



**Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği**

**Yazılım Mühendisliği Dersi Proje Ödevi**

**Dersin Yürütücüsü: Dr.Öğr.Üyesi Yunus Emre SELÇUK**

**Mayıs, 2023**

**Konu: 2 - YSOFT Yazılım Evi İnsan Kaynakları Sistemi**

**19011049 - Emircan Kirez**

**20011001 - Buse Nur Pekmezci**

**20011019 - Muhammed Eren Şekkeli**

**20011024 - Sait Yalçın**

**20011041 - Recep Selim Ağırman**

1. Proje Alan Tanımı .....	3
2. Kabul ve Kısıtlar .....	3
3. Proje İş-Zaman Çizelgesi .....	4
4. Ekip Organizasyon Şeması ve Görev Dağılımları .....	5
5. Proje Risk Tablosu .....	6
6. Analiz .....	7
6.1. Kullanım Senaryosu Diyagramı .....	7
6.2. Kullanım Senaryoları .....	8
a. Senaryo – Çalışan Ekleme .....	8
b. Senaryo – Proje Ekleme .....	9
c. Senaryo – Proje Bitirme .....	9
d. Senaryo – Çalışanı Projeye Bağlama .....	10
e. Senaryo – Çalışan Çıkarma .....	11
f. Senaryo – Maaş Hesapla .....	11
g. Senaryo – Tazminat Hesaplama .....	12
6.3. İzlenebilirlik Tablosu .....	13
6.4. Kavramsal Sınıf Diyagramı .....	14
7. Tasarım .....	15
7.1. Ayrıntılı Sınıf Diyagramı .....	15
7.2. Sıralama (Sequence) Diyagramları .....	16
7.3. Durum (State) Diyagramları .....	16
7.4. Etkinlik (Activity) Diyagramları .....	17
8. Birim Testi Sınamaları .....	19

## 1. Proje Alan Tanımı

YSOFT Yazılım evi için insan kaynaklarının ihtiyaçlarını karşılayacak bir proje gerçekleştirilmesi istenmektedir. Bu projede çalışanların bilgileri ve çalışanların çalıştığı projelerin bilgisi tutulacaktır. İnsan kaynakları bu sistemi kullanarak yeni çalışan ekleyebilecek ve sistem tarafından otomatik proje atanmasını gerçekleştirilebilecektir. Projelerle ilgili gereksinimler ve kısıtlar insan kaynakları tarafından girilebilmekte ve gerektiği zaman çalışanların maaşları harici muhasebe sistemleri vasıtasıyla hesaplanıp insan kaynakları tarafından girilerek güncellenebilmektedir. Bir proje bittiğinde projede bulunan çalışanların atanma ve işten çıkarılma işlemlerini gerçekleştirecektir.

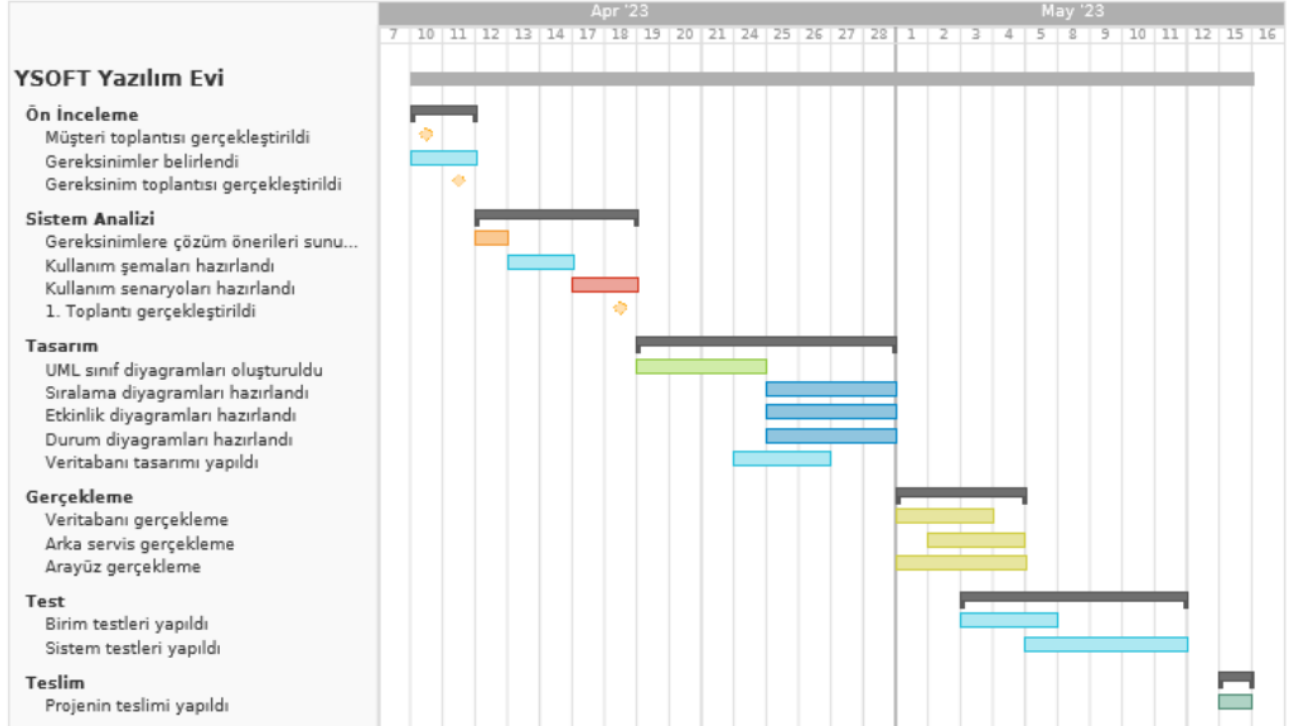
## 2. Kabul ve Kısıtlar

Bu kısımda projeyi gerçekleştirirken kabul ettiğimiz bilgiler yer almaktadır.

1. Bir çalışan aynı anda yalnızca tek bir projede çalışabilir.
2. İnsan kaynakları, projenin tamamlandığını ya da sonlandırıldığını sisteme girer ve projede yer alan ilgili çalışanlar işten çıkarılabilir veya başka bir projeye aktarılabilir.
3. Maaş hesaplamaları ve benzeri işlemler dış hizmet olarak sunulacaktır.
4. Çalışan maaşları çalışan eklenirken brüt maaş olarak sisteme kaydedilir. Daha sonra insan kaynakları kullanılan muhasebe programına göre net maaşı ilgili yerlere bildirir.
5. Net maaş sistemde tutulmaz, gerek duyulmamıştır.
6. Şirketten bir çalışanın çıkarılması, ilgili çalışan(lar)ın atandığı projelerin bulunmadığı durumlarda gerçekleştirilebilir.
7. İnsan kaynakları, pozisyon eksikliği veya ihtiyaçlarına göre personel alımı yapacaktır.
8. Çalışanlar işe alınırken, projeye atanamazsa işe alınmaz. Uyarı verilir.
9. Projeler boş olarak açılabilir ve projelerin geçerli olup olmadığı bir etiket ile belirtilir.
10. Çalışan eklenirken ve herhangi bir proje tamamlandığında, çalışan(lar) projelerde bulunan uygun slotlara otomatik olarak atanacaktır.
11. Projeler tamamlandıktan sonra projeye dair bilgiler veritabanından silinmeyecek, sadece proje inaktif hale getirilecek.
12. Kişisel Verilerin Korunması Kanunu (KVKK) gereği, çalışan işten çıkarıldığında çalışanların bilgileri veritabanından silinecek. Tazminat hesabı için çalışan bilgileri ekranda görüntülenecektir.
13. İnsan kaynakları, tazminat miktarını yönlendirilen sitede manuel olarak hesaplayacaktır. Tazminat miktarı sisteme girilmeyecektir.

### 3. Proje İş-Zaman Çizelgesi

Bu kısımda projeye dair gantt diyagramı bulunmaktadır.



Şekil 3.1 - Proje Gantt Diyagramı

#### 4. Ekip Organizasyon Şeması ve Görev Dağılımları

Bu kısımda ekip organizasyon şeması ve görev dağılımları bulunmaktadır. Şekil 4.1 şemasında yazılanlar asıl görevler olup ekip üyelerinin görev aldıkları diğer alanlar şemanın altında bulunur.



Şekil 4.1 - Ekip Organizasyon Şeması

- Emircan Kirez: İş analisti, sistem tasarımcısı
- Buse Nur Pekmezci: İş analisti, sistem tasarımcısı
- Muhammed Eren Şekkeli: Yazılım geliştirici
- Sait Yalçın: Yazılım geliştirici
- Recep Selim Ağırman: İş analisti, sistem tasarımcısı

## 5. Proje Risk Tablosu

Bu kısımda projeye dair risk ve riske dair alınabilecek tedbirleri içeren risk tablosu bulunmaktadır.

Risk	Etki	Olasılık	Yönetim Stratejisi	Tür
Proje ihtiyaçları doğru tespit edilemez.	Gecikme, işlevsiz sistem	Orta	Daha önce benzer projeler üzerinde çalışmış deneyimli personel ile proje gereksinimlerinin doğru bir şekilde belirlenmesi	İş
Çalışanların sayısı yeterli değil veya fazla çalışan var.	Projelerin tamamlanması için yeterli personel olmaması, kaynak israfı	Yüksek	Çalışan sayısının doğru bir şekilde belirlenmesi ve işe alım sürecinin dikkatli yürütülmesi	İş
Tazminat hesaplama hizmeti sunan şirketten beklenmeyen bir hizmet kesintisi yaşanması.	Maaş ve tazminatların zamanında ödenmemesi	Düşük	Yedek bir hizmet sağlayıcısı ile anlaşma yapılması, hizmet sağlayıcısı ile sürekli iletişim halinde olunması	Teknik
Muhasebe programlarının uyumsuzluğu.	Maaşların doğru hesaplanamaması	Orta	Herhangi bir sorunun hızlı bir şekilde çözülebilmesi için uzman bir ekip oluşturulması	Teknik
Sistem güvenliği ihlal edilirse, hassas verilerin kaybedilmesi veya zarar görmesi.	Bilgi güvenliği ihlalleri, müşteri veya çalışanların bilgilerinin çalınması veya sızdırılması	Orta	Güvenlik açıklarının sürekli olarak taranması ve düzeltilmesi, sıkı kimlik doğrulama ve erişim denetimi prosedürleri uygulanması	Teknik
Sistem arızaları veya kesintileri.	Sistem kullanımının engellenmesi, veri kaybı, müşteri veya çalışanların mağduriyeti	Yüksek	Düzenli yedeklemeler, sistem güncellemelerinin ve bakımının yapılması, hızlı müdahale prosedürleri oluşturulması	Teknik

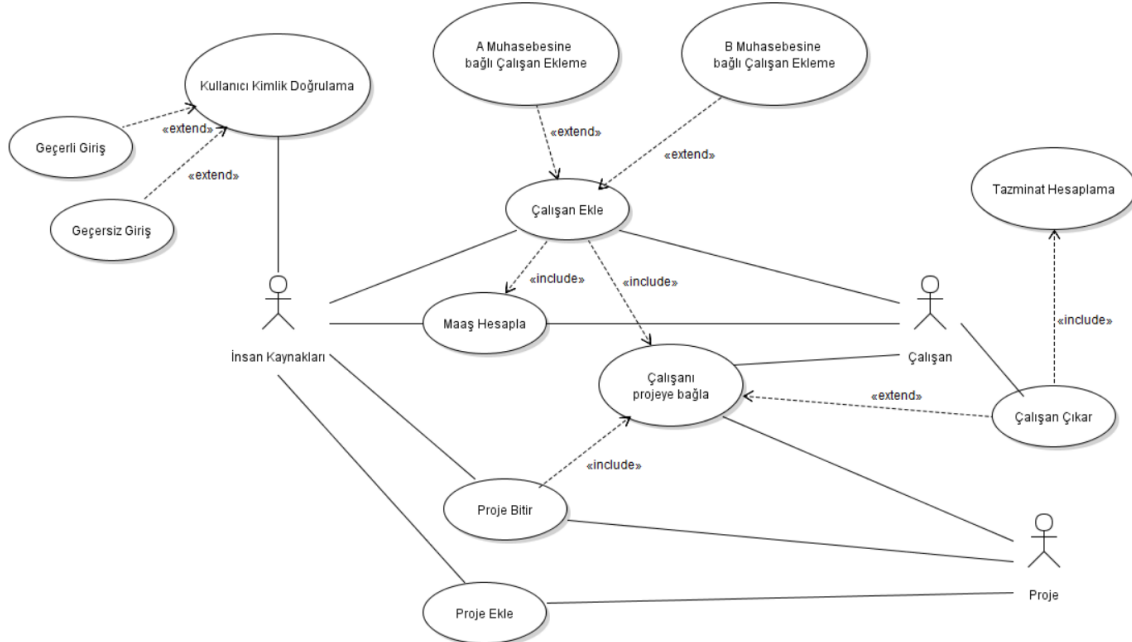
Tablo 5.1 - Proje Risk Tablosu

## 6. Analiz

Bu kısımda analize dair diyagramlar ve kullanım senaryoları bulunmaktadır.

### 6.1. Kullanım Senaryosu Diyagramı

Bu kısımda sistemin genel kullanım senaryosu bulunmaktadır.



Şekil 6.1.1 - Use Case Diyagramı

## 6.2. Kullanım Senaryoları

Bu kısımda kullanım senaryoları bulunmaktadır.

### a. Senaryo – Çalışan Ekleme

Kullanım Senaryosu	Çalışan Ekleme
Birincil Aktör	İnsan Kaynakları
İlgililer ve Beklentileri	İnsan Kaynakları: Şirkette yeni çalışacak olan çalışanın bilgilerini girebilmek ve çalışanı uygun bir projeye atamak ister. Çalışan: Bilgilerinin sisteme girilip uygun projeye atanmak ister.
Ön Koşullar	İnsan kaynakları kendisine daha önceden verilmiş kullanıcı erişim bilgileri ile kendini doğrular. Ekleyeceği çalışana projelerde uygun bir pozisyon bulunması gerekir. Çalışan, işe kabul edilir.
Son Koşullar	Çalışanın bilgileri sisteme kaydedilir ve çalışan uygun bir projeye atanır.
Ana Senaryo	<ol style="list-style-type: none"><li>1. İnsan kaynakları çalışanın bilgilerini girer.</li><li>2. Sistem, istem, girilen çalışan bilgilerinin sistemdeki başka bilgilerle çakışıp çakışmadığını kontrol eder.</li><li>3. Çalışanı uygun projeye bağlar.</li><li>4. Çalışan için harici muhasebe vasıtasıyla hesaplanmış ilk maaş bilgisi insan kaynakları tarafından girilir.</li><li>5. Sistem çalışanın son bilgileri sisteme kaydeder.</li></ol>
Alternatif Akış	<p>2.a. Eklenmeye çalışılan çalışan sistemdeki bilgilerle çakışıyordur.</p> <ol style="list-style-type: none"><li>1. Sistem, çalışanın bilgilerini kaydetmeden işlemi başarısız olarak sonlandırır.</li></ol> <p>3.a. Sistem çalışanın atanabileceği uygun bir proje bulamaz.</p> <ol style="list-style-type: none"><li>1. Atanacak proje olmadığı için sistem, ekrana bir uyarı mesajı döndürür.</li></ol>



b. Senaryo – Proje Ekleme

Kullanım Senaryosu	Proje Ekleme
Birincil Aktör	İnsan Kaynakları
İlgililer ve Beklentileri	İnsan Kaynakları: Projeyi oluşturup projenin kaynak gereksinimlerini girebilmek ister.
Ön Koşullar	Projeyi oluştururken projenin yöneticisi belirlenmelidir.
Son Koşullar	Proje oluşturulup sisteme kaydedilmiş olur.
Ana Senaryo	<ol style="list-style-type: none"><li>1. İnsan kaynakları, proje bilgilerini gereksinimler dahil olmak üzere sisteme girer.</li><li>2. Sistem, projeyi sisteme ilk seferde boş olarak kaydedip geçersiz olarak işaretler.</li></ol>

c. Senaryo – Proje Bitirme

Kullanım Senaryosu	Proje Bitirme
Birincil Aktör	İnsan Kaynakları

İlgililer ve Beklentileri	İnsan Kaynakları: Proje tamamlandığında ya da iptal edildiğinde projeye dahil olan çalışan bilgilerini gözlemlemek ve projeyi sonlandırabilmek ister.
Ön Koşullar	İnsan Kaynakları, birime ait kullanıcı bilgileriyle sisteme giriş yapar.
Son Koşullar	Proje sonlandırılır. Projede görevli çalışanlar farklı projelere atanır ya da işten çıkarılır.
Ana Senaryo	<ol style="list-style-type: none"> <li>1. İnsan Kaynakları çalışanı sisteme giriş yapar.</li> <li>2. İnsan kaynakları sonlandırılmasına karar verilen projeyi bulur.</li> <li>3. İnsan kaynakları projenin sonlandırılması için işlemde bulunur.</li> <li>4. Sistem, projeyi sonlandırır.</li> <li>5. Sistem, sonlandırılan projedeki çalışanları başka projelere atamaya çalışır.</li> </ol>

d. Senaryo – Çalışanı Projeye Bağlama

Kullanım Senaryosu	Çalışanı Projeye Bağlama
Birincil Aktör	İnsan Kaynakları
İlgililer ve Beklentileri	<p>İnsan kaynakları: Yeni çalışanın düzgün bir şekilde uygun projeye atanmasını ister.</p> <p>Çalışan: Kendisinin çalışabileceği bir projeye atanmak ister.</p>
Ön Koşullar	Çalışanın başka bir projede çalışmaması gerekir.
Son Koşullar	Çalışan en uygun projeye atanır. Atanamazsa işten çıkarılır.

Ana Senaryo	<ol style="list-style-type: none"> <li>1. Sistem, çalışan için en uygun projeyi bulur.</li> <li>2. Sistem, çalışanı ilgili projeye atar.</li> </ol>
Alternatif Akış	<p>2.a. Sistem çalışan için uygun proje bulamaz.</p> <ol style="list-style-type: none"> <li>1. İnsan kaynakları, çalışan için tazminat hesabı yapar.</li> <li>2. Sistem çalışanı tazminatıyla beraber işten çıkartır.</li> </ol>

e. Senaryo – Çalışan Çıkarma

Kullanım Senaryosu	Çalışan Çıkarma
Birincil Aktör	Çalışan
İlgililer ve Beklentileri	Çalışan: Atanacağı bir proje yoktur, tazminatını almak ister.
Ön Koşullar	<p>Çalışanın bulunduğu proje sonlandırılır.</p> <p>Çalışanın atanacağı uygun bir proje yoktur.</p>
Son Koşullar	<p>Çalışan tazminatı belirlenir.</p> <p>İnsan kaynakları çalışanın çıkarılma raporunu elde eder.</p>
Ana Senaryo	<ol style="list-style-type: none"> <li>1. İnsan kaynakları, çalışana dair bilgileri girer.</li> <li>2. İlgili servis aracılığıyla tazminat miktarı hesaplanır.</li> <li>3. Sistem, çalışanın bilgilerini insan kaynaklarına ekran çıktısı olarak verir.</li> <li>4. Sistem, çalışan bilgilerini sistemden siler.</li> <li>5. İnsan kaynakları, yazdırılabilir formatın kopyalarını yazdırıp yönetime ve çalışana verir.</li> </ol>

f. Senaryo – Maaş Hesapla

Kullanım Senaryosu	Maaş Hesapla
Birincil Aktör	İnsan Kaynakları

İlgililer ve Beklentileri	<p>İnsan Kaynakları: Çalışanın seçmiş olduğu muhasebe programını kullanarak çalışanın maaşını hesaplamak ister.</p> <p>Çalışan: Seçmiş olduğu muhasebe programı aracılığıyla maaşının doğru hesaplanmasını ister.</p>
Ön Koşullar	İnsan kaynakları, kendisine ait kullanıcı bilgileriyle sisteme giriş yapar.
Son Koşullar	Sistem çalışanın bilgilerini kaydeder.
Ana Senaryo	<ol style="list-style-type: none"> <li>1. İnsan kaynakları, çalışanın seçmiş olduğu muhasebe programını öğrenir.</li> <li>2. Sistem, çalışan için kullanılacak muhasebe programının linkini insan kaynaklarına gösterir.</li> <li>3. İnsan kaynakları, muhasebe programı üzerinden çalışanın net maaşını hesaplar.</li> <li>4. İnsan kaynakları, hesapladığı net maaşı ilgili yerlere bildirir.</li> </ol>

g. Senaryo – Tazminat Hesaplama

Kullanım Senaryosu	Tazminat Hesaplama
Birincil Aktör	İnsan Kaynakları
İlgililer ve Beklentileri	<p>İnsan kaynakları: Çalışanın tazminatını harici bir link ile hesaplamak ister.</p> <p>Çalışan: Tazminatının doğru bir şekilde hesaplanmasını ister.</p>
Ön Koşullar	İnsan kaynakları, kendisine ait kullanıcı bilgileriyle sisteme giriş yapar.
Son Koşullar	Hesaplanan tazminat değerine göre çalışana ödeme yapılması planlanır.
Ana Senaryo	<ol style="list-style-type: none"> <li>1. Sistem, insan kaynaklarına tazminat hesaplamak için kullanacağı web sitesinin linkini gösterir.</li> <li>2. İnsan kaynakları, web sitesine erişim sağlar.</li> <li>3. İnsan kaynakları, web sitesi aracılığıyla çalışanın tazminatını hesaplar.</li> </ol>

### 6.3. İzlenebilirlik Tablosu

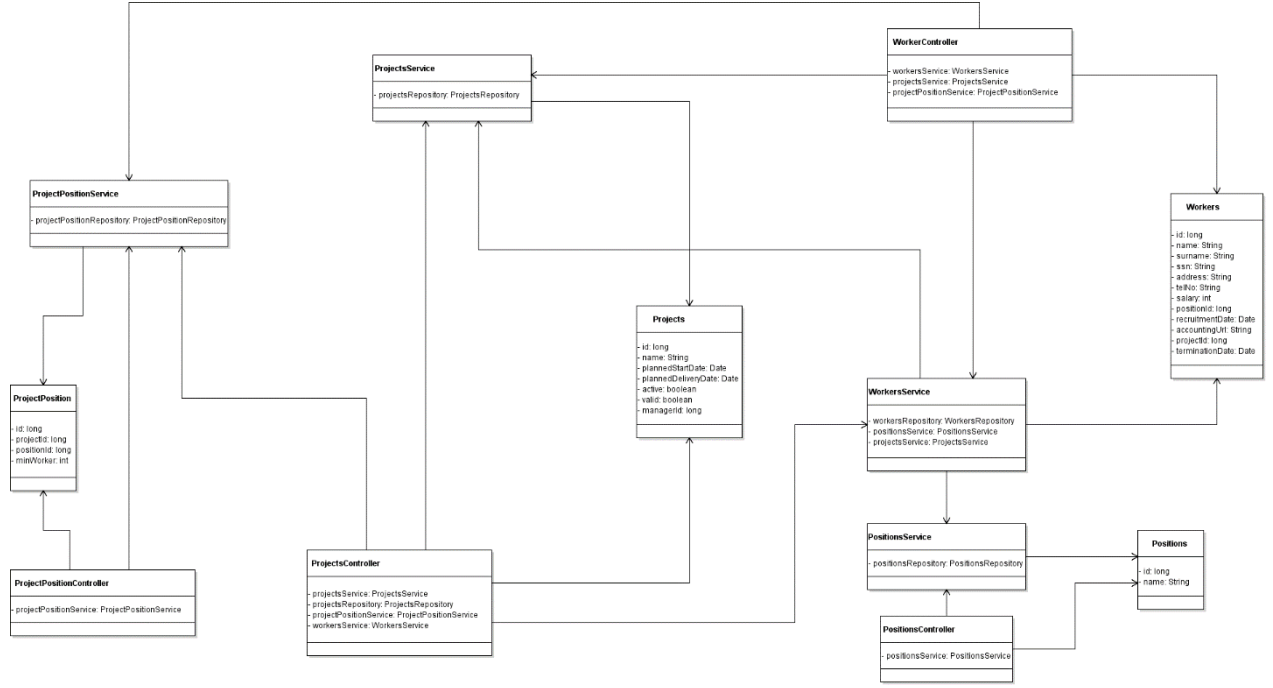
Bu kısımda projeye ait gereksinimlerin ve gereksinimleri sağlayan servis, modül ve sınıfların yer aldığı izlenebilirlik tablosu bulunmaktadır.

Gereksinimler	Authentication Modülü	Proje Sınıfı	Çalışan Sınıfı	Üçüncü Parti Tazminat Hesaplama Servisi	Üçüncü Parti Muhasebe Servisi
İK'nın sisteme girebilmesi	X				
Sisteme yeni çalışan eklenebilmesi			X		
Yeni çalışanın özelliklerinin sisteme girilebilmesi			X		
Çalışanların uygun projelere atanabilmesi		X	X		
Projelerin oluşturulabilmesi		X			
Proje bitiminde çalışanların diğer projelere atanabilmesi		X	X		
Proje bitiminde çalışanların işten çıkarılabilmesi		X	X		
Çalışan tazminatlarının hesaplanabilmesi			X	X	
Çalışanların farklı muhasebe programlarıyla maaşlarının hesaplanabilmesi			X		X

Tablo 7.1 - İzlenebilirlik Tablosu

## 6.4. Kavramsal Sınıf Diyagramı

Bu kısımda sınıflara dair methodların yer almadığı kavramsal sınıf diyagramı bulunur.



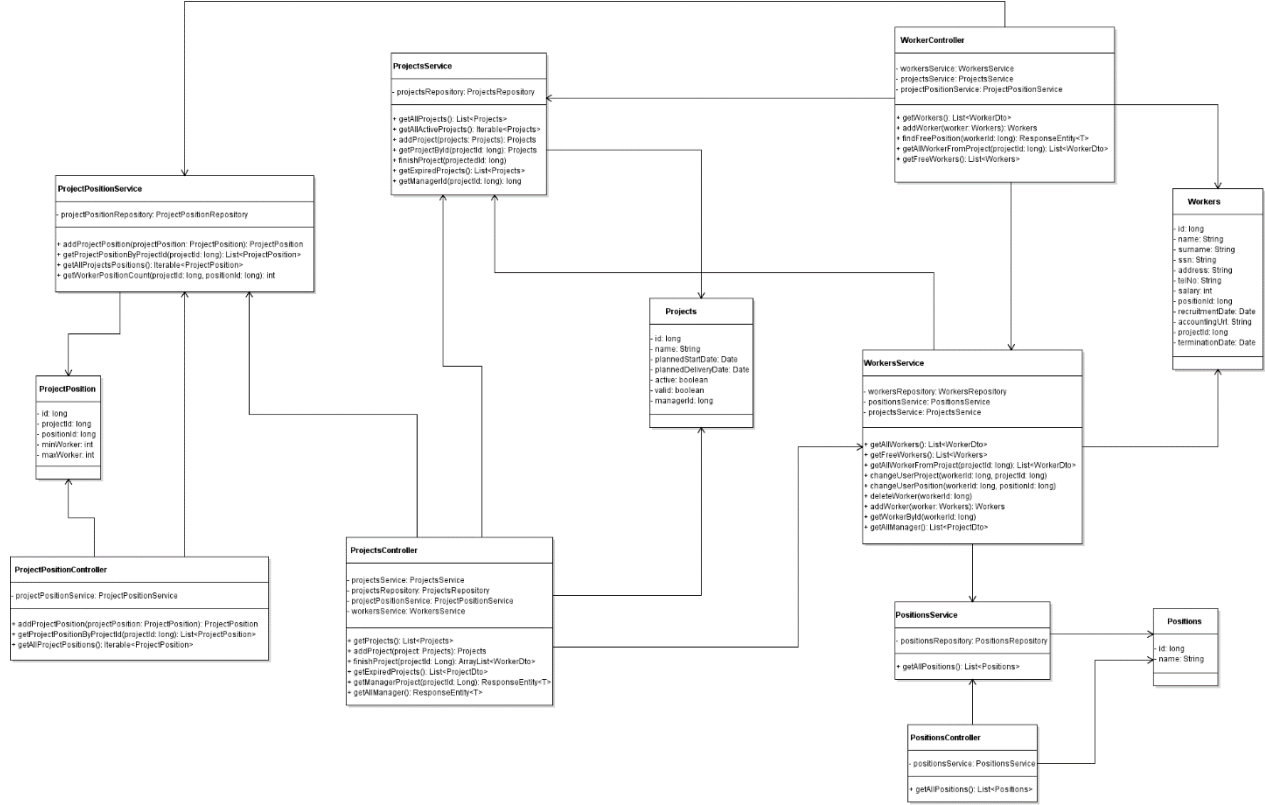
Şekil 6.4.1. - Kavramsal Sınıf Diyagramı

## 7. Tasarım

Bu kısımda tasarıma dair diyagramlar bulunmaktadır.

### 7.1. Ayrıntılı Sınıf Diyagramı

Bu kısımda ayrıntılı sınıf diyagramına yer verilmektedir.

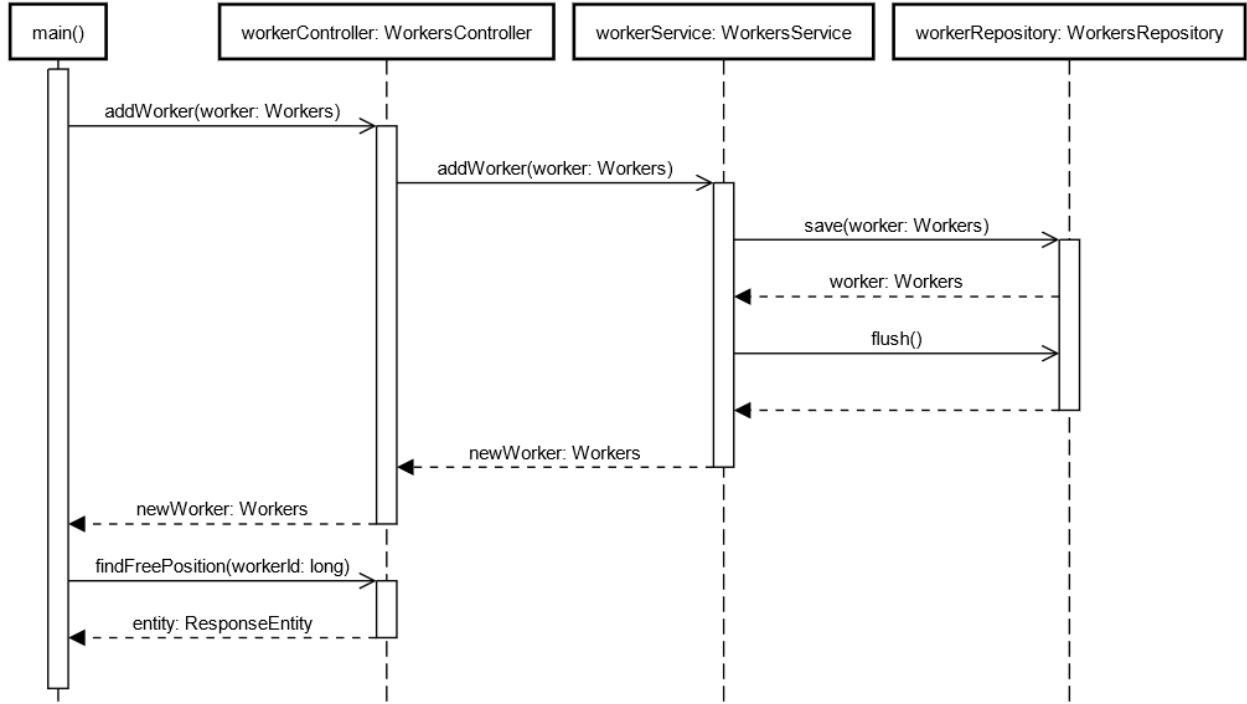


Şekil 7.1.1. - Detaylandırılmış UML Sınıf Diyagramı

## 7.2. Sıralama (Sequence) Diyagramları

Bu kısımda ardışıl diyagramlar bulunmaktadır.

### a. Çalışan Ekleme



Şekil 7.2.1. - Çalışan Ekleme Ardışıl Diyagram

## 7.3. Durum (State) Diyagramları

Bu kısımda durum diyagramları bulunmaktadır

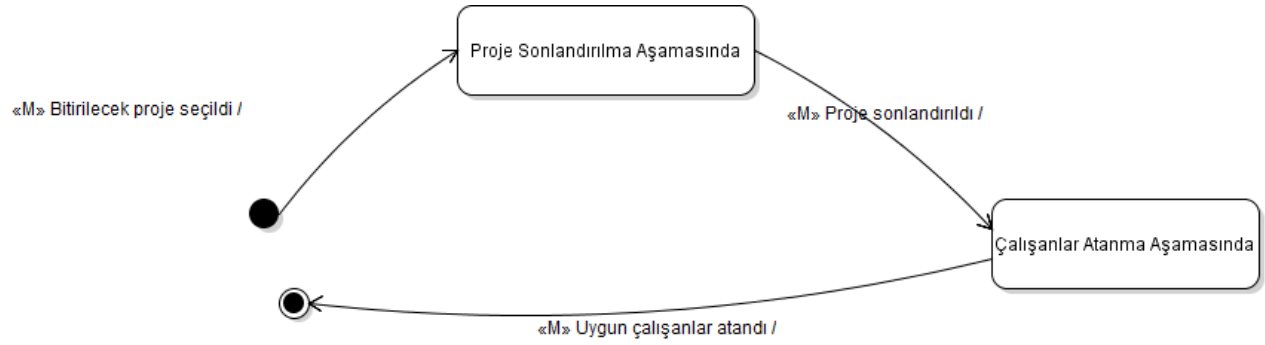
### a. Proje Ekleme



Şekil 7.3.1. - Proje Ekleme Durum Diyagramı

### b. Proje Bitirme



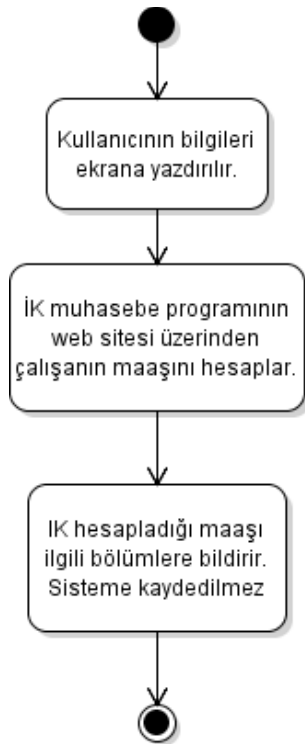


Şekil 7.3.2. - Proje Bitirme Durum Diyagramı

#### 7.4. Etkinlik (Activity) Diyagramları

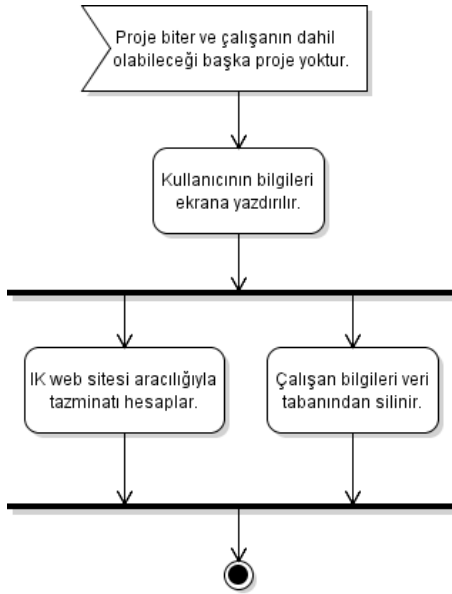
Bu kısımda etkinlik diyagramları bulunmaktadır.

##### a. Maaş Hesaplama



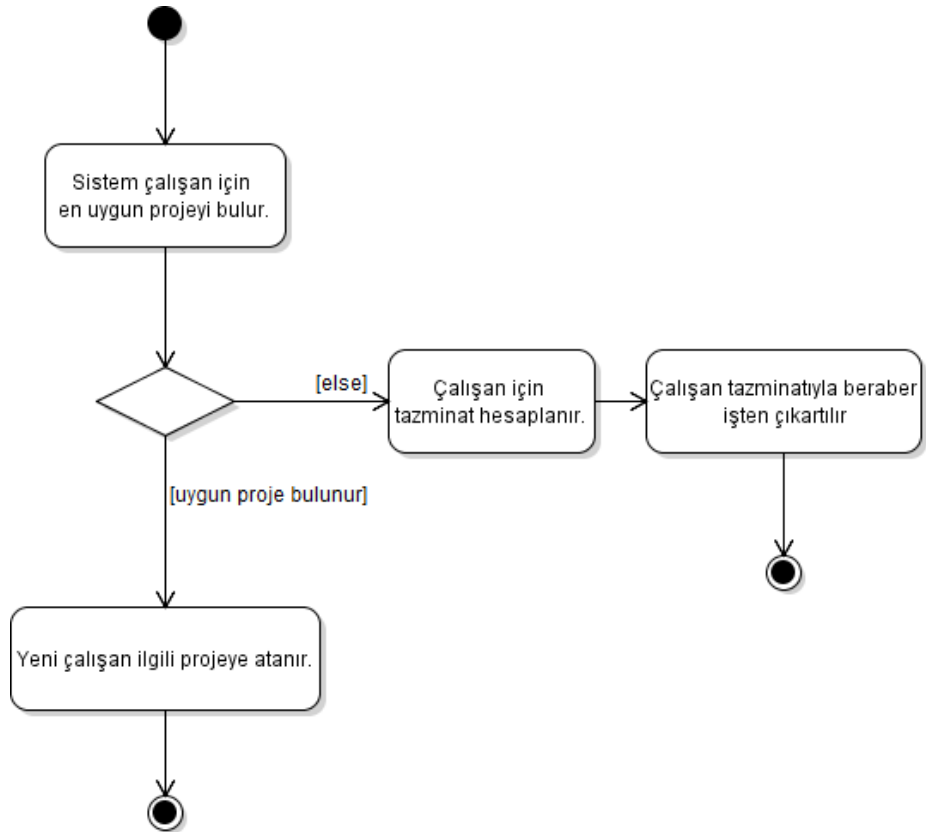
Şekil 7.4.1. - Maaş Hesaplama Etkinlik Diyagramı

##### b. Tazminat Hesaplama



Şekil 7.4.2. - Tazminat Hesaplama Etkinlik Diyagramı

### c. Çalışanı Projeye Bağlama



Şekil 7.4.3. - Çalışanı Projeye Bağlama Etkinlik Diyagramı

## 8. Birim Testi Sınamaları

Bu bölümde birim testlerinin kodları ve testlerin toplu başarısının ekran çıktısı bulunmaktadır. Testler kodlandıkları öğrencinin isim ve numarasıyla başlıklandırıldı.

### a. Emircan Kirez

Veri tabanından çalışanların ve projelerin çekilmesine dair 2 adet testin kaynak kodları:

```
@Test

void getAllWorkers() {

    List<Workers> allWorkers = workersRepository.findAll();

    assertEquals(0, allWorkers.size());

}
```

```
@Test

void getAllProjects() {

    List<Projects> allProjects = projectsRepository.findAll();

    assertEquals(0, allProjects.size());

}
```

### b. Buse Nur Pekmezci

Yeni çalışan eklenmesi ve çalışanı proje atamaya dair 2 adet testin kaynak kodları:

```
@Test

void addNewWorker() {

    Workers worker = new Workers();

    worker.setName("Test");    worker.setSurname("Test");

    worker.setSsn("12345678901");    worker.setTelNo("123456789");

    worker.setAddress("Test");    worker.setAccountingUrl("Test");

    worker.setPositionId(2);    worker.setRecruitmentDate(new Date());

    worker.setSalary(1000);

    List<Workers> oldAllWorkers = workersRepository.findAll();

    workersController.addWorker(worker);

    List<Workers> newAllWorkers = workersRepository.findAll();

}
```

```

        assertEquals(oldAllWorkers.size() + 1, newAllWorkers.size());
    }

    @Test
    void findProjectToNewWorker() {
        List<Workers> allWorkers = workersRepository.findAll();
        Workers lastWorker = allWorkers.get(allWorkers.size() - 1);
        workersController.findFreePosition(lastWorker.getId());
        Workers newLastWorker = workersRepository.findById(lastWorker.getId()).get();
        assertEquals(0, newLastWorker.getProjectId());
    }

```

c. Muhammed Eren Şekkeli

Yeni proje eklenmesi ve çalışan ünvanlarının çekilmesine dair 2 adet testin kaynak kodları:

```

    @Test
    void addNewProject() {
        Projects project = new Projects();
        project.setName("Test Project");
        project.setPlannedStartDate(new java.sql.Date(new Date().getTime()));
        project.setPlannedDeliveryDate(new java.sql.Date(new Date().getTime()));
        project.setActive(true);
        project.setValid(false);
        project.setManagerId(702);
        List<Projects> oldAllProjects = projectsRepository.findAll();
        projectsController.addProject(project);
        List<Projects> newAllProjects = projectsRepository.findAll();
        assertEquals(oldAllProjects.size() + 1, newAllProjects.size());
    }

```

```

    @Test
    void getAllPositions() {
        List<Positions> allPositions = positionsRepository.findAll();
    }

```

```
        assertEquals(0, allPositions.size());  
    }  
}
```

#### d. Sait Yalçın

Yeni ünvan ekleme ve bir projeye aktarılmayan çalışanların silindiğinin sağlanmasına dair testlerin kaynak kodları:

```
@Test  
void savePosition() {  
    List<Positions> allPositions = positionsRepository.findAll();  
    Positions positions = new Positions();  
    positions.setName("Test");  
    List<Positions> oldAllPositions = positionsRepository.findAll();  
    Positions positions1 = positionsRepository.save(positions);  
    List<Positions> newAllPositions = positionsRepository.findAll();  
    positionsRepository.deleteById(positions1.getId());  
    assertEquals(oldAllPositions.size() + 1, newAllPositions.size());  
}  
  
@Test  
void getFreeWorkers(){  
    List<WorkerDto> workersList = workersController.getWorkers();  
    for(WorkerDto workers: workersList){  
        assertEquals(0, workers.projectId);  
    }  
}
```

#### e. Recep Selim Ağırman

Her projenin bir yöneticisi olduğunu kontrol eden test ve proje pozisyonlarının bulunduğu tablonun satır sayısı ile tüm projelerdeki pozisyon toplamının eşit olup olmadığını kontrol eden testin kaynak kodları:

```
@Test
```

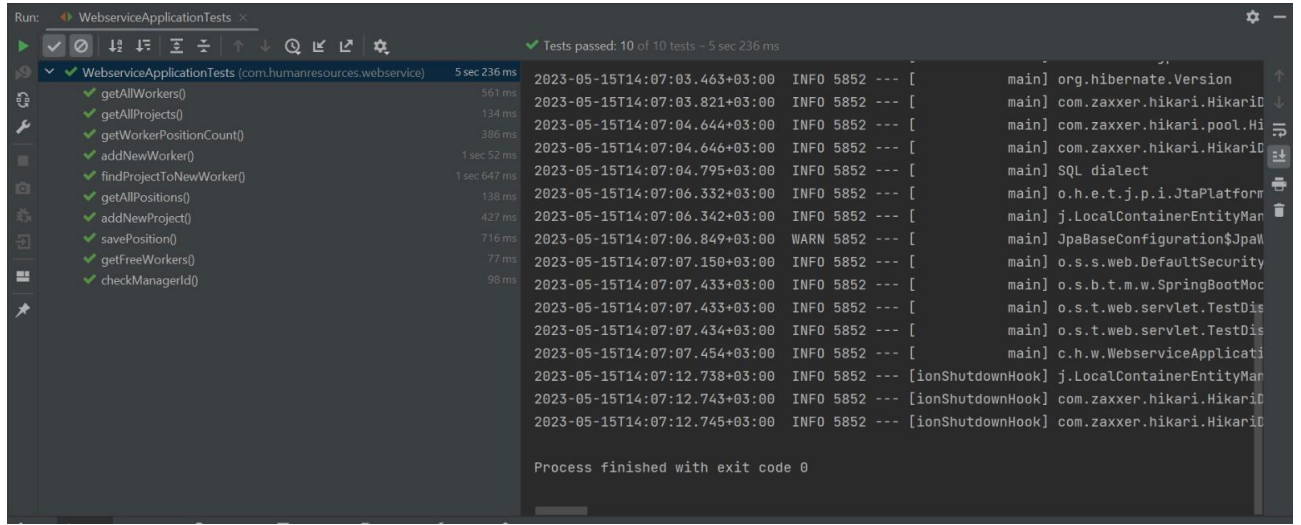
```

void checkManagerId(){
    List<Projects> projectsList = projectsController.getProjects();
    for(Projects projects: projectsList){
        assertEquals(0, projects.getManagerId());
    }
}

@Test
void getWorkerPositionCount(){
    Random random = new Random();
    Iterable<Positions> positionsList = positionsController.getAllPositions();
    List<Projects> projectsList = projectsController.getProjects();
    int randomNumber = random.nextInt(projectsList.size() - 0 + 1) + 0;
    Projects projects = projectsList.get(randomNumber);
    int positionsSize = 0;
    Iterator<Positions> iterator = positionsList.iterator();
    while (iterator.hasNext()) {
        iterator.next();
        positionsSize++;
    }
    int positionsOfProjectSize =
        projectPositionController.getProjectPositionByProjectId(projects.getId()).size();
    assertTrue(positionsSize >= positionsOfProjectSize);
}

```

f. Test Sonuçlarına Dair Ekran Görüntüsü



Şekil 8.f. - Test Sonuçlarının Ekran