.4

**Use Case 5:**

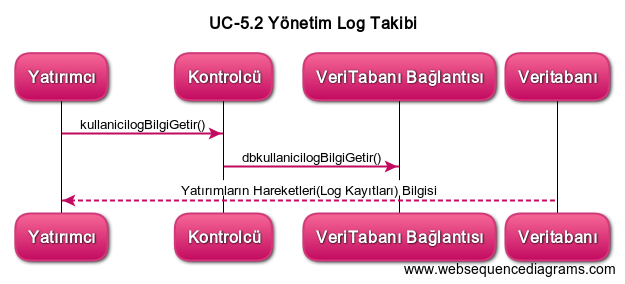
* Yönetim alanı kullanıcı (yatırımcı) ların yetkilendirilmesi, takibi ve mali kontroller için tanımlanacaktır.

**Kullanıcı işlemleri;**



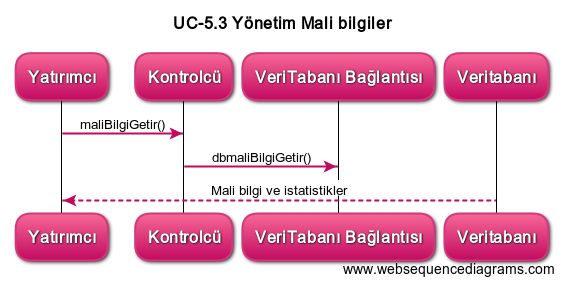
Şekil 8:UC-5.1

**Log Kayıtlar;**



Şekil 9:UC-5.2

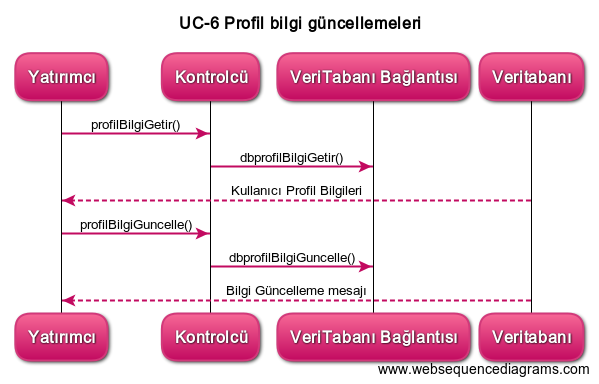
**Mali Bilgiler;**



Şekil 10:UC-5.3

**Use Case 6:**

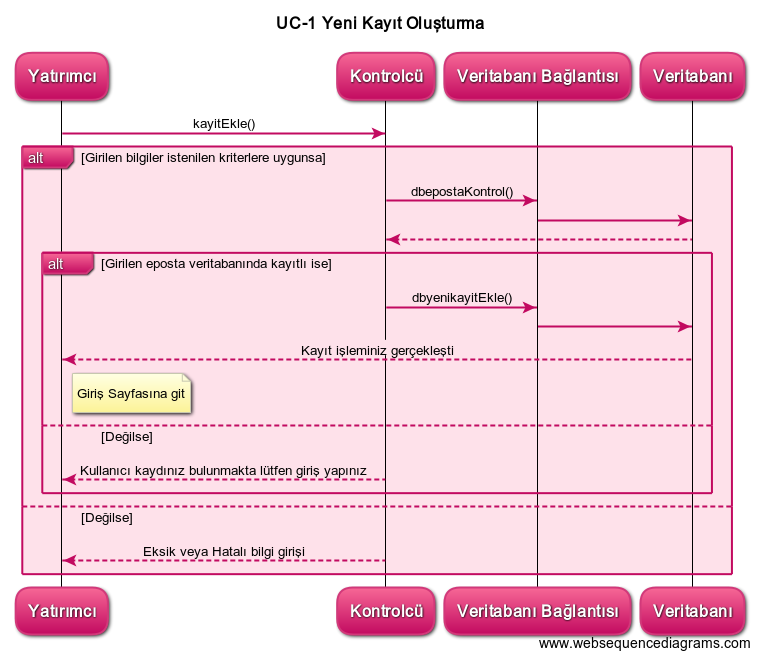
Profil Bilgileri alanı kullanıcının kayıt olurken verdiği bilgilerin değiştirilmesi ve şifre değiştirme işlemleri için kullanılır. Kontrolcü veri tabanı bağlantısı aracılığı ile veri tabanından profil bilgilerini getirir. Değişiklik işlemi yine veri tabanı bağlantısı aracılığı ile veri tabanına kaydedilir.

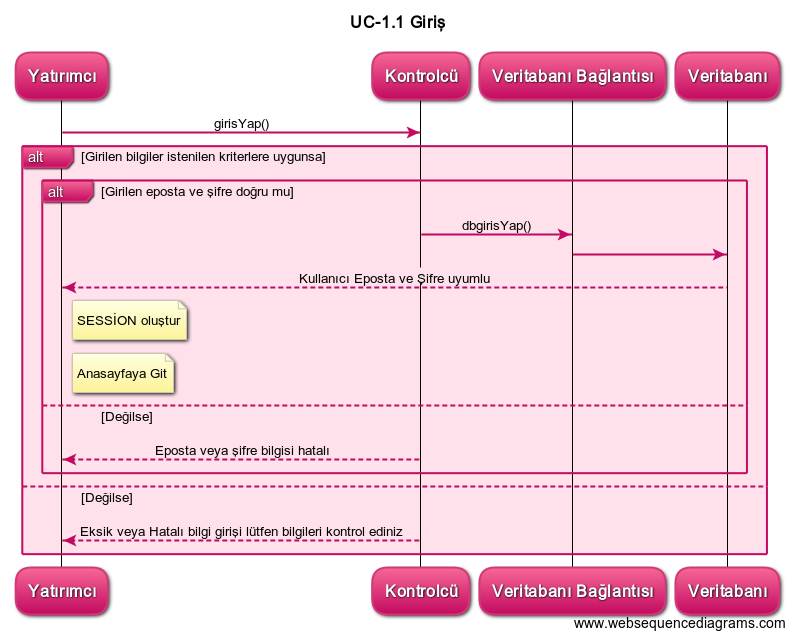


Şekil 11: UC-6

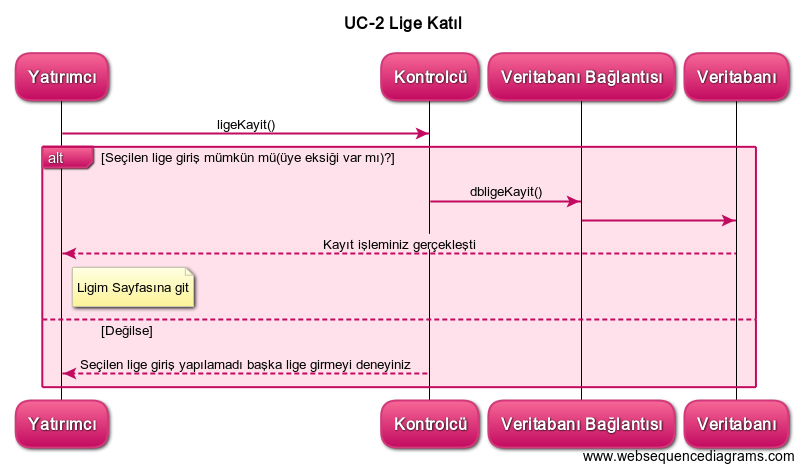
## **3. Alternatif Çözüm Şeması**

Yazılım tasarımı, ilk fikrinizi seçmek ve onunla devam etmekle ilgili olmamalıdır. Eldeki göreve alternatif çözümler düşünmeniz ve bilinen kriterlere göre en iyisini seçmeniz gerekir.

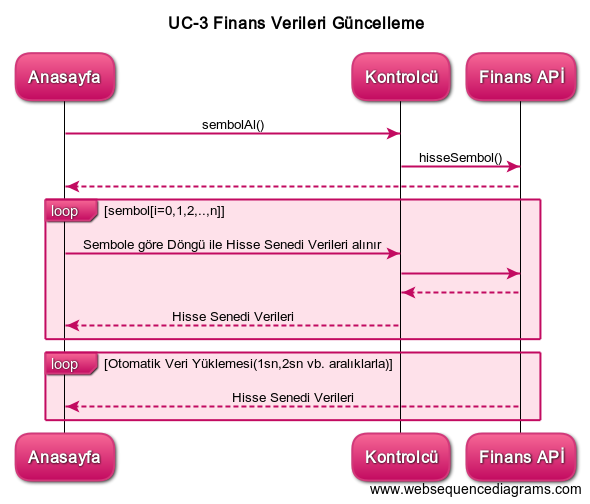


Şekil 12: UC-1 alternatif

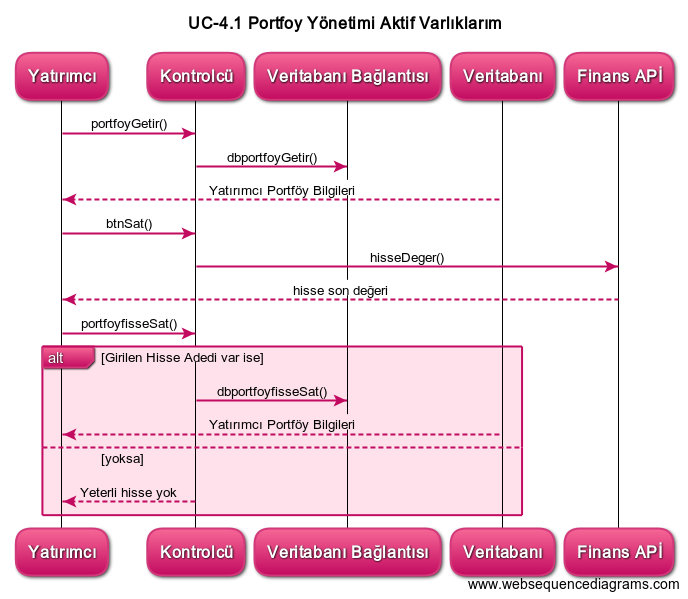
Şekil 13: UC-1.1 alternatif



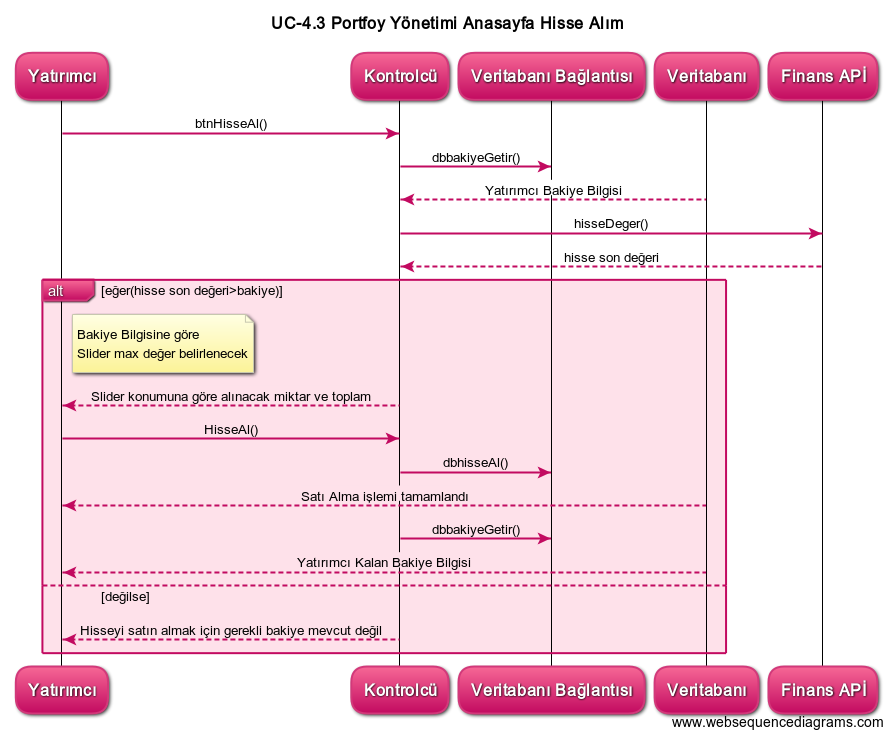
Şekil 14: UC-2 alternatif



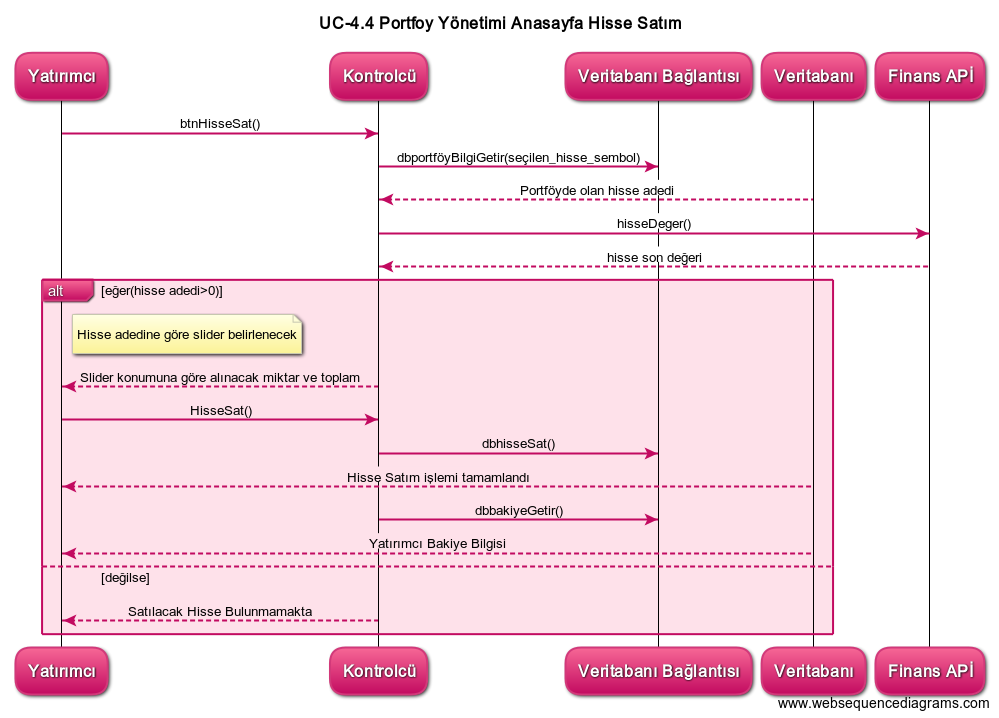
Şekil 15: UC-3 alternatif



Şekil 16: UC-4.1 alternatif



Şekil 17: UC-4.3 alternatif



Şekil 18: UC-4.4 alternatif

## **4. Tasarım Desenleri**

Kimlik doğrulama, görüntü oluşturma ve nesne modelleme verimli sayfa gibi şeyler için işlevsellik sağlamak amacıyla çeşitli standart ve standart olmayan tasarım desenleri kullanılmıştır.

**Nesne İlişkisel Model Kalıbı**

Bir veri tabanı erişim tasarım modelinin akıllı bir uygulaması olan Nesne İlişkisel Model (ORM) modeli, yalnızca sistemde kullanılan kalıcı depolama teknolojileriyle etkileşim kurmak için kullanılmıştır. Bu model, veri tabanına özgü sorguların sabit kodlanmasına gerek kalmamasının en büyük avantajını sundu. ORM aracılığıyla yapılan tüm istekler, halihazırda kullanılan DB sisteminin diline çevrilir ve veriler doğrudan nesne formunda döndürülür. Doğrudan sorgular yazma gereksiniminin olmaması, büyük bir yan etkiye, yani çeşitli geliştirme uygulamalarının farklı aşamalarında test edilmesine izin veren veri tabanı agnostisizmine yol açar. Geliştirme sırasında SQLite, geliştiriciler makinesindeki hafif ayak izi için kullanıldı, daha sonra üretim için MySQL, daha fazla miktarda veriyle uğraşırken çok daha verimli olduğu için kullanıldı. Bu tasarım, sayısız saat geliştirme süresinden tasarruf ederek gelişimimizi kesinlikle geliştirdi.

**Responsive UI Patern**

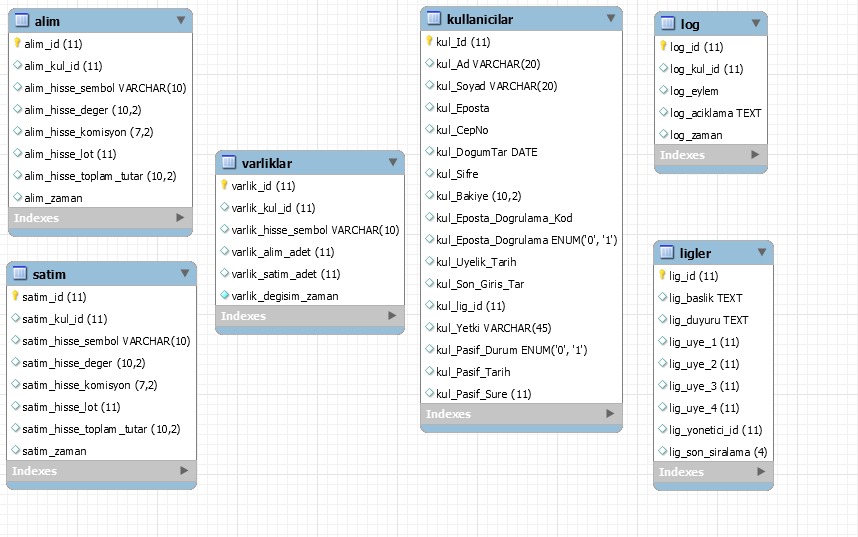
Bootstrap UI çerçevesi, görsel sunumu içerik ve kullanıcı deneyiminden tamamen ayıran bir tasarım deseni uyguladı. Bu, masaüstlerinden akıllı cihazlara kadar farklı istemci cihazlarına uyarlanmış güzel ve duyarlı bir tasarım sağladı. Desen, sayfa Javascript ve CSS kullanarak tarayıcıda işlenirken HTML5'in esnek biçimlendirmesinin avantajını kullanır. Bu, ekibimizin fazladan uygulama çabası olmadan hızla büyüyen mobil kullanıcıları hedeflemesini sağladı. Ayrıca, ilk sayfa oluşturma sırasında minimum işlem yapıldığından ve sayfa zaten kullanıcı tarafından görüldüğünde çoğunlukla eş zamansız olarak yapıldığından, doğası gereği daha hızlı bir kullanıcı deneyimi üretmiştir. Bu hedeflerin her ikisine de aktif olarak ulaşmaya çalıştık.

# **BÖLÜM 2**

# **SINIF DİYAGRAMI VE ÖZELLİKLERİ**

## **1.Sınıf Diyagramı Ve Tanımlamalar**

### **1.1 Sınıf Diyagramı**



Şekil 19 : Sınıf Diyagramı

## **2. Veri Türleri Ve İşlem İşaretleri**

### **2.1 Tanımlamalar**

Alım : Alınan hisseler burada yer alır.

Alim\_id(11): Alınan hissenin adı

Alim\_kul\_id(11): Hisseyi alan kişinin kullanıcı adı

Alim\_hisse\_sembol(varcahar(10)): Hisseyi alan kullanıcı veya kullanıcının şirketinin sembolü

Alim\_hisse\_deger(10,2):alınan hissenin değeri

Alim\_hisse\_komisyon(7,2): alınana hissedeki komisyon oranı

Alim\_hisse\_lot(11): alınan hisseden kaç adet alındığını gösterir

Alim\_hisse\_toplam\_tutar(10,2);alınan hissenin toplam tutarını gösterir.

Alim\_zaman: hissenin ne zaman alındığını gösterir.

Satım : satılan hissenin yer aldığı yer burasıdır.

Satim\_id (11) : satılan hissenin adı

Satim\_kul\_id (11) : hisseyi satan kişinin kullanıcı adı

Satim\_hisse\_sembol(varchar(10)): hisseyi satan kişi veya şirketin sembolü

Satim\_hisse\_deger(10,2); hissenin satım değeri

Satim\_hisse\_komisyon(7,2): satılan hissedeki komisyon değeri

Satim\_hisse\_lot(11): satılan hissenin kaç adet olduğu

Satim\_hisse\_toplam\_tutar(10,2): satılan hisselerin toplam tutarı

Satim\_zaman: hissenin satıldığı tarih

### **2.2 Varlıklar**

Varlik\_id(11): var olan hissenin id si

varlik\_kul\_id(11): var olan hissenin adı

varlik\_hisse\_sembol(varchar(10)): Var olan hisseye sahip kişi veya şirketin sembolü

varlik\_alim\_adet(11): Kaç adet hisse alınmış onu gösterir

varlik\_satim\_adet(11):Kaç adet hisse satılmış onu gösterir

varlik\_degisim\_zaman: Hisselerin zaman a göre değişimi

### **2.3 Kullanıcılar**

Kul\_id(11): Kullanıcı id si

Kul\_Ad(varchar(20)): Borsaya giren kulanıcının adı

Kul\_Soyad(varchar(20)): Borsaya giren kullanıcının soyadı

Kul\_Eposta: Kullanıcının e-postası

Kul\_Cepno: Kullanıcının telefon numarası

Kul\_DogumTar(Date): Kullanıcının doğum tarihi

Kul\_Sifre: Kullanıcının sisteme giriş şifresi

Kul\_Bakiye(10,2): Kullanıcının hesaptaki bakiyesi

Kul\_Eposta\_Dogrulama\_Kod: Kullanıcı üye olurken e-postasına giden kod

Kul\_Eposta-Dogrulama(Enum(“0”, “1”)):Doğrulama kodu

Kul\_Uyelik\_Tarih: Kullanıcının üye olduğu tarih

Kul\_Son\_Giris\_Tarih: Kullanıcının en son sisteme giriş yaptığı tarih

Kul\_lig\_id(11): Kullanıcının mevcut bulunduğu yatırım liginin adı

Kul\_Yetki(varcahar(45)): Kullanıcıya verilen yetkiler

Kul\_Pasif\_Durum(Enum(“0”, “1”)):

Kul\_Pasif\_Tarih:

Kul\_Pasif\_Sure:

### **2.4 Loglar**

Log\_id: Giriş yapan kullanıcının id si

Log\_kul\_id: Giriş yapan kullanıcının adı

Log\_eylem: Giriş yapan kullanıcının yaptığı gerçekleştirdiği eylemler

Log\_aciklama: Giriş yapan kullanıcının açıklaması

Log\_zaman: Kullanıcının ne zaman giriş yaptığı

### **2.5 Ligler**

Lig\_id: Yatırım yapılacak lig id si

Lig\_baslik: Ligin başlığı

Lig\_duyuru: Lig de yapılan duyurular

Lig\_uye\_1: Yatırım ligindeki uye-1

Lig\_uye\_2: Yatırım ligindeki uye-2

Lig\_uye\_3: Yatırım ligindeki uye-3

Lig\_uye\_4: Yatırım ligindeki uye-4

Lig\_yonetici\_id: Yatırım liginin yöneticisi(admin)

Lig\_son\_siralama: Yatırım liginde oluşan sıranın en son hali

## **3. İZLENEBİLİRLİK MATRİSİ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASS | ALIM | SATIM | LOG | LİGLER |
| ALIM | X | X | X | X |
| SATIM | X |  | X | X |
| VARLIKLAR | X | X |  |  |
| KULLANICILAR | X | X | X | X |

Şekil 20: İzlenebilirlik Matrisi

Alım ve satım tüm diğer bölümlerle bağlantılıdır. Çünkü tüm işlemleri neredeyse alım ve satım yapan kişi yapar.

Varlıklar ise sadece alım ve satımla ilgilidir.

Kullanıcılar da tüm bölümlerle bağlantılıdır. Çünkü alım, satım, giriş yapma ve lige katılma kullanıcı tarafından gerçekleştirilir.

# **SİSTEM MİMARİSİ VE SİSTEM TASARIMI**

## **1. Mimari Tarzlar**

Yazılımı gerçekleştirirken bazı mimari tarzlardan yararlanacağız bu kısımda bu mimari tarzları ele aldık.

### **1.1 SQL**

SQL (Structured Query Language – Yapılandırılırmış Sorgu Dili) kavramı, bir nevi veri tabanı yönetimini ifade eder. Burada amaç verileri yönetebilmek ve tasarlayabilmektir. Esasen SQL bir programlama dili olmasa bile pek çok kişi tarafından bu şekilde kabul görür. SQL aracılığıyla sadece veri tabanı üstünden işlem gerçekleştirilir.

SQL sayesinde veriler üstünde esnek bir şekilde işlem yapmak mümkündür. Hem çok düzenli hem de pratik şekilde sonuç alınabilmektedir.

Projemizle finans verilerini, kullanıcı bilgileri gibi verileri çekerken sql den yararlanacağız.[1]

#### **1.1.1 Veri Merkezli Tasarım**

Veriler sql management veri tabanında tutulacaktır. Kullanıcı her oturum açtığında, kendi kişisel ana bilgisayarına erişmesi gerekir. Verilerin erişilebilecek şekilde depolanması gerekir gerektiğinde çoklu alt sistemler. Bu yöntemi kullanırken verileri tutabiliriz

#### **1.1.2 İstemci Sunucu Erişimi**

Kullanıcı sürekli olarak ara yüzle etkileşime girecektir. SQL ile tüm etkileşimler gerçekleştirilir, böylece, istemci sunucu bazında. Kullanıcı birincil istemci olarak kalır ve bu nedenle sürekli olarak diğer alt sistemlerle etkileşime girebilir. Kullanıcı tarafından erişilmesi gereken yerlere düzgün iletişim sağlanır. Ligler, tablolar oluşturulurken ara yüz ve veriler arasındaki iletişimi sağlar.

#### **1.1.3 REST**

Temsili durum transferi (REST) Web Hizmetleri oluşturmak için kullanılacak kısıtlamalar kümesini tanımlayan bir yazılım mimari tarzıdır . RESTful Web Hizmetleri olarak adlandırılan geri kalan mimari stile uygun Web hizmetleri, internetteki bilgisayar sistemleri arasında birlikte çalışabilirlik sağlar . RESTful Web Hizmetleri, istekte bulunan sistemlerin, tekdüzen ve önceden tanımlanmış durum bilgisi olmayan işlemler kümesini kullanarak Web kaynaklarının metinsel gösterimlerine erişmesine ve manipüle etmesine izin verir. SOAP Web hizmetleri gibi diğer Web Hizmetleri türleri, kendi rasgele işlem kümelerini ortaya çıkarır. [2]Projemizi gerçekleştirirken RESTful mimari tarzından yararlanacağız.

## **2. Alt Sistemlerin Tanımlanması**

Sistemler genellikle kullanıcı arayüzü ve kullanıcı ile nesne etkileşimi içerir. Arka uç referans alacak tüm veritabanı şeması, uygulama ve ilgili donanım ile etkileşimler. Ayrıca dahili sistemimizin başarısı için gerekli olan ilişkisel olmayan öğelerdir.

Ön uç sistemleri resmen sadedir. Görünümleri ve belirli verileri görüntüleyen kullanıcı arayüzü kullanıcıya birden fazla platformda dahil edilmiştir. Yani, farklı eşlemeler içerecek ve

iOS ve Android için olduğu kadar Web için de yerel uygulamalar. Ön uç sistemi tutarlılığı sağlamak için arka uç sistemle sürekli iletişimi sürdürmek zorunda kalacak. Aşağıdaki bilgileri başarıyla iletebilmesi gerekir. Komutları kullanıcı tarafından verilir ve arka uç ile iletişim kurar.

Arka uç sistemi denetleyici ve veritabanı alt sistemlerine bölünür. Buna ek olarak, daha önce belirtildiği gibi finansal alım sistemine ve kuyruk sistemlerine sahip olacağız. Bu nedenle, komut işlemenin büyük kısmı arka uç alt sistemimiz tarafından işlenir. Arka uç sistemi yalnızca iletişim kurmakla kalmamalı kendi içindeki alt sistemler arasında, ancak aynı zamanda ön uç UI sistemiyle de iletişim kurması gerekir komutlara yanıt vermek ve aynı zamanda ilişkili olmayan sistemlerle de iletişim kurmak.

Alt sistemi daha da yıkarak, finansal erişim sisteminin ve kuyruk sisteminin önemini vurguluyoruz. Aynı zamanda sıraya koyma ve doğru komutların doğru işlendiğinden emin olmak için arka uç işlemleri ve monitörleri bu modüllerin başarısı, tüm arka uç sistemin başarısı ve Sistemler arasındaki iletişim, yazılımın genel başarısı için çok önemli olacaktır.

## **3. Alt sistemleri Donanıma Eşleme**

Sistemin birden fazla cihazda çalışması gerekir. Özellikle, sistemin vale Yardımcısı arayüzü ile iletişim kurabilmesi gerekecektir. Toplanan bilgilerin geçerli olduğundan ve veritabanına doğru şekilde saklandığından emin olmak sorumludur.[3]

Sistem iki ayrı bölüme ayrılır: istemcilerin web tarayıcısında çalıştırılan bir ön uç sayfası ve veritabanının sunucu tarafında çalışan bir arka uç. Ön uç ana sistem ve istemci arasındaki grafik kullanıcı arabirimi (GUI). Ön uç şunlardan sorumludur. Piyasa emirlerini onaylama gibi amaçlar için GUI ve veri tabanı arasındaki iletişim ve bir yatırımcının portföyünün güncellenmesi. Ön uçtaki bu değişiklikler arka uçta yansıtılır sunucu tarafı. Arka uç, piyasa emirlerinin ve güncellemelerinin uygun şekilde yürütülmesini ele alacak.

## **4. Kalıcı Veri Depolama**

Veri tabanı basit olarak bilgi depolayan bir yazılımdır. Bir çok yazılım bilgi depolayabilir ama aradaki fark, veri tabanın bu bilgiyi verimli ve hızlı bir şekilde yönetip değiştirebilmesidir. Veri tabanı, bilgi sisteminin kalbidir ve etkili kullanmakla değer kazanır. [4] Yazılımımız geliştirilmiş ve güncellenmiş verilere bağlı olduğundan, veri tabanı şemamızın ilgili tüm nesneleri doğru bir şekilde temsil etmesi projemiz açısından önemli yer tutar. Veriler tüm kullanıcı verilerini, hisse alma verilerini, hisse satma verilerini, lig ayarlarını, en çok düşen ve yükselen tabloların verileri ve diğer tüm ilgili verileri doğru bir şekilde depolamalıdır.

Günümüzde verileri toplamak, saklamak ve değerlendirmek eskisine göre çok daha kolaydır. Bir analizi verimli yapabilmek için en önemli olan etken, mümkün olduğunca çok verinin toplanabilmesidir. Veri analizinin elle veya yavaş bilgisayarlarla yapıldığı zamanlarda örnekleme önemli bir konuyken, bugün hızlı çalışan sayısal bilgisayar yardımıyla olaylar ve ilişkili hadiseler hakkında kesintisiz veri toplamak ve bunları çabuk bir şekilde analiz etmek mümkün olmaktadır. Böylece, örnekleme yöntemiyle yapılacak analizlerde, örneklerin bazı önemli detayları dikkate almaması nedeniyle oluşabilecek yanlışlar engellenebilmektedir. Spektral analizin borsa uygulamalarıyla ilgili pek çok ülkede çalışmalar yapılmıştır; ancak, BİST’in genç bir borsa olması ve dolayısıyla bu analiz için yeterli verinin yokluğu, analizin ülkemizde yapılabilirliğini kısıtlamıştır.

Borsa Yatırım Fantezi Lig’inde ilişkisel veri tabanı MySQL'i yoğun şekilde kullanacaktır. İlişkisel veri tabanları, bu yazılımın ihtiyaçları için çok daha pratiktir. Veriler şu şekilde alınacaktır. Bir kullanıcı oturum açmaya çalışırsa ilgili veriler veri tabanımızla uyuşuyorsa kullanıcı giriş yapmış olur. İşlem onaylandıktan sonra veri tabanında saklanılacak ve değişiklik yapılan bilgiler de ara yüzde gösterilecektir. Gösterilmiş olan verilen isteğe göre güncellenip ekleme ve silme yetkisini de gerçekleştirecektir. Borsa İstanbul’ un veri yayınını sadece yetkilendirdiği kuruluşlar aracılığı ile yapması ve bu kuruluşlarında veriler için yüksek fiyatlar istemesi nedeniyle 100 adet yerin verisi 15 dakika gecikmeli olarak paylaşılan siteler üzerinden sağlanacaktır. Bir teknik analist portföyünü oluşturmakla ilgili bir karar problemi ile karşı karşıyadır. Kullanıcı uygulamada veri tablosu olarak kullandığımız bilgileri kolayca elde edebilir.

## **5. Ağ Protokolü**

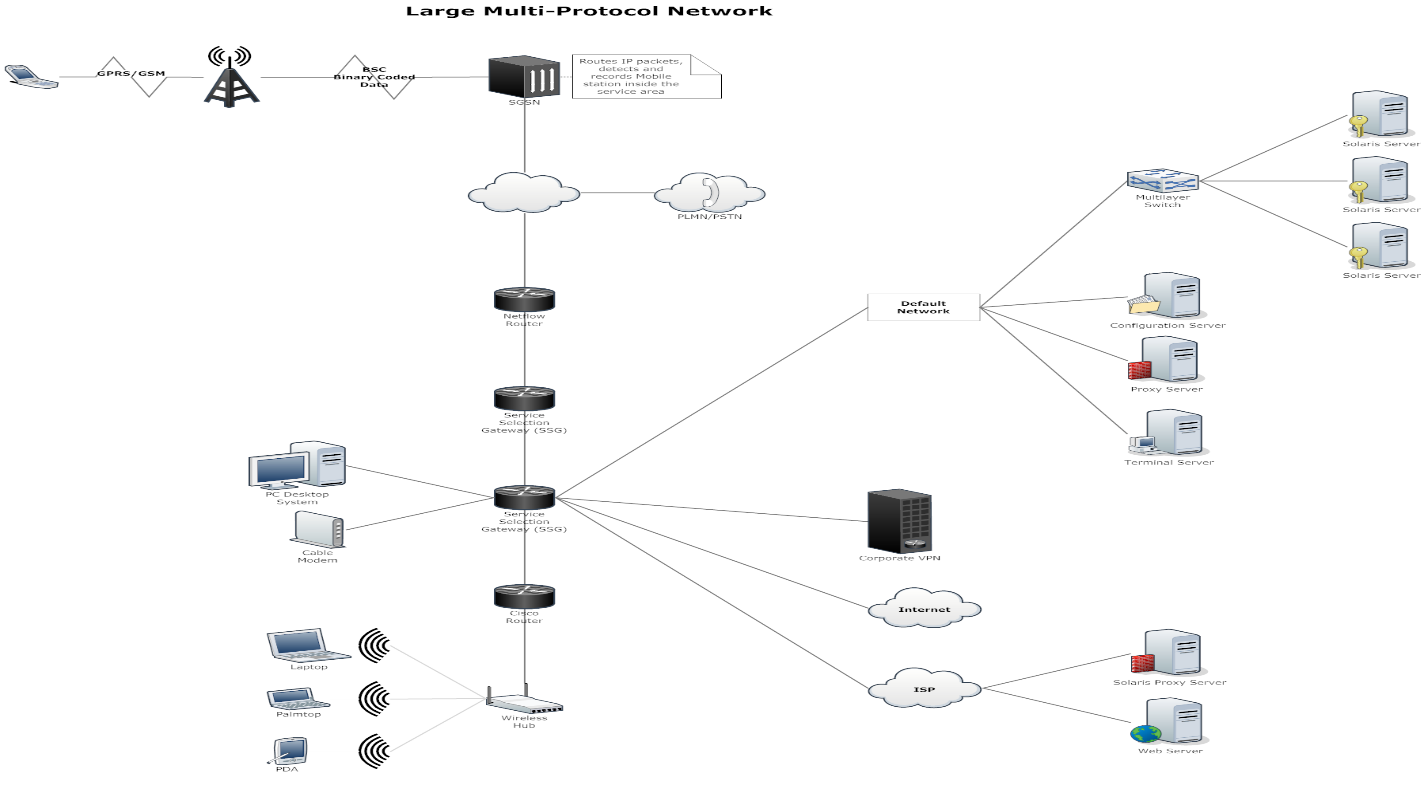
Bu tür yazılımlar da genellikle http kullanıldığı gibi Borsa Yatırım Fantezi Lig’i projesinde de HTTP kullanılmıştır. HTTP protokolü ağ üzerinden web sayfalarının görüntülenmesini sağlayan protokoldür. HTTP protokolü istemci (PC) ile sunucu (server) arasındaki alışveriş kurallarını belirler. Port olarak ise 80 portunu kullanır. İstemci sunucuya bir istek gönderir. Bu istek Internet Explorer, Google Chrome veya Mozilla Firefox gibi web browser’lar aracılığıyla iletilir. Sunucu bu isteği alır ve Apache veya IIS gibi web sunucuprogramları aracılığıyla cevap verir. [5] İstemci konumunda olan bilgisayarlar ile bilginin sağlandığı sunucular arasında bir köprü oluşturan http sistemi aynı zamanda bu kaynaklar arasındaki bilgi alışverişinin kurallarını da belirlemektedir. Bu kuralların ilki, verinin güvenli bir şekilde aktarımıdır. Bizim de önceliğimiz güvenilir bir şekilde aktarım olduğu için http protokolü önceliğimiz oldu.

## **6.Global Kontrol Akışı**

### **6.1 Yürütme Düzeni**

Bu sistemin en önemli özelliği olayların uygulanmasıdır. Sistemin oluşması için kullanıcı tarafından çalıştırılması gerekir. Kullanıcı veya sistemin herhangi bir parçasında olayın çalışması kullanıcı tarafında gerçekleşmesi sonucu oluşur. Sistemimiz olaya bağlı olarak çalışmaktadır örneğin kişi butona tıkladığında alım satım gerçekleşmesi gibi.

Kullanıcı başlamadan önce giriş yapmalıdır hesabı yoksa yeni bir kayıt oluşturmalıdır.



Şekil 21: Yürütme Düzeni

### **6.2 Zamana Bağlılık**

Genel olarak sistemimiz olaya bağlı bir zamanlı sistemdir. Zamana bağlı sistemimizde hisse verilerinin zamanlı olarak 3 saniye 2 saniye ’de bir güncellenmesidir.

## **7. Donanım Gereksinimleri**

Bir web projesinin bir browser’da(chrome,opera vb.) çalıştırılabilmesi için gerekli sistemin minimum özelliklerini içermesidir.

Sistem gereksinimi nedir ?

Bir yazılım veya oyunun sizin bilgisayarınızın sahip olduğu donanımda çalışıp çalışamayacağını gösteren yani yazılım veya oyunun çalışması için gereken donanım bilgileridir. İndirdiğiniz ve kullandığınız bütün programlar bunlara işletim sistemlerinde dahil hepsi yazılımsal ve donanımsal açıdan bir gereksinime sahiptir.[2]

### **7.1 İnternet Bağlantısı**

Bu projenin temel işlemlerinden birini kullanabilmesi için işlemlerden kastımız kullanıcının hisse verileri, kullanıcı girişi, kayıt bilgilerinin tutulduğu vb. işlemleri gerçekleştirmek için minimum düzeyde bir internet bağlantısına sahip olmalıdır.

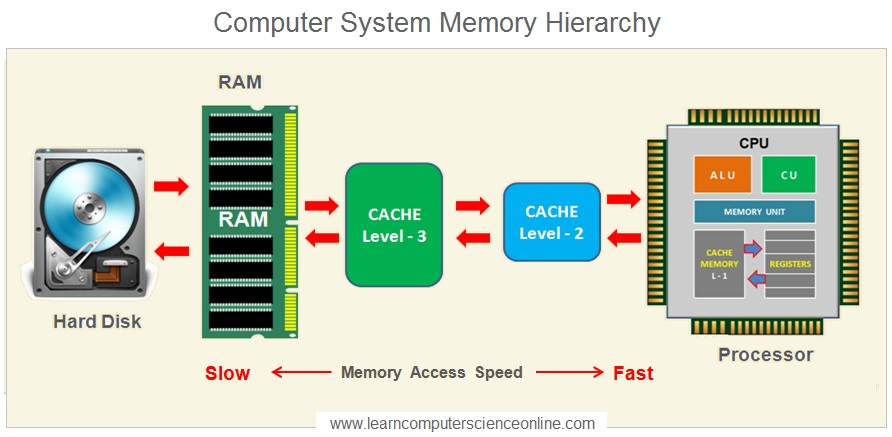
### **7.2 Disk Alanı**

Sunucu verilerini(bilgilerini) ve veritabanı bilgilerini saklayabilmesi için bir disk alanına ihtiyaç vardır. Saklanan bu veriler için minimum 15 GB kapasiteye sahip olması yeterli olacaktır.

### **7.3 Sistem Belleği**

Bu ana başlık altında bilgilerin daha iyi verim alabilmek için ve belleği en az kullanarak yönetimini gerçekleştirmektir. Bu başlık altındaki en önemli iki öğe hız ve az yer kaplamaktır. Ayrıca sistem belleği için en az 512 Mb bir yerimiz olması yeterli olacaktır.

Sistem belleği, RAM ya da bellek olarak adlandırın; bellekler teknik anlamda basit donanımlar gibi görünse de bu donanımı parçasında da pek çok teknik detay yer almaktadır. Saat hızı, Kanallar, Voltaj, Soğutucu, Kapasite, Registered DIMM gibi.[5]

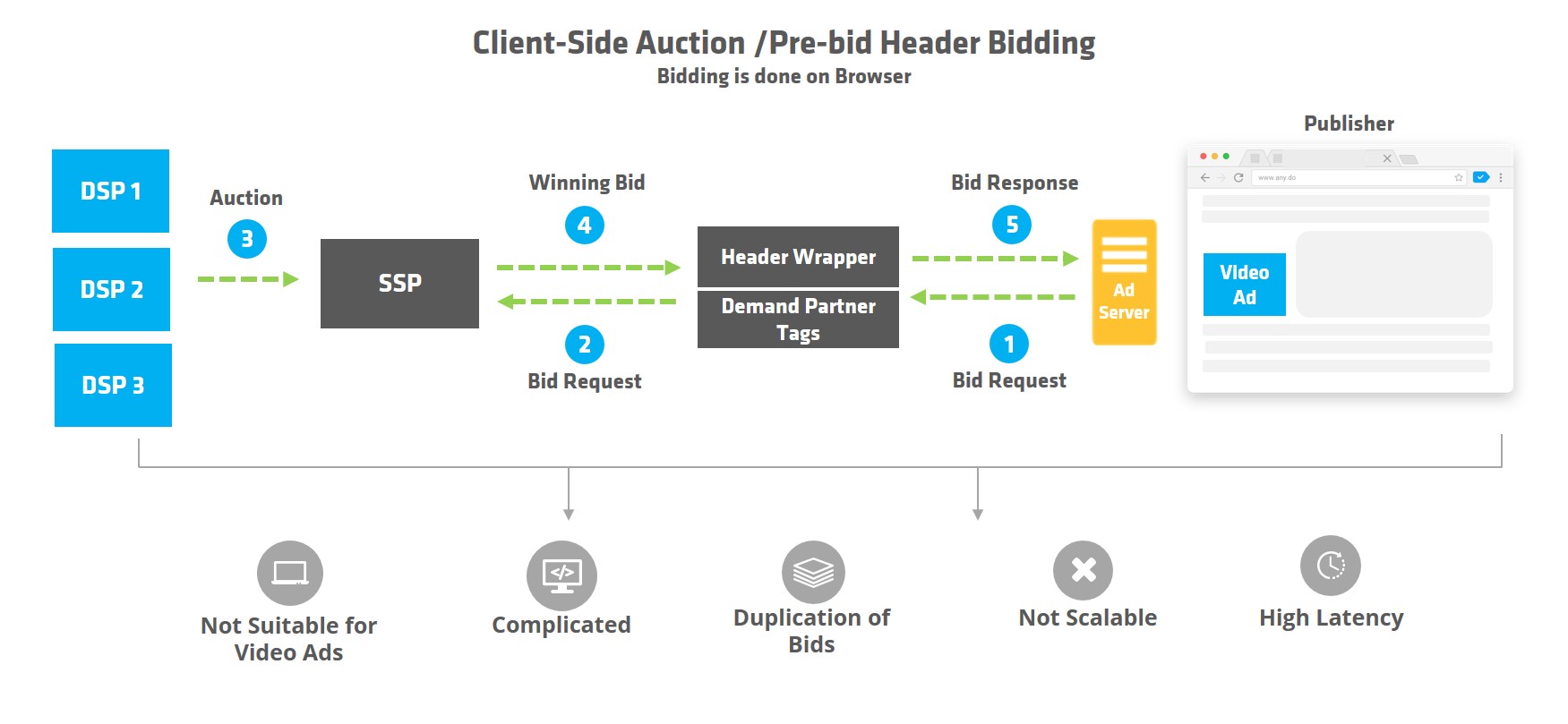


Şekil 22: Sistem Belleği

### **7.4 İstemci Tarafı Donanım Gereksinimleri**

İstemci tarafındaki en temel işlem bir internet bağlantısının olmasıdır eğer olmazsa sunucuya bağlanamaz. İnternet olmadan hiçbir müşteri web tarayıcısını ziyaret edemez. İnternet bağlantısına ek olarak fare ve klavyeye sahip olmalıdır.

İstemci tarafında 512 Mb’lık bir belleğe sahip olması gereklidir bunların altıda mevcutsa istemci gerekli verimi alamaz. Web sitesini görüntüleyebilmesi içinde minimum 800x600 piksel çözünürlüklü bir ekrana sahip olmalıdır.

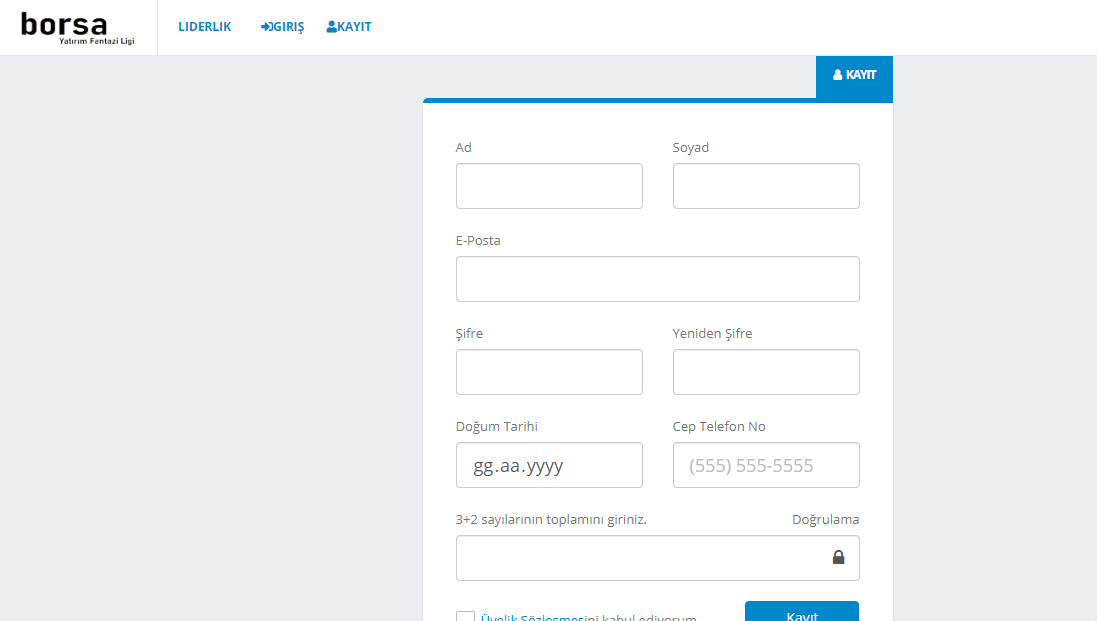


Şekil 23: İstemci

# **BÖLÜM 3**

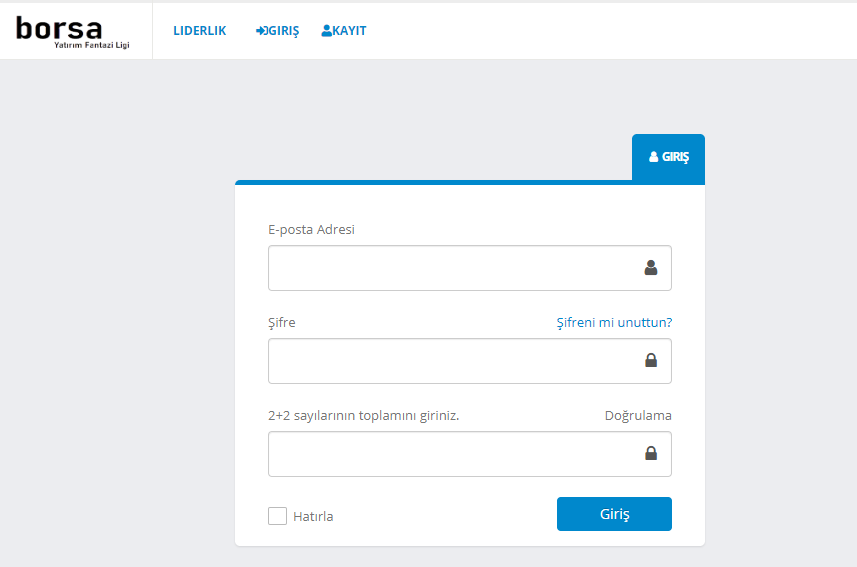
# **KULLANICI ARAYÜZÜ TASARIMI VE UYGULAMASI**

## **Giriş Sayfası**



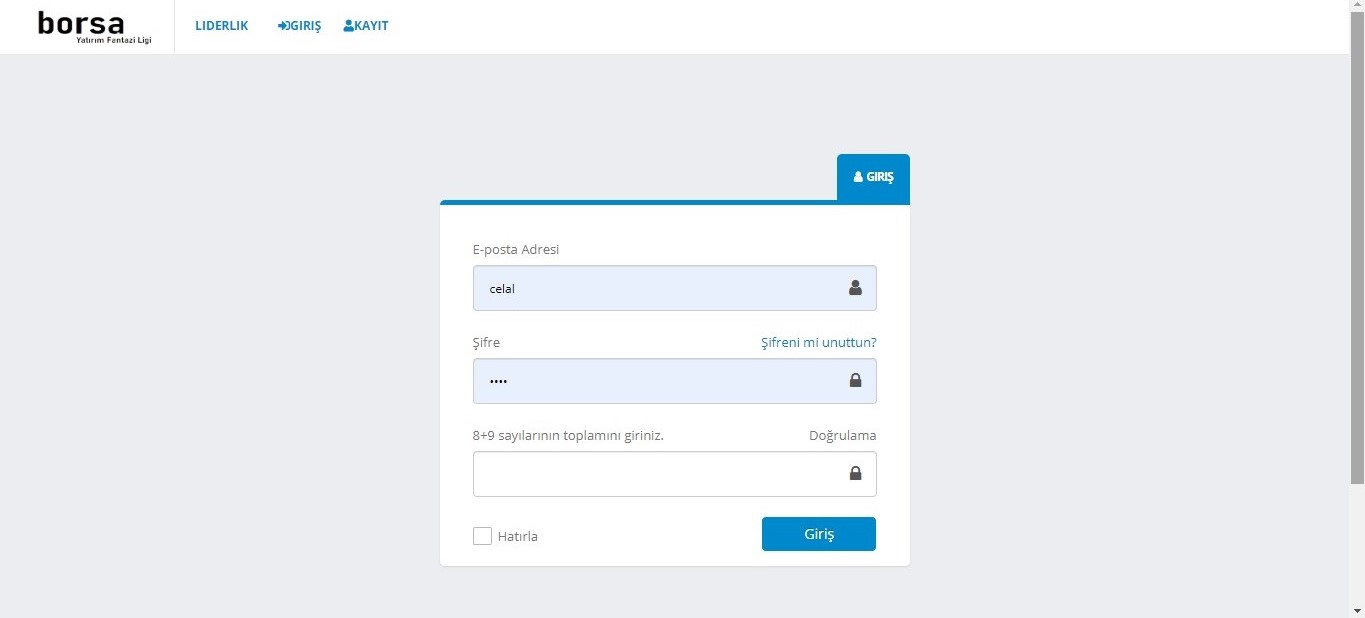
Şekil 24: Kayıt Sayfası

Bu görüntü, giriş sayfası için ekran gereksinimlerinin genel bir temsilini vermektedir. Burada bunun kullanıcı kayıt ekranında kullanıcının **adı, soyadı, eposta adresi, doğum tarihi,** telefon numarası, şifre istenecek olup bunlar dışında istenebilecek bilgiler kullanıcı bilgileri ekranı aracılığı ile istenebilecektir.



Şekil 25: Giriş Sayfası

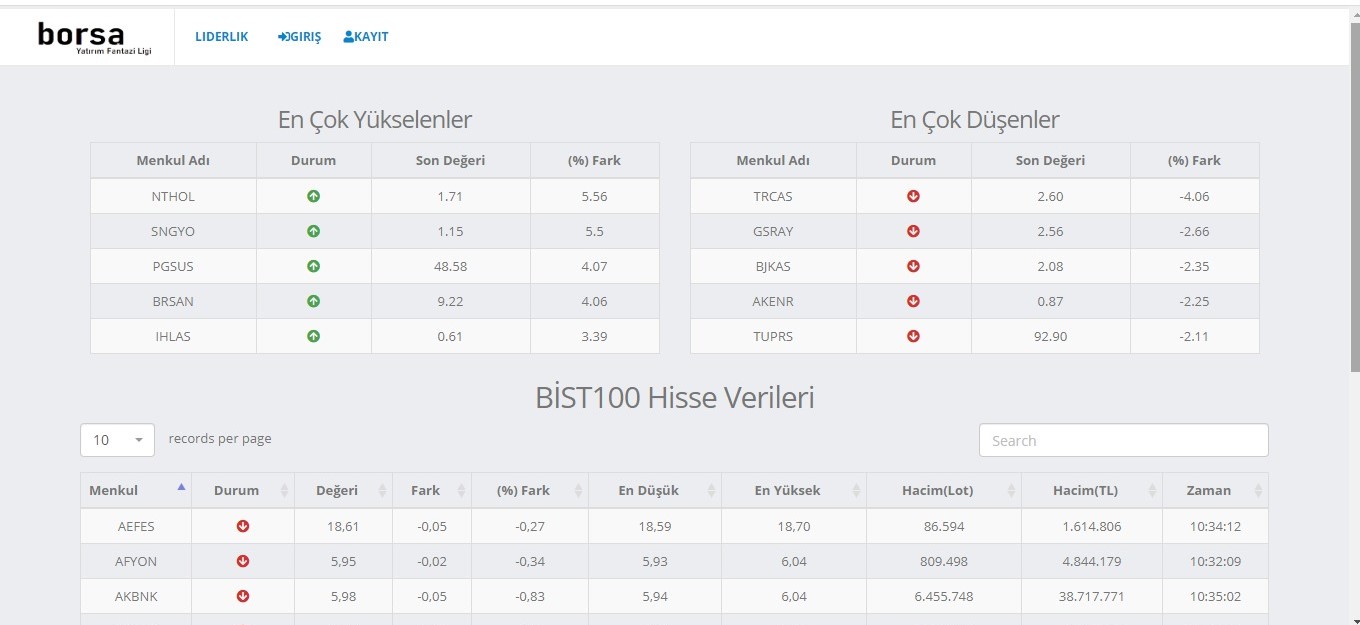
Kayıt yapıldıktan sonra sisteme giriş için eposta adresi ve şifre yeterli olacak ve hatırla butonu ile kullanıcının hatırlanması sağlanmaktadır.



Şekil 26: Giriş

Giriş işlemi yapıldıktan sonra karşımıza aşağıdaki görseldeki ana sayfamız gelecektir.

## **Ana sayfa**



Şekil 27: Anasayfa1

Bu ekranda da kayıt işlemi gerçekleştikten sonra bu ekran karşımıza çıkacaktır ve burada al sat işlemleri gerçekleştiği gibi hisse değerleri ve en çok yükselenler – en çok düşenler de görülecektir.



Şekil 28: Anasayfa2

Borsa Yatırım Fantezi Ligi kullanıcılar ve ziyaretçilerin kolayca BİST100 verilerini takip etmesi için tasarlanmıştır.

### **2.1 Genel Üstbilgi**

Kullanıcının yetki durumuna göre üst bilgiler değişecektir.

Ziyaretçi için ;



Kullanıcı için ;



Yönetici için ;



Şeklinde tasarlanmıştır.

## **Alım - Satım İşlem Ekranı**



Şekil 29: Alım-Satım İşlem Ekranı

Hisse senedi alım satımları ana sayfa üzerinden gerçekleşecektir.

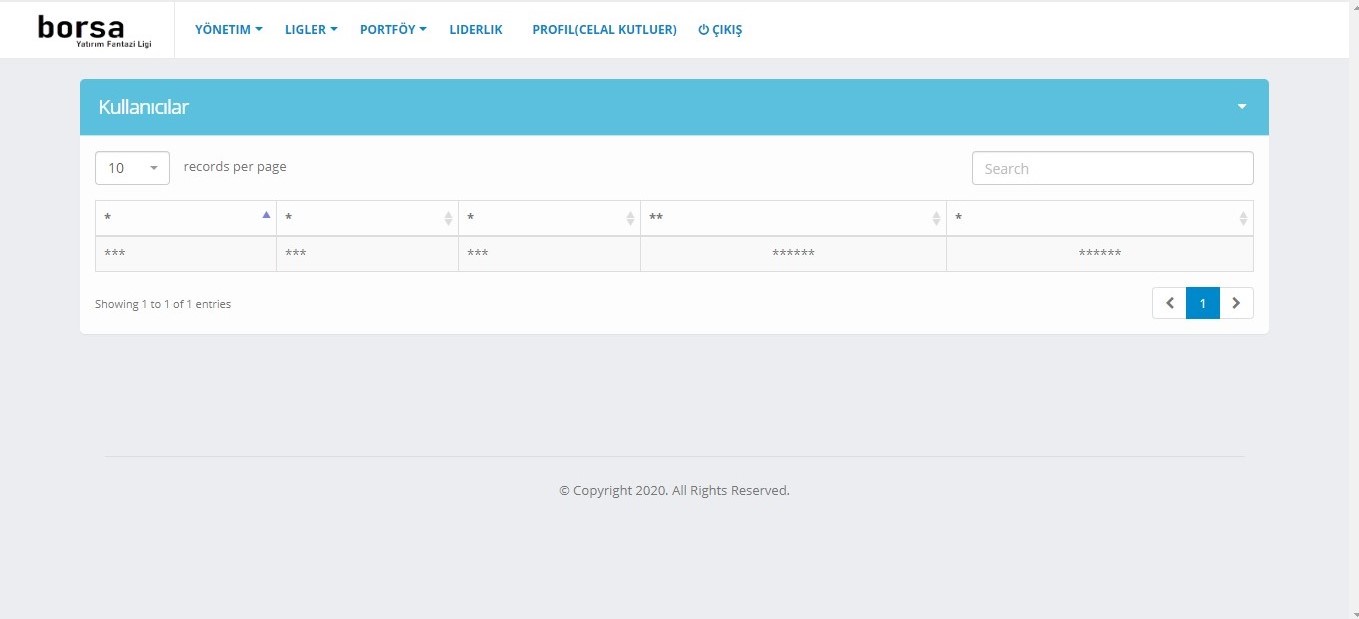
## **Lig Ekranı**



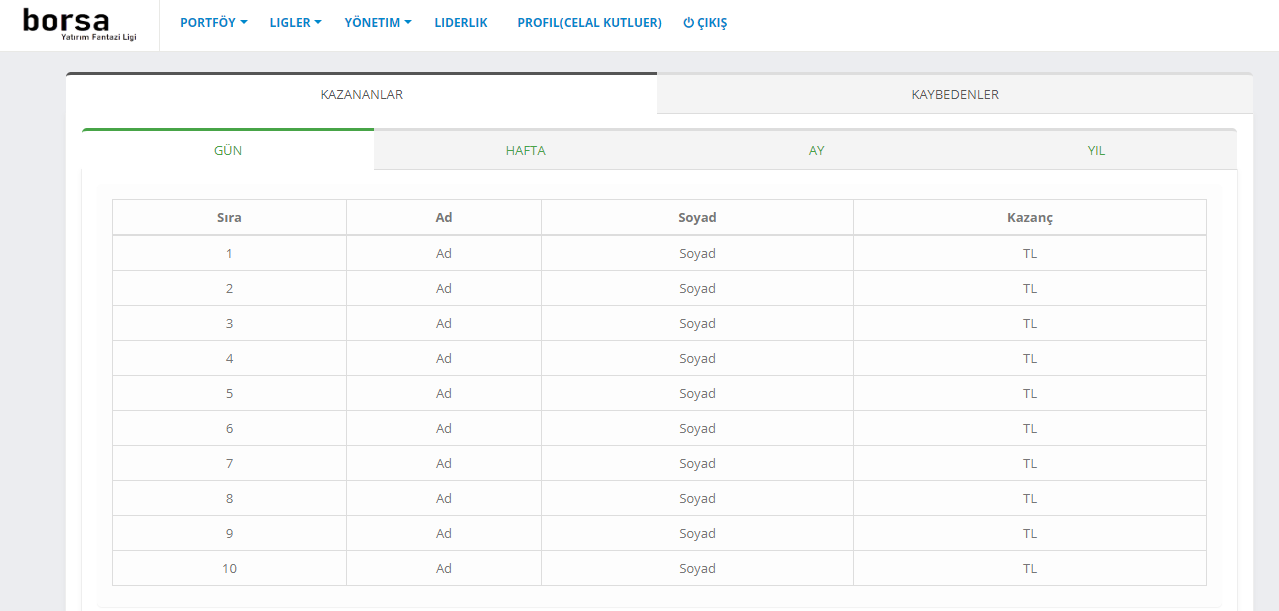
Şekil 30: Lig Ekranı

Lig ekranında veriler gözüktüğü gibi ek olarak yeni lig kaydı ve lig giriş işlemleri de gerçekleşmektedir.

## **Liderlik Gösterim Ekranı**



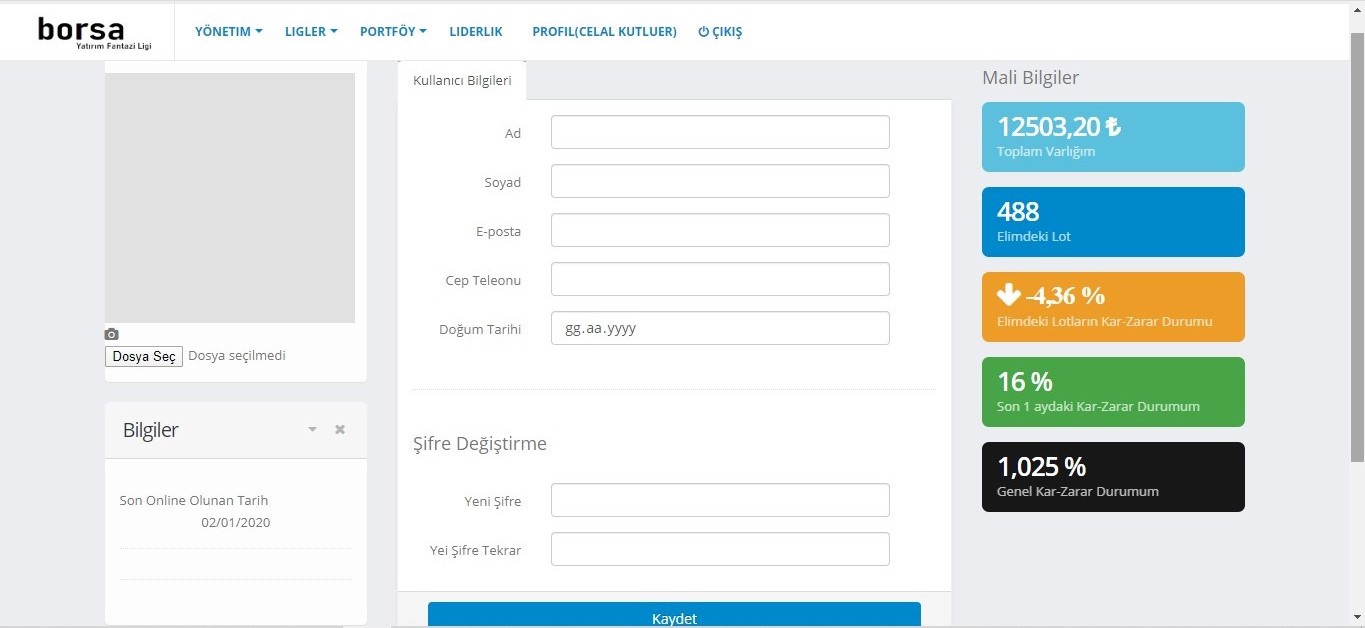
Şekil 31: Liderlik Gösterim Ekranı-1



Şekil 32: Liderlik Gösterim Ekranı-2

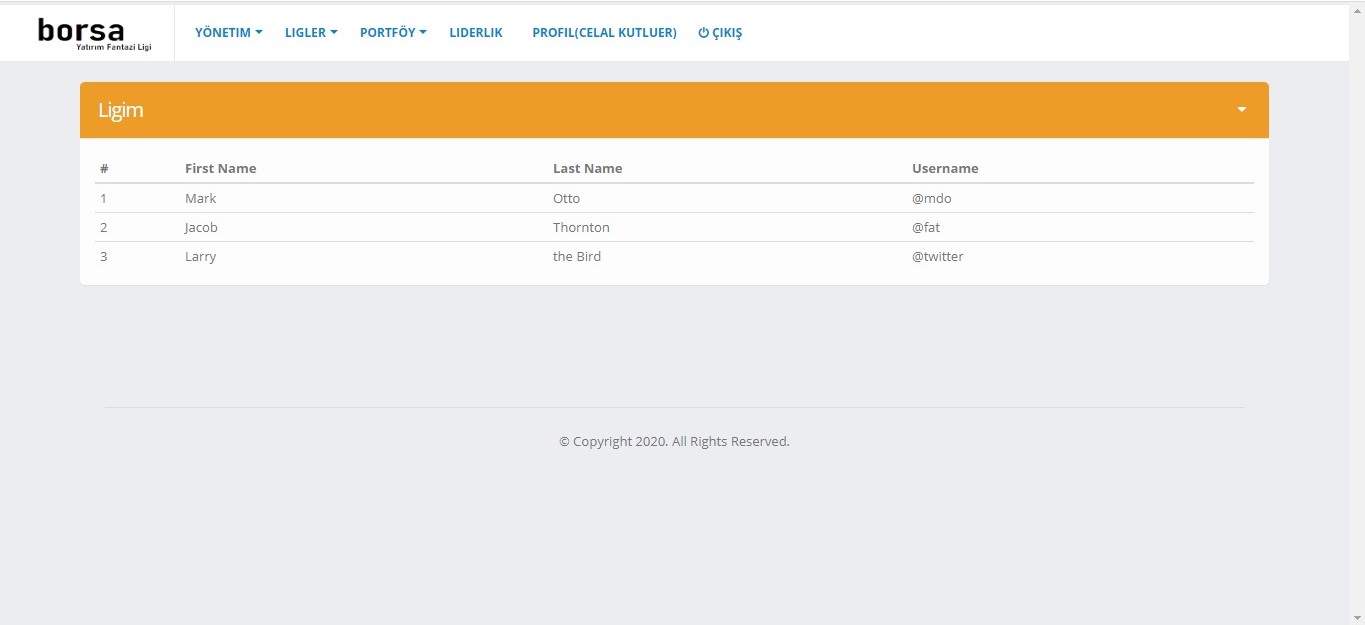
Liderlik ekranında; en çok kazananlar ve en çok kaybedenler şeklinde iki sıralama mevcut olmuştur. 10’ ar kişilik bu listelerde günlük, haftalık, aylık ve yıllık olarak kar ve zarar oranlarına göre belirlenen sıralamalar sunulmuştur.

## **Kullanıcı Profil Ekranı**



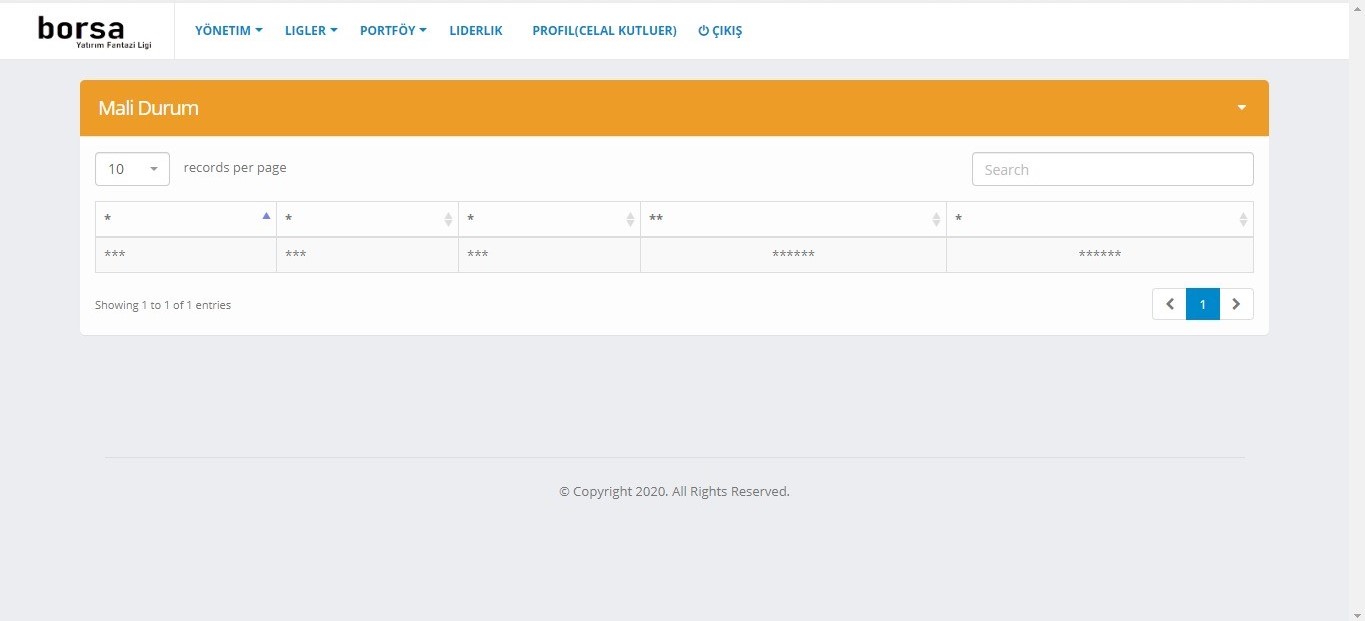
Şekil 33: Kullanıcı Profil Ekranı

Profil Bilgileri ekranında kullanıcının üyelik oluşturur iken kullandığı veriler ile sonradan talep edilen verilerin giriş-düzeltme ve silme işlemleri yapılabilecektir. Fakat üyelik kaydında zorunlu olarak alınan ***adı, soyadı, e-posta adresi, doğum tarihi, şifre*** bilgileri silinemeyecek sadece güncellenebilecektir.



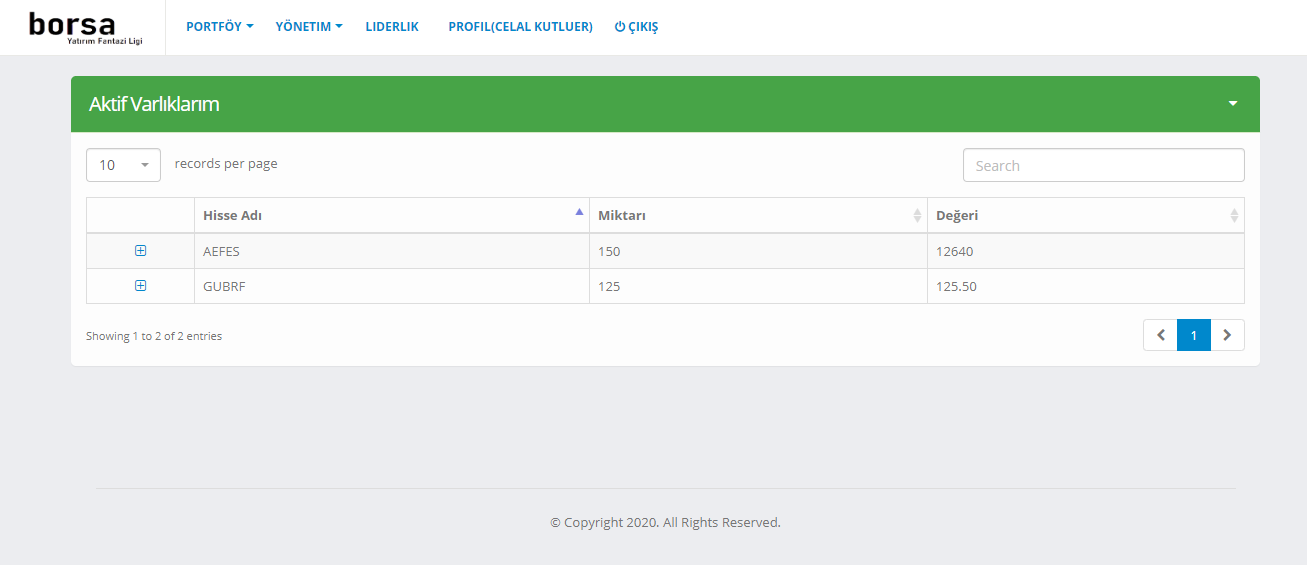
Şekil 34: Lig Ekranı

Kullanıcının kendine ait oluşturduğu lig ekranı isim, soy isim ve kullanıcı adı ile birlikte bu ekranda gösterilmiştir.



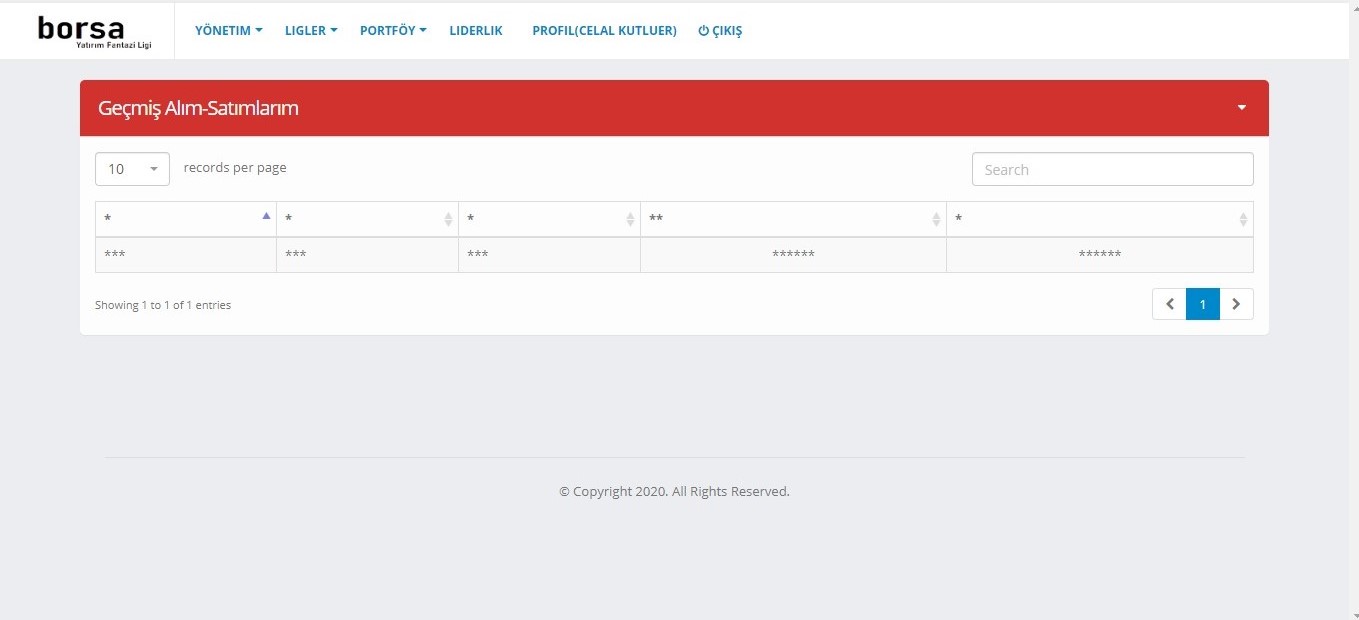
Şekil 35: Mali Durum Ekranı

Kullanıcının mali durum bilgilerinin gösterildiği ekrandır.



Şekil 36: Aktif Varlık Ekranı

Kullanıcının aktif varlıklarının bilgilerinin gösterildiği ekrandır. Aldığı hisse adı, miktarı ve değeri gösterilmiştir.



Şekil 37: Geçmiş Alım-Satım Ekranı

Kullanıcının geçmiş alım satım bilgilerinin gösterildiği ekrandır. Gerçekleştirdiği alım satım işlemlerinin kaydedilip sunulduğu tablodur.

# **BÖLÜM 4**

# **TEST TASARIMI**

**Yazılım Test Tasarımı** aşaması, test koşulları ve test durumlarını türetmek ve seçmek için test tasarım tekniklerinin uygulanmasını içeren test hazırlama aşamasını ele almaktadır. Yürüteceğimiz ya da koşacağımız test durumları için gerekli olan test koşulları test tasarım tekniklerini kullanarak yapabiliriz. Test koşulları belirlendikten sonra test planlanması aşamasında yaptığımız ürün risklerine bakarak önceliklendiririz.

**Yazılım Test Tasarımı** aşamasında test durumları oluşturulması önemli bir süreçtir. Bu test durumları bize yol haritası çizecektir. Nelerin test edileceğini planladığımız bu aşamada test teknikleri kullanılarak test senaryoları belirlenir.[8] Hiçbir uygulama tamamlanmamıştır, ancak bir projeyi uygulanabilir bir projeye yönlendirmenin büyük bir kısmı testtir. Test, beklenen işlevselliği sağlamamıza, olası güvenlik açıklarını kontrol etmemize,ve proje ilerledikçe gerilemeyi önleyecektir. Geliştiriciler, entegrasyon ve birim testlerini verimli ve verimli bir şekilde gerçekleştirmek için test paketleri kullanırlar.

## **1.Test Kutuları**

Borsa Yatırım Fantezi Ligi uygulaması aktif olarak geliştirilmektedir, bu nedenle her test belirtilen vaka yalnızca bu geliştirme aşamasında mevcut işlevler için geçerlidir.

Bileşenleri birim olarak test etmek verimli değildir. Bunun yerine, birbirleriyle nasıl etkileştiklerini görmek için bu birimler üzerinde entegrasyon testleri yapacağız. Borsa Yatırım Fantezi Ligi web sitemizi geliştirmek için Java / Scala Play Framework kullanılacak. Play Framework'ü seçmenizin ana nedeni, minimum kaynak tüketimi sağlar (CPU, bellek, iş parçacığı) ve ayrıca büyük veritabanlarını destekler. Yani, ekip üyelerinin çoğu C ++ konusunda yeterlidir, bu nedenle Java'ya geçiş yapılabilir.

## **2. Ünite Testleri**

### **2.1 Veritabanı Yöneticisi**

Aşağıda listelenen testler MySQL veritabanımızla etkileşime girer, ancak bunlarla hiçbir ilişkisi yoktur.

**1. Test örneği tanımlayıcısı TC-1:**

Test Edilen İşlev: oyuncu bilgilerini alma (kullanıcı kimliğinde: int): sınıf Kullanıcı

Başarı / başarısız kriterleri başarılı bir test, istenen bilgiler hakkında bilgi alan bir testtir

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| arama fonksiyonu (başarı) | bilgileri arama ölçütleriyle eşleşir |
| arama fonksiyonu (arıza) | bilgileri arama ölçütleriyle eşleşmiyor. |

**2. Test örneği Tanım İdentifierlayıcısı TC-2:**

Test Edilen İşlev: oyuncu bilgilerini güncelle (kullanıcı kimliğinde: int, upd: sınıf kullanıcısında): bool Başarı / başarısız - başarılı bir test, bir oyuncunun bilgilerini güncelleyen bir testtir.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | oyuncunun profili yeni bilgilerle güncellenir. |
| Çağrı fonksiyonu (arıza) | Oyuncu profil bilgileri, güncelleme denendikten sonra değişmez |

**3. Test örneği tanımlayıcısı TC-3:**

İşlev test edildi: sipariş bilgisi alın (işlem kimliği : int): sınıf işlem başarısı / başarısız kriterleri-başarılı bir test, ilişkili bilgileri döndüren testtir

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | İşlem bilgileri iade edilir |
| Çağrı hata(hata) | işlem bilgileri kullanıcıya iade edilmez. |

**4. Test örneği tanımlayıcısı TC-4:**

Fonksiyon test: güncelleme Ligi (Lig bilgisi : sınıf Ligi): bool Başarı / başarısızlık kriterleri-bu yöntem, bir ligin güncellenmesi gerektiğinde kullanılır.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Lig bilgileri başarıyla güncellendi. True değeri dönderir. |
| Çağrı hata(arıza) | Lig bilgileri daha önce değişmedi. False değeri döner. |

**5. Test örneği tanımlayıcısı TC-5:**

Fonksiyon test: dönüş Lig güncellemeleri (Lig ıd:int): sınıf Lig Başarı / başarısız kriterleri - başarılı bir test istenen kullanıcıya lig güncellemelerini döndürecektir.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Lig güncellemeleri talep eden kullanıcıya sunulur. |
| Çağrı hata(arıza) | Kullanıcıya istediği veri sunulmaz. |

### **2.2 Sipariş Yöneticisi**

Sipariş Yöneticisi, siparişler ve işlemlerle ilgili tüm görevleri yapmaktan sorumludur.

Sipariş Yöneticisi sisteme yeni siparişler vermek ve eski siparişi arşivlemekten sorumludur

**1. Test Durumu Tanımlayıcısı TC-6:**

Test Edilen İşlev: Sırayı kontrol edin (sembollerde: sınıf Sırası): bool Başarı / Başarısızlık Kriterleri Başarılı bir test, aşağıdaki değere karşılık gelen bir Boolean true değeri döndürür.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Kullanıcı geçerli bir işlem gerçekleştirebilir. Fonksiyon çağrısından sonra true değeri döndürülür. |
| Çağrı hata(arıza) | Kullanıcıya geçerli bir işlem yapamayacağı bildirilir. İşlem fonksiyon çağrısından sonra false döndürülür. |

**2. Test Durumu Tanımlayıcısı TC-7:**

Test Edilen İşlev: yer sırası (sembollerde: sınıf Sırası): bool Başarı / Başarısızlık Kriterleri - Başarılı bir test kullanıcının pazar emri vermesini sağlar.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Piyasa emri verilir ve kullanıcıya onay gönderilir. true işlevi işlev çağrısından sonra döndürülür. |
| Çağrı hata(hata) | Piyasa emri verilmez ve kullanıcı bilgilendirilir. Fonksiyon çağrısından sonra false değeri döndürülür. |

**3. Test Durumu Tanımlayıcısı TC-8:**

Test Edilen İşlev: Siparişi yürüt (işlem kimliği: int'de): bool Bir kullanıcı portföyünü buna göre finanse edin ve güncelleyin.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Sistem verileri alır ve kullanıcı portföyünü günceller. Değeri fonksiyon çağrısından sonra true döndürülür. |
| Çağrı hata(hata) | Sistem veritabanından bilgi almıyor veya  kullanıcı portföyü güncellenmez. False değeri döndürülür |

## **3. Lig Teknik Direktörü**

Bu sınıf, sistemdeki tüm ligleri yönetmekle sorumludur. Yetkisi vardır ligler oluşturun, ligleri silin ve eğer talimat verilirse ligleri değiştirin.

**1. Test Durumu Tanımlayıcısı TC-9:**

Test Edilen İşlev: Lig oluştur (): Sınıf lig

Başarı / Başarısızlık Kriterleri - Başarılı bir test, kullanıcının sıfırdan bir lig oluşturabildiği zamandır.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Kullanıcı artık lig yöneticisi |
| Çağrı hata(hata) | Sistemde hiçbir yeni lig kayıtlı değil ve kullanıcı bilgilendirilecek lig oluşturma girişimlerinin başarısız olduğu. |

**2. Test Durumu Tanımlayıcısı TC-10:**

Test Edilenler: reaturn lig güncellemeleri (lig id: int cinsinden): sınıf ligi

Başarı / Başarısızlık Kriterleri - Başarılı bir test seçilen ligi silecektir.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Seçilen lig, ligin kullanıcı listesinden silinir. Fonksiyon çağrısından sonra true döndürülür. |
| Çağrı hata(hata) | Lig, ligin kullanıcı listesinde kalacaktır. False değeri işlev çağrısından sonra döndü. |

Çağrı İşlevi (Başarı) Seçilen lig, ligin kullanıcı listesinden silinir. Değeri fonksiyon çağrısından sonra true döndürülür.

Çağrı İşlevi (Hata) Ligi, ligin kullanıcı listesinde kalacaktır. False değeri işlev çağrısından sonra döndü.

**3. Test Durumu Tanımlayıcısı TC-11:**

Test Edilen İşlev: lig adını değiştir (lig id: int'te): bool

Başarı / Başarısızlık Kriterleri - Başarılı bir testte, geçerli lig adı değiştirilmiş olarak güncellenir.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Lig adı değiştirildi ve veritabanına yansıtıldı. İşlev çağrısından sonra true değeri döndürülür. |
| Çağrı hata(hata) | Lig adı değişmedi. Fonksiyon çağrısından sonra False değeri döndürülür. |

**4. Test Durumu Tanımlayıcısı TC-12:**

Test Edilen İşlev: Lig yöneticisini değiştir (lig id: int, usr: sınıf Kullanıcı): bool

Başarı / Başarısızlık Kriterleri - Başarılı bir test, mevcut lig menajerini yenisiyle değiştirir

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Lig 'in yeni bir menajeri var ve tüm değişiklikler veritabanına yansıtılır. İşlev çağrısından sonra true değeri döndürülür. |
| Çağrı hata(hata) | Lig yöneticisi değişmeden kalır. Fonksiyon çağrısından sonra false değeri döndürülür. |

**5. Test Durumu Tanımlayıcısı TC-13:**

Test Edilen İşlev: kural ekleme (lig id: int'te): bool

Başarı / Başarısızlık Kriterleri -Başarılı bir test, zaten lig kuralları listesine yeni bir kural ekleyecektir.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Yeni eklenen kural veritabanına yansıtılır. İşlev çağrısından sonra true değeri döndürülür. |
| Çağrı hata(hata) | Eklenecek yeni kural eklenmedi ve veritabanı listesinde değişiklik yok Fonksiyon çağrısından sonra false değeri döndürülür. |

**6. Test Durumu Tanımlayıcısı TC-14:**

Test Edilen İşlev: Kuralları sil (lig id: int'te): bool

Başarı / Başarısızlık Kriterleri - Başarılı bir test, ligler kurallar listesindeki bir kuralı siler.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Seçilen kural silinir ve veritabanı güncellenir  değiştirin. İşlev çağrısından sonra true değeri döndürülür. |
| Çağrı hata(hata) | Seçilen kural kaldırılmadı ve veritabanı  değiştirir. Fonksiyon çağrısından sonra false değeri döndürülür. |

## **4. Hesap Denetleyici**

Bu sınıf, herhangi bir kullanıcı hesabını içeren işlevlerle ilgilenmek için vardır. Fonksiyonlar arasında, hesap ekleme, değiştirme veya silme işlemleri bulunur.

**1. Test Durumu Tanımlayıcısı TC-15:**

Test Edilen İşlev: Giriş (kullanıcı kimliğinde: int): bool

Başarı / Başarısızlık Kriterleri - Başarılı bir test, kullanıcının Paramount Investments League global portföyünü ziyaret etmesini sağlar.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Arama İşlevi (Başarılı) | Kullanıcı sistemde oturum açar ve hesaplarını görüntüleyebilir. İşlev çağrısından sonra true değeri döndürülür. |
| Arama İşlevi (Hata) | Kullanıcı web sitesinde oturum açmadı. Kullanıcı girmemiş olabilir şifrenizi doğru girmiş veya kayıtlı bir kullanıcı değil. Fonksiyon çağrısından sonra false değeri döndürülür. |

**2. Test Durumu Tanımlayıcısı TC-16:**

Test Edilen İşlev: oturumu kapatma (kullanıcı kimliğinde: int): bool

Başarı / Başarısızlık Kriterleri - Başarılı bir test, tüm kullanıcıların Paramount Investments League hesabından çıkış yapmalarını sağlar.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Arama İşlevi (Başarılı) | Kullanıcı sistemde oturum açar ve hesaplarını görüntüleyebilir. İşlev çağrısından sonra true değeri döndürülür. |
| Arama İşlevi (Hata) | Kullanıcı web sitesinde oturum açmadı. Kullanıcı girmemiş olabilir şifrenizi doğru girmiş veya kayıtlı bir kullanıcı değil. Fonksiyon çağrısından sonra false değeri döndürülür. |

**3. Test Durumu Tanımlayıcısı TC-17:**

Test Edilen İşlev: Hesap oluştur (): sınıf Kullanıcı

Başarı / Başarısızlık Kriterleri - Başarılı bir test yeni bir kullanıcı hesabı oluşturur.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Arama İşlevi (Başarılı) | Paramount Investments League web sitesini ziyaret eden eski bir ziyaretçi şimdi kayıtlı bir yatırımcıdır. İşlev çağrısından sonra true değeri döndürülür. |
| Arama İşlevi (Hata) | Yeni hesap oluşturma isteği başarısız oldu ve yeni hesap yok veritabanına yansıtılacaktır. Fonksiyon çağrısından sonra false değeri döndürülür. |

**4. Test Durumu Tanımlayıcısı TC-18:**

Test Edilen İşlev: hesabı sil (kullanıcı kimliğinde: int içinde): bool

Başarılı / Başarısız Kriterler - Başarılı bir test seçilen kullanıcı hesabını siler.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Bir yatırımcı hesabını silmeyi seçer ve tüm portföyler veritabanından silinecek. İşlev çağrısından sonra true değeri döndürülür. |
| Çağrı hata(hata) | Seçilen hesap sistemde kalır, veritabanı kaybolmaz. Fonksiyon çağrısından sonra false değeri döndürülür. |

## **5.Finans Bağdaştırıcısı**

Bu sınıf, Finance API'sinden piyasa verilerini elde etmekten sorumludur. Üç oluşur

fiyat teklifi alma, şirket bilgisi alma ve sektör bilgisi alma işlevlerini yerine getirir.

**1. Test Durumu Tanımlayıcısı TC-19:**

Test Edilen İşlev: Fiyat teklifi alın (stokta yer kodu id: string): sınıf teklifi

Başarı / Başarısızlık Kriterleri - Başarılı bir test, istenen teklif (stok) bilgilerini kullanıcıya

Döndürür.

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Teklif bilgileri kullanıcıya sunulur. Sistem erişim istekleri |
| Çağrı hata(hata) | Teklif bilgisi isteği gerçekleşmez ve kullanıcıya hata bildirilir. Sistem iletişim kuramadı |

**2. Test Durumu Tanımlayıcısı TC-20:**

Test Edilen İşlev: şirket bilgisini alın (stoktaki kimlik numarası: dize): sınıf Şirket

Başarı / Başarısızlık Kriterleri - Başarılı bir test, istenen şirket bilgilerini kullanıcıya geri gönderir

|  |  |
| --- | --- |
| Test Prosedürü: | Beklenen Sonuçlar |
| Çağrı fonksiyonu (başarı) | Bir yatırımcı hesabını silmeyi seçer ve tüm portföyler veritabanından silinecek. İşlev çağrısından sonra true değeri döndürülür. |
| Çağrı hata(hata) | Seçilen hesap sistemde kalır, veritabanı kaybolmaz. İşlev çağrısından sonra false değeri döndürülür. |

## **6.Test Kapsamı**

İdeal test kapsamı, her yöntemin her uç durumunu kapsayan bir teste sahip olmak olacaktır. Bu sadece mümkün değil, aynı zamanda imkansız çünkü tüm kenarı bilmek mümkün değildir. Bu nedenle, temel miktarda test sağlamak için çekirdek işlevselliğini test etmeyi planlıyoruz. Daha sonra Son kullanıcılarla alfa ve beta derleme etkileşimlerini kullanarak, öngörülmeyen sistemle etkileşime girme yollarını, daha sonra da ek testler ekleyebiliriz. Bu gelecekte hata ayıklamaya yardımcı olacaktır.

## **7. Entegrasyon Testi**

Entegrasyon testinin amacı birbirine bağlı modülleri bir bütün olarak test edip genel yapıda bir hata var mı kontrol etmektir.[9]

Entegrasyon testi, sunucu ortamını taklit ederek yerel bir geliştirici makinesinde yapılacaktır. Mevcut sistem entegrasyon ortamında çalışana kadar sistem canlı çalışmayabilir.

Bunu, kaynak kodun iki dalı olan master ve dev. dev, tüm yeni işleri yapılacaktır. Oradan yerel entegrasyon makinesine çekilecek, test edildi ve hata ayıklandı. Sistem hata ayıklandıktan sonra, herhangi bir tanesinin 1 ayrıntılı günlüğünü tuttuğunuzdan emin olunması gerekir.

Sistem yapılandırma değişiklikleri gerekli, kaynak kodu master itilir. Bir kez ustalaşmaya itildiyse, yeni sisteme uyum sağlamak için üretim sunucusunda sistem yapılandırma değişiklikleri yapılacaktır. Bu değişiklikler yapıldıktan sonra, master üretim makinesine çekilecek. Entegrasyon testinin ikinci aşaması, hizmeti bir geliştirici bağlantı noktasında başlatarak başlayacaktır. Eğer tüm testleri geçtikten sonra geliştirici bağlantı noktası kapanacak ve sistem web sitesi normal http bağlantı noktasında bulunabilir.

# **BÖLÜM 5**

# **PROJE YÖNETİMİ**

## **1.Ekip Üyelerinin Bireysel Katkıları**

### **1.1 Beyzanur TAŞKÖPRÜ**

Proje yönetimini benim grup liderliğimle ve grup arkadaşlarım Celal KUTLUER, Tarık ERDEN, Merve TOKAT ve Haydar BULUT ile birlikte yapmaya çalıştık. Her arkadaşım malum nedenlerden dolayı yüz yüze toplantı yapamamamız nedeniyle çalışmalarımızı Whatsapp , Classroom ve GitHub uygulamaları üzerinden görev dağılımlarımızı yapıp konuları grup üyelerine paylaştırdık bu sayede herkesin aktif şekilde katkı vermesine vesile olmak istedik. Projenin yönetimi, veri depolama, ağ protokolü ve kullanıcı arayüzü tasarımı ile ilgili konuları tarafımdan sağlandı.

Bir analizi verimli yapabilmek için en önemli olan etken, mümkün olduğunca çok verinin toplanabilmesidir. Projemizde 100 adet veri kullanarak işlemler gösterilmiştir. Böylelikle en çok yükselen ve en çok düşen değerleri daha net görebilmemiz hedeflenmiştir.

Gerçekleştirdiğimiz yazılımda yani Borsa Yatırım Fantezi Lig’i projesinde de HTTP kullanılmıştır. Http kullanmamızın sebebi istemci konumunda olan bilgisayarlar ile bilginin sağlandığı sunucular arasında bir köprü oluşturmaktır.

Kullanıcı arayüz tasarımı; kullanıcı arayüz çeşitleri arasında (Grafiksel, işitsel, dokunsal vb.) iletişim ve etkileşim köprüsü görevi gören kullanıcı arayüzlerinin tasarlandığı tasarım disiplinidir. Projenin giriş sayfası, kayıt sayfası, anasayfa, portföy bilgileri, alım-satım işlem hareketleri ve durum hareketlerinin gösterildiği ekranları raporda detaylı bir şekilde gösterilmiştir.

### **1.2. Celal KUTLUER**

Borsa Yatırım Fantezi Ligi projemizde rapor 1 de bahsetmiş olduğumuz kullanım durumları rapor 2 de diyagramlara dönüştürüldü. Yatırımcı, yönetici ve ziyaretçilerin web sitesine girdiğinde yapabilecekleri buton, nav-menü vb. etkileşimlerine ait uml diyagramları çıkarıldı. Diyagramlar ücretsiz olarak uml diyagramları çizimine izin veren <https://www.websequencediagrams.com/> üzerinden çizildi. Etkileşimlerde temelde **Yatırımcı(Görünen Yüz)---Kontrolcü(Ara Katman)---veritabanı veya Finans APİ(Son katman)** olarak düşünüldü ve buna göre diyagramlar oluşturuldu. Sadelik ön planda tutulmaya çalışıldı. Veritabanı olarak php ile çalışacağımız öngörüsü ile Mysql düşünüldü.

### **1.3 Tarık ERDEN**

Class diyagramları ve tanımlamaları ben ve grup arkadaşlarım Celal KUTLUER, Beyza Taşköprü , Merve TOKAT ve Haydar BULUT ile birlikte yapmaya çalıştık. Her arkadaşım malum nedenlerden dolayı yüz yüze toplantı yapamamamız nedeniyle çalışmalarımızı Whatsapp , Classroom ve GitHub uygulamaları üzerinden görev dağılımlarımızı yapıp konuları grup üyelerine paylaştırdık bu sayede herkesin aktif şekilde katkı vermesine vesile olmak istedik. Class diyagramları, UML 'in en sık kullanılan diyagramlardan biri olup nesne yönelimli analiz, tasarım ve programlamadaki sınıfları net ve anlaşılabilir şekilde temsil etmeyi amaçlar . Tanımlamalar da Class diyagramı içinde bulunan değişkenlerin ne tür işlemler yaptığı ve nerelerde kullanılacağını belirttim.

### **1.4 Merve TOKAT**

Grup arkadaşlarımla hazırlamış olduğumuz projeyi hazırlarken whatsapp ve Google meets’i kullanarak projeyi tamamlamaya çalıştık.

Benim projedeki katkım sistem mimarisi ve sistem tasarımı kısmında mimari tarzları tanıtmak, alt sistemlerin tanımlanması, alt sistemleri donanıma eşleme ve test tasarımı başlıkları altına yer almaktadır.

Projemizi hazırlarken çeşitli internet sitelerinden araştırma yaptık ve projemizi oluştururken çeşitli tablolar ekledim.

### **1.5 Haydar BULUT**

Projemizin adı Borsa yatırım fantezi ligidir. Projemizin konusu yatırımcıların finansal hisse senedi alım satımını öğrenmelerini sağlayan bir fantezi oyunu geliştirir ve oyun alım, satım faaliyetlerini destekler. Bu projeye başlamadan önce classroom, Whatsap ve google meet gibi program araçlarıyla grup arkadaşlarımla iletişime geçerek görev paylaşımını yaptık bana düşen görev şudur: Global kontrol akışı ve Donanım gereksinimleridir. Projemin konu başlığını yaparken bir çok kaynaktan yardım aldım gerekli gördüğüm yerlerde grup arkadaşlarıma danışıp yardım aldım.

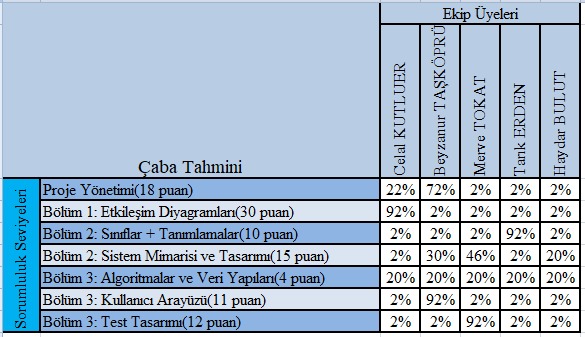
## **2. Proje Koordinasyon**

## **3.Çalışma Planı**

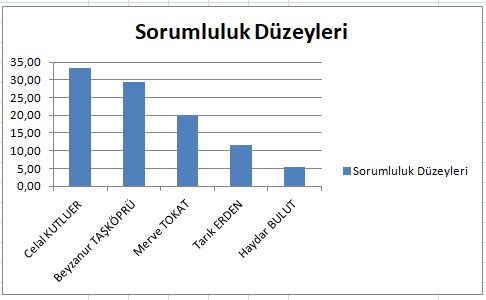
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AYLAR | MART | | | | | | | | | | |
| GÜNLER | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| CELAL KUTLUER | Eksiklerin belirlenip görev dağılımının yapılması- Toplantı | Etkileşim Diyagramları |  |  | Arayüz Tasarım ve Toplantı |  |  |  |  |  | Belge Birleştirme |
| BEYZANUR TAŞKÖPRÜ | Eksiklerin belirlenip görev dağılımının yapılması- Toplantı | Bölümsel Konu Toplantısı |  | Sistem Mimarisi ve Tasarımı | Arayüz Tasarım ve Toplantı |  |  |  | Kullanıcı Arayüzü Tasarım Ve Uygulaması |  | Belge Birleştirme |
| TARIK ERDEN | Eksiklerin belirlenip görev dağılımının yapılması- Toplantı |  |  |  | Arayüz Tasarım ve Toplantı |  |  |  | Sınıf Diyagramları |  | Belge Birleştirme |
| MERVE TOKAT | Eksiklerin belirlenip görev dağılımının yapılması- Toplantı | Bölümsel Konu Toplantısı | Sistem Mimarisi ve Tasarımı |  | Arayüz Tasarım ve Toplantı |  |  |  |  | Test Tasarımı | Belge Birleştirme |
| HAYDAR BULUT | Eksiklerin belirlenip görev dağılımının yapılması- Toplantı |  |  |  | Arayüz Tasarım ve Toplantı | Sistem Mimarisi ve Tasarımı |  |  |  |  | Belge Birleştirme |

## **4. Sorumlulukların Dağılımı**

### **4.1. Çaba Tahmini**



### **4.2 Sorumluluk Düzeyi**



# **BÖLÜM 6**

# **KAYNAKÇA**

[1] <https://www.smartdraw.com/network-design/examples/?id=358478>

[2] <https://forum.turkmmo.com/konu/3690460-sistem-gereksinimleri-nedir-minimum-ve-onerilen-sistem-gereksinimi-ne-anlama-gelir/>

[3] <https://brightlineit.com/business-two-internet-connections/>

[4] <https://snapcraft.io/install/disk-space-saver/debian>

[5] <http://www.dijitalteknoloji.net/bilgisayar/sistem-bellegi-ram-nedir-tum-detaylar.html>

[6] <https://www.learncomputerscienceonline.com/what-is-computer-memory/>

[7] <https://chocolateplatform.com/client-side-auction-vs-server-side-auction-which-is-more-suited-for-video-advertising/>

[8] https://www.mobilhanem.com/yazilim-test-tasarimi-yapilmasi/

[9]<http://www.burakavci.com.tr/2017/01/integration-testing.html>

* 1-Örnek Proje
* <https://www.websequencediagrams.com/>
* 2-Ders Notları
* <https://medium.com/gokhanyavas/interaction-diagram-etkile%C5%9Fim-diyagramlar-47f7241d96bd>
* https://classroom.google.com/u/0/c/NjIzMTcwMTQzMDBa( örnek pdf)
* <https://github.com/celalkutluer/Borsa_Projesi_Raporlari>