**BOZOK ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

****

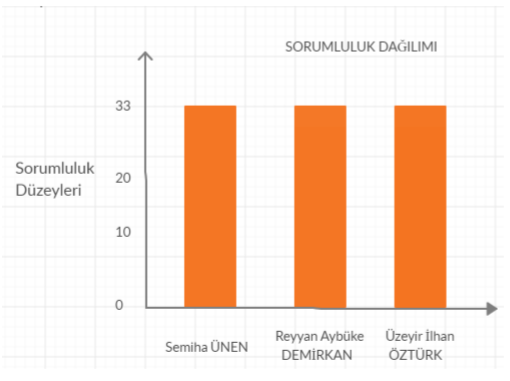
**ONLİNE OTOPARK SİSTEMİ**

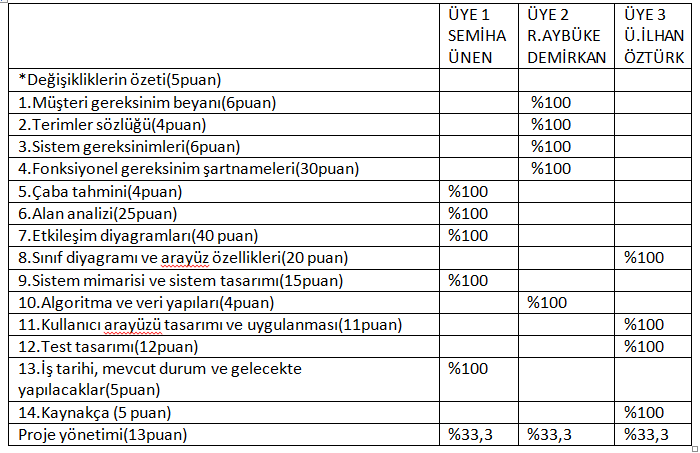
**SEMİHA ÜNEN**

**REYYAN AYBÜKE DEMİRKAN**

**ÜZEYİR İLHAN ÖZTÜRK**

**2019-2020 BAHAR DÖNEMİ**

****



İçindekiler

[1.Müşteri Gereksinim Beyanı 4](#_Toc40188157)

[2.Terimler Sözlüğü 4](#_Toc40188158)

[3.Sistem Gereksinimleri (Kullanıcı Öyküleri) 5](#_Toc40188159)

[4.İşlevsel Gereksinimler Spesifikasyonu 7](#_Toc40188160)

[5. Kullanıcı Çaba Tahmini 12](#_Toc40188161)

[6.Alan analizi: 15](#_Toc40188162)

[7.Madde: Etkileşim Diyagramları: 21](#_Toc40188163)

[Araç Girişi: 22](#_Toc40188164)

[Araç listeleme: 22](#_Toc40188165)

[Ücretlendirme Sistemi: 22](#_Toc40188166)

[8. Sınıf Diagramı ve Arayüz Spesifikasyonu 24](#_Toc40188167)

[Ana sınıf diagramı 23](#_Toc40188168)

[Araç Listeleme 24](#_Toc40188169)

[Ücret Bilgilendirme Sistemi 25](#_Toc40188170)

[b.Veri Türleri ve İşlem İşaretleri 26](#_Toc40188171)

[c.İzlenebilirlik Matrisi 28](#_Toc40188172)

[9. Sistem Mimarisi ve Sistem Tasarımı 30](#_Toc40188173)

[a.Mimari Tarzlar 30](#_Toc40188174)

[b. Alt Sistemlerin Tanımlanması 30](#_Toc40188175)

[Sunucu (Veri Tabanı): 31](#_Toc40188176)

[c. Alt Sistemleri Donanıma Eşleme 34](#_Toc40188177)

[d. Kalıcı Veri Depolama 34](#_Toc40188178)

[e. Ağ Protokolü 36](#_Toc40188179)

[f. Global Kontrol Akışı 37](#_Toc40188180)

[g. Donanım Gereksinimleri 37](#_Toc40188181)

[10. Algoritmalar ve Veri Yapıları 38](#_Toc40188182)

[11.Kullanıcı Arayüzü Tasarımı ve Uygulaması 39](#_Toc40188183)

[12 Test Tasarımı 45](#_Toc40188184)

[KAYNAKÇA 53](#_Toc40188185)

# 1.Müşteri Gereksinim Beyanı

Otoparka kendisinin kayıt olabilmesi ve personelin kayıt etmesi.

Bunlardan ilki ise kendisi online olarak sistem ara yüzünden kayıt olabilir.

İkincisi ise müşteri otoparka geldiği zaman personel tarafından sisteme kayıt edilebilir.

İstediği zaman sisteme giriş yapıp parkın doluluk boşluk oranını görüp rezervasyon yapabilir.

Sistemde kayıtlı olan kişisel bilgilerini güncelleyebilir.

Rezervasyon bilgilerini görebilir.[1]

# 2.Terimler Sözlüğü

Rezervasyon: Otel, lokanta, gazino, tiyatro gibi yerlerde, uçak, tren, otobüs gibi taşıtlarda yer ayırtma ya da ayırma işi.

Otopark: Motorlu taşıtların belli bir süre için bırakılabildiği yer.

Hizmet Kalitesi: Hizmet, gereksinimleri karşılama ve üretildiği anda tüketilme özelliklerine sahip her türlü etkinlik olarak tanımlanabilir. Hizmet kalitesi ise bir işletme kavramıdır. Hizmet kalitesi, hizmetin istenen amaca ulaşma derecesidir.

Ücret: İşgücünün kullanımı karşılığında ödenen bedeldir. Kurum ve organizasyonlar tarafından çalışanlara ödenen maaşlar, primler ve komisyonlar ücret kapsamına girerler.

Personel: Bir kuruluşun hizmetin görevlileri, bir işyerinde çalışanların tümü.

Sistem: Birbiriyle etkileşen veya ilişkili olan, bir bütün oluşturan cisim veya varlıkların bileşkesidir. Bu varlıklar soyut veya somut olabilirler.[2]

# 3.Sistem Gereksinimleri (Kullanıcı Öyküleri)

**Fiziksel Özellikler:** Otoparkların somut özellikleri, kullanılan malzemeler, araç gereçler ve personelin dış görünüşü gibi fiziksel unsurlardır.

Modern ve teknolojik donanıma sahiptir.

Binaları ve çalışma alanları her zaman temizdir.

Çalışanları temiz ve şık görünüşlüdür.

Tabelalar ile yönlendirmeler yapılmıştır.

**Güvenilirlik:** Otoparklarda verilen hizmetlerdeki kusursuzluktur. Hizmetin her seferinde aynı standartlarda, dikkatli, güvenilir ve zamanında yapılmasıdır. Güvenilirlik aynı zamanda sorunların çözümü ve müşterilere verilen fiyat garantisini de kapsamaktadır.

Müşteri problemlerini zamanında çözerler.

Müşterilerin sorunlarını çözmede etkilidirler.

Hizmet karşılığı alınan ücretler uygundur.

Hizmetlerin tam olarak ne zaman verileceği müşterilere söylenir.

Müşterilerin kayıtları hatasız tutulur.

**Heveslilik:** Otopark çalışanlarının hızlı hizmet sunmaları, müşteriye gösterilen ilgi, hizmetin doğru ve hızlı verilmesi gibi unsurlardır.

Çalışanlar fiziki ekipmanları kullanacak bilgiye sahiptirler.

Hizmetleri mümkün olan en kısa sürede gerçekleştirirler.

Müşterilere yardım etmeye her zaman isteklidirler.

Müşterilerin şikâyetlerini her zaman dikkate alırlar.

**Güvence:** Otoparklarda çalışanların bilgisi, anlayışı, müşterilere karşı saygılı olmaları, müşterilerin kendilerini otoparkta güvende hissetmeleri gibi unsurlardır.

Çalışanların davranışları müşterilerde güven duygusu uyandırır.

Müşteriler otoparkta kendilerini güvende hissederler.

Çalışanlar müşterilere her zaman nazik davranırlar.

Çalışanlar müşterilerin sorularına cevap verecek bilgiye sahiptirler.

**Empati:** Otopark çalışanlarının müşterilere saygılı davranmaları, müşteri çıkarlarını gözetmeleri ve ulaşılabilirlik gibi unsurlardır.

Müşterilere her zaman değer verirler.

Çalışma saatleri müşterilere uygun olacak şekildedir.

Müşterilere karşı saygılıdırlar.

Müşterilerin çıkarlarını her şeyin üstünde tutarlar.

Çalışanlar yaptıkları iş ile ilgili yeterli eğitime sahiptirler.

# 

# 4.İşlevsel Gereksinimler Spesifikasyonu

**A. Paydaşlar**

Müşteriler, Personel, Patronlar

**B. Aktörler ve Hedefler**

**Müşteri:**

İki yolla sisteme kayıt olabilir:

Bunlardan ilki personel tarafından sisteme kayıt edilebilir.

İkincisi ise kendisi online olarak sistem ara yüzünden kayıt olabilir.

İstediği zaman sisteme giriş yapıp parkın doluluk boşluk oranını görüp rezervasyon yapabilir.

Sistemde kayıtlı olan kişisel bilgilerini güncelleyebilir.

Rezervasyon bilgilerini görebilir.

**Personel:**

Personel giriş kısmından sisteme giriş yapar.

Personel Sisteme kayıtlı olmayan müşterileri sisteme kayıt edebilir.

Personel park bilgisi ekranından parkın boş olup olmadığını görebilir ayrıca gelen araçların girişlerini sisteme kaydeder.

Personel parktan çıkış yapmak isteyen araçların ücret tutarını sistemden bakıp müşteriden alır.

Personel mesaisi bittiğinde sistemden çıkış yapar.

**Otopark Sahibi (Patron):**

Sistem tarafından kaydedilen ücretlendirmeleri, otoparka gelen araç sayısını, araçların ödediği tutarları ve araçların plakalarını kaydedilen veri tabanında görebilir.

Otopark fiyatlandırmasını istediği zaman artırıp azaltabilir.[3]

**C. Kullanım Durumları**

1. **Günlük Tanım**

**1: Kayıt**

REQ-22

Sistem, müşterilerin kendi bilgilerini görüntülemesine ve düzenlemesine izin verecektir.

REQ-26

Sistem, kullanıcının web sitesinde bir hesap oluşturmasına izin verecektir.

REQ-27

Sistem, kullanıcının bilgilerini veritabanına kaydetmelidir.

REQ-22, REQ-26, REQ-27 kullanılarak sistem müşterinin kayıt yapmasını sağlar ilgili formu çevrimiçi bir forma gönderir bir hesap için.

**2: Rezervasyon**

REQ-1

Sistem, bir kullanıcı hesabının rezervasyon yapmasına izin vermelidir.

REQ-2

Sistem, rezervasyon başarısı veya başarısızlığı için metin veya e-posta onayı gönderecektir.

REQ-8

Sistem, hangi alanların dolu olup olmadığını haritada gösterecektir

REQ-1, REQ-2 ve REQ-8 kullanan sistem, müşterinin çevrimiçi rezervasyon sayfamıza bir zaman belirterek bir park yeri ayırmasına olanak tanır.

**3: Garaja giriş**

REQ-3

Sistem, giriş kapısına yaklaştığında personel arabanın bilgilerini alacaktır.

REQ-6

Sistem, müşterinin garaja girdiği zamanı personel kayıt eder araç limiti olmadan istediği süre kadar içerde kalabilir.

REQ-7

Sistem, Personel sürücüyü gitmesi gerektiği noktaya yönlendirecektir

REQ-8

Sistem, hangi alanların dolu olup olmadığını haritada gösterecektir.

REQ-3, REQ-6, REQ-7 ve REQ-8 kullanarak bir sürücü garaja arabası ile girebilir. Sistem bilgilerini gösterecek ve hangi personel boş yeri seçerek aracın o noktaya gitmesi gerektiğini söyleyecektir.

**4: Durum Kontrolü**

REQ-6

Sistem, müşterinin garaja girdiği zamanı personel kayıt eder araç limiti olmadan istediği süre kadar içerde kalabilir.

REQ-9

Sistem, müşterilere işgal ettikleri yeri gösterecektir.

REQ-12

Sistem, müşterinin garajdan ayrıldığı zamanı gösterecektir.

Sistem, REQ-6, REQ-9 ve REQ-12 ’yi kullanarak, bir müşteriye otoparkın durumunu, araçlarının konumunu ve ilgili diğer bilgileri gösterir.

**5: Ayrılmış Park Yeri**

REQ-3

Sistem, giriş kapısına yaklaştığında personel arabanın bilgilerini alacaktır.

REQ-6

Sistem, müşterinin garaja girdiği zamanı personel kayıt eder araç limiti olmadan istediği süre kadar içerde kalabilir.

REQ-7

Sistem, Personel sürücüyü gitmesi gerektiği noktaya yönlendirecektir

REQ-8

Sistem, hangi alanların dolu olup olmadığını haritada gösterecektir.

REQ-3, REQ-6, REQ-7 ve REQ-8 kullanılarak sistem, giriş yapan müşterilerin rezervasyon yapıp yapmadığını söyleyerek personel aracın giriş şeklini belirler. Daha sonra onları uygun şekilde otoparkta yönlendirir.

**6: Park yerine giriş**

REQ-8

Sistem, hangi alanların dolu olup olmadığını haritada gösterecektir.

REQ-11

Sistem, bir araba bir noktayı terk ettiğinde veya doldurduğunda haritayı güncelleyecektir

REQ-8 ve REQ-11’i kullanarak sistem, müşteriler belirlenmiş park yeri noktalarına gidip oraya park ediyorlar. Personel, araçların park edileceği yeri sistemden seçerek harita üzerinden o yeri dolu gösterilir.

**7: Garajdan ayrılma**

REQ-11

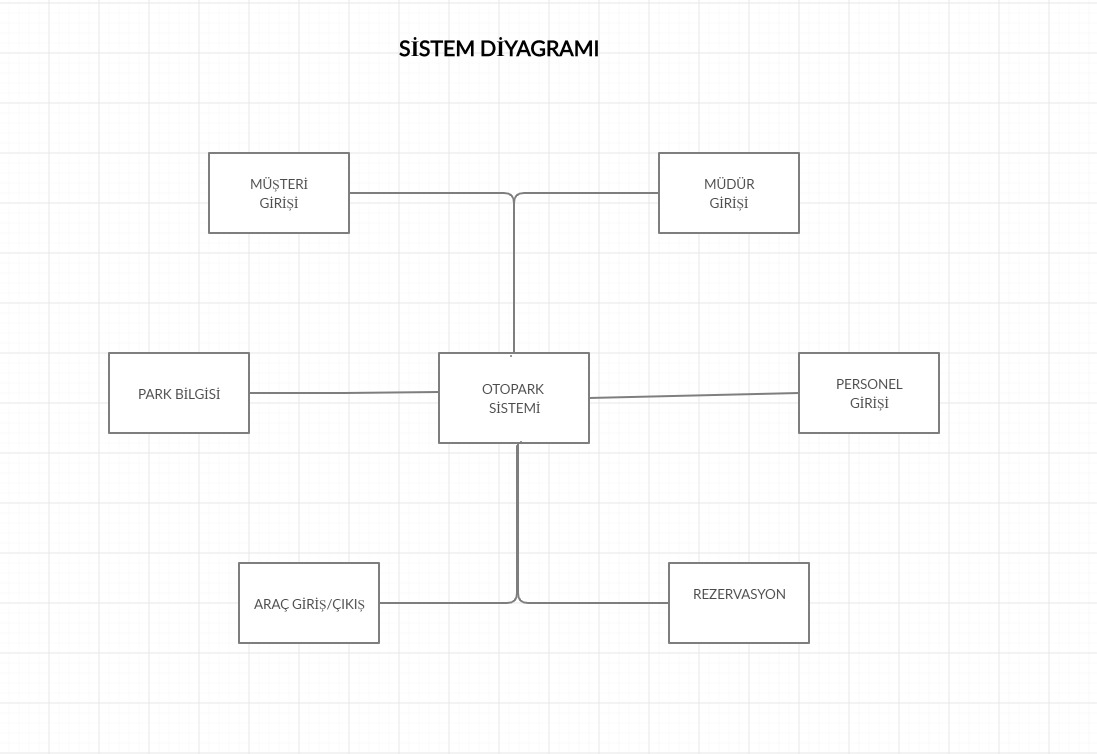
Sistem, bir araba bir noktayı terk ettiğinde veya doldurduğunda haritayı güncelleyecektir.

REQ-12

Sistem, müşterinin garajdan ayrıldığı zamanı gösterecektir.

REQ-11 ve REQ-12’yi kullanarak sistem, müşteriler park etmeyi bitirir ve garajı terk ederler.[4]

1. Sistem Dizisi Diyagramı



1. İzlenebilirlik Matrisi

Müşteri, personel ve otopark sahibinin giriş yapması-Otopark ile ilgili bilgileri görebilmeleri.

Şifremi unuttum-Herhangi bir üye şifresini unuttuğu zaman kolayca yeni bir şifre alarak kolayca sisteme girebilir.

Rezervasyon-Müşteri otoparkın doluluk boşluk oranını görüp buna göre istediği yere rezervasyon yapabilir. Böylece otoparka geldiği zaman, fazla zaman kaybetmeyerek yerine geçer.

Araç giriş çıkış-Personel araç giriş çıkışlarını görüp yeni gelen araç için park alanının boş yerini gösterecek.

Ücret hesaplama-Araç giriş çıkış saati hesaplanarak bunun karşılığında müşteri ne kadar ödeyeceğini öğrenir Personel yardımıyla. Otopark sahibi de ödenecek tutarı görebilecek.[5]

# 

# 5. Kullanıcı Çaba Tahmini

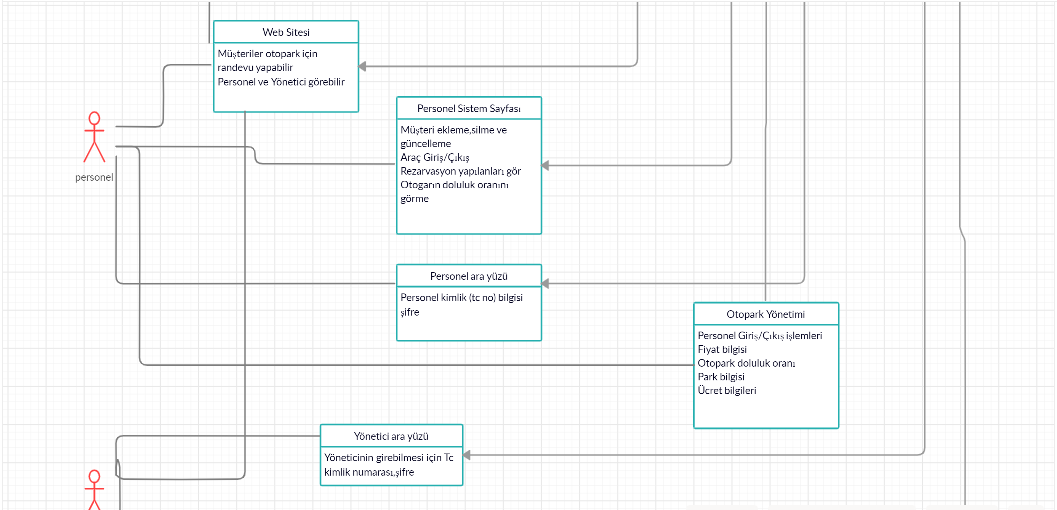
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | İlgili özelliklerin tanımı | Karmaşıklık | Ağırlık |
| Use Case1:Yönetici ekleme | Ekran arayüzü. Ana başarı senaryosu için 7adıım.1 katılımcı | Basit | 5 |
| Use Case2:Yönetici giriş | Ekran arayüzü. Ana başarı senaryosu için 8adım.1 katılımcı | Basit | 5 |
| Use Case3:Yönetici çıkış | Ekran arayüzü. Ana başarı senaryosu için 3adım.1 katılımcı | Basit | 5 |
| Use Case4: park yeri bilgisi | Ekran arayüzü.Ana başarı senaryosu için 3adım.2 katılımcı | Basit | 5 |
| Use Case5: araç girişi | Ekran arayüzü.Ana başarı senaryosu için 3adım.3 katılımcı | Basit | 5 |
| Use Case6: uygun park yeri | Ekran arayüzü.Ana başarı senaryosu için 4adım.3 katılımcı | Karmaşık | 10 |
| Use Case7: :araç çıkış ve ücretlendirme | Ekran arayüzü.Ana başarı senaryosu için 4adıım.3 katılımcı | Karmaşık | 10 |
| Use Case8: araçları listele | Ekran arayüzü.Ana başarı senaryosu için 2adım.2 katılımcı | Kompleks | 10 |
| Use Case9:yönetici güncelleme | Ekran arayüzü.Ana başarı senaryosu için 4adım.2 katılımcı | Karmaşık | 10 |
| Use Case10:yönetici şifre sıfırlama | Ekran arayüzü.Ana başarı senaryosu için 5adım.1 katılımcı | Basit | 5 |

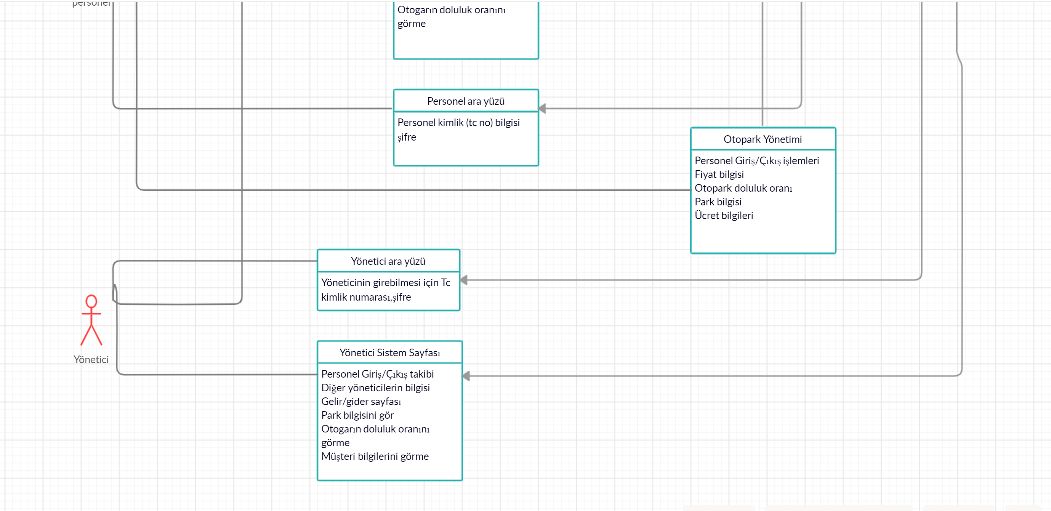
Teknik karmaşıklık faktörleri:

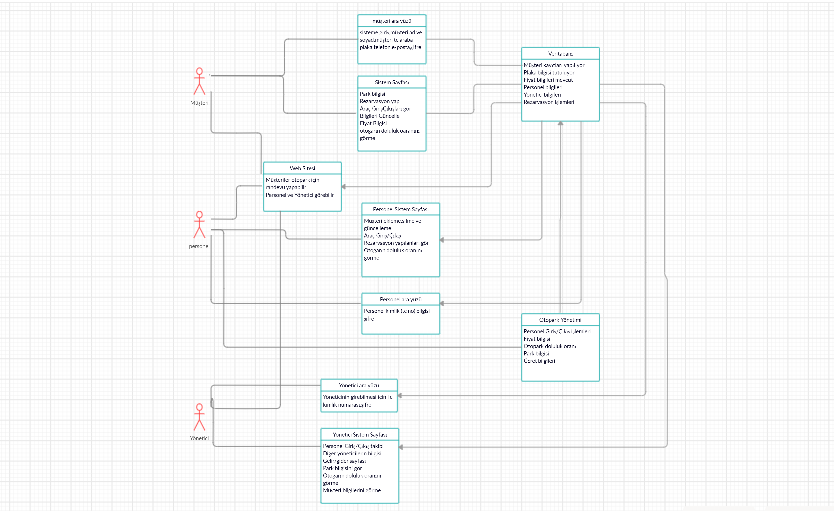
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Teknik faktörler | Açıklama | Ağırlık | Algılanan karmaşıklık | | Faktörler (A\*AK) |
| T1 | Dağıtılmış, Web tabanlı sistem | 2 | 3 | | 6 |
| T2 | Kullanıcı iyi performans bekler | 1 | 3 | | 3 |
| T3 | Son kullanıcı verimlilik bekliyor | 1 | 3 | | 3 |
| T4 | Dahili işleme basittir | 1 | 3 | | 3 |
| T5 | Yeniden kullanılabilirlik gerektirmez | 1 | 0 | | 0 |
| T6 | Kurullum kolaylığı basittir | 0.5 | 3 | | 1.5 |
| T7 | Kullanım kolaylığı çok önemlidir | 0.5 | 5 | | 2.5 |
| T8 | Taşınabilirlik gereksinimi yoktur sistem web tabanlıdır. | 2 | 0 | | 0 |
| T9 | Sisteme ödeme yöntemleri ekleme | 3 | 2 | | 6 |
| T10 | Eğitim ihtiyacı yoktur | 1 | 0 | | 0 |
| T11 | Üçüncü taraflar dahil | 1 | 3 | | 3 |
| T12 | Güvenlik önemlidir | 1 | 5 | | 5 |
| T13 | Eşzamanlı kullanım gerektiriyor. | 1 | 4 | | 4 |
|  | Toplam teknik faktörler: | | | 37 | |

[7]

# ym-16.Alan analizi:







[8]

İzlenebilirlik matrisi

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Gereklilik | Öncelik | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 |
| SG4 |  |  |  |  |  | ✔ |  |  |  |
| SG5 |  |  |  |  |  |  | ✔ |  |  |
| SG6 |  |  |  |  |  |  | ✔ |  |  |
| SG7 |  |  |  |  | ✔ | ✔ | ✔ |  |  |
| SG8 |  |  |  |  | ✔ | ✔ | ✔ | ✔ | ✔ |
| SG9 |  |  |  | ✔ | ✔ | ✔ | ✔ |  |  |
| SG10 |  |  |  |  | ✔ |  |  |  |  |
| SG11 |  |  |  |  | ✔ | ✔ |  |  |  |
| SG12 |  |  |  |  |  |  | ✔ |  |  |
| SG13 |  |  |  |  |  | ✔ |  |  |  |
| SG14 |  |  |  |  |  | ✔ |  |  |  |
| SG15 |  |  |  | ✔ | ✔ | ✔ |  |  |  |
| SG16 |  |  |  |  | ✔ |  |  |  |  |
| SG17 |  | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| SG18 |  | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| SG19 |  | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| SG20 |  | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| SG21 |  |  |  |  | ✔ | ✔ | ✔ |  |  |

Müşteri, personel ve otopark sahibinin giriş yapması-Otopark ile ilgili bilgileri görebilmeleri.

Rezervasyon-Müşteri otoparkın doluluk boşluk oranını görüp buna göre istediği yere rezervasyon yapabilir. Böylece otoparka geldiği zaman, fazla zaman kaybetmeyerek yerine geçer.

Araç giriş çıkış-Personel araç giriş çıkışlarını görüp yeni gelen araç için park alanının boş yerini gösterecek.

Ücret hesaplama-Araç giriş çıkış saati hesaplanarak bunun karşılığında müşteri ne kadar ödeyeceğini öğrenir Personel yardımıyla. Otopark sahibi de ödenecek tutarı görebilecek.

Sistem Gereksinimleri:

SG 4: Personel park bilgisi ekranından parkın boş olup olmadığını görebilir ayrıca gelen araçların girişlerini sisteme kaydeder.

SG -5: Personel parktan çıkış yapmak isteyen araçların ücret tutarını sistemden bakıp müşteriden alır.

SG -6: Personel mesaisi bittiğinde sistemden çıkış yapar çıkış yaptığı saati ve tarihi sistem tarafından kaydedilir.

SG -7: Sistem tarafından kaydedilen ücretlendirmeleri, personelin giriş çıkış kaç saat çalıştığını.

SG -8: Otoparka gelen araç sayısını, araçların ödediği tutarları ve araçların plakalarını kaydedilen veri tabanında görebilir.

SG -9:Müşteri sisteme kendi bilgileri ile giriş yapıp kaydolabilir.

SG -10:Müşteri kayıt işlemlerini personel kendisi yapar yapar

SG -11: Sistem, bir kullanıcı hesabının rezervasyon yapmasına veya rezervasyonu iptal etmesine izin vermelidir.

SG -12: Sistem giriş müşterileri için ödeme yöntemleri sağlayacaktır

SG-13 Sistem, hangi alanların dolu olup olmadığını haritada gösterecektir

SG-14 Sistem müşterilere işgal ettikleri yeri gösterecektir.

SG-15 Sistem, bir araba bir noktayı terk ettiğinde veya doldurduğunda haritayı güncelleyecektir

SG-16 Sistem, müşterilerin kendi bilgilerini görüntülemesine ve düzenlemesine izin verecektir.

SG -17Sistem, kullanıcının bilgilerini veritabanına kaydetmelidir.

SG-18 Sistemin internet erişimi olmalıdır

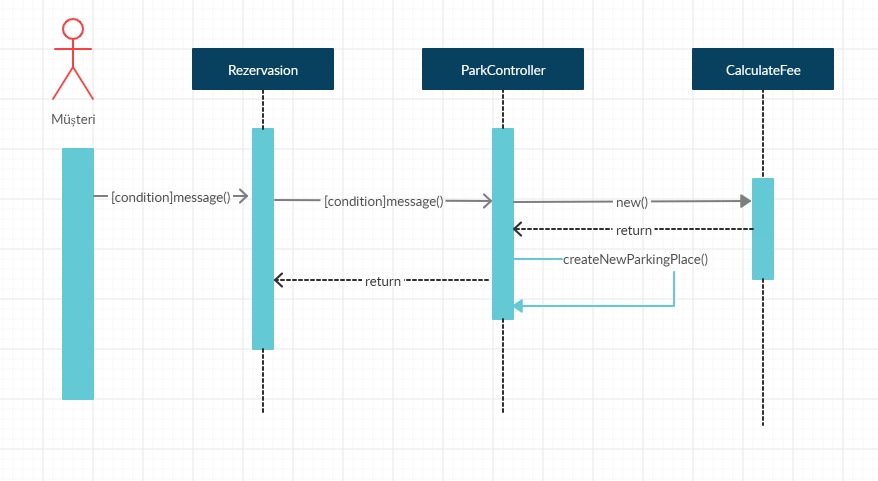
SG-19 Sistem, yöneticinin davranışını yapılandırmasına izin vermelidir

SG -20Sistem otomatik olarak çevrimiçi işlem yapabilmelidir

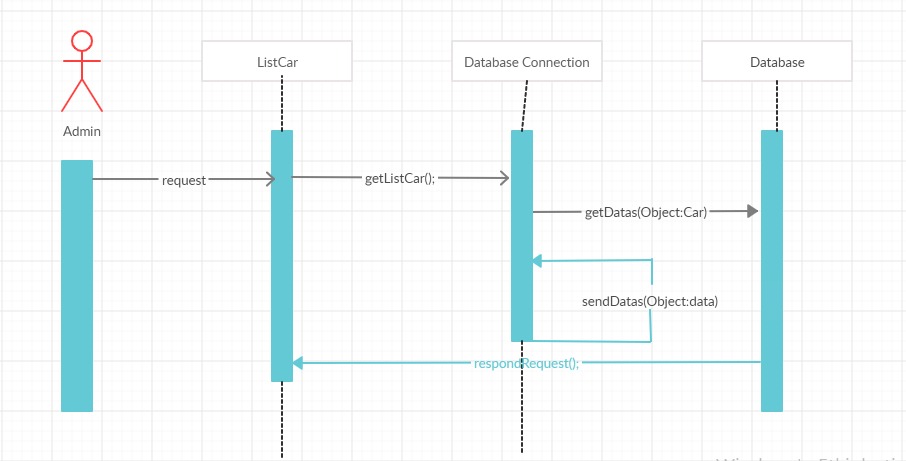
SG-21 Ana sayfada, kayıtlı olmayan kullanıcıların kaydol düğmesine tıklayarak ve kullanıcı adı ile şifre girerek kaydolmalarına izin verin.[9]

# 7.Madde: Etkileşim Diyagramları:

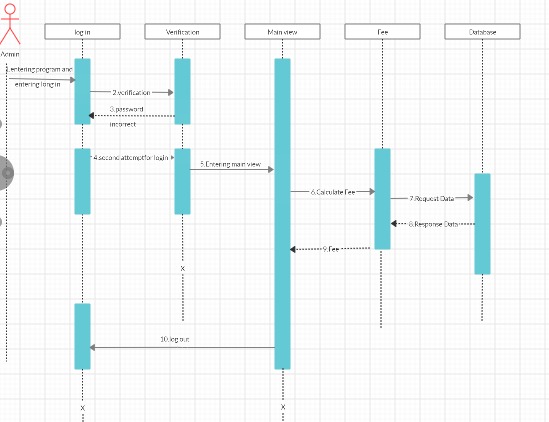
### Araç Girişi:



### Araç listeleme:

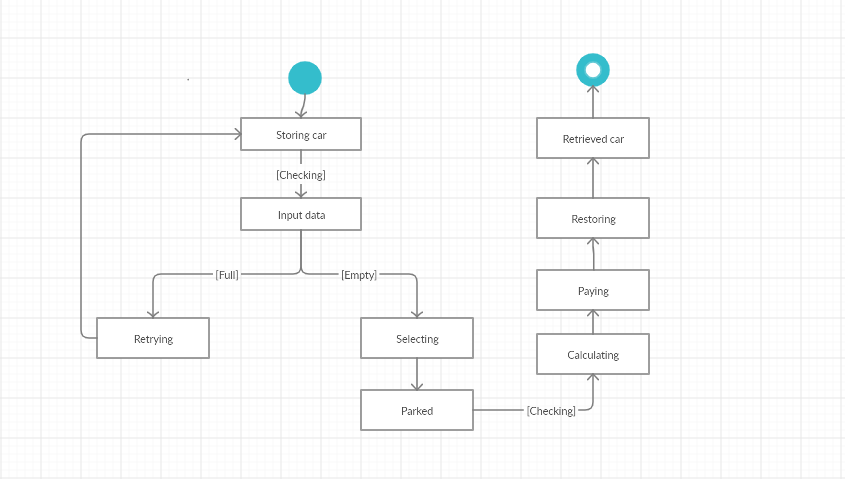


### Ücretlendirme Sistemi:



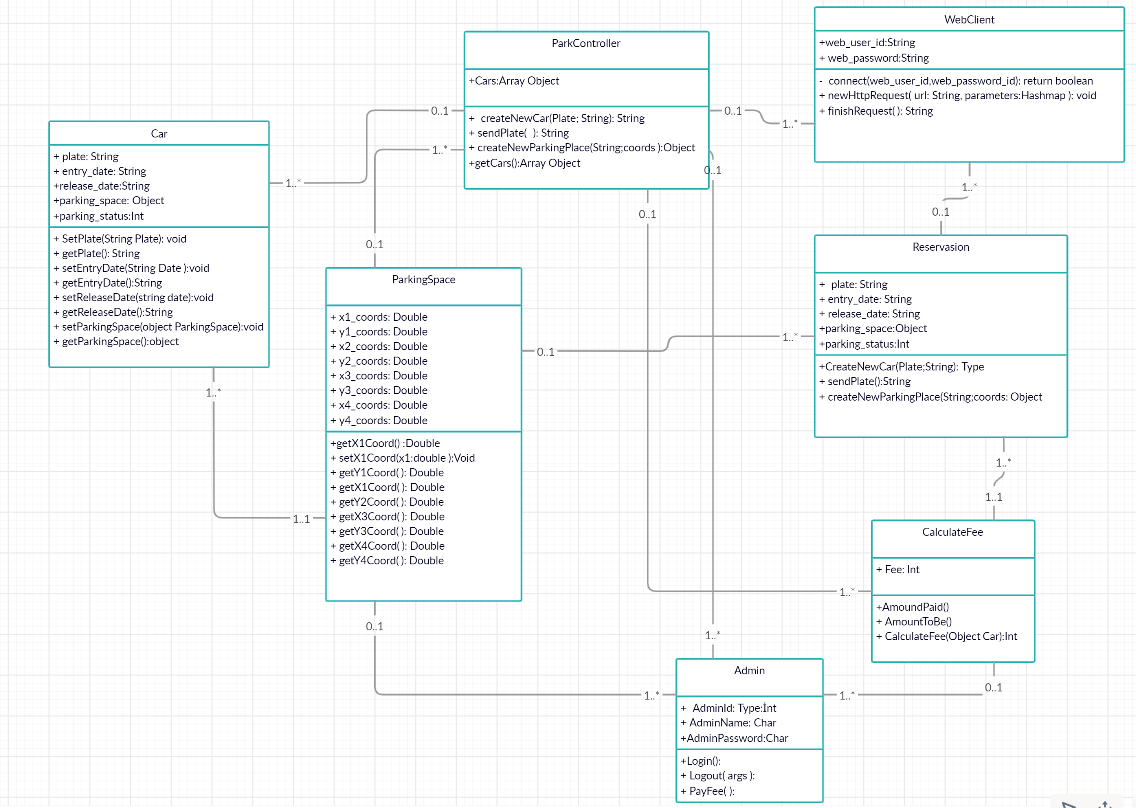
**[9]**

**Durum Dinamiği Bakış Açısı:**

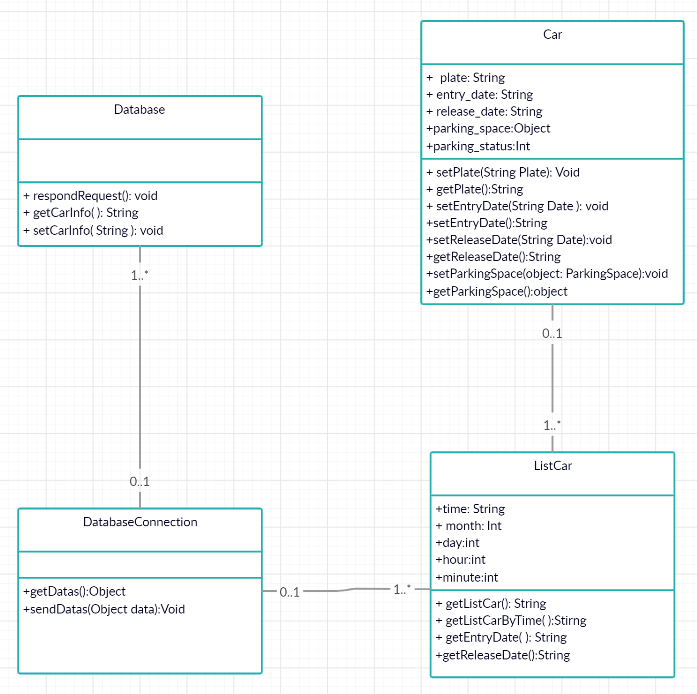
****

# 8. Sınıf Diagramı ve Arayüz Spesifikasyonu

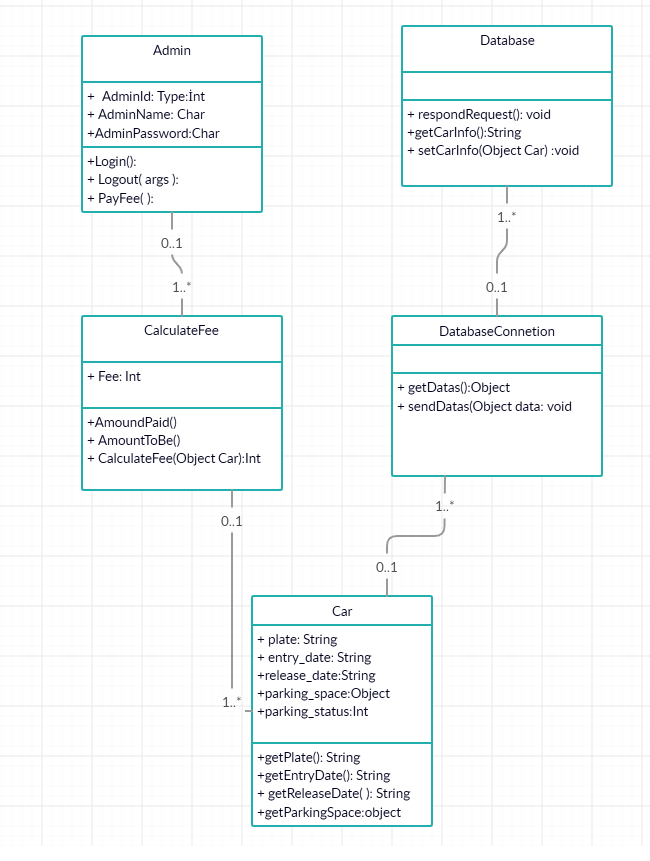
### Ana sınıf diagramı

****

### Araç Listeleme

****

### Ücret Bilgilendirme Sistemi



[10]

## b.Veri Türleri ve İşlem İşaretleri

**ParkingSpace:**

Car ve ParkController sınıfları ile ilişkilidir. (x1\_coord:Double),(y1\_coord:Double)(x2\_coord:Double),(y2\_coord:Double),(x3\_coord:Dou ble),(y3\_coord:Double),(x4\_coord:Double)(y4\_coord:Double) tipinde değişkenlere sahiptir. Park alanı oluşturmada x ve y leri tanımlamada kullanılır. (getX1Coord():Double),(setX1Coord(x1:double):Void),(getY1Coord():Double),(getX2Coord():Double),(getY2Coord():Double),(getX3Coord():Double),(getY3Coord():Double),(getX4Coor d():Double),(getY4Coord():Double) metotlarına sahiptir. Park alanı oluşturmada x ve y leri tanımlandıktan sonra belirlenen değerler için park alanı oluşturur.[2]

**Park Controller:**

Diyagramdan anlaşılacağı üzere en çok işlem gören sınıftır. Park işleminin ana hatları bu sınıf diyagramı altında işlenir. Admin,WebClient, ParkingSpace ve CarPlate sınıfları ile ilişkilidir. (Cars: Array Object) dizi değişkenine sahiptir. Araç objeleri bu dizide tutulur. (createNewCar(Plate:String): String) metotu yeni aracı, plaka bilgisiyle döndürür. (sendPlate:String) metotu yeni plaka girişini servera gönderir. (createNewParkingPlace(String:coords):Object) metotu girilen koordinatlara göre yeni park alanı belirler (getCars(): Array Object) metotu arabaların bilgilerinin dizi tipinde döndürüldüğü metottur.

**Car**

ParkingSpace ve ParkController sınıfı ile ilişki halindedir. (plate:String) aracın plaka bilgisini String tipinde tutar. (entry\_date:String) araç giriş tarihini String tipinde tutar. (release\_date:String) araç çıkış tarihini String tipinde tutar. (parking\_space:Object) aracın hangi alanda park halinde olduğunu ParkingSpace nesnesi olarak saklayabilir. (parking\_status: Int) aracın park durumunu belirtir.

(setPlate(String Plate):void) metotu aracın plakasına değer atar.

(getPlate():String) metotu aracın plaka bilgisini döndüren metottur.

(setEntryDate(String Date): void) ----

(setEntryDate(): void) entry\_date özelliğine aracın giriş yaptığı tarihi atar. (setReleaseDate(String Date): void) release\_date özelliğine aracın çıkış yaptığı tarihi atar. (getReleaseDate(): string) release\_date özelliğini döndürür.

(setParkingSpace (object:ParkingSpace): void) ---- parking\_space özelliğine parametre olarak gönderilen ParkingSpace nesnesini gönderir.

(getParkingSpace(): object) ---- parking\_space özelliğini döndürür.

**Reservasion**

(plate): Plaka alınır.

(entry\_date):Aracın geleceği tarih girilir.

(release\_date):Aracın çıkacağı tarih bilgilerini alır.

(parking\_space): Otoparkta boş bir yerleri gösterir.

(parking\_status): Otoparkın durumunu gösterir.

sendPlate():Rezervasyon yapmak istediği yeri gönder.

CateNewParkingPlace: Rezervasyon yapmak istediği yeri seç.

**WebClient**

String id: Bir müşteri tarafından web sitesine giriş yapmak için kullanılır

String Password: Müşterinin hesap şifresi veritabanındaki hesap numarasıyla bağlantılı

String emailAdress: Müşterinin eposta adresi

Connect(web\_user\_id,web\_password) : Giriş yapmak için kullanıcı adı ve şifreni gir.

newHttprequest: Sorguyu gönder.

Finishrequest: sorguyu doğrulat. Ve gir.

**Admin**

String email: Kayıt için müştrerinin epostası

String name:Müşteri adı.

String lastname: Müşteri soyadı

String Phone\_number: Müşteri Cep numarası

String password: Müşteri giriş şifresi

Int TCNO: Müşterinin tc kimlik numarası

Login(): Admin giriş yapar.

Logout():Admin çıkış yapar.

PayFee():Ücreti alır.

**CalculateFee**

Fee: Ücret girilir.

Amoundpaid(); Ücreti çeker.

AmoundtoBe():Ücreti hesaplatır.

## c.İzlenebilirlik Matrisi

**Anasafya:**

Giriş yapan yönetici bu sayfada karşılanır.

Yönetici,otoparkta park halinde olan araç sayısına erişebilir.

Yönetici, otoparkta giriş yaptığı güne kadar kaç arabanın otoparka giriş yaptığını görebilir.

Yönetici işlem yapmak istediği kısımlara menülerden ulaşır.

Otoparkın günlük, haftalık, aylık istatistiklerine ulaşabilir.

Plaka aranmak istenirse, araba listesinden arama yapabilir.

Yönetici çıkış yapmak isterse “çıkış yap” butonuna basar.

**Araç Girişi Yap:**

Yönetici, araç girişi için “Plaka” , “Ad”, “TC” bilgilerini alır ona verilen konuma gider.

Yönetici “Park Yeri” bilgisini tanımlanan yerden seçer.

“Kaydet” butonuna basılır ve araç girişi bilgileri ile birlikte veri tabanına kaydedilir.

**İçerideki Araçları Listele**

Yönetici, otoparkta hali hazırda park durumunda olan araçları listeler.

“İçerde geçirilen süre”, “giriş tarihi” ve “park alanı” bilgisine “plaka” bilgisi ile ulaşabilir[11]

# 9. Sistem Mimarisi ve Sistem Tasarımı

## a.Mimari Tarzlar

Projemiz Web Sunucusu, Otopark Veri tabanı ve Masaüstü Ekranı'ndan oluşan 3 ana modülden oluşmaktadır. Otoparkta bulunan masaüstü uygulaması, web paneli ve web servisler mvc yapısı ile yazılacaktır ve MSSQL server veri tabanı kullanılacaktır.

**Katmanlı (n-katmanlı) mimari**

Popüler web çerçevelerinin çoğunun sunduğu standart yazılım geliştirme yaklaşımı olan Model-View-Controller (MVC) yapısı açıkça katmanlı bir mimaridir.

 Katmanlı mimaride ise uygulamalar ayrı katmanlar halinde yazıldığından, yönetilebilmesi ve  değiştirilmesi oldukça kolay ve hızlıdır. Bu yapıda eğer programın çok uzun kodları olacaksa alt parçacıklara (fonksiyonlara) bölünür. Böylece hangi fonksiyonun ne iş yaptığı  belli olur ve kodlarda bir hata varsa ilgili fonksiyon değiştirilir, yazılımın tüm kodlarını değiştirmek gerekmez. Böylece kodların yönetimi kolaylaşır. MVC ile popüleritesini arttıran bir mimari olan katmanlı mimari yine en çok kullanılan yazılım mimarileri dizisinin başında gelmektedir.

Bizlerde projemizde katmanlı mimariyi kullanmayı tercih ettik.

Otopark yazılımı, bir otoparka giriş yapan tüm araçların kaydedilebileceği, otomatik park yerleşiminin yaptırılabileceği bir yazılım olarak geliştirilecektir. Yazılımın veri tabanı web üzerinde saklanılacaktır. Yazılım ise web yazılımları sayesinde gerçekleştirilecektir.

Otopark sistemi, otoparklarda ki yer arama problemini çözmek amacıyla kullanılacaktır. Uygulama sayesinde kullanıcı otopark alanının girişinden park etme sürecine kadar çeşitli bileşenler yardımıyla kolayca, aracını en uygun park alanına park edebilecektir. Bu sayede park arama süresini minimuma düşürerek zamandan tasarrufta sağlanmaktadır.

Sistem otoparklarda boş park alanı bulma işleminin hızlı olması için üretilmiştir.

1. Araç otoparkımıza geldiğinde web üzerinden aldığı randevu ile yada girişte personel tarafından kayıt olur.

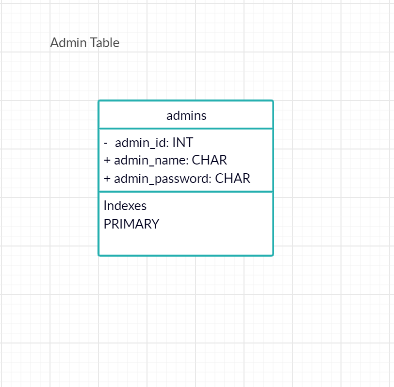
1. Web Sunucusu tarafından Otopark Veri tabanı ile bağlantı kurulur.

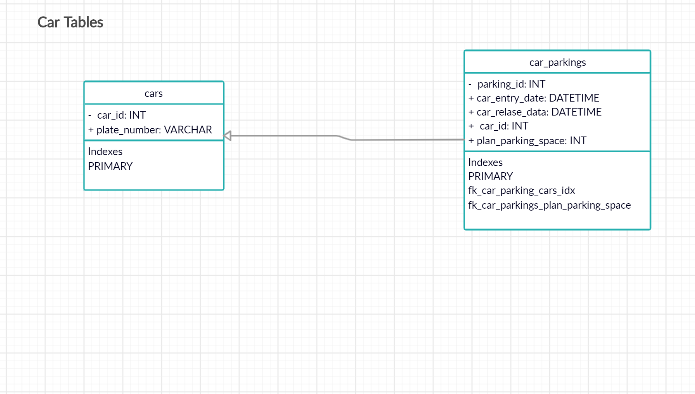
1. En uygun park yeri seçilir ve plaka o yer bilgisine işlenir.[12]

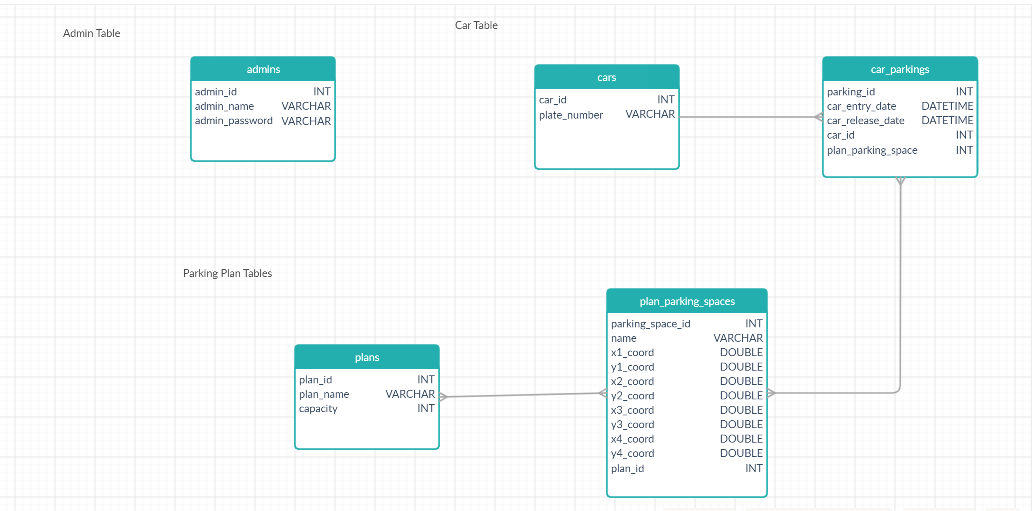
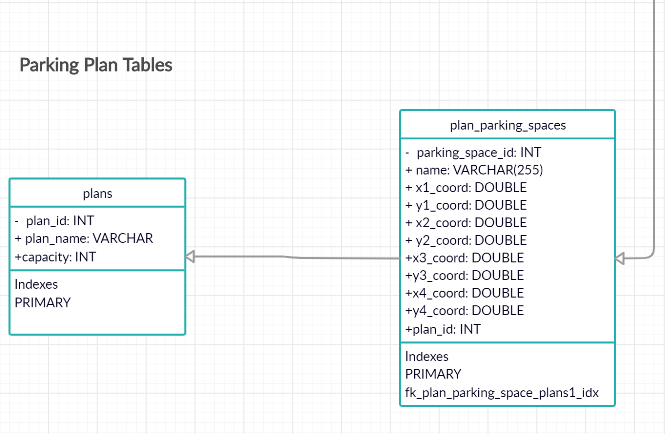
## b. Alt Sistemlerin Tanımlanması

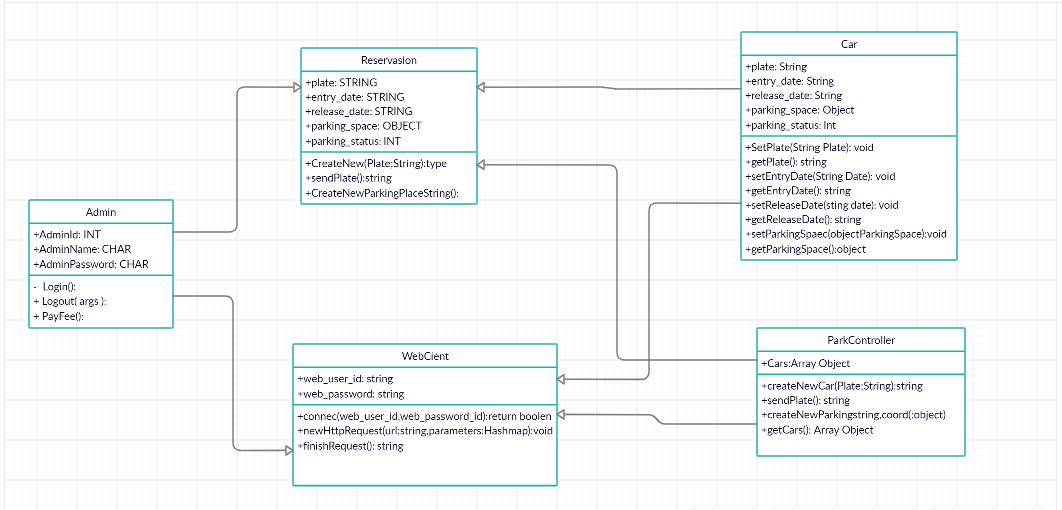
Yapımızda iki ana alt sistem olacak: Sunucu (veri tabanı); web sitesi (Kullanıcı arayüzü).

### Sunucu (Veri Tabanı):







 [13]

### c. Alt Sistemleri Donanıma Eşleme

Projede kullandığımız bir donanım cihazı bulunmamaktadır.

### d. Kalıcı Veri Depolama

Sistemimizdeki kullanıcıların ve noktaların bilgilerinin saklanması için bir veritabanı kullanılacaktır. kullanıcıların ödeme / iletişim / araç bilgileri depolanır. Ayrıca, noktaların mevcut durumu rezervasyon için her zaman güncellenecek ve saklanacaktır.

VERİ SÖZLÜĞÜ

ADMİN

|  |  |
| --- | --- |
| Admin\_id | Int | NN |
| Admin\_name | String(55) || NN |
| Admin\_password | String(55) || NN |

CARS

|  |  |
| --- | --- |
| Car\_id | Int | NN |
| Plate\_number | String(25) || NN |

CAR\_PARKINGS

|  |  |
| --- | --- |
| Parking\_id | Int | NN |
| Car\_id | Int | NN |
| Car\_entry\_date | Datetime | NN |
| Car\_release\_date | Datetime |
| Parking\_space\_id | Int | NN |

PLANS

|  |  |
| --- | --- |
| Plan\_id | Int | NN |
| Plan\_name | String(255) | NN |
| Capacity | Int | NN |

PARKING\_SPACES

|  |  |
| --- | --- |
| Parking\_space\_id | Int | NN |
| name | String(255) | NN |
| X1\_coord | Double | NN |
| Y1\_coord | Double | NN |
| X2\_coord | Double | NN |
| Y2\_coord | Double | NN |
| X3\_coord | Double | NN |
| Y3\_coord | Double | NN |
| X4\_coord | Double | NN |
| Y4\_coord | Double | NN |

\*NN = Not null boş geçilemez. [15]

### e. Ağ Protokolü

Biz otomasyonumuzda TCP/IP modeli kullanmak istedik, çünkü daha güvenilir ve http kullanır ve farklı katmanlarda çalışır.

### f. Global Kontrol Akışı

Yürütme düzeni:

Araç sahibi otoparka ister randevu oluşturarak yada otoparkımızı ziyaret edip kayıt oluşturarak arabasını park edebilir.

Zamana bağlılık:

Otoparka web sitesinden online olarak yapılan randevulu sistemlerde aracın belli bir zaman içerisinde otoparka girmemmiş olması durumunda sistemden çıkışı yapılacaktır.

Eş zamanlılık:

İşlemlerimizin hepsi web üzerinden veya admin girişi tarafından yapılmaktadır ve her bir işlem takip edilir ve günceldir. İşlemler veri tabanında güncel bir şekilde tutulmaktadır.

### g. Donanım Gereksinimleri

1. Website Host Server

1GB RAM

10GB Disk Space

100Mbps Connection Speed

2. Database

4GB RAM

64GB Disk Space

1Mbps

**[16]**

# 10. Algoritmalar ve Veri Yapıları

1. **Algoritmalar**

Projemizde matematiksel model uygulayarak yapılan toplam 4 adet algoritmamız vardır.

Bunlar;

1. Otoparkta bulunan toplam araç sayısının hesaplanması
2. Müşterinin çıkarken ödeyeceği ücretin hesaplanması
3. Gün içerisindeki toplam gelirin hesaplanması
4. Herhangi bir park yerine giriş, çıkış ve rezervasyon yapma işlemleri

1- Otoparkta bulunan toplam araç sayısı hesaplanması

Eğer otoparka araç giriyorsa;

Otoparktaki araç sayısı = Otoparktaki araç sayısı + 1

Eğer otoparktan araç çıkıyorsa;

Otoparktaki araç sayısı = Otoparktaki araç sayısı + 1

2- Müşterinin çıkarken ödeyeceği ücretin hesaplanması

(Çıkış zamanı – Giriş zamanı) \* Birim zamandaki ücret = Ödenecek ücret

3- Gün içerisindeki toplam gelirin hesaplanması

Araç çıkış yaparken toplam gelir güncellenir.

Toplam gelir = Toplam gelir + Ödenecek ücret

4- Herhangi bir park yerine giriş, çıkış ve rezervasyon yapma işlemleri

Araba x\*park yerine giriş yaptığı zaman;

x\*park yeri = Dolu

Araba x\*park yerinden çıkış yaptığı zaman;

x\*park yeri = Boş

Kullanıcı x\*park yerini rezerve ettiği zaman;

x\*park yeri = Rezerve

15dk içinde geldiğinde

x\*park yeri = Dolu

15dk içinde gelmediğinde

x\*park yeri = Boş [17]

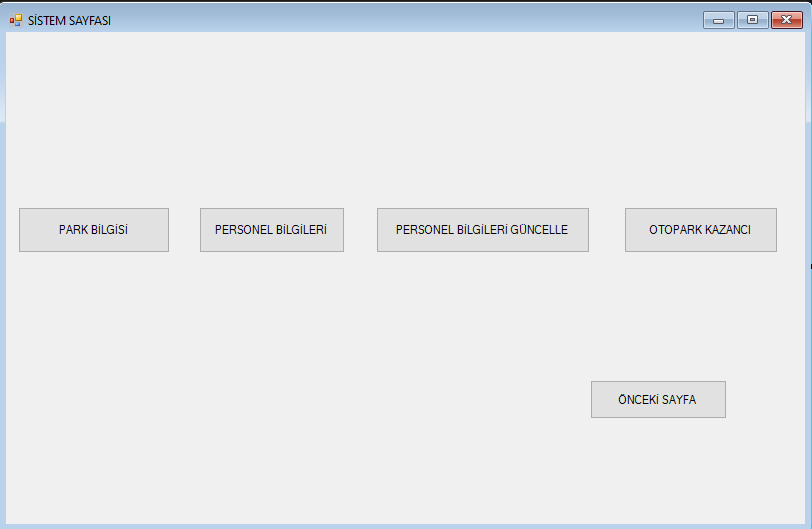
1. **Veri Yapıları**

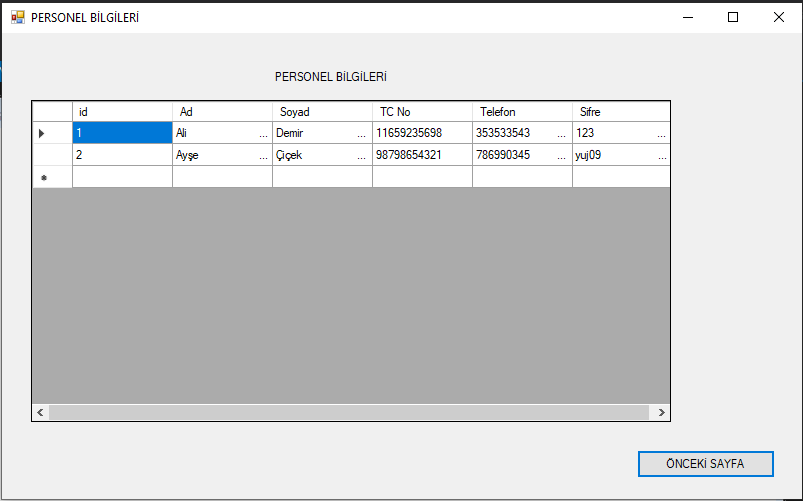
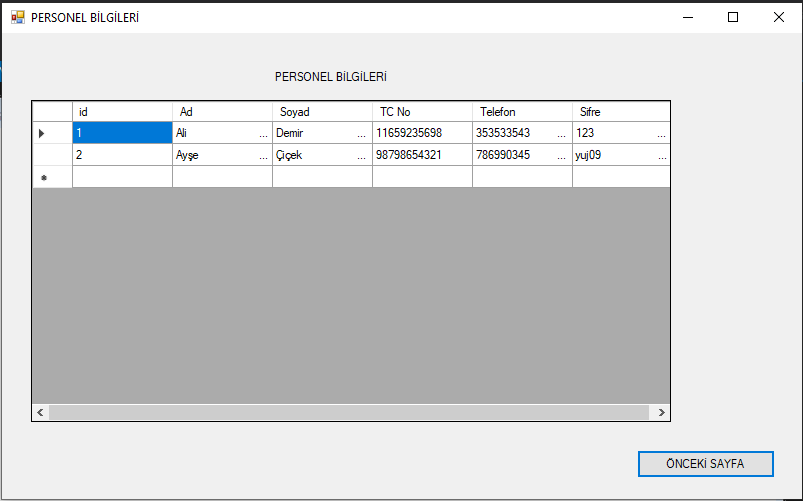
Sistemimiz çok boyutlu bir dizi veri yapısını kullanıyor.

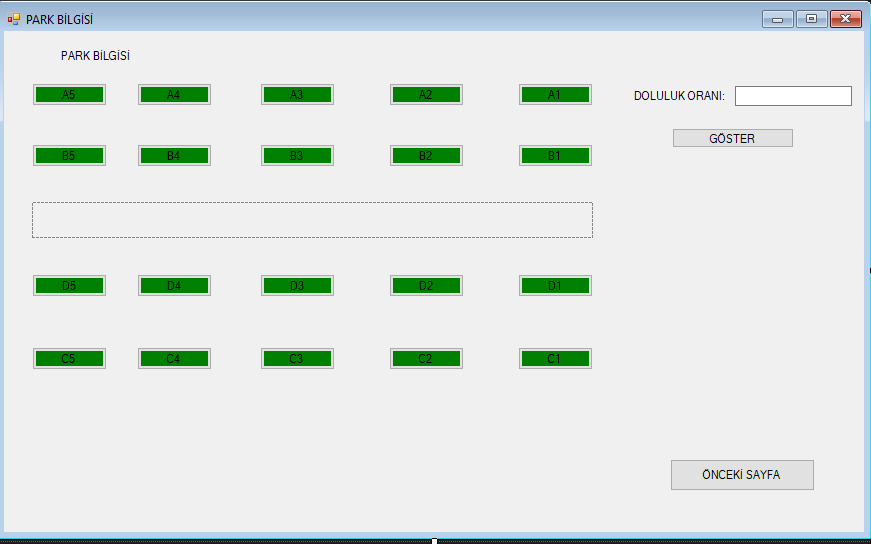
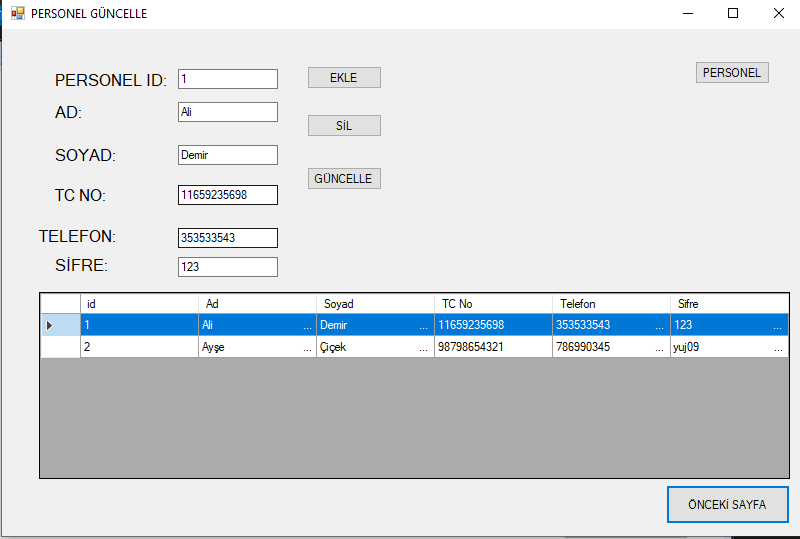
* 1. Çok boyutlu bir dizi veri yapısını kullanmaya karar verirken diziye erişim hızı ve performans kriterlerine göre kara verdik.[18]

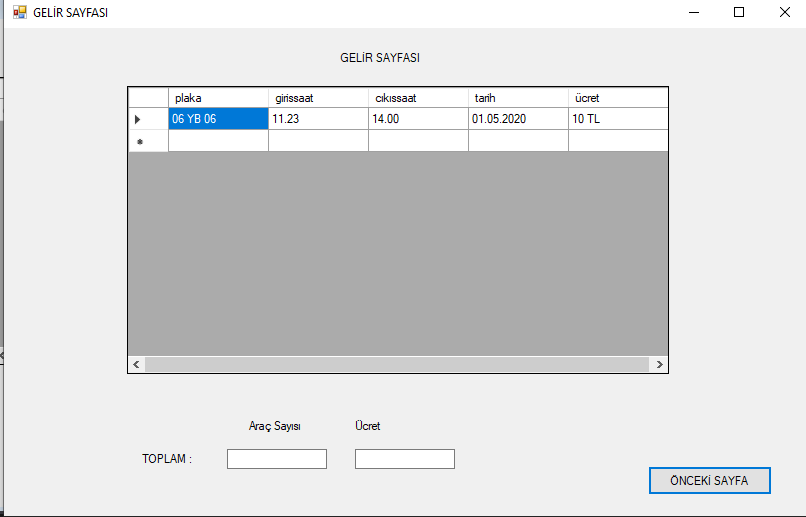
# 11.Kullanıcı Arayüzü Tasarımı ve Uygulaması



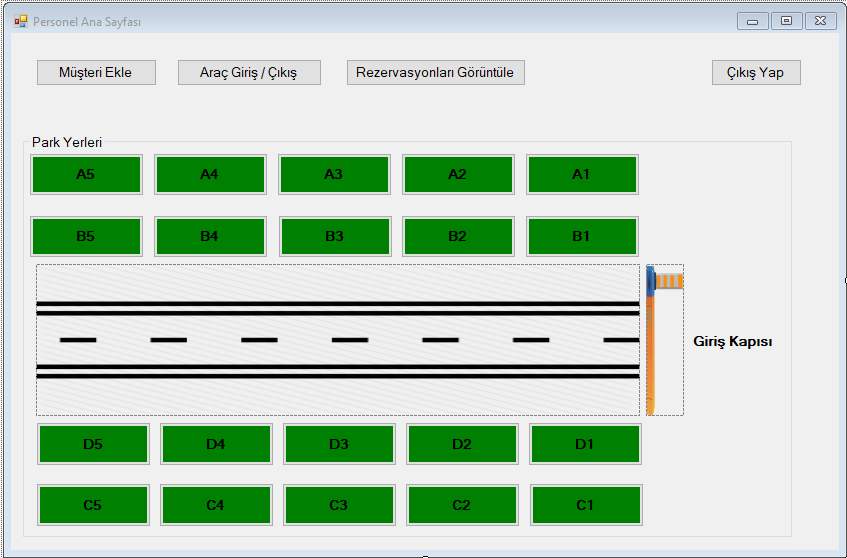


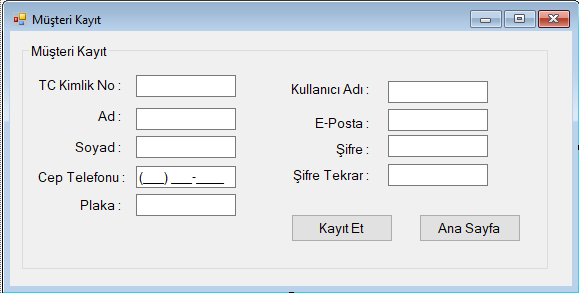


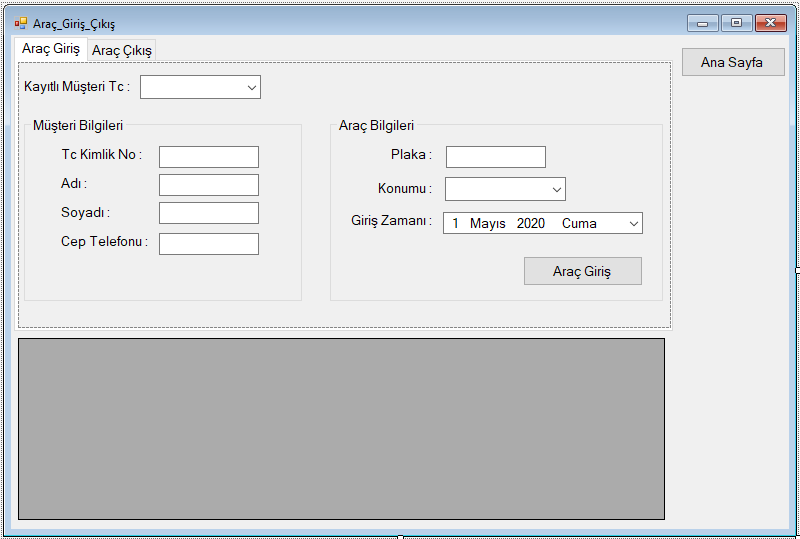


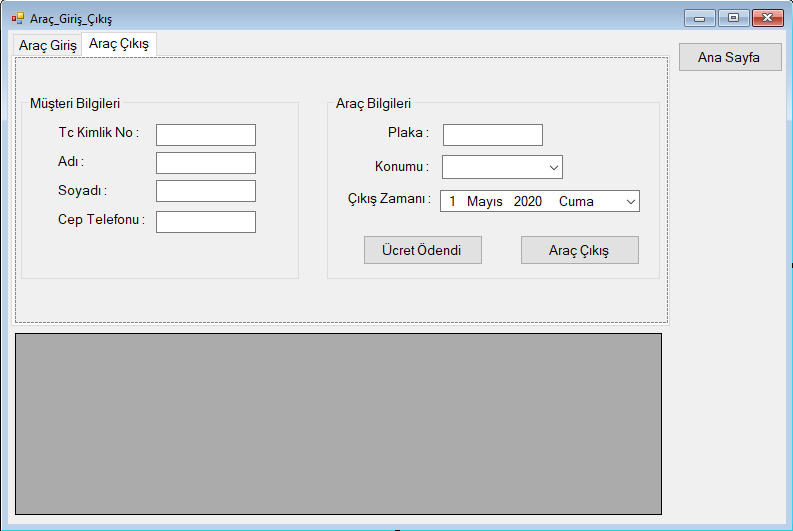












# 12 Test Tasarımı

1. **Giriş**

Yazılımın tüm test prosedürlerini, yazılımın hedeflenen ihtiyaçları karşıladığını doğrulamak için gerekli olan test ortamının hazırlanışı, test durumları ve yapılacak olan kullanıcı test faaliyetlerini göstermektedir.

1. **Test Edilecek Birimler**

Yönetici Giriş Testi

Ücretlendirme Testi

Uygun Park yeri

Yönetici Ekleme Testi

Geçmiş Kayıtları Listeleme Testi

Araç Giriş Testi

Araş Çıkış Testi

Yöneticileri Listeleme Testi

Web Servis Bağlantı Testi

Veri Tabanı Bağlantı Testi

1. **Yazılıma İlişkin Riskler**

**3.1 Zaman Riskleri**

Projenin, yanlış görev ve malzeme paylaşımından dolayı beklenen süre içerisinde gerçekleşmesine veya tamamlanmasına engel olan veya sebep olan risk çeşididir.

* 1. **Bütçe(maliyet) riskleri**

Gerçekçi olmayan bütçe tahminleri sonucu finansal sorunlara yol açan risklerdir.

**3.3 Yönetim riskleri**

Yönetim riskleri; amaçların net olmayışı, planlama eksikliği, yönetim tecrübesi ve eğitim eksikliği, iletişim sorunları, örgütsel sorunlar, otorite eksikliği ve kontrol problemlerini kapsamaktadır.

**Teknik riskler**

Genelde fonksiyonların yanlış olmasından kaynaklanır. Müşteri taleplerinin sürekli değişmesi, gelişmiş tekniklerin kullanılmaması ve geliştirilecek olan projenin zor faaliyetler içermesi gibi sebeplerden kaynaklanmaktadır.

* 1. **Program riskleri**
  2. Proje kapsamının dışına çıkan, kontrol dışı durumlardan veya önceliklerin sürekli değişmesinden doğan risklerdir.
  3. **Personel riskleri**

Personel riskleri; personel duraklamaları, deneyim ve eğitim sorunları, etik ve ahlak

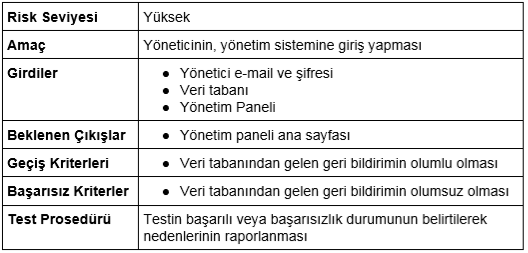
konularını, personel çatışmalarını ve verimlilik sorunlarını içermektedir.

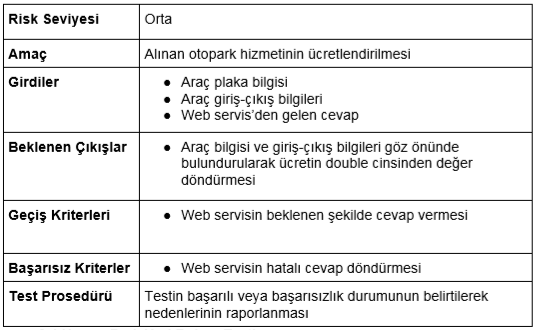
**3.6 Diğer kaynaklı riskler**

Diğer kaynaklı riskler, mevcut olmayan veya geç teslim edilen ekipman ve sarf malzemeleri, yetersiz aracı, yetersiz tesisleri, dağıtılan bölgeleri, bilgisayar kaynaklarının olmayışı ve yavaş tepki sürelerini kapsamaktadır.

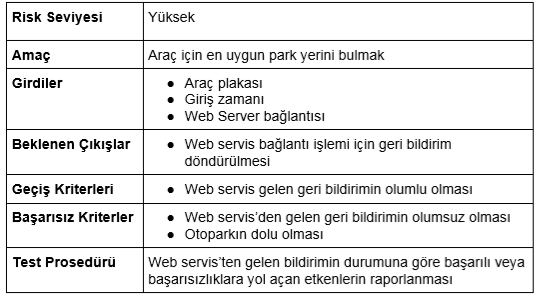
**4.Test edilecek özellikler**

**4.1 Yönetici Giriş Testi**

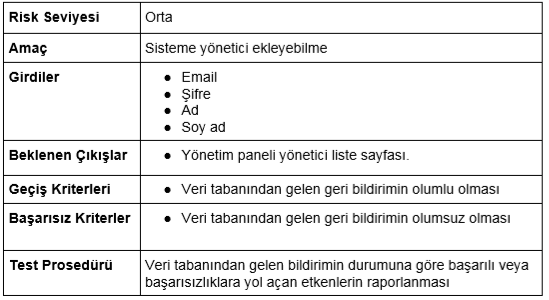
****

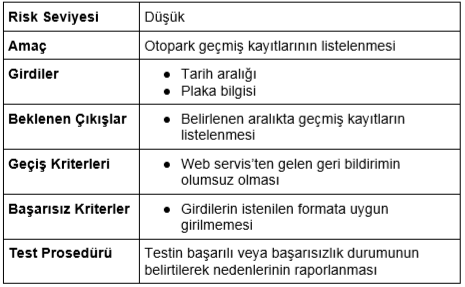
**4.2 Ücretlendirme Testi**

**4.3 Uygun Park Yeri Bulma Testi**

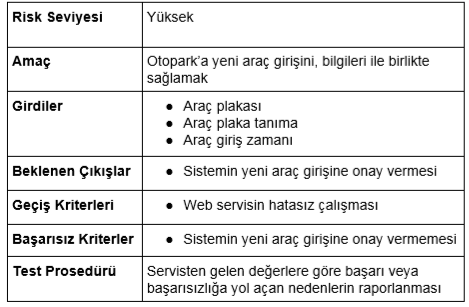


**4.4 Yönetici Ekleme Testi**

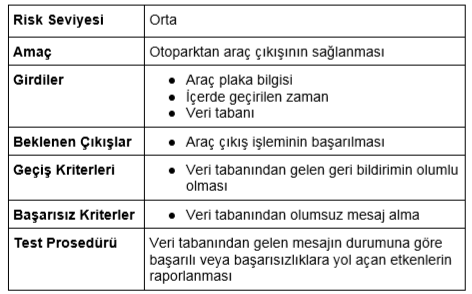


**4.5 Geçmiş Kayıtları Listeleme Testi**

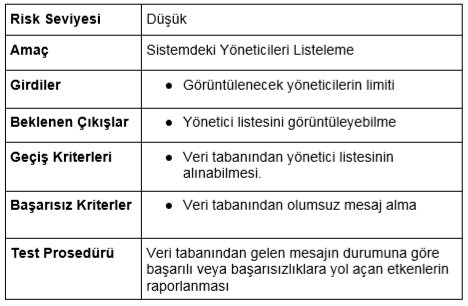
**4.6 Araç Giriş Testi**

****

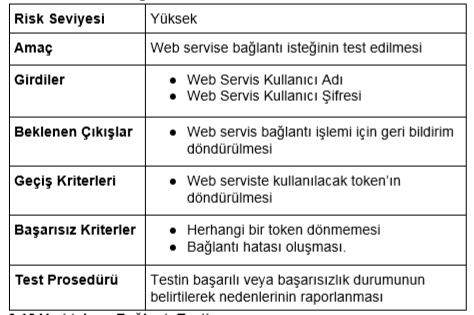
**4.7 Araç Çıkış Testi**

****

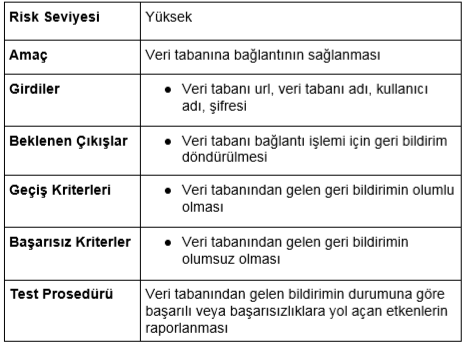
**4.8 Yönetici Listeleme Testi**

****

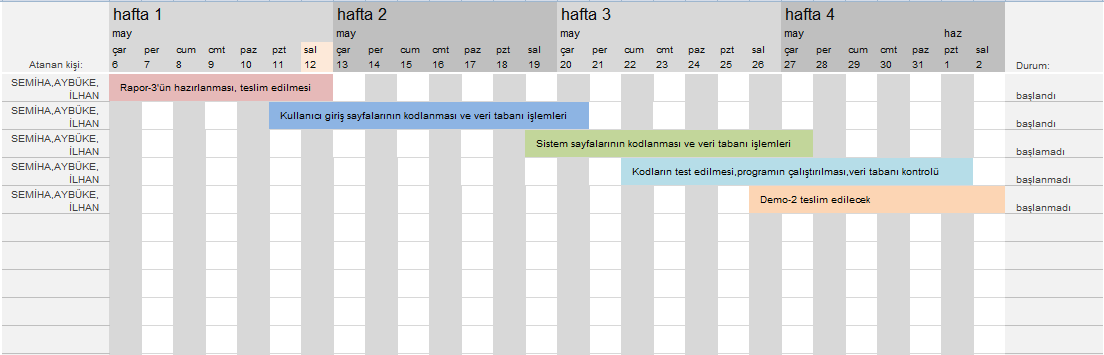
**4.9 Web Servise Bağlantı Testi**

****

**4.10 Veri Tabanı Bağlantı Testi**

****

[19]

13 .İş Planı

# KAYNAKÇA

[1]<https://www.youtube.com/watch?v=Gw35c29JCpk>

[2]<https://ramazanakbuz.com/c-otopark-otomasyonu-sql-veritabanli-kaynak-kod-2020/>

[3] **Örnek Garaj Otomasyonu**

[4]<https://www.akinsoft.com.tr/programlar/detay.php?code=akinsoft-otopark-programi--WOP2>

[5]<https://www.wottopark.com/?gclid=EAIaIQobChMI48HBgu-L6AIVFeDtCh05tAjxEAAYASAAEgIfYvD_BwE>

[6]<http://www.adnanmenderesairport.com/tr-TR/otopark>

[7]<https://www.parkkolay.com/>

[8]<https://www.youtube.com/watch?v=F7QnpyvdvgQ>

[9]<http://www.yazilimcilardunyasi.com/2017/01/use-case-puan-use-case-points-ucp.html>

[10]<http://www.eee.ktu.edu.tr/bitirme.dosyalar/bitirme_projeler_archive/06_2013-2014_Bahar/228495%20Fatih%20YAL%C3%87INKAYA/228495%20Fatih%20YAL%C3%87INKAYA_vqtwws.pdf>

[11]<http://univera-ng.blogspot.com/2009/11/uml-ve-modelleme-bolum-3-use-case.html>

12]<https://www.gliffy.com/>

[13]<https://www.ikinciyeni.com/blog/iyi-surucu-rehberi-detay/nereye-park-edecegiz-otopark-sorununu-cozmenin-yollari>

[14]<https://www.fidanelektronik.com.tr/otopark-otomasyon-sistemleri.html>

[15]<http://www.gorselprogramlama.com/otopark-takip-programi-10-cnet/>

[16]<https://www.wikihow.com.tr/%C3%87al%C4%B1%C5%9Fma-Plan%C4%B1Nas%C4%B1l-Yaz%C4%B1l%C4%B1r>

[17]<https://drive.google.com/file/d/1RQ26huGqVRWvY2e1KdBET1fYBYtIKs60/view>

[18]<https://office.live.com/start/excel.aspx?omkt=tr-TR>

[19]<https://office.live.com/start/Word.aspx?omkt=tr-TR>