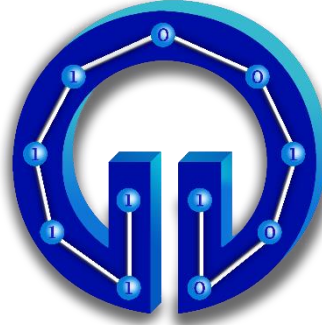


**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**YEMEK YEMEK: VERİ MADENCİLİĞİ YÖNTEMLERİYLE
BESİNLERİ TANIYAN VE EN UYGUN YEMEK TARİFİ
ÖNERİLERİNİ SUNAN MOBİL UYGULAMA**

MÜHENDİSLİK TASARIMI

**Mehmet Alperen ÇUBUK
Muhammet Mücahit AKTEPE**

2018-2019 GÜZ DÖNEMİ

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YEMEK YEMEK: VERİ MADENCİLİĞİ YÖNTEMLERİYLE
BESİNLERİ TANIYAN VE EN UYGUN YEMEK TARİFİ
ÖNERİLERİNİ SUNAN MOBİL UYGULAMA**

MÜHENDİSLİK TASARIMI

**Mehmet Alperen ÇUBUK
Muhammet Mücahit AKTEPE**

2018-2019 GÜZ DÖNEMİ



IEEE Etik Kuralları IEEE Code of Ethics



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

“Yemek Yemek: Veri Madenciliği Yöntemleriyle Besinleri Tanıyan ve En Uygun Yemek Tarifi Önerilerini Sunan Mobil Uygulama” isimli bu çalışma, Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü’nde güz dönemi tasarım projesi olarak hazırlanmıştır.

Eğitim ve öğretim hayatımız boyunca bize destek veren ailelerimize ve çalışmalarımız boyunca bizi yönlendiren hocamız Dr. Öğr. Üyesi Murat AYKUT’a teşekkür ederiz.

Mehmet Alperen ÇUBUK
Muhammet Mücahit AKTEPE
Trabzon 2018

İÇİNDEKİLER

ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
1. GENEL BİLGİLER	1
1.1 Giriş.....	1
1.2 Veri Tabanı	2
1.2.1 MySQL	2
1.2.2 ORM	2
1.3 Web Sunucu	2
1.3.1 Flask.....	2
1.3.2 REST.....	2
1.3.3 JSON.....	3
1.4 Mobil Uygulama	3
1.4.1 Flutter.....	3
1.5 Öneri Algoritmaları.....	4
1.5.1 Besin Listesinden Yemek Önerisi.....	4
1.5.2 A-priori Algoritması	5
2. PROJE TASARIMI	9
2.1 Gereksinim Analizi	9
2.1.1 Veri Tabanı Gereksinimleri	9
2.1.2 Web Sunucu Gereksinimleri	9
2.1.3 Mobil Uygulama Gereksinimleri	9
2.1.4 Öneri ve Örüntü Tanıma Sistemi Gereksinimleri	10
2.2 Mimari Tasarım	10
2.2.1 Veri Tabanı Tasarımı	11
2.2.2 Web Sunucu Tasarımı.....	11
2.2.3 Mobil Uygulama Tasarımı.....	13
2.2.4 Öneri ve Örüntü Tanıma Sistemi Tasarımı	23
2.3 UML Nesne Modeli	24
2.4 Yapılan Çalışmalar.....	24
3. KAYNAKLAR	26
STANDARTLAR ve KISITLAR FORMU	27

ÖZET

Proje konusu seçilirken özgün bir proje olması ve mümkün olduğunca fazla kullanıcıya hitap edecek bir uygulama olması istenmiştir. Günümüzde akıllı telefonlar oldukça popüler olduğundan, uygulamanın mobil platform için hazırlaması daha büyük kitlelere ulaşmamızda etkili olacaktır. Bu sebeple hem iOS hem Android işletim sistemli cihazlarda çalışacak bir mobil uygulama Flutter ile hazırlandı.

Uygulamada kategorize edilmiş besinler listeler halinde tutulmaktadır. Kullanıcı bu besinlere tıklayarak besin hakkında çeşitli bilgiler edinebilir. Ayrıca istenilen besinler içindekiler listesine eklenebilmektedir. Yemek önerisi yapılırken içindekiler listesine eklenen besinleri içeren yemekler kullanıcıya önerilir. Kullanıcı önerilen yemekler listesinden seçtiği bir yemeğin tarifine erişerek bu tarif yardımıyla yemeği hazırlayabilmektedir.

Uygulamanın arka tarafında bir web sunucusu ve bir veri tabanı barınmaktadır. Veri tabanında besinler, tarifler ve kullanıcı bilgileri gibi uygulamamız için gerekli bilgiler tutulmaktadır. Veri tabanı ile mobil uygulamanın iletişimi, web servis aracılığıyla gerçekleştirilmektedir. Uygulama başlatıldığında gerekli verileri web sunucusuna gönderdiği bir GET isteğiyle veri tabanından almaktadır.

Öneri sistemi için iki farklı algoritma kullanılmıştır. İlk öneri algoritması seçilen besinlerin yemek tarifindeki besinlerle uyumluğunu kontrol edip uyumluluk oranı yüksek yemekleri üst sıralarda göstermektedir. Ancak yalnızca bu algoritmanın kullanılması her seferinde aynı besinler için aynı yemeklerin önerilmesine yol açacaktır. Bunu önlemek ve farklı lezzetler denemek isteyen kullanıcılara daha iyi öneriler sunmak için ikinci bir öneri algoritması daha kullanıldı.

Uygulamamız A-priori algoritmasını kullanarak kullanıcılara sevdiği yemekleri seven diğer kullanıcıların sevdiği alternatif yemekleri de önermektedir. Böylece kullanıcıların ilgisini çekebilecek farklı yemekler de önerilerek kullanıcıların uygulamadan en iyi şekilde faydalanması hedeflenmiştir. Bu algoritmanın kullanılabilmesi için her bir kullanıcının bir favori listesi vardır ve kullanıcılar beğendikleri yemek tariflerini favorileri arasına alabilmektedir.

Uygulamada yapılacak tüm bu işlemlerin kullanıcı tarafından kolayca gerçekleşmesi için sade ve göze hitap eden bir arayüz ile mobil uygulama tasarımı tamamlanmıştır. Gerekli tüm menülere kolayca erişilebilir. Örneğin herhangi bir öneri yapılmadan da tüm yemek tarifleri görüntülenebilmektedir.

Projenin ilerleyen kısımlarında akıllı tarama menüsü ile besinler kameradan otomatik olarak tanınacaktır. Uygulamada kayıtlı bir besinin fotoğrafı çekildiğinde görüntüden besin tanınarak ilgili besinin bilgileri ekranda görüntülenecektir. Ardından besin yine listeye eklenebilecek ve yemek önerisi alınabilmektedir. Böylece kullanıcı elindeki besinlerin sadece fotoğraflarını çekerek hızlı bir şekilde uygulamaya tanıtabilecek ve bu besinleri kullanarak yapabileceği yemekler hakkında öneriler alabilecektir.

1. GENEL BİLGİLER

1.1 Giriş

Yıllardır tartışılan ve üzerine araştırmalar yapılan sağlıklı yaşam konusu günümüzde iyice önem kazanmış ve insan sağlığının iyiye gitmesinin ancak tüketilen besinlere dikkat edilmesiyle mümkün olacağı defalarca kez vurgulanmıştır. Tüm bu bilgiler ışığında insanların çeşitli besinlerin kalorilerini, makro değerlerini (karbonhidrat, protein, yağ) ve sağlık açısından yarar/zararlarını hızlı bir şekilde öğrenebilmelerini ve bu besinlerle yapılabilecek yemeklerin tariflerine erişebilmelerini sağlamak projemizin temel amacı olmuştur.

Proje kapsamında, hem iOS hem Android işletim sistemli telefonlar üzerinde çalışacak bir mobil uygulama ve veri tabanından verileri çekip uygulamaya sunacak bir web sunucu hazırlanmıştır. Kullanıcı, uygulamada daha önceden tanımlı olan besinleri seçip onlarla ilgili çeşitli bilgilere erişebilmekte ve yine aynı besinleri bir listeye ekleyerek bu listedeki malzemelerden yapılabilecek yemek tariflerine ulaşabilmektedir. Uygulama, yemek tariflerini seçilen besinlere göre en alakalı yemeği en üstte gösterecek şekilde özel bir sıralama ile önermektedir.

Besin seçimleri uygulama içerisinden elle yapılabileceği gibi, akıllı tarama menüsünden kamera açılarak ilgili besinin fotoğrafı çekilerek de yapılabilecektir. Uygulama otomatik olarak çekilen fotoğraftaki besini tanıyıp ilgili besin bilgilerini içeren sayfayı açacaktır. Bu sayede kullanıcı elindeki besinlerin fotoğraflarını çekerek hızlı bir şekilde uygulamaya tanıtabilecek ve bu besinler ile hangi yemekleri yapabileceğini öğrenebilecektir. Bu sayede daha etkileyici bir kullanıcı deneyimi sağlanacaktır.

Proje temel olarak bir veri tabanı, bir web sunucusu ve bir mobil uygulamadan oluşmaktadır. Gerekli tüm arka plan işlemleri web sunucusu tarafından yapılmaktadır. Mobil uygulama yalnızca veri gönderip, gönderilen verilere göre gelen yanıtı görüntüleyen bir arayüz görevi üstlenmiştir. Besinler mobil uygulamadan seçildikten sonra web sunucusuna gönderilmekte ve öneri algoritması web sunucusu üzerinde çalıştırılarak sonuç üretilmektedir. Üretilen sonuçta bulunan tarifler mobil uygulamaya gönderilip kullanıcıya gösterilmektedir.

1.2 Veri Tabanı

Uygulamamızdaki tüm verileri saklayacağımız bir veri tabanı MySQL ile oluşturuldu. Besin bilgileri ve yemek tarifleri veri tabanımızdaki ilgili tablolara eklendi. Yemek tarifleri için Türkçe bir API bulunmadığı için veriler bizim tarafımızdan elle veri tabanına girildi. İlk etapta sınırlı sayıda yemek tarifi ve bu tariflerde kullanılan besinler veri tabanına eklendi.

1.2.1 MySQL

MySQL, altı milyondan fazla sistemde yüklü bulunan çoklu iş parçacıklı, çok kullanıcı, hızlı ve sağlam bir veri tabanı yönetim sistemidir. [1] Ücretsiz olması ve stabil bir şekilde bir çok sistemde çalışabilmesi sebebiyle projemizin veri tabanı ihtiyacı için kullanılmıştır.

1.2.2 ORM

ORM, Object Relational Mapping anlamına gelmektedir. ORM ilişkisel veri tabanı (RDBMS) ile nesneye yönelik programlamanın (OOP) arasında bir tür köprü özelliği gören ve ilişkisel veri tabanındaki bilgilerimizi yönetmek için, nesne modellerimizi kullandığımız bir tekniktir. [2] Projemizde Python için yazılmış bir ORM kütüphanesi olan SQLAlchemy kullanılarak veri tabanı işlemleri sorgular yerine Python kodu yazılarak yapıldı.

1.3 Web Sunucu

Veri tabanı ile mobil uygulama arasındaki iletişimi sağlayacak bir web sunucusu Python ile Flask framework'ü kullanılarak hazırlandı. Mobil uygulama için gerekli kullanıcı bilgileri, besin bilgileri, yemek tarifleri gibi verileri mobil uygulama tarafından gelen bir GET isteğine karşılık olarak veri tabanından çekip uygulamaya göndermektedir. Sunucu REST mimarisi kullanmaktadır. Gönderilen ve alınan veriler JSON formatı kullanır.

1.3.1 Flask

Flask web uygulamalarını geliştirmemizi sağlayan Python ile yazılmış bir mikro framework'tür. Küçük bir framework olmasına karşın oldukça hızlıdır. Flask ile projemiz için gerekli web servis REST mimarisi kullanarak hazırlandı.

1.3.2 REST

REST, servis yönelimli mimari üzerine oluşturulan yazılımlarda kullanılan bir veri transfer yöntemidir. HTTP üzerinde çalışır ve diğer alternatiflere göre daha basittir, minimum içerikle veri alıp gönderdiği için de daha hızlıdır. İstemci ve sunucu arasında XML veya JSON verilerini taşıyarak uygulamaların haberleşmesini sağlar. [3]

REST mimarisinde HTTP metotlarından yararlanılır. DELETE, GET, POST ve PUT metotları kullanılır. [3] Projemizde ihtiyacımız olan yalnızca veri tabanına veri yazmak veya veri çekmek olduğu için web sunucumuzun bu işlemleri hızlıca yapmasını sağlayacak REST mimarisi tercih edilmiştir.

1.3.3 JSON

Json, Javascript uygulamaları için oluşturulmuş bir veri formatıdır. Javascript Object Notation'ın kısaltmasıdır. Json'ın çıkış amacı veri transferlerinde verilerin XML'den daha az yer kaplamasını sağlamaktır. Şu an sadece Javascript uygulamalarında değil, yazılım geliştirmede kullanılan birçok teknolojiye Json formatındaki veriler tercih edilmektedir. [4]

Json türündeki veriler iki parçadan oluşur: key (anahtar) ve value (değer). Anahtarda nesnenin hangi özelliğinin olduğu (koddaki değişken ismi gibi düşünülebilir) tanımlanırken değerde ise anahtar özelliğinin değeri (değişkenin değeri) tanımlanır. Nesnelerdeki anahtar ve değerler string türünde tanımlanır. Aşağıda projede kullanılan basit bir json nesnesi örneği bulunmaktadır.

```
{
  "calorie": 34,
  "carb": 4.98,
  "category": 2,
  "description": "Patlıcan, folik asit, potasyum, lif açısından zengindir. Antioksidan içeriği yüksektir. Bu sayede kanserden koruyucu, vücut direncini artırıcı etki gösterir.",
  "fat": 0.36,
  "health_point": 5,
  "image": "lib/assets/foods/patlican.png",
  "name": "Patlıcan",
  "protein": 2.48
}
```

1.4 Mobil Uygulama

Bilindiği üzere iOS ve Android işletim sistemleri mobil pazarında en çok kullanılan sistemlerdir. Hedefimiz maksimum kullanıcıya erişmek olduğu için her iki sistem için de uygulamamızın çalışması istenmiştir. Her iki platform için ayrı uygulamalar hazırlamaktansa hibrit bir uygulama hazırlayarak bir kere yazıp her iki platformda çalışan bir mobil uygulama yazmak daha mantıklı olacaktır. Bu sebeple projemizde yaptığımız araştırmalar sonucunda Google tarafından geliştirilen Flutter kullanıldı.

1.4.1 Flutter

Flutter, içinde framework, widget ve diğer araçları barındıran, geliştiricilere Android ve iOS platformu için uygulama geliştirme imkanı sunan bir mobil uygulama SDK'sıdır. Google tarafından geliştirilen Flutter, açık kaynak kodlu olup, tüm geliştirici ve tasarımcılar tarafından ücretsiz olarak kullanılabilir. [5]

Flutter, profesyonel bir SDK'dır ve Windows, Linux ya da Mac ortamında çalıştırılabilir. Mobile-first 2D render motoru, modern react-style framework'ü, Android ve iOS ortamında kullanılan widget desteğiyle, zengin bir paket içeriğine sahiptir. Ek olarak Flutter hafif katmanlı C/C++ ile hazırlanmış kod yapısına sahip olup, nesne yönelimli Dart programlama dilini kullanır. [5]

1.5 Öneri Algoritmaları

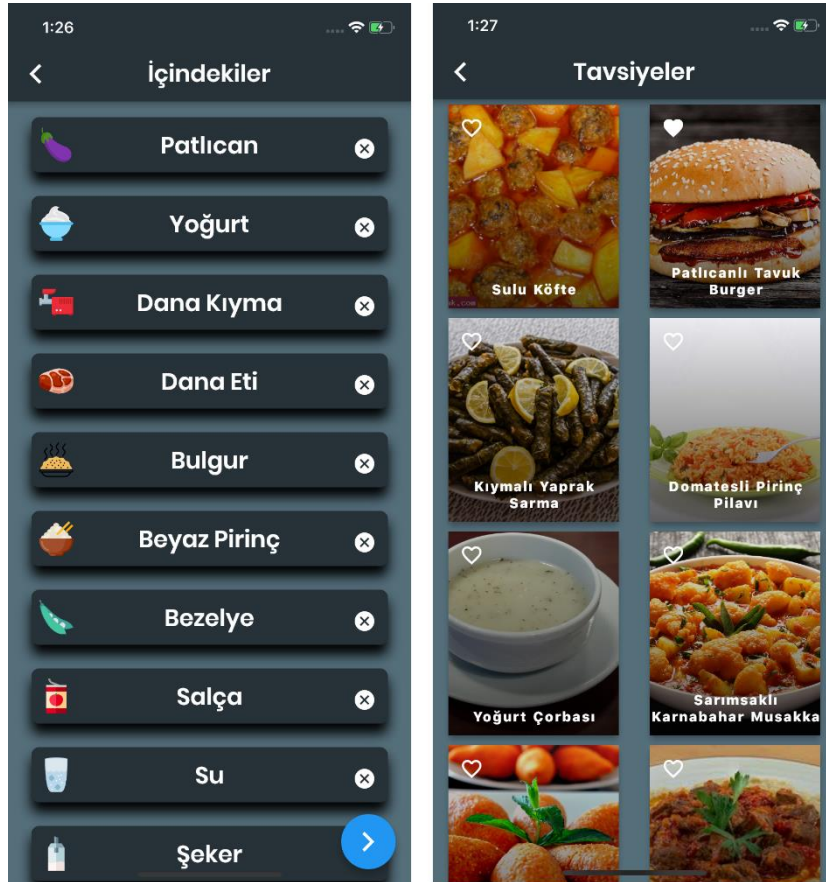
Uygulamanın yemek önerisi yapmak için kullandığı iki algoritma mevcuttur. Bu algoritmalarından birisi basit bir mantık kullanılarak tarafımızca yazılmıştır. İkinci algoritma için ise veri madenciliğinde kullanılan A-priori algoritması tercih edilmiştir.

1.5.1 Besin Listesinden Yemek Önerisi

Yemek önerisi için kullanılan ilk algoritma tarafımızca yazıldı. Seçilen besinlerden en fazlasını içeren yemeğin en üst sıralarda görüntülenmesi sağlandı. Böylece kullanıcının elindeki besinlerden en iyi şekilde faydalanabilmesi hedeflendi. Veri tabanındaki tarif sayısı arttıkça algoritmanın önerileri daha da iyileşmektedir.

Algoritmamızın mantığını kısaca açıklamak gerekirse ilk olarak kullanıcının seçtiği besinlerden en az birini içeren tüm yemek tarifleri veri tabanından getirilir. Ardından içindekiler listesindeki ilk besin seçilerek bir döngü ile bu tarifler üzerinde gezilir. Eğer seçilen besin ilgili yemek tarifinde var ise tarife 1 puan eklenir. İlk besin için döngü tamamlandıktan sonra sırayla diğer besinler için işlem tekrarlanır.

Bu işlem sonucunda yapılan puanlama yemeğin kaç adet besini içerdiği bilgisini bize sağlamaktadır. Bu puanlar kullanılarak listemizdeki yemek tarifleri en çok besin içeren en az besin içerene doğru sıralanır. Sıralanmış tarifler bir liste halinde tutulur.



Şekil 1.1 Öneri sistemi örneği

Öneri yapılırken sıralanmış liste kullanıcıya gösterilir. Algoritmanın dezavantajı seçilen besinlerin dışında ekstra malzeme gerektiren yemek tariflerini de sıralamasıdır. Bu durumda kullanıcının eksik besinleri temin etmesi gerekmektedir.

1.5.2 A-priori Algoritması

Veri madenciliğinde kullanılan ve veri kümeleri veya veriler arasındaki ilişkiyi çıkarmak için geliştirilmiş bir algoritmadır. A-priori algoritması, özellikle çok büyük ölçekli veri tabanları üzerindeki veri madenciliği çalışmalarında geliştirilmiştir. Genel anlamda münasebet kuralı (association rule, birliktelik kuralı) çıkarımında kullanılan bir algoritmadır. Algoritmanın amacı, veri tabanında bulunan satırlar arasındaki bağlantıyı ortaya çıkarmaktır. [6]

A-priori algoritmasının en bariz örneği alışveriş sepetidir. Bir alışveriş sepetinde bulunan ürünlerden ya da alışveriş fişlerinden yola çıkarak hangi ürünün ne sıklıkla satıldığı ve hangi ürünlerin birlikte satıldığını öğrenmemiz mümkündür.



Şekil 1.2 Apriori örneği. [7]

Bizim uygulamamız düşünüldüğünde ise kullanıcıların favori listesi, alışveriş sepeti gibi düşünülebilir. Yani kullanıcıların hangi yemekleri birlikte favori listesine aldığı kontrol edilerek, A yemeğini beğenen kullanıcılar B yemeğini de beğeniyor, şeklinde bir veri üretilir, A yemeğini favori listesine alan kullanıcılara B yemeği önerilebilir.

A-priori algoritmasının ilk aşamasında tüm kullanıcıların favori listeleri kontrol edilerek her bir yemeğin kaç kişinin favori listesinde olduğu bilgisi hesaplanır. Bu değer toplam kullanıcı sayısına bölünerek 0-1 aralığında bir frekans değeri oluşturulur.

Aşağıdaki örneği inceleyelim. 4 kullanıcımızın favori listesi aşağıdaki gibi olsun.

Kullanıcı 1	{A,B}
Kullanıcı 2	{A,B,C,D}
Kullanıcı 3	{A,C}
Kullanıcı 4	{A,B,C}

Tablo 1.1 Apriori örneği.

Öyleyse A,B,C ve D yemeklerinin frekans değerleri aşağıdaki gibi olacaktır.

A	1
B	0.75
C	0.75
D	0.25

Tablo 1.2 Apriori örneği.

Görüldüğü üzere A tüm kullanıcıların favori listesinde, B ve C kullanıcıların %75'inin favori listesinde ve D kullanıcıların %25'inin favori listesinde. Bu tablo hazırlanırken önceden belirlenen bir destek (support) değeri belirlenmelidir. Bu değer bizim eşik değerimiz olacak ve büyük veri tabanları için tablomuzun biraz daha sade bir hal almasını sağlayacaktır.

Destek değeri 0-1 aralığında olmalıdır. Örneğin destek değerimizin 0.5 olduğunu varsayalım. Bu durumda yukarıdaki tablo aşağıdaki gibi sadeleşecektir.

A	1
B	0.75
C	0.75

Tablo 1.3 Apriori örneği.

Görüldüğü üzere D değeri destek değerimizin altında olduğu için tablomuzdan silindi. Algoritmayı bir adım daha öteye taşımak için bu sefer 2'li kombinasyonlar kullanılır.

A,B	0.75
A,C	0.75
B,C	0.50

Tablo 1.4 Apriori örneği.

Yukarıdaki tabloda görüldüğü gibi A ve B'nin birlikte favori olduğu kullanıcı yüzdesi %75'tir. En düşük oranımız B ve C'nin birlikte favori olduğu kullanıcı yüzdesi, destek değerinin altına düşmediği için bu tabloda herhangi bir sadeleştirme yapılmaz.

Destek değerinin altına düşülene kadar algoritma çalışmaya devam eder. Bir sonraki adımda 3'lü kombinasyonlar kontrol edilir.

A,B,C	0.50
-------	------

Tablo 1.5 Apriori örneği.

Yukarıda görüldüğü gibi yine destek değerin altına düşülmediği için bu tablo da verilerimiz arasına eklenir ve son durumda aşağıdaki tablo elde edilmiş olur.

A	1
B	0.75
C	0.75
D	0.25
A,B	0.75
A,C	0.75
B,C	0.50
A,B,C	0.50

Tablo 1.6 Apriori örneği.

Son tablodaki tüm verilerin frekans değerlerinin, destek değerimiz olan 0.5'in üzerinde olmasına dikkat edilmelidir.

Bu tabloya bakılarak çeşitli veriler elde edilebilir. Örneğin A,B'nin frekans değeri 0.75 olduğundan A yemeğini favorilerine almış kullanıcıların %75'i B ürününü de favorilerine almıştır diyebiliriz. Bu durumda A yemeğini favoriye alıp B yemeğini almamış olan kullanıcılara B yemeği önerilebilir.

Son tablo (Tablo 1.6) üzerinden öneri yapmak için bir güven (confidence) değeri belirlenerek güvenilir veriler üzerinden öneri yapılır. Tabloda 2'li veya daha fazla yemek içeren satırlar içerisinde güven değerinden yüksek frekans değerine sahip yemekler iyi bir öneri yapmak için kullanılır. Örneğin güven değerimiz 0.7 olarak seçilirse tablomuzdaki A,B ve A,C satırları 0.75 frekans değerine sahip olduğu için öneri sistemine uygundur.

Bu durumda favori listesinde A yemeği olan kullanıcılara B ve C yemekleri önerilmektedir. Favori listesinde B veya C yemeklerinden en az birisi bulunan kullanıcılara ise A yemeği önerilmektedir. Veri sayısı arttıkça algoritma daha doğru sonuçlar üretmektedir. Destek ve güven değerlerinin seçimi de algoritmanın verimi etkilemektedir. Favori listeleri birbiriyle alakasız ise destek değerinin düşük seçilmesi gerekir. Favori listelerinin benzerlik oranı arttıkça destek değeri artırılmalıdır. Güven değeri de destek değeriyle doğru orantılı olarak seçilir.

A-priori büyük veri tabanlarında daha iyi sonuçlar üretmekte olup bizim uygulamamızdaki veri ve kullanıcı sayımız kısıtlı olduğundan ilk etapta beklenildiği kadar performanslı çalışmamaktadır. Kullanıcı sayısı ve tarif sayısı arttıkça kullanıcılara daha iyi öneriler sunulacaktır.

Özetlemek gerekirse A-priori algoritmasının girdi ve çıktıları aşağıdaki gibidir.

Girdi:

- 1) Tüm kullanıcıların favori listeleri

Kullanıcı 1	{A,B}
Kullanıcı 2	{A,B,C,D}
Kullanıcı 3	{A,C}
Kullanıcı 4	{A,B,C}

Tablo 1.7 Apriori girdi.

Çıktılar:

- 1) (A,B) 0.75 tarzı birlikte favoriye alınan yemek kümeleri ve frekans değerleri

A,B	0.75
A,C	0.75
B,C	0.50
A,B,C	0.50

Tablo 1.8 Apriori çıktı.

- 2) $A \Rightarrow B$ tarzı önerilecek yemeklerin listesi (Kullanıcı A'yı seviyorsa B'yi öner gibi)

A	B
A	C
B	A
C	A

Tablo 1.9 Apriori çıktı.

İlk çıktı hata ayıklama aşamasında kontrol amaçlı kullanılmış olup, ikinci çıktıdan gelen veriler kullanılarak, uygulamamız kişiye özel yemek önerilerini kullanıcılara sunmaktadır.

2. PROJE TASARIMI

2.1 Gereksinim Analizi

Proje Gereksinimleri 4 başlık altında incelenebilir. Projemizin gerçekleşmesi için veri tabanı, web sunucusu ve mobil uygulama gerekmektedir. Hedefimiz olan öneri sisteminin gerçekleşmesi ve örüntü tanıma teknikleriyle besinlerin uygulamamıza tanıtılması gereksinimlerimiz arasındadır.

2.1.1 Veri Tabanı Gereksinimleri

- 1) Uygulama için gerekli olan kullanıcı, besin ve yemek bilgilerinin ilişkisel bir veri tabanı ile gerçekleşmesi.
- 2) A-priori öneri sisteminde kullanılacak veri setinin sağlanması için veri tabanının ilgili şekilde düzenlenmesi ve gerekli tabloların eklenmesi.
- 3) Veri tabanını üzerinde barındıracak ve yeterli seviyede kaynak sağlayacak bir bulut.

2.1.2 Web Sunucu Gereksinimleri

- 1) Veri tabanı ile mobil uygulama arasında köprü görevi görecektir web sunucusunun gerçekleşmesi.
- 2) Web sunucunun gelen istekleri mümkün olduğunca hızlı bir şekilde cevaplaması için gerekli optimizasyonların yapılması.
- 3) Web sunucuyu üzerinde barındıracak ve yeterli trafiği kaldırabilecek kadar kaynak sağlayacak bir bulut.

2.1.3 Mobil Uygulama Gereksinimleri

- 1) Uygulamayı ilk kez yükleyen kullanıcıların üyelik alabileceği bir kayıt ekranının ve daha önceden üyeliği bulunan kullanıcılar için bir giriş ekranının oluşturulması.
- 2) Kullanıcıların kayıt ve giriş işlemlerini mail adresleri ve şifreleri ile yapması için gerekli kontrol mekanizmasının kurulması.
- 3) Kullanıcıların giriş yaptıktan sonra herhangi bir besin seçebileceği ve seçilen besinin bilgilerini öğrenebileceği bir ekranın oluşturulması.
- 4) Seçilen besinin bilgilerinin web sunucusu tarafından veri tabanından çekilmesi ve mobil uygulamaya sunulması.
- 5) Kullanıcıların besin bilgilerini öğrendiği ekran üzerinde, yemek tarifinde bulunmasını istediği besinleri arka plandaki listeye ekleyebileceği bir sistem oluşturulması.

6) Kullanıcının besinleri eklemeyi bitirdiği zaman bir buton yardımıyla, daha önceden arka plandaki listeye ekleme yaptığı besinleri içeren yemek tariflerinin listelendiği ekrana geçebilmesi.

7) Yemeklerin listelendiği ekranda herhangi bir yemek üzerine tıklanması durumunda ilgili yemeğin tarifinin ve çeşitli bilgilerinin (toplam kalori değeri, makro değerleri gibi) bulunduğu ekrana geçiş yapılması.

8) Her kullanıcının kendine özgü favori yemekler listesinin oluşturulması ve yemek tariflerinin favorilere alınabilmesi.

9) Kullanıcının besin seçmeden (öneri sistemini kullanmadan) de tüm yemekleri görebileceği bir ekranın oluşturulması.

10) Akıllı tarama işlevi için telefon kamerasının açılıp fotoğraf çekerek, fotoğrafı sunucu tarafına göndermesi.

2.1.4 Öneri ve Örüntü Tanıma Sistemi Gereksinimleri

- 1) Web sunucu üzerinde çalışacak besin öneri algoritmalarının implementasyonu.
- 2) Akıllı tarama için gelen fotoğraflardan besinin tanınması ve mobil uygulamaya cevap gönderilmesi.
- 3) Öneri ve örüntü tanıma sistemlerinin gerçek zamanlı çalışması.
- 4) Besin tanıma sisteminin her durumda (örneğin farklı ışıklandırma koşulları altında) doğru çalışması.
- 5) Besin tanıma sisteminin hazırlanması için yüksek performanslı bir GPU.

Yukarıdaki gereksinimler göz önüne alındığında uygulamamızın temel anlamda girdi ve çıktıları aşağıdaki gibidir.

Girdiler:

- 1) Kullanıcı bilgileri
- 2) Besin isimleri/görüntüleri
- 3) Favoriler

Çıktılar:

- 1) Besin bilgileri
- 2) Yemek tarifleri
- 3) Favori listesi
- 4) Öneri listesi

2.2 Mimari Tasarım

Projemizin mimari tasarımı 4 başlık altında incelenebilir. Veri tabanı, web sunucu ve mobil uygulama tasarımları gerçekleştirilmiştir. Örüntü tanıma teknikleri ile besin tanıma sistemi projenin ilerleyen aşamalarında gerçekleştirilecektir.

İstek:

```
curl --user mail:şifre http://0.0.0.0:5000/api/v1/foods
```

Yanıt:

```
{
  "Foods": [
    {
      "calorie": 19,
      "carb": 2.86,
      "category": 2,
      "description": "Domates, C ve A vitamini, potasyum, mangan minerallerinin iyi bir kaynağıdır. Düşük kalorili ve yüksek lifli olması dolayısıyla kilo kontrolünün istendiği diyetler için iyi bir seçenektir. Antioksidan özelliği ile kanser oluşumuna karşı niteliktedir.",
      "fat": 0.23,
      "health_point": 5,
      "image": "lib/assets/foods/domates.png",
      "name": "Domates",
      "protein": 1.05
    },
    . . .
  ]
}
```

İstek:

```
curl --user mail:şifre http://0.0.0.0:5000/api/v1/recipes
```

Yanıt:

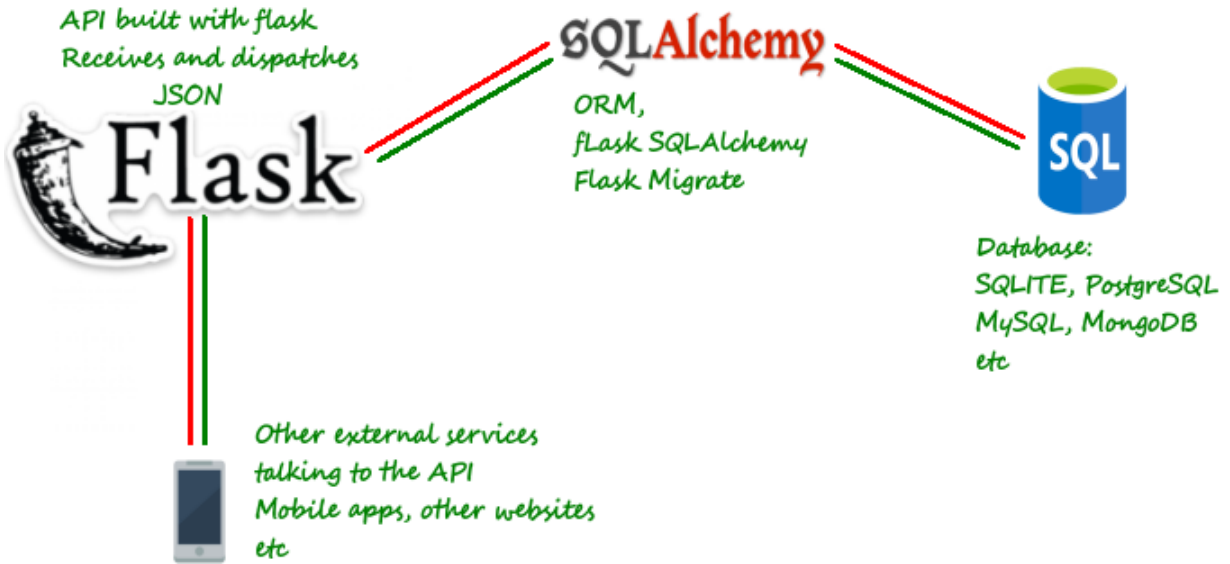
```
{
  "Recipes": [
    {
      "name": "Tavuk Sote",
      "image": "lib/assets/recipes/tavuk_sote.png",
      "rating": 5,
      "calorie": 288,
      "carb": 7.67,
      "protein": 28.46,
      "fat": 15.2,
      "cooking_minutes": 30,
      "description": "Tavuk Sote için; tavukları küçük kuşbaşı şekliyle doğra, bir tavada 2 tatlı kaşığı zeytinyağ ile 5 dakika
```

```

sotele. Ardından tavukları bir tabağa al. Aynı tavaya 2 tatlı kaşığı
zeytinyağını ekleyip küçük küçük doğradığın biberleri sotelemeye
başla. Biberler yumuşadıktan sonra, biber salçasını tavaya ekle ve
karıştırarak kavurmaya devam et. Sarımsakları küçük küçük doğrayıp
ekle ve karıştır. Karıştırdıktan sonra tavukları tavaya ekle.
Malzemeleri iyice karıştırdıktan sonra küp küp doğradığın domatesleri
ekle. Domatesler yumuşadıktan sonra baharatları ekleyip karıştır.
Ardından sıcak suyu tavaya ekleyip karıştır. Suyunu çektikten sonra
tavuk sotenin altını kapat. Tavuk Sote servise hazır."
    },
    . . .
  ]
}

```

Görüldüğü gibi web sunucusu gerekli verileri JSON formatında çıktı olarak sunmaktadır. Bu durum başka projelerde besin ve yemek tarifi bilgilerine ihtiyaç duyulması durumunda kolayca verileri almak için bir API oluşturmamızı sağlamıştır.

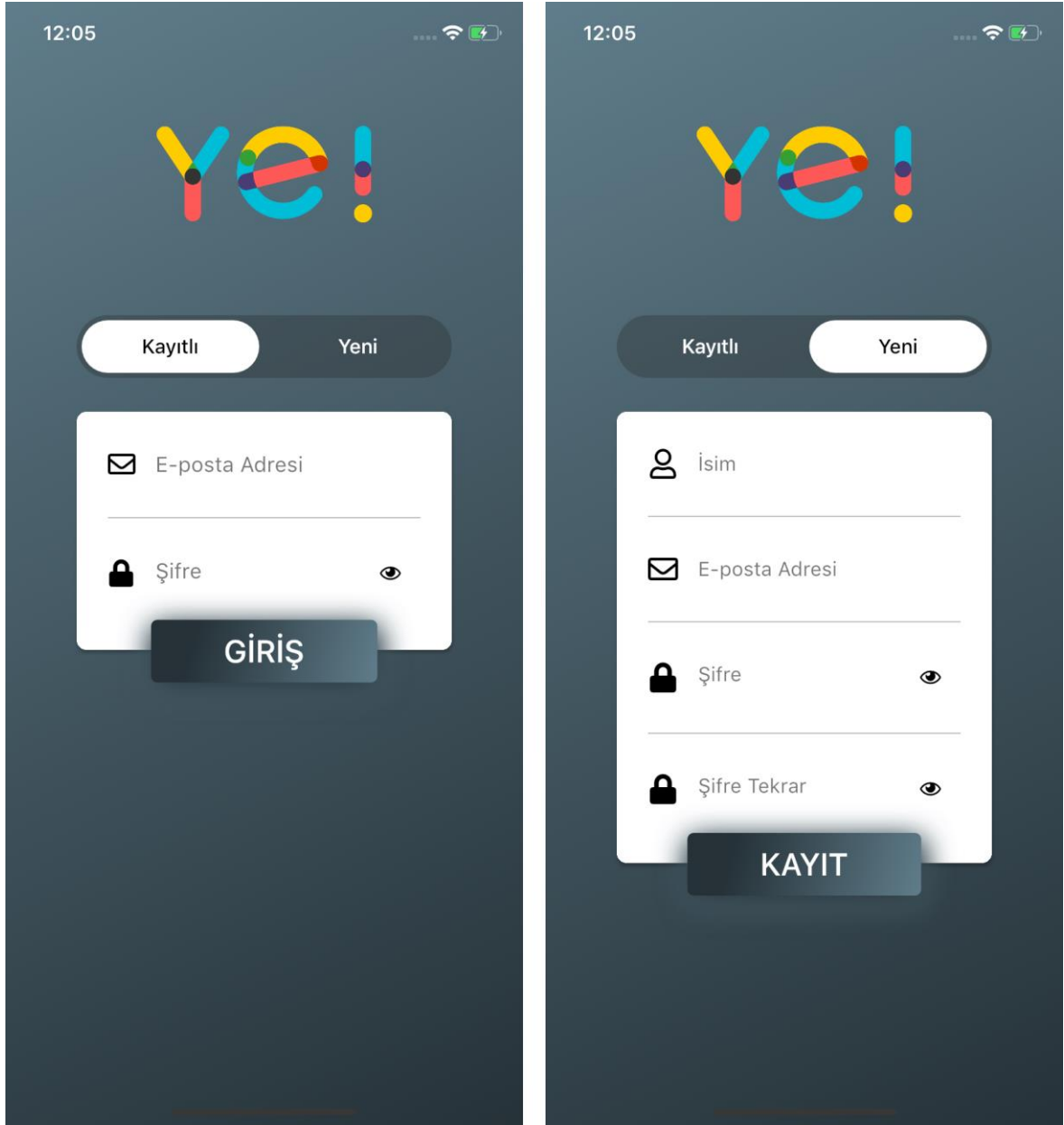


Şekil 2.2 Web Sunucu Tasarımı [8]

2.2.3 Mobil Uygulama Tasarımı

Mobil uygulamayı kullanabilmek için üye olunması gerekmektedir. Uygulama ilk açıldığında bir üye girişi ekranı ve yana kaydırıldığında üye olma ekranı ile karşılaşılmaktadır. Uygulamanın yemek önerileri için kullandığı A-priori algoritmasının daha verimli çalışabilmesi için kullanıcı verilerine (favori listesi) ihtiyaç vardır. Veri tabanımızın büyüyen uygulamamızın daha iyi öneriler sunması için üyelik zorunlu tutulmuştur.

Uygulamanın ilk ekranı kullanıcı giriş ekranıdır. İlk defa uygulamayı kullanacak kullanıcılar bu ekran üzerinden kaydolabilir.

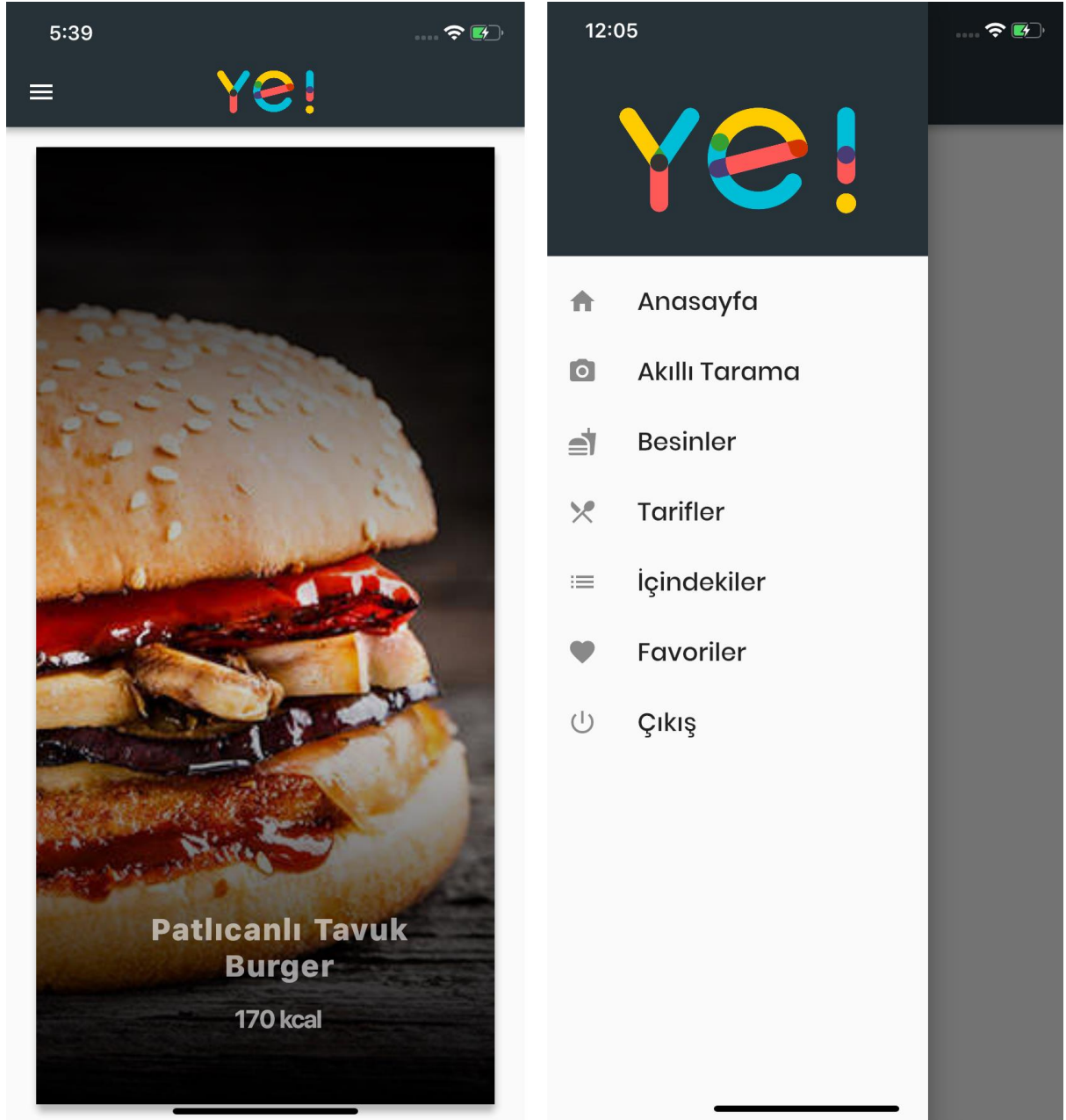


Şekil 2.3 Mobil uygulama giriş ve kayıt ekranı

Üye olmak için yalnızca bir isim, mail adresi ve şifre bilgileri yeterlidir. Üye girişi yapıldıktan sonra menüden çıkış yapılana kadar kullanıcı uygulamada aktif kalır. Uygulama kapanıp açıldığında tekrar giriş yapılması gerekmez.

Giriş yapıldıktan sonra kullanıcılar uygulamanın ana sayfasına yönlendirilmektedir. Üye olan her kullanıcı uygulamanın tüm özelliklerinden kolayca faydalanabilmektedir.

Ana Sayfa:

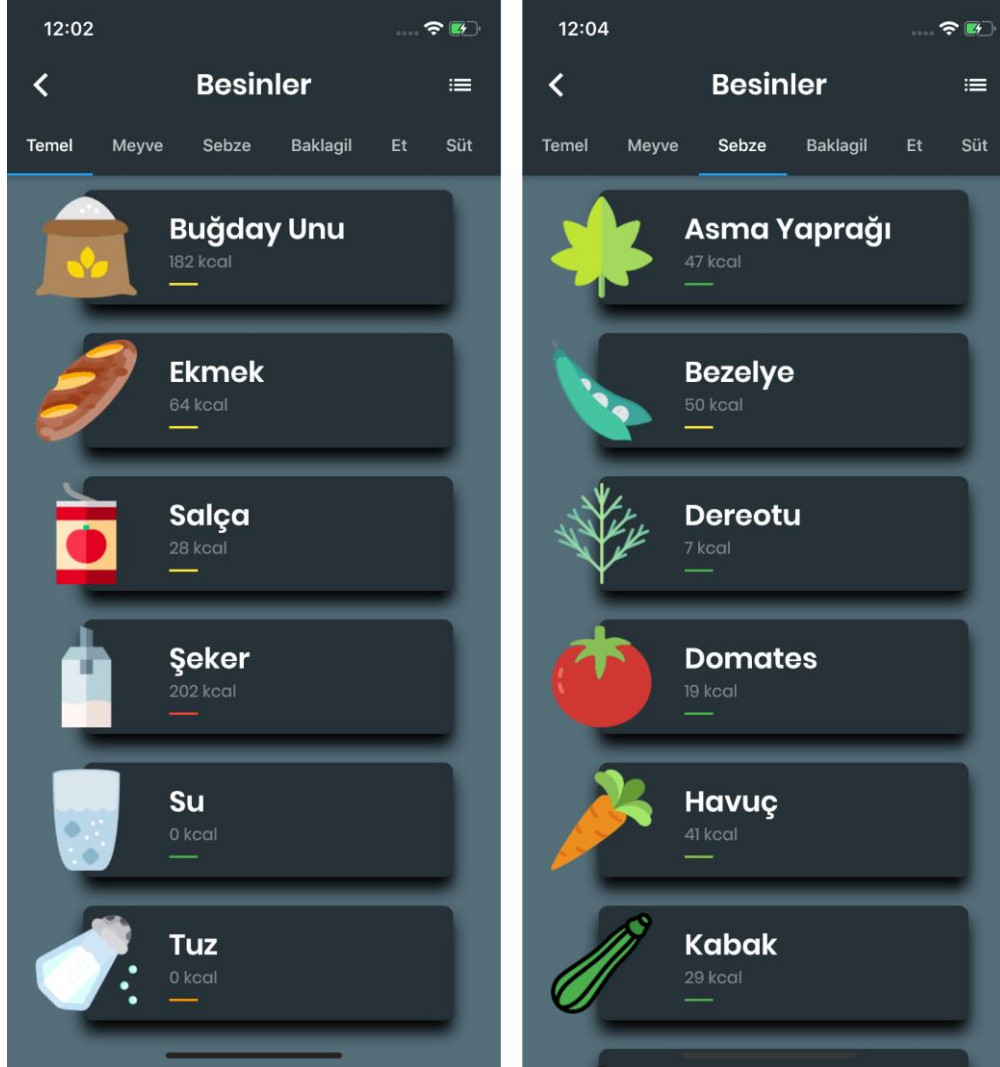


Şekil 2.4 Mobil uygulama ana ekranı

Uygulamanın ana ekranında, veri tabanındaki yemeklerden rastgele birisi seçilerek günün yemeği olarak gösterilir. Yandan açılan bir çekmece menüler arası gezinti yapacak şekilde tasarlanmıştır. Her bir menü uygulamanın amacı doğrultusunda hazırlanmış ve kullanıcı deneyimini en üst seviyeye çıkaracak görsel açıdan zengin bir arayüz sunmaktadır.

Mobil uygulama ařağıdaki menülerden oluřmaktadır.

Besinler Menüsü:



Şekil 2.5 Mobil uygulama besinler menüsü

Bu menüde kategoriler halinde besinler listelenmiştir. Toplam 8 kategori ařağıdaki şekildedir.

- 1) Temel
- 2) Meyve
- 3) Sebze
- 4) Baklagil
- 5) Et
- 6) Süt
- 7) Çerez
- 8) Yağ

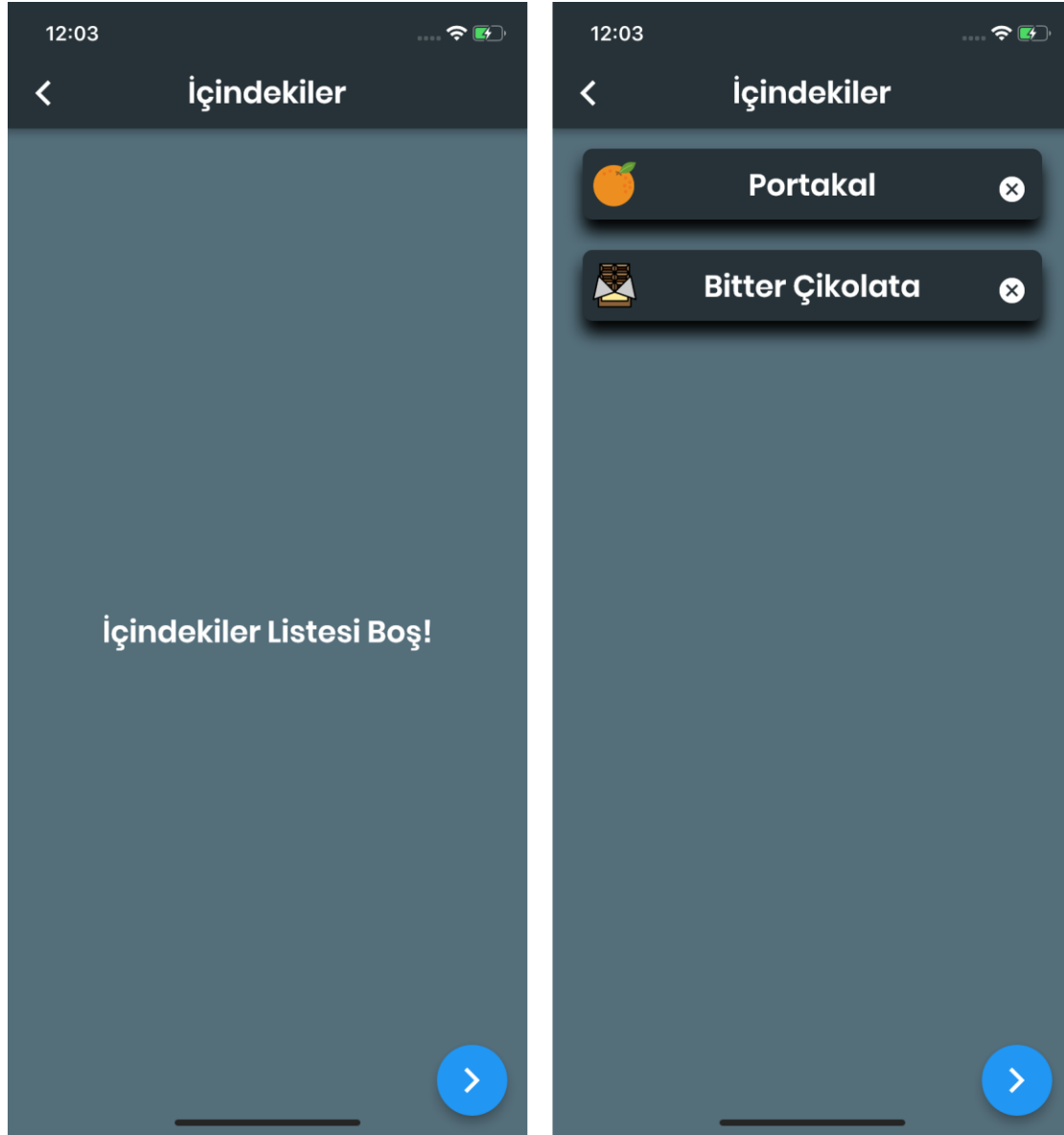
Görsel açıdan zenginleştirmek için besinlere ikon eklenmiştir. Ayrıca besinlerin altında sağlık puanını simgeleyen bir bar mevcuttur. Bu bar, sağlıklı besinler için yeşil, orta seviyedeki besinler için sarı, sağlıksız besinler için kırmızı rengini almaktadır.

Listeden herhangi bir besine tıklandığında o besinin bilgilerini içeren ekran açılmaktadır. Bu ekranda besinin kalori, protein, karbonhidrat ve yağ değerleri ile sağlık seviyesi öğrenilebilir. Yine bu ekranın altında bulunan listeye ekle butonu ile besin içindekiler listesine eklenebilir.



Şekil 2.6 Mobil uygulama besin detayları sayfası

İçindekiler Menüsü:

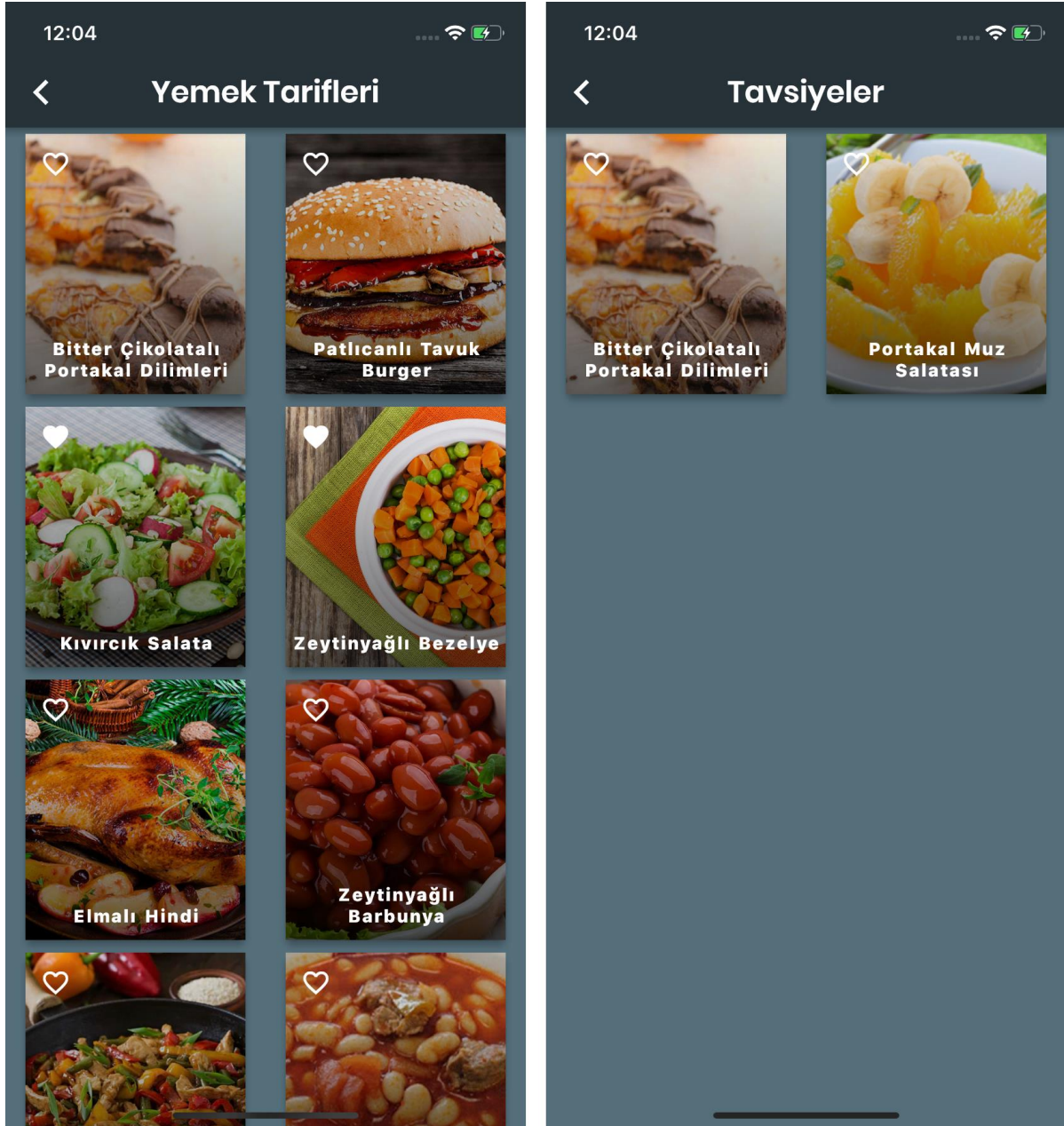


Şekil 2.7 Mobil uygulama içindekiler menüsü

Bu menüde içindekiler listesine alınan besinler görüntülenir. İstenilen besinler bu listeden silinebilir. İçindekiler listesi en fazla 10 adet besin içerebilir. Daha fazla besin eklenmeye çalışıldığında uygulama uyarı mesajı verecektir.

İçindekiler menüsünün sağ altındaki buton ile yemek önerileri alınır. Bu butona basıldığında web sunucuya besin listesi gönderilir ve bu besin listesi kullanılarak çalışan algoritma tarafından uygun yemek tarifleri mobil uygulamada görüntülenir.

Tarifler Menüsü:



Şekil 2.8 Mobil uygulama tarifler ve tavsiyeler ekranı

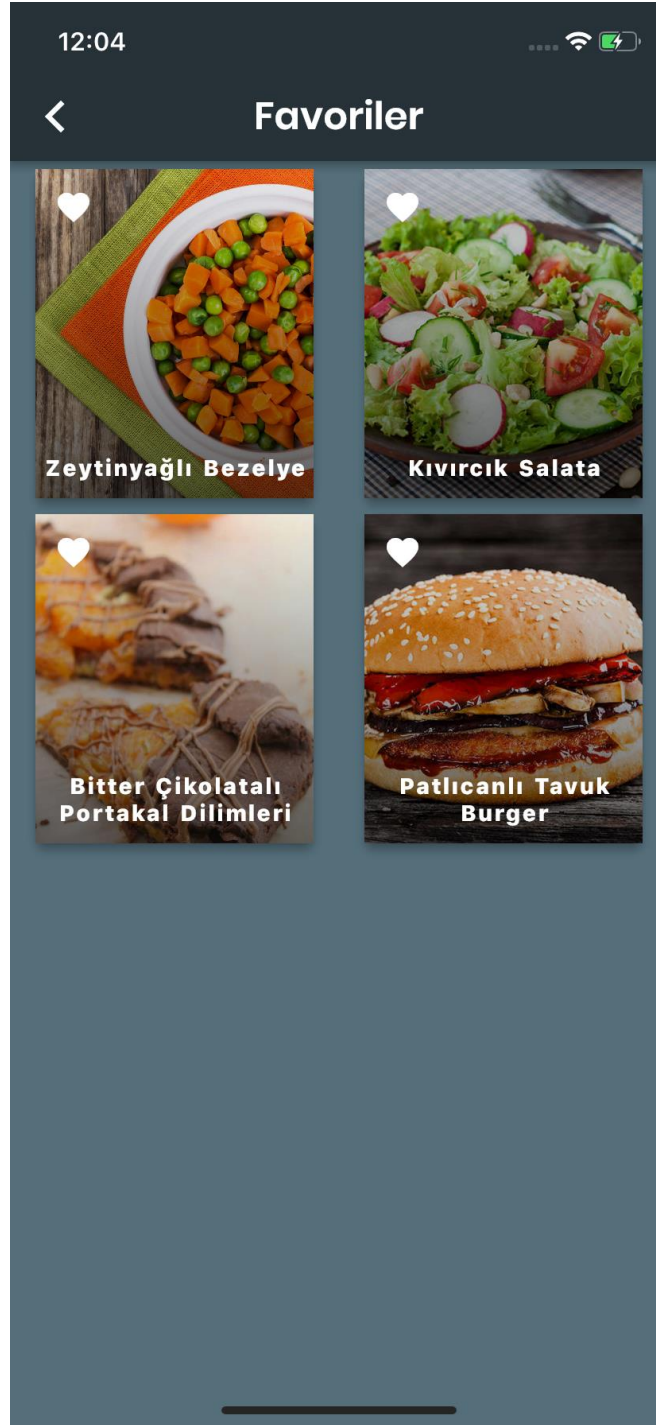
Herhangi bir öneri yapılmadan da uygulamadaki yemek tariflerine ulaşılmasını sağlayan bu menü, mevcut tüm yemek tariflerini listelemektedir. Bu menü üzerinden istenilen yemekler sol üstünde bulunan kalp ikonu ile favoriler listesine eklenebilir. Favoriler listesi daha iyi yemek önerileri sunmak için kullanılmaktadır.



Şekil 2.9 Mobil uygulama yemek tarifi sayfası

Yemek tarifleri menüsündeki herhangi bir yemek seçilerek yemekle ilgili bilgileri içeren ve yemek tarifinin yer aldığı sayfaya geçilebilir. Bu sayfanın altında o yemek için gerekli besinlerin de bir listesi bulunmaktadır. Ayrıca yemeğin kaç kişi tarafından favori listesine alındığı bilgisine de yine bu ekran üzerinden ulaşılabilir.

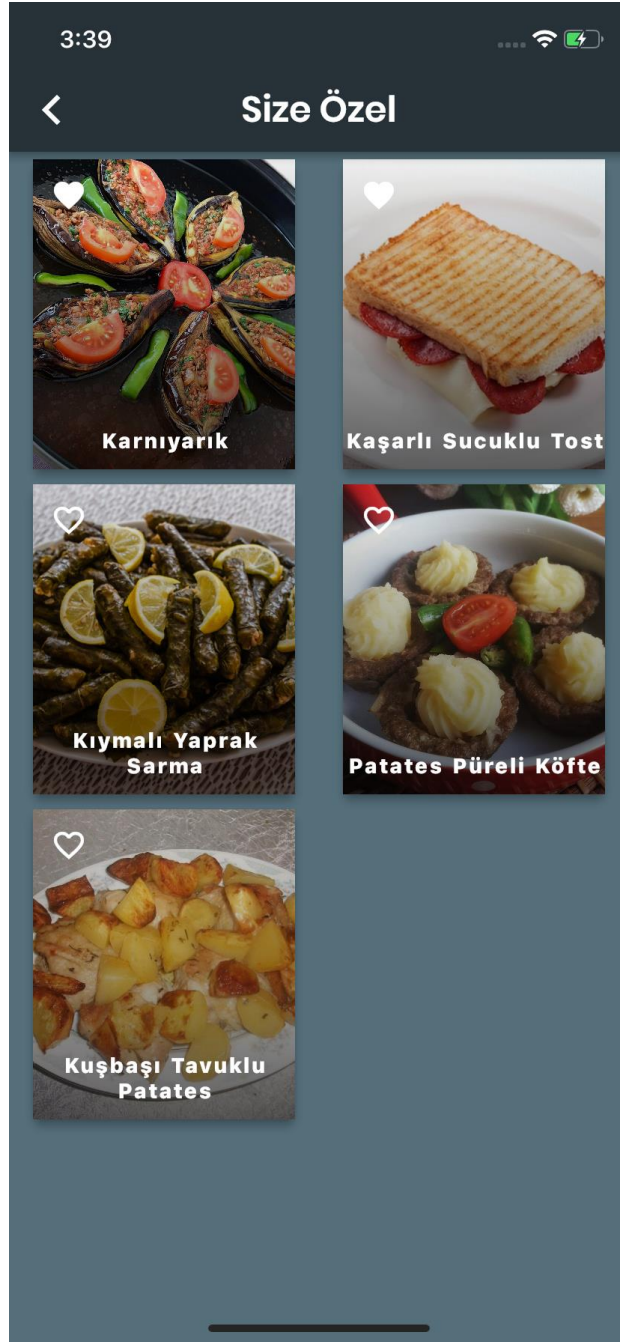
Favoriler Menüsü:



Şekil 2.10 Mobil uygulama favoriler menüsü

Burada kullanıcı tarafından favori listesine alınmış yemekler listelenir. Böylece kullanıcı en sevdiği yemek tariflerine hızlıca erişebilir. İster tarifler menüsünde ister bu menüden kullanıcı istediği takdirde bir yemeği favori listesinden çıkarabilir.

Size Özel Menüsü:



Şekil 2.11 Mobil uygulama size özel menüsü

Size özel menüsünde A-priori algoritması kullanılarak oluşturulmuş yemek önerileri yer almaktadır. Kullanıcının mevcut favoriler listesi göz önünde bulundurularak beğenebileceği farklı lezzetler burada kullanıcıya önerilir. Bu menüdeki öneriler tamamen diğer kullanıcıların favori listelerine bağlı olduğundan kullanıcı sayısı arttıkça öneriler daha etkili olacaktır.

2.2.4 Öneri ve Örüntü Tanıma Sistemi Tasarımı

Öneri sistemi için “1.5 Öneri Sistemleri” başlığında anlatılan algoritmalar kullanılarak kullanıcılara yemek önerilerinin sunulması sağlandı.

Projenin ilerleyen aşamalarında mobil uygulamamızda bulunan akıllı tarama menüsü üzerinden bir besin tanıma sistemi gerçekleştirilecektir. Uygulama, akıllı tarama menüsüne girildiğinde mobil cihazın kamerasını kullanarak bir fotoğraf çekme arayüzü sunacaktır. Bu arayüz üzerinden kullanıcı elindeki besinlerin fotoğrafını çekerek uygulama tarafından tanınmasını sağlayacaktır.

Örnek bir senaryoda, kullanıcı dolapta bulunan besinlerin fotoğraflarını çekerek hızlı bir şekilde içindekiler listesine ekleyebilir. Ardından uygulamadan tanınan bu besinleri kullanarak yemek önermesini isteyebilecektir. Her bir besinin fotoğrafı çekildiğinde ekranda o besinin bilgilerinin bulunduğu ekran açılacaktır. Böylece çekilen besinin doğru bir şekilde tanınıp tanınmadığı kullanıcı tarafından da kontrol edilebilir.

Örüntü tanıma teknikleri kullanılarak işlem yapılacağından tanıma işlemi uygulamaya bir yük getirmektedir. Mobil bir cihaz bu tarz bir işlemi gerçekleştirmek için yeterince performanslı olmayabilir. Bu sebeple besin tanıma işlemini yapacak modelin uygun bir sunucu üzerinde çalıştırılması düşünülmektedir.

Planlanan tasarımda, mobil uygulamadan çekilen fotoğraf sunucuya gönderilecek ve sunucu tarafında besin tanıma işlemi gerçekleştirildikten sonra geriye bir cevap dönecektir. Mobil uygulama, gelen cevaptaki besinin bilgilerini ekranda görüntülenecek olup kullanıcıya besin listesine ekleme seçeneği sunacaktır.

Bu konuda daha önce yapılmış tanıma sistemleri araştırılarak örnek projeler incelendi. [9] Bitirme çalışması kapsamında görüntüden besin tanıma sistemi uygulamaya eklenecektir.

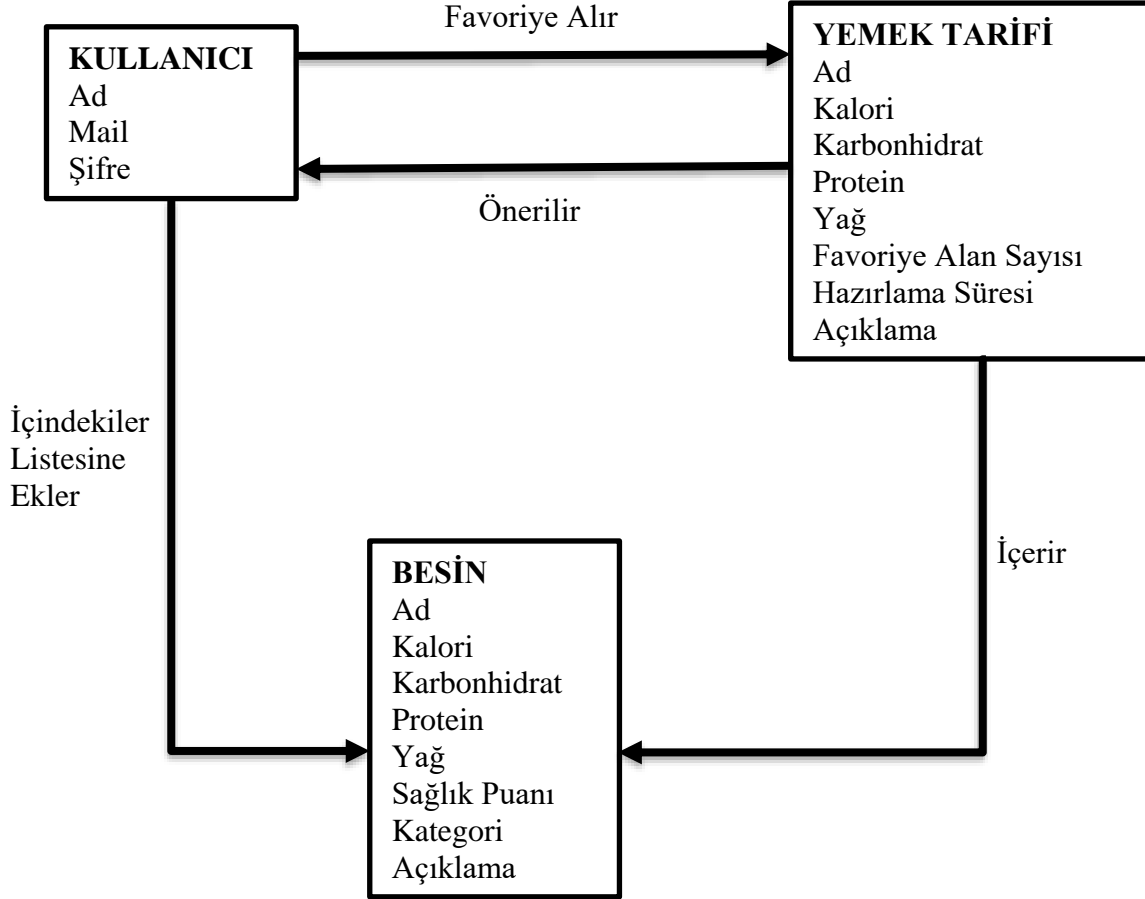


Prediction	Probability
sushi	99.91083%
cup cakes	0.08749%
sashimi	0.00166%
grilled salmon	0.00000%
dumplings	0.00000%

Şekil 2.12 Örnek yemek tanıma sistemi [9]

2.3 UML Nesne Modeli

Uygulamamız 3 temel nesneden oluşmaktadır. Nesne modeli aşağıdaki gibidir.



2.4 Yapılan Çalışmalar

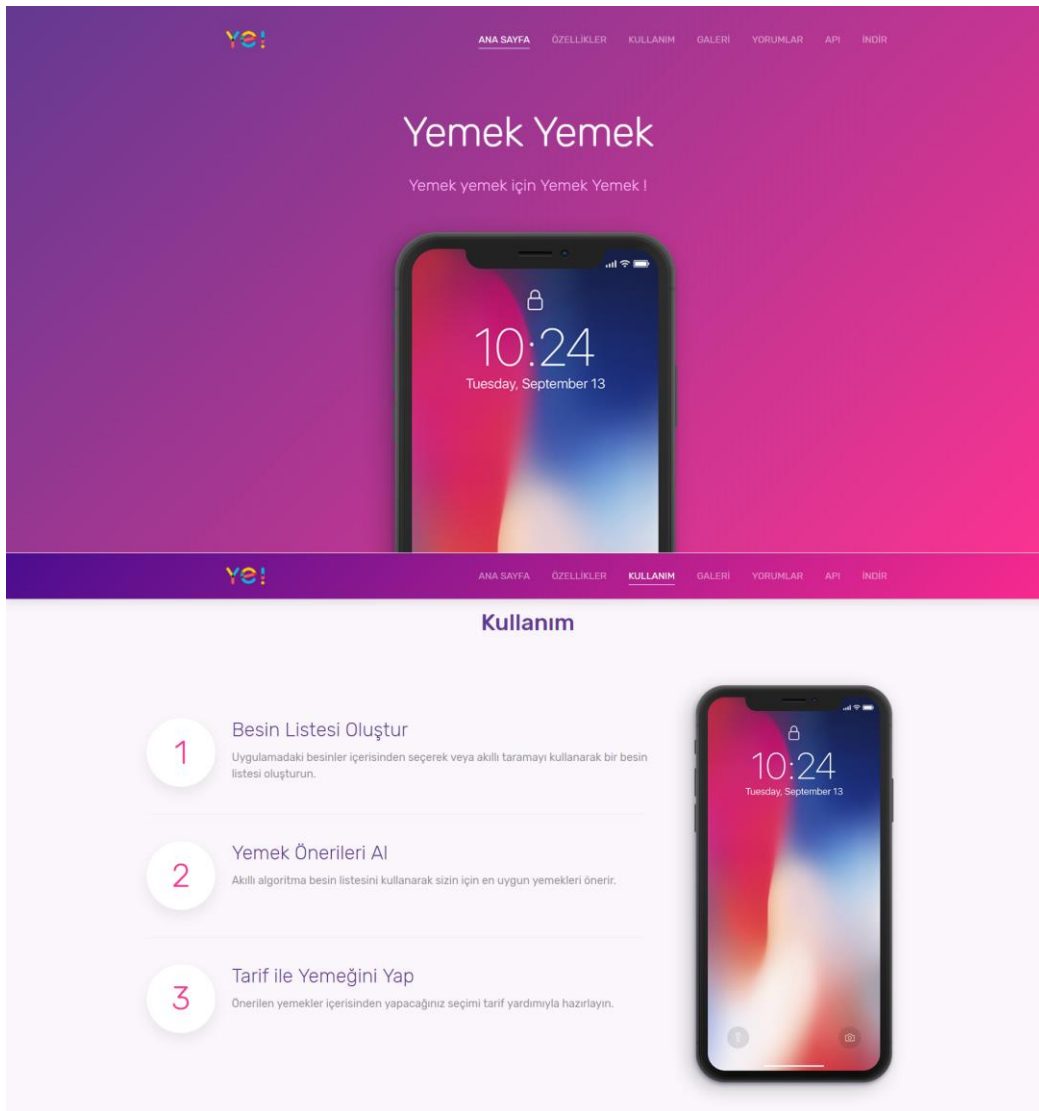
Her şeyden önce projemiz için gerekli olan veri tabanı modeli oluşturuldu. Daha sonra bu veri tabanına veri ekleme, veri silme ve veri çekme işlemlerini gerçekleştirecek bir web sunucusu gerçekleştirildi. Ardından veri tabanına başlangıçta belirli sayıda veri elle girildi. Bu işlemler yapılırken bir taraftan Flutter ile mobil uygulama yazma üzerine araştırmalar yapılarak projeye yardımcı olacak kaynaklar bulundu.

Flutter kullanılarak uygulamanın arayüzü tasarlandı. Besinler, yemek tarifleri, içindekiler, favoriler, gibi gerekli menüler oluşturuldu. Ardından web sunucusu yardımıyla veri tabanından çekilen veriler mobil uygulama üzerinde gösterildi. Besin bilgileri, yemek tarifleri gibi detay içeren sayfalar ayarlandı. Uygulama gerekli tüm verileri göstermeye başladıktan sonra favori listesine ekleme/çıkarma işlemi ve besinleri içindekiler listesine ekleme/çıkarma gibi fonksiyonel işlemler gerçekleştirildi.

Yapılan testler ile uygulamanın hem iOS hem Android cihaz üzerinde görsel olarak ayarlamaları yapıldı. Farklı ekran boyutlarında da uygulamanın düzgün görüntülenmesi sağlandı. Görsel ve temel fonksiyonel işlemler tamamlandıktan sonra uygulamanın ana amacı olan öneri sistemi hazırlandı.

Öneri sisteminde iki farklı algoritma kullanıldı. Bunlardan birisi seçilen besinlerden adet olarak en fazlasını içeren yemek tariflerini daha ön sıralarda listeleyerek öneri yapan ilk algoritmadır. Bu algoritma gerçekleştirildikten sonra ek olarak A-priori algoritması da web sunucuda gerçekleştirildi. Böylece kullanıcılara beğenebilecekleri yemek önerileri sunacak uygulamamız tamamlandı.

Son olarak uygulamanın tanıtımı için statik bir web sayfası hazırlanarak kısaca uygulamanın özellikleri anlatıldı. Ayrıca Türkçe besin bilgileri ve yemek tariflerini geliştiriciler için sağlayacak bir API oluşturulup kullanımı web sayfasına eklendi. Böylece gelecekte diğer geliştiriciler, bu verilere ihtiyacı olması halinde sıfırdan bir veri tabanı oluşturmak yerine bizim verilerimiz API'den çekilerek projelerinde kullanabileceklerdir.



Şekil 2.12 Web sitesi örnek görsel

3. KAYNAKLAR

- 1) <https://tr.wikipedia.org/wiki/MySQL> MySQL. 30 Aralık 2018.
- 2) <http://www.barisdere.com/2010/12/orm-nedir-ve-ne-zaman-kullanmamaliyiz-2/> ORM. 30 Aralık 2018.
- 3) <http://devnot.com/2016/rest-mimarisi-ve-restful-servisler/> REST. 30 Aralık 2018.
- 4) <http://www.ugurkizmaz.com/YazilimMakale-1878-JSON--JavaScript-Object-Notation--Nedir--Nasil-ve-Nerede-Kullanilir-.aspx> JSON. 30 Aralık 2018.
- 5) <https://wmaraci.com/nedir/flutter> Flutter. 30 Aralık 2018
- 6) <http://bilgisayarkavramlari.sadievrenseker.com/2011/09/07/apriori-algoritmasi/> A-priori. 30 Aralık 2018
- 7) https://cdn-images-1.medium.com/max/1000/1*dbfgpeaPYcyeST9xmVuc-g.png A-priori örneği görseli. 30 Aralık 2018
- 8) <https://danidee10.github.io/images/flask-5-overview.png> Web Sunucu Tasarımı Görseli. 30 Aralık 2018
- 9) <https://github.com/stratospark/food-101-keras> Food Classification. 30 Aralık 2018

STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Yeni bir projedir. Tasarımımız projenin %100'ünü oluşturmaktadır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Öneri sistemi için kullanılan algoritma tarafımızca yazılmıştır.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Nesne yönelimli programlama prensiplerini, İşletim Sistemleri dersinde öğrendiğimiz multithreading işlemini, Windows Programlama dersinde öğrendiğimiz mobil uygulama geliştirme prensiplerini ve Veri Tabanı dersinde öğrendiğimiz ilişkisel cebir bilgilerini kullandık.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

Python Flask ile web sunucusu hazırlandı, Flutter ile mobil uygulama yazıldı.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

Uygulamamızın yalnızca yeterli seviyede bir web sunucu maliyeti ve Play Store, App Store gibi marketlere eklenmesi için gerekli geliştirici hesabı maliyeti olacaktır.

b) Çevre sorunları:

Projenin çevreye olumsuz bir etkisi bulunmamaktadır.

c) Sürdürülebilirlik:

Uygulamanın kullanıcı sayısı arttıkça hızla büyümeye devam edecektir.

d) Üretilebilirlik:

Proje sunucu hariç donanımsal gereksinime ihtiyaç duymadığı için üretilebilirliği yüksektir.

e) Etik:

Proje etik açıdan bir problem teşkil etmemektedir.

f) Sağlık:

Proje insan sağlığına yarar sağlamak amacıyla tasarlanmıştır.

g) Güvenlik:

Proje güvenlik açısından bir problem teşkil etmemektedir.

h) Sosyal ve politik sorunlar:

Projenin herhangi bir sosyal ve politik bir sorunu bulunmamaktadır.