

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



YAZILIM MÜHENDİSLİĞİ

16011002 – Aykut AKDENİZ

16011033 – Yusuf ANI

16011081 – Fazlı BOZATAY

YSOFT Yazılım Evi İnsan Kaynakları Sistemi

Mayıs, 2020

İÇİNDEKİLER

ŞEKİL LİSTESİ	iii
TABLO LİSTESİ	iv
1 Proje Planı	1
1.1 Proje Alan Tanımı	1
1.2 Kabul ve Kısıtlar	1
1.2.1 Proje İş-Zaman Çizelgesi	1
1.3 Ekip Organizasyon Şeması ve Görev Dağılımı	1
1.4 Teknik Fizibilite	3
1.4.1 Yazılım fizibilitesi	3
1.4.2 Donanım fizibilitesi	3
1.5 Zaman fizibilitesi	4
1.6 Ekonomik fizibilite	4
1.7 Yasal fizibilite	4
1.8 Risk Tablosu	5
1.8.1 Organizasyon	5
1.8.2 Kullanışlılık	6
1.8.3 Pazarlama	6
1.8.4 Bakım	7
1.8.5 Doğruluk	7
1.8.6 Personel	8
2 Kullanım Senaryoları	9
2.1 Kullanım Senaryosu Modeli	9
2.2 Kullanım Senaryoları	9
2.2.1 Yeni Çalışanın Bilgilerinin Girilmesi	9
2.2.2 Proje Başlatma	11
2.2.3 Proje Bitirme	11
2.2.4 Tazminat Hesaplama	12
3 Nesneye Dayalı Modelleme	14
3.1 Sınıf Diyagramı	14

4 Nesneye Dayalı Tasarım	16
4.1 Sıralama Diyagramı	16
4.1.1 Proje Oluşturma Sıralama Diyagramı	16
4.1.2 Proje Bitirme Sıralama Diyagramı	17
4.1.3 Projeleri Görüntüleme Sıralama Diyagramı	17
4.1.4 Proje Başlatma Sıralama Diyagramı	18
4.1.5 Çalışan Oluşturma Sıralama Diyagramı	19
4.1.6 Belirli Tipteki Çalışanları Görüntüleme Sıralama Diyagramı . .	20
4.1.7 Tüm Çalışanları Görüntüleme Sıralama Diyagramı	20
4.1.8 Çalışanı Projeye Atama Sıralama Diyagramı	22
4.2 Aktivite Diyagramı	23
4.3 Durum Diyagramı	24
4.4 Tasarım Sınıf Diyagramı	25
A Gantt Diyagramı	27

ŞEKİL LİSTESİ

Şekil 1.1	Ekip Organizasyon Şeması	2
Şekil 1.2	Görev Dağılımı	2
Şekil 1.3	Risk Tablosu	5
Şekil 1.4	Risk Grafiği	5
Şekil 2.1	Use Case Diyagramı	9
Şekil 3.1	Nesneye Dayalı Modelleme	15
Şekil 4.1	Proje Oluşturma Sıralama Diyagramı	16
Şekil 4.2	Proje Bitirme Sıralama Diyagramı	17
Şekil 4.3	Projeleri Görüntüleme Sıralama Diyagramı	17
Şekil 4.4	Proje Başlatma Sıralama Diyagramı	18
Şekil 4.5	Çalışan Oluşturma Sıralama Diyagramı	19
Şekil 4.6	Belirli Tipteki Çalışanları Sıralama Diyagramı	20
Şekil 4.7	Tüm Çalışanları Görüntüleme Sıralama Diyagramı	21
Şekil 4.8	Çalışanı Projeye Atama Sıralama Diyagramı	22
Şekil 4.9	Aktivite Diyagramı	23
Şekil 4.10	Durum Diyagramı	24
Şekil 4.11	Durum Diyagramı	26
Şekil A.1	Proje Gantt Diyagramı -Detaylar-	27
Şekil A.2	Proje Gantt Diyagramı -Şema-	28

TABLO LİSTESİ

Tablo 1.1	Minimum Sistem Gereksinimleri	3
Tablo 1.2	Tavsiye Edilen Sistem Gereksinimleri	4
Tablo 1.3	Personel Gider Tablosu	4
Tablo 1.4	Donanım Gider Tablosu	4

1

Proje Planı

1.1 Proje Alan Tanımı

FAY Yazılım şirketi olarak müşterilere çalışanlarının ve projelerinin koordinasyonunun sağlandığı bir İnsan Kaynakları(İK) sistemi üretilecektir. Bu sistem sayesinde insan kaynakları sorumluları şirketin projelerine çalışan atayabilecek, çalışanların maaş ve tazminatlarını hesaplayabileceklerdir.

1.2 Kabul ve Kısıtlar

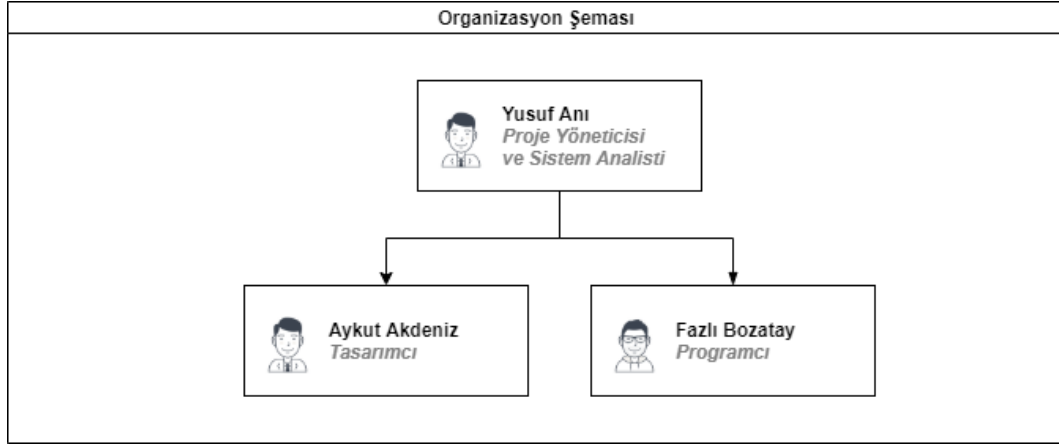
Üretilen bu insan kaynakları sisteminin başka şirketlere de satılabileceği kabul edilecektir. Yönetim çalışanları da bu sistemi kullanabilecektir.

1.2.1 Proje İş-Zaman Çizelgesi

Proje gantt diyagramı Appendix A.1 ve A.2 de bulunmaktadır.

1.3 Ekip Organizasyon Şeması ve Görev Dağılımı

Organizasyonumuz 3 kişiden oluşmakta olup şeması Şekil 1.1’de gösterilmektedir. Görev dağılımı Şekil 1.2’de gösterilmektedir.



Şekil 1.1 Ekip Organizasyon Şeması



Şekil 1.2 Görev Dağılımı

1.4 Teknik Fizibilite

1.4.1 Yazılım fizibilitesi

- İşletim Sistemi: İşletim sistemi tercih edilirken kullanıcı dostu arayüzü ve birçok programa desteği sebebiyle Windows tercih edilmiştir. Windows'a alternatif olarak MacOS veya Linux de tercih edilebilirdi.
- Programlama Dili: Web uygulamalarında tercih edilmesi sebebiyle ve programcıların bu projeden deneyim kazanabilmesi için Javascript dili tercih edilmiştir. Javascript'e alternatif olarak python, java, .NET tercih edilebilecek seçeneklerdir.
- Geliştirme Ortamı: Öğrenciye ücretsiz olmasının yanı sıra kullanıcıya birçok öneri ve yardımcı olması için eklenen özellikleri sebebiyle WebStorm tercih edilmiştir. WebStorm'a alternatif olarak Visual Studio tercih edilebilirdi.
- Kütüphaneler:
 - Server: Server tarafında, Javascript kullanımında server geliştirme sürecinde kullanım sıklığı ve bağımlılık desteklerinin fazlalığı sebebiyle Node.js tercih edilmiştir.
 - Client: Client tarafında, yazılımcıların daha önce kısa bir süre deneyimlemesi sebebiyle ve bu alanda kullanımının yanı sıra Redux bağımlılığına desteği sebebiyle React.js tercih edilmiştir.

1.4.2 Donanım fizibilitesi

Minimum Sistem	
Gereksinim	Niceliği
İşletim Sistemi	Windows veya Linux
RAM	4GB
Gerekli Disk Alanı	1GB
CPU	2.0GHz

Tablo 1.1 Minimum Sistem Gereksinimleri

Tavsiye Edilen Sistem	
Gereksinim	Niceliği
İşletim Sistemi	Windows/Linux
RAM	8GB
Gerekli Disk Alanı	5GB
CPU	2.5GHz

Tablo 1.2 Tavsiye Edilen Sistem Gereksinimleri

1.5 Zaman fizibilitesi

Zaman fizibilitesi gantt diyagramı ile Appendix A.1 ve A.2’de bulunmaktadır.

1.6 Ekonomik fizibilite

Personel Giderleri				
Personel	Kişi	Gün	Günlük ücret	Toplam ücret
Proje Yöneticisi	1	70	180 TL	12600 TL
Tasarım Personeli	1	14	100 TL	1400 TL
Yazılım Personeli	1	63	150 TL	9450 TL
Toplam gider				23450 TL

Tablo 1.3 Personel Gider Tablosu

Donanım Giderleri			
Donanım	Adet	Birim ücret	Toplam ücret
Laptop	3	4200 TL	12600 TL
Toplam gider			12600 TL

Tablo 1.4 Donanım Gider Tablosu

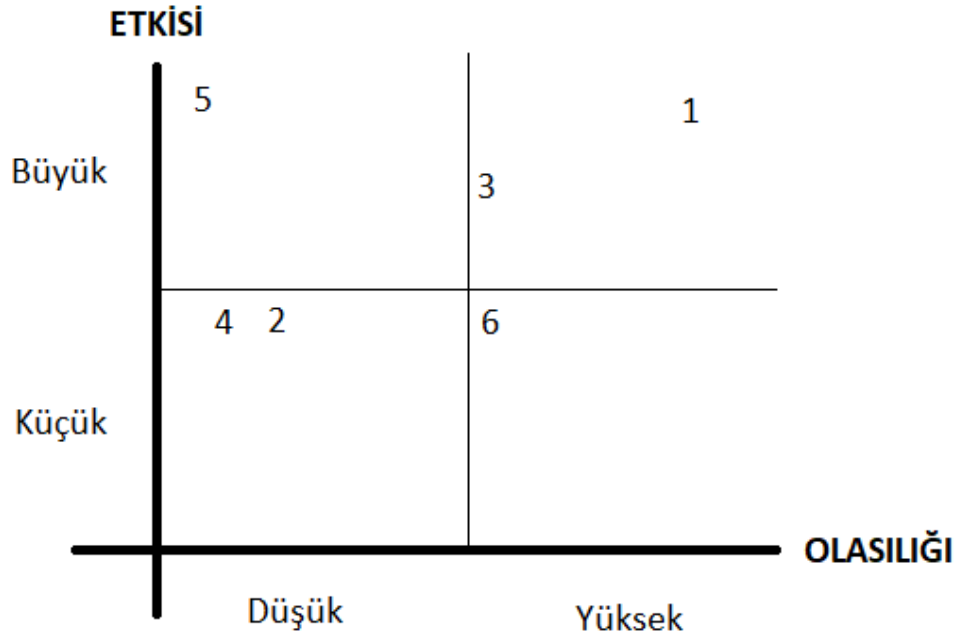
1.7 Yasal fizibilite

Sistem geliştirilirken açık kaynak kodlu yazılım dili kullanılmasından dolayı herhangi yasal bir bariyer bulunmamaktadır. Sistemin görsel arayüzü de sistem programcıları tarafından geliştirilip hazır alınmadığı için herhangi yasal bir engel bulunmamaktadır.

1.8 Risk Tablosu

Risk ID	Adı	Türü, Grubu	Etkisi	Olasılık
1	Organizasyon	Proje	Büyük	Yüksek
2	Kullanışlılık	Teknik	Orta	Düşük
3	Pazarlama	İş	Büyük	Orta
4	Bakım	Teknik	Orta	Düşük
5	Doğruluk	Teknik	Büyük	Düşük
6	Personel	Proje	Orta	Orta

Şekil 1.3 Risk Tablosu



Şekil 1.4 Risk Grafiği

1.8.1 Organizasyon

Olasılık: Yüksek

Etki: Büyük

Türü: Proje

İşaretleri:

- Farklı görevlerdeki çalışanların uyum göstermemesi.
- Görevlerin açık bir şekilde anlaşılması.

- Projede yapılması gereken işlerin sık sık aksaması ve ertelenmesi.

Önlemler:

- Çalışanları daha önce başka projelerde çalışmış kişiler arasından seçmek.
- Çalışanların görev tanımlarını iyi anladığından emin olmak.
- Zamanlamada sıkıntı çekiliyorsa projeye daha fazla çalışanı dahil etmek.

1.8.2 Kullanışlılık

Olasılık: Düşük

Etki: Orta

Türü: Teknik

İşaretleri:

- Müşterilerin sistemi kullanmakta zorlanması
- Müşterilerin sistemi tam kapasitede kullanamaması.
- Müşteriler tarafından sistemin kullanımına yönelik çok fazla soru gelmesi.

Önlemler:

- Sistemin kullanıcı arayüzünü kolay tasarlayarak kullanıcıya daha rahat bir deneyim sağlamak.
- Sistemi kullanacak şirkete veya departmana sistemi iyi bilen birinin gönderilmesi ve sistemin kullanımının iyi bir şekilde anlatılması.
- Sistem içerisinde kullanıma dair ipuçların olduğu bilgi kutuları bulunması.

1.8.3 Pazarlama

Olasılık: Orta

Etki: Büyük

Türü: İş

İşaretleri:

- Satışların düşük olması.

- Sistemin varlığının insanlar tarafından bilinmiyor olması.

Önlemler:

- Pazarlama departmanında nitelikli insanları çalıştırmak.
- Pazarlama bütçesini artırmak.
- Sosyal medya üzerinden reklam vermek.

1.8.4 Bakım

Olasılık: Düşük

Etki: Orta

Türü: Teknik

İşaretleri:

- Sistemin çalışmasında aksaklıkların başlaması.
- Sistemin yeni donanımlar üzerinde çalışamaması.
- Yeni ihtiyaçların karşılanamaması.

Önlemler:

- Sistemin düzenli güncellemeler ile stabilleştirilmesi.
- Sistem ilk ortaya konurken iyi bir kod altyapısına sahip olması.
- Test yapacak daha fazla kişi görevlendirmek.

1.8.5 Doğruluk

Olasılık: Düşük

Etki: Büyük

Türü: Teknik

İşaretleri:

- Müşteri şirketin çalışanları tarafından sisteme yönelik yapılan itirazlar.
- Maaşların veya tazminatların yanlış hesaplanması.

- Proje atama işleminde yanlışlıklar olması.

Önlemler:

- Sistem, değişen koşullara göre güncellemeler almalıdır.
- Müşterinin istekleri iyi bir şekilde sisteme yansıtılmalıdır.

1.8.6 Personel

Olasılık: Orta

Etki: Orta

Türü: Proje

İşaretleri:

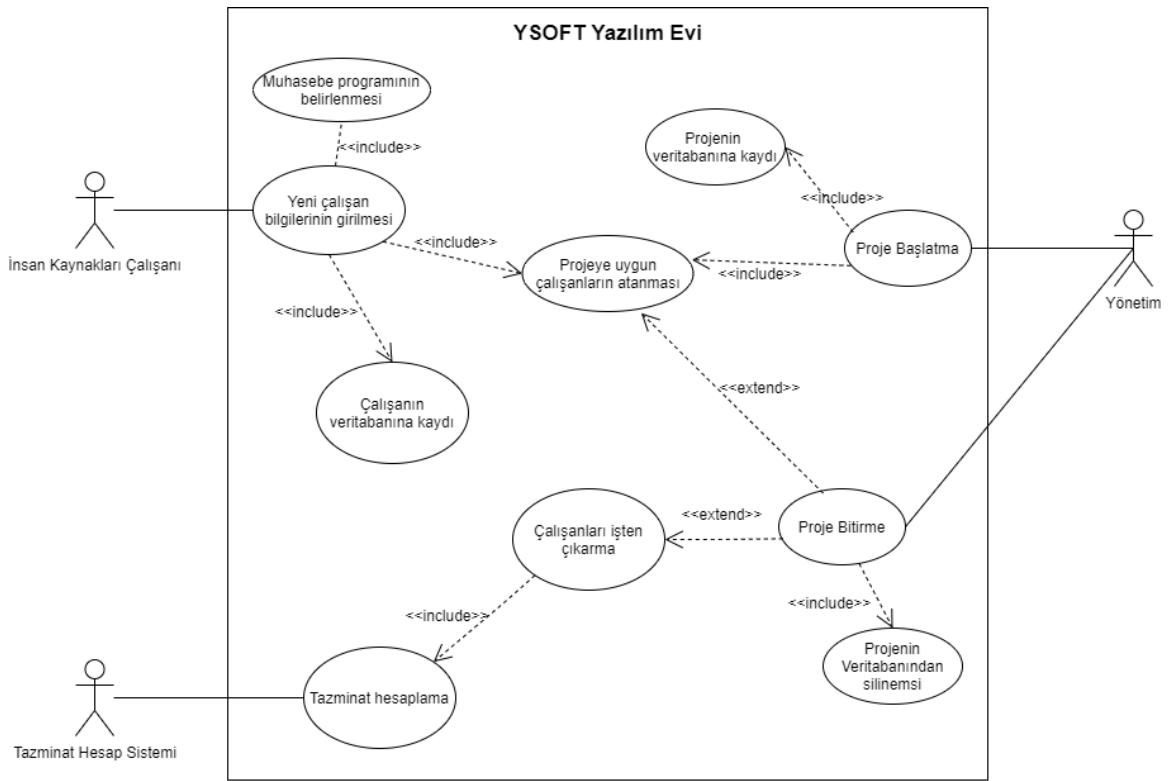
- Sistem üzerinde çalışan programcı, tasarımcı gibi kişilerin projeyi terk etmesi ya da işten ayrılması.
- Çalışan kişilerin sisteme yeterli işgücünü sunamaması.

Önlemler:

- Varolan çalışanlara daha fazla iş vermek.
- Yeni çalışanlar işe almak.
- Çalışanları motive edecek imkanlar sunmak.

2 Kullanım Senaryoları

2.1 Kullanım Senaryosu Modeli



Şekil 2.1 Use Case Diyagramı

2.2 Kullanım Senaryoları

2.2.1 Yeni Çalışanın Bilgilerinin Girilmesi

Birincil Aktör: İnsan Kaynakları Çalışanı

İlgililer:

- İnsan Kaynakları Çalışanı: Yeni işe alınan çalışanın bilgilerinin sisteme aktarılmasını ister.

Ön Koşullar:

- İnsan kaynakları çalışanı sisteme giriş yapmıştır.

Son Koşullar:

- Çalışanın bilgileri veritabanına işlenmiştir.
- Çalışanın maaşının hangi muhasebe programı ile hesaplanacağı belirlenmiştir.
- Çalışan uygun bir projeye atanmıştır.

Ana Senaryo:

1. Yeni işe alınan çalışanın isim, soyisim gibi kimlik bilgileri girilir,
2. Çalışanın hangi pozisyonda çalışacağı(programı,tasarımcı vs.) girilir.
3. Çalışanın maaşının hangi muhasebe programıyla hesaplanacağı bilgisi girilir.
4. Çalışanın iş tecrübesi yıl olarak eklenir.
5. Tüm bilgilerin doğru girildiğinden emin olunduktan sonra onay tuşuna basılır.

Alternatif Senaryolar:

2a. Çalışan birden fazla pozisyon için başvurmuştur.

1. Çalışanın sahip birden fazla pozisyon işaretlenir.
2. Projeler içerisinde hangi pozisyona daha fazla ihtiyaç var ise sistem bu pozisyon için kayıt yapılacağını söyler.

4a.Varolan çalışanın girilmesi

1. Kullanıcıya varolan bir çalışanın girilmeye çalışıldığına dair uyarı gösterilir.
2. Kullanıcı hata yaptığını düşünerek bilgileri kontrol eder.

2.2.2 Proje Başlatma

Birincil Aktör: Yönetim Çalışanı

İlgililer:

- Yönetim Çalışanı: Başlatılmak istenen yeni projenin bilgilerini sisteme aktarmak ve projeyi başlatmak ister.

Ön Koşullar:

- Yönetim Çalışanı sisteme giriş yapmıştır.

Son Koşullar:

- Proje bilgileri veritabanına işlenmiştir.
- Girilen projeye uygun çalışanlar atanmıştır.

Ana Senaryo:

1. Projenin ne tür bir proje olduğu bilgisi girilir.
2. Projenin yöneticisinin kim olacağı yöneticiler listesinden seçilir.
3. Projede hangi pozisyondan kaç kişi çalışacağı bilgisi girilir.
4. Girilen bilgilerin doğruluğundan emin olununca onay tuşuna basılır.

Alternatif Senaryolar:

2a. Yönetici pozisyonunda boşta kimse bulunmamaktadır.

1. Varolan çalışanlar arasından belirli koşulları sağlayan bir kişiler (tecrübe gibi) gösterilerek bu kişiler arasından seçim yapılması istenir.

2.2.3 Proje Bitirme

Birincil Aktör: Yönetim Çalışanı

İlgililer:

- Yönetim Çalışanı: Aktif projeler arasından istediği projeyi bitirmek ister.

Ön Koşullar:

- Yönetim Çalışanı sisteme giriş yapmıştır.

Son Koşullar:

- Proje aktif projeler içerisinde çıkarılmıştır.
- Projeye işi biten çalışanlar yeni projelere atanmıştır.
- Kalan çalışanlar işten çıkarılmıştır ve tazminatları hesaplanmıştır.

Ana Senaryo:

1. Aktif projeler ekranda gösterilir.
2. Bitirilmek istenen proje seçilir ve onay tuşuna basılır.

2.2.4 Tazminat Hesaplama

Birincil Aktör: Tazminat Hesap Sistemi

İlgililer:

- Tazminat Hesap Sistemi: Tazminat miktarını hesaplamak ister.
- Çalışan: Tazminat miktarının hesaplanmasını ister.

Ön Koşullar:

- Çalışanın işten çıkarılmış olması.

Son Koşullar:

- Tazminat miktarının hesaplanarak gösterilmesi.

Ana Senaryo:

1. Yönetimce bir projenin bitirilme kararı alınır.

2. Yönetim çalışanı proje bitirme ekranından bu projeyi bulur.
3. Onay tuşuna basar.
4. Bitirilen projedeki çalışanlar uygun projelere atanır.
5. Herhangi bir projeye atanamayan çalışanlar işten çıkarılır.
6. İşten çıkarılan kişilerin bilgileri tazminat hesap sistemine gönderilir.
7. Tazminat hesap sisteminden gelen tazminat miktarları ayrı bir menüde görüntülenir.

Alternatif Senaryolar:

2a. Tüm kişiler projeye atanır.

1. Tazminat hesap sistemine herhangi bir bilgi gitmez.
2. Tazminat miktarları görüntülenmez.

7a. Gönderilen bilgiler yetersizdir.

1. Yetersiz görülen çalışan bilgileri doldurularak tekrar tazminat hesap sistemine gönderilmesi istenir.
2. Sisteme daha sonra giriş yapacak olan insan kaynakları çalışanına eksik bilgisi bulunan çalışanlar listelenir.
3. İnsan kaynakları çalışanı bu bilgileri girer ve tazminat hesap sistemine tekrar gönderilir.
4. Hesaplanan tazminat miktarları menüdeki tazminatlar listesine eklenir.

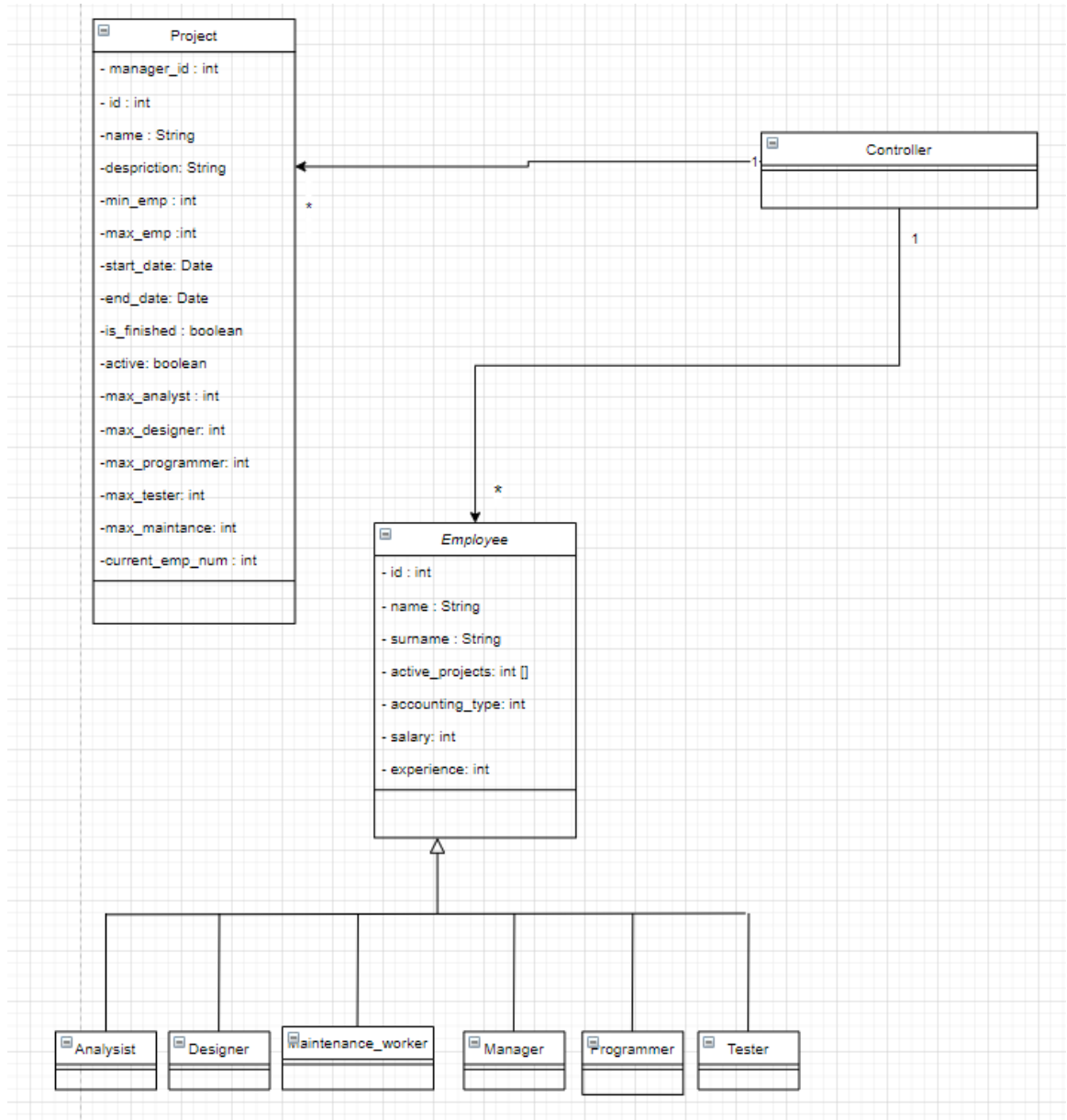
Yapılan sistemde 2 ana aktör - proje ve çalışan - ve bunların arasındaki bağlantıları kuran kontrolcü sınıf ihtiyacı vardır.

Her projede yönetici , id , isim , açıklama , minimum çalışan sayısı , maksimum çalışan sayısı , başlangıç ve bitiş tarihleri , bitip bitmediği durumu , başlayıp başlamadığı durumu , maksimum eleman sayıları ve sahip olduğu anlık eleman sayısı tutulması gerekmektedir.

Her çalışanda id, isim , soyad , aktif projeleri , muhabebe türü , maaşı ve deneyimi tutulması gerekmektedir.

Kontrolcü sınıfın bir özelliği bulunmasına gerek yoktur. Amacı sadece kullanılacak fonksiyonlar ile haberleşmeyi kullanıcı arayüzü , proje ve çalışan arasında gerçekleştirmektir.

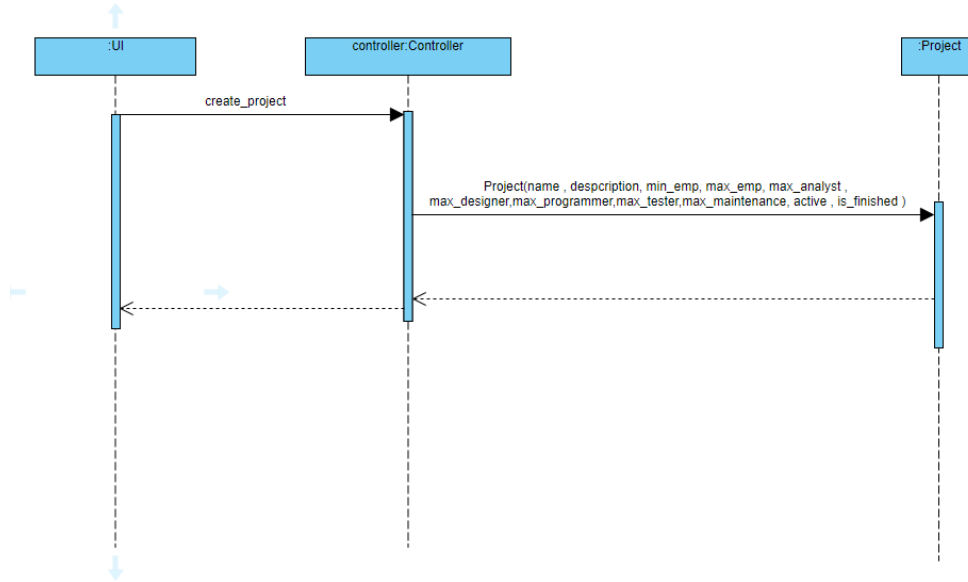
3.1 Sınıf Diyagramı



Şekil 3.1 Nesneye Dayalı Modelleme

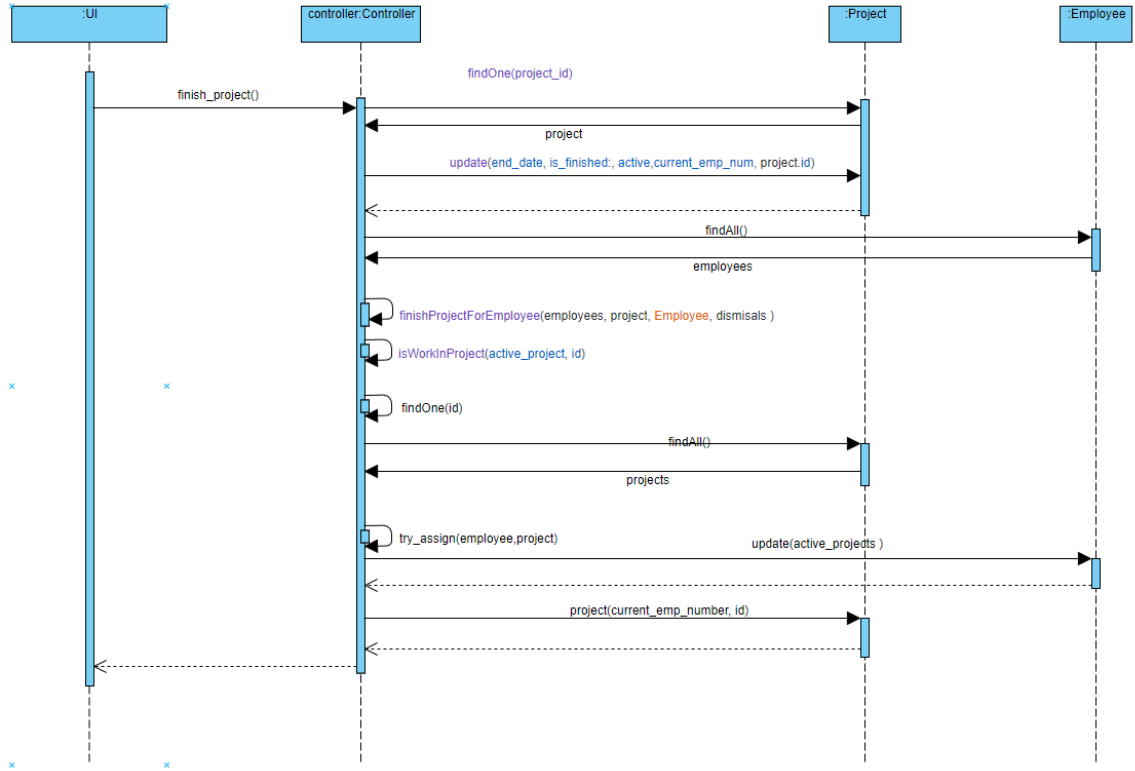
4.1 Sıralama Diyagramı

4.1.1 Proje Oluşturma Sıralama Diyagramı



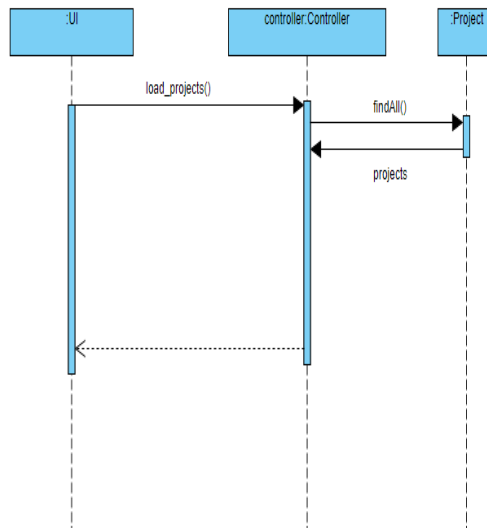
Şekil 4.1 Proje Oluşturma Sıralama Diyagramı

4.1.2 Proje Bitirme Sıralama Diyagramı



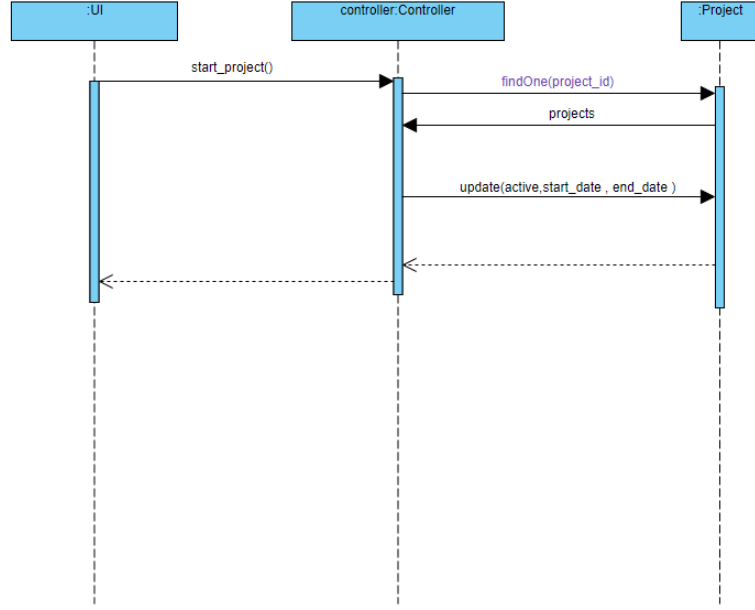
Şekil 4.2 Proje Bitirme Sıralama Diyagramı

4.1.3 Projeleri Görüntüleme Sıralama Diyagramı



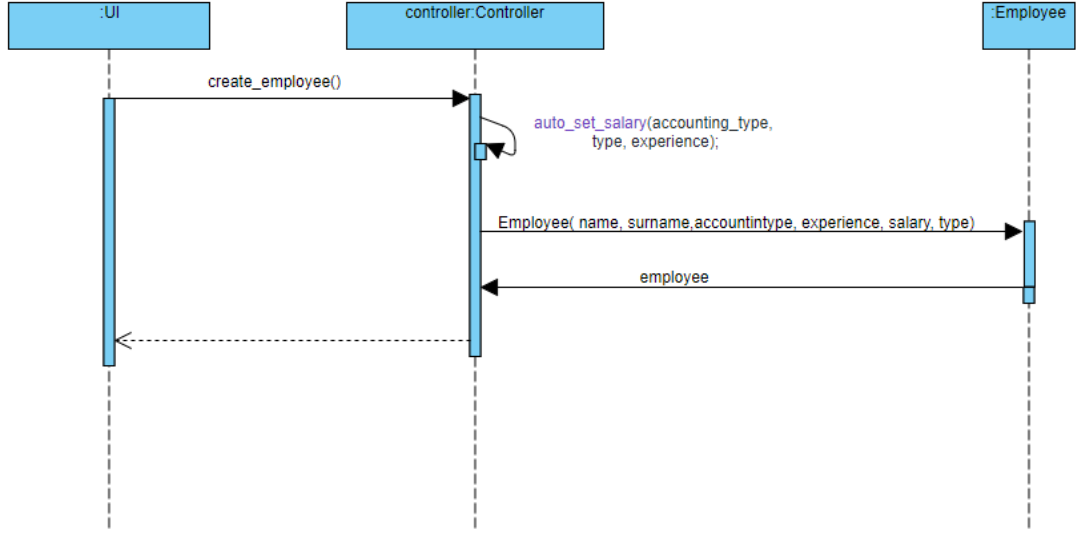
Şekil 4.3 Projeleri Görüntüleme Sıralama Diyagramı

4.1.4 Proje Başlatma Sıralama Diyagramı



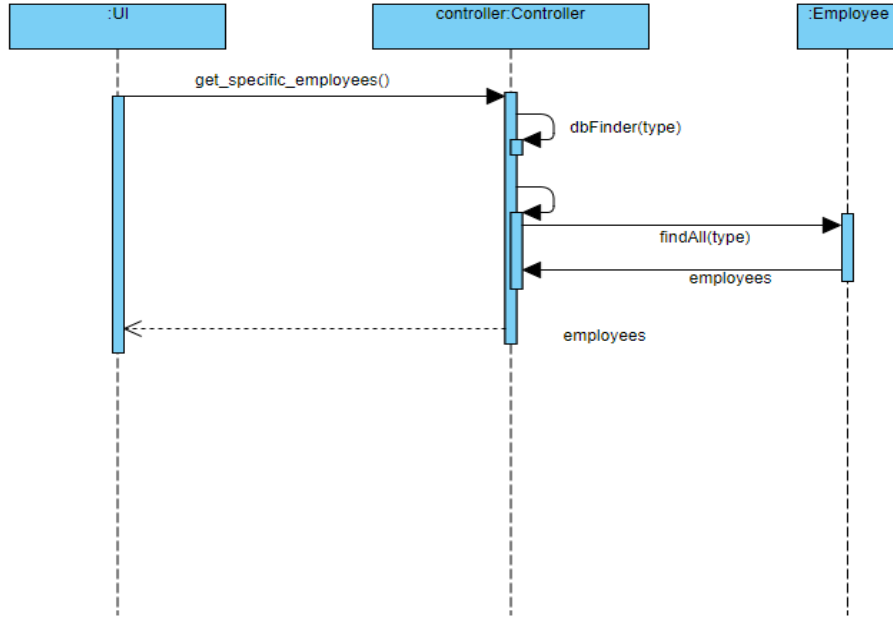
Şekil 4.4 Proje Başlatma Sıralama Diyagramı

4.1.5 Çalışan Oluşturma Sıralama Diyagramı



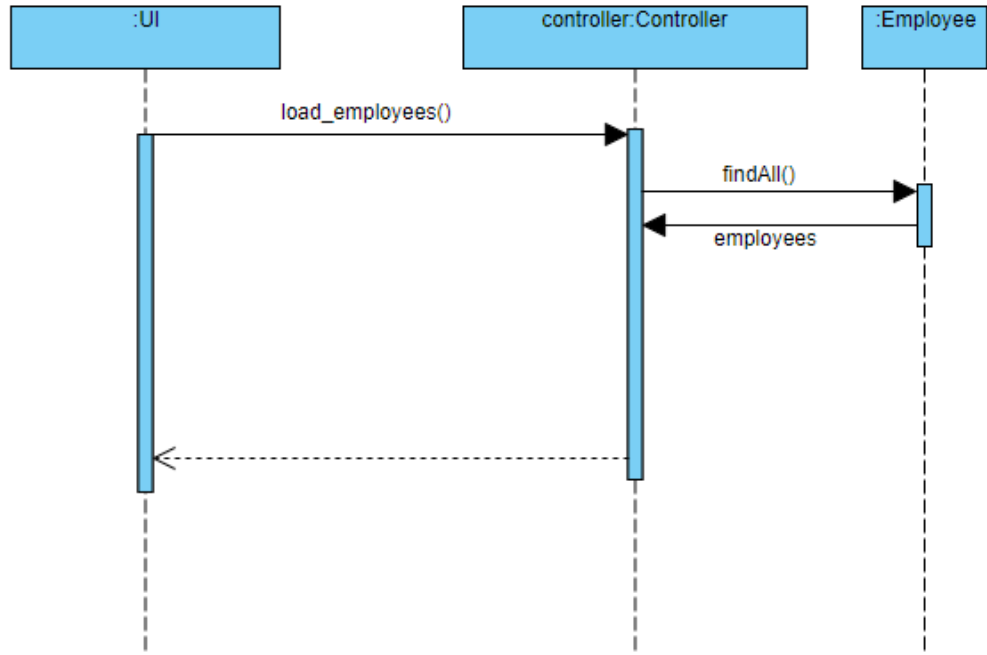
Şekil 4.5 Çalışan Oluşturma Sıralama Diyagramı

4.1.6 Belirli Tipteki Çalışanları Görüntüleme Sıralama Diyagramı



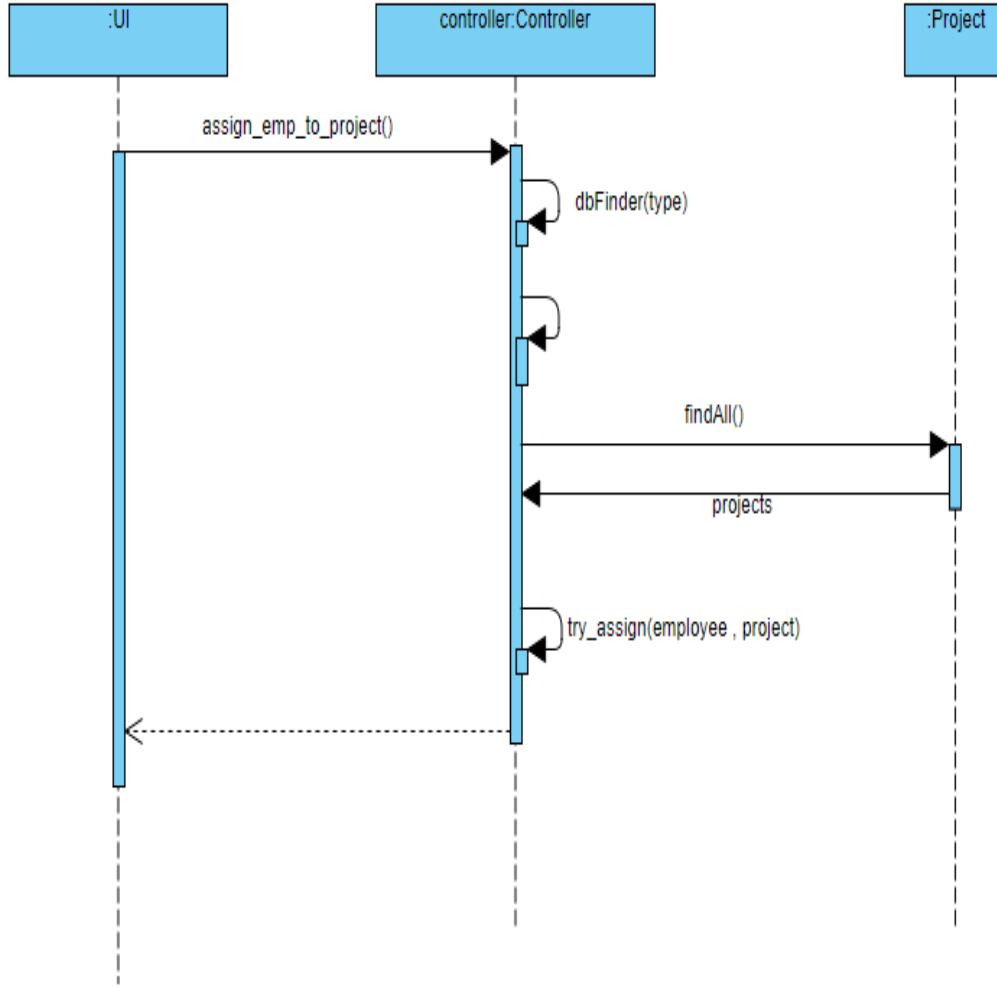
Şekil 4.6 Belirli Tipteki Çalışanları Sıralama Diyagramı

4.1.7 Tüm Çalışanları Görüntüleme Sıralama Diyagramı



Şekil 4.7 Tüm Çalışanları Görüntüleme Sıralama Diyagramı

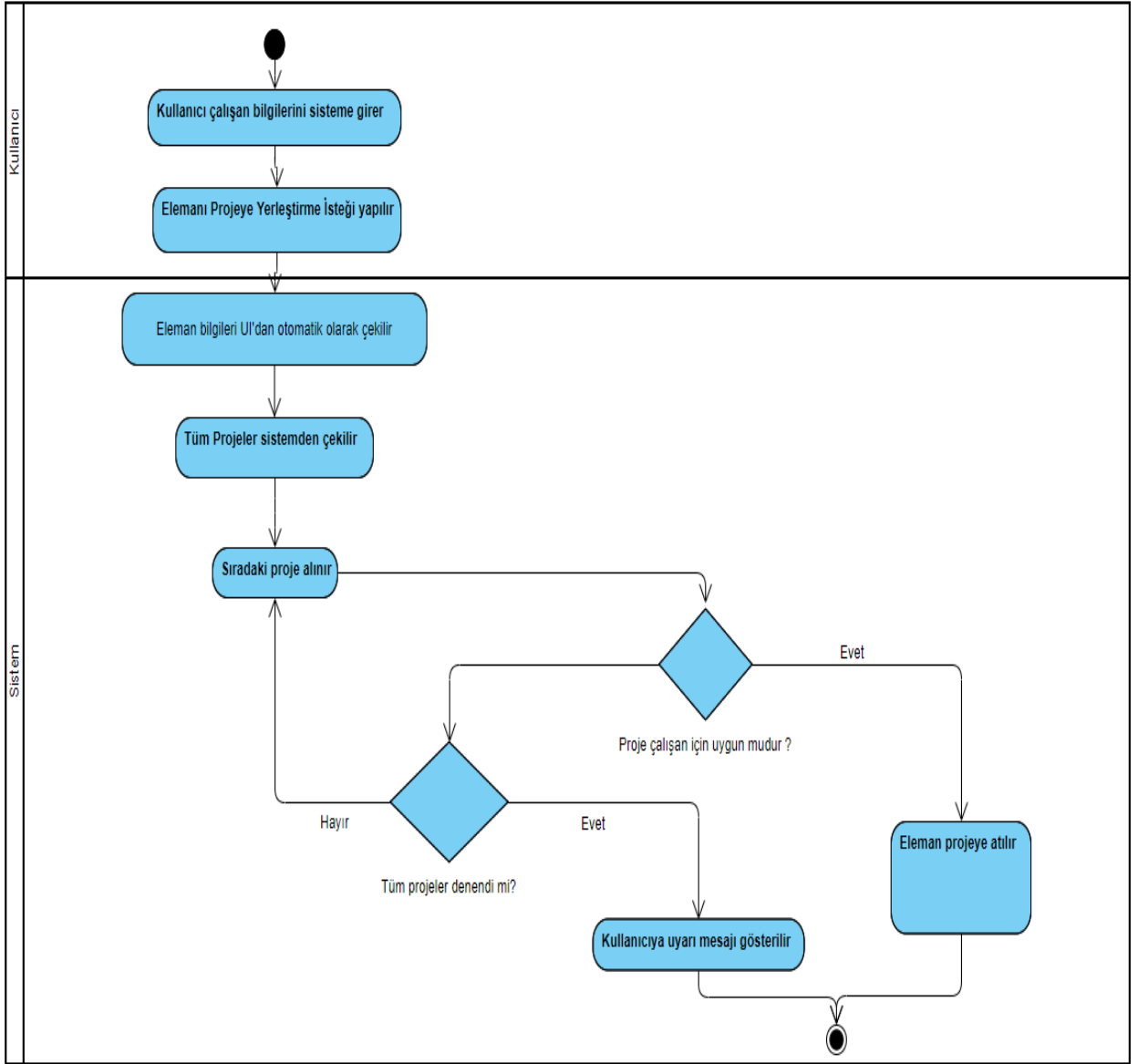
4.1.8 Çalışanı Projeye Atama Sıralama Diyagramı



Şekil 4.8 Çalışanı Projeye Atama Sıralama Diyagramı

4.2 Aktivite Diyagramı

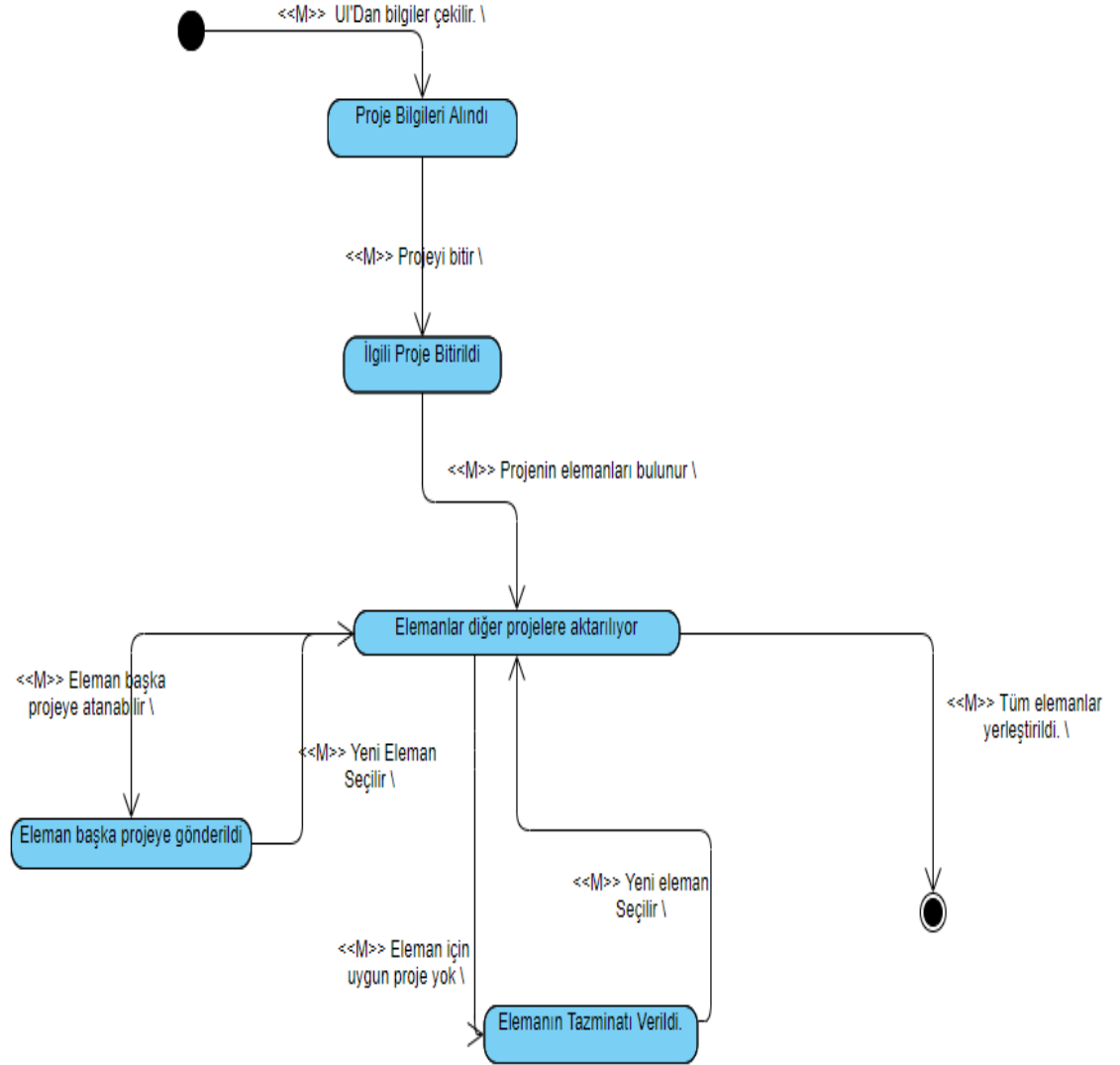
Aktivite diyagramı için çalışanın kendisi için en uygun projeye atanması seçilmiştir. Kullanıcı "Assign Employee to Project" tuşuna basıldığında sistem kullanıcı arayüzünden elemanın bilgilerini çeker. Ardından veritabanında yüklü bütün projeler sisteme getirilir. Tüm bu projeler içinde otomatik olarak elemanın proje için uygunluğu aranır. Eğer eleman için uygun proje bulunabilirse atanır. Bulunamazsa sistem kullanıcıyı atanamadığına dair bilgilendirir.



Şekil 4.9 Aktivite Diyagramı

4.3 Durum Diyagramı

Durum diyagramı gösterimi için bir projenin bitirilmesi gösterilmiştir. Seçilen projenin bilgileri kullanıcı arayüzünden getirilir. Ardından ilgili bilgileri ile sistemdeki kayıtlı proje bulunur ve bitirilir. Projenin bitmesinin ardından açıkta kalan her eleman için diğer projelere uygun ise aktarma işlemi, uygun değil ise tazminat verilmesi işlemi yapılır ve işten çıkarılır. Bu işlem eleman sayısı bitinceye kadar devam eder.



Şekil 4.10 Durum Diyagramı

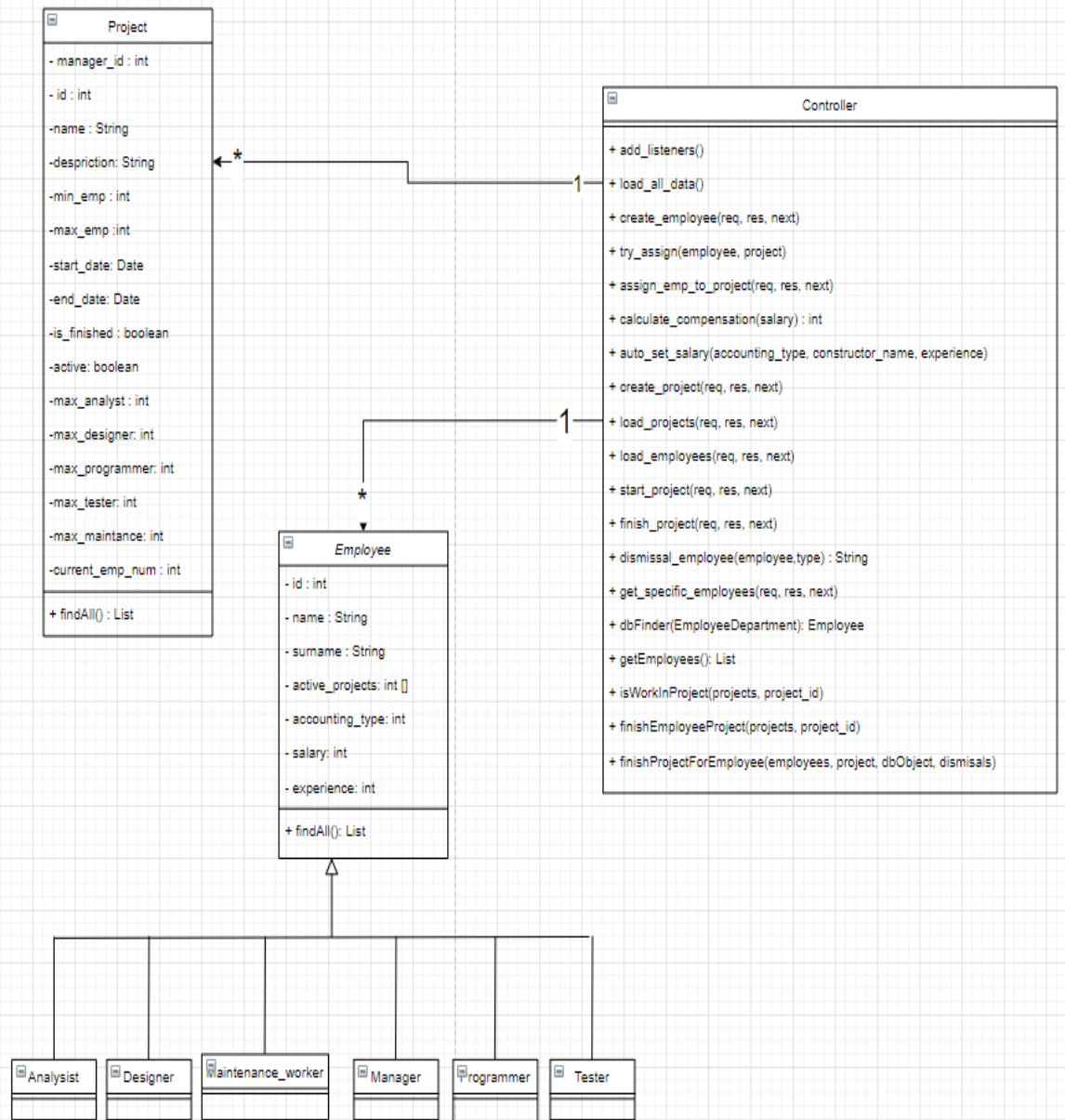
4.4 Tasarım Sınıf Diyagramı

Tasarım sınıf diyagramında ek olarak gerekli olan fonksiyonlar diyagrama eklenmiştir.

Proje ve çalışan sınıfında gerekli objeleri veri tabanından çeken findAll fonksiyonu eklenmiştir.

Kontrolcü sınıfta :

1. add listeners : Arayüzdeki dinamik bağlantıları ayarlayan fonksiyon
2. load all data : Veritabanından proje ile ilgili bilgileri çeken fonksiyon
3. create employee : Yeni çalışan üreten fonksiyon
4. try assign : Çalışanı verilen projeye atamayı deneyen fonksiyon
5. assign emp to project : çalışanı sırayla tüm projeleri atama için çalışan fonksiyon
6. calculate compensation : Tazminat hesabı yapan fonksiyon
7. auto set salary : Çalışanın maaşını belirleyen fonksiyon
8. create project : Yeni proje üreten fonksiyon
9. load project : Projeleri veri tabanından çeken fonksiyon
10. load employees : Çalışanları veritabanından çeken fonksiyon
11. start project : İlgili projeyi başlatan fonksiyon
12. finish project : İlgili projeyi bitiren fonksiyon
13. dismissal employee : Verilirse kullanıcın belirttiği web servisten verilmezse sistemin dahili tamzinat fonksiyonu çağıran çalışanı işten çıkaran fonksiyon
14. get spesific employee : Spesifik olarak çalışan türündeki tüm çalışanları döndüren fonksiyon
15. dbFinder : Çalışan türüne göre veritabanını getiren fonksiyon
16. getEmployees : Tüm çalışanları getiren fonksiyon
17. isWorkInproject : İlgili çalışanın verilen projede olup olmadığını kontrol eden fonksiyon
18. finishEmployeeforProject : Bir çalışanın ilgili projesini bitiren fonksiyon
19. finishProjectForEmployee : İlgili projenin tüm çalışanları için o projeyi bitiren fonksiyon eklenmiştir.



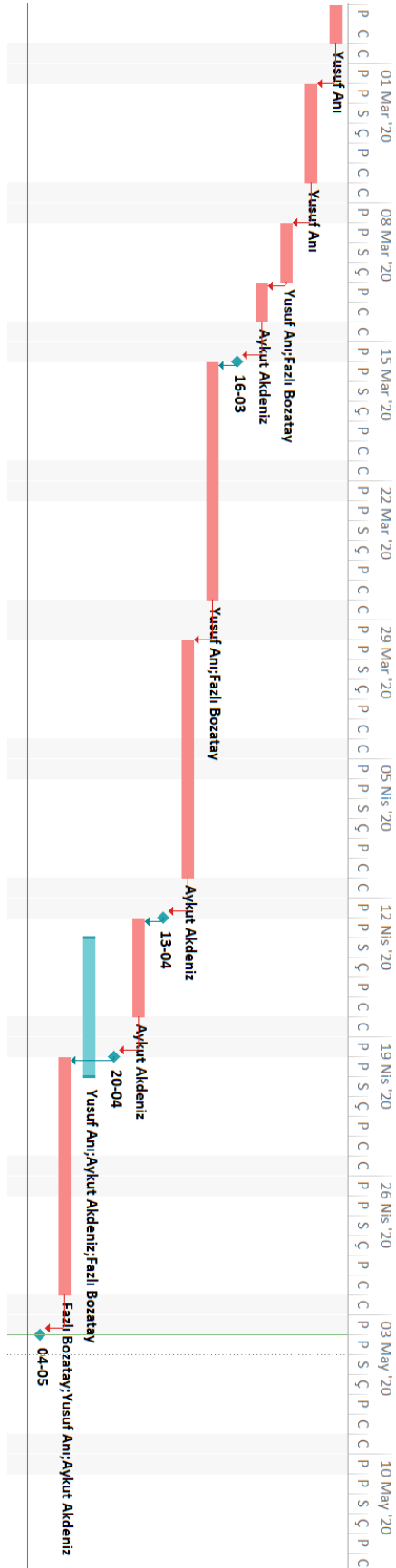
Şekil 4.11 Durum Diyagramı

A

Gantt Diyagramı

Görev	Görev Adı	Süre	Başlangıç	Bitiş	Öncüller	Kaynak Adları
1	On Tasarım	2 gün	Per 27-02-20	Cum 28-02-20		Yusuf Anı
2	Sistem Mimarisini Modelleme	5 gün	Pzt 02-03-20	Cum 06-03-20	1	Yusuf Anı
3	Nesneye Yönelik Modelleme	3 gün	Pzt 09-03-20	Çar 11-03-20	2	Yusuf Anı; Fazlı Bozatay
4	Veritabanı Modelleme	2 gün	Per 12-03-20	Cum 13-03-20	3	Aykut Akdeniz
5	Modelleme Tamamlanması	0 gün	Pzt 16-03-20	Pzt 16-03-20	4	
6	Sistemin Programlanması	2 hf	Pzt 16-03-20	Cum 27-03-20	5	Yusuf Anı; Fazlı Bozatay
7	Veritabanı Bağlantıları	2 hf	Pzt 30-03-20	Cum 10-04-20	6	Aykut Akdeniz
8	Programlama ve Veritabanı Tamamlanması	0 gün	Pzt 13-04-20	Pzt 13-04-20	7	
9	Görsel Arayüz Yapımı	1 hf	Pzt 13-04-20	Cum 17-04-20	8	Aykut Akdeniz
10	Arayüz Tasarımının Tamamlanması	0 gün	Pzt 20-04-20	Pzt 20-04-20	9	
11	Dokümantasyon	5 gün	Sal 14-04-20	Cum 20-04-20		Yusuf Anı; Aykut Akdeniz; Fazlı Bozatay
12	Rapor	10 gün	Pzt 20-04-20	Cum 01-05-20	10	Fazlı Bozatay; Yusuf Anı; Aykut Akdeniz
13	Proje Teslimi	0 gün	Pzt 04-05-20	Pzt 04-05-20	12	Yusuf Anı; Aykut Akdeniz; Fazlı Bozatay

Şekil A.1 Proje Gantt Diyagramı -Detaylar-



Şekil A.2 Proje Gantt Diyagramı -Şema-