

**YALOVA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

- BİTİRME TEZİ-

TARA GELSİN SİPARİŞ SİSTEMİ

Mert NACAR

Bitirme Tezi Danışmanı: Dr. Öğr. Üyesi Murat OKKALIOĞLU

YALOVA, 2020

**YALOVA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

TARA GELSİN SİPARİŞ SİSTEMİ

**Mert NACAR
150101060**

- 1. Bitirme Tezi Danışmanı: Dr. Öğr. Üyesi Murat OKKALIOĞLU**
- 2. Jüri Üyesi :**
- 3. Jüri Üyesi :**

Bitirme Tezinin Dönemi: 2019 – 2020 Bahar Yarıyılı

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ	iii
KISALTMA LİSTESİ	iv
ÖNSÖZ	v
ÖZET	vi
ABSTRACT	vii
ŞEKİL LİSTESİ.....	viii
1. GİRİŞ	1
2. MATERYALLER VE YÖNTEMLER	2
2.1 Veri Tabanı ve PSQL	2
2.2 Visual Studio Code	2
2.3 JavaScript	2
2.4 Node.js	3
2.4.1 Express JS.....	3
2.4.2 Sequelize.....	3
2.5 React Native	4
2.5.1 Redux.....	4
2.5.2 React Redux.....	5
2.5.3 React Navigation.....	5
2.5.4 React Native Elements.....	5
2.5.5 React Native Vector Icons.....	5
2.5.6 React Native Qrcode Scanner.....	5
2.5.7 Twilio.....	6
2.5.8 Cloudinary.....	6
2.5.9 Regular Expression.....	6
3. TARA GELSİN İÇERİĞİ VE KULLANIMI	7
3.1 Veri Tabanı	7
3.1.1 Veri Tabanı Tablo İlişkiler Diyagramı.....	7
3.2 Giriş, Kayıt Olma ve Şifre Kurtarma	8
3.2.1 Giriş.	8

3.2.2 Kayıt Olma.....	9
3.2.3 Şifre Kurtarma.....	10
3.3 Kare Kod ve Profil İşlemleri	11
3.3.1 Kare Kod İşlemleri.....	11
3.3.2 Profil İşlemleri.....	12
3.4 Menü, Mekân ve Sepet İşlemleri	15
3.4.1 Menü İşlemleri.....	15
3.4.2 Mekân İşlemleri.....	16
3.4.3 Sepet İşlemleri.....	17
4. SONUÇLAR VE TARTIŞMA.....	19
KAYNAKLAR	20

SİMGE LİSTESİ

`{ }` : JavaScript Code Scope

KISALTMA LİSTESİ

ORM	: Object Relational Mapping
JSON	: JavaScript Object Notation
PSQL	: Postgre Structured Query Language
RGX	: Regular Expression
RN	: React Native
TG	:Tara Gelsin
JS	: JavaScript

ÖNSÖZ

Bu sipariş sistemi projesi Yalova Üniversitesi Bilgisayar Mühendisliği bölümünde Bilgisayar Mühendisliği Bitirme Tezi olarak hazırlanmıştır. Bu projenin amacı, günümüz çapraz platform mobil uygulama teknolojilerinden biri olan React Native aracı (framework) ile kafe ve restoran gibi masaya sipariş ile çalışan firmaların menü basım ve müşterilerin masada sipariş vermek için masada bekleme süresini indirmektir. Bu projede danışmanlığımı yapan ve her konuda bana yardımcı olan saygıdeğer hocamız Dr. Öğr. Üyesi Murat Okkalıoğlu, okul hayatım boyunca benden yardımını esirgemeyen aileme, arkadaşlarıma ve hocalarıma teşekkür ederim.

Mert NACAR

ÖZET

Günümüzde birçok insan hem sosyalleşmek hem de yakınlarıyla yemek ve içecek tüketimi için kafe ve restoranları tercih etmektedirler. İnsanlar kafe veya restoranlara oturduktan sonra sipariş vermek için garsona ulaşmak ile vakit kaybetmektedirler. Bu durum kafe ve restoranlar için para kaybı, müşteriler için ise zaman kaybına neden olmaktadır. Tara Gelsin sipariş sistemi, bu zaman ve para kaybını ortadan kaldırmayı amaçlayan uygulamalardan biridir. Kullanıcıların kafe veya restoranda masaya geçtiklerinde masanın üzerinde bulunan kare kodu uygulama aracılığı ile okutarak zaman kaybetmeden menüye ve kafe veya restoran bilgilerine erişebilmektedirler. Böylelikle müşteriler garsonları beklemeden siparişlerini geldikleri gibi verebileceklerdir. Müşterilerin hızlı sipariş verebilmeleri ve kafe veya restoranların daha fazla kâr yapmalarını sağlamak amacıyla React Native aracı ile uygulama gerçekleştirilmiştir.

Anahtar Sözcükler: Mobil Uygulama, React Native, Sipariş

ABSTRACT

Today, many people prefer cafes and restaurants both for socializing and eating and beverage consumption with their relatives. People lose their time by reaching the waiter to order after sitting in cafes or restaurants. This situation causes loss of money for cafes and restaurants and time loss for customers. Tara Gelsin ordering system is one of the applications that aims to eliminate this waste of time and money. When users go to the table in the cafe or restaurant, they can access the menu and cafe or restaurant information without wasting time by reading the square code on the table through the application. Thus, customers will be able to place their orders as they come without waiting for waiters. In order to enable customers to order quickly and to make cafes or restaurants to make more profit, the application was implemented with React Native framework.

Key Words: Mobile Application, React Native, Order

ŞEKİL LİSTESİ

Şekil 2.1 Redux şeması.....	4
Şekil 3.1 Veri Tabanı Tablo İlişkileri Diyagramı.....	7
Şekil 3.2 Giriş Yap ekranı	8
Şekil 3.3 Kayıt Ol ekranı	9
Şekil 3.4 Hash işlemleri.....	10
Şekil 3.5 Şifre kurtarma ekranları.....	10
Şekil 3.6 Ana Ekran.....	11
Şekil 3.7 Kare kod tarama Ekranı.....	12
Şekil 3.8 Profil ekranı.....	12
Şekil 3.9 Profili Güncelle ve Şifre Değiştir ekranları.....	13
Şekil 3.10 Ödeme Yöntemleri ve Sipariş Geçmişim ekranları.....	14
Şekil 3.11 Menü, Ürünler ve Ürün Detay ekranları	15
Şekil 3.12 Mekân ve Garson ekranları	16
Şekil 3.13 Sepetim ve Ödeme ekranları	17

1. GİRİŞ

Tara Gelsin (TG) kullanıcıların gittikleri kare kod sisteminin kurulu olduğu mekânlarda garsonları beklemeden direkt olarak kendi cep telefonu ile sipariş verebilmesini sağlayan mobil sipariş uygulamasıdır. Geliştirilen bu uygulama ile kullanıcılar gittikleri mekânın bilgilerine, puanlamasına ve menü hizmetine ulaşabilmektedirler. Seçtikleri ürünleri sepetlerine ekleyip siparişlerini tamamlayabilirler. TG uygulaması çeşitli teknolojilerin harmanlanarak kullanılmasıyla geliştirilmiştir. Uygulamanın arka tarafında (Back-End) JavaScript (JS) dilinin sunucu tarafı çalıştırma ortamı olan Node.js platformu ve veri tabanı için açık kaynak olan Postgre Structured Query Language (PSQL) veri tabanı kullanılmıştır. Uygulamanın istemci tarafında (Front-End) ise Facebook'un geliştirdiği çapraz platform mobil uygulama JS aracı olan React Native (RN) kullanılmıştır. TG uygulaması ilk etapta zorunlu olarak kullanıcıların üye olmalarını veya zaten üyelerse giriş yapmalarını beklemektedir. Aynı zamanda bu ekranda şifremi unuttum özelliği ile yeni şifresini telefon numarası ve e-posta adresini doğru girerek oluşturabilmektedir. Giriş yapmış kullanıcı, isim, e-posta adresi ve telefon numarası gibi profil bilgilerini düzenleyebilir, kredi kart bilgilerini ekleyebilir veya silebilir, şifresini değiştirebilir gibi profil işlemlerine ve kare kod tarama ekranına ulaşabilmektedir. Bir mekâna giden kullanıcı giriş yaptıktan sonra kare kod tarama ekranından masada bulunan kare kodu tarayarak ilgili mekânın bilgilerine, garson çağırma özelliğine ve menüsüne erişim sağlayabilmektedir. Kullanıcının isteğine göre sipariş vermeden sadece garson çağırabilmekte ve ilgili mekânın bilgilerini inceleyebilmektedir. Kullanıcı menülerde yemek veya içecekler için ekstralar ekleyebilir, içindekileri görüntüleyebilir, fotoğrafına erişebilir ve sepetine ekleyebilmektedir. Menü seçimlerini yapan kullanıcı ödeme ekranına gidip sepetinde değişiklik yapabilir veya direkt olarak siparişini tamamlayabilmektedir. Özet olarak TG, kullanıcıların garson beklemeden menüye ulaşip kolayca sipariş verebildikleri bir mobil sipariş sistemidir.

2. MATERYALLER VE YÖNTEMLER

2.1 Veri Tabanı ve PSQL

Veri tabanı bilgilerin birtakım prosedürlere uygun olarak barındırıldığı ortamdır. Veri tabanı bilgileri uygun biçimlerde kaydetmeye, güncellemeye ve bilgiler üzerinde gelişmiş sorgulamalar yapmaya olanak sağlar. Veri tabanı, Veri Tabanı Yönetim Sistemi (Database Management System) denilen yazılımlar aracılığı ile oluşturulur ve yönetilir. Veri tabanı sistemlerinin en temel birimi tablolarıdır [1]. Tablolar bilgilerin tutulduğu nesnelerdir ve her veri tabanında en az bir tablo bulunur. Tablolarda bulunan her bir verinin ayırt edici bir kimlik verisi bulunur. Bu kimlik verileri aracılığıyla (primary key) diğer tabloda eşleştirilmek istenen veriler ile aynı kimlik değeri ile (foreign key) ilişkilendirilir. Bu ilişkilerin bulunduğu veri tabanına ilişkisel veri tabanı denir. İlişkisel veri tabanları birçok mobil uygulama ve web sitesi gibi halka açık platformlarda kullanılmaktadır. Verilerin tekrar edilmeyip referans olarak kullanıldığı ve sadece istenilen bilgilere ulaşmak için ilişkisel veri tabanlarının performansı yüksektir. PSQL, bir ilişkisel veri tabanı örneğidir. Uygulamanın ilişkisel veri tabanı ihtiyacı için PSQL kullanılarak uygulamanın veri tabanı tasarlanmıştır. Açık kaynak kodlu olması ve platform ayırt etmediği için PSQL seçilmiştir.

2.2 Visual Studio Code

Visual Studio Code Microsoft tarafından Windows, Linux ve MacOS için geliştirilen bir kaynak kodu düzenleyicisidir. Hata ayıklama, gömülü Git kontrolü, sözdizimi vurgulama, akıllı kod tamamlama ve kod yeniden yapılandırma desteği içerir. Ayrıca özelleştirilebilir, böylece kullanıcılar editörün temasını, klavye kısa yollarını ve tercihlerini değiştirebilir. Resmi indirme işlemi tescilli bir lisans altında olmasına rağmen, ücretsiz ve açık kaynaktır [2].

2.3 JavaScript

JS, yaygın olarak web tarayıcılarında kullanılmakta olan bir betik dilidir. JS ile yazılan istemci tarafı betikler sayesinde tarayıcının kullanıcıyla etkileşimde bulunması, tarayıcının kontrol edilmesi, asenkron bir şekilde sunucu ile iletişime geçilmesi ve web sayfası içeriğinin değiştirilmesi gibi işlevler sağlanır. JS, Node.js gibi platformlar sayesinde sunucu tarafında da yaygın olarak kullanılmaktadır [3]. Uygulama, hem ön tarafta RN aracı ile hem de arka tarafta Node.js platformunda JS ile yazılmıştır.

2.4 Node.js

Node.js, açık kaynaklı, sunucu tarafında çalışan ve ağ bağlantılı uygulamalar için geliştirilmiş bir çalışma ortamıdır. Node.js uygulamaları genelde istemci taraflı betik dili olan JS kullanılarak geliştirilir. Bu teknolojiler sık sık gerçek zamanlı Web uygulamalarında tercih edilmekle beraber kullanım alanı popülaritesiyle orantılı olarak genişlemiştir. Node.js, Google V8 JS motorunu kullanarak betik dilini yorumlar ve içerisinde standart olarak dağıtılan kütüphaneler sayesinde ek bir sunucu yazılımına gerek kalmadan uygulamanın Web sunucusu görevini görür [4]. Sunucu için node.js platformu üzerinde JS kullanılarak sunucu tarafı oluşturulmuştur. Sunucuda, istemciden gelen istekleri yönetmek, cevap vermek ve veri tabanı işlemleri gerçekleştirmek ve ilerleyişi kolaylaştırmak için bazı kütüphanelere ihtiyaç duyulmaktadır.

2.4.1 Express JS

Express JS, Web ve Mobil uygulamalar için sağlam özellikler sağlayan sade ve esnek bir Node.js web uygulama çatısıdır. Sınırsız HTTP yardımcı araç ve katmanlar sayesinde sağlam bir API oluşturmak hızlı ve kolaydır [5]. İlk başta API terimini kavramalıyız. Uygulama programlama arayüzü (application programming interface, API), yoğunlukla web tabanlı uygulamalarda istemci ve sunucu arasındaki iletişimi sağlayan bir sözleşme olarak kullanılmaktadır. Express-JS bize sunucumuzun istekleri yönetmek, istekler doğrultusunda belirli işlemleri gerçekleştirmek ve istemci tarafına cevap vermemizi sağlar.

2.4.2 Sequelize

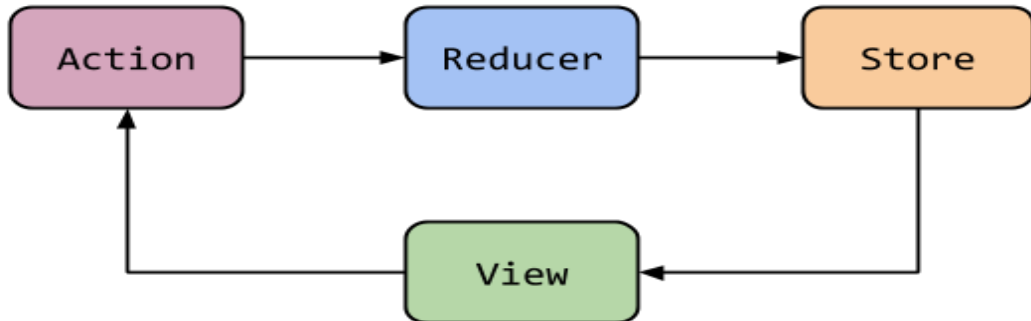
Object Relational Mapping (nesne ilişkisel eşleştirme, ORM), nesne tabanlı programlama dilleri yapısına uygun olmayan katı veri tabanı sorguları yerine veri tabanı yapısının bir nesne gibi düşünülerek yazılım geliştirilmesine olanak sağlayan programlama teknolojisidir. Temel olarak klasik veri sorguları ile uygulama arasında bir katman oluşturarak, veri yapılarının bir nesne gibi kontrol edilmesini sağlar [6]. Sequelize ORM, Node.js sunucumuz ile PSQl veri tabanımız arasında köprü görevi görerek veri tabanındaki bilgilerimizi yönetmek için sunucu tarafında nesne modelleri kullanmamıza olanak sağlar. Sunucu taraflı oluşturduğumuz bu nesnelerle veri tabanında bilgileri getirme ve eş zamanlı olarak ekleme, silme ve güncelleme işlemlerimizi gerçekleştirmemizi sağlar.

2.5 React Native

RN, Facebook tarafından oluşturulan açık kaynaklı bir mobil uygulama aracıdır. Geliştiricilerin JS web araçlarından olan React aracının yerel platform yetenekleriyle birlikte kullanmasını sağlayarak Android, iOS ve Windows ve benzeri platformlar için uygulamalar geliştirmede kullanılır. RN kullanıcının göreceği görünüm kısmı için JSX söz dizimini kullanır. Bu söz dizimi Extensible Markup Language (XML) / Hyper Text Markup Language (HTML) söz dizimine benzer yapıda kullanılır. Her görünüm için tek bir View etiketli görünümle sarmalanması gerekmektedir. View etiketi ise RN için bir görünüm açmaktır. TG'nin ön tarafı yani mobil uygulama tarafı için RN kullanılmıştır. Görünüm için JSX söz diziminden ve mantıksal işlemler için ise JS betik dili kullanılarak kodlanmıştır. Yeniden kullanılabilir kodlar için RN her yeniden kullanılacak parça için bileşen (component) oluşturmayı öneriyor. TG kodlama yapılırken uygulamanın her bir parçası azami düzeyde bileşenlerine ayrıştırılıp yeniden kullanılabilir hale getirilmiştir. Mobil uygulama için veri yönetimi, navigasyon, kod tarama, ikonlar ve stiller için çeşitli RN kütüphanelerinden faydalanılmıştır.

2.5.1 Redux

Redux, uygulama durum yönetimi için açık kaynak bir JS kütüphanesidir. Sadece RN için geliştirilmemiştir. Birçok başka teknolojiyle birlikte kullanılabilir. RN'nin her bir bileşenin kendi içinde durumları (state) vardır. İç içe geçmiş bileşenler çoğaldıkça durum yönetimi yapmak bir o kadar zorlaşır. Redux bu kısımda devreye girerek küresel durum (store) oluşturarak tüm bileşenlerin direkt olarak ulaşmasını sağlar. Güncellenen bu küresel durum görünüme yansıtılır. Action ile ne yapılacağı, reducer ile nasıl yapılacağı belirlenip store güncellenir. Redux için akış tek yönlüdür. Redux şeması Şekil 2.1'de görülmektedir.



Şekil 2.1 Redux şeması

2.5.2 React Redux

React Redux kütüphanesi de bir JS kütüphanesi olup RN ile Redux kütüphanesinin birleşmesinde rol oynar. React Redux kütüphanesinin Provider etiketi, RN'nin kök görünüm olan View etiketini sarmalayarak Redux'un store objesini parametre olarak birleştirmeyi gerçekleştirir.

2.5.3 React Navigation

React Navigation, RN aracı için mobil uygulama ekranları arasında animasyonlu dolaşım, navigasyon bar oluşturma, bileşenler arası durumların aktarımı ve navigasyon geçmişi gibi genel olarak kullanıcının ekranlar arası geçiş yapmasını sağlayan bir JS kütüphanesidir. Uygulamanın navigasyon kısmında React Navigation'ın çeşitli modüllerinden yararlanılarak mobil uygulama içi navigasyon sağlandı.

2.5.4 React Native Elements

React Native Elements, RN aracı için girdiler, butonlar ve listeler gibi çeşitli kullanıcı arayüz elemanlarının stillerini bileşenler oluşturarak çeşitli varyasyonlarda hazır olarak sunan bir JS kütüphanesidir. TG'nin mobil uygulama kısmında butonların hazır stilleri, listeler, girdiler ve ürün kartları gibi kullanıcı arayüz bileşenlerinde React Native Elements kütüphanesinden yararlanılmıştır.

2.5.5 React Native Vector Icons

React Native Vector Icons, RN aracı için mobil platformda çeşitli ikonların çeşitli stiller ile gösterilmesini sağlayan bir JS kütüphanesidir. TG'nin mobil uygulama kısmında navigasyon bar kısmında, profil etkileşimlerinde, butonlarda ve kullanıcının bilgi gireceği girdilerde daha anlaşılır olması için ikonlar kullanılmıştır. Bu ikonlar kullanılırken React Native Vector Icons kütüphanesinden yararlanılmıştır.

2.5.6 React Native Qrcode Scanner

React Native Qrcode Scanner, RN aracında kare kod taraması yapılabilmesi için kullanılan bir JS Kütüphanesidir. TG'nin mobil uygulamasında kare kod taraması yapıp ilgili mekânın menü ekranına geçiş yapmak için React Native Qrcode Scanner kütüphanesi kullanılmaktadır.

2.5.7 Twilio

Twilio, bir bulut iletişim hizmeti veren şirkettir. Yazılım geliştiriciler Twilio API kullanarak programlı olarak telefon görüşmesi yapmaları, mesaj göndermeleri ve onay mesaj sistemi gibi birçok özelliği uygulamalarında kullanabilmektedirler. TG'nin mobil uygulama kısmında şifremi unuttum ve kaydolma işlemleri sırasında telefon doğrulaması için onay mesajı Twilio API kullanarak sağlanmaktadır. Kullanıcı telefon numarasını girdikten sonra istek TG'nin Node.js sunucusuna gitmekte ve sunucudan twilio API ile iletişim kurmaktadır. Onay mesajı hem kullanıcının telefonuna hem de TG'nin sunucusuna ulaşır. Kullanıcı onay kodunu tekrar girdiğinde tekrar sunucuya onay mesajıyla istek atıldığında doğrulama sunucuya, sunucu ise Twilio API ile iletişim kurarak istemci tarafına onay sonucunu dönmektedir.

2.5.8 Cloudinary

Cloudinary, bulut tabanlı video ve fotoğraf yönetimi için yazılım hizmeti sağlayan bir şirkettir. Dosyalama ve çeşitli fotoğraf kırpma gibi işlemleri kolayca sağlamaktadır. TG'nin her bir mekânı için ayrı bir dosya açılıp her bir ürünün fotoğraflarının ayrı tutularak her mekâna kendi ürününün fotoğrafını koyma özgürlüğü sağlanmıştır. Her bir mekânın içinde ise kategoriler dosyalara ayrılarak ürünler ayrı bir şekilde tutulmaktadır. TG'nin mobil uygulama kısmında menü ekranında bulunan kategori ve ürünlerin fotoğrafları Cloudinary'nin sunmuş olduğu bağlantılar ile direkt olarak sergilenmektedir.

2.5.9 Regular Expression

Regular Expression (Kurallı ifadeler, RGX), sisteme sızma veya zarar verme amacıyla olan insanlardan ve sunucu tarafına gitmeden girdilerin kontrolünü sağlamak amacıyla kullanılır. TG'nin hem mobil uygulama hem sunucu kısmında RGX yapısından faydalanarak performans ve korunma sağlanmaktadır. Telefon numarası, e-posta ve şifre gibi kullanıcı bilgileri RGX ile doğruluğu sağlanarak sunucu tarafına gönderilmekte ve sunucu tarafında da RGX kontrolü başka yollarla direkt sunucuya istek atılmasından korunmak için kontrolü tüm girdiler için sağlanmaktadır.

Şekil 3.1 Veri Tabanı Tablo İlişkileri Diyagramı

3.2 Giriş, Kayıt Olma ve Şifre Kurtarma

3.2.1 Giriş

Kullanıcıyı ilk olarak önceden giriş yapıp yapmadığını telefonun hafızasından (cache) gerekli bilgileri alarak kontrolü sağlanır. Eğer giriş yapmışsa uygulamanın ana ekranına giriş yapmamış ise Şekil 3.2’de görülmekte olan giriş yap ekranına yönlendirilecektir. Kullanıcının TG uygulamasına kayıtlı bir hesabı var ise telefon numarası ve şifre bilgilerini doldurarak giriş yap butonuyla ana ekrana geçebilmektedir. Uygun telefon numarası ve şifre girmediğinde kullanıcı formun üst tarafında uyarılar almaktadır. Her bir kullanıcı giriş yaptığından itibaren hangi kullanıcının sunucu tarafına istekte bulunduğu ve uygulama güvenliği için JSON web token (JWT) ayırt edici ve isteğe bağlı sonlu anahtar oluşturulur. Giriş yaptıktan sonra her sunucu isteğinde bu anahtar sunucuya gönderilerek kontrol edilir. Böylelikle hangi kullanıcının istekte bulunduğu öğrenilip güvenlik de sağlanmaktadır. Kullanıcı redux yapısı kullanılarak verileri giriş yapmış olduğu sürece saklanmaktadır.

Şekil 3.2 Giriş Yap ekranı

3.2.2 Kayıt Olma

Eğer kullanıcının kayıtlı bir hesabı yok ise Şekil 3.2’de Kayıt Ol bağlantısı ile Şekil 3.3’te gözükmekte olan Kayıt Ol ekranına gidebilmektedir. Kullanıcı Zaten üye misin? adlı bağlantı ile Giriş Yap ekranına geri dönebilmektedir. Kayıt Ol ekranında bulunan isim, soy isim, telefon numarası şifre, e-posta, kullanım koşulları ve KVKK veri işlenmesini eksiksiz doldurması ve onaylaması gerekmektedir. Kullanıcıdan alınan bu bilgiler hiçbir şekilde başka kullanıcılar ile paylaşılmamaktadır. Kullanıcının şifresi SHA-512 algoritmasına göre karıştırma (hash) işlemini sunucu tarafında Şekil 3.4’te 1 numaralı görüntüde rastgele iterasyon sayısı oluşturulup hash işlemi gerçekleştirilip veri tabanına eklenir, 2 numaralı görüntüde ise kullanıcı giriş yaparken şifre kontrol sağlanır. Kısaca SHA-512 algoritması kullanıcının girdiği yalın veriyi anlaşılması güç bir formata dönüştürür. Kayıtlanan kullanıcı giriş yap ekranına yönlendirilmekte ve uygulamaya giriş yapması istenmektedir.

Şekil 3.3 Kayıt Ol ekranı

```

// Rastgele olarak hash şifreye eklemek için tuz(salt) değeri oluşturulur.
crypto.randomBytes(byte, (err, salt) => {
  if (err) {
    return reject(err);
  }

  let saltHex = salt.toString("hex");

  // Kayıt için şifre hash işlemi geçirir.
  crypto.pbkdf2(password, saltHex, iteration, byte, digest, (err, hash) => {
    if (err) {
      return reject(err);
    }
    // tuz (salt), hashlenmiş şifre ve iterasyon değeri veri tabanı için formatlanır.
    let hashHEX = hash.toString("hex");
    userPassword = `${iteration}:${hashHEX}:${saltHex}`;
    // formatlanmış veri döndürülür.
    return resolve(userPassword);
  });
});

```

```

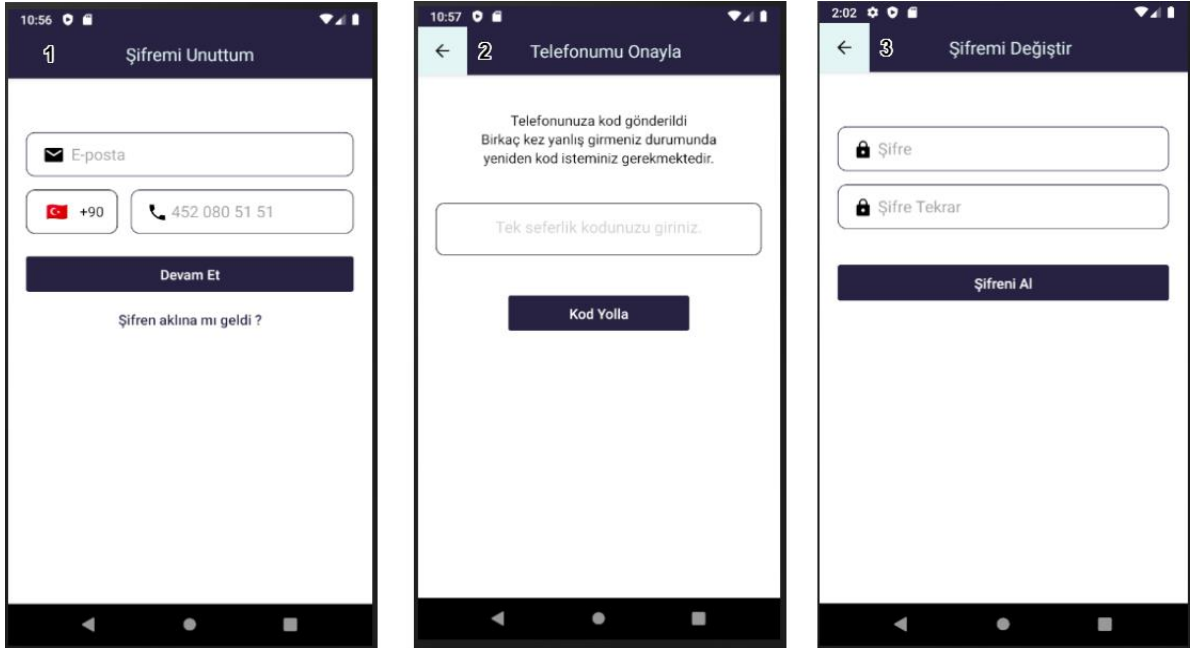
// Girilen şifrenin hashlenerek kontrol edilmesi
crypto.pbkdf2(password, salt, iteration, byte, digest, (err, verify) => {
  if (err) {
    return reject(err);
  } else {
    // giriş için doğru(true) veya yanlış(false) değer döner
    return resolve(verify.toString("hex") === hash);
  }
});

```

Şekil 3.4 Hash işlemleri

3.2.3 Şifre Kurtarma

Eğer kullanıcı şifresini hatırlayamadığı için giriş yapamıyorsa giriş ekranında bulunan şifremi unuttum butonu yardımıyla Şekil 3.5'te gözüken 1 numaralı şifremi unuttum ekranına yönlendirilmektedir. Eşleşmeyen veya uygun formatta olmayan e-posta veya telefon numarası girildiğinde ilgili hata mesajı formun üstünde çıkmaktadır. Şifremi unuttum 1 numaralı ekranda doğru olarak eşleşen e-posta ve telefon numarasını girdikten sonra Şekil 3.5'te gözüken 2 numaralı Telefonumu Onayla ekranına yönlendirilir. Telefonumu Onayla ekranında kod yolla butonu ile kullanıcının telefonuna kod yollanır. Uygun kod girilmez ise hata mesajı gösterilir eğer uygun kod girilmişse Şekil 3.5'te gözüken 3 numaralı yeni şifre oluşturma ekranına gidilerek kullanıcının yeni şifre oluşturması beklenmektedir. Oluşturulan yeni şifre veri tabanına eski şifre yerine kaydedilmektedir.

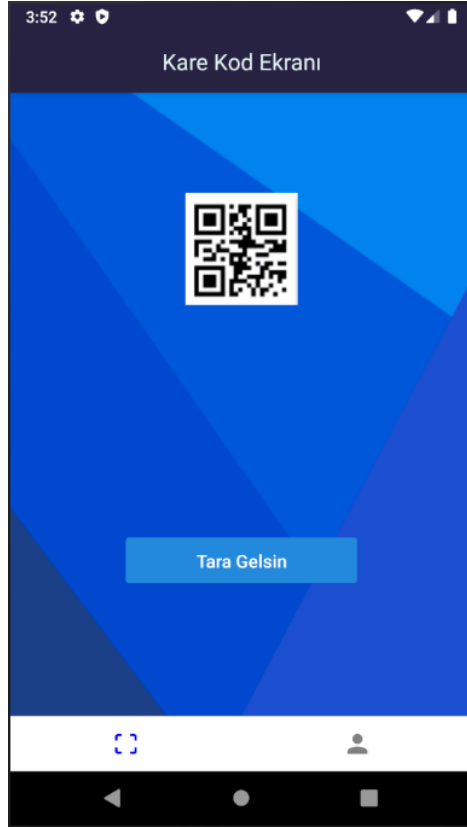


Şekil 3.5 Şifre kurtarma ekranları

3.3 Kare Kod ve Profil İşlemleri

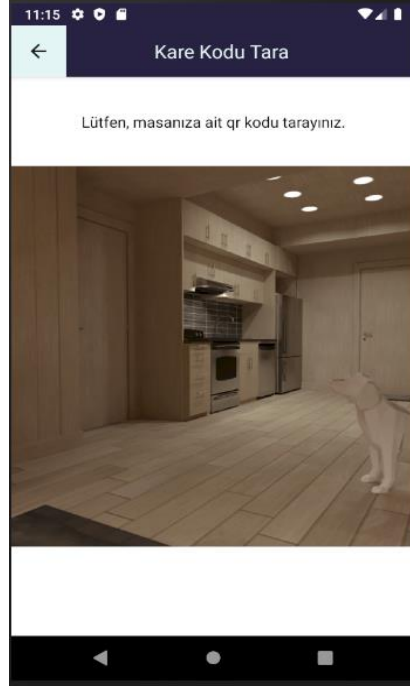
3.3.1 Kare Kod İşlemleri

Giriş işlemi gerçekleşmiş olan kullanıcıyı Şekil 3.6’te gözüken iki seçenekli mobil navigasyon barlı ana ekran karşılamaktadır. Eğer kullanıcı mobil uygulama içindeki herhangi çıkış yap butonunu kullanmadan uygulamayı kapatırsa geri uygulamaya girdiğinde yeniden kullanıcı adı ve şifresini girmeden direkt olarak otomatik giriş sağlanarak Şekil 3.6’te gözüken ana ekrana yönlendirilecektir.



Şekil 3.6 Ana Ekran

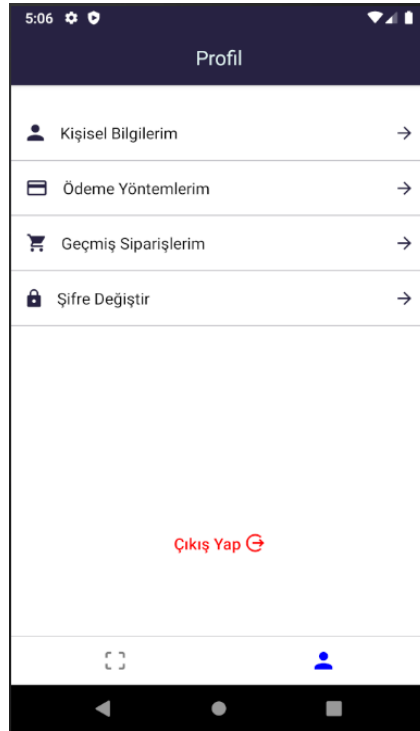
Kullanıcı eğer bir mekâna gidip masaya geçtiyse masada bulunan TG kare kodunu mobil uygulamayı yoluyla kullanması gerekmektedir. Masada bulunan kodu ise Tara Gelsin adlı butona basarak Şekil 3.7’de gözüken kare kod tarama ekranına geçiş yapacaktır. Mobil uygulamanın kare kod taraması için kamera izninin mobil uygulamaya verilmiş olması gerekmektedir. Kamera izni verilen mobil uygulamada Şekil 3.7’de gözüken kamera kısmı açılmakta ve kullanıcının masada bulunan kare kodu okutarak ilgili mekânın ana ekranına yönlendirme yapılmaktadır.



Şekil 3.7 Kare kod tarama Ekranı

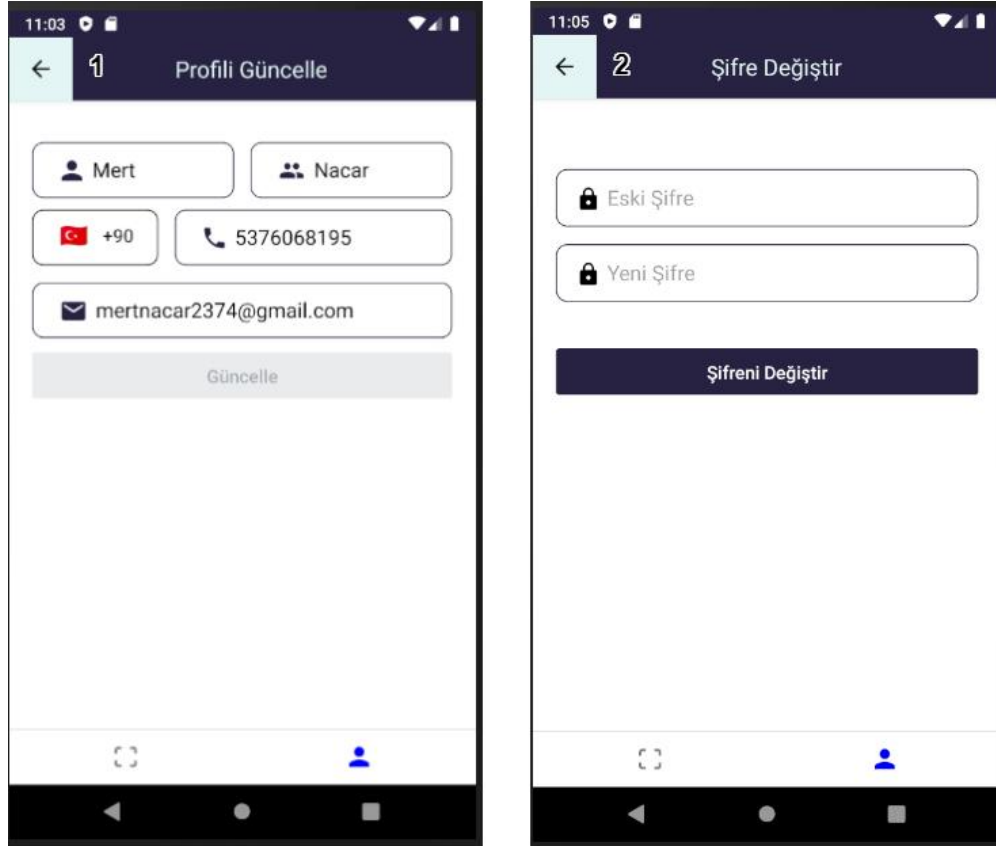
3.3.2 Profil İşlemleri

Kullanıcı Şekil 3.6’de gözüken ana ekran üzerinde alt navigasyonda sağda bulunan insan simgesine tıklandığında ise Şekil 3.8’de gözüken profil ekranı karşılamaktadır. Kullanıcı Çıkış Yap adlı butona basarak hesabından çıkarak Giriş Yap ekranına geçebilmektedir.



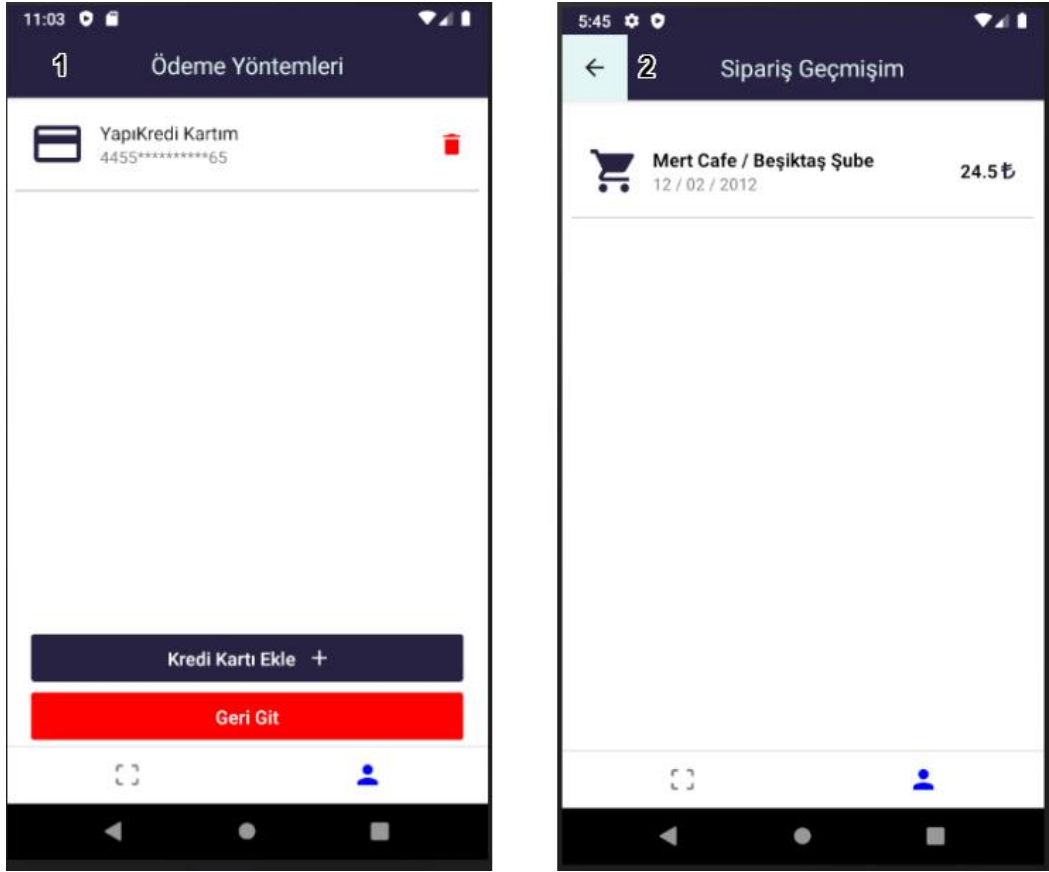
Şekil 3.8 Profil ekranı

Kullanıcı profil ekranı üzerinden isim, soy isim, e-posta, şifre ve telefon numarası gibi bilgileri düzenleyebilmekte, tüm kredi kartı bilgilerini görüntüleyip silme işlemi uygulayabilmekte ve kredi kartı ekleme işlemi yapabilmekte, şifre değişimini ayrı ekrandan yapabilmekte ve geçmiş verdiği siparişleri mekân ve şube ismi, sipariş verilme tarihi ve tutarını görüntüleyebilmektedir. Kullanıcı profil ekranından Kişisel Bilgilerim bağlantısına basarak Şekil 3.9’de gözüken Profili Güncelle ve Şifre Değiştir ekranlarından 1 numaralı Profili Güncelle ekranına geçiş sağlayabilmektedir. 1 numaralı ekran üzerinden isim, soy isim, telefon numarası ve e-posta bilgilerini uygun bilgiler girerek düzenleyebilmektedir. Kullanıcı aynı şekilde profil ekranından Şifre Değiştir bağlantısına basarak Şekil 3.9’de gözüken Profili Güncelle ve Şifre Değiştir ekranlarından 2 numaralı Şifre Değiştir ekranına geçiş sağlayabilmektedir. 2 numaralı ekran üzerinden kendisine yeni şifre sağlayabilmektedir. Böylelikle kullanıcıya bu ekranlarda tüm bilgilerini istediği gibi değiştirebilme olanağı sağlanmaktadır. Şifre değişikliği hash işlemi yapılarak değiştirilmekte ve diğer bilgilerinde düzenlenmesinde JWT anahtarı kullanılarak sağlanır ve veri tabanında ilgili yerler değiştirilmektedir.



Şekil 3.9 Profili Güncelle ve Şifre Değiştir ekranları

Kullanıcı Şekil 3.8’de gözüken Ödeme Yöntemlerim bağlantısına basarak Şekil 3.10’da gözüken Ödeme Yöntemlerim ve Sipariş Geçmişim ekranlarından 1 numaralı Ödeme yöntemlerim ekranına geçiş yapılabilmekte ve kredi kartlarını direkt olarak görüntüleme, Kredi Kartı Ekle adlı butona basarak kredi kartı ekleme ve çıkarılmak istenen ilgili kredi kartının kırmızı çöp kutusu ikonuna basarak silme işlemlerini gerçekleştirebilmektedir Kullanıcı Şekil 3.8’de gözüken Geçmiş Siparişlerim bağlantısına basarak Şekil 3.10’da gözüken Ödeme Yöntemlerim ve Sipariş Geçmişim ekranlarından 2 numaralı Sipariş Geçmişim ekranına geçiş yapılabilmekte ve yapılan önceki siparişlerin hangi mekân ve şubede, siparişin oluşturulma tarihi ile sipariş tutarı tüm siparişler için gözükmektedir. Üstten ekranın çekilmesi ile siparişler için tekrar istek sunucuya atılır ve güncel siparişler gelmektedir.

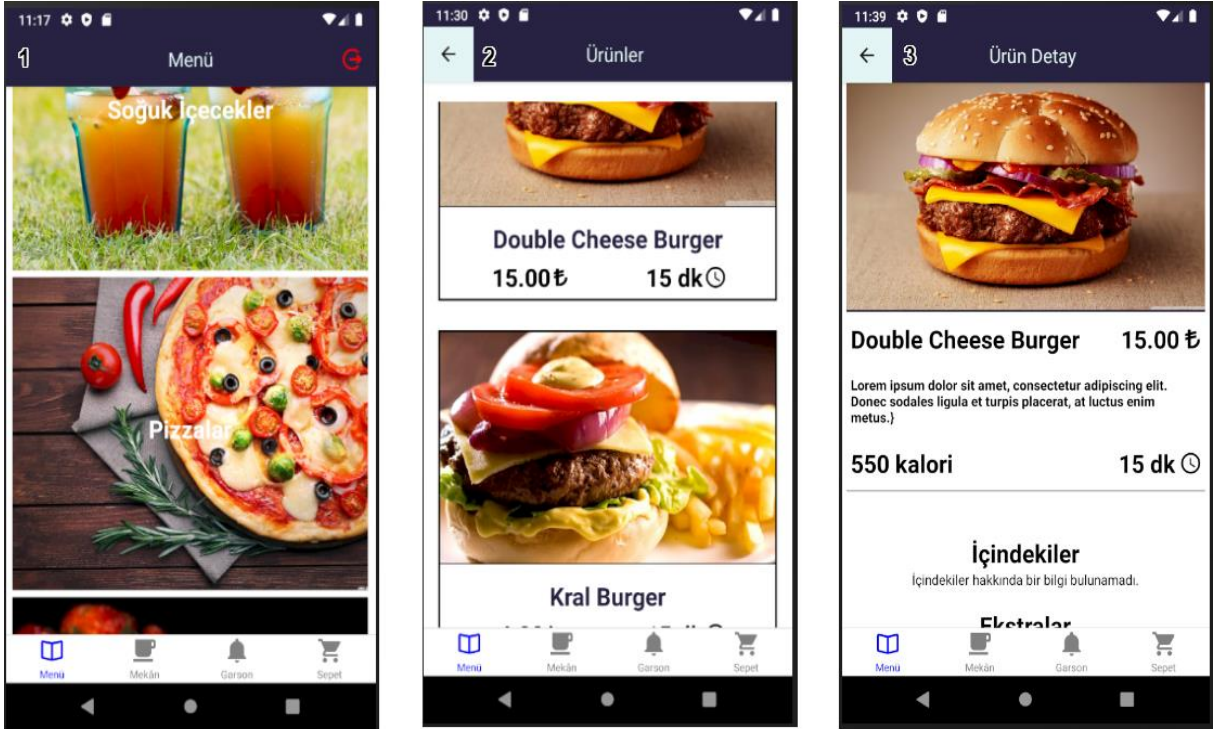


Şekil 3.10 Ödeme Yöntemleri ve Sipariş Geçmişim ekranları

3.4 Menü, Mekân ve Sepet İşlemleri

3.4.1 Menü İşlemleri

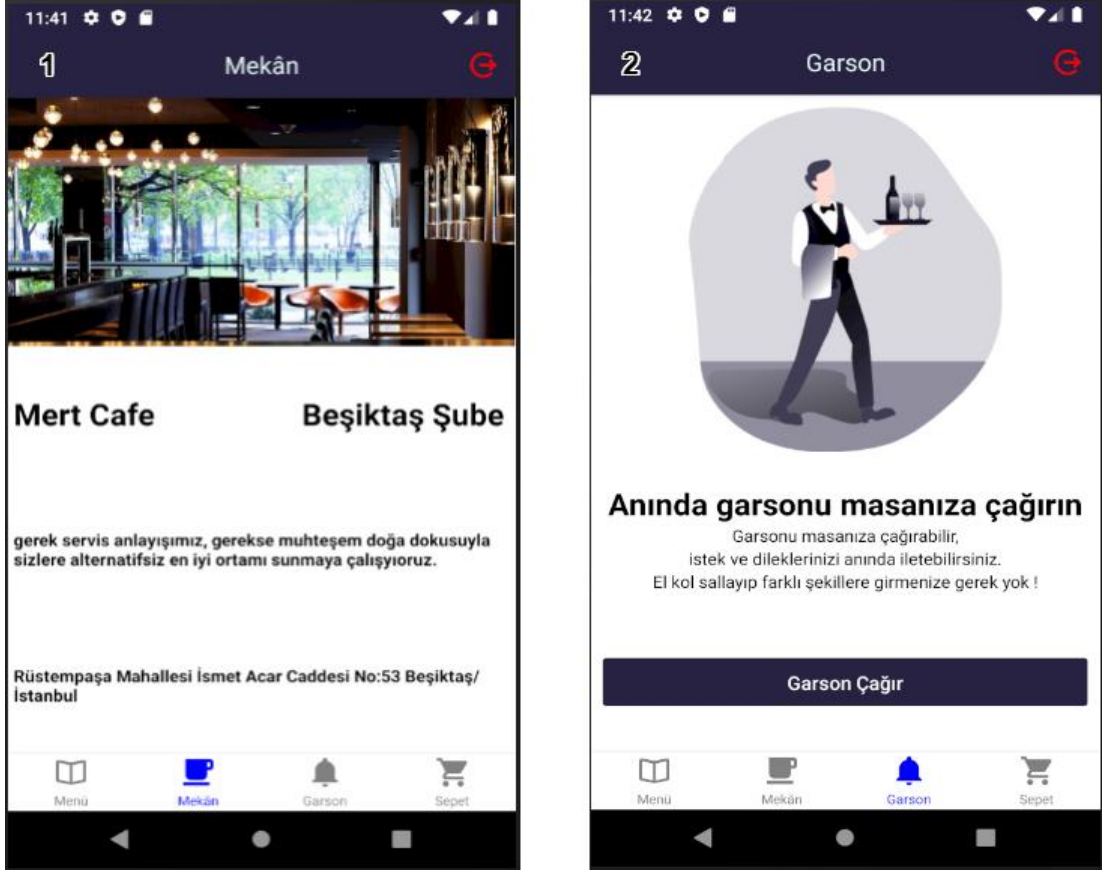
Kullanıcı Şekil 3.7’da ilgili mekânda geçilmiş olan masanın üzerindeki kare kod taratıldıktan sonra ilk olarak Şekil 3.11’da gözüken Menü, Ürünler ve Ürün Detay ekranlarından 1 numaralı olan Menü ekranı karşılamaktadır. Kare kod bilgisi ise mekânın, mekân bölümünün ve bölümün masa kimliklerini taşımaktadır. Böylelikle ilgili mekân, bölüm ve masa eşleşmesi doğru olarak sağlanıp kullanıcıya geçiş sağlanmaktadır. İlgili mekânın menüsü Menü ekranında çıktıktan sonra kullanıcı menünün kategorileri arasından istediğini seçerek Şekil 3.11’da gözüken 2 numaralı Ürünler ekranına geçişi sunucu tarafında mekân bilgisi verilerek ilgili kategoriye göre gelmektedir. Ürünler sayfasında gelen kategorilerin içindeki ürünlerin fiyatı ve hazırlanma süresi kullanıcılara belirtilmektedir. Kullanıcı Ürünler ekranından istediği ürünü seçerek Şekil 3.11’da gözüken 3 numaralı Ürün Detay ekranına geçiş yapmaktadır. Geçiş yapılan Ürün Detay ekranında ise kullanıcının seçmiş olduğu ürün bilgileri sunucu tarafından alınıp ürün ismi, fiyatı, açıklaması, kalori değeri, hazırlanma süresi, içindekiler ve ekstra seçim listesi bilgileri gözükmektedir. Kullanıcı isteğe bağlı olarak eğer ürünün ekstraları varsa ekleyebilmekte ve ürün adetini ayarlayabilmektedir. Ürün ile ilgili ayarlamalarını bitiren kullanıcı ürünü sepetine ekleyebilmektedir. Sepet verileri ise redux yapısı kullanılarak mekân ekranından çıkana kadar tutulmaktadır.



Şekil 3.11 Menü, Ürünler ve Ürün Detay ekranları

3.4.2 Mekân İşlemleri

Kullanıcı Şekil 3.11’de gözüken herhangi bir ekrandayken altta bulunan navigasyonu kullanarak diğer ekranlara geçiş sağlayabilmektedir. Alt navigasyonun ikinci Simgesi olan Mekân ikonuna bastığında ise Şekil 3.12’de gözüken ekranlardan 1 numaralı olan Mekân ekranına geçiş sağlamaktadır. Mekân ekranı üzerinde gidilen mekânın ve şubesinin ismi, mekân açıklaması ve adres bilgileri kullanıcı tarafına sunulmaktadır. Bu ekran ile kullanıcı gidilen mekân doğruluğunu buradan kontrol edebilmektedir.

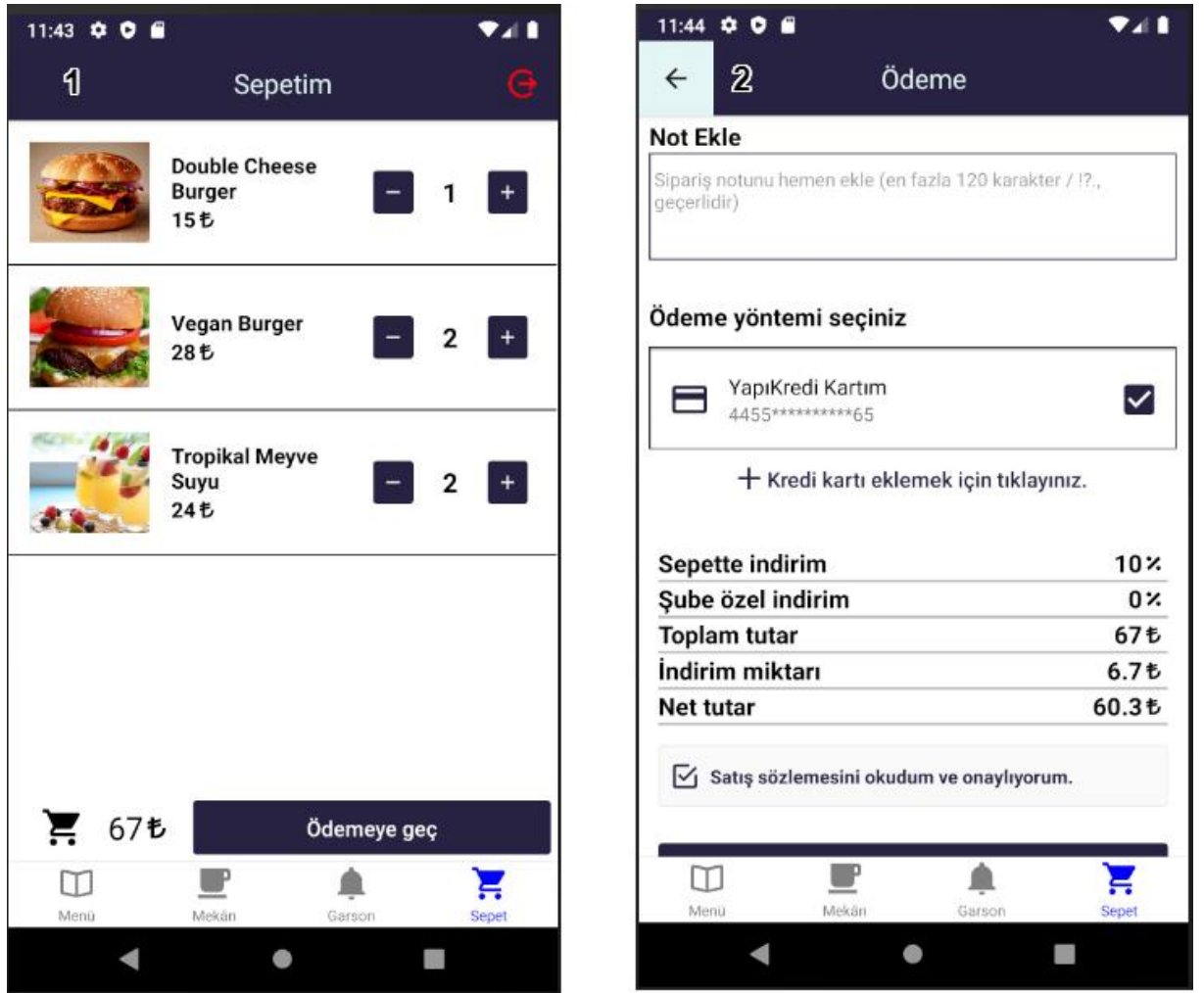


Şekil 3.12 Mekân ve Garson ekranları

Ayrıca kullanıcı menü, mekân veya sepet ekranlarından herhangi birinde sağ üstte bulunan çıkış butonuna basarak Şekil 3.6’te bulunan ana ekrana geri dönebilmektedir. Kullanıcı Şekil 3.11’de gözüken herhangi bir ekrandan veya Şekil 3.12’de gözüken 1 numaralı Menü ekranından altta bulunan navigasyonun üçüncü sırasında zil ikonuna bastığında Şekil 3.12’de gözüken 2 numaralı Garson Ekranına geçiş sağlayabilmektedir. Garson ekranı üzerinden kullanıcı sipariş harici bir şey istemek ve masasını sildirmek istediğinde Garson Çağır butonuna basarak sağlamaktadır. Butona basıldığında ise mekânın yönetim paneline ilgili masanın garson çağırıldığı gözükmektedir. Bu bilgiyi alan garson ise ilgili masaya gidebilmektedir.

3.4.3 Sepet İşlemleri

Kullanıcı Şekil 3.11 ve Şekil 3.12’de gözüken herhangi bir ekrandayken alt navigasyondan en sağda gözüken alışveriş arabası ikonuna bastığından Şekil 3.13’te gözüken 1 numaralı Sepetim ekranına geçiş sağlamaktadır. Eğer kullanıcı sepetine ürün eklememişse sepetiniz boş diye uyarı mesajı ve görsel gösterilecektir. Sepetine ürünlerini eklemiş olan kullanıcı sepet ekranına geldiğinde eklenmiş bu ürünlerin her biri için artı ve eksi ikonları ile adetlerini arttırabilir, azaltabilir ve adeti 1 olan ürünü azaltınca silebilmektedir. Ürünlerin üstüne direkt olarak bastığında ise ürünlerin detay içeriğini göstermek için ekran boyutundan daha küçük bir açılır kapanır pencere belirmektedir.



Şekil 3.13 Sepetim ve Ödeme ekranları

Bu pencere üzerinde seçilmiş ürünün fotoğrafı, içeriği, ekstraları ve adet üzerinden fiyatı gösterilmektedir. Kullanıcı bu pencerenin dışında bir yere basarsa ürün detay penceresi kapanmakta ve sepet ekranı tamamıyla gözükmemektedir. Kullanıcı Sepetim ekranının sol aşağısında bulunan alışveriş sepet ikonunun yanında bulunan fiyat toplam sepet miktarını göstermektedir. Kullanıcı Ödemeye geç butonuna bastığında Şekil 3.13'te gözüken 2 numaralı Ödeme ekranına geçiş sağlayabilmektedir. Kullanıcı ödeme ekranındayken alt navigasyonu kullanarak ürün ekleme, mekân ekranına gidebilme ve garson ekranına gidip garson çağırabilme işlemlerine devam edebilmektedir. Sepetim veya Ödeme ekranında iken sağ üst tarafta bulunan çıkış ikonundan Şekil 3.6'te bulunan ana ekrana geri dönebilmektedir. Kullanıcı Ödeme ekranında sipariş ile ilgili sipariş notu eklemek isterse Not Ekle kısmından ekleyebilmektedir. Kullanıcının kredi kartı tanımlı değil ise kredi kartı eklemek için tıklayınız kısmından kredi kartı ekleyebilir veya hali hazırda kredi kartı varsa bir adet kredi kartı seçmek zorundadır. Kullanıcı sepet indirim yüzdesi, şubenin özel indirim yüzdesi, toplam tutar, indirim miktarı ve net tutarını ödeme yöntemi seçiniz kısmının altında görebilmektedir. Son olarak satış sözleşmesini onaylayan kullanıcı tüm bilgilerinin kontrolünü sağlayarak Sipariş Ver butonu ile siparişini tamamlayabilmektedir. Kullanıcı sipariş verdikten sonra sepetin içindeki son siparişe ait ürünler silinmekte ve kullanıcı Şekil 3.6'da bulunan ana ekrana yönlendirmesi sağlanmaktadır. Ana ekrana yönlendirilen kullanıcı Tara Gelsin butonuyla Şekil 3.7'da bulunan ekrandan tekrar kare kod taratıp Şekil 3.11'da gözüken 1 numaralı Menü ekranına geçiş sağlayabilmektedir. Kullanıcı verdiği siparişi ana ekran üzerinden profil simgesine basarak Şekil 3.8'de bulunan ekrana ardından Geçmiş Siparişlerim bağlantısına basarak Şekil 3.10'da gözüken 2 numaralı Sipariş Geçmişim ekranına ulaşabilmektedir. Kullanıcının verdiği tüm siparişler bu ekran üzerinden gösterilmektedir.

4. SONUÇLAR VE TARTIŞMA

Bu tezde amaç günümüzde kafe ve restoran gibi mekânlarda sipariş verirken uzun beklemelemler sonunu getirmek ve mobil uygulama dünyasına yeni bir sipariş uygulaması sunmaktır. Kullanıcılar mekânların çoğunda garson beklemek, garsona kendini göstermek ve menü adettinin yetmemesi gibi durumların ortadan kaldıracak bir mobil uygulama geliştirmiş olduk. Ayrıca Facebook teknolojilerinin günümüz yazılım dünyasındaki rolünün yadsınamayacak durumda olduğunu farkına varmış olduk. Uygulama yoluyla kullanıcıdan aldığımız hassas bilgilerin JS kodlarıyla sunucu tarafında nasıl hash işlemi geçirip korunduğunu, regex yöntemiyle hem istemci hem sunucu tarafında veri kontrolü sağlandığı ve kullanıcıların sistemde dolaştıkları süre içinde JWT yardımı ile hangi kullanıcıların hangi isteklerde bulunduğunu sağlayarak sistem güvenliği fazlasıyla sağlandığını görmüş olduk. Sequelize ORM ile hem sunucu ve veri tabanı arasında kolayca bağlama hem de veri tabanı sorgularını sunucu tarafında kolayca çalıştırılabildiğini böylelikle ORM faydalarını görmüş olduk. İstemci tarafında React Native aracı ve sunucu tarafında Node.js platformunda sadece JS kullanarak yapılan uygulamada JS'in gücünü de böylelikle görmüş olduk. Projeye ek bazı fikirlerle daha da genişletme sağlanabilir. Örnek olarak kullanıcıdan adres bilgisi de alınarak kullanıcının eve sipariş vermesi sağlanabilir.

KAYNAKLAR

- [1] <https://www.teknologweb.com/veritabani-nedir>, Eriřim Tarihi: 2 Mayıs 2020.
- [2] https://tr.wikipedia.org/wiki/Visual_Studio_Code, Eriřim Tarihi: 2 Mayıs 2020.
- [3] <https://gelecegiyazanlar.turkcell.com.tr/konu/web-programlama/egitim/301-javascript/javascript-nedir>, Eriřim Tarihi: 2 Mayıs 2020.
- [4] <https://tr.wikipedia.org/wiki/Node.js>, Eriřim Tarihi: 3 Mayıs 2020.
- [5] <https://expressjs.com/tr/>, Eriřim Tarihi: 3 Mayıs 2020.
- [6] https://tr.wikipedia.org/wiki/Nesne-%C4%B0li%C5%9Fkisel_E%C5%9Fleme, Eriřim Tarihi: 3 Mayıs 2020