7 Sistem Etkileşim Diyagramları

7.1 Giriş

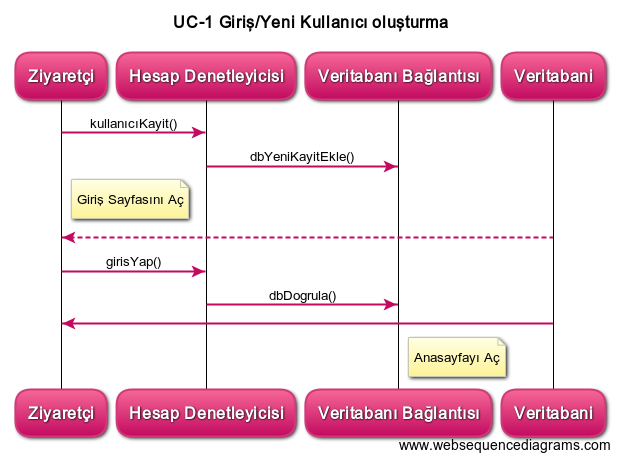
Aşağıdaki bölümdeki etkileşim diyagramları, yazılımımızın en önemli bölümlerindeki sistem etkileşimlerini açıklayacaktır. Her özel kullanım durumu için, sistemler ve veritabanları arasındaki etkileşimleri özetleyeceğiz. Ayrıca, sistemlerin farklı senaryoları ele alacağı birçok durumu analiz edeceğiz.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

7.2 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, veritabanı bağlantı modülü aracılığıyla veritabanında yinelenen oturum açma bilgilerinin bulunmadığından emin olmaya çalışır(burada e-posta adresini kontrol eder) ve eğer değilse yeni kullanıcı bilgilerini veritabanına kaydeder. Bu gerçekleştikten sonra kullanıcıya bir onay e-postası gönderilir. Ardından Hesap Denetleyicisi Giriş Sayfasına gider.

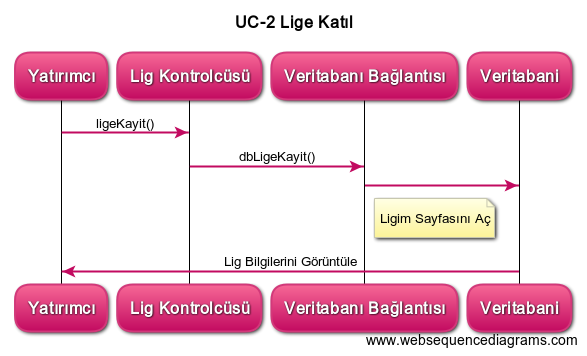
Bir kullanıcı oturum açmaya çalışırsa, Hesap Denetleyicisi DB Bağlantısı modülü aracılığıyla veritabanında bulunan ayrıntılarla oturum açma ayrıntılarını doğrulamaya çalışır. Ayrıntılar doğru bir şekilde eşleşirse, Hesap Denetleyicisi misafiri kullanıcı(yatırımcı) moduyla Ana sayfaya gönderir.



Şekil 7.1: UC-1

Use Case 2

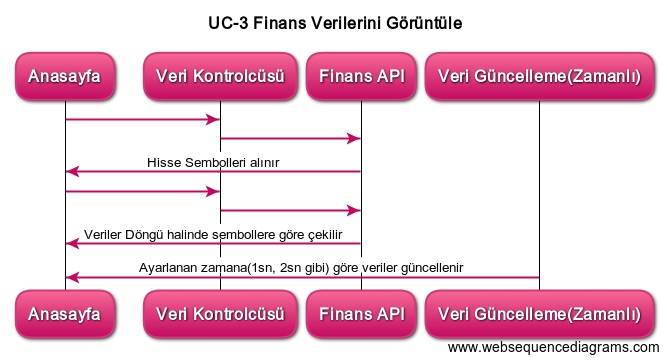
UC-2 dizi diyagramında gösterilen, bir yatırım liginin nasıl oluşturulacağının akışıdır. Bir yatırımcı Ligler sayfasından bir lig oluşturmayı seçtiğinde Lig Kontrolcüsü ile temasa geçilecektir. Lig Kontrolcüsü Ligler arasında bağlantı kurulur. Bundan sonra, ligi yaratacak ve DB Bağlantısı üzerinden veritabanında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



Şekil 7.2: UC-2

Use Case 3

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

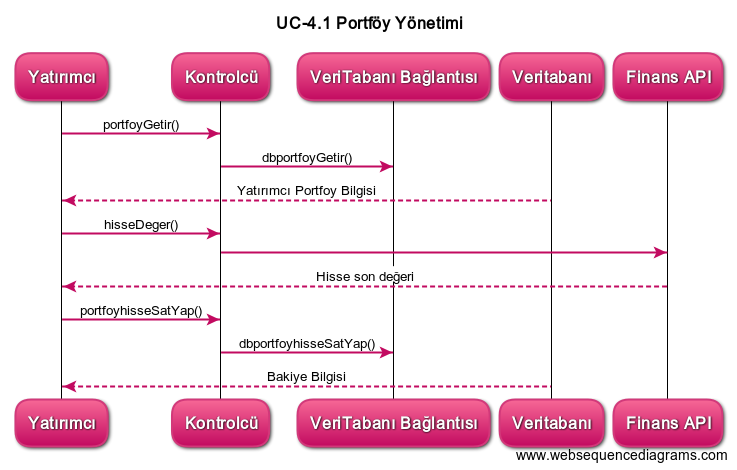


Şekil 7.3: UC-3

Use Case 4

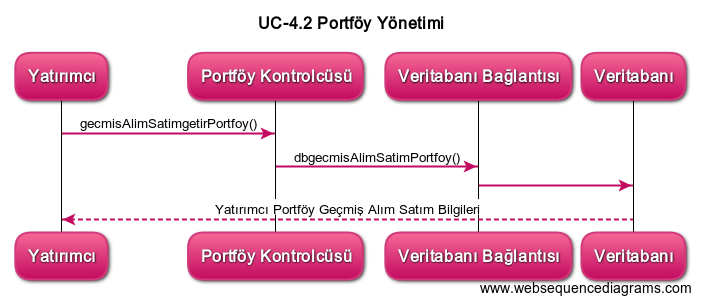
Yatırımcı, Portföyünü görüntüleyebilmeli ve üzerinde değişiklik yapabilmelidir. Portföy yönetimi 2 şekilde sağlanır. Bunlar;

* Kullanıcı portföyünü görüntülemek için Aktif Varlıklarıma tıkladığında, Portföy Denetleyicisi Veritabanı Bağlantısı modülü aracılığıyla yatırımcı portföyü stoklarını veritabanından alır. Satım yapılmak istenirse Portföy Denetleyicisi Veritabanı Bağlantısı modülü aracılığıyla veritabanını günceller.



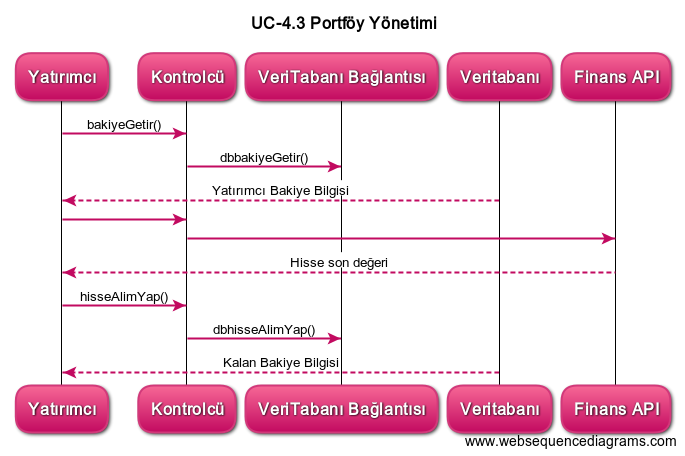
Şekil 7.4: UC-4.1

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



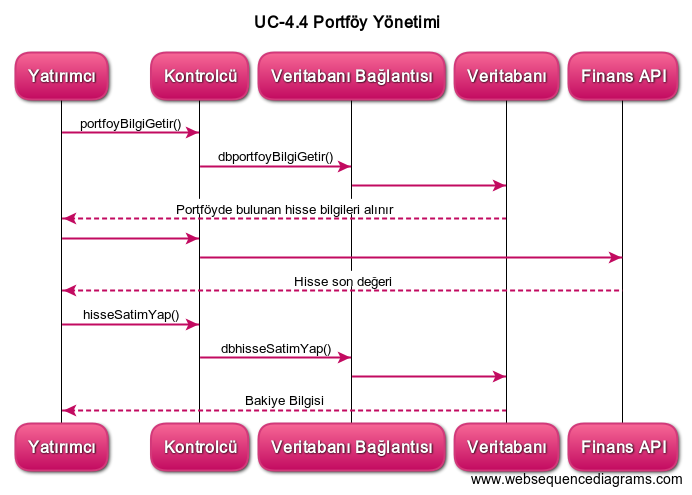
Şekil 7.5: UC-4.2

* Anasayfa üzerinden portföy alım satımları yapılabilir Kontrolcü Veritabanı Bağlantısı aracılığı ile Veritabanından bakiye bilgisini alır.Hisse alımı sırasında Kontrolcü Veritabanı Bağlantısı aracılığı ile Veritabanına alınan hisse bilgilerini kaydeder ve kalan bakiye bilgisini gönderir.



Şekil 7.6: UC-4.3

* Bu alanda satım işlemi de yapılabilir Kontrolcü Veritabanı Bağlantısı aracılığı ile Veritabanından portföyde bulunan hisse bilgilerini alır. Hisse satımı sırasında Kontrolcü Veritabanı Bağlantısı aracılığı ile Veritabanına alınan hisse bilgilerini düşer ve bakiye bilgisini gönderir.

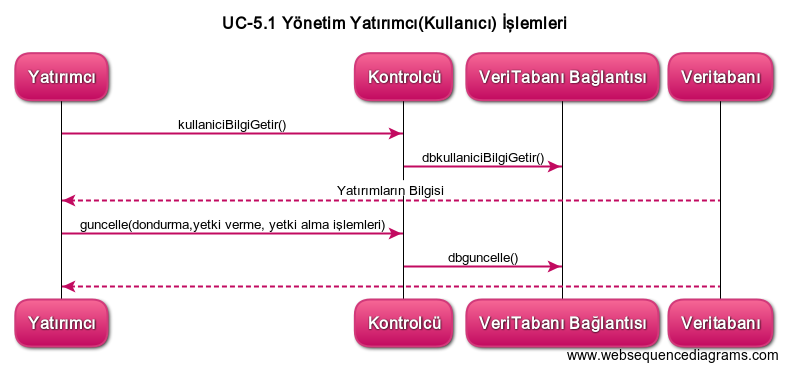


Şekil 7.7: UC-4.4

Use Case 5

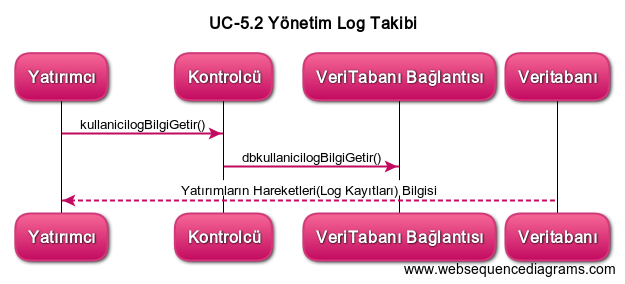
* Yönetim alanı kullanıcı(yatırımcı)ların yetkilendirilmesi, takibi ve mali kontoller için tanımlanacaktır.

Kullanıcı işlemleri;



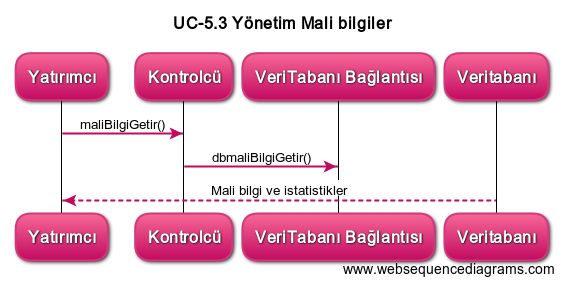
Şekil 7.8:UC-5.1

Log Kayıtlar;



Şekil 7.9:UC-5.2

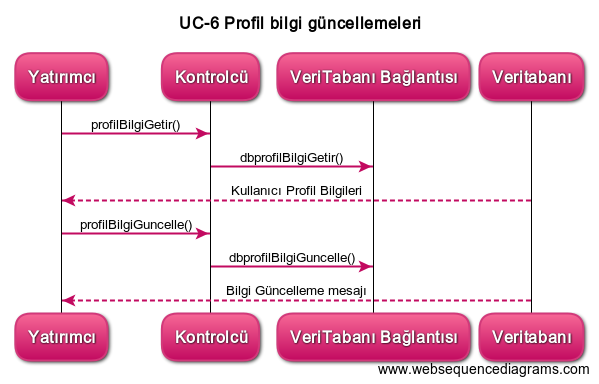
Mali Bilgiler;



Şekil 7.10:UC-5.3

Use Case 6

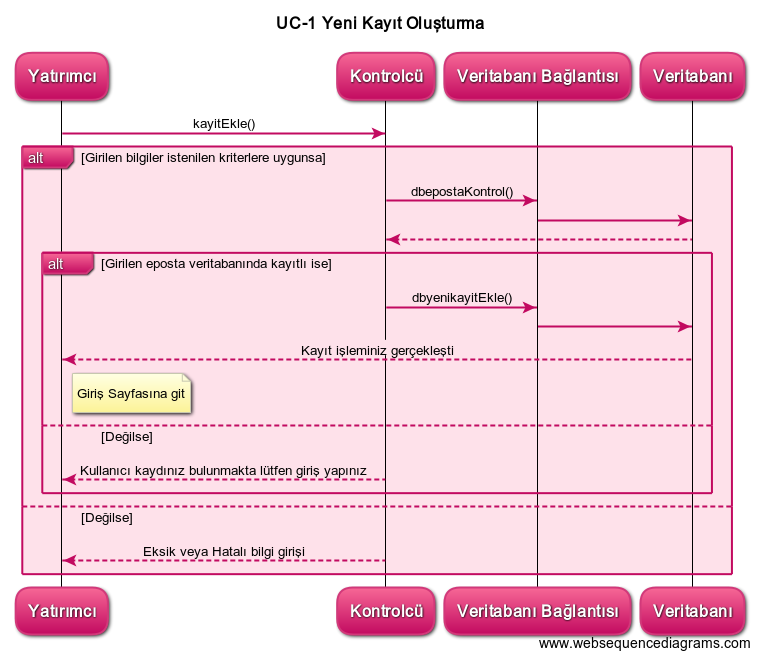
Profil Bilgileri alanı kullanıcının kayıt olurken verdiği bilgilerin değiştirimesi ve şifre değiştirme işlemleri için kullanılır. Kontrolcü veritabanı bağlatısı aracılığı ile veritabanından profil bilgilerini getirir. Değişiklik işlemi yine veri tabanı bağlantısı aracılığı ile veritabanına kaydedilir.

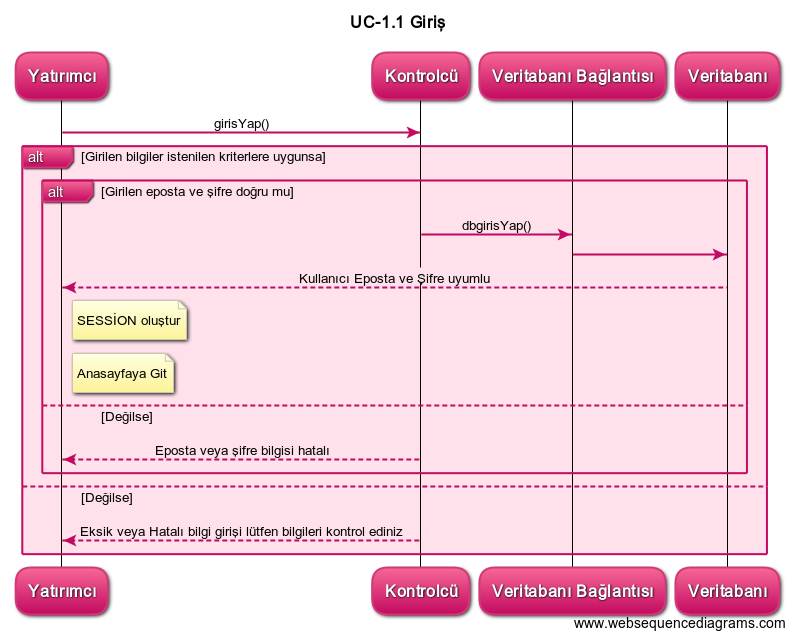


Şekil 7.11: UC-6

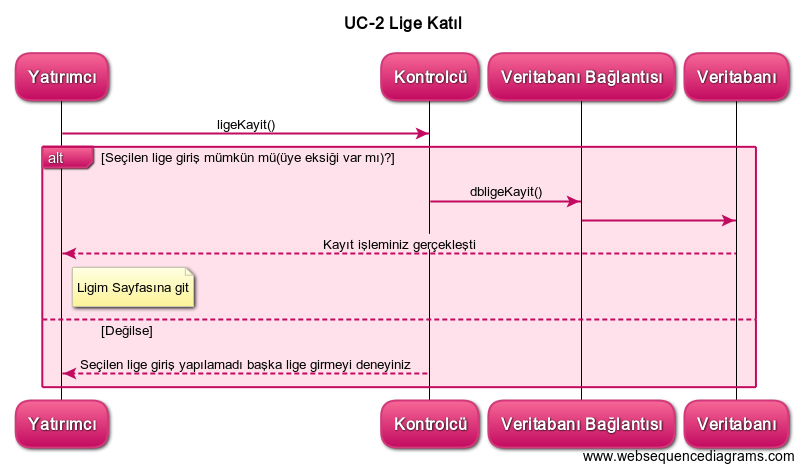
7.2 Alternatif Çözüm Şeması

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

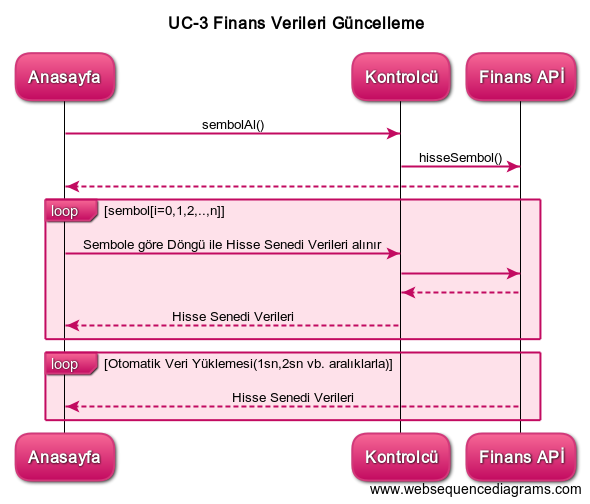


Şekil 7.12: UC-1 alternatif

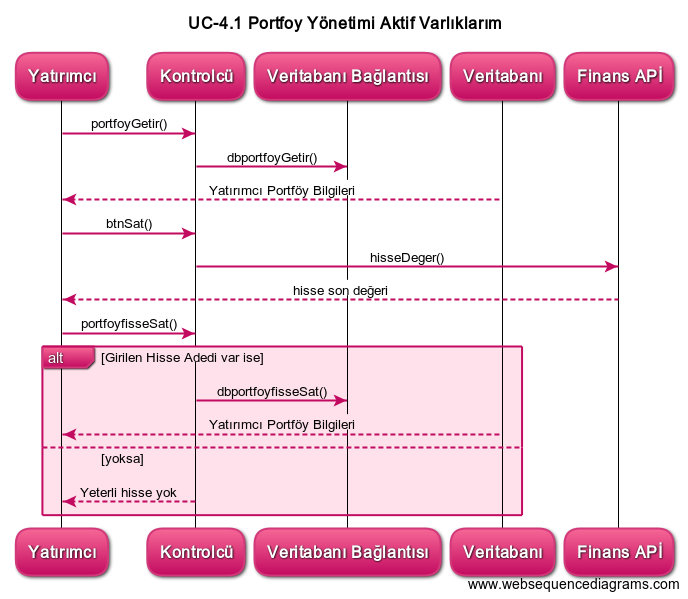
Şekil 7.13: UC-1.1 alternatif



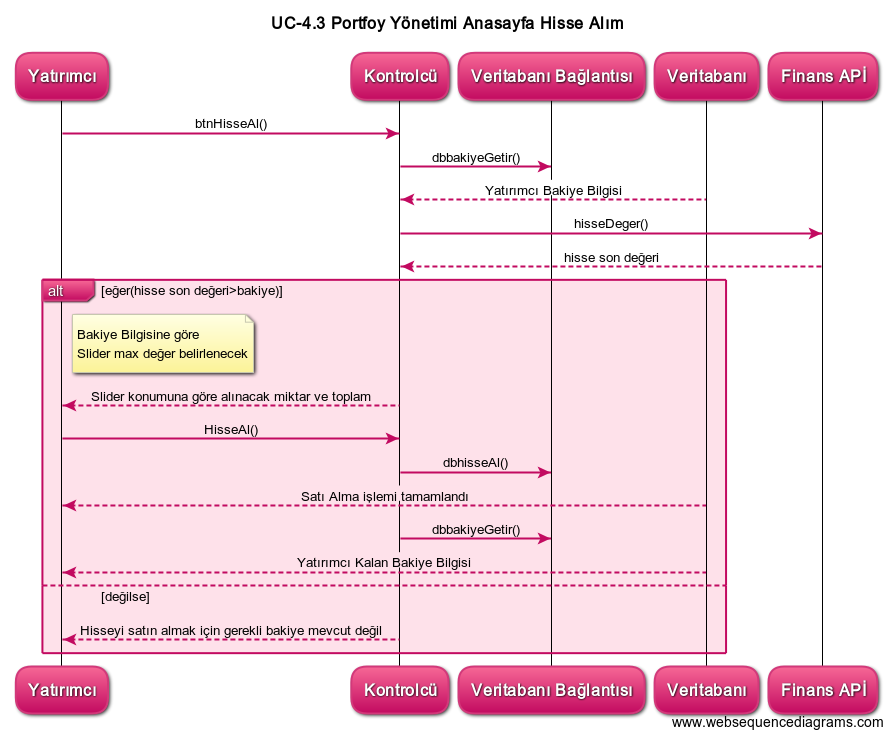
Şekil 7.14: UC-2 alternatif



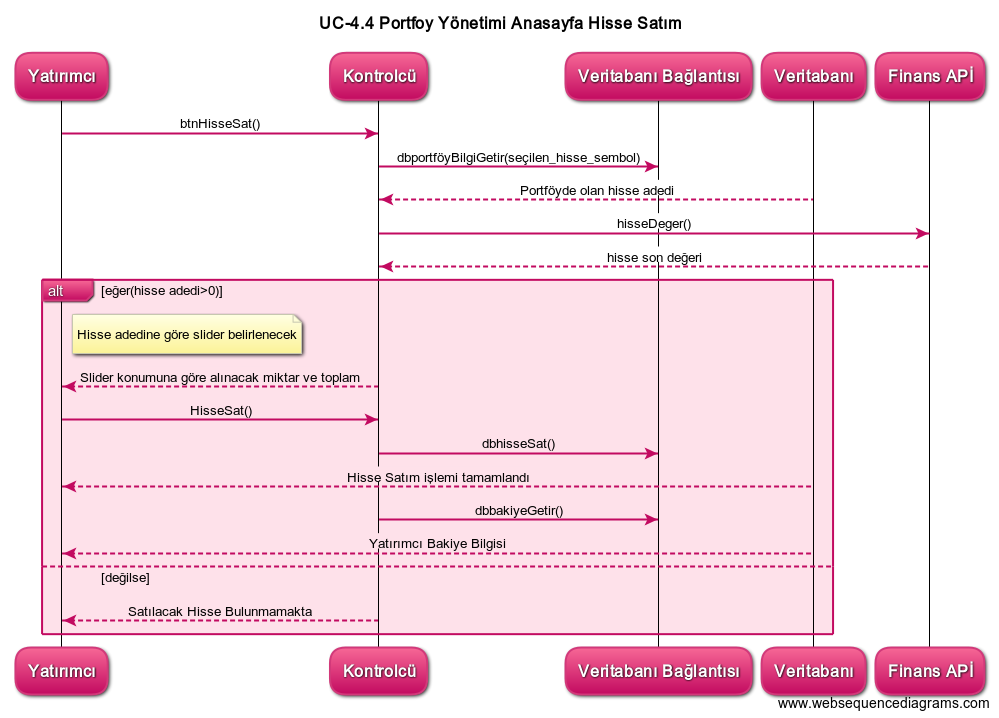
Şekil 7.15: UC-3 alternatif



Şekil 7.16: UC-4.1 alternatif



Şekil 7.17: UC-4.3 alternatif



Şekil 7.18: UC-4.4 alternatif

7.3 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 veritabanı erişim tasarım modelinin akıllı bir uygulaması olan Nesne İlişkisel Model (ORM) modeli, yalnızca sistemde kullanılan kalıcı depolama teknolojileriyle etkileşim kurmak için kullanılmıştır. Bu model, veritabanına 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 veritabanı 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 eşzamansız olarak yapıldığından, doğası gereği daha hızlı bir kullanıcı deneyimi üretmiştir. Bu hedeflerin her ikisine de aktif olarak ulaşmaya çalıştık.

KAYNAKÇA

1-Örnek Proje

2-<https://www.websequencediagrams.com/>

3-Ders Notları

4-<https://medium.com/gokhanyavas/interaction-diagram-etkile%C5%9Fim-diyagramlar-47f7241d96bd>