

**BAŞKENT ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ  
BİTİRME PROJESİ RAPORU**

**AFİYET : Sipariş Verme ve Hesap Ödeme Uygulaması**

**Mehmet Umut İnceer - 21995537  
Öncüm Korkmaz Yılmaz - 21997287  
Emin Umut Gerçek - 21997548  
Barış İlkinci - 21995510  
Umut Çiloğlu - 21994740**

<b>Bölümü: Bilgisayar Mühendisliği</b>	
<b>Proje Danışmanı: Dr. Öğr. Üyesi Didem ÖLÇER</b>	
<b>Ders Kodu ve Adı: BIL494 Bitirme Projesi II</b>	
<b>Proje Başlangıcı: 2022-2023 Güz</b>	<b>Proje Süresi (Yarıyıl): 2/2</b>
<b>Rapor Sunumu: 2022-2023 Bahar</b>	

## **ONAY SAYFASI**

Bu Rapor, 15/05/2023 tarihinde aşağıda üye adları yazılı jüri tarafından kabul edilmiştir.

Unvan

Dr. Öğr. Üyesi

Adı Soyadı

Didem ÖLÇER (Danışman)

İmza

Prof. Dr.

Nizami GASİLOV

Doç. Dr.

Mustafa SERT

## **TEŞEKKÜR**

Tüm eğitim sürecimizin yanında, bitirme projesinde de bize destek olduğu ve bu proje sürecinde şu ana kadar verdiği destekten ötürü sayın Dr. Öğr. Üyesi Didem Ölcer'e teşekkür ederiz. Ayrıca bizlere verdiği desteklerden dolayı bölümümüzün tüm öğretim üyeleri ve görevlilerine de teşekkürlerimizi iletiriz.

Afiyet Ekibi

15/05/2023

## ÖZ

Proje, günümüzde farklı alanlardaki birçok farklı sürecin otonom ve personelsiz hale gelmesinden esinlenmektedir. Menü inceleme ve sipariş verme süreçlerinin müşteri tarafından garsonsuz yapılabilmesi olanağının tanınması projenin ana unsurunu oluşturmaktadır. Proje, müşteriler açısından bakıldığından bir restoranın menüsünü önceden görebilmelerini, böylece bütçelerine uygun bir restoran seçimi yapabilmelerini sağlamak ve sipariş sürecindeki personel unsurundan dolayı ortaya çıkan hata ihtimali ve yavaşlığı azaltmak gibi hedefler barındırmaktadır. Bunun yanında restoran açısından bakıldığından personel maliyetini azaltmak ve daha hızlı bir sipariş süreci ortaya koymak da projenin hedefleri arasında yer almaktadır.

Proje geliştirme süreci, iki yarıyıyla ayrılmıştır. İlk yarıyilda yapılan çalışmalar: versiyon kontrol sistemi kurulması, arayüz tasarımlarının yapılması ve bir kısmının hayatı geçirilmesi, veritabanı tablolarının oluşturulması, veritabanına bağlı arka yüz kodlarının yazılması ve QR kod geliştirmelerinin tamamlanmasından oluşmaktadır. İkinci yarıyilda yapılan çalışmalar ise: arayüzlerin tamamlanması, harita eklenmesi, harita üzerinden restoran bulma algoritmalarının yazılması, mobil uygulama, sunucu ve web sitesi bağlantılarının yapılması ve en sonunda bulut teknolojileri kullanarak uygulamanın hazır hale getirilmesinden oluşmaktadır.

## **ABSTRACT**

The project is inspired by the fact that many different processes in different fields have become autonomous and require less staff today. The main element of this project is to provide the possibility that the menu review and ordering processes can be done by the customer without a waiter. The project includes objectives such as giving the customer the ability to see a restaurant's menu in advance, which in turn enables them to choose a restaurant that fits their budget, and reducing the possibility of error and slowness caused by the human element in the ordering process. In addition, for the restaurant, reducing personnel costs and providing a faster ordering process are among the objectives of the project.

The project development process is divided into two semesters. The work done in the first semester includes establishing a version control system, making interface designs and implementing some of them, creating database tables, writing the back-end codes connected to the database and completing the QR code developments. The work completed for the second semester consists of completing the interfaces, adding maps, writing the algorithms for finding restaurants on the map, finishing the mobile application, server and website connections, and finally making the application ready to use by utilizing cloud technologies.

# **İÇİNDEKİLER**

<b>ONAY SAYFASI</b>	<b>i</b>
<b>TEŞEKKÜR</b>	<b>ii</b>
<b>ÖZ</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>İÇİNDEKİLER</b>	<b>0</b>
<b>1.GİRİŞ</b>	<b>1</b>
1.1. Genel Motivasyon	1
1.2. Fonksiyonel Gereksinimler	1
1.3. Fonksiyonel Olmayan Gereksinimler	2
1.4. Sistem Kısıtları	2
1.5. Başarım Ölçütleri	2
1.6. Görev Dağılımı	3
1.7. Sistemin Genel Yapısı	3
1.8. Proje Planı	5
1.9. Raporun Sonraki Bölümleri ile İlgili Genel Açıklama	5
<b>2.ARAÇLAR VE YÖNTEMLER</b>	<b>6</b>
2.1. Proje Organizasyonu	6
2.2. Arka Yüz (Backend) Tasarımı ve Geliştirilmesi	8
2.2.1 Bulut Kullanımı	9
2.3. Ön Yüz (Frontend) Tasarımı ve Geliştirilmesi	17
2.4. Veri Tabanı Tasarımı ve Entegrasyonu	21
2.5. Mobil Uygulama Tasarımı ve Geliştirilmesi	23
2.5.1 Tanım	23
2.5.2 Kapsam	24
2.5.3 Kullanılan Teknolojiler	24
2.5.4 Ekran Mimarisi	25
2.5.5 Redux Veri Deposu Mimarisi	26
2.5.6 Bileşen (Component) Tasarımı	27
<b>3. SONUÇLAR</b>	<b>31</b>
<b>4. TARTIŞMA</b>	<b>32</b>
<b>5. ŞEKİLLER DİZİNİ</b>	<b>33</b>
<b>6. SİMGELER ve KISALTMALAR DİZİNİ</b>	<b>34</b>
<b>7. KAYNAKLAR</b>	<b>35</b>
<b>8. EKLER</b>	<b>37</b>

# **1.GİRİŞ**

## **1.1. Genel Motivasyon**

Proje genel anlamıyla yemek seçme ve sipariş etme sürecinin verimli bir hale getirilmesine odaklanmaktadır. Bu açıdan hem müşterilere hem işletmecilere yarar sağlamak projenin başlıca amaçlarındanandır. Mobil Uygulama, Web sitesi, QR kod ve harita teknolojileri kullanılarak birçok bakımından kullanım kolay, sürece pozitif etki eden bir uygulama ortaya konmaya çalışılmaktadır.

Projenin müşteri tarafından kullanılacak olan özelliklerinden bahsetmek gerekirse; Müşteri birçok restoranın menüsünü, fiyat bilgisini, restoran puanlarını ve diğer kullanıcıların restoranındaki düşüncelerini restorana gitmeden görebilecek ve kendine uygun restoranı seçebilecektir. Dilerse uzaktan sipariş vererek restoranda yerini rezerve edebilecek dilerse restoranın masaları üzerinde bulunan QR kod ile siparişlerini garsona ihtiyaç duymayacak bir şekilde verebilecek ve ödemeyi isteğine bağlı yine garsona ihtiyaç duymayacak bir şekilde uygulama üzerinden yapabilecektir. Aynı zamanda müşteriler uygulamayı açtıklarında yakınındaki restoranları harita üzerinden keşfetebileceklerdir.

Proje işletmeler tarafından web sitesi üzerinden kampanya yayılama, siparişleri anlık görebilme, garson talep eden masaları görebilme, müşteriyi başka masaya taşıyabilme, menüyü güncelleme, yorumlarını okuma, masalar için QR kod oluşturma ve oluşturulmuş QR kod aracılığıyla her masanın siparişlerini takip etme fonksiyonelitelerini sağlayacaktır.

Projemiz, işletmelerde sipariş alma sürecinin otomatikleştirilmesini ve bu sayede işletmenin personel maliyetini azaltmayı hedeflemektedir. İşletmelerin güzel görünen menülere ve bu menülerin güncellenmesine özellikle enflasyonist bir ekonomide gereksiz kaynak sarf ettiklerini düşünmekteyiz ve uygulamamız sayesinde bu gereksiz harcamaların işletme içerisinde daha verimli şekilde kullanılmasını sağlamayı hedeflemekteyiz. İşletmelerin eve yemek siparişi veren kullanıcılar tarafından puanlanması yerine restoranın atmosferini deneyimleyen, yemekleri plastik kaplarda yemek yerine olması gereği şekilde yiyen kullanıcılar tarafından puanlanmalarının daha isabetli olacağını düşünmekteyiz. Bu şekilde, uygulamamız sayesinde iyi işletmeler daha öne çıkarken başarısız işletmelere ise alacakları kullanıcı geri bildirimleri sayesinde eksiklerini giderme imkanı sunmayı amaçlamaktayız.

## **1.2. Fonksiyonel Gereksinimler**

- Sistem, kullanıcının ve işletmenin kaydını ve girişini sağlamalıdır.
- Sistem, kullanıcıya ve işletmeye şifre değiştirme imkanı vermelidir.
- Sistem, kullanıcıya ve işletmeye dil değiştirme imkanı vermelidir.
- Sistem, kullanıcılarla en güncel menüyü gösterebilmelidir.
- Sistem, kullanıcıların QR kodları sorunsuz bir şekilde okutabilmesini ve doğru adrese yönlendirmesini sağlamalıdır.

- Sistem, kullanıcıya restorana gitmeden siparişlerde kart ile, restorandan siparişlerde ise kart ve nakit ile sipariş verme ve ödeme yapabilme imkanı vermelidir.
- Sistem, kullanıcıya restoran arama imkanı vermelidir.
- Sistem, yapılan siparişin ödeme kabulünü yapabilmelidir.
- Sistem, kullanıcıya yorum yapma imkanı vermelidir.
- Sistem, kullanıcıya en yakın Afiyet kullanan işletmeleri gösterebilmelidir.
- Sistem, işletmeye menüyü güncelleme şansı vermelidir.
- Sistem işletmeye, menüdeki yiyecek ve içeceklerin sipariş edilebilir olup olmadığına karar verme imkanı vermelidir.
- Sistem, işletmeye masa ekleme, silme, güncelleme ve ilgili masalar için QR kod oluşturmaya izin vermelidir.
- Sistem, işletmeye beklenen siparişleri gösterebilmelidir.
- Sistem, işletmeye yorumlarını okuma imkanı vermelidir.
- Sistem, işletmeye müşterilerin masalarını rezerve etme ve müşterilerin siparişlerini başka masaya taşıma imkanı vermelidir.
- Sistem, işletmeye kampanya yayinallyama imkanı vermelidir.

### **1.3. Fonksiyonel Olmayan Gereksinimler**

- Kullanıcı ve işletme parolası en az sekiz en çok on dört karakterden oluşmalı, en az bir rakam, bir büyük harf, bir küçük harf, bir noktalama işareteti içermelidir.
- Sistem %99.5 oranında çalışma süresine (*uptime*) sahip olacaktır
- Sistemin çökmesi durumunda 7 dakika içerisinde düzeltilmelidir.
- Sistem kullanıcı için Android ve iOS platformları üzerinde çalışabilecektir
- Sistem işletme için Web üzerinde çalışabilecektir.
- Tek bir mail adresi ile tek bir hesap açılabilir.

### **1.4. Sistem Kısıtları**

- Bu uygulamayı kullanacak olan işletmenin gerekli donanıma ve bunun yanında internet bağlantısına ihtiyacı vardır.
- Kullanacak olan müşterinin ise QR kod okutmak ve uygulamaya giriş yapmak için kameraya sahip, internet bağlantısı olan bir telefona ihtiyacı vardır.
- Müşterinin telefonunda GPS özelliği açık durumda olmalıdır.
- Sistem sürekli olarak interne bağı olmalıdır.

### **1.5. Başarım Ölçütleri**

- Kullanıcı aktivitesinin yoğun olduğu zamanlarda sistemin performans sorunu vermemesi.
- Ürün ve menülerin en güncel hallerinin kullanıcıya gösterilmesi sağlanmalıdır.
- Müşteriler ile işletme arasında oluşturabilecek hataların en düşük seviyede tutulması.
- QR Kodlarının restoran sahibi tarafından doğru bir şekilde oluşturulması ve müşteri tarafından QR kod ile doğru adrese yönlendirmenin sağlanması.

## 1.6. Görev Dağılımı

Mehmet Umut İnceer; mobil uygulama tasarımları, mobil uygulama arayüzünün kodlanması,döküman hazırlama konularında,

Barış İkinci; web arayüzü tasarımları, web sitesi arayüzü kodlanması, döküman hazırlama konularında,

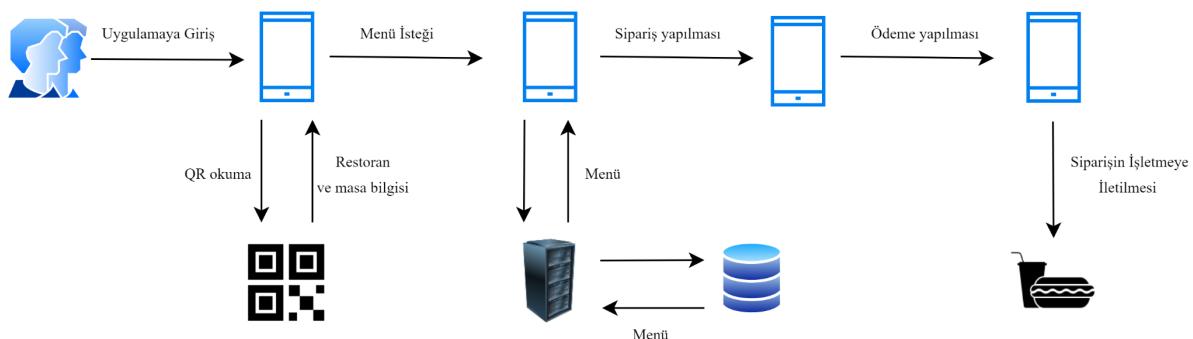
Emin Umut Gerçek, arka uç mimarisini kurma, bulut teknolojileri geliştirme, genel arka uç kodlaması,döküman hazırlama konularında,

Umut Çiloğlu, veritabanı oluşturma, veritabanı ile arka uç arası bağlantıyı kurma, genel arka uç kodlaması, döküman hazırlama,

Öncüm Korkmaz Yılmaz, genel arka uç kodlaması, bulut teknolojileri geliştirme, döküman hazırlama konularında katkı sağlamıştır.

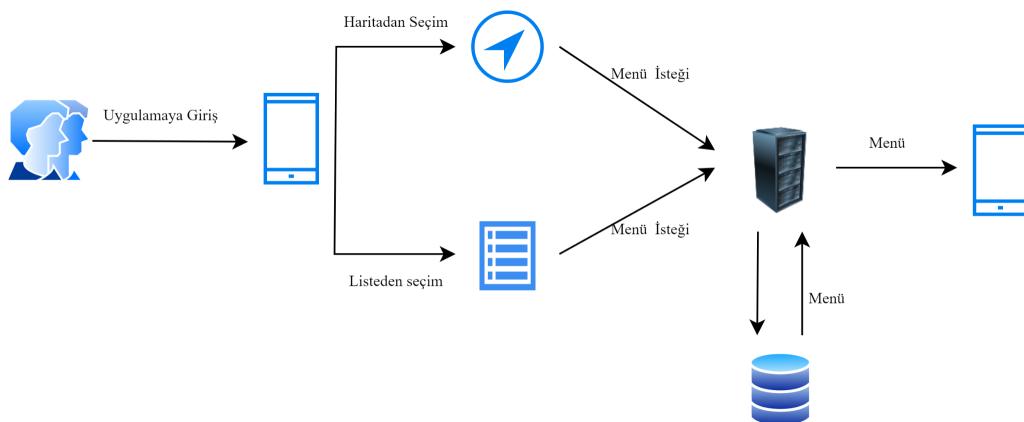
## 1.7. Sistemin Genel Yapısı

Kullanıcı tarafından bakmak gerekirse; Şekil 1.1'de gösterildiği gibi kullanıcı restoranın bir masasındaki QR kodu okutur. Bu QR sayesinde istek sunucudan geçerek veritabanına iletilir. Veritabanından alınan menü bilgisi yine sunucu üzerinde kullanıcıya döndürülür. Bu noktada kullanıcı menüden siparişini hazırlar ve gönderir. Gönderildikten sonra eğer kullanıcı kart ile ödeme seçti ise ödeme sayfası doldurularak onaylanır, nakit ile ödeme seçti ise sipariş nakit olarak işaretlenir. Eğer hiçbir hata yapılmadıysa sipariş işletmeye geçer. Sipariş hazırlandığında garson siparişleri masaya getirir. Nakit ödeme seçildi ise kullanıcı uygulama üzerinden garson çağrıbilir, kart ile ödeme yapıldı ise kullanıcı için yapacak hiçbir şey kalmamıştır.

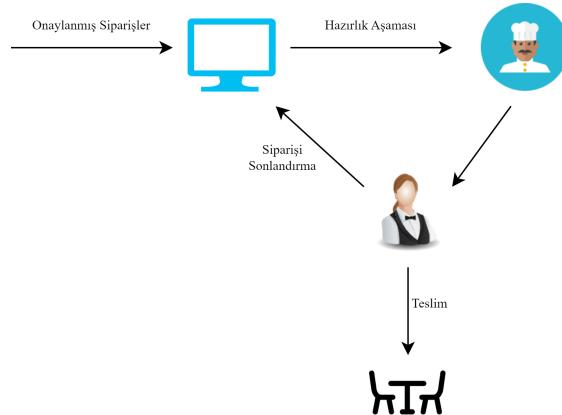


Şekil 1.1 : Sistemin Genel Yapısı

Kullanıcı ayrıca şekil 1.2'de gözüktüğü gibi herhangi bir restorana girmeden haritadan veya listeden restoranı seçip menülerine bakabilme şansına sahiptir. Eğer kullanıcı kart ile ödeme yapılacaksa ve restorana belli bir mesafede ise uzaktan sipariş verebilir.

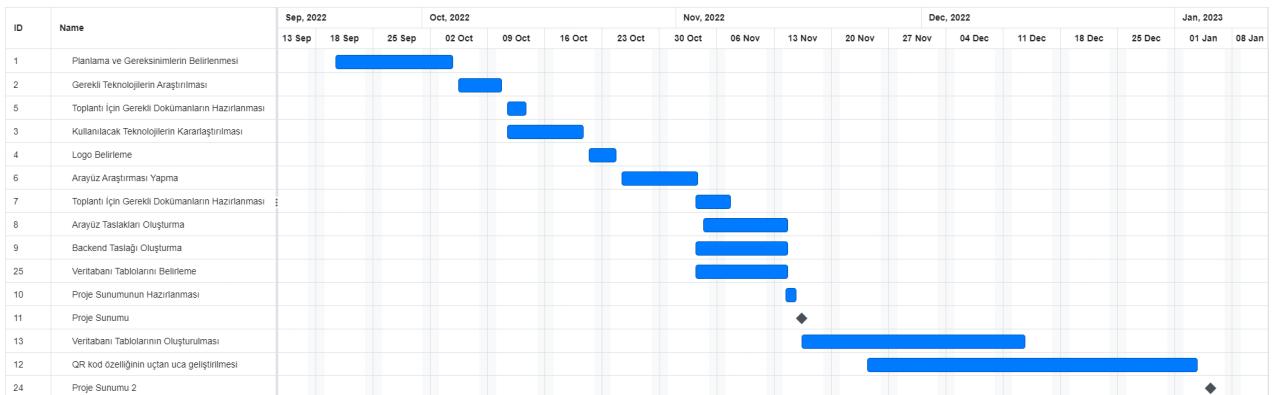


Restoran tarafında gelen siparişler Şekil 1.3'de görüldüğü üzere web sitesinde üzerinde gözükmektedir. Gözükken siparişler hazırlanma sürecinin bitirilip teslim edildiğinde sipariş ekrandan onaylanır ve ekranдан silinmiş olur.



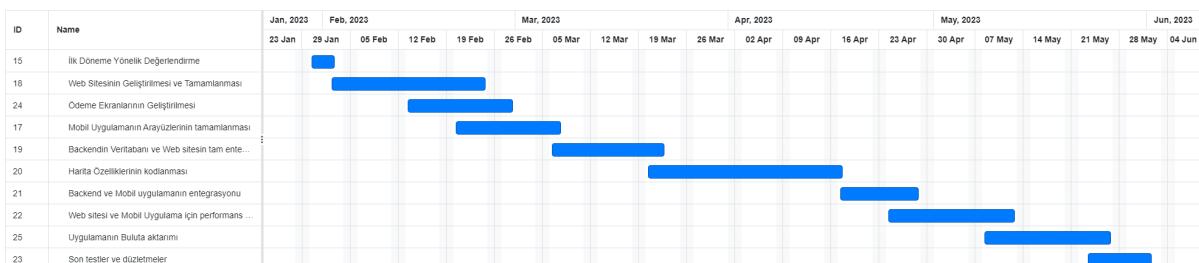
## 1.8. Proje Planı

Şekil 1.4'de görülen Gantt Çizelgesi 2022-2023 güz dönemine ait iş-zaman planlamasını göstermektedir.



Şekil 1.4 : Güz Dönemi İş-Zaman Planı

Şekil 1.5'te görülen Gantt Çizelgesi ise 2022-2023 bahar dönemine ait iş-zaman planlamasını göstermektedir.



Şekil 1.5 : Bahar Dönemi İş-Zaman Planı

## 1.9. Raporun Sonraki Bölümleri ile İlgili Genel Açıklama

Projeyle ilgili açıklamalar 3 ana başlıkta verilmiştir. Araçlar ve yöntemler bölümünde, proje organizasyonu, arka uç tasarımını ve geliştirilmesi, ön uç tasarımını ve geliştirilmesi, veritabanı tasarımını ve entegrasyonu, mobil uygulama tasarımını ve geliştirilmesi bölmeleriyle projenin genel olarak nasıl ilerlediği ve bu ilerleme de kullanılan yöntemler açıklanmıştır. Sonuçlar ve tartışma bölmelerinde projenin son durumu, avantajları ve dezavantajlarından bahsedilmiştir.

## 2.ARAÇLAR VE YÖNTEMLER

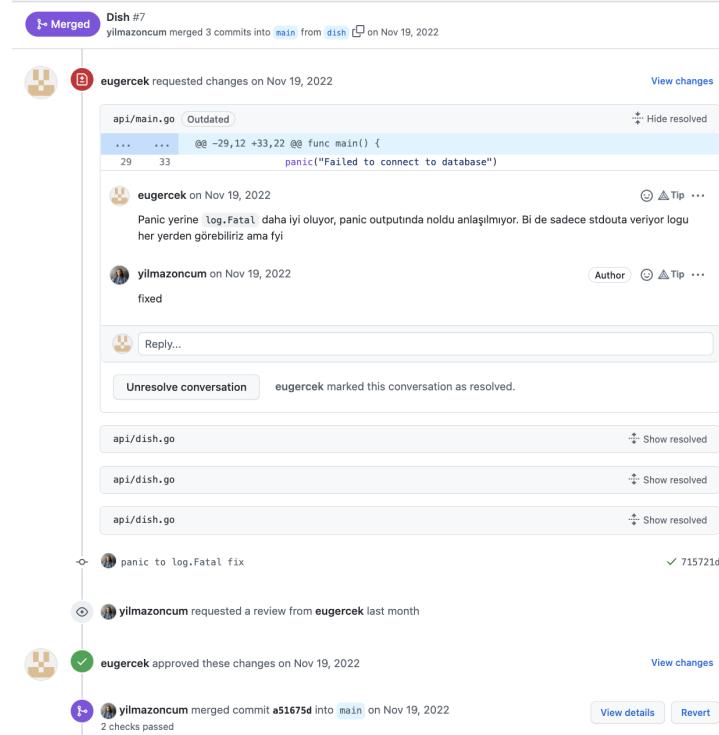
### 2.1. Proje Organizasyonu

Proje bir grup tarafından yapıldığı için bir kaynak kod sisteminin (*version control system*) kullanılması neredeyse zorunludur. İşbirliği (*collaboration*) amacı dışında da yapılan değişikliklerin kaybolmaması için bir şekilde kayıt altına alınması ve aynı anda birden fazla değişikliğin de kayıt altına alınabilmesi gereklerinden ötürü bir kaynak kod sistemi kullanılmıştır. Git yazılımı, geliştiriciler tarafından kullanılan yazılımlar arasında neredeyse tek olmuş bir yazılımdır [1]. Projede işbirliği yapılması için bir Git sağlayıcısı (*host*) kullanımına karar verilmiştir. Projede Git sağlayıcısı olarak GitHub kullanılmıştır. Bu seçimdeki ana nedenler şunlardır: En bilinir Git sağlayıcısı olması, sayısal limitlerinin alternatiflerine göre çok yukarıda olması, gelişmiş bir Boru Hattı (*pipeline*) sisteminin olması (GitHub Actions) [2].

Proje temel olarak 4 parçadan oluşmaktadır. Bir arka uç programı, bir web arayüzü ve bir mobil arayüz ve bir rapor programı. Bu kodların hepsi tek bir Git deposunda bulunur, (*monorepo Git repository*). Ölçeklenebilirlik (*scalability*) açısından da bir problem teşkil etmez, çok büyük ölçekte (*scale*) şirketler tek depoyu (*monorepo*) kullanmışlardır ve kullanmaya devam etmektedirler [3]. Bu depoda bulunan dosyalar şunlardır:

- Go arka uç kodları
- Rapor uygulaması kodları
- Web ve mobil arayüz kodları
- Boru hattı (pipeline dosyaları)
- İyzico html dosyaları

Projenin başlarında “Trunk Based Development” kullanılmış olsa da paralel bir şekilde geliştirme yapmayı kolaylaştırmak için sonradan Git Flow kullanılmıştır, Git sağlayıcısı olarak GitHub kullanıldığı için GitHub Flow'un Pull Request (PR) özelliği de kullanılmıştır. Örnek bir geliştirme şu şekilde yapılmaktadır: İlk olarak dev dalından (*branch*) yeni bir özellik (*feature*) dalı açılır ve burada geliştirme yapılır. Geliştirme bittiği zaman ise bir PR açılır ve bir değerlendirmeli (*reviewer*) atanır. Değerlendiricinin istekleri doğrultusunda kodda değişiklik yapılır. Değerlendiricinin onayından geçen dal dev'e birleştirilir (*merge*). Projenin ana dalı olan main'de sadece hata olmadığından emin olunan kodlar bulunur. Müşteriler main dalının dağıtılmış (*deploy*) halini kullanıyor olacaklar, bu main dalında çalışan 3 tane boru hattı sayesinde sağlanır.



Şekil 2.1: Örnek bir PR

GitHub sağlayıcısından kullanılan yazılımlar şunlardır: Projects (Jira benzeri bir proje takip yazılımı), GitHub Actions (Boru hattı yazılımı).

Todo	In Progress	Done	Long
Implement login endpoint and logic	Add insomnia files	Move to layered architecture Move to parameters -> json in put like handlers Move one of the endpoints to proper repository pattern Implement dish, rating	Investigate loggers Investigate json validators Investigate more go native architectures
Add echo's popular middlewares to improve ops			
standardize where queries in repositories			
Add eager loading to table and order etc. list and gets			
Take a look at orm foreign keys there is ambiguity			
gorm doesn't add foreign keys to orders for dishes, there is need for a mediatory table.			

Şekil 2.2: Örnek bir GitHub Projects

## 2.2. Arka Yüz (*Backend*) Tasarımı ve Geliştirilmesi

Projenin arka yüzünün geliştirmesinde programlama dilleri ve çerçeveye (*framework*) iki kısım üzerinde değerlendirilmiştir. İlk kısım insan faktöridür, kullanılacak programlama dili ve kütüphanelerin ekip içindeki kişiler tarafından hızlı öğrenilmesi ve katkıda bulunma süresinin azaltılması istenmiştir. Go dilinin yalınlığı sayesinde en iyi kalitede (*best practice*) kod yazılmasına başlanması diğer programlama dillerine nazaran daha kısa olması nedeniyle, Go'nun en uygun dil olduğunda karar verilmiştir. Go dilinin yaratıcılarından biri olan Rob Pike'ın bu konuda pek çok söylemi vardır [4, 20:40-21:05][5]. İkinci kısım ise mühendislik kalitesinin olabildiğince yüksek olmasıdır. Bu kısmada Go dili; performans, bağımlılık (*dependency*) sayısının az olması, dağıtımının (*deployment*) kolay olması ve kalıtım (*inheritance*) olmayan mimarilerin kurulması konularında öne geçmiştir.

Go programlama dilinde diğer programlama dillerinde olduğu gibi komünitenin (Ruby on Rails) veya dili yaratan şirketin tavsiye ettiği (ASP.NET Core) bir mimari yaklaşım bulunmamaktadır. Ve bu dili uzun süredir kullanıp, komünitede görüşlerine saygı duyulan insanların verdikleri tavsiyeler her zaman daha basit mimariler kurma yönündedir [6][7]. Arka ucu geliştirirken bu görüşlere kulak verilip Go'nun özüne uygun bir "Katmanlı Mimari" (*Layered Architecture*) geliştirilmiştir. Proje için mimariyi oluştururken daha modern mimariler de araştırılmış (Hexagonal Architecture, Clean Architecture, Onion Architecture) ve bu mimarilerden de kazanımlar elde edilmiştir. Bu mimarilerin hepsinin en önemli noktası olan, iş mantığına (business logic) hiçbir girdi (input) ve çıktıının (output) dahil olmaması prensibidir, bu prensibi benimseyip iş mantığı girdi ve çıktılardan soyutlanmıştır.

İş mantığını her türlü girdi ve çıktıdan ayırmak için Repository Pattern ve Servis mantığı kullanılmıştır. Repository Pattern ve GORM kütüphanesinin birlikte kullanılması ile proje hiçbir veritabanına bağımlı halde olmamıştır. Servis katmanın da ayrı olması sayesinde iş mantığı içerisinde HTTP'ye dair hiç bir kalıntı yoktur. Proje şu an HTTP ve PostgreSQL ile çalışsa da istenildiği zaman başka yazılımlar, protokoller ile entegre olabilir durumdadır. Bu pek çok mimarinin en büyük amaçlarından biridir [8].

Go programlama dilinde kalıtım (*inheritance*) olmadığı için “Composition over Inheritance” prensibine dayalı bir mimari yapılmıştır. Bu paragrafta EK-1'deki sınıf diyagramının User sınıfı ile alakalı bölümü açıklanacaktır. Go dilinde ortak düşünce (*convention*) olarak, çalışacak programlar, yani main fonksiyonuna sahip programlar cmd paketinde (*package*) bulunur. İleride yeni programların da yazılması gerekirse onlar da bu pakete eklenecektir. cmd paketinde App isimli sınıf (*class*), veritabanı ve Echo HTTP çerçevesi (*framework*) sınıflarını içerir. Veritabanına uygulamanı başlangıcında bir kere bağlanılır ve bu bağlantı üzerinden diğer bütün işlemler yapılır. Echo nesnesi (sınıftan yaratılmış somut obje, *instance*) çalışma zamanında (*runtime*) kullanacağı Handler sınıflarından bir obje yaratır. Bu Handler sınıfları içerisinde de o Handler'a yönelik bir servis vardır. Bu servislerde iş mantığı uygulanmıştır (*implementation*). Örneğin bir e-posta ile sadece bir tane hesap açılabilmesi gibi. Servis sınıfları veritabanına doğrudan erişemezler, bir Repository sınıfı aracılığı ile

erişim sağlarlar. Bu Repository sınıflarında tekrarı azaltmak için GenericRepository isimli bir sınıf oluşturulmuştur. Bu sınıf kullanılan Nesne-İlişkisel Eşleme (ORM) kütüphanesi olan GORM sayesinde yazılmıştır. Data paketinde ek olarak bir de Model diye bir paket vardır, bu sınıf iş mantığındaki parçaları, kodda temsil eder (representation). Go dilinde Bağımlılık Enjeksiyonu (Dependency Injection) açık (*explicit*) olarak yapılır, bu yüzden mimarinin her yerinde bağımlılıklar Yapıcı Metotlarda (*constructor*) argüman olarak verilir ve objenin içerisinde saklanır (*composition*). Herhangi bir Görüntü katmanı (*View*) bulunmamaktadır. Görüntü bölümü React ile yapılmaktadır. React kodu ile bağlantı ise hazırlanan REST API ile sağlanır, bu bağlantıda yük'ün (*payload*) içerisinde JSON kullanılmıştır.

Rapor programı, UTC+3 11:45'te bütün restoran hesaplarına günlük raporlarını pdf haline getirip e-posta olarak atar. Rapor programı sistemde programının zamanlayıcısı (timer) ile çalıştırılır, tamamen otomatik bir sistemdir.



Şekil 2.3: Örnek Bir Rapor

## 2.2.1 Bulut Kullanımı

Afiyet projesinin arka uç, depolama ve ön yüz parçaları bulut teknolojileri ile müşteriye sunulmaktadır. Altyapı mimarisinin diyagramı ekler bölümünde EK-5 olarak bulunmaktadır.

### **2.2.1.1 Arka Uç**

Afiyet projesinin arka ucu 3 parçadan oluşmaktadır: REST API sunucusu (*server*), rapor uygulaması ve ödeme entegrasyonu servisi. Bulardan REST API ve rapor uygulaması Go programlama dili ödeme entegrasyon servisi ise Node.js (JavaScript) ile geliştirilmiştir.

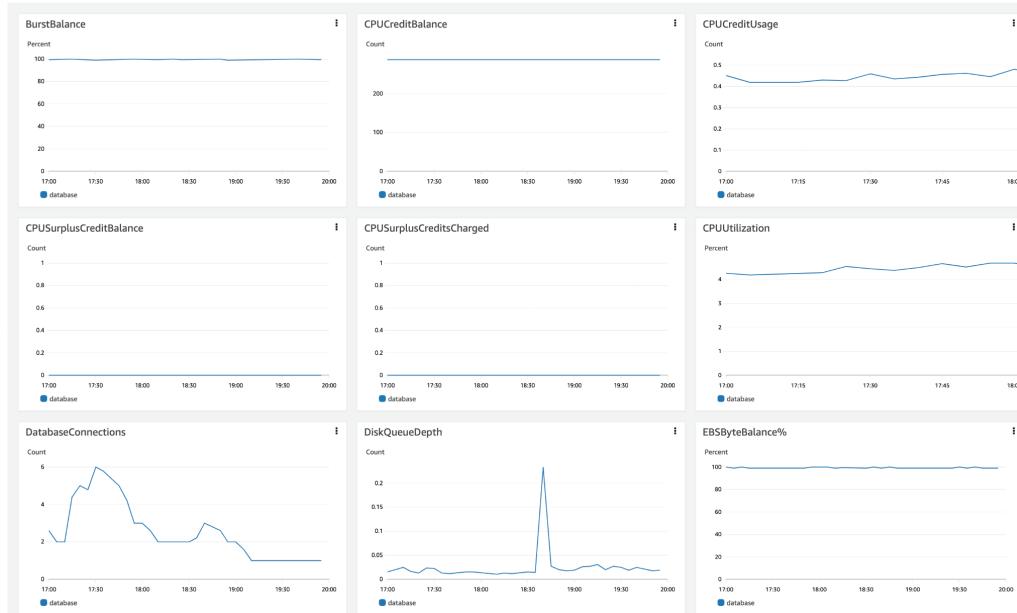
REST API, bir AWS EC2 sanal makinesi üzerinde çalışmaktadır. Go derleyicisinin hem çapraz derleme (*cross compilation*) hem de tek statik çalıştırılabilir (*single static binary*) üretmesi sebebiyle, altyapı son derecede minimal bir biçimde hazırlanabilmiştir.

### **2.2.1.2 Veritabanı**

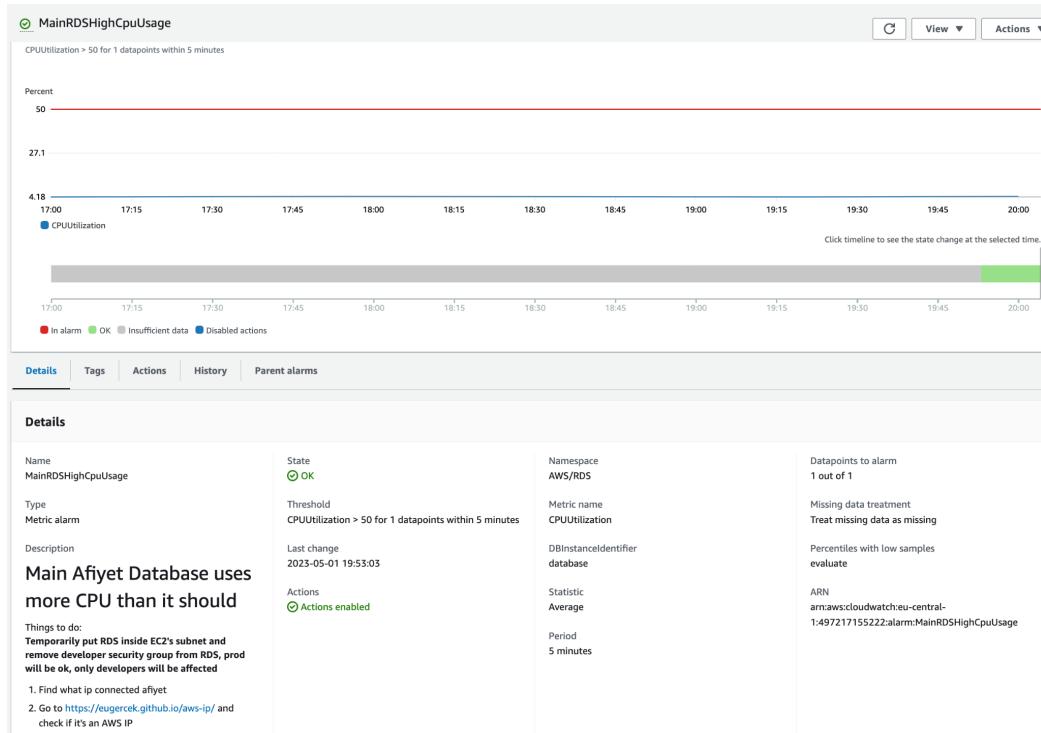
Afiyet projesi AWS üzerine kurulmasından önce de lokal veritabanı kullanılmadan geliştirilmiştir. Bu yapının kurulması ElephantSQL bulut şirketi üzerinde bir yönetilen (*managed*) PostgreSQL veritabanı kullanılması ile sağlanmıştır. Bu veritabanından taşınmasının nedenleri şunlardır: fiyat olarak daha pahalı olması, daha az veritabanı yöneticisi soyutlaması olması (DBA (Database Administrator) *abstraction*), uygulamanın çalıştığı sanal makinelere AWS'in sunduğu kadar yakın olmaması sebebiyle daha fazla gecikmesi (*latency*) olması, gelişmiş gözlemeleme altyapısının olmaması. Afiyet buluta taşınırken, pg\_restore ve pg\_backup programları ile veritabanı AWS'in sunduğu RDS'e taşınmıştır.

Afiyet'in bölge (*region*) seçimi zorunlu olan bulut servislerinin hepsi AWS'in Frankfurt bölgesinde (eu-central-1)'de kurulmuştur. RDS de bu bölgdededir. db.t3.micro isimli RDS sınıfı (*class*) seçilmiş ve eu-central-1b isimli kullanılabilirlik (*AZ, Availability Zone*) bölgesinde veritabanı kurulmuştur. Sınıf seçiminde fiyat, AZ seçiminde ise Afiyet'in arka ucuna yakın olması gözetilmiştir.

Veritabanlarının gözlemlenmesi (*monitoring*) fonksiyonel olmayan gereksinimlerin sağlanması için kritik öneme sahiptir. Örneğin işlemcinin normalde olması gerekenden çok kullanılmaya başlandığının farkedilmemesi, sistemler bozulduğunda yani işlemci tamamen tükeğinde farkedilir bu da aksama sürelerine (*down time*) sebep olur. Aksama süresi ile sonuçlanabilecek problemler yaşanmadan çözülmesi için RDS ile AWS CloudWatch ve AWS SNS servisleri bağlanarak bir çözüm üretilmiştir. RDS'in metrikleri, CloudWatch üzerinden bir zaman aralığında bir eşigi geçtiği zaman SNS üzerinden afiyet geliştiricilerine e-posta atılmaktadır.



Şekil 2.4: Veritabanı Metrikleri



Şekil 2.5: Örnek Veritabanı Alarmı

Ne kadar önlem alınsa da, her sistemin bozulabileceği varsayımlı ile sistemlerin kurulması aksama sürelerini bir kaza (*incident*) anında en aza indirecektir. Afiyet bulutta çalışan bir uygulama olduğu için bütün dünyadaki siber güvenlik tehditlerine karşı önlem alınması gereklidir. Bu önlemlerden veritabanı konusundaki bir önlem de veritabanının hepsinin günlük yedeğinin alınmasıdır (*daily backup*). Bu yedekler, artımlı (*incremental*) olduğundan bütün veritabanının yedeğinin alınmasından daha hızlıdır. Bir kaza durumunda önceki

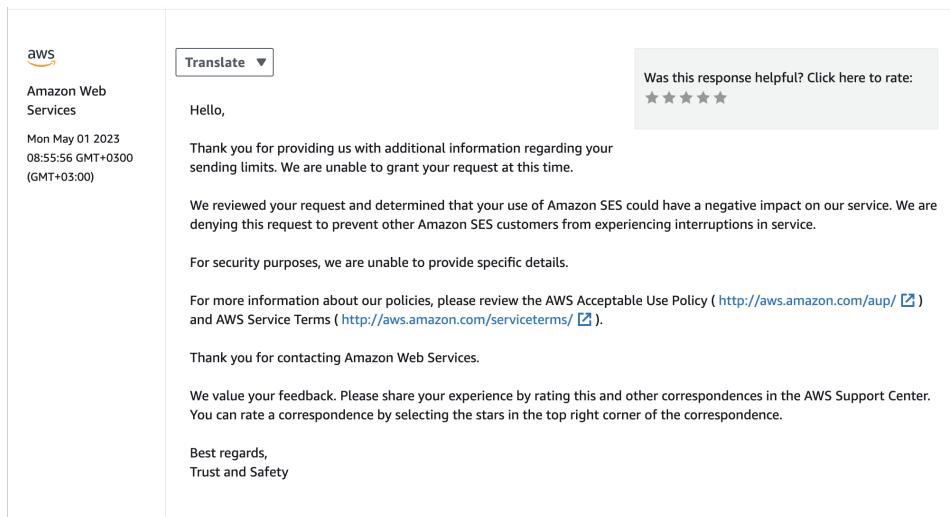
yedeklerin enstantelerinden (*snapshot*) veritabanı yedek alındığı zamanki haline döndürülebilir (*restore*). Yedekleme mekanızması bir ayda toplam kaç GB alan tutulduğundan ücretlendirilmektedir, bu da mimari kararlarda bir etkendir.

System snapshots (3)						
Name	DB source	Creation time	Status	Progress	VPC	
rds:database-2023-05-07-21-36	database	Mon May 08 2023 00:36:44 GMT+0300	available	Completed	vpc-0a88669b1379fa464	
rds:database-2023-05-08-21-37	database	Tue May 09 2023 00:37:20 GMT+0300	available	Completed	vpc-0a88669b1379fa464	
rds:database-2023-05-09-21-36	database	Wed May 10 2023 00:36:28 GMT+0300	available	Completed	vpc-0a88669b1379fa464	

Şekil 2.6: Veritabanı Enstantaneleri

### 2.2.1.3 E-posta İşlemleri

AWS'in SES (*Simple Email Service*) servisi kullanarak e-posta işlemlerinin geliştirilmesi planlanmıştır lakin, AWS'in şirket politikaları nedeniyle SES servisi kullanılmamıştır. Bu durumun düzeltilmesi amacıyla AWS'in müşteri hizmetlerine etiket (*ticket*) açılmıştır, ama fayda sağlanamamıştır.



Şekil 2.7: Müşteri Hizmetleri Etiketi Sonucu

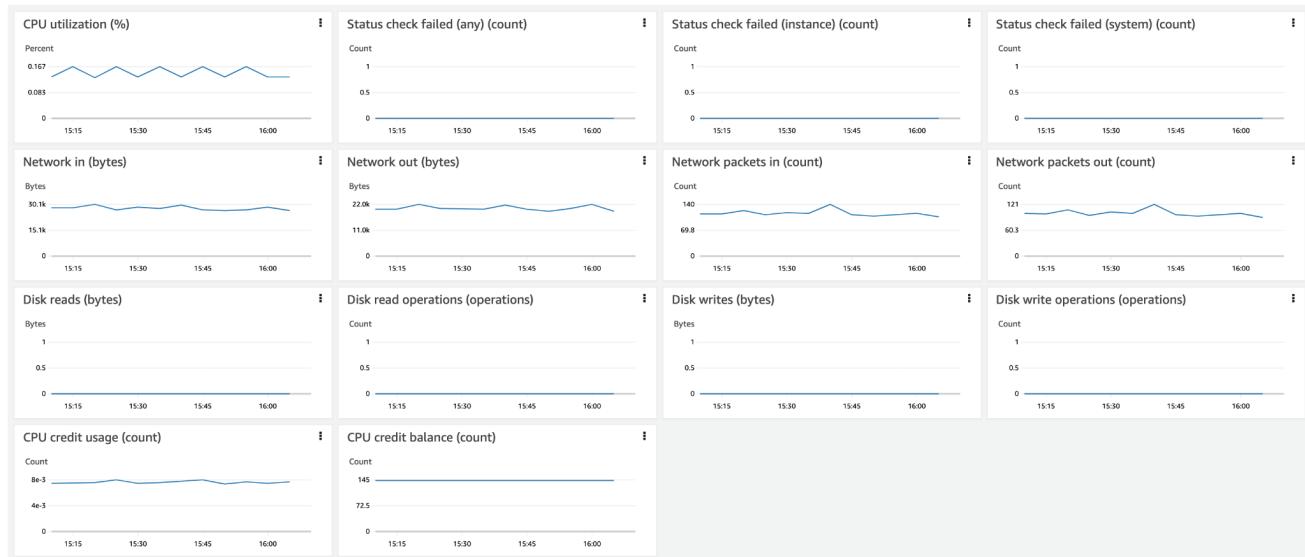
Bu problemin çözümü için profesyonel bir e-posta hizmetine en yakın alternatifler değerlendirilmiş ve Gmail üzerinden e-posta işlemlerinin yapılması sağlanmıştır. Afiyet REST API'na OAuth2 kimlik doğrulama (*authentication*) yükü eklememek için, App Passwords isimli doğrulama metodu kullanılmıştır ve standart SMTP protokolü üzerinden müşterilere e-posta atma özelliği geliştirilmiştir.

#### 2.2.1.4 Boruhatları

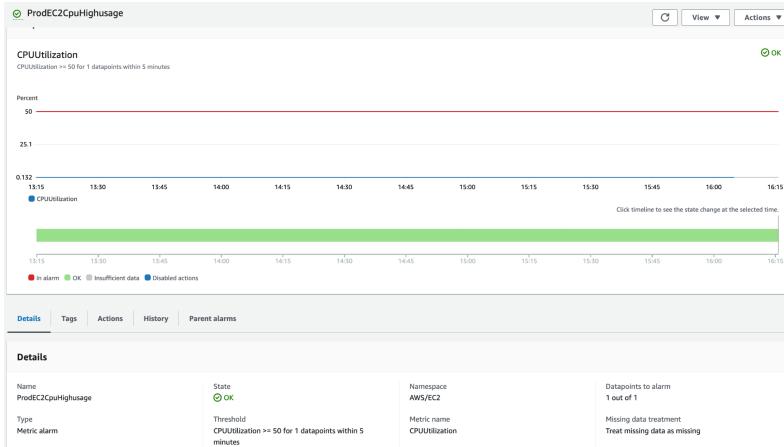
Afiyet uygulamasının kaynak kodları GitHub üzerinde bulunmaktadır. GitHub'ın GitHub Actions boru hattı (*pipeline*) ürünü ile, Sürekli Teslimat, Sürekli Dağıtım (*CI/CD Continuous Integration Continuous Delivery*) teknikleri kullanılmıştır. Üç tür boru hattı geliştirilmiştir: Arka uç ve rapor uygulaması için go kodunu derleyip, sanal makinaya atan iki tane pipeline geliştirilmiştir. Web için JavaScript, CSS, HTML ve kullanılan resimleri paketleyip (*bundle*) [www.afiyet.site](http://www.afiyet.site) isimli S3 obje deposuna atan bir pipeline.

#### 2.2.1.5 Sanal Sunucu

Afiyet arka ucu ve rapor uygulaması bir EC2 (*Elastic Compute Cloud*) sanal sunucusunda bulunmaktadır. Fiyat nedenlerinden ötürü t2.micro sınıfı seçilmiştir, arka uç Go programlama dili ile geliştirildiği için 1 CPU ve 1 GB RAM'e sahip bir sanal sunucu kullanılması bir engel değildir. Bu sunucuda yaşanacak problemler hem web ön yüzünü hem de mobil ön yüzünü etkileyeyecek için yapılabilecek hatalar minimize edilmeye çalışılmıştır. Örneğin sanal sunucunun yanlışlıkla kapatılmasını önlemek için “Termination Protection” özelliği aktive edilmiştir, sanal sunucunun CPU yükü gözlemlenmiştir (*monitoring*) ve belirli bir eşikten sonra geliştiricilere e-posta atılması sistemi kurulmuştur.



Sekil 2.8: Sunucu Metrikleri



Şekil 2.9: Sunucu Alarmı

### 2.2.1.6 TLS Sertifikası ve Alan Adı

Restoranların erişebilmesi için **afiyet.site** ve arka uç için ise **backend.afiyet.site** alan adları kullanılmıştır, **afiyet.site** alan adı Namecheap şirketinden alınmıştır.

Type	Host	Value	TTL
A Record	backend	52.57.220.100	Automatic
CNAME Record	@	d1kr8uzt8dadw2.cloudfront.net.	Automatic

Şekil 2.10: Afiyet Projesinin DNS Kayıtları

Bulutta çalışan bir arka uç programının HTTPS kullanmaması durumunda tüm iletişim dünyadaki herkes tarafından görülebilir olacaktır, bunun önüne geçmek için afiyet arka ucu sadece HTTPS üzerinden fonksiyonalite vermektedir, HTTP isteği geldiği durumda arka uç tarafından gene HTTPS'e yönlendirme yapılmaktadır. Bir diğer önemli neden ise müşterinin tarayıcı üzerinden **afiyet.site**'a eriştiği duruma tarayıcısında kilit ikonu çıkmasıdır, bütün modern tarayıcılar (Google Chrome, Edge, Brave, Safari, Firefox ...) HTTP kullanılması durumunda kilit ikonu göstermemektedir, ve CA (*Certificate Authority*) tarafından imzalanmamış (*sign*) sertifikalarda ise uyarı ekranı göstermektedir.

Namecheap üzerinden fiyat nedeniyle sertifika alınmamıştır. Sertifika alınma işlemi Let's Encrypt üzerinden yapılmıştır. Let's Encrypt ACME (*Automated Certificate Management Environment*) protokülü ile ücretsiz sertifika hizmeti veren, 300 milyondan fazla sertifika vermiş bir şirkettir. Bazı alan adı kayıt edicisi (*domain registrar*) şirketlerinin Let's Encrypt entegrasyonu vardır, Namecheap'in entegrasyonu olmadığı için DNS üzerinden ACME

protokolü ile kimlik sorgulama (*challenge*) yapılamamıştır. Bu problemin aşılması için bir EC2 sanal sunucusunda auto-cert programı kurulup, alınacak alan adı için bir A kaydı (*record*) açılıp bu kaydı açılan EC2 sanal sunucunun genel (*public*) IP'sine atanmıştır.

### 2.2.1.6 Obje Depolanması

Afiyet projesinin pek çok parçası obje depolarında (*object storage*) bulunmaktadır. AWS'in S3 servisi obje deposu olarak kullanılmıştır. Bu servisin kullanılması sayesinde S3'ün şu avantajları elde edilmiştir: on bir dokuz (%99.999999999) olarak bilinen on milyon yılda on bin tane objeden en fazla bir tanesini kaybolma garantisi, güvenlik için uyarı veren bir UI'ın AWS konsolu üzerinden sunulması, CloudFront CDN entegrasyonu kolaylığı.

3 tane obje deposu bulunmaktadır:

1. **afiyet-media:** Restoran, yemek ve kampanya resimlerinin bulunduğu depo
2. **www.afiyet.site:** Afiyet web arayüzünün derlenmiş halinin bulunduğu depo
3. **iyzico-mock:** “İyzico sandbox” servisinin doğru çalışmaması durumunda kullanılacak sistemin bulunduğu obje deposu

Afiyet projesinde medyalar obje deposunda tutulmuştur bunun nedenleri şunlardır: veritabanını hantallaştırmamak, veritabanın yedekleme mekanizmasının ücretlendirme biçimini, CDN entegrasyonu kolaylığı, eşzamanlı (*concurrent*) medya isteği (*request*) atabilme özelliği.

afiyet-media ve **www.afiyet.site** depoları genel (*public*) değildir, bulutta depo güvenliği konusunda önlemler alınmıştır. Bir bulut servisi kullanıldığı ücret göz önüne alınmıştır, istek (*request*) başı ücretlendirmeye sahip bir servis bot taramalarında afiyet ekibine büyük bir ücret olarak dönebilirdi.

Objects (4)	
Objects are the fundamental entities stored in Amazon S3. You can use <a href="#">Amazon S3 inventory</a> to get	
<input type="button" value="C"/>	<input type="button" value="Copy S3 URI"/>
<input type="button" value="Copy URL"/>	<input type="button" value="Download"/>
<input type="button" value="Open"/>	
<input type="text"/> <i>Find objects by prefix</i>	
<input type="checkbox"/> Name	Type
<input type="checkbox"/> campaign/	Folder
<input type="checkbox"/> dish/	Folder
<input type="checkbox"/> other/	Folder
<input type="checkbox"/> restaurant/	Folder

Şekil 2.11: afiyet-media Obje Deposu

## 2.2.1.7 İçerik Dağıtım Ağı

İçerik Dağıtım Ağı (*CDN Content Delivery Network*) kullanımı sistemin gecikmeleri (*latency*) azaltmak konusunda önemli bir etkendir. Afiyet AWS'in CloudFront servisini kullanılmıştır. İki tane CDN dağıtımını (*distribution*) kurmuştur bunlar, websitesi sunmak (*hosting*) ve medya istekleri içindir.

ID	Domain name	Alternate domain names	Origins	Status	Last modified
EP50A88U2X8MZ	d1kr8uzt8dadw2.cloudfront.net	afiyet.site	www.afiyet.site.	Enabled	May 9, 2023 at 7:44:19 AM UTC
E2531XH1PUP8G6	d33virtamuit9v.cloudfront.net	-	afiyet-media.s3.	Enabled	April 23, 2023 at 7:35:45 PM UTC

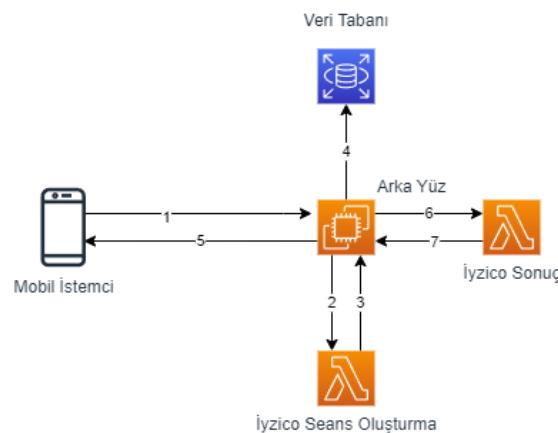
Şekil 2.12: CloudFront Dağıtımları

Dağıtımlar OAC (*Origin Access Controller*) üzerinden gizli (*private*) S3 obje deposuna erişim sağlarlar. CDN genel (*public*) olduğu için botlar tarafından taraması çok olağandır, bu nedenden ötürü ücretlendirme araştırılmıştır.

Web arayüzüne **afiyet.site** alan alan adından erişim sağlandığı için bu adres için "Alternate Domain Name" özelliği kullanılmıştır, bu özellik için **afiyet.site** için alınan TLS sertifikası ACM (*AWS Certificate Manager*) servisine aktarılmıştır, CloudFront ACM'i kullanarak dağıtımını yapar.

## 2.2.1.8 Ödeme Altyapısı

Ödeme altyapısı kartla ve nakit para ile olmak üzere iki ana ödeme şeklini desteklemektedir. Kart ile ödeme altyapısında İyzico ödeme sistemi kullanılmıştır. İyzico yerli bir ödeme sistemi olması ve hafif (*lightweight*) bir sistem olması sebebi ile tercih edilmiştir. Kartla ödeme metodunda aynı zamanda müşteri restorana gelmeden de sipariş verip ödeme yapabilmektedir. İyzico ödeme sistemi arka yüzün yazıldığı go dilini desteklemediği için AWS lambda üzerinden Node.js ile yazılmıştır ve iki fonksiyondan oluşmaktadır. Bu fonksiyonlardan ilki ödeme seansının başlatılmasına ve ödeme sayfasının oluşturulmasına yarayan lambda fonksiyonu iken diğeri ise ödeme sonucunu öğrenmeye yarayan fonksiyondur.



Şekil 2.13: Ödeme Sistemi Mekanizması

Kartla ödeme yapılması durumunda mobil istemci sipariş oluşturmak için gereken kullanıcı, restoran, masa, yemek ve uzaktan sipariş olup olmadığı bilgisi arka yüzde ilgili uç noktaya gönderilir. Arka yüz bu bilgiler ile AWS'de yayınlanan İyzico Lambda fonksiyonuna istek atılır ve iyzico seansı oluşturulur. İyzico ödeme bilgilerini arka uca döndürür ve sipariş bilgileri İyzico bilgileri ile veritabanına kayıt edilir. Veritabanına sipariş bilgileri kaydedildikten sonra İyzico tarafından oluşturulan ödeme sayfası mobil istemciye döndürülür ve kullanıcıya gösterilir. Kullanıcı ödeme işlemini tamamlandığında mobil sayfası "callback" uç noktasına yönlendirilir ve uç nokta İyzico sonuç kontrolünü yapmak için yine AWS'de yayınlanan sonuç lambda fonksiyonuna istek atar. Bu sonuca göre mobil istemciye ödeme sonucu döndürülür ve ödeme işlemi bitirilir.

Uzaktan kartla sipariş aynı süreçleri takip eder ancak uzaktan sipariş verilmeden önce restoranda boş masa olup olmadığı kontrol edilir ve boş masa yoksa süreç ilerletilmez. Boş masa olması halinde sipariş rastgele bir masaya atanır ve süreç normal bir şekilde ilerler. Uzaktan verilen sipariş ve atandığı masa restorana bildirilir.

Nakit ödeme senaryosunda kullanıcı, masa, restoran ve yemek bilgileri arka yüze gönderilir ve siparişin nakit ile ödeneceği siparişe işlenip restorana bildirilir. Kullanıcı ödeme yapmak istediği zaman uygulama üzerinden bir garson çağrıbilir ve ödeme işlemini yapabilir.

### 2.3. Ön Yüz (*Frontend*) Tasarımı ve Geliştirilmesi

Projenin ön yüz tasarımı projenin ilk aşamasında yapılan taslağa kıyasla çok daha işlevli bir hale getirilmiştir. Projenin son halinde taslak aşamasında arka yüzde bulunmasına karar verilen işlemlerin ve fonksiyonların hepsi ön yüzde iletişim kurabilir hale gelmiştir. Arayüz, restoran çalışanlarının işlerini yapmak için kullanması gereken fonksiyonları barındırdığı için tasarlanırken öncelikli olarak kullanıcıya kolaylık sağlayan, pratik bir web sitesi tasarımlı tercih edilmiştir. Bunun yanında her başarılı arayüz tasarımda olduğu gibi renk uyumu göz önünde bulundurulmuş ve ekip olarak kırmızı ve beyaz renk tonlarının arayüzde kullanılması kararlaştırılmıştır. Bu seçimin sebebi ise AFİYET logosunun renklerinin kırmızı ve beyaz olmasıdır. Siteye stil eklenmesi işleminde Material UI [9] adlı React bileşenleri (component) içeren bir kütüphane kullanılmış ve Flexbox yardımıyla bileşenleri sayfada olmaları gereken yerlere koyma işlemi yapılmıştır [10].

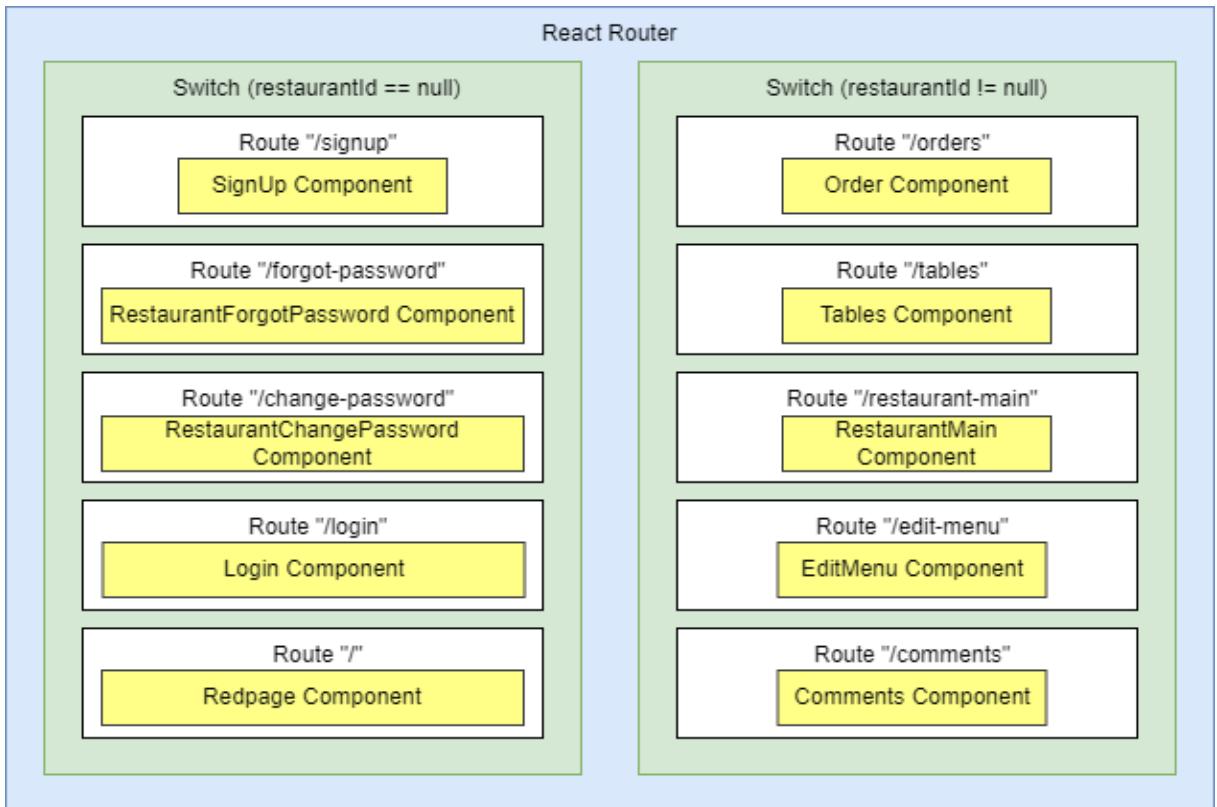
Web sitesi React [11][12] kullanılarak geliştirilmiş ve her sayfa ve sayfalarda sıkça kullanılan öğeler bileşen olarak tanımlanmıştır. Sitede bulunan sayfalar şunlardır:

- Giriş ekranı sayfası
- Kayıt olma sayfası
- Şifremi unuttum sayfası
- Restoran bilgileri sayfası
- Siparişler sayfası

- Masalar sayfası
- Menü düzenleme sayfası
- Yorumlar sayfası

Site yukarıda bahsedilen sayfalar dışında da sık kullanıldığı için bileşen haline getirilmiş sayfa öğelerine sahiptir.

Sitenin React Router kullanılarak oluşturulan sayfa mimarisi aşağıdaki gibidir:



Şekil 2.14: React Router Mimarisi

Sitenin giriş ekranı sayfasında popüler sitelerde bulunanlara benzer modern bir sayfa kullanıcıyı karşılamaktadır. Bu sayfada kullanıcı Material UI kullanılarak dizayn edilmiş e-posta alanına şirket e-postasını, şifre alanına ise kayıt olurken oluşturduğu şifreyi girerek giriş yapabilmektedir. Bu sayfada, bahsedilenler haricinde kayıt olma sayfasına yönlendirmeye, şifrenin unutulması durumunda şifre yenileme sayfasına yönlendirmeye ve tercih durumunda dil değiştirmeye yarayan butonlar da bulunmaktadır. Giriş yapmadan kullanılan verilerin doğru olup olmadığını kontrolü ise Axios yardımcıyla AWS'de bulunan arka yüze atılan POST istekleri (request) sayesinde yapılmaktadır. Arka yüze atılan POST isteğine dönen cevap bilgileri Redux'da "restaurantState" içerisinde ilgili değişkenlere kaydedilmekte ve site içerisinde kullanılmaktadır.

Sitenin kayıt olma sayfasında kullanıcı, restoranının fotoğrafını ve düzenlediği kampanyaları temsil eden bir resmi profiline ekleyebilmektedir. Bunlar dışında restoranının adını,

kategorisini, restoranın kullandığı şirket mailini ve kendi belirlediği şifreyi de belirtilen alanlara girebilmektedir. Sayfada son olarak restoranın haritada görüntülenebilmesini sağlamak için restoranın adresiyle enlem ve boylam bilgisi de kullanıcidan alınmaktadır. Kayıt olma işlemi de giriş yapma sayfasında olduğu gibi arka yüze POST isteği atma yoluyla çalışmaktadır.

Şifremi unuttum sayfasında ise site ilk önce kullanıcıya şifre yenileme isteğini gerçekleştirmek için atacağı URL'yi hangi maile atması gerektiğini sormaktadır. Bundan sonra kullanıcı mailine giriş yaparak oraya yollanan URL sayesinde şifresini değiştirebildiği sayfaya yönlendirilmektedir. Bu sayfaya maile atılan URL kullanılmadığı sürece giriş yapılamamaktadır. Bu sayfanın içerisinde ise kullanıcından yeni şifresini girmesi için bir alan, şifresini tekrar girmesi için bir başka alan ve şifre değişikliğini onaylamak için bir buton bulunmaktadır. Şifrenin unutulduğunun bildirilmesi kullanıcının mailini de gönderisinde (payload) içeren bir POST isteğiyle, değiştirilen şifrenin bildirilmesi işlemi de yine POST isteği aracılığıyla yapılmaktadır.

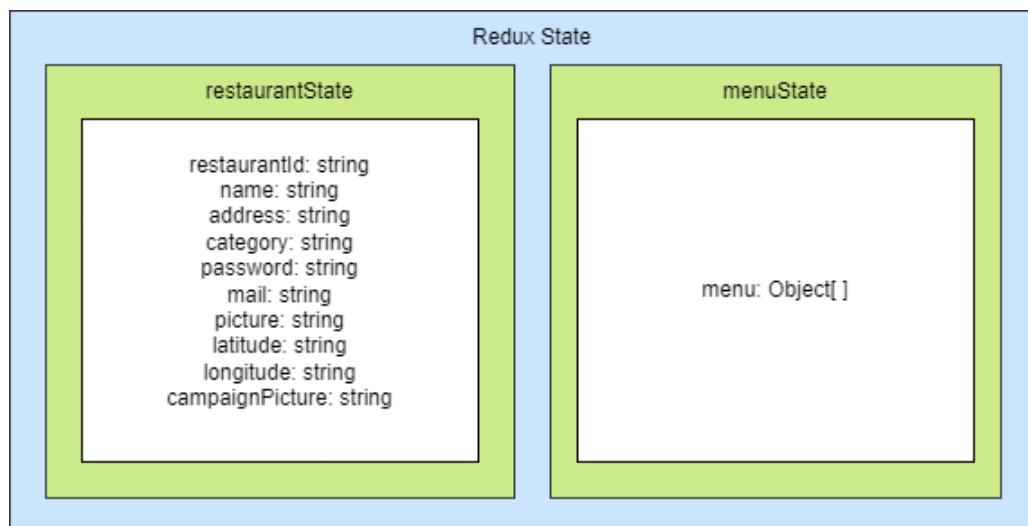
Kayıt olma sırasında kullanılan bilgilerin düzenlenmesi için kullanılan restoran bilgileri sayfasında kullanıcı, sitenin kayıt olma sayfasında bulunan her türlü restoran bilgisini güncelleyebilmektedir. Bu sayfada kullanıcı aynı zamanda restoranını Leaflet adlı bir kütüphane aracılığıyla sunulan hizmetler sayesinde haritada görebilmektedir. Kullanıcı güncellemek istediği bilgileri girdikten sonra güncelleme butonuna basarak PUT isteği aracılığıyla bilgilerini güncelleyebilmektedir. Kullanıcı eğer şifresini değiştirmek isterse şifre değiştirme butonuna basarak mailine gönderilen URL aracılığıyla şifre değiştirme işlemini gerçekleştirebilmektedir.

Siparişler sayfasında ise kullanıcı masalara gelen siparişlerin içeriğini ve fiyatını görüntüleyebilmektedir. Bu sayfadaki her siparişe bir numara verilmiştir ve sayfada bu siparişleri müşterilerin masa değiştirmesi durumunda farklı masalara taşıma seçeneği bulunmaktadır. Sayfadaki en önemli özellik ise kullanıcıya masanın durumunu bildiren simgelerdir. Bu simgeler sırasıyla, ödemenin yapılması durumunda yeşile dönen bir para simgesi, nakitle ödeme yapmak için garsonu çağırın bir müşteri olması durumunda turuncuya dönen bir garson simgesi ve masaya siparişin teslim edilmesi durumunda yeşile dönen bir yemek simgesinden oluşmaktadır. Ödeme yapılmasının ardından çalışanlar web sitesinde bulunan ödeme alındı butonuyla ödemeyi onaylayarak ödeme simgesinin yeşile dönmesini sağlayabilmekte ve siparişi teslim ettikten sonra teslim edildi butonuyla teslim simgesini yeşile döndürebilmektedir. Bunlar yapıldıktan sonra da kullanıcı siparişlerle ilgili başka bir iş kalmaması durumunda siparişi silmekteydi. Burada sipariş silme işlemi için DELETE, siparişleri web arayüzüne yansıtma için GET, kalan işlemler için ise POST isteği kullanılmıştır.

Masalar sayfası kullanıcıya masalar hakkında bilgi verme ve onları düzenleme imkanı tanımaktadır. Bu sayfada kullanıcı restoranının masa yönetimini sayfanın sunduğu masa ekleme, silme, isim değiştirme ve arama imkanlarıyla gerçekleştirebilmektedir. Sayfada aynı

zamanda her masaya özel olan QR kodları da görüntülenebilmekte ve indirilebilmektedir. Kullanıcının bu indirdiği QR kodunu gerekli masaya yerleştirmek suretiyle siparişleri görüntülemesi uygulamanın ana fikrini oluşturduğu için bu sistem kullanıcı için son derece basit hale getirilmiştir. Sayfada aynı zamanda masaların müsaitlik ve doluluk bilgileri de görüntülenebilmektedir. Normalde müsait olan bir masa bir müşterinin QR kodunu okuttuktan sonra sipariş vermesi durumunda sayfada dolu olarak gösterilmektedir. Bu sayfada da masaları web arayüzünde göstermek için GET, masa eklemek için POST, masa adı değiştirmek için PUT ve masa silmek için DELETE istekleri kullanılmıştır.

Menü düzenleme sayfası restoranın menüsünde müşterilere sunduğu ürünleri düzenlemesi işlevini yerine getirmektedir. Restoranın müşterilerle olan ilişkisi açısından çok kritik bir sayfa olmasından ötürü menü düzenlemek için sayfanın sunduğu araçların kullanımının basit olmasına dikkat edilmiştir. Bu sayfada menü kategorilere bölünmüştür. Menünün sağındaki kategori ekle butonuna basılarak yeni bir kategori oluşturulabilmekte ve kategori oluştuktan sonra sağdaki çöp kutusu butonuya da kategori silinebilmektedir. Bu kategorilerin altında ise yemekler bulunmaktadır. Eklenen kategorinin sağında yemek ekleme butonu ile yemek eklenebilmektedir. Yemeklerin her birisinin fotoğraf, isim ve fiyat bilgisi kullanıcı tarafından girilmekte ve aynı zamanda kullanıcı yemeğin sağındaki butonları kullanarak bu yemekler hakkındaki bilgileri güncelleyebilmekte, silebilmekte ve stokta kalmaması gibi durumlarda sipariş edilebilirliğini yemeği silmeden kaldırabilmektedir. Bu sayfada bileşen yapısı hem geliştirme aşamasını hem de kullanıcının sayfadaki kullanıcı tecrübesini iyileştirmek amacıyla yoğun bir biçimde kullanılmıştır. Menünün altındaki her kategori birer bileşen, bu kategorilerin oluşturdukları yemekler de aynı şekilde birleşendir. Bu sayede kod tekrarı düşürülmüş ve geliştirme açısından kolaylıklar sağlanmıştır. Bu sayfa da öbür sayfalar gibi güncelleme işlemleri için PUT, ekleme işlemleri için POST, silme işlemleri için DELETE, veritabanındaki yemekleri web arayüzünde görüntülemek için ise GET isteklerini kullanmaktadır. Menü düzenleme sayfası diğer sayfalardan farklı olarak her menü değişikliğinde verilerin ekraná tekrar işlenmesi (*render*) sırasında kısa süreli kaybolmasının önlenmesi için Redux'da menuState altında güncellenmektedir.

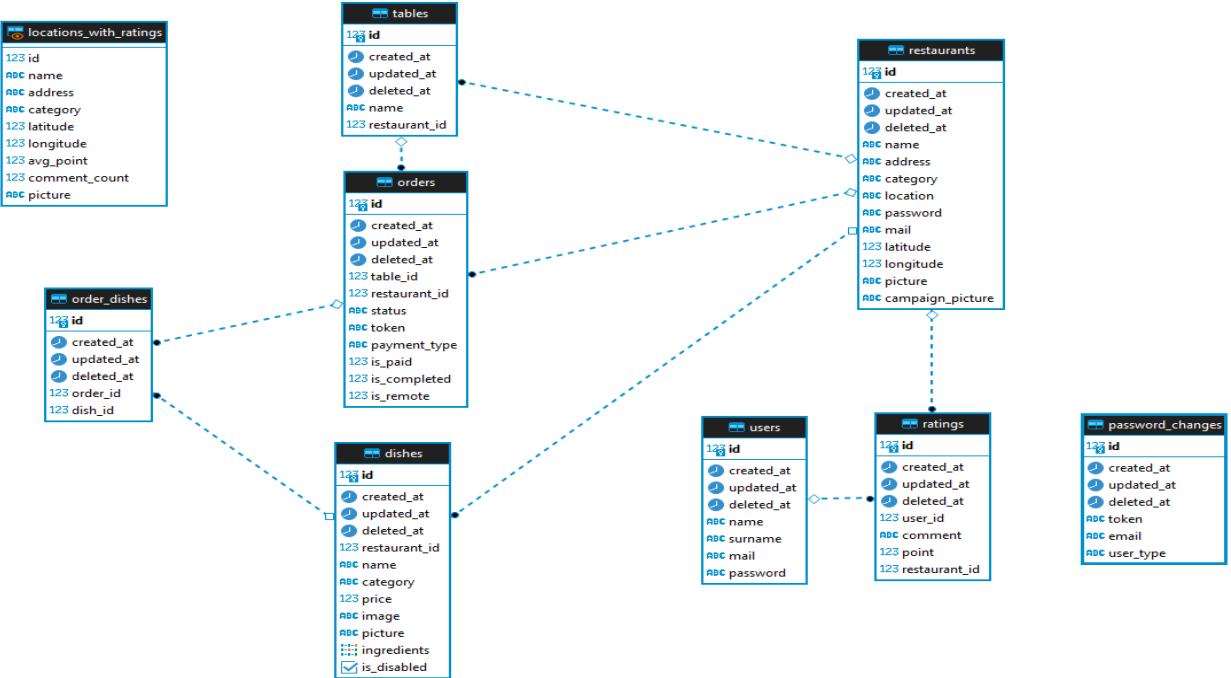


Web arayüzündeki son sayfa olan yorumlar sayfasında ise kullanıcı restoran hakkında yapılan yorumları görüntüleyebilmekte, yorumları puanlarına göre filtreleyebilmekte ve aynı zamanda verilen puanların ortalamasını görebilmektedir. Bu sayfada yalnızca görüntüleme işlemleri yapıldığı için GET isteğin kullanım tek başına yeterli olmuştur.

Web sitesindeki bir diğer önemli unsur da sayfalardaki navigation çubuğuudur. Bu çubuk aracılığıyla kullanıcılar yukarıda bahsedilen sayfalar arasında geçiş yapabilmekte, modern web sitelerinde olduğu gibi çubuğun sağında bulunan profil fotoğrafına basmaları durumunda açılan menü aracılığıyla sayfanın dil ayarını değiştirme, çıkış yapma ve restoran düzenleme sayfasına gitme işlemlerini gerçekleştirebilmektedir. Çubuktaki sayfa geçiş butonlarına ek olarak AFİYET yazısına tıklanması durumunda restoran bilgileri sayfasına geçilebilmekte, AFİYET logosuna basılması durumunda da siparişler sayfası görüntülenebilmektedir.

## 2.4. Veri Tabanı Tasarımı ve Entegrasyonu

Projede kullanılması için PostgreSQL [13] seçilmiştir. Bu seçimin arkasındaki neden PostgreSQL veri tabanının açık kaynak kodlu ve bedava bir yazılım olmasıdır. Veri tabanı uygulama geliştirme süresince ElephantSQL [14] servisi ile bulutta çalışmaktadır ve böylece uygulamanın geliştirilmesi sırasında tüm ekip üyeleri istedikleri yerden örnek bir veri tabanına ve içerisinde işlem yapabilecekleri yeterlilikte veriye kolayca erişebilmektedirler. Uygulama geliştirme sürecinin sonunda veri tabanının yine bir bulut servisi ile ayakta tutulması planlanmaktadır. Veri tabanında tutulacak tablolar Şekil 2.3’de gösterildiği gibi tutulmaktadır. Bu tablolar uygulamanın geliştirilmesi süresince Gorm [15] adlı ORM’nin otomatik migrasyon özelliği sayesinde sık sık ihtiyaca göre değiştirilmekte ve geliştirilmektedir.



Şekil 2.16: Veritabanı ER Tablosu

User tablosu uygulama kullanıcılarının uygulamayı kullanmaları için gereken isim soyisim, şifre ve mail bilgilerini tutar, şifre saltlanmış ve hashlenmiş bir şekilde tutulmaktadır. PasswordHash alanının 71 byte binary uzunluğunda olmasının sebebi kullanmayı planladığımız Bycrpyt [16] isimli şifreleme kütüphanesinin gereksinimlerinden dolayıdır. Bycrpyt arka planda blowfish algoritmasını kullanarak salt ve hash işlemlerini kendi içerisinde yaparak güvenlik açıklarını minimuma indirir ve veri tabanında tutulacak olan şifre hashini tutmamızda kolaylık sağlar.

Restaurant tablosu User tablosunun restoran karşılığıdır. Restoranların isim, adres, lokasyon(koordinat halinde), şifre, restoran resmi, kampanya resmi ve tür bilgilerini tutar. restoran resmi ve kampanya resimleri, resimlerin tutulduğu cdn URL'lerini tutar. restoran şifresi user şifreleri ile aynı mekanizmalar ile güvence altına alınır. Dish, Table ve Order tabloları ile “one to many” ilişkiye sahiptir.

Dish tablosu restoranların menülerinde bulunan yemeklerin bilgilerinin tutıldığı tablodur. RestaurantID yabancı anahtar görevi görmektedir. Name yemeğin ismini, Ingredients yemeğin içinde bulunan malzemeleri karakter dizisi şeklinde tutmaktadır. Category yemek türü, Picture ise yemeğin fotoğrafının cdn urlini tutar. Yemeğin fiyatı da bu tablo içerisindeki Price sütununda tutulur. Order tablosu ile OrderDish tablosu aracılığı ile “many to many” ilişkiye sahiptir.

Table tablosu restoranlarda bulunan masaların bilgilerini bulundurmaktadır. Bu tablo içerik olarak basit bir tablodur ve RestaurantID yabancı anahtarı dışında sadece masayı diğerlerinden ayırmak için kullanılan ve karakter dizisi şeklinde saklanan TableName sütununu bulundurmaktadır. Table tablosu Order tablosu ile “one to many” ilişkiye sahiptir

ve restoranların masalarda bulunan siparişleri takip etmelerini sağlayan ana mekanizma bu ilişki üzerinden çalışır.

Order tablosu bir restoranda o an bulunan siparişlerin tutulduğu tablodur. Bu tablo Restaurant ve Table tablosuna yabancı anahtarlarla bağlıdır ve bu sayede restoranlar hem masalarda bulunan siparişleri hem de o an tüm restoran genelindeki siparişleri gözlemlayabilirler. IsPaid, IsCompleted, IsRemote siparişin sırasıyla ödenip ödenmediğini, tamamlanıp tamamlanmadığını ve uzaktan olup olmadığını gösteren bool tutan sütunlardır. PaymentType ise ödemeyi “CARD” veya “CASH” değerlerini tutar ödemeyi kartla mı nakit ile mi yapıldığını belirtir. Status “INPROGRESS”, “PAYMENT\_FAILED”, “PAYMENT\_ACCEPTED” değerlerini tutar ve siparişin ödeme durumunu belirtir. RestaurantID ve TableID yabancı anahtar görevini görmektedir. Dish Tablosu ile OrderDish tablosu aracılığı ile “many to many” ilişkiye sahiptir.

OrderDish tablosu Order tablosu ile saklanan bir siparişin içerisinde bulunan yemekleri saklayan tablodur. Bu tablonun ana amacı Order tablosunun Dish tablosu ile “many to many” ilişkisini sağlamaktır.

PasswordChanges tablosu şifre değişikliklerinin kaydedildiği tablodur. E-posta şifre yenileme mailinin gönderildiği mailldir. UserType şifre değiştiren hesabın kullanıcı mı restoran mı olduğunu tutar. Token ise şifre değiştirme sayfasının sadece bir defa kullanılmasını garanti eden bir karakter dizisidir.

Ratings tablosu kullanıcıların restoranlara yaptıkları yorumların kaydedildiği tablodur. Point verilen puanı, Comment ise yapılan yorumu tutar. RestaurantId ve UserId yabancı anahtar görevini görmektedir.

LocationsWithRatings uygulamanın harita kısmında kullanılacak gerekli bilgileri tutan bir viewdird. Restoran bilgileri ile Rating bilgilerini birleştirir.

## 2.5. Mobil Uygulama Tasarımı ve Geliştirilmesi

### 2.5.1 Tanım

Afiyet mobil uygulaması restoran arayan kullanıcılarımızın (müsteri) Afiyet'e kayıtlı olan restoranları keşfedebildikleri, menülerini görebildikleri, diğer kullanıcıların restoran hakkındaki düşüncelerini okuyabildikleri, yakındaki restoranlara rezerve yapabildikleri veya masaya oturarak sipariş verebildikleri, dilerlerse uygulama üzerinden kredi kartı dilerlerse nakit olarak ödeme yapabildikleri ve deneyimlerini paylaşmak için restoranı değerlendirebildikleri çevrimiçi bir yemek siparişi deneyimi uygulamasıdır.

## 2.5.2 Kapsam

Projemizin geliştirme süreci mobil uygulamamızın geliştirilmesi, gereksinimlerin belirlenmesi, gereksinimler ile en uyumlu teknoloji altyapısının seçilmesi, proje isterlerine göre uygulama sayfalarının kararlaştırılması, mimarinin tasarlanması ve uygulanması, güncel tasarım ilkeleri takip edilerek önyüz tasarımlarının oluşturulması ve uyumlu bir şekilde paketlenerek isterleri karşılayan bir mobil uygulama ortaya çıkarmayı kapsamaktadır.

## 2.5.3 Kullanılan Teknolojiler

### 2.5.3.1 React Native

Afiyet ekibi olarak uygulamamızı ulaşabildiğimiz kadar kullanıcıya ulaştırmayı hedefliyoruz. Bu hedef doğrultusunda mobil uygulama geliştiren ekip arkadaşı sayımız ve ekibin çeşitli teknolojilere olan aşinalığı da dikkate alındığında hedeflerimize görece en az eforla ulaşmamızı sağlayacak bir teknoloji arayışı başlamıştır. Bu arayışın sonucunda mobil uygulamamızın Meta tarafından geliştirilmiş olan açık kaynak kodlu React Native [17] uygulama geliştirme çerçevesi kullanılarak geliştirilmesi kararlaştırılmıştır. Uygulamamızı geliştirmek için React Native altyapısının seçilmesindeki en önemli etken Web teknolojileri (JavaScript, HTML, CSS) kullanarak yazılmış tek kaynak kodunu iOS ve Android işletim sistemlerine derlenebilme olanağıdır.

### 2.5.3.2 Expo

Uygulamamızın baştan sona geliştirme aşamalarında React Native tarafından da önerilen açık kaynak kodlu Expo [18] platformu kullanılmıştır. Expo, React Native çerçevesiyle oluşturulmuş mobil uygulamaları geliştirmek ve dağıtmak için bir dizi araç ve hizmet sağlayan bir platformdur. Geliştirme sürecini basitleştiren ve geliştiricilerin uygulamalarını hızlı bir şekilde oluşturmmasına ve test etmesine olanak tanıyan bir dizi özellik ve araç içermektedir. Bu özelliklerden bazıları şunlardır:

- Yerleşik bir kod düzenleyiciye ve canlı yeniden yükleme özelliklerine sahip bir geliştirme ortamı
- Önceden oluşturulmuş bileşenlerden ve API'lerden oluşan geniş bir kitaplığa (*library*) erişim
- Bir dizi test araçları ve uzaktan hata ayıklama (*debug*) desteği

Expo Platformundan hazır olarak alınıp kullanılan API'lar ve seçilme nedenleri aşağıdaki gibidir:

- **Expo-Camera:** Farklı işletim sistemlerine sahip mobil cihazların kameralarına erişebilmek için farklı kodlar yazmamak amacıyla Expo-Camera API kullanılmıştır.
- **Expo-Location:** Farklı işletim sistemlerine sahip mobil cihazların GPS bilgilerine erişmek için farklı kodlar yazmamak amacıyla Expo-Location API kullanılmıştır.

- **Expo-Localization:** Farklı işletim sistemlerine sahip mobil cihazların varsayılan dil seçenekleri bilgisine erişmek için Expo-Localization API kullanılmıştır.
- **Expo-BarcodeScanner:** Farklı barkod türlerini tarayabilmek ve içeriği bilgiyi bilgisayar tarafından anlaşılabılır forma dönüştürebilmek için görüntü işleme altyapısı yazılmasını gerektirmiştir, böyle bir özelliğin sıfırdan geliştirilmesinin projeye katacağı ek iş yükünü üstlenmemek için Expo-BarcodeScanner API kullanılmıştır.

#### **2.5.3.3 i18next**

Mobil uygulamamıza yerelleştirme özelliğinin eklenmesi ve bu sayede uygulamamıza Türkçe – İngilizce dil desteğini kazandırmak ve dil değişikliği işlemini program çalışırken (runtime) yapabilmek için en popüler yerelleştirme çerçevelerinden birisi olan i18next kullanılmıştır.

#### **2.5.3.4 Axios**

Mobil uygulamamız, kiraladığımız sunucularımızdaki kod parçalarıyla HTTP protokolü ile iletişim kurmaktadır. Bu iletişimın sağlanabilmesi için Axios HTTP istemcisinin kullanılmasına karar verilmiştir. Axios'un seçilmesinde sağladığı JavaScript promise desteği, otomatik JSON dönüşümü, fotoğraf / resim gönderimi için FormData ve Blob desteği etkili olmuştur.

#### **2.5.3.5 Redux**

React ve React Native çerçeveleri, performans çekincelerinden dolayı ekrandaki bileşenlerin tekrar işlenmelerini söz konusu bileşenleri kontrol eden değişkenlerin değerlerinin değişmesine sağlamıştır. Bu değişkenlere React ve React Native geliştiricileri “state” adını vermiştir. Bu “state”ler bileşenlere özgü olup global erişilebilir değildir. “State”lerin başka bileşenler tarafından erişilmesini sağlamak için zincir halde bir bileşenden diğer bileşene aktarılması gerekmektedir. Bileşen zinciri çok uzun olduğunda uygulama kodlarını sürdürülebilir tutmak zorlaşmaktadır. Bu sebeple uygulama genelinde “state” kontrolünü kolaylaştırmak için Redux [21] kütüphanesi kullanılmıştır.

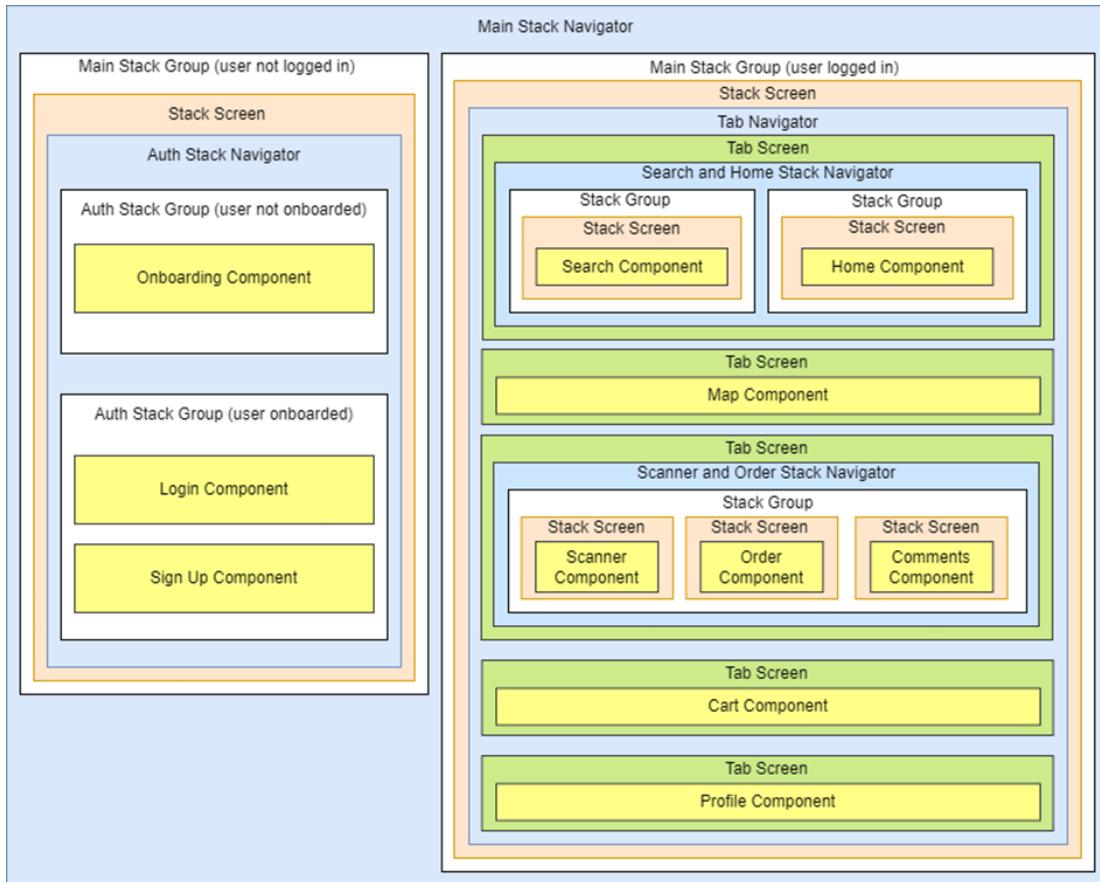
#### **2.5.3.6 React Navigation**

Mobil uygulamamızın sayfa geçişlerinin sağlanması ve sayfalar arası veri aktarımının yapılabilmesi için React Router ve React Navigation [19][20] projeleri karşılaştırılmış, mobil uygulamalar için daha optimize olduğu konusunda karar verilen React Navigation projesi seçilmiştir. React Navigation içerisinde birçok geçiş mekanizması barındırmaktadır. Bunlardan “Tab Navigation” ve “Stack Navigation” seçilmiş ve kullanılmıştır.

### **2.5.4 Ekran Mimarisi**

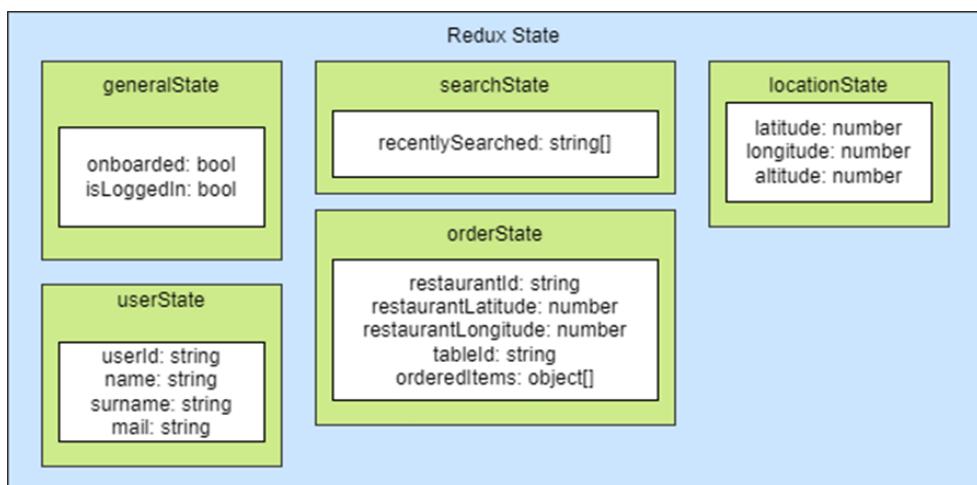
Yığın (*Stack*) ve Sekme (*Tab*) ekran geçiş mekanizmaları (*Navigator*) çocuk bileşen olarak bulundurdukları sayfalar (*Screen*) arasında geçişini sağlamaktadır. Kullanıcının tanıtım ekranlarını (*Onboarding Component*) veya kendi hesabına giriş yapmadığı durumlarda diğer

sayfalara erişimini kısıtlamak için birden fazla ekran geçiş mekanizması kullanılmıştır. Örneğin, hesabına giriş yapmamış fakat tanıtım ekranlarını geçmiş bir kullanıcı “Ana Yığın Grubu” içerisindeki (*Main Stack Group*) “Auth Yığın Ekran Geçiş Mekanizması”ndaki (*Auth Stack Navigator*) ikinci grupta bulunan bileşenleri görecektir.



Şekil 2.17: Ekran Mimarisi

### 2.5.5 Redux Veri Deposu Mimarisi



Şekil 2.18: Redux Veri Deposu Mimarisi

## **2.5.6 Bileşen (*Component*) Tasarımı**

Raporun mobil uygulama ile ilgili bölümünde kullanıcının ekranda gördüğü uygulama bileşenlerinden de ekran olarak bahsedilecektir. Bu ekranlar özünde kendi içindeki bileşenleri kapsayan bileşenlerdir ve Ekran Mimarisi başlığı altında olan figürde sarı renkli “Component”lara denk gelmektedirler. Ekran geçiş mekanizmalarındaki Yığın Ekran (Stack Screen) ve Sekme Ekran (*Tab Screen*) yapılarıyla karıştırılmamalıdır.

### **2.5.6.1 Karşılama (*Onboarding*) Ekranları**

Karşılama ekranları içlerinde birer resim, ana başlık ve açıklama barındıran beş adet kayar ekran, kullanıcının hangi ekranda olduğunu gösteren koyulaşan noktalar, bütün ekranları geçmeyi sağlayan “Atla”, ekranları tek tek geçmeyi sağlayan “Geç” ve sadece son ekrana gelindiğinde görünen “Bitir” butonlarından oluşmaktadır. Karşılama ekranlarında bulunan resimler derlenmiş uygulama dosyalarında yüklü olarak gelmektedir. “Bitir” butonuna dokunulduğunda Redux “state”lerinden “onboarded” değişkeni “true” değerini almakta ve “Auth Yığın Ekran Geçiş Mekanizması”ndaki ikinci grup ekranda görünülmektedir.

### **2.5.6.2 Giriş Yap Ekranı**

Giriş Yap ekranının genel tasarımları figürde görüldüğü gibidir. E-posta girilen “TextInput” bileşeni girilen e-postanın format olarak düzgün olup olmadığını düzenli ifade (*regular expression*) kullanarak test etmekte, geçersiz e-posta girilirse uyarı mesajını görünür kılmaktadır. Şifre girme alanı boş bırakılamaz olarak tanımlanmış ve boş bırakıldığı takdirde uyarı mesajını görünür kılmaktadır. Kullanıcı tarafından girilen bilgiler kullanıcının “Giriş Yap” butonuna dokunmasıyla arka uca gönderilmiştir. Arka uçtan dönen cevap olumsuz ise ekranın alt kısmında hata uyarısı görünür olmakta, dönen cevap olumlu ise cevapta bulunan veriler Redux’da “userState” içerisindeki ilgili bölgelerde kullanıcının uygulamayı kullanma süresi boyunca bu veriye ihtiyacı bulunan bileşenlere sunulmak üzere saklanmaktadır. “Şifrenizi mi unuttunuz?” yazısına dokunulması sonucunda e-posta alanının doğru bir şekilde dolu olması kontrol edilmekte, doğru olması durumunda arka uca istek göndererek kullanıcının e-posta adresine şifre değişikliği talebinin gönderilmesini sağlamaktadır. Girilen e-posta bilgisi düzenli ifade (*regular expression*) ile test edilen formata uygun değilse hata mesajı görünür olmaktadır. Kullanıcının “Kayıt Ol” butonuna dokunmasıyla yığın ekran geçiş mekanizması “Kayıt Ol” ekranını yığının en üstüne taşımakta ve ilgili ekranın görünür kılınmasını sağlamaktadır.

### **2.5.6.3 Kayıt Ol Ekranı**

Kayıt Ol ekranı kullanıcıdan ad, soyad, e-posta, şifre, şifre tekrarı bilgilerini eksiksiz, tutarlı, beklenen formata uygun ve güvenlik parametrelerine uygun olarak beklemektedir. Kullanıcı tarafından girilen bilgilerin kabul edilebilirliği ilgili bilginin girildiği alanın en sağındaki çarpı veya tik işaretleriyle belirtilmektedir. Kullanıcının “Kayıt Ol” butonuna dokunmasıyla girilen bilgilerin kabul edilebilirliği tekrar kontrol edilmekte, eğer bütün bilgiler bekendiği gibi ise arka uca istek atılmakta, değilse hata mesajı çıkarmaktadır. Kullanıcı kayıt olduktan

sonra “Giriş Yap” butonuna dokunarak yiğin ekran mekanizmasının Giriş Ekranı’nı en tepeye getirmesini sağlamaktadır.

#### **2.5.6.4 Ana Ekranı**

Ana ekran kullanıcının “Giriş Yap” ekranını geçtikten sonra varsayılan olarak geldiği ekrandır. Bu ekranda arama çubuğu, restoranların kampanyaları, öne çıkarılan restoranlar ve kullanıcının GPS konumuna dört km yakınıkta olan restoranlar görülmektedir. Kullanıcı kampanya fotoğraflarına veya restoranlardan birine dokunarak ilgili restoranın sayfasına gidebilir. Arama çubuğu dokunulduğunda arama sayfasına yönlendirilir. Ana sayfada gösterilen restoranlar listesi arka uçtan alınmakta, alt alta restoran kartları olarak gösterilmektedir. Yakındakı restoranlar ise aynı listenin kullanıcının GPS konumuna dört km yakınılığında olan restoranları içerecek şekilde filtrelenmesi sonucu kullanıcıya gösterilmektedir. Arama çubuğu ve restoranlar arasında kalan bölümde yakındakı restoranların kampanya fotoğrafları gösterilmektedir. Söz konusu kampanya fotoğraflarına dokunularak ilgili restoranın sipariş sayfasına gidilebilmektedir.

#### **2.5.6.5 Arama Ekranı**

Arama ekranında arama çubuğu ve Redux’da “searchState” içerisinde bulunan “recentlySearched” dizisindeki geçmişte yapılmış aramalar gösterilmektedir. Arama çubüğunda aranan her sonuç “recentlySearched” dizisine kayıt edilir. Arama çubüğunda aranan kelimeler arka ucta aranan harflerin bulunduğu bütün restoranları mobil uygulamaya cevap olarak döner. Dönen cevaplar ekranda bir liste olarak gösterilir. Kullanıcı listelenmiş restoran kartlarından birine dokunarak ilgili restoranın sayfasına gidebilir.

#### **2.5.6.6 Harita Ekranı**

Harita ekranı ilk açıldığında arka uca bütün restoranların listesini almak için istek atılmaktadır. Arka uçtan dönen cevap her restoran için içerisinde restoranın adı, kategorisi, koordinatları, fotoğrafı, yorum sayısı, ortalama puanı, restoranId’si ve adres bilgilerini barındırmaktadır. Restoranların bilgileri alındıktan sonra Google Maps haritasının ilgili koordinatlarına harita iğnesi koyulmakta ve ekranın alt kısmında cevapta dönen her restoran için restoran kartı oluşturulmaktadır. Bu kartlarda koordinatlar hariç diğer bilgiler gösterilmektedir. Kullanıcı bu kart listesini yatay eksende kaydırarak haritadaki iğneler arasında gezebilmekte, dilerse kartın altında bulunan “Restoranı Gör” butonuna dokunarak ilgili restoranın sayfasına gidebilmektedir.

#### **2.5.6.7 Tarayıcı Ekranı**

Tarayıcı ekranının ilk açılışında mobil cihazın sistem ayarlarından kamera kullanımı için Afiyet uygulamasına izin verilip verilmediği kontrol edilir. Afiyet uygulamasının cihaz kamerasına erişim izni yoksa kullanıcından izin istenmekte, halihazırda izin verilmişse cihaz kamerası uygulama içerisinde açılmaktadır. Afiyet uygulaması telefonun okutulduğu barkod çeşitlerinden sadece QR kod türünde olanları anlamdıracak şekilde programlandırılmıştır.

Restoran arayüzü tarafından oluşturulan QR kodlar “**“restoranId:masaId”** şeklinde kodlanmıştır. Afiyet QR formatına uygun QR kod kamera tarafından okunduğu takdirde tarayıcı ekranı kullanıcıyı ilgili restoranın sayfasına yönlendirmeden önce QR kodun okutulduğu masa uzaktan sipariş vermiş birisi tarafından rezerve edilmiş olup olmadığını kontrol etmektedir. Eğer masa uzaktan sipariş vermiş birisine rezerve edilmişse bu durumu kullanıcıya bildiren bir hata mesajı görünür olmakta, değilse barkodun içeriği restoranId ve masaId bilgisi Redux’da “orderState” içerisindeki ilgili bölmelere kayıt edilmekte ve kullanıcı QR kodu okutulan restoranın sayfasına yönlendirilmektedir.

#### **2.5.6.8 Sipariş Ekranı**

Sipariş ekranında restoranın kullanıcıyı ilgilendirecek bütün bilgileri gösterilmektedir. Bu bilgiler restoran adı, fotoğrafı, kategorisi, adresi, yorum sayısı, ortalama puanı, kullanıcının konumuna olan uzaklığın kilometre cinsinden değeri ve menü içeriğidir. Sipariş ekranı açıldığında bir dizi hesaplama olmaktadır. Bu hesaplamalar restoranın kullanıcıya olan uzaklığının hesaplanması ve kullanıcı menü içeriğini görüntülemek için sipariş ekranını açık bıraktığı her 10 saniyede bir menüde değişiklik olup olmadığı kontrol edilmesidir. Menü içeriğinin kontrol edilmesi her 10 saniyede bir kullanıcıya gösterilen yemeklerin son güncelleme tarihi ile arka uçtan yeniden alınan yemeklerin son güncellenme tarihleri karşılaştırılarak hesaplanmaktadır. Hesaplamalar sonucunda arka ucta bulunan menünün herhangi bir bileşeninde değişiklik olduğu saptanırsa kullanıcıya menünün restoran tarafından güncellendiği bilgisi gösterilmekte ve menü en güncel versiyonla değiştirilmektedir. Sipariş ekranında siparişler yemek kartlarına dokunma sonucunda ekranın alt kısmından kayarak açılan çekmece aracılığıyla verilebilmektedir. Açılan çekmecede seçilmiş yemeğin adı, fotoğrafı, fiyat bilgisi gösterilmekte yemekten kaç adet sipariş verileceğinin belirtilebilmesi için bir sayaç ve sepete eklemek için bir buton bulunmaktadır. Sepete ekle butonuna dokunulduğunda Redux’da “orderState” içerisinde bulunan “orderedItems” dizisine seçilmiş olan yemek adı, birim fiyatı, kaç adet sipariş edildiği, yemekIdsi, restoranIdsi bilgilerini içeren bir obje olarak eklenmektedir. Restoran bir yemeğin tüketliğini bildirdiğinde sipariş ekranı ilgili yemeğin kartını gri renk olarak göstermektedir.

#### **2.5.6.9 Yorum Ekranı**

Yorum ekranında restorana kullanıcılar tarafından yapılmış değerlendirmeler gösterilmektedir. Kullanıcılar bu ekrana Sipariş ekranında bulunan puan ve inceleme sayısını gösteren yazıya dokunarak gelebilmektedirler. Yorum ekranı açıldığında arka uca restorana ait yorumları almak için istek gönderilmekte, bu isteğe dönen cevaptaki her yorum için yorum, tarih, puan, yorum yapan kullanıcının adı ve soyadı ekranda listelenmektedir. Yorum ekranının en tepesinde beş ile bir puan arasında toplamda beş adet filtre butonu bulunmaktadır. Bu butonlara kullanıcı tarafından dokunulduğunda ekranda listelenen yorumlar seçilen filtreye göre listelenmektedir. Ekranın altında bulunan “Yorum Ekle” butonuna dokunulduğunda kayarak açılan bir çekmece ekran kullanılarak yemek deneyimi paylaşılabilir ve restoran puanlanabilir.

### **2.5.6.10 Sepet Ekranı**

Sepet ekranı iki bölümden oluşmaktadır. İlk bölümde sepete eklenen yemekler adetleriyle birlikte görünmekte, kullanıcı dilerse sayaç kullanılarak adetleri değiştirebilmekte veya ilgili yemeğin kartını sola kaydırarak sepetten silebilmektedir. Sepet ekranının alt kısmında kredi kartı ile ödeme veya nakit olarak ödeme yöntemlerinin seçilebilmesi için radyo butonları bulunmaktadır. Uzaktan sipariş veren kullanıcılar için sadece kredi kartı, restoranda QR kod okutup sipariş veren kullanıcılar için her iki ödeme yöntemi de kullanılabilir şekilde tasarlanmıştır. Ekranın alt kısmında “Sepet Durumunu Kontrol Et” butonu bulunmaktadır. Bu butonun işlevi, kullanıcının yemekleri sepete ekleme zamanından sonra restoran tarafından fiyatı veya tükenmişlik durumu değiştirilen ürünlerin var olup olmadığını kontrol etmek, eğer varsa kullanıcıya hangi ürünlerde hangi değişikliklerin yapıldığını gösteren bir uyarı ekranı çıkartmak ve sepette ilgili değişiklikleri otomatik olarak yapmaktadır. Kontroller bittikten sonra sepet ekranının ikinci bölümüne geçilir. Bu bölüme geçilmeden önce nakit ödeme yöntemi seçilmişse kullanıcıya dilediği zaman garson çağırması için bir buton gösterilir. Kredi kartı ile ödeme yöntemi seçilmişse arka uca kredi kartı ile ödeme altyapısının hazırlanması için bir istek atılır ve o siparişe özel bir token ile ödeme altyapısının URL bağlantısı cevap olarak alınır. Cevaptaki URL ekranada WebView bileşeni içerisinde açılarak React Native uygulamamızın içerisinde bir Web ekranının çalıştırılması sağlanır. Kullanıcı kendisinden istenen kredi kartı bilgilerini girdikten sonra ödeme başarılı olması durumunda ise kendisine ödemeyi başarıyla gerçekleştigini belirten bir sembol gösterilir.

### **2.5.6.11 Profil Ekranı**

Profil ekranı kullanıcı ile ilgili bilgilerin göründüğü ve uygulamaya özel dil değiştirme, destek ile iletişime geçme gibi özellikleri ve hesaptan çıkış yapma fonksiyonlarını barındırmaktadır. Kullanıcının adı, soyadı ve e-posta adresi uygulamaya giriş yapılırken Redux'a kaydedilen verilerden alınmıştır. Kullanıcının bulunduğu konumun şehir ve ülke bilgisine dönüştürülmesi Expo-Location API'ından alınan koordinat bilgilerinin tersine işlenerek adrese çevrilmesi ile yine aynı API'nin farklı bir fonksiyonu tarafından gerçekleştirılmıştır. “Şifre Değiştir” butonu dokunulduğunda kullanıcının sisteme kayıtlı e-posta adresine tipki “Şifremi Unuttum” mantığındaki gibi bir şifre değişikliği talebi gönderilmesi ile tasarılmıştır. Dil değiştir butonuna dokunulduğunda i18next kütüphanesi kullanılarak uygulama genelindeki bütün yazılar önceden çevirileri hazırlanmış birer JSON dosyasından okunarak Türkçe ve İngilizce arasında değiştirilmektedir. “Destek” butonuna dokunulduğunda mobil cihazın varsayılan e-posta istemcisi alıcımasına Afiyet şirket destek hesabının adresi yazılı bir şekilde açılmaktadır. “Çıkış Yap” butonuna dokunulduğunda Redux'da kayıtlı olan bütün kayıtlar sıfırlanmakta ve kullanıcı giriş ekranına yönlendirilmektedir.

## **3. SONUÇLAR**

Bölüm “1.8. Proje Planı”nda anlatıldığı gibi ortaya konulan plana sadık kalınmış ve proje sonu için planlanan hedefler gerçekleştirılmıştır.

- Düzenli olarak ana daldan (*main branch*) dağıtım (*deployment*) yaparak her an çalışır bir yazılım ortaya koymak için altyapı (*infrastructure*) kurulumu üzerine deneyim elde edilmiştir.
- Katmanlı Mimarının program büyündükçe programcıya kazandırdığı okunabilirlik (*readability*) avantajı deneyim edilmiştir.
- Modern bir UI/UX dizaynı yapılmıştır.
- Modern teknolojiler kullanılarak tam fonksiyonlu bir uygulama geliştirilmiştir.
- Grup üyeleri proje içinde kullanılacak teknolojiler konusunda gerekli eğitimleri almış ve bu teknolojileri projede kullanmak için araştırmıştır.
- Web sitesi için React ve Javascript konularında araştırmalar yapılmıştır.
- Mobil Uygulama için React Native ve Redux araştırılmış ve üstüne eğitim alınmıştır.
- Arka Uç’da Golang dilinde geliştirme sağlanmış ve SQL diliyle alakalı araştırmalarda bulunulmuştur.

## 4. TARTIŞMA

Projenin geliştirmelerinin sonunda kullanıcıya sağladığı avantajlar;

- Hızlı bir sipariş süreci
- Restoranlar hakkında daha geniş bilgi yelpazesi
- Süreci insansız hale getirerek hata olasılığını azaltılması
- Restoran tarafından maliyet azalması

sayılabilir.

Ürünün dezavantajları ise;

- Sistem kısıtlarında da bahsedildiği gibi projenin kullanılması için internete ve gerekli kapasitede sunucuya ihtiyacı olması.
- Potansiyel kullanıcıların restoranlara hepsinde uygulamayı kullanabilmesi için birçok restoranın projeyi kullanmaya başlamaya ikna olması gerekmesi.

bunlarla örneklenirilebilir.

Proje geliştirmelerinin öğrencilere olan katkılarına örnek olarak;

- CDN kullanımından dolayı önbellek geçersiz kılma (*Cache Invalidation*) problemleri ile ilgilenilmiştir ve bunların çözüm yöntemleri öğrenilmiştir.
- Hızlı bir geliştirme sürecinde, önceki özelliklerini bozmadan yeni özellik geliştirme konusunda kod kalitesinin önemi anlaşılmıştır.
- React ve React Native teknolojilerinin farklarından ötürü farklı yapıarda benzer sistemlerin kurulmasının zorluğu kavranmıştır.

bunlar verilebilir.

## **5. ŞEKİLLER DİZİNİ**

Şekil No	Açıklama	Sayfa No
1.1	Sistemin Genel Yapısı	3
1.2	Sistemin Genel Yapısı-2	3
1.3	Sistemin Genel Yapısı-3	4
1.4	Güz Dönemi İş-Zaman Planı	5
1.5	Bahar Dönemi İş-Zaman Planı	5
2.1	Örnek bir PR Ekran	7
2.2	Örnek bir GitHub Projects	7
2.3	Örnek Bir Rapor	9
2.4	Veritabanı Metrikleri	11
2.5	Örnek Veritabanı Alarmı	11
2.6	Veritabanı Enstantaneleri	12
2.7	Müşteri Hizmetleri Etiketi Sonucu	12
2.8	Sunucu Metrikleri	13
2.9	Sunucu Alarmı	14
2.10	Afiyet Projesinin DNS Kayıtları	14
2.11	afiyet-media Obje Deposu	15
2.12	CloudFront Dağıtımları	16
2.13	Ödeme Sistemi Mekanizması	16
2.14	React Router Mimarisi	18
2.15	Redux Veri Deposu	20
2.16	Veritabanı ER Tablosu	22
2.17	Ekran Mimarisi	26
2.18	Redux Veri Deposu Mimarisi	26

## **6. SİMGELER ve KISALTMALAR DİZİNİ**

Simge / Kısaltma	Açıklama
QR	Quick Response Code
iOS	iPhone Operating System
GPS	Global Positioning System
PR	Pull Request
HTTP	Hypertext Transfer Protocol
ASP.NET	Active Server Pages Network Enabled Technologies
SQL	Structured Query Language
ORM	Object-relational mapping
REST	Representational State Transfer
API	Application Programming Interface
JSON	JavaScript Object Notation
UI	User Interface
CSS	Cascading Style Sheet
GIMP	GNU Image Manipulation Program
ER	Entity Relationship
HTML	Hyper Text Markup Language
SDK	Software Development Kit
DBA	Database Administrator
AWS	Amazon Web Services
RDS	Relational Database Service
CPU	Central processing unit
SES	Simple Email Service
TLS	Transport Layer Security
HTTPS	Hypertext Transfer Protocol Secure
CA	Certificate Authority
ACME	Automated Certificate Management Environment
S3	Simple Storage Service
CDN	Content Delivery Network
OAC	Origin Access Control
ACM	AWS Certificate Manager

## 7. KAYNAKLAR

- [1] Stack Overflow Developer Survey 2022. (2022). Stack Overflow.  
<https://survey.stackoverflow.co/2022/#section-version-control-version-control-systems>  
Son Erişim Tarihi: 31/12/2022
- [2] Lisowski, T. (2021, December 8). Top Git Hosting Services for 2022.  
<https://gitprotect.io/blog/top-git-hosting-services-for-2022>  
Son Erişim Tarihi: 30/12/2022
- [3] Why Google stores billions of lines of code in a single repository | Communications of the ACM. (2016). Communications of the ACM. <https://dl.acm.org/doi/10.1145/2854146>
- [4] Rob P. Concurrency (2014) Rob Pike - From Parallel to Concurrent [Video]. Youtube.  
[https://www.youtube.com/watch?v=iTrP\\_EmGNmw](https://www.youtube.com/watch?v=iTrP_EmGNmw)  
Son Erişim Tarihi: 20/10/2022
- [5] Go at Google- The Go Programming Language. (2012). Go.dev.  
<https://go.dev/talks/2012/splash.article>  
Son Erişim Tarihi: 20/10/2022
- [6] Merovius Router. (2017, June 18). Merovius.de; Mero's Blog.  
<https://blog.merovius.de/posts/2017-06-18-how-not-to-use-an-http-router>  
Son Erişim Tarihi: 10/10/2022
- [7] Ryer, M. (2018). How I write HTTP services after eight years by Mat Ryer - PACE. Pace Blog. <https://pace.dev/blog/2018/05/09/how-I-write-http-services-after-eight-years.html>  
Son Erişim Tarihi: 5/10/2022
- [8] Clean Architecture. (2021). Cleancoder.com.  
<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>  
Son Erişim Tarihi: 7/10/2022
- [9] Overview - Material UI. (2014). Mui.com.  
<https://mui.com/material-ui/getting-started/overview/>  
Son Erişim Tarihi: 11/11/2022
- [10] Basic concepts of flexbox - CSS: Cascading Style Sheets | MDN. (2022, September 28). Mozilla.org.  
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Basic\\_Concepts\\_of\\_Flexbox](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)  
Son Erişim Tarihi: 18/12/2022

[11] Getting Started – React. (2021). Reactjs.org. <https://reactjs.org/docs/getting-started.html>  
Son Erişim Tarihi: 21/12/2022

[12] Tutorial v6.6.1. (2023). Reactrouter.com. <https://reactrouter.com/en/main/start/tutorial>  
Son Erişim Tarihi: 22/11/2022

[13] PostgreSQL: Documentation. (2022). Postgresql.org. <https://www.postgresql.org/docs/>  
Son Erişim Tarihi: 19/11/2022

[14] Documentation | ElephantSQL. (2022). Elephantsql.com.  
<https://www.elephantsql.com/docs/>  
Son Erişim Tarihi: 4/12/2022

[15] Jinzhu. (2020, July 2). GORM Guides. GORM. <https://v1.gorm.io/docs/>  
Son Erişim Tarihi: 4/12/2022

[16] bcrypt packages. (2022). Go.dev. <https://pkg.go.dev/golang.org/x/crypto/bcrypt>  
Son Erişim Tarihi: 21/12/2022

[17] Introduction · React Native. (2022, September 6). Reactnative.dev.  
<https://reactnative.dev/docs/getting-started>  
Son Erişim Tarihi: 17/11/2022

[18] Expo Documentation. (2023). Expo Documentation. <https://docs.expo.dev/>  
Son Erişim Tarihi: 13/11/2022

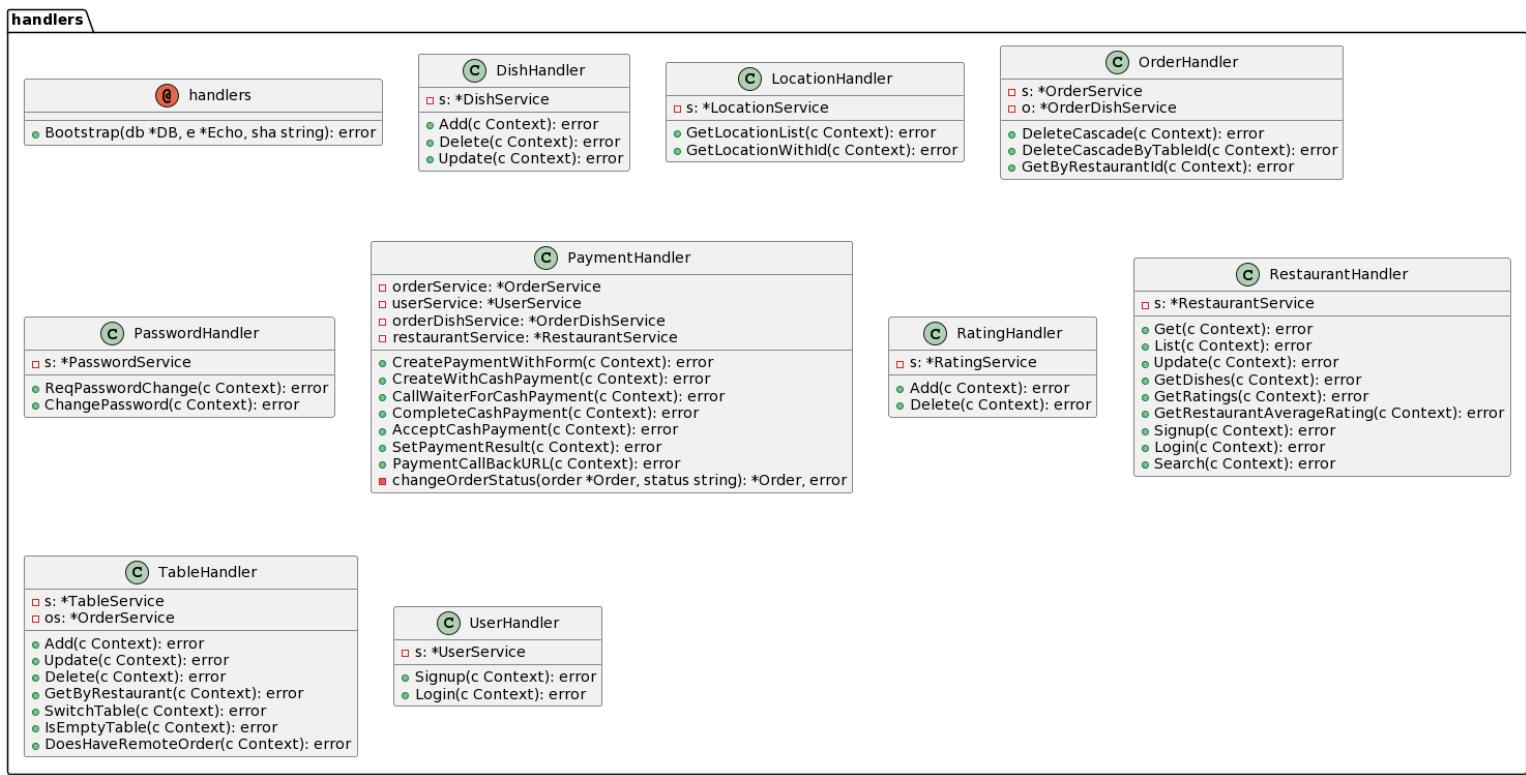
[19] Getting Started - React Navigation (2023). Reactnavigation.org.  
<https://reactnavigation.org/docs/getting-started/>  
Son Erişim Tarihi: 13/11/2022

[20] Main Concepts v6.6.1. - React Router (2023). Reactrouter.com.  
<https://reactrouter.com/en/main/start/concepts>  
Son Erişim Tarihi: 21/12/2022

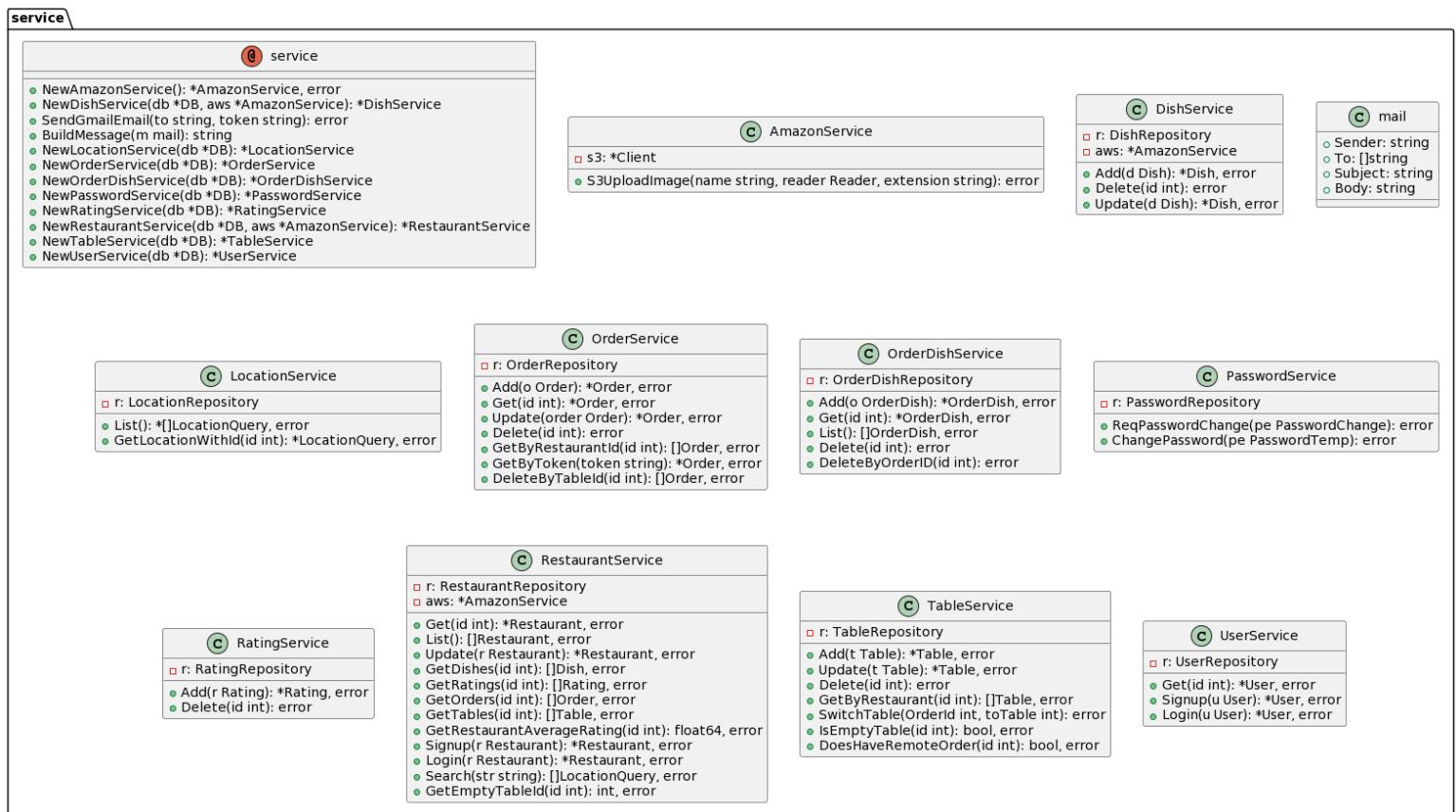
[21] Getting Started with Redux | Redux. (2022, August 18). Js.org.  
<https://redux.js.org/introduction/getting-started>  
Son Erişim Tarihi: 5/10/2022

## 8. EKLER

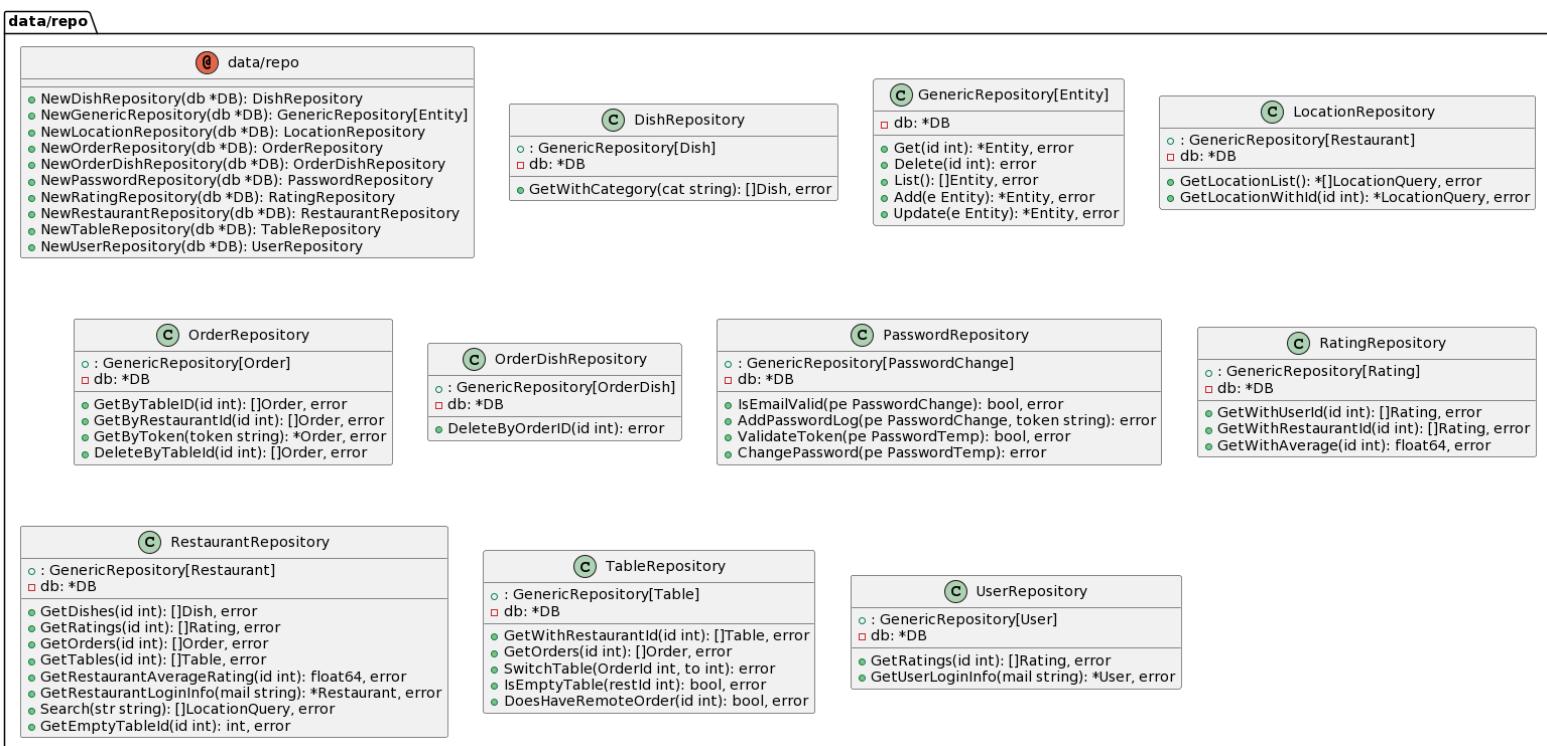
### EK-1 Handler UML Class Diagram



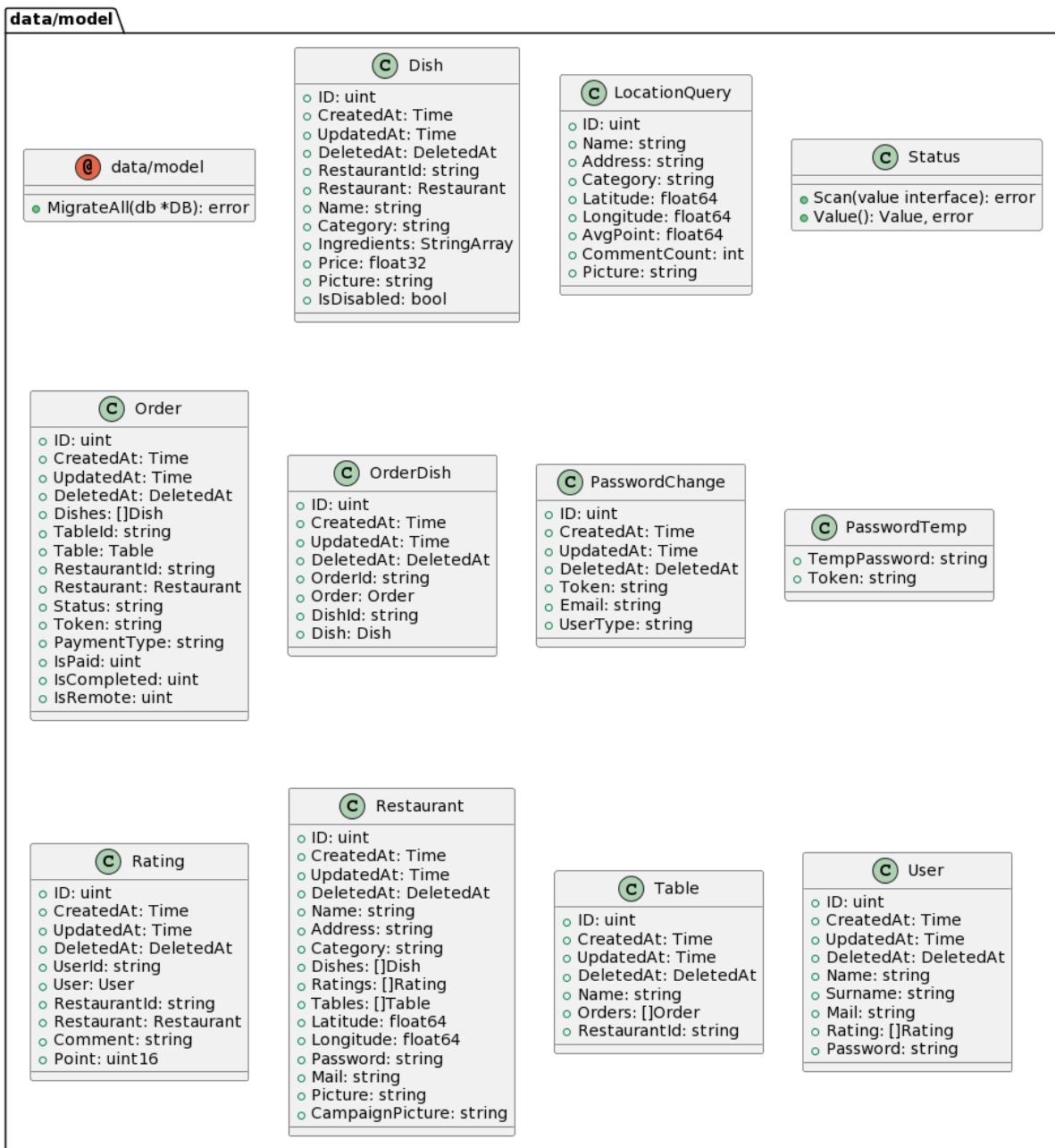
## EK-2 Service UML Class Diyagramı



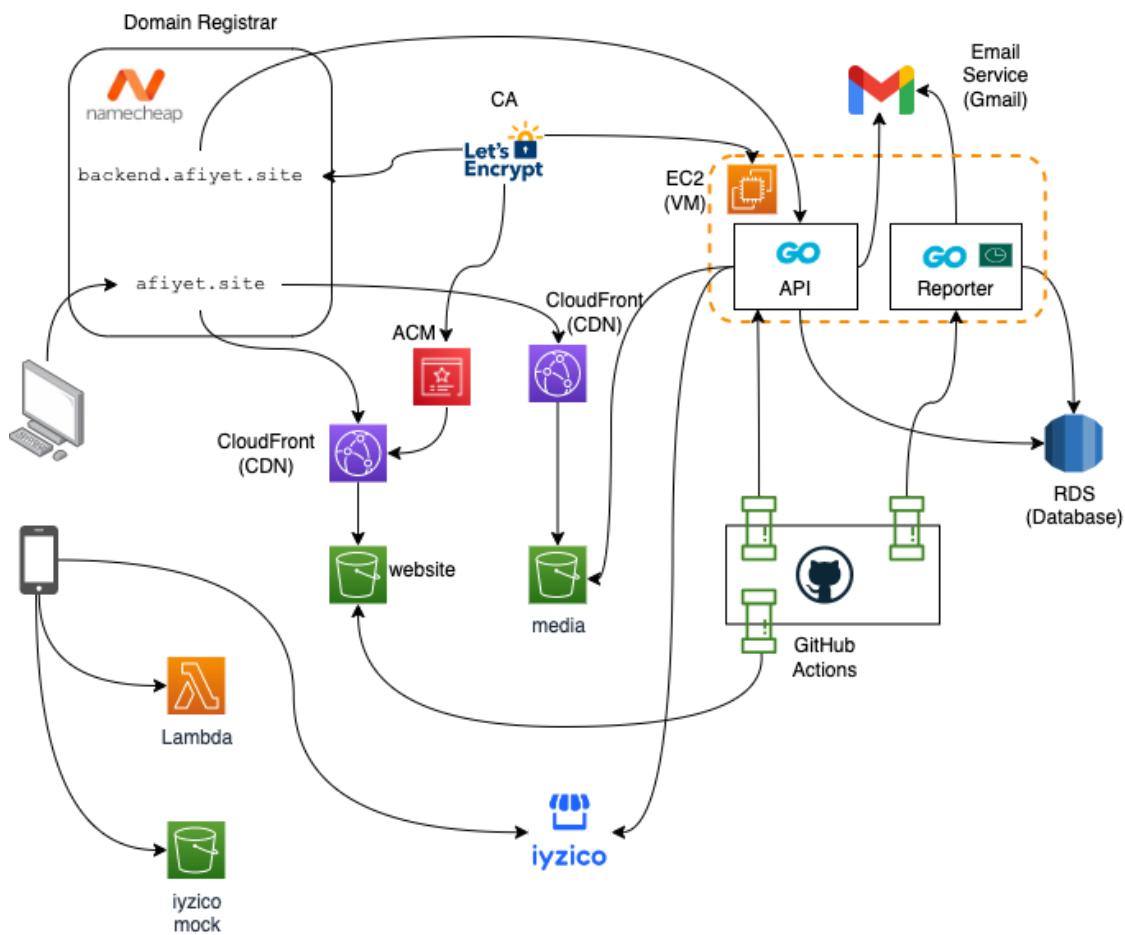
## EK-3 Repository UML Class Diyagramı



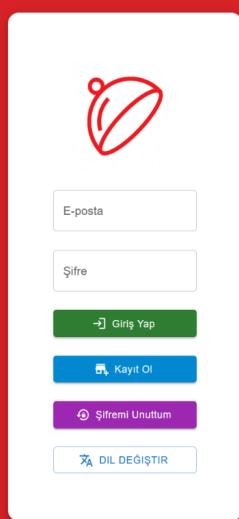
## EK-4 Model UML Class Diagram



## EK-5 Altyapı Mimarisi



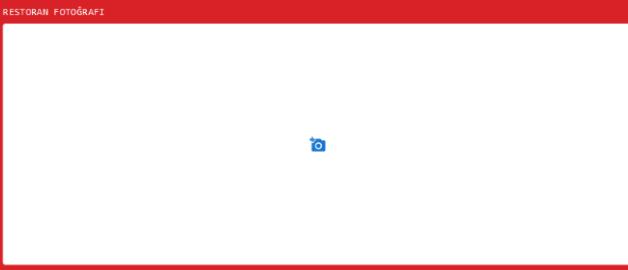
## EK-6 Giriş Yapma Sayfası



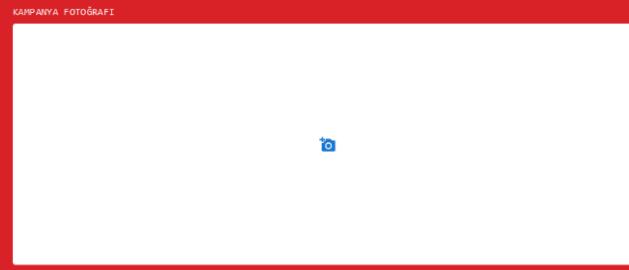
## EK-7 Kayıt Olma Sayfası

Kayıt Ol

RESTORAN FOTOĞRAFI



KAMpanya FOTOĞRAFI



Restoran Adı

Restoran Kategorisi

Restoran e-postası

Şifre

Restoran Adresi

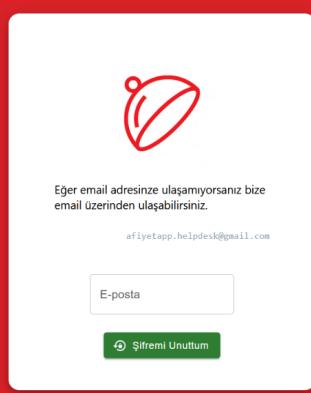
Enlem

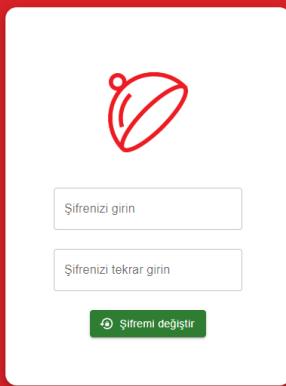
Boylam

 KAYIT OL

## EK-8 Şifremi Unuttum Sayfası



## EK-9 Şifre Yenileme Sayfası



## EK-10 Restoran Bilgileri Sayfası

AŞA  
AFİYET SİPARİŞLER MASALAR MENÜ DÜZENLEME YORUMLAR

Restoran Bilgileri

**RESTORAN FOTOĞRAFı**

**KAMPANYA FOTOĞRAFI**

Restoran Adı: Bang Aspava, Çankaya (Çayolu Yaşamkent Mah.)  
Restoran Kategori: Kebap

Restoran e-postası: afiyetbusiness@gmail.com  
Restoran şifre: 32 64522323165274  
Restoran Adres: Yaşamkent Mh., 3207. Cd. No:11, 06810 Yenimahalle/Çankaya/Ankara  
Restoran Telefon: 39 86987830224515  
Restoran Boylari: 32 64522323165274

**GÜNCELLE**

**MAP**

## EK-11 Siparişler Sayfası

The screenshot shows the Afiyet app's order management screen. At the top, there's a red header bar with the Afiyet logo, menu items like 'SİPARİŞLER', 'MASALAR', 'MENÜ DÜZENLEME', and 'YORUMLAR', and a search bar labeled 'Masa Ara'.

The main area is titled 'Siparişler' (Orders) and displays three separate order sections:

- bahçe-1 Siparişleri**:
  - Sipariş 171: Yemeğe (Beyti Kebap) - 1 adet, Toplam (TL) 155.
  - Masa Toplamı (TL): 155.
- teras-1 Siparişleri**:
  - Sipariş 173: Yemeğe (Mercimek Çorbası, İskender Eti Döner, Fırında SSK Eti Döner Dürüm, Coca Cola (330 ml)) - 2 adet, Toplam (TL) 431.
  - Masa Toplamı (TL): 0.
- köşe-1 Siparişleri**:
  - Masa Toplamı (TL): 0.

Each order section includes buttons for 'SİPARİSLERİ BAŞKA MASAYA TAŞI' (Move to another table), 'TESLİM EDİLDİ' (Delivered), and 'SİL' (Delete).

## EK-12 Sipariş Taşıma Sayfası

The screenshot shows the Afiyet app interface for managing orders. The main screen displays three order lists: 'bahçe-1 Siparişleri', 'teras-1 Siparişleri', and 'köşe-1 Siparişleri'. A modal dialog titled 'Siparişleri Başka Masaya Taşı' (Move Orders to Another Table) is open over the 'teras-1' list. This dialog has two input fields: 'Taşınan Masa' (Moving Table) set to 'bahçe-1' and 'Masa Seç' (Select Table) with a dropdown menu showing 'Masa 5'. It also contains a checkbox labeled 'Sipariş 171' and two buttons at the bottom: 'İPTAL' (Cancel) and 'TAŞI' (Move). The background shows the total order amounts for each table: bahçe-1 (165), teras-1 (431), and köşe-1 (0).

## EK-13 Masalar Sayfası

A screenshot of a web-based restaurant management system. At the top, there is a red header bar with the logo 'AFİYET' on the left, followed by menu items: 'SİPARİŞLER', 'MASALAR', 'MENÜ DÜZENLEME', and 'YORUMLAR'. On the far right of the header is a user profile icon. Below the header, there is a search bar labeled 'Masa Ara' and a blue button labeled '+ MASA EKLE'. The main content area displays three QR codes, each associated with a table name and a small circular status indicator:

- The first QR code is labeled "bahçe-1" and has a red circular status indicator.
- The second QR code is labeled "teras-1" and has a red circular status indicator.
- The third QR code is labeled "köşe-1" and has a green circular status indicator labeled "Masafat".

Each QR code is accompanied by a set of small icons for managing the table.

## EK-14 Menü Düzenleme Sayfası

AŞİYET SİPARİŞLER MASALAR MENÜ DÜZENLEME YORUMLAR

**Menü**

**İçecek**

<b>Kategori Adı</b> İçecek	<b>Yemek Adı</b> Sprite (330 ml)	<b>Fiyatı</b> 25	<b>İçindekiler</b> İçindekiler	<b>YEMEK EKLE</b>
	<b>Yemek Adı</b> Ayran (270 ml)	<b>Fiyatı</b> 15	<b>İçindekiler</b> İçindekiler	
	<b>Yemek Adı</b> Soda (200 ml)	<b>Fiyatı</b> 15	<b>İçindekiler</b> İçindekiler	
	<b>Yemek Adı</b> Şalgam Suyu (250 ml)	<b>Fiyatı</b> 25	<b>İçindekiler</b> İçindekiler	

**Tavuk Tantuni**

<b>Kategori Adı</b> Tavuk Tantuni	<b>Yemek Adı</b> Tavuk Tantuni Dürüm	<b>Fiyatı</b> 60	<b>İçindekiler</b> Tavuk,Lavaş,Soğan,Domates,Maydanoz	<b>YEMEK EKLE</b>
	<b>Yemek Adı</b> Somun Ekmek Arası Tavuk Tantuni	<b>Fiyatı</b> 60	<b>İçindekiler</b> Tavuk,Ekmek,Soğan,Domates,Maydanoz	

**Et Tantuni**

<b>Kategori Adı</b> Et Tantuni	<b>Yemek Adı</b> Et Tantuni Dürüm	<b>Fiyatı</b> 90	<b>İçindekiler</b> Et,Lavaş,Soğan,Domates,Maydanoz	<b>YEMEK EKLE</b>
	<b>Yemek Adı</b> Somun Ekmek Arası Et Tantuni	<b>Fiyatı</b> 90	<b>İçindekiler</b> Et,Ekmek,Soğan,Domates,Maydanoz	

## EK-15 Yorumlar Sayfası

The screenshot shows a red-themed website for 'AFİYET'. At the top, there's a navigation bar with icons for a magnifying glass, a person, and a lock, followed by the word 'AFİYET' and links for 'SİPARİŞLER', 'MASALAR', 'MENÜ DÜZENLEME', and 'YORUMLAR'. On the right side of the header is a small circular profile picture. Below the header, the main content area has a white background and a red border. It features a title 'Yorumlar' and a row of five smiley face rating icons. To the right of these icons is a horizontal bar with five colored stars (yellow, blue, orange, green, purple) representing different rating levels. Below this bar, eight review entries are listed, each enclosed in a red-bordered box. Each entry contains the name 'ahmet kitu', a timestamp ('25/04/2023'), a short comment, and a star rating. The reviews are as follows:

- ahmet kitu (25/04/2023): Yemekler soğuk geldi. berbat! ★★★★☆☆
- ahmet kitu (25/04/2023): Her şey mükemmel! ★★★★★
- ahmet kitu (25/04/2023): Pena değil ★★★★★☆
- ahmet kitu (25/04/2023): Berbat! ★★☆☆☆
- ahmet kitu (25/04/2023): İyi! ★★★★★☆
- ahmet kitu (25/04/2023): Restorandan kovulduk! Bıdaha da Davosa gelmem! ★☆☆☆☆
- ahmet kitu (25/04/2023): Begeniemedim! ★★★★★☆
- ahmet kitu (25/04/2023): Çok basarılı! ★★★★★

## EK-16 Mobil Karşılama (*Onboarding*) Ekranları



## EK-17 Giriş Ekranı

The first screenshot shows a red error bar at the top with the message "Hoşgeldiniz!". Below it, the login form fields are empty. The second screenshot shows a red error bar at the top with the message "Hoşgeldiniz!". The "E-posta" field contains "@ qwe". A red error message "Bu geçerli bir e-posta değil!" is displayed below the field. The third screenshot shows a green checkmark icon next to the "E-posta" field, indicating a valid email address. The "Şifre" field contains "AfiyetTest123!". A red error message "Şifre alanı boş bırakılamaz!" is displayed below the field.

**Hoşgeldiniz!**

E-posta  
@ E-postanız

Şifre  
🔒 Şifreniz

Şifrenizi mi unuttunuz?

Giriş yap

Kayıt ol

Bu geçerli bir e-posta değil!

**Hoşgeldiniz!**

E-posta  
@ qwe

Şifre  
🔒 Şifreniz

Şifre alanı boş bırakılamaz!

Şifrenizi mi unuttunuz?

Giriş yap

Kayıt ol

Şifrenizi mi unuttunuz?

**Hoşgeldiniz!**

E-posta  
@ ogrenciumut@gmail.com

Şifre  
🔒 AfiyetTest123!

Şifrenizi mi unuttunuz?

Giriş yap

Kayıt ol

The left screenshot shows a red error bar at the top with the message "Hoşgeldiniz!". Below it, a modal window displays the message "Şifre değiştirme bağlantısı e-posta adresinize gönderildi." with a "Kapat" button. The right screenshot shows a red error bar at the top with the message "Hoşgeldiniz!". Below it, a modal window displays the message "Girdiğiniz e-posta geçerli bir e-posta adresi değil!" with a "Kapat" button. Both screenshots show the same login form fields as the top row.

**Hoşgeldiniz!**

E-posta  
@

Şifre  
🔒

Şifre değiştirme bağlantısı e-posta adresinize gönderildi.

Kapat

Giriş yap

Kayıt ol

Girdiğiniz e-posta geçerli bir e-posta adresi değil!

Kapat

Şifrenizi mi unuttunuz?

Giriş yap

Kayıt ol

## EK-18 Kayıt Ol Ekranı

**Şimdi kayıt ol!**

Ad  
 Adınız

Soyad  
 Soyadınız

E-posta  
 E-postanız

Şifre  
 Şifreniz

Şifrenizi tekrar giriniz  
 Şifrenizi tekrar giriniz

**Şimdi kayıt ol!**

Ad  
 Soyadınız

E-posta  
 E-postanız

Şifre  
 Şifreniz

Şifrenizi tekrar giriniz  
 Şifrenizi tekrar giriniz

**Kayıt ol**

**Giriş yap**

**Şimdi kayıt ol!**

Ad  
 Umut

Soyad  
 INCEER

E-posta  
 umut@gmail.com

Şifre  
 .....

Şifrenizi tekrar giriniz  
 .....

**Şimdi kayıt ol!**

Ad  
 INCEER

E-posta  
 umut@gmail.com

Şifre  
 .....

Şifrenizi tekrar giriniz  
 .....

**Kayıt ol**

**Giriş yap**

## EK-19 Ana Ekran

The image shows the McDonald's mobile application's main screen. At the top is a red header with a white leaf logo and a search bar containing a magnifying glass icon and the placeholder text "Ara...". Below the header is a white promotional box for a "2'li Double Fırsat Menüsü" (Double Double Offer Menu). It features two large burgers, a Coca-Cola bottle, and a side of fries. Text inside the box indicates "Kişi Başı 96 TL | 2 Kişilik Toplam 192 TL". Below the promotional box is a yellow banner for "McDelivery" with the text "www.mcdonalds.com.tr".

Öne Çıkan

Yakındakiler

**Maydonoz Döner**

★ 3.2(4)

Döner

BAĞLICA MAH. BAĞLICA BLV. NO:55, 06000 Etimesgut/Ankara



**Afiyet Restoran**

★ 3.5(10)

Kebap

Test mah. 111.sokak Test iş merkezi 11/09



Ana Sayfa



Harita



Tarayıcı

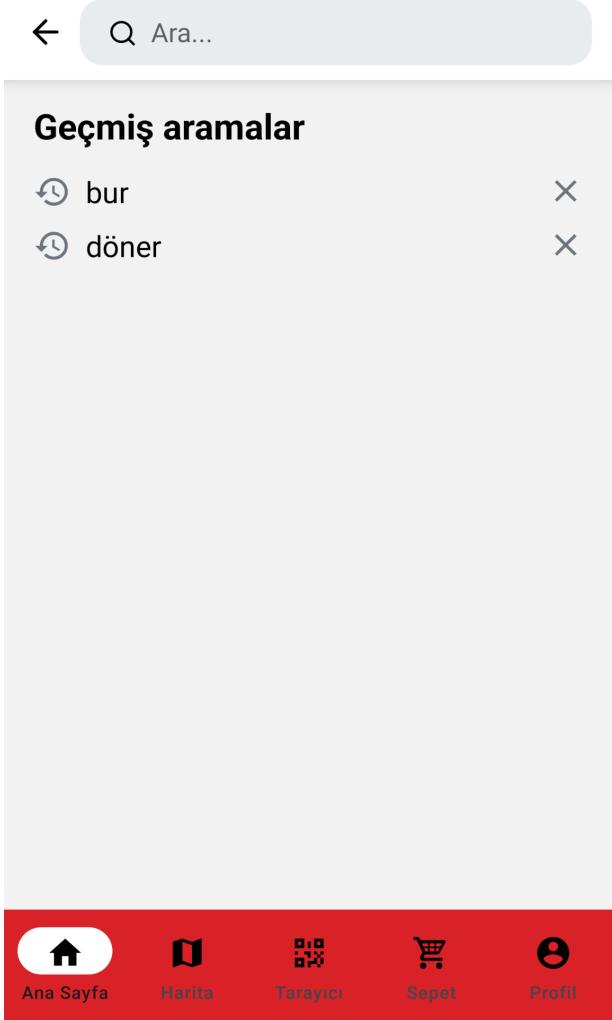


Sepet



Profil

## EK-20 Arama Ekranı



## EK-21 Harita Ekranı

İh Müzesi ve Parkı

ORA'STEAK & BURGERS

Gordion Alışveriş Merkezi

Başkent Üniversitesi

Arcadium Alışveriş Merkezi

Quick China Çayyolu

Tay Ümr

Köfteci Yusuf

Tepe-Mesa Park Mozaik Evleri

D200

Osmanoğulları Pide & Kebap

0 (0)

Kebap

Bağlıca Mah. Mermeroğlu Cad. No:46/B1, Ankara

Restoranı Gör

Ana Sayfa

Harita

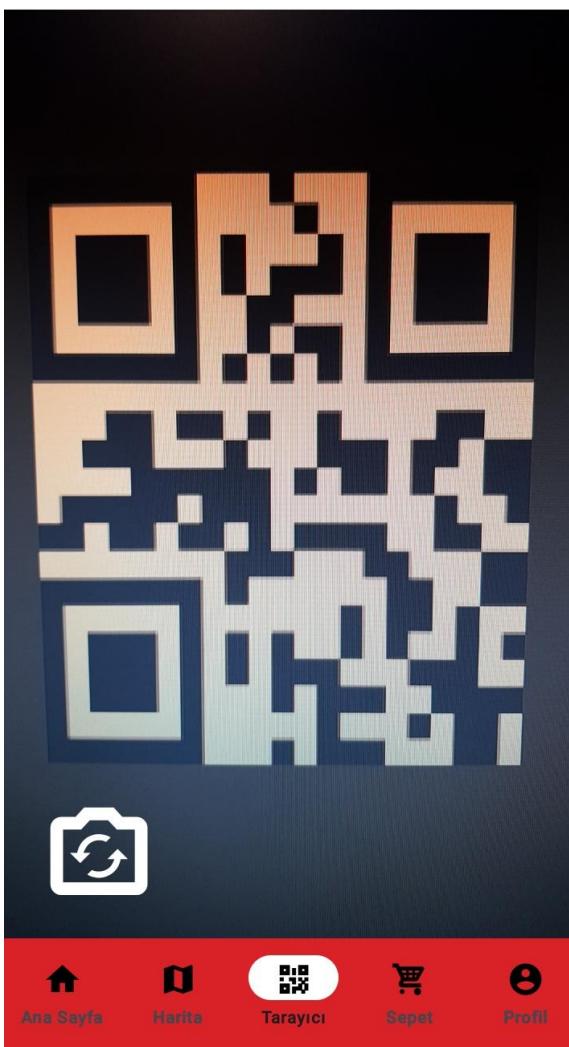
Tarayıcı

Sepet

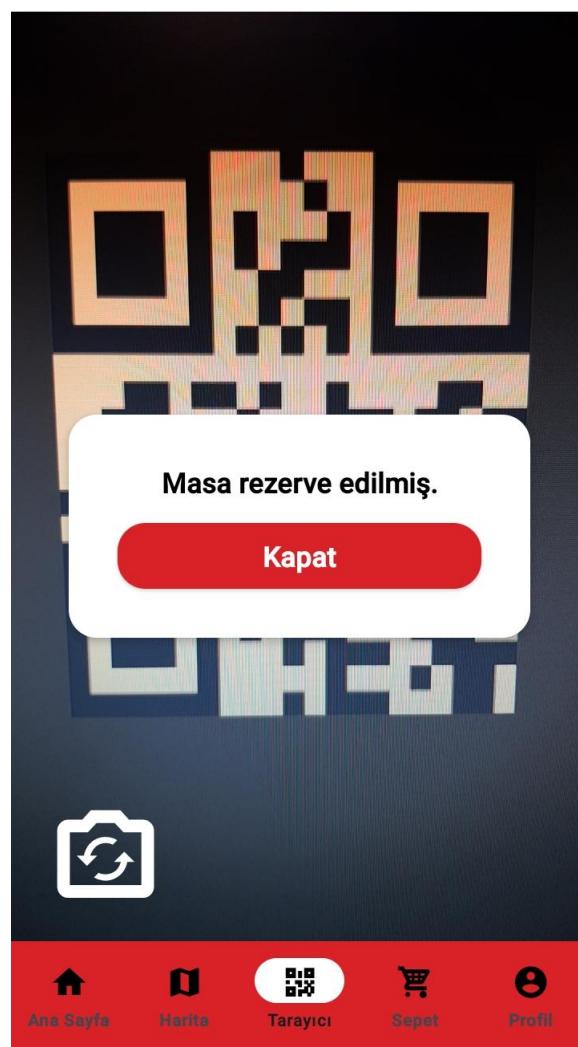
Profil

## EK-22 Tarayıcı Ekranı

Tarayıcı



Tarayıcı



## EK-23 Sipariş Ekranı

← Sipariş

Barış Aspava, Çankaya (Çayyolu  
Yaşamkent Mah.)

Kebap  
Yaşamkent Mh., 3207. Cd. No:11, 06810 Yenimahalle/  
Çankaya/Ankara

★ 0 (0 İnceleme) 2 km

Ana Yemek Menüler İçecekler Pideler

Beyti Kebap  
Adana, Lavaş, Közlenmiş Biber  
155 TL

Ana Sayfa Harita Sipariş Sepet Profil

← Sipariş

Barış Aspava, Çankaya (Çayyolu  
Yaşamkent Mah.)

Restoran Menüde Değişiklik  
Yaptı

Yeni Menüyü Gör

Ana Sayfa Harita Sipariş Sepet Profil

← Sipariş

Barış Aspava, Çankaya (Çayyolu)

Beyti Kebap  
Adana, Lavaş, Közlenmiş Biber  
155 TL

Ana Sayfa Harita Sipariş Sepet Profil

Beyti Kebap 155 TL

Adana, Lavaş, Közlenmiş Biber

- 1 + Sepete Ekle

Ana Sayfa Harita Sipariş Sepet Profil

## EK-24 Yorum Ekranı

← Yorumlar

★ 5 ★ 4 ★ 3 ★ 2 ★ 1

**Christopher Anderson** ★ 3  
21/11/2022  
Its was juicy and delicious but very hot

**Umut Ciloglu** ★ 2  
13/11/2022  
Bir ıslak mendil bile koymamışlar. Soğuktu.

**Emin Umut Gercek** ★ 4  
21/05/2022  
Vazgeçemediğim bir lezzet...

**Öncum Korkmaz Yilmaz** ★ 4  
21/09/2022  
Tatlar güzeldi, porsiyonlar biraz daha büyük olabilirdi.

**Yorum Ekle**

Home Harita Sipariş Sepet Profil

← Yorumlar

★ 5 ★ 4 ★ 3 ★ 2 ★ 1

**Deneyiminizi paylaşın!**

★ 5 ★ 4 ★ 3 ★ 2 ★ 1

Yorumunuz...

**Gönder**

Home Harita Sipariş Sepet Profil

## EK-25 Sepet Ekranı

1      2

Sepet      Ödeme

### Sepet

Toplam: 120 TL

Tavuk Döner XLarge  
Dürüm (45 cm)  
95 TL

Pepsi Max (330ml)  
25 TL

Kredi Kartıyla Şimdi Öde  
 Nakit Olarak Sonra Öde

[Sepet Durumunu Kontrol Et](#)

[Ana Sayfa](#) [Harita](#) [Sipariş](#) [Sepet](#) [Profil](#)

1      2

Sepet      Ödeme

### Sepet

Toplam: 115 TL

Beyti  
80 TL

Tükenen ürünler: Fırında Beyti  
Fiyatı değişen ürünler: Ayran  
Sepetiniz otomatik olarak güncellenmiştir.

[Kapat](#)

Kredi Kartıyla Şimdi Öde  
 Nakit Olarak Sonra Öde

[Sepet Durumunu Kontrol Et](#)

[Ana Sayfa](#) [Harita](#) [Sipariş](#) [Sepet](#) [Profil](#)

1      2

Sepet      Ödeme

### Ödeme

Kart Üzerindeki Ad Soyad  
Kart Numarası  
Ay / Yıl      CVC

Taksit seçenekleri kart numaranızı girildikten sonra görünecektir.

3D Secure

[iyzico güvencesi ile kartımı sakla.](#)

**120,00 TL ÖDE**

[Sepete Dön](#)

[Ana Sayfa](#) [Harita](#) [Sipariş](#) [Sepet](#) [Profil](#)

1      2

Sepet      Ödeme

### Ödeme

Ödeme Başarılı

[Sepete Dön](#)

[Ana Sayfa](#) [Harita](#) [Sipariş](#) [Sepet](#) [Profil](#)

1      2

Sepet      Ödeme

### Ödeme

Ödeme yapmak için garson çağrı

[Sepete Dön](#)

[Ana Sayfa](#) [Harita](#) [Sipariş](#) [Sepet](#) [Profil](#)

## EK-26 Profil Ekranı

