

Spark

2.0

04.01.2021

Suzan Nur Ayaydın
Elif Ece Erdem
İsmail Enes Kırlı
Ezgi Uysal

Revizyon Geçmişi

Tarih	Açıklama	Yazar	Yorumlar
30/10/20	Proje konusunun belirlenmesi	Suzan Ayaydın	Proje konusu belirlendi, proje kapsamında gerçekleştirilebilecek özellikler üzerinde beyin fırtınası yapıldı.
9/11/20	Uygulama özellikleri	İsmail Enes Kırılı	Uygulamada olması gereken/olacak bütün özellikler belirlendi.
23/11/20	Uygulamanın kodlanmaya başlanması	Elif Ece Erdem	Belirlenen özelliklerin gerçekleştirilmesi için uygulama kodlanmaya başlandı.
30/11/20	Ara rapor teslimi	Ezgi Uysal	Proje ile ilgili gerekli maliyet analizleri, pazar araştırması, teknik analiz vs. tamamlandı.
26/12/20	Kodlamanın tamamlanması	İsmail Enes Kırılı	Yapılan analizler sonucu uygulama için belirlenen özellikler tamamlanmış, uygulama ayağa kaldırılmıştır.
03/01/21	Yazılım tasarım raporunun tamamlanması	Elif Ece Erdem	Yönergeler doğrultusunda yazılım tasarım raporu tamamlanmıştır.

İçindekiler Tablosu

REVİZYON GEÇMİŞİ	II
1. GİRİŞ	3
1.1 AMAÇ	3
1.2 KAPSAM	4
1.3 GENEL BAKIŞ	6
2. SİSTEME GENEL BAKIŞ	6
3. SİSTEM MİMARISI	7
3.1 MİMARİ TASARIM	7
3.2 MİMARİ BİLEŞENLERİN AÇIKLANMASI	11
4. VERİ TASARIMI (DATA DESIGN)	13
4.1. VERİ BELRTİMİ	13
4.2 VERİ TABANI TASARIMI	16
5. KULLANICI ARAYÜZLERİ	19
5.1 KULLANICI ARAYÜZLERİNE GENEL BAKIŞ	19
5.2 EKRAN RESİMLERİ	21
6. KULLANILAN KÜTÜPHANELER VE ARAÇLAR	32

1. Giriş

Bu belge, Spark isimli mobil uygulamasının genel sistem mimarisini ve veri mimarisini içermektedir. Galatasaray Üniversitesi Bilgisayar ve Endüstri Mühendisliği son sınıf öğrencileri tarafından gerçekleştirilecek olan bu projenin fonksiyon ve özelliklerinin tanımlarını belirtmektedir.

1.1 Amaç

Özellikle İstanbul’da uzun süredir ciddi bir sorun olan, kalabalık semtlerde park yeri bulamama sorununu çözmeyi ya da azaltmayı amaçlayan bir uygulama geliştirmek hedeflenmiştir. Uygulama temel olarak İstanbul’da yer alan park yerlerinin kullanıcı konumuna uzaklığını, park yerinin saatlik ücretlerini ya da aylık üyelik ücretlerini kullanıcılarla paylaşmaktadır, ayrıca otoparkların anlık doluluk durumlarını da ekranda bilgi olarak sunmaktadır. Bunun yanında eğer mümkünse kullanıcıya otoparkta bulunan boş yeri rezerve etme imkanı sunmaktadır. Ayrıca günümüzde Covid-19 pandemisi sebebiyle de insanların teması azaltma isteğine de değinerek, kullanıcılar otoparktan ayrılırken kaldıkları süre boyunca olan ücreti mobil uygulama üzerinden kart kaydederek ya da Cüzdan’a (wallet) para yükleyerek online ödeme imkanı da sunulmaktadır. Bu sayede İstanbul trafiğinde yaşanan stresin büyük sebeplerinden biri olan park sorununu çözmek ve sürücülerin bu esnada yaşadığı zaman kaybını azaltmak amaçlanmaktadır.

1.2 Kapsam

Bu yazılım tasarım raporu, Spark yazılımı ile ilgili sistem ve yazılım mimarisi, modüller, sınıflar, fonksiyonlar, veritabanı modelleri, kullanıcı arayüzleri, kullanılan araçlar ile ilgili ayrıntılı bilgi içermektedir.

Spark yazılımı ile ilgili aşağıdaki bilgileri vermek faydalı olacaktır:

- (1) Bu proje kapsamında iki adet yazılım üretilecektir. Bunlar kullanıcı yani otopark arayan kişi için ve otoparkta çalışan görevli için özelleştirilmiş yazılımlardır.
- (2) Otopark arayan kişinin kullanacağı uygulamada misafir ve kayıtlı kullanıcı olmak üzere iki farklı şekilde giriş yapmak mümkündür. Bunu yapmaktaki amaç, sisteme kayıt olmak istemeyen kullanıcıların da uygulamayı indirmesini sağlamaktır. Hedefimiz otopark arayan herkese hizmet etmektir. Burada kişi kayıtlı olmadan da konumuna en yakında bulunan otoparkları sıralı şekilde görebilir, bu otoparklarla ilgili ayrıntılı bilgiye ulaşabilir. Kayıtlı veya kayıtlı olmayan kullanıcıya en yakın otoparkların sıralanması şu algoritma takip edilerek yapılmaktadır:
 - Kullanıcının güncel konum bilgisi alınır,
 - Veritabanında hali hazırda bulunan otoparkların konum bilgileri ile kullanıcının güncel konum bilgisi arasındaki mesafeler belirlenir,
 - En kısa mesafeye sahip otoparkın bilgileri veritabanından çekilir.

Ancak kişinin rezervasyon/önceden ödeme yapma seçeneklerinden yararlanabilmesi için sisteme kayıtlı olma zorunluluğu vardır. Buradaki amaç bilgi güvenliğini sağlamaktır. Kayıtlı kullanıcıların kredi kartı bilgileri ve Cüzdan bakiyeleri sistem üzerinde şifreli ve güvenli bir şekilde tutulmaktadır. Eğer kullanıcı kayıtlı değil ve rezervasyon yaptırmak istiyorsa kayıt olma ekranına yönlendirilir.

Kayıtlı kullanıcı eğer rezervasyon yaptırmak istiyorsa, istediği rezervasyon seçeneğini, ardından ödeme seçeneğini seçer. Burada sistemde kayıtlı olan kredi

kartı ile ödeme yapma ve Cüzdan’a daha önceden yüklediği bakiye ile ödeme yapmak üzere iki ödeme seçeneği mevcuttur. Sistem üzerinde kredi kartını kaydetmek istemeyecek kullanıcılar için alternatif bir ödeme seçeneği olan Cüzdan sistemi uygulama üzerinde kullanılmaktadır.

Eğer kredi kartı ile ödeme sağlanacak ise, sistemde varsayılan kredi kartından ilgili ücret banka ile gerekli işlemlerin yapılması sonucunda çekilir. Eğer Cüzdan ile ödeme sağlanacak ise, cüzdanda yeterli bakiye olup olmadığı kontrol edildikten sonra rezervasyon işlemi gerçekleştirilir, kullanıcıya onay QR’ı gönderilir ve otoparka konu ile ilgili bilgi gider.

Ayrıca kayıtlı kullanıcı otopark ile ilgili yorum yapabilir, daha önceden yapmış yorumları okuyabilir. Buradaki amaç, yapılan yorumlar doğrultusunda otoparkların hizmet kalitesini artırmasına yardımcı olmaktır.

Son olarak kullanıcı; kullanıcı bilgileri ekranını kullanarak kayıtlı kredi kartlarını görüntüleyebilir, Cüzdan’ına para ekleyebilir, ve İspark’ın Borç/Ceza sayfasına yönlendirilebilir.

- (3) Yazılımın otoparkçı ayağında ise; otoparkçı uygulama üzerinden ödeme yapmamış araç otoparka girdiğinde bu aracın plakasını sisteme girer. Burada plaka ve saat bilgisi veritabanında tutulur. Araç çıkarken yine aracın plakasını ilgili yere girer, aracın plaka ve çıkış saat bilgisi veri tabanına gönderilir. Veritabanındaki otoparkın saatlik ücret bilgisi ve aracın kaç saat otoparkta kaldığı bilgisi kullanılarak kullanıcının ödeme yapması için bir QR kodu oluşturulur.

Eğer kullanıcı SPark ile rezervasyon yaptırmış, dolayısıyla ödeme yapmış ise; kullanıcıya ödeme sonrası verilen onay kodu ilgili yere girilerek bunun kontrolü otoparkçı tarafından yapılır. Veritabanında onay kodu olup olmama duruma göre otoparkçının ekranına “Onay” ya da “Red” mesajı gönderilir.

1.3 Genel Bakış

Yazılım Tasarım Raporunun geri kalanı, sırasıyla; sistem hakkında genel bilgileri, sistem mimarisini, veri tasarımını, kullanıcı arayüz tasarımlarını, uygulama geliştirilirken kullanılan kütüphaneler, framework ve teknolojilerle ilgili genel bilgileri içermektedir.

2. Sisteme Genel Bakış

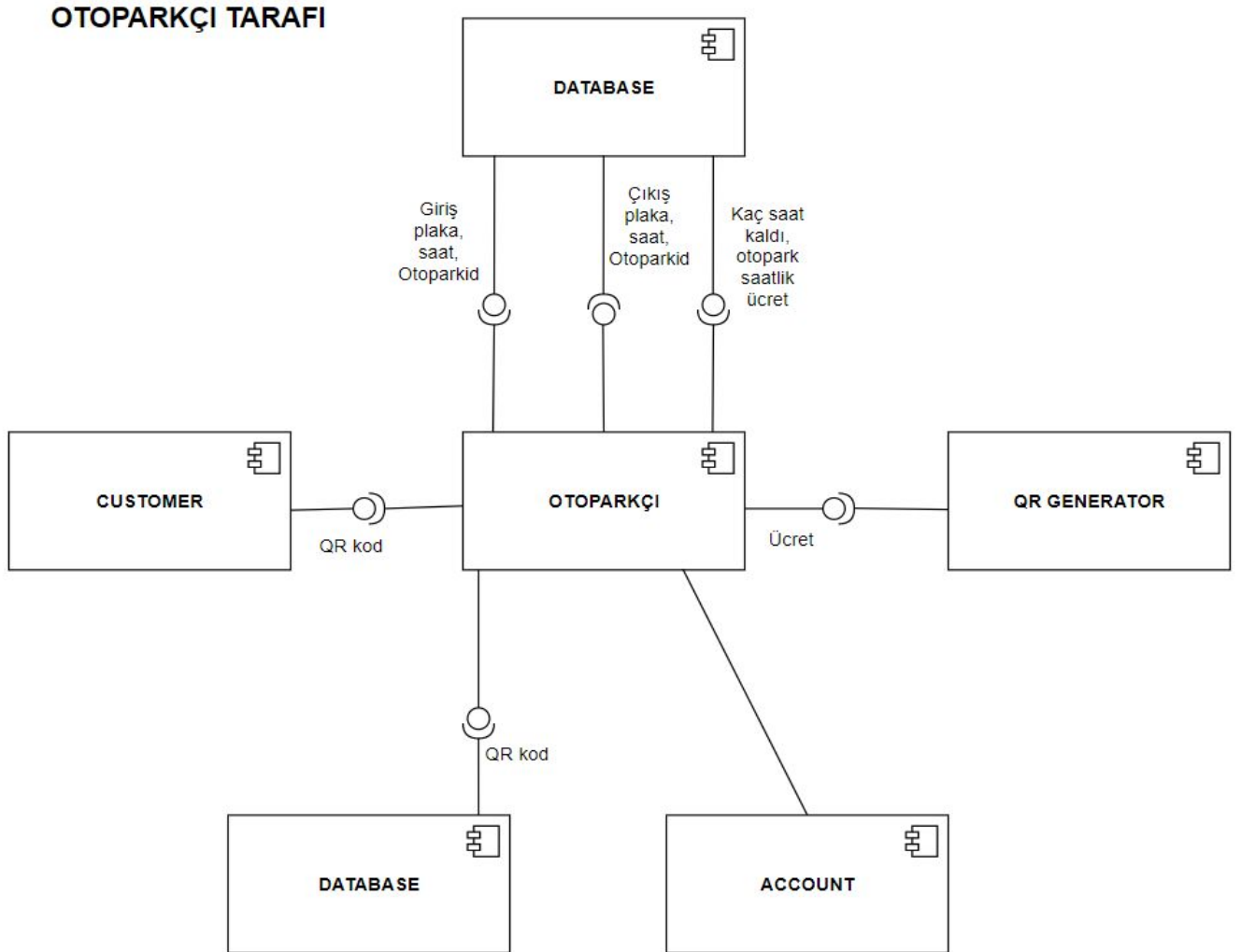
Proje kapsamında gerçekleştirilecek uygulamanın kullanıcı tarafında öncelikle kullanıcının konum koordinatlarını alarak yakındaki park yerlerini tespit eder. Bu park yeri/otoparkların konumları ile arasındaki mesafeyi hesaplayarak kullanıcıya en yakın otoparkları sıralar. Bu sıralama sırasında otoparkların kapalı/açık otopark olmaları, saatlik fiyat bilgisi, ne kadar uzakta olduğu, kullanıcı oylarıyla elde ettiği puan ve doluluk durumu görülür. Bunun yanında daha detaylı bilgi almak için seçili otoparka tıklandığında otoparkın iletişim bilgileri, adresi, saatlik ücret detayları, aylık abonelik olma durumu ve vale hizmeti olup olmadığı görülür. Ayrıca adresin üzerine tıklanma durumunda Google Maps'e yönlendirme yapılarak Google Maps'in sunmuş olduğu navigasyon hizmetinden de faydalanılabilir. Bunun yanında da dilerse otoparkta kendisine yer ayırtabilir, bunun için rezerve ettiği andan itibaren otopark saati işlemeye başlar ve saatlik ücret üzerinden ücretlendirme yapılır. Kullanıcı otoparkta işi bittikten sonra ödemesini pratik olarak uygulama üzerinden gerçekleştirebilir. Bunun için 2 seçenek bulunur biri önceden kayıtlı kredi kartı ile ödeme diğeri ise kredi kartı ile para yüklenen cüzdan sistemidir. Bu sayede kullanıcılar boş yer olan ve fiyatını tercih ettiği otoparka gidebilir ve kolaylıkla ödemesini de mobil uygulama üzerinden gerçekleştirebilir. Ayrıca, kullanıcı bilgileri ekranında İspark Borç/Ceza butonu ile İspark'ın ilgili sitesine yönlendirme olanağı sunularak kullanıcının plakasına ait İspark'ta kayıtlı herhangi bir ceza veya borç olup olmadığını kontrol edebilir.

Proje kapsamında gerçekleştirilecek uygulamanın otoparkçı kullanıcısı tarafında, otoparkçı; otoparka giren ve otoparktan çıkan aracın plakasını yazarak ücretlendirmenin

yapılmasını sağlar. Bu noktada, QR kodu oluşturularak müşterinin QR kodu tarayıp ödeme yapması sağlanır. Ayrıca, otoparka gelmeden mobil olarak ödeme yapan müşterinin ödemesinin alınıp alınmadığının kontrolünü de yapabilir. Bu kontrol, ödeme yapan müşterinin ekranına gelen onay kodunun otoparkçı tarafından sisteme girilmesi ile gerçekleştirilir. Son olarak otoparktaki doluluk oranını da görebilmektedir.

3. Sistem Mimarisi

3.1 Mimari Tasarım

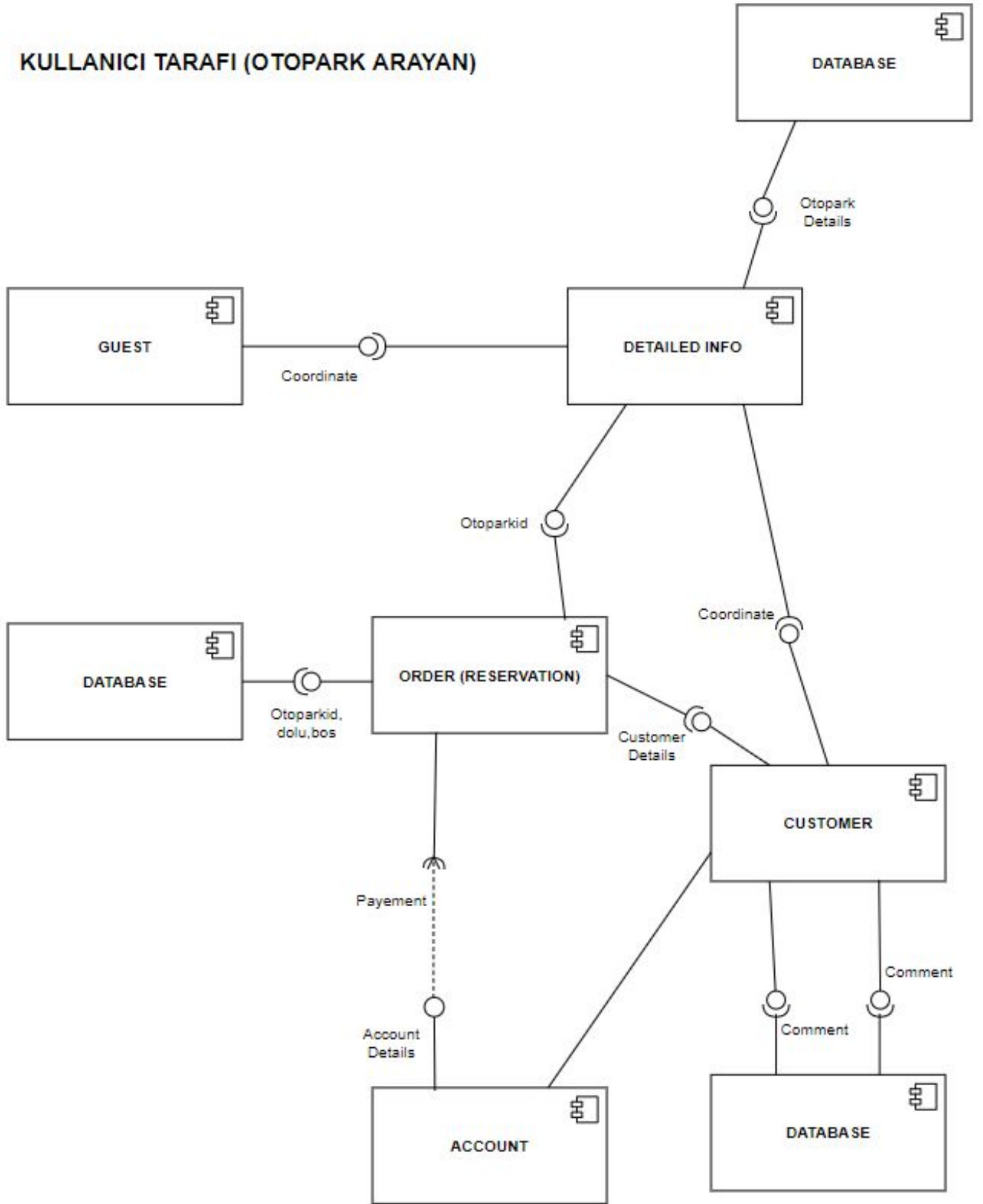


Şekil - 1

Uygulamanın Otoparkçı tarafında temel olarak otoparkçının yapabildiği 4 işlem vardır.

Bunlar:

1. Giriş yapan aracın plakasının sisteme girilmesi, giriş plaka bilgisinin sistem tarafından otomatik olarak saat bilgisi ile birlikte veritabanına kaydedilmesi
2. Çıkış yapan aracın plakasının sisteme girilmesi, çıkış plaka bilgisinin sistem tarafından otomatik olarak saat bilgisi ile birlikte veritabanına kaydedilmesi
3. Çıkış yapan araca ödeme yapabilmesi için QR kodu üretilmesi. Bu işlem veri tabanından gelen otoparkta kalma saati (çıkış saatinden giriş saati çıkarılarak bulunur) ve o otoparka ait saatlik ücret bilgisinin sistem içindeki fonksiyona koyulması ile ücretin bulunması ve bu ücrete ait QR kodunun üretilmesi ile gerçekleştirilir.
4. SPark sistemini kullanarak önceden ödeme yapmış kullanıcının ödeme yapıp yapılmadığını kontrol etmek için kullanıcının gösterdiği QR kodun okunması. Bu QR kodunun, veri tabanında kayıtlı olup olmadığına göre kullanıcının ödeme yapıp yapmadığı anlaşılmaktadır.



Şekil - 2

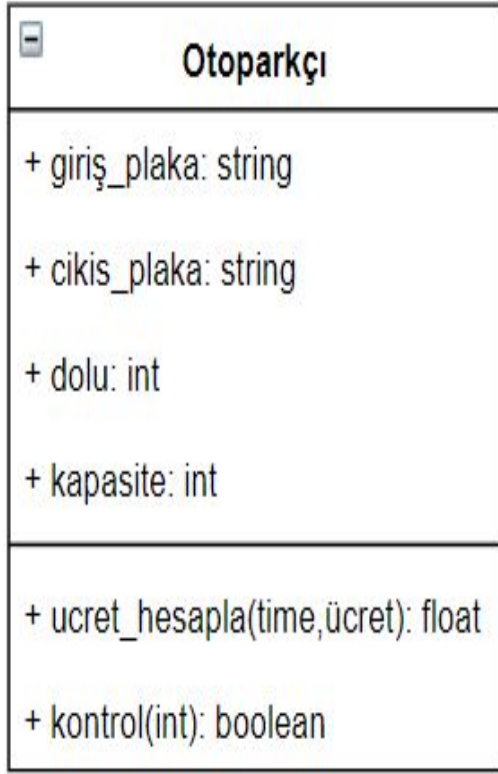
Kullanıcı tarafında uygulamayı kullanan kimse uygulamaya kayıt olmadan yani misafir olarak ya da kayıtlı kullanıcı olarak giriş yapabilir.

Misafir kullanıcı, konum bilgisini sisteme göndererek sistem tarafından kullanıcıya yakınlık derecesine göre sıralanmış otopark verilerini görür. Ayrıca sıralanmış otoparklardan istediğine basarak veri tabanından o otopark ile ilgili verilerin çekilmesi ile o otopark ile ilgili ayrıntılı bilgiye ulaşabilir. Ancak o otopark için herhangi bir ödeme ya da rezervasyon yapamaz.

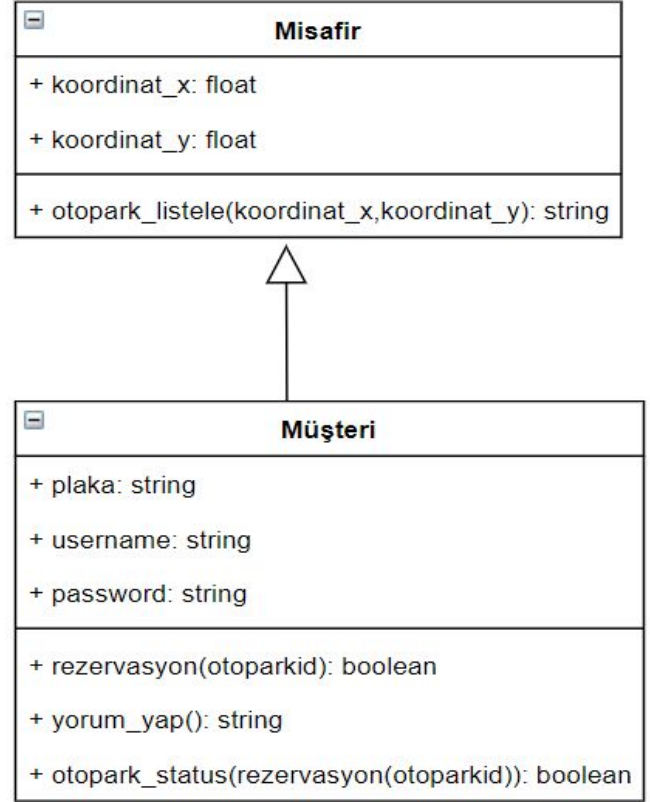
Kayıtlı kullanıcı (müşteri), misafir kullanıcının yaptığı işlemleri yapabilir (otoparkların kullanıcının konumuna yakınlığına göre sıralanması ve sıralı otoparklardan herhangi biri hakkında ayrıntılı bilgiye ulaşılması). Bunun yanında o otopark için rezervasyon yaptırabilir. Kullanıcının sistemde kayıtlı bilgileri (kayıtlı kredi kartı ya da Cüzdan) ile bu işlem gerçekleştirilir. O araca ait bir rezervasyon oluşturulduğunda veri tabanında bu bilgi gönderilir. Böylece otoparktaki dolu/boş yer bilgisi güncellenir.

Ayrıca kayıtlı kullanıcı o otoparka dair yorum da yapabilir. Kullanıcının yaptığı yorumların diğer kullanıcılar tarafından görüntülenebilmesi adına yorumlar veritabanına kaydedilir. Daha önce yapılmış yorumlar da ekranda görüntüleyebilmek için veritabanından çekilir.

3.2 Mimari Bileşenlerin Açıklanması



Şekil - 3



Şekil - 4

Yazılımın otoparkçı tarafında kullanıcı olarak otoparkçı bulunmaktadır. Otopark kullanıcısı girdi olarak giriş yapan ve çıkış yapan aracın plakasını yazar, o plakaya ait ücret bilgisini içeren QR kodu oluşturur ve ödemenin alınıp alınmadığını yine plaka yazarak ya da QR kodu taratarak kontrol eder. Bu işler şu şekilde gerçekleşmektedir:

1. Otopark kullanıcısının kodlandığı alanda giriş plakası yazıldığı anda sistem anlık saatini o plakanın kayıtlı olduğu kullancının Person tablosunda Pid'sinin olduğu alana yazar.
2. Otopark kullanıcısının kodlandığı alanda çıkış plakası yazıldığı anda sistem anlık saatini o plakanın kayıtlı olduğu kullancının Person tablosunda Pid'sinin olduğu alana yazar.
3. Ücret hesaplaması yapılırken veritabanına yazılmış çıkış saatinden giriş saati çıkarılır otoparkta kalınan toplam süre hesaplanır.

4. O otoparka ait saatlik ücret ile toplam kalınan süre çarpılarak ödenecek ücret belirlenir.
5. Ücretin ödenip ödenmediğini kontrol etmek için de ödemeyi yapan kullanıcının plaka bilgisi ya da gösterdiği QR kod taranarak veritabanında onay kodu tablosunda böyle bir kullanıcı olup olmadığı bilgisine göre bu kontrol gerçekleştirilir.

Yazılımın kullanıcı kısmında kullanıcı misafir ya da kayıtlı kullanıcı olarak giriş yapar. kayıtlı kullanıcı ya da misafir kullanıcı konumuna en yakın otopark listesini görüntüleme ve istediği otoparkla ilgili ayrıntılı bilgi edinme olanağına sahiptir. Ancak yorum yapma, rezervasyon yapma, ödeme özelliklerinden sadece kayıtlı kullanıcılar yararlanabilmektedir.

Otoparkların listelenmesi ve seçilen otoparka ait detaylı bilgilerin görüntülenmesi aşağıdaki algoritma takip edilerek yapılır:

1. Kullanıcının konum bilgisi koordinat olarak kodun içinde alınır.
2. Kullanıcıdan alınan koordinat bilgisi veritabanında kayıtlı otoparkların koordinat bilgisi ile karşılaştırılır.
3. Karşılaştırma sonucu otoparklar en yakından uzağa doğru sıralanır.
4. Kullanıcının seçtiği otopark, otoparkid bilgisi kullanılarak veri tabanında bulunur ve bu otopark hakkında detaylı bilgi ekranda görüntülenir.

Otoparka yorum yapılması/yorumların görüntülenmesi aşağıdaki algoritma takip edilerek yapılır:

1. Veritabanında kayıtlı yorumların ekranda görünmesi sağlanır.
2. Kullanıcı yorum yaptığında yapılan yorum veritabanına yazılır.

Otoparkta rezervasyon yapıp ödemenin yapılması aşağıdaki algoritma takip edilerek yapılır:

1. Rezervasyon yapılmak istenen otoparkta ne kadar süre rezervasyon yapılacağı bilgisi ve bunun ücreti koda gömülmüştür. Tutar varsayılan ödeme biçiminden (Kayıtlı kredi kartı/Cüzdan) tahsil edilir.

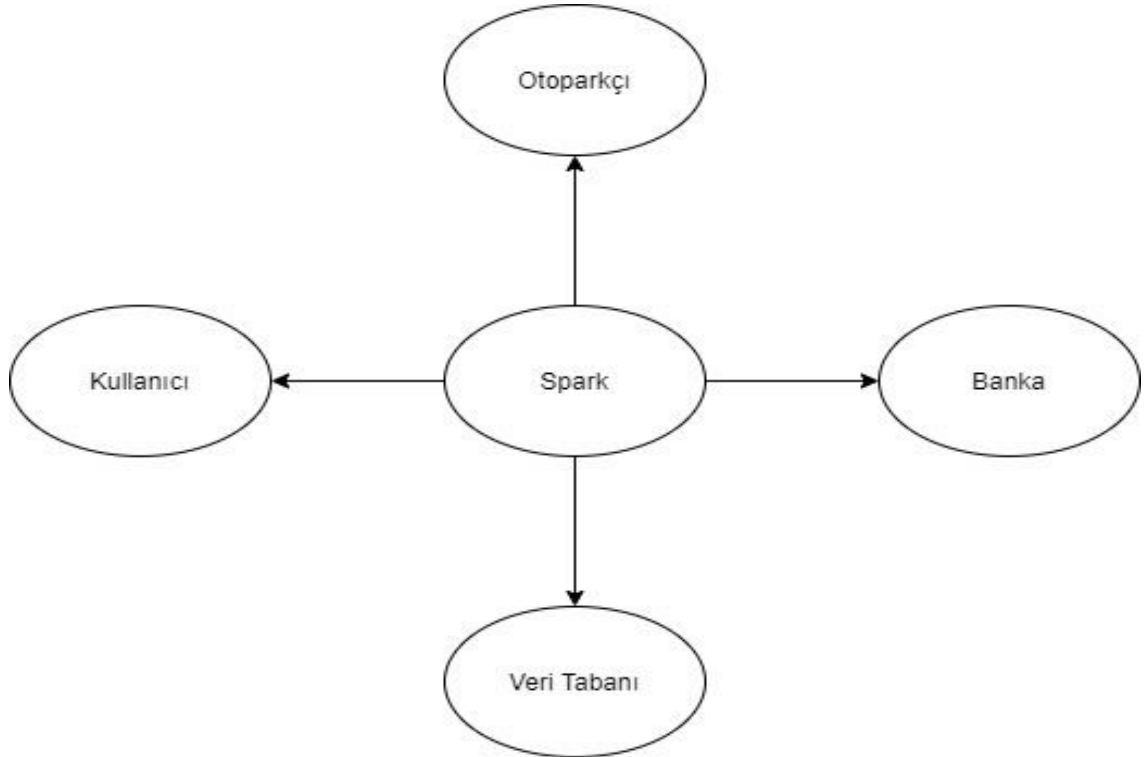
2. Ücret tahsil edildiğinde veritabanındaki otopark tablosundaki ilgili otoparkid'nin olduğu satırda dolu/boş bilgisi güncellenir.

4. Veri Tasarımı (Data Design)

4.1. Veri Belirtimi

Bu bölüm, diyagramlar ve tablolar kullanılarak veri nesnelerinin niteliklerini (attribute) ve veriler ile fonksiyonlar arasındaki ilişkiyi gösterir. Bu veri objeleri, gereksiz niteliklerden (attribute) ve normalleştirme faktörlerinden kurtulma düşüncesi altında yapılır.

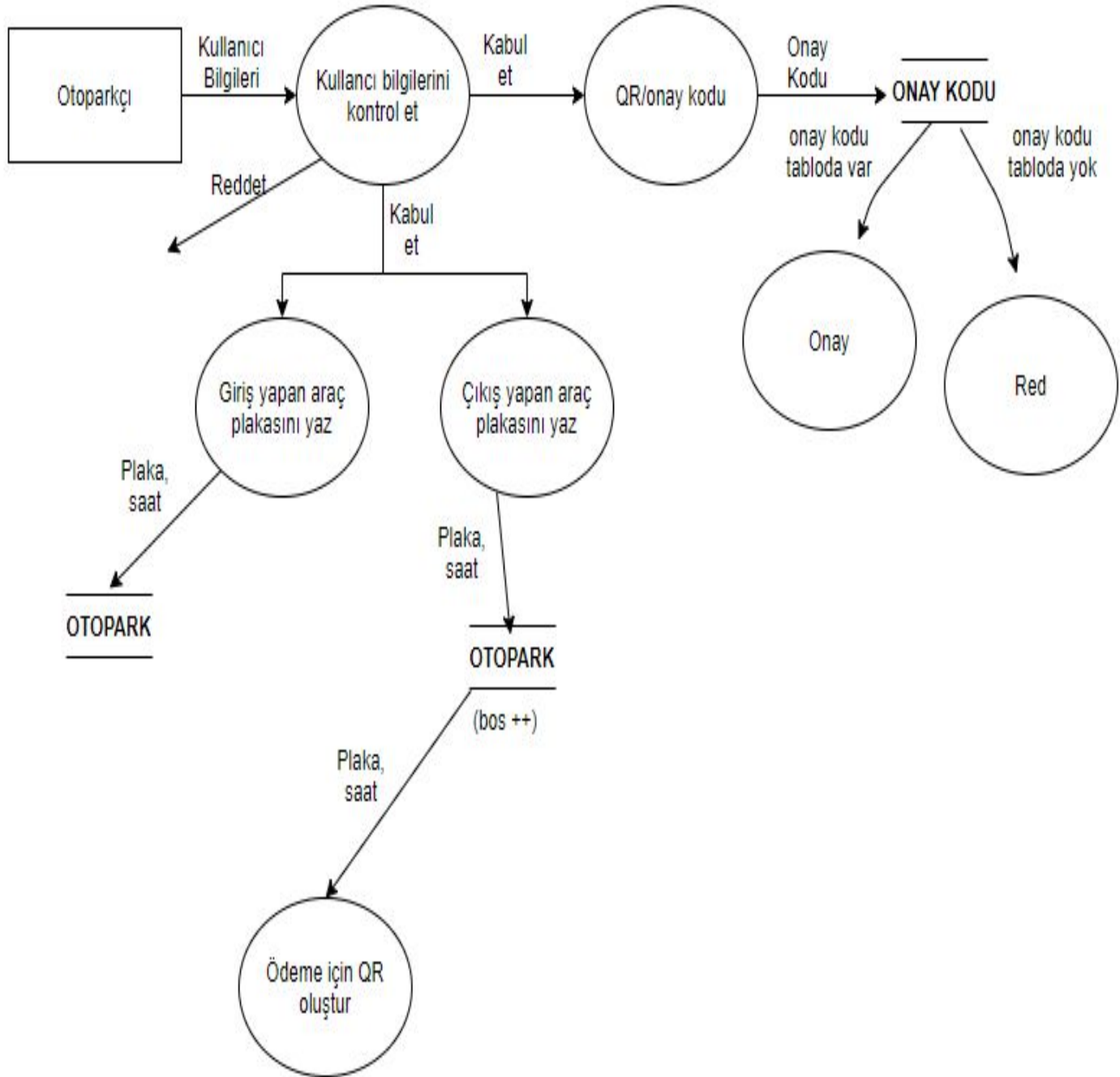
4.1.1 Context Diyagramı



Şekil - 5

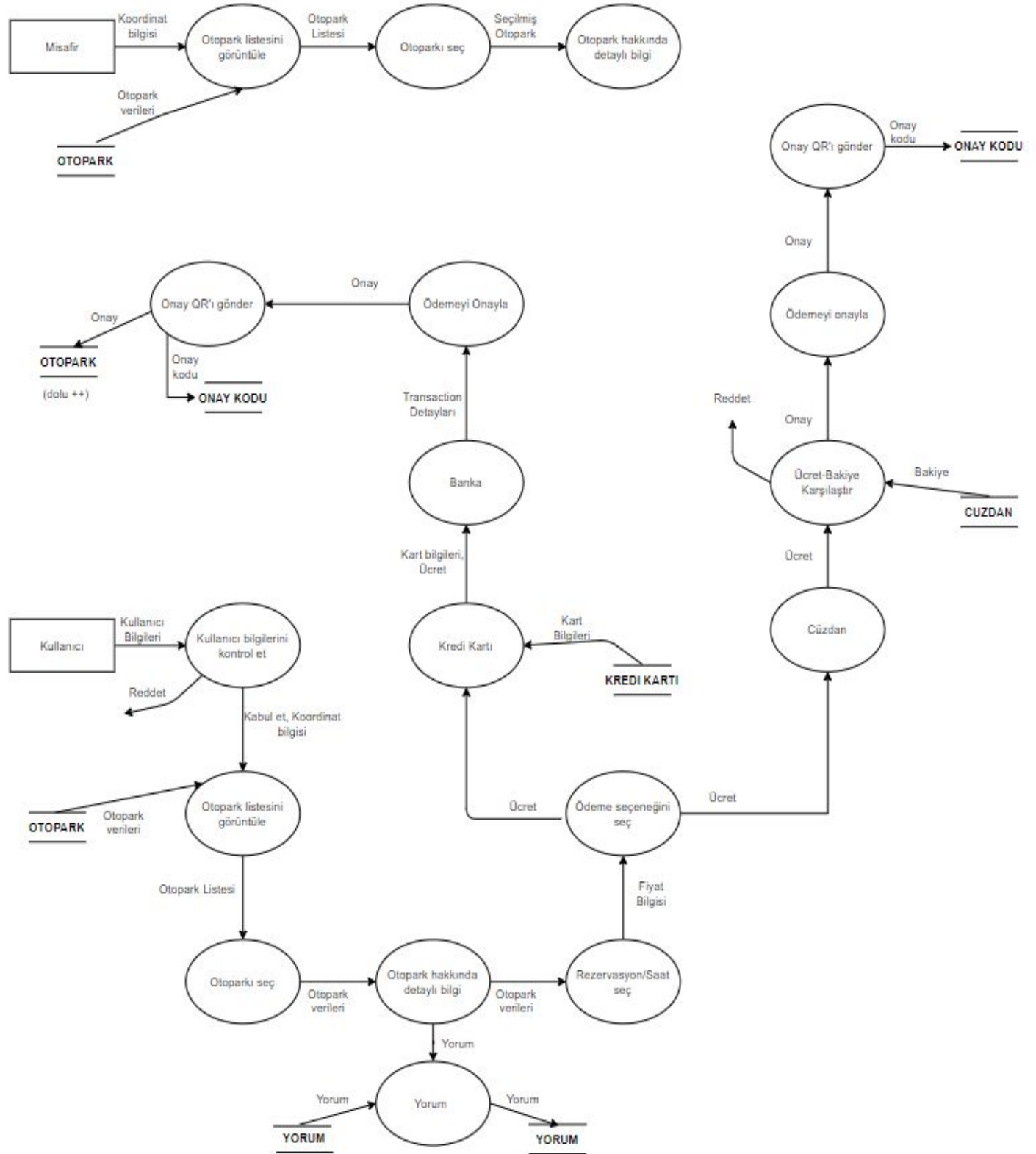
4.1.2 Veri akışı Diyagramı

OTOPARKÇI TARAFI



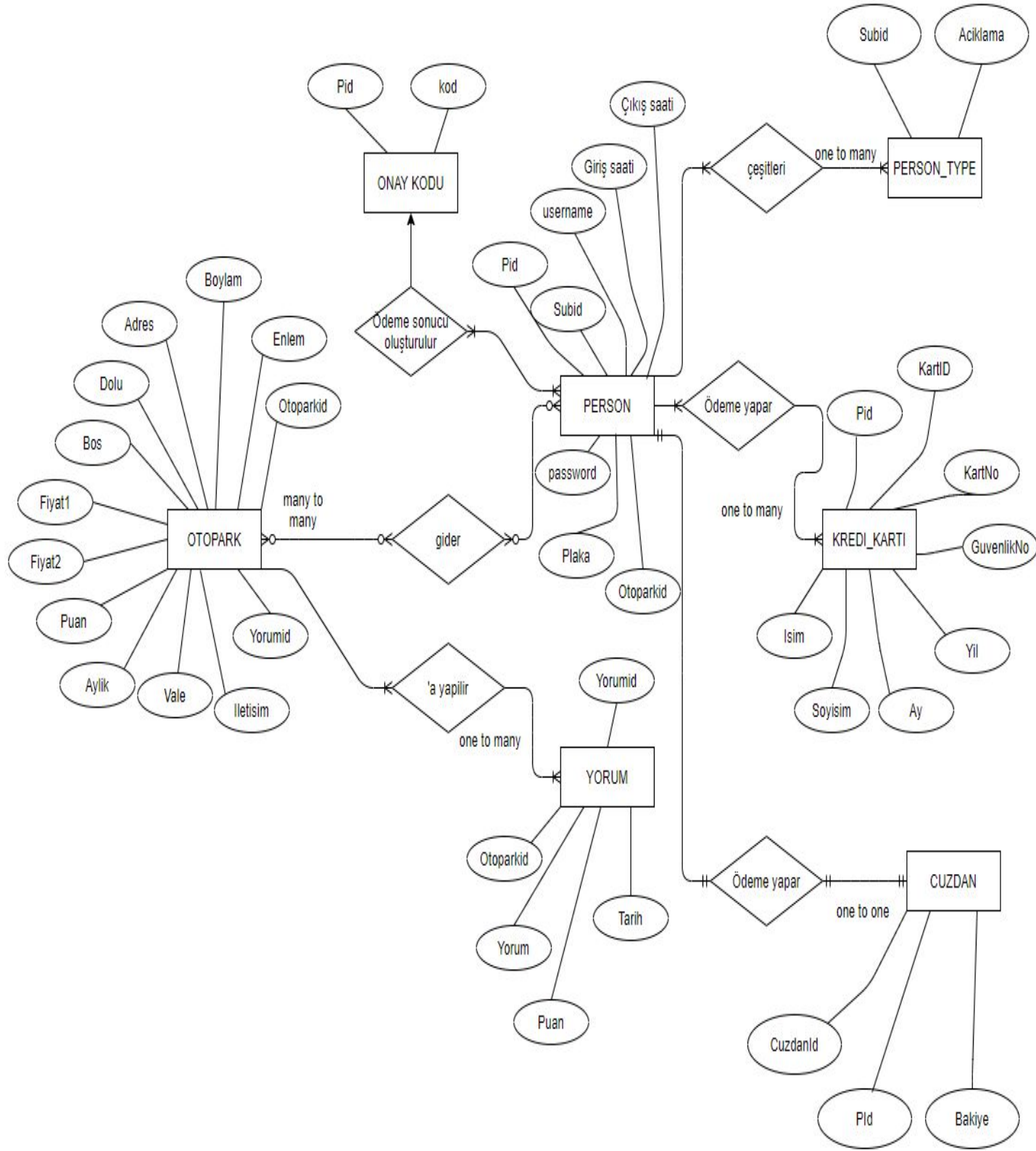
Şekil - 6

KULLANICI TARAFI (OTOPARK ARAYAN)



Şekil - 7

4.2 Veri Tabanı Tasarımı



Şekil - 8

Uygulamada kullanacağımız veri tabanında toplamda 6 adet tablo bulunmaktadır. Bunlar aşağıdaki gibidir:

1. PERSON tablosu: Bu tabloda primary key olan Pid, giriş yapan kişinin otopark görevlisi mi yoksa otopark arayan kullanıcı mı olduğunu ayırt eden Subid, kişinin gideceği ve aynı zamanda otopark görevlisinin çalıştığı otoparkı belirten Otoparkid, plaka, giriş saati, çıkış saati bilgisi, kullanıcı adı ve parola bilgileri bulunmaktadır.

2. PERSON_TYPE tablosu: Burada PERSON tablosunda foreign key olan Subid primary key'dir. Bu tablonun amacı, tablodaki açıklama sütunu ile uygulamaya giriş yapan kişinin otopark çalışanı mı yoksa otopark arayan bir kullanıcı olup olmadığını belirlemektir. (0-> otoparkçı, 1-> otopark arayan kullanıcıdır.)

3. OTOPARK tablosu: Burada PERSON tablosunda foreign key olan Otoparkid primary key'dir. Bu tabloda otoparka ait enlem, boylam bilgisi mevcuttur. Bu bilgi, otoparkın güncel konuma olan uzaklığının tespiti için kullanılacaktır. Ayrıca; adres, dolu yer sayısı, boş yer sayısı, 1 saatlik ücret (fiyat1), 2 saatlik ücret (fiyat2), otoparkın puanı, otoparkta aylık ücretlendirmenin olup olmadığı, otoparkta vale hizmeti olup olmadığı, otoparkın telefonu (iletişim) ve otoparka yapılan yorumları görüntülemek için Yorumid (foreign key) bilgileri yer almaktadır.

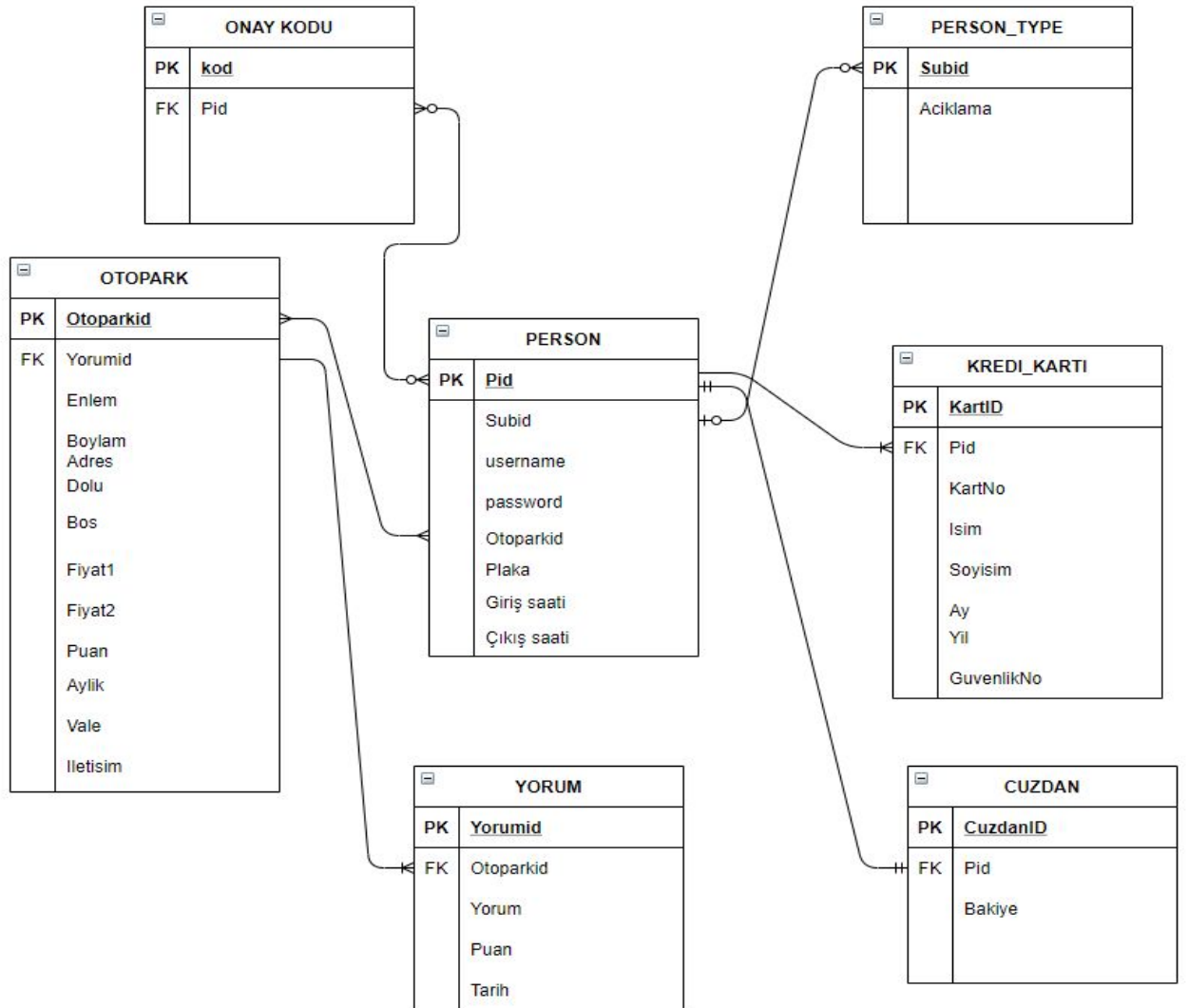
4. YORUM tablosu: Burada OTOPARK tablosunda foreign key olan Yorumid primary key'dir. Bu tabloda, otoparka yapılan yorumun hangi otoparka yapıldığını tutmak için Otoparkid, yapılan yorum, otoparka verilen puan ve yorumun ne zaman yapıldığı bilgileri yer almaktadır.

5. KREDI_KARTI tablosu: Sistemde kayıtlı olan kredi kartı bilgilerini tutmak için kurulmuş bir tablodur. Her kredi kartının kendine özel bir Cid'si (primary key), kartın hangi kullanıcıya ait olduğunu belirten Pid, kart numarası, kartın sahibinin adı-soyadı, kartın son kullanım tarihi (ayy/yıl) ve güvenlik kodu bilgileri bulunmaktadır. Burada kullanıcıların hassas bilgisi olan kredi kartı bilgileri güvenliğin sağlanması amacıyla şifrelenerek tutulmalıdır.

6. CUZDAN tablosu: Cüzdan ile ödeme yapmak isteyen kullanıcıların, cüzdanlarındaki bakiye bilgisinin, cüzdanın kime ait olduğu bilgilerinin tutulduğu tablodur.

7. ONAY KODU tablosu: Ödemesini yapan kullanıcıya ödemesinin onaylandığını belirten ve otoparkçı tarafından ödemenin yapılıp yapılmadığının kontrolünü sağlayan kodların tutulduğu tablodur. Ödeme onayı verilince üretilen kod bu tabloya eklenir. Otoparkçı kontrol için sorgu gönderdiğinde eğer tabloda böyle bir kod varsa ödeme başarıyla alınmış demektir.

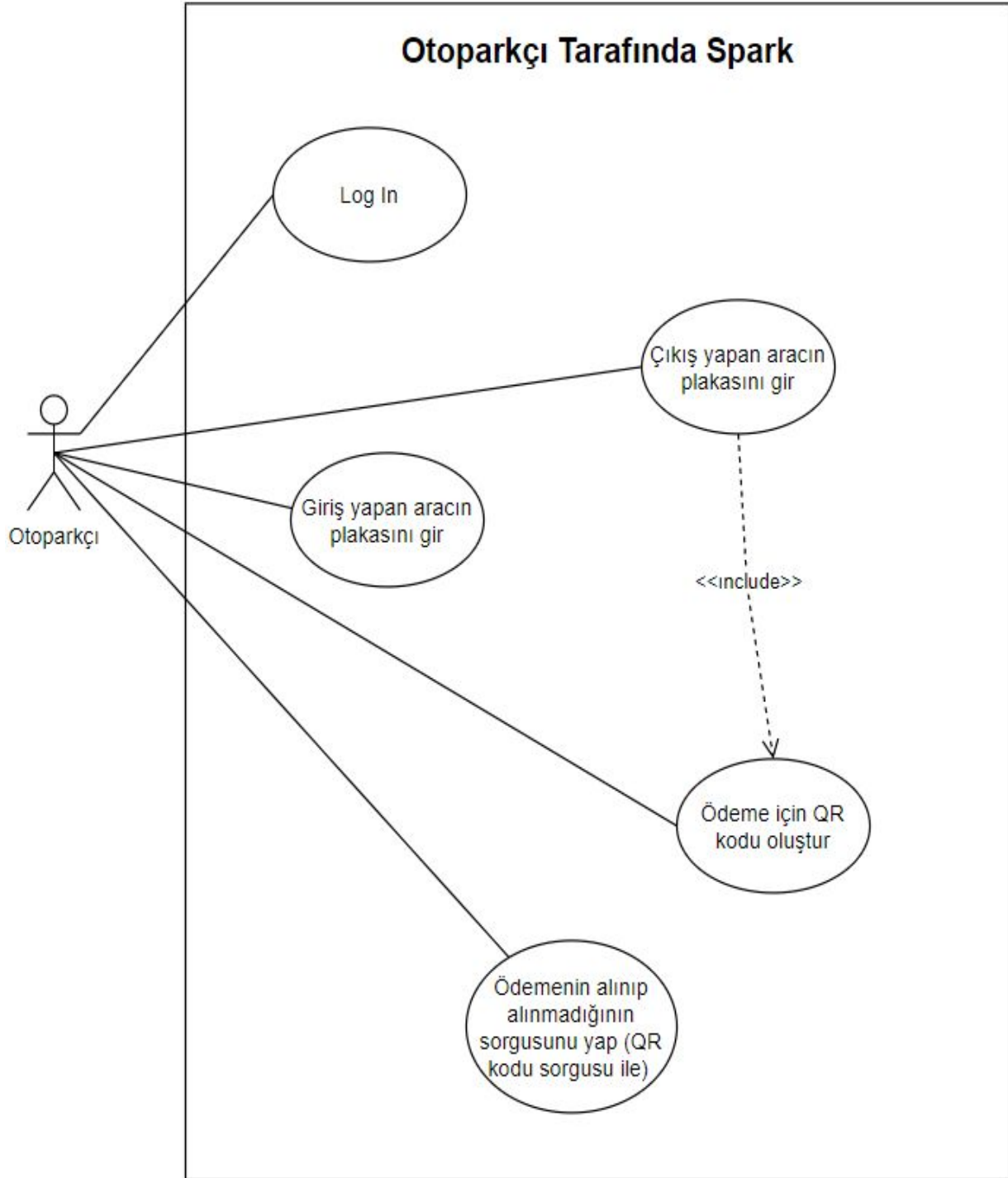
Veri tabanı üzerindeki bağlantılar aşağıdaki şekilde görselleştirilmiştir:



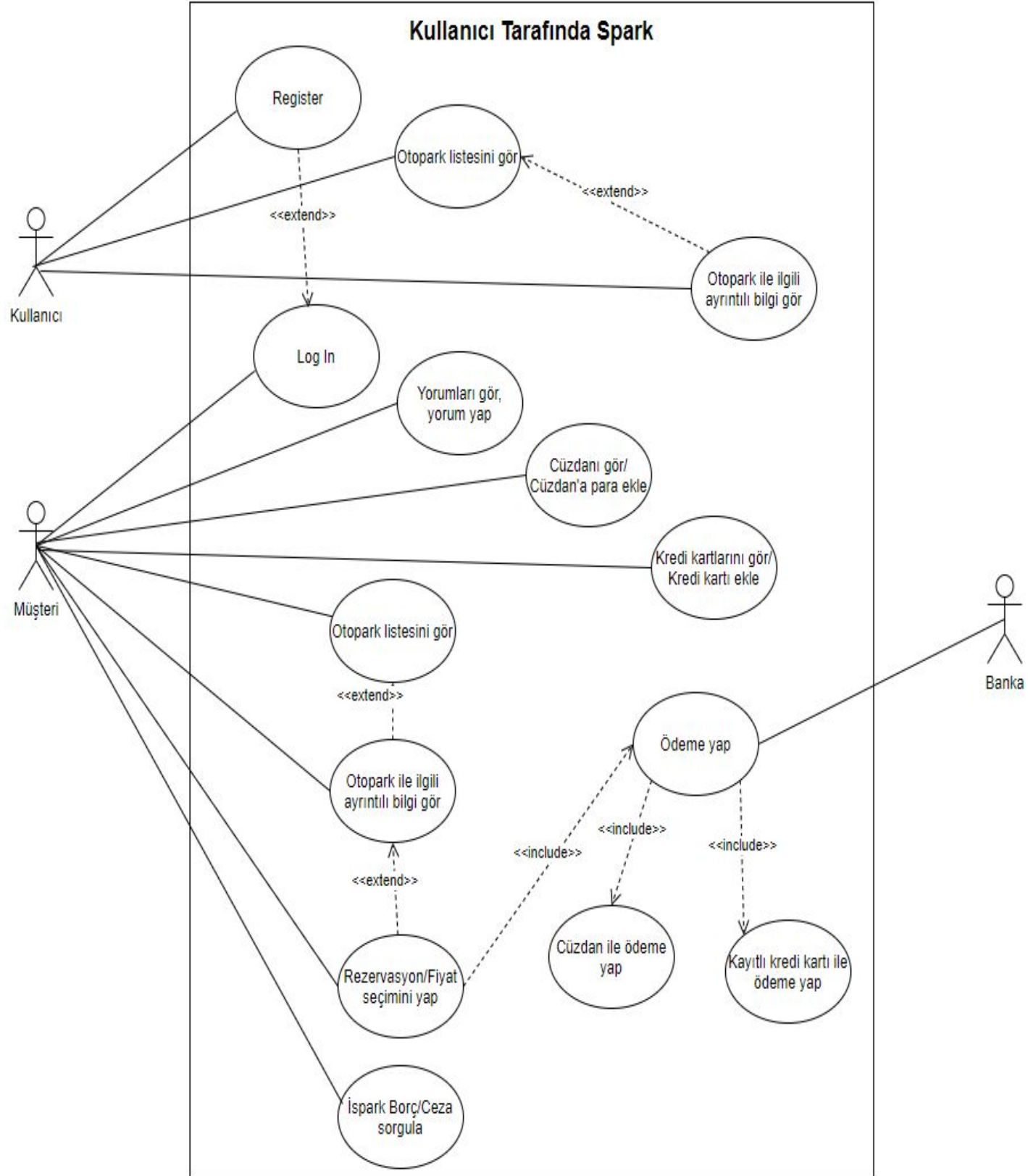
Şekil - 9

5. Kullanıcı Arayüzleri

5.1 Kullanıcı Arayüzlerine Genel Bakış



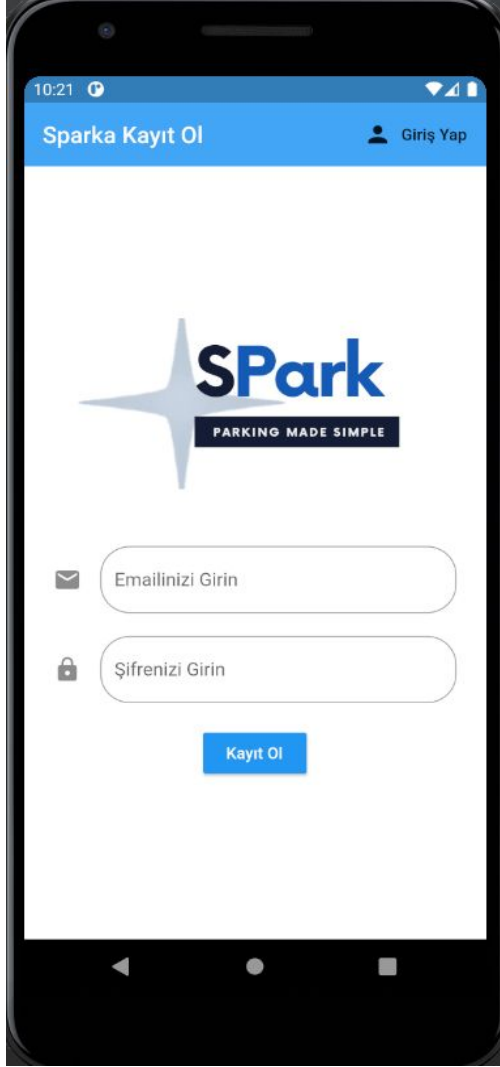
Şekil - 10



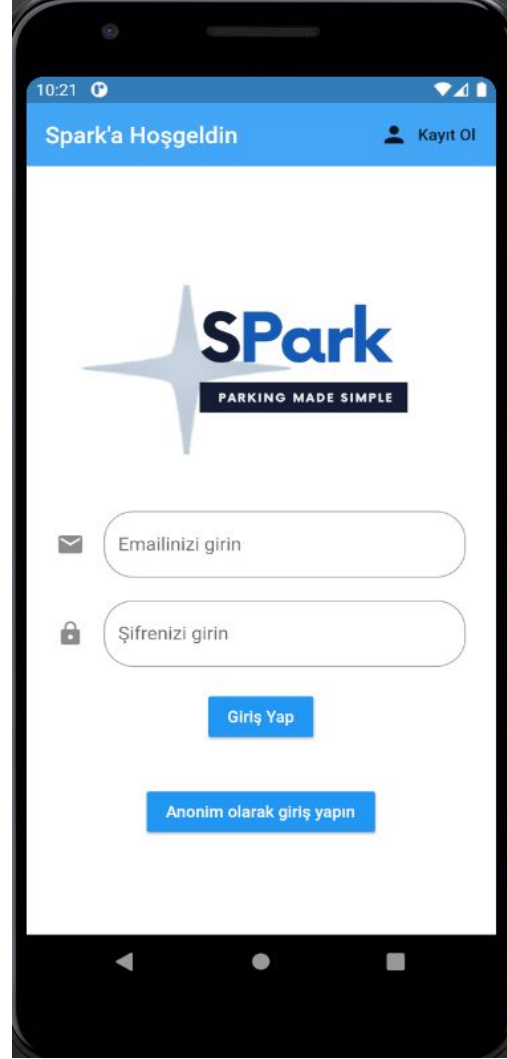
Şekil - 11

5.2 Ekran Resimleri

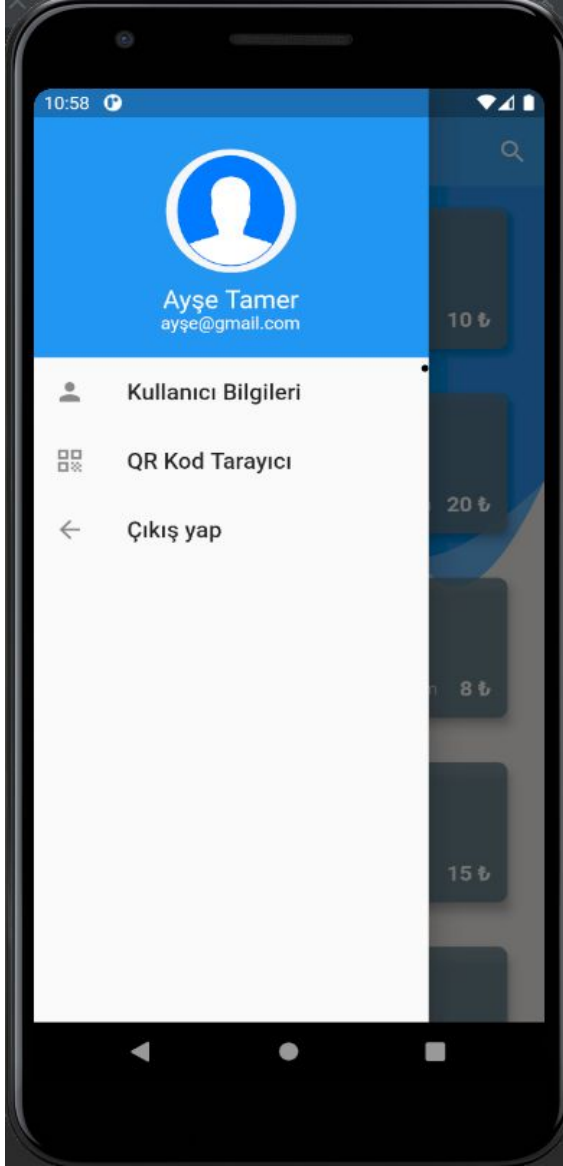
Kullanıcı tarafı (Otopark arayan kişi)



(1) Kayıt Ekranı



(2) Giriş Ekranı



(3)



(4) Kullanıcı Bilgileri Ekranı

7:26

← Kredi Kartınızı Kaydedin

XXXX XXXX XXXX XXXX

MM/YY

CARD HOLDER

Kartınızı kaydetmek için ilerleyin

Card number

Expired Date

CVV

Card Holder

Kredi Kartını Kaydet

(5) Kredi Kartı kayıt ekranı

7:27

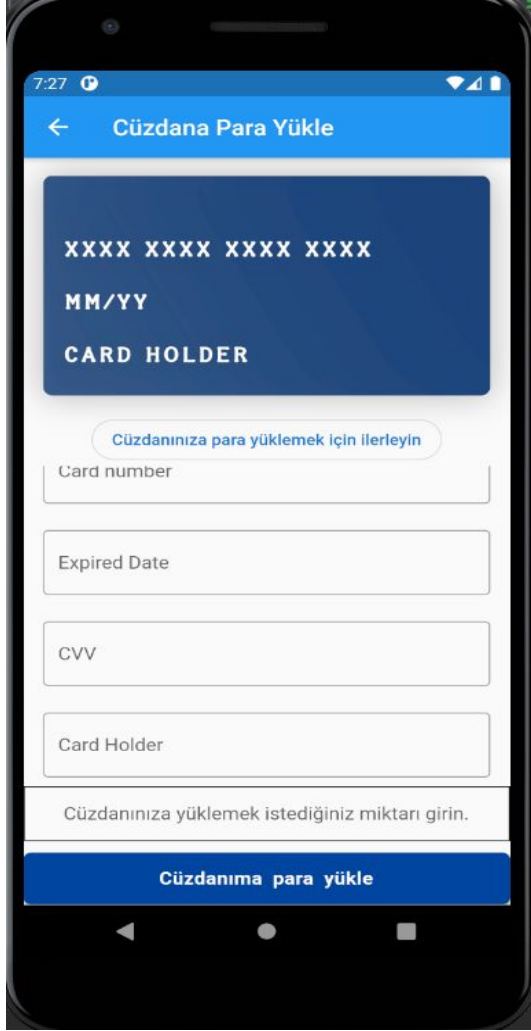
← Kayıtlı Kartlarım

Kayıtlı Kartlarım

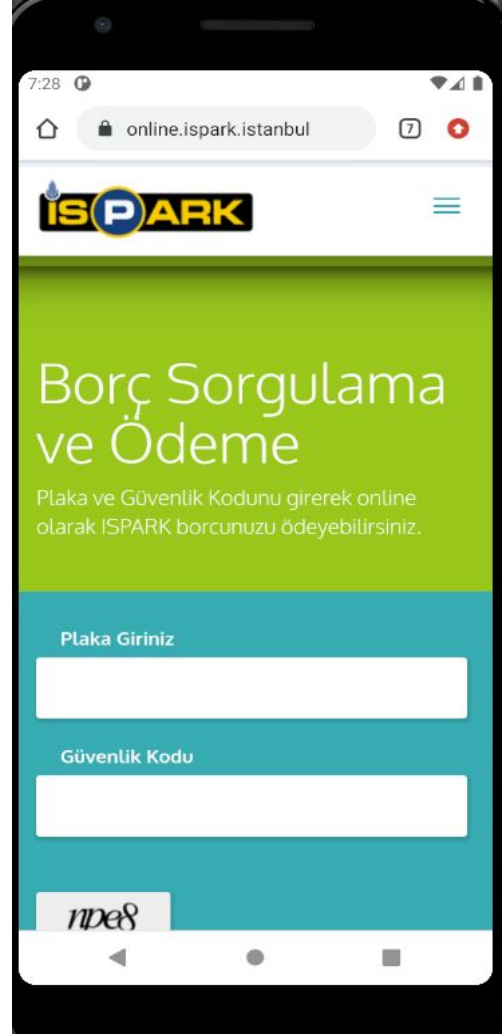
QNB Enpara Kartım 2565 **** *

\$ Varsayılan

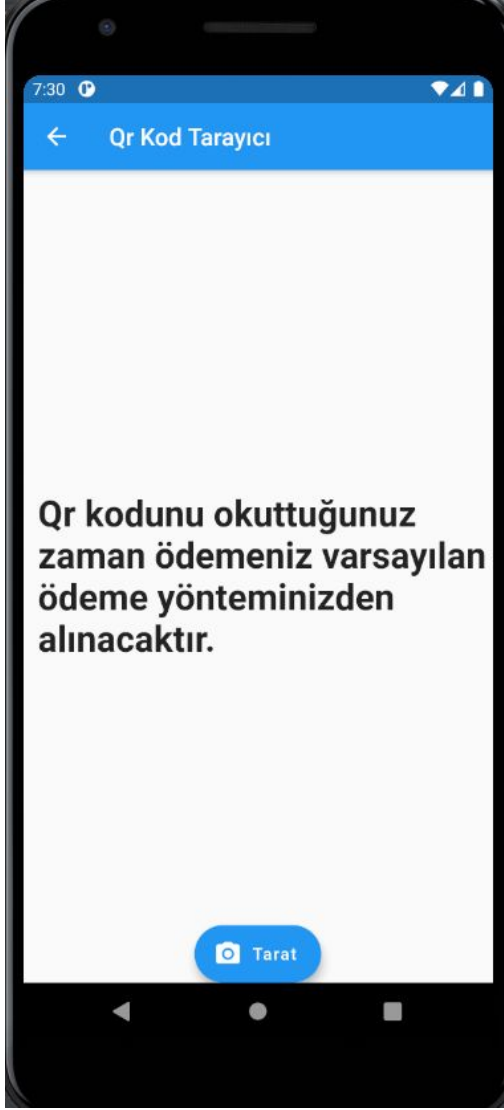
(6) Kayıtlı Kredi Kartı Ekranı



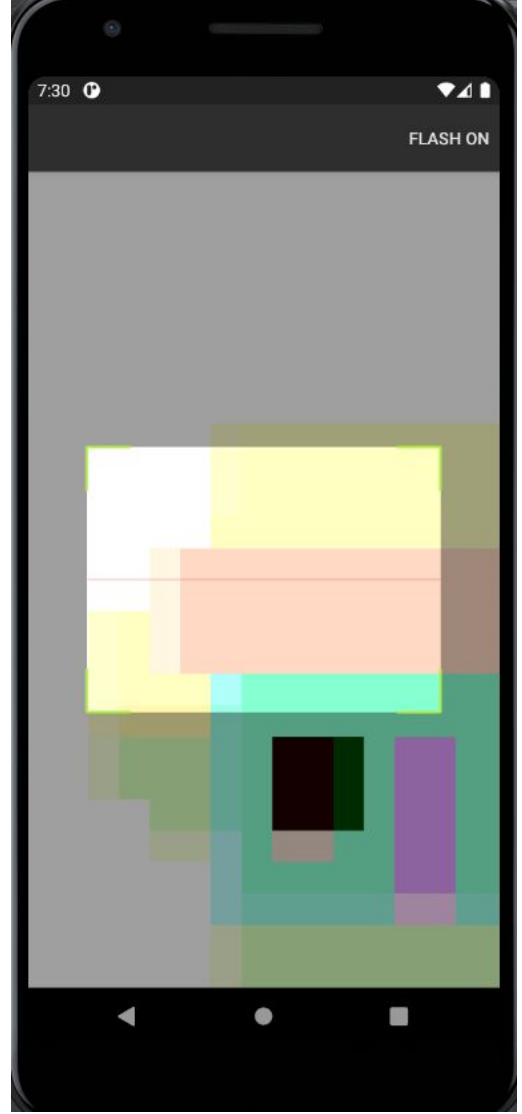
(7) Cüzdan'a para yükleme ekranı



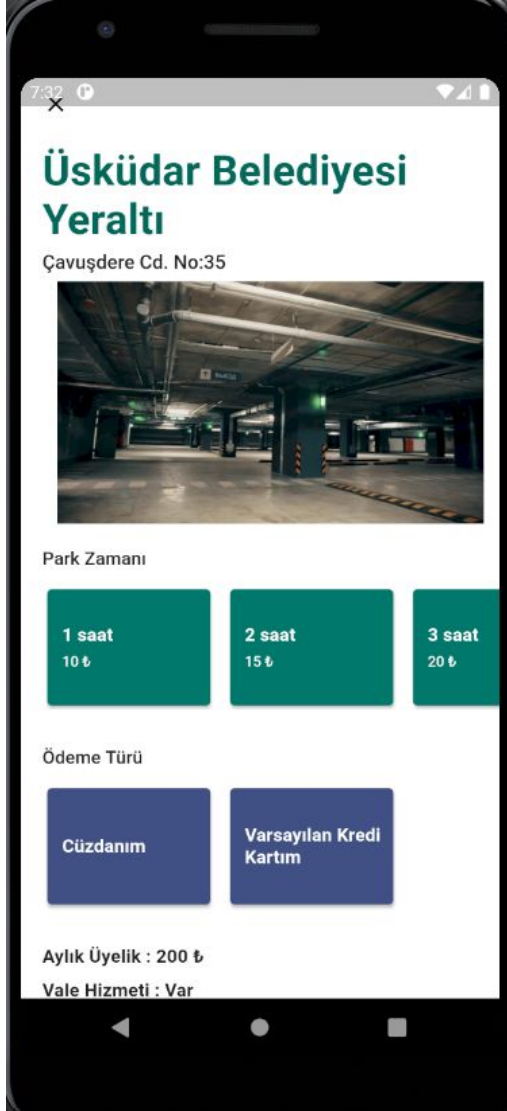
(8) Borç/Ceza sorgusu sonucu yönlendirilen ekran



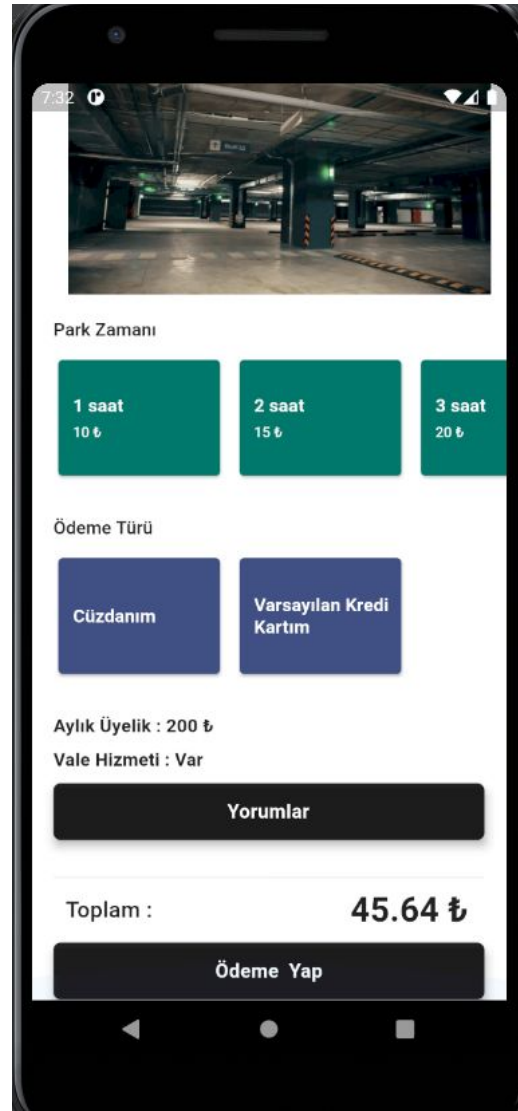
(9) Otoparkta ödeme için QR kod tarama ekranı



(10) QR tarandığında kameranın açılması



(11) Seçilen otopark ile ilgili ayrıntılı bilgi



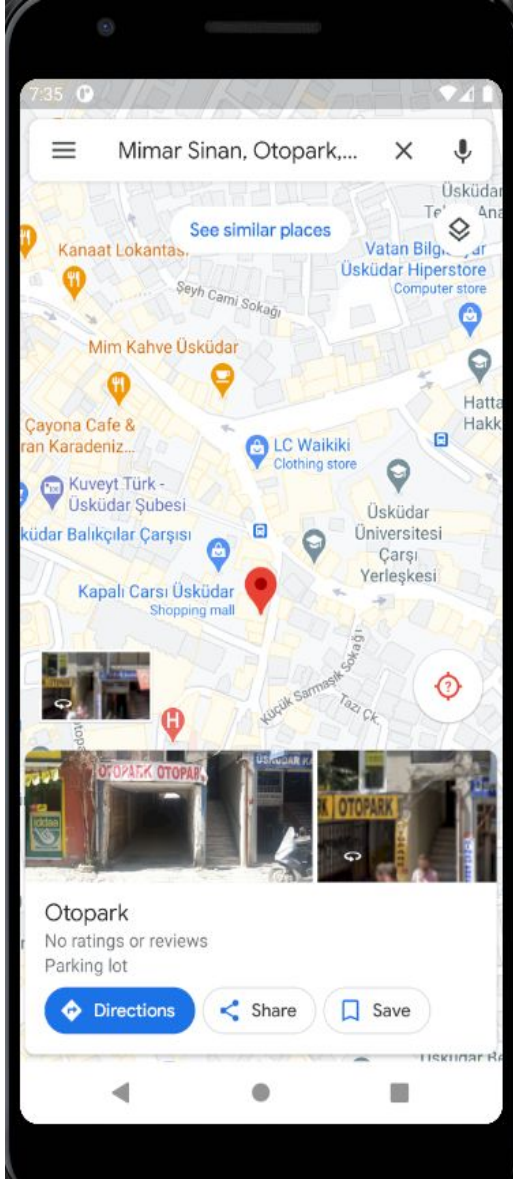
(12) Seçilen otopark ile ilgili ayrıntılı bilgi (devamı)



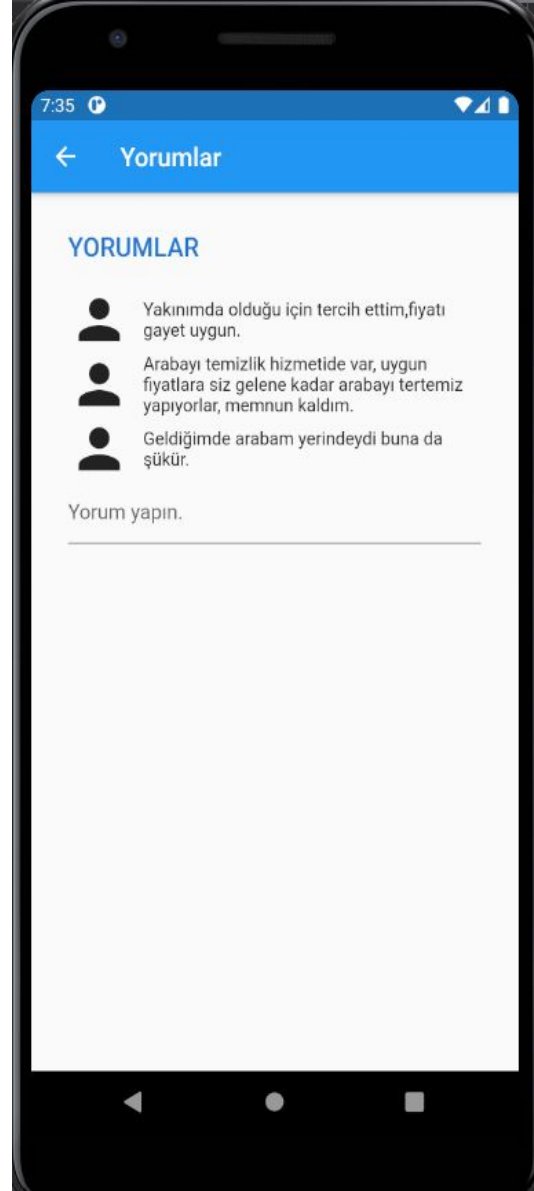
(13) Yorumların görüntülediği ekran



(14) Uygulamadan ödeme yapılırken onay QR'ı, hangi saat aralığında otoparktan yararlanılacağını, otoparkın adresini gösteren ekran

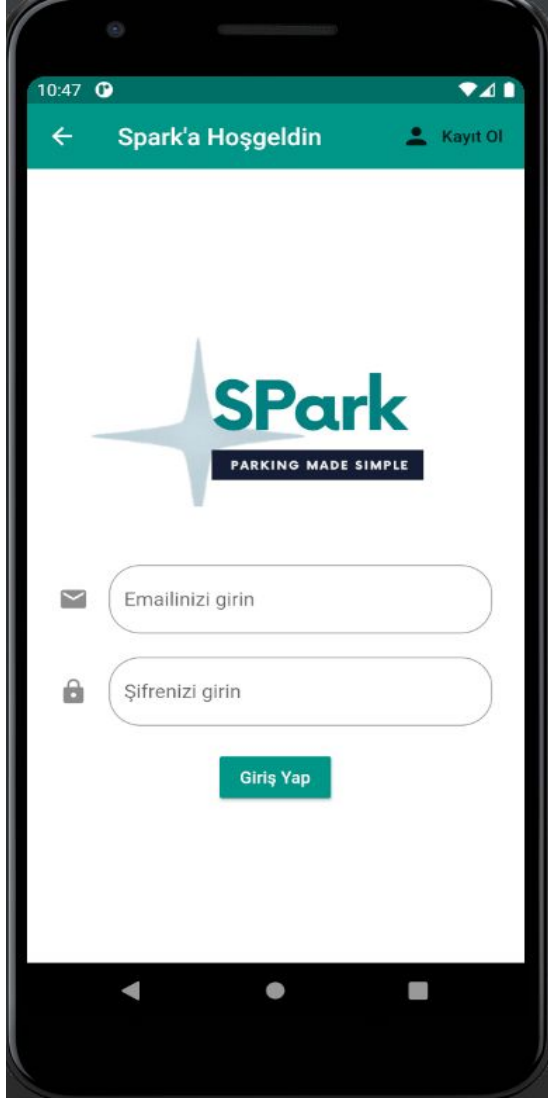


(15) Gidilecek otoparkın adres bilgisini
Google Maps’de gösteren ekran

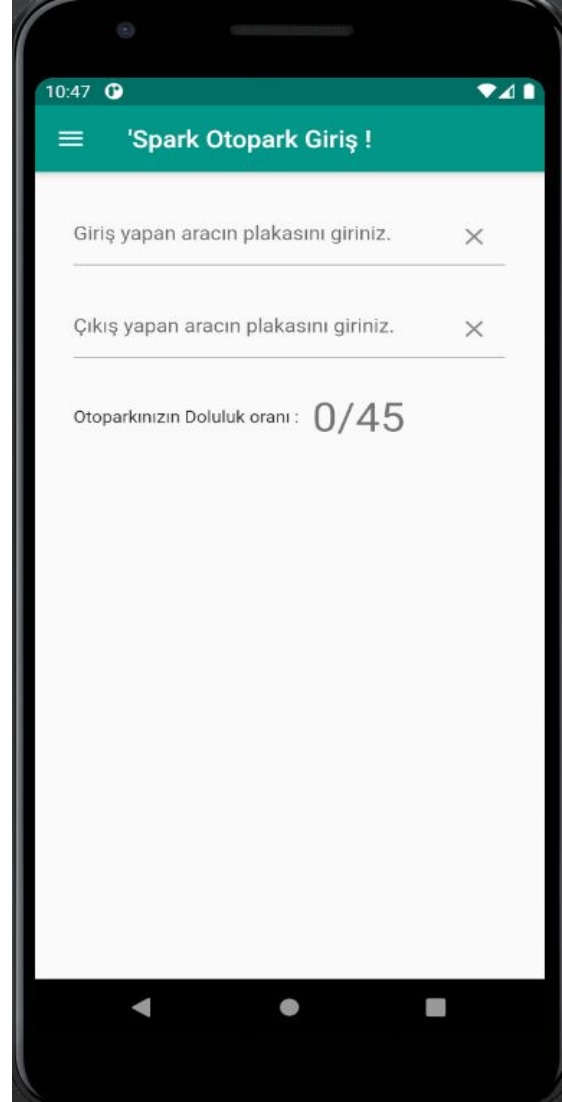


(16) Yorum ekleme ekranı

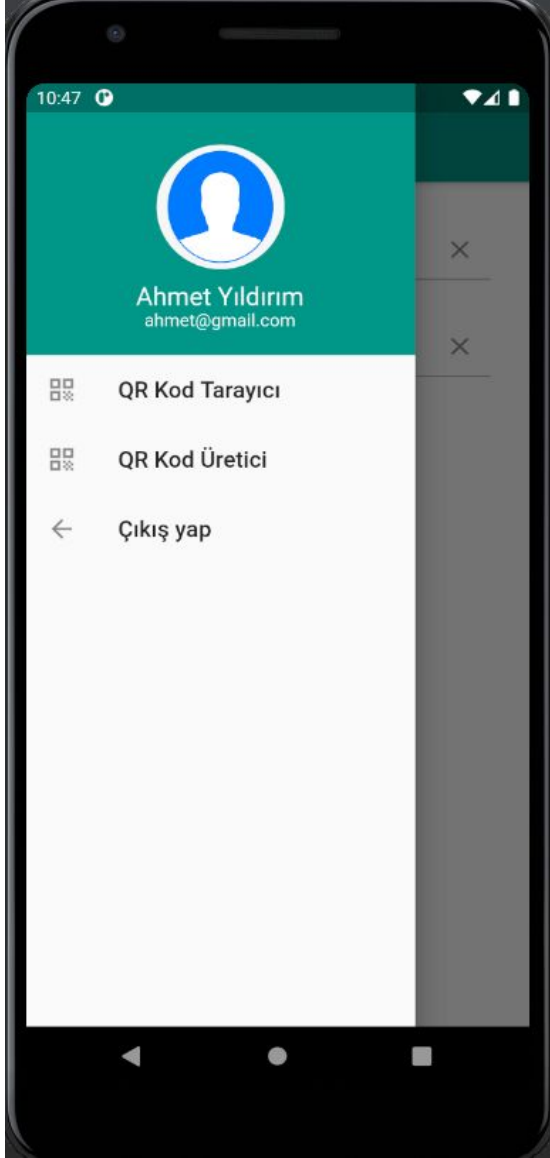
Otoparkçı tarafı



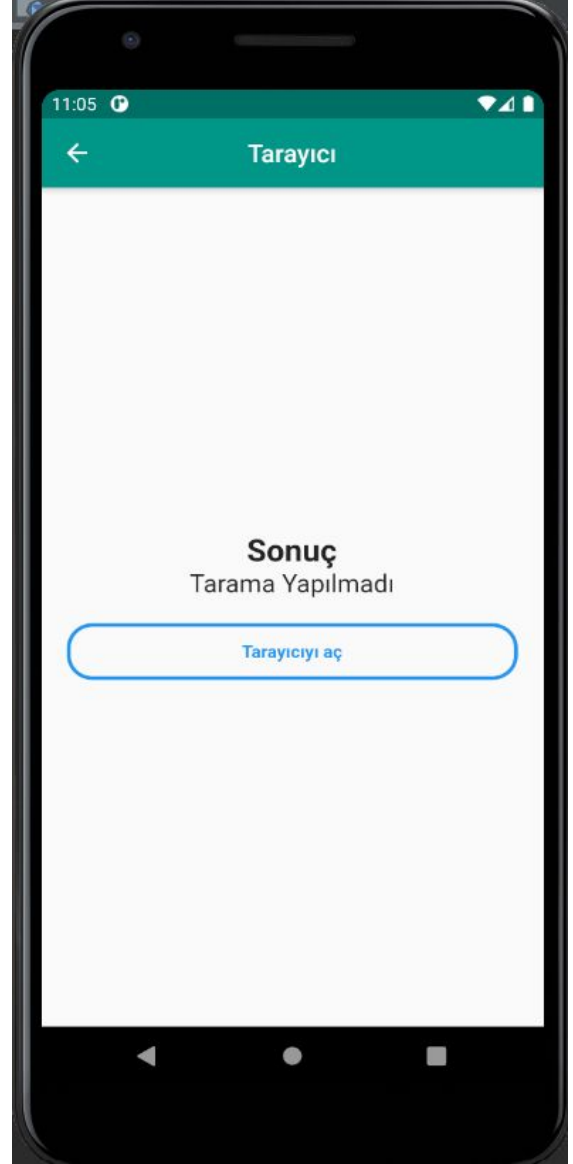
(1) Otoparkçı giriş ekranı



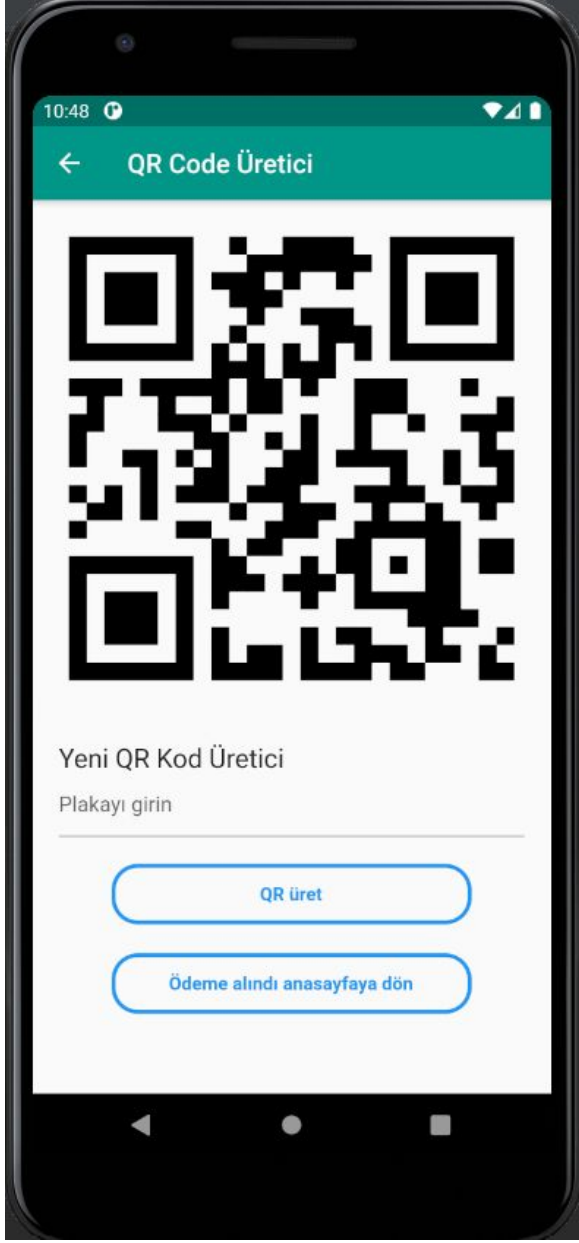
(2) Otoparkçı veri giriş ekranı



(3)



(4) QR kod tarayıcı ekranı



(5) QR kod üretici

6. Kullanılan Kütüphaneler ve Araçlar

Proje gerçekleştirilirken kullanılan geliştirme ortamları, araçları, programlama dilleri aşağıda sıralanmıştır:

- **Android Studio:**

Android Studio, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım geliştirme ortamıdır. Flutter framework'ü ile geliştirme yapabileceğimiz bir IDE'dir.

- **Flutter:**

Flutter, Google tarafından geliştirilen açık kaynak kodlu UI geliştirme yapabileceğimiz bir framework'tür. Tek bir codebase den Android, IOS, Linux, Mac, Windows, Google Fuchsia işletim sistemleri için uygulama çıkarmaya yarar. Uygulama geliştirme sürecinde hot reload özelliği, bütün platformlarda aynı UI ve business logic'in kullanılması, içindeki hazır widget'ları gibi özellikleriyle ön plana çıkmaktadır.

- **Dart:**

Dart, birden çok platformdaki uygulamalar için istemci için optimize edilmiş bir programlama dilidir. Google tarafından geliştirilmiştir ve mobil, masaüstü, sunucu ve web uygulamaları oluşturmak için kullanılır.

Dart'ın C'ye benzer bir syntax'ı vardır. Nesne yönelimli, sınıf tabanlı bir dilidir.

- **JustInMind:**

Uygulama gerçekleştirilmeye başlamadan önce ekran tasarımlarını yapmak amacıyla web ve mobil uygulamalar için bir prototip aracıdır. İçindeki Android telefon seçenekleri, sürükle-bırak işlemi ile tasarımı hızlandırması ve kolaylaştırması yönüyle ön plana çıkmaktadır. Ayrıca, projenin sadece Android telefonlarda değil, ilerleyen süreçte IOS

telefonlarda da gerçekleştirilmesi hedeflenmektedir. JustInMind, IOS telefonlar için de prototip seçenekleri barındırmaktadır.

- **Firestore:**

Firestore, uygulamanın ihtiyacı olduğu ve ürettiği verilerin saklandığı, yönetildiği platformdur. Firestore'in uzaktan yapılandırma özelliği ile, geliştiricilerin uygulamadaki belirli değişiklikleri uzaktan uygulamasına olanak tanır. Bu sayede değişiklikler mevcut sürüme yansıtılır ve kullanıcının en son güncellenmiş sürümü indirmesine gerek kalmaz. Firestore üzerinde gerçek zamanlı veriler senkronize ve performanslı olarak çalışmaktadır.

- **NoSQL:**

NoSQL, veriler için kullanılan veri tabanıdır. NoSQL veritabanları, harika kullanıcı deneyimleri sunulması amacıyla esnek, ölçeklenebilir, yüksek performanslı ve yüksek oranda işlevsel veritabanları gerektiren mobil uygulamalar için idealdir.