**Fonksiyonel Gereksinimler**

1. **Sisteme Giriş:** Kullanıcı/Müşterilerin sisteme gerekli bilgilerini girdikten sonra giriş yapabilmeli ve sistem özelliklerini kullanabilmeli.
2. **Kullanıcı/Müşteri Kaydı Oluşturma:** Müşterilerin sistem üzerinden araç kiralama işlemlerini yapabilmesi için sisteme kayıt oluşturması sağlanmalı.
3. **Araçların Listelenmesi:** Sistemdeki farklı sınıf ve modeldeki araçlar listelenmeli. Listelenen sayfada araç hakkında detaylı bilgi verilmeli.
4. **Rezervasyon İşlemleri:** Müşteriler gerekli doğrulama işlemlerini yaptıktan sonra seçtikleri araca ait rezervasyon yapabilmeli.
5. **Doğrudan Kiralama İşlemi:** Müşteri, aracı hemen kiralama durumunda kredi kartı ile ödeme işlemini yaparak aracı doğrudan kiralayabilmeli.
6. **Doğrulama İşlemi:** Müşterilerin kimliği, telefonlarına gelen SMS ile doğrulanabilmeli.
7. **Müşteriye Mail Gönderimi:** Müşteriye rezervasyon yaptıktan sonra veya diğer gerekli durumlarda otomatik mail gönderimi yapılabilmeli. (Kiralama işleminden sonra mail üzerinden e-fatura gönderimi)
8. **Yetkilendirme:** Yetki sahibi çalışanlar, sistem üzerinden gerekli yetkilendirmeleri yapabilmeli. (Yeni araç Ekleme, Güncelleme, Silme yetkileri)

**Teknik Gereksinimler**

1. **Multi Platform Desteği:**

* Kullanılacak sisteme hem mobil uygulama üzerinden hem de web sitesi üzerinden erişilebilmeli.
* Mobil tarafta Android ve iOS işletim sistemlerinin ikisinde de sorunsuz kullanılabilmeli.
* Web ve Mobil arasındaki uyumluluk sağlanmalı.

1. **Performans:** Sistem yoğun kullanım olduğu durumlarda bile sorunsuz şekilde çalışacak bir altyapıya sahip olmalı.
2. **Güvenlik:**

* Müşteri bilgilerinin güvende kalması ve dışarıdan erişilememesi için gerekli ve güncel protokollere sahip bir güvenlik duvarı olmalı.
* Yapılacak herhangi bir siber saldırıda güvenlik sağlanabilmeli ve güvenlik açıkları belli aralıklarla test edilip sistem güvenliği her daim sağlanmalı.

1. **Ödeme İşlemi:**

* Kredi kartı ile ödeme yapılması durumunda güvenliği sağlamak için gerekli ödeme teknolojileri kullanılmalı.
* Kullanılacak teknoloji *PCI DSS* standartlarına uymalı.

1. **Veri tabanı:** Araç, müşteri ve log bilgilerini depolayabilmek için performanslı ve veri bütünlüğü olan bir veri tabanı kullanılmalı.
2. **Loglama İşlemleri:** Sistemde yapılan her hareketi sonradan incelenebilecek şekilde kayıt altına alınabilmeli.
3. **Server/Sunucu:**

* Server’ın performanslı çalışabilmesi için yeterli donanım ve altyapı kullanılmalı.
* Düzenli aralıklarla yedeklemeler alınmalı.
* Server 7/24 sorunsuz çalışabilecek kapasitede olmalı.
* Server’da oluşacak sorunlar sebebiyle sistemde. Kesintiler yaşanacak olursa bu durum yetkili kişilere uyarı şeklinde iletilebilmeli.

1. **API Kullanımı:**

* Kullanılacak olan API’nin detaylı şekilde dokümantasyonu yapılmalı.
* API’nin güvenliği sağlanmalı.
* API’nin performanslı çalışması için gerekli optimizasyonlar yapılmalı.

1. **Uygun Yazılım Mimari Stil Seçimi:**

* Projeye en uygun mimariyi seçip mimariye uygun altyapı sağlanmalı.

**Kalite Nitelikleri**

**Performans**

1. **Veri Tabanıyla İlgili Performans Senaryosu:**

* Büyükverilere sahip Veri tabanını karmaşık sorgular ile zorlayıp performansının yeterliliğini test edeceğiz.
* Eğer karmaşık sorgulara rağmen hızlı şekilde sonuçları geri döndürebiliyorsa Veri tabanımız performanslı şekilde çalışıyor diyebiliriz.

**Senaryo işleyişi:**

* Müşteri sisteme giriş yapıp herhangi bir aracı filtreleyerek araçları listeler.
* Bu filtreleme sonucunda araçlar sorunsuz ve hızlı şekilde müşterinin karşısına çıkmalıdır.

1. **Sistem Yoğunluğuyla İlgili Performans Senaryosu:** Sistemde çok yoğun trafik olduğu bir durum olduğunu varsayalım.

* Sistemin yoğunluk durumlarında işlevini yerine getirip getirmediği test edilecektir.

**Senaryo İşleyişi:**

* Müşteri sisteme giriş yapıp herhangi bir işlem gerçekleştirsin.
* Sistemde binlerce müşterinin aynı anda işlem yaptığı durumu ele alıp analiz yapalım
* Server kapasitesi, sistemdeki kullanıcı sayısı ve işlem yanıt süreleri gibi değerlere bakılarak test edilmeli.
* Kullanıcı sayısı ve işlem yanıt süreleri en optimum duruma gelince test tamamlanır ve sistemin yoğun isteklere cevap verebildiği doğrulanır
* Aksi durumda ise sistemin yeterli donanım ve altyapıya sahip olmadığı anlamına gelir. Bu durumda gerekli güncellemeler ve optimizasyonlar yapılıp server daha performanslı hale getirilmelidir.

**Güvenlik**

1. **Verilerin Korunmasına İlişkin Senaryo:** Müşterilerin kişisel bilgilerine siber saldırı ile erişmeye çalışılan bir senaryoyu düşünelim.

* Bu senaryo sonucunda kişisel verilere erişimin güvenliğinin sağlandığını ve saldırgan işlemlerin tespit edildiği test edilecektir.

**Senaryonun İşleyişi:**

* Sisteme yetkisi olmadan sızmaya çalışan bir hacker var.
* Sisteme girmeye çalışan hacker sistem tarafından fark edilir ve gerekli güvenlik önlemleri devreye girer.
* Bu saldırı sistemdeki yetkili kişilere hemen uyarı şeklinde bildirilir.
* Hackerin işlemleri tespit edilir ve kayıt altına alınarak güvenlik önlemlerinin arttırılması amacıyla kullanılır. (Sistem zafiyetlerini en aza indirgemek için)
* Bu test sonucunda eğer hacker sisteme erişim sağlayamazsa sistemin güvenlik önlemleri sorunsuz şekilde çalışıyor demektir.

**Süreklilik**

1. **Sistemin 7/24 Çalışma Durum Senaryosu:** Sistemin herhangi bir hata veya sorun karşısında dahi kesintisiz çalışmaya devam edebilmesini test edeceğimiz senaryo.

* Sistemdeki olağanüstü durumlar sebebiyle kesinti olma durumlarında en kısa sürede ikincil sistem veya önlemler devreye girerek sistemin çalışmaya devam etmesi sağlanmalıdır.

**Senaryonun İşleyişi:**

* Sistem normal şekilde çalışırken herhangi bir kesinti olması durumunu ele alalım.
* Sistem kesintiye uğradığı zaman bu durum sistemdeki yetkili kişilere bildirilir ve müşterilere de giriş ekranında belirtilir.
* Sistem kesintisi durumunda devreye ikincil sistem girerek sistem kaldığı yerden çalışmaya devam eder.
* Bu kesinti süreci raporlanıp analiz edilir ve muhtemel kesintilerin önlenmesi için kullanılmak üzere kayıt altına alınır.