

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

**GİRİLEN VERİLERE GÖRE
TAHMİN YAPAN REACT NATIVE
TABANLI MOBİL UYGULAMA “BÜTCEP”**

G1512.10077

**Bölüm : BİLGİSAYAR MÜHENDİSLİĞİ
Danışman : Dr.Öğr.Üyesi Mustafa AKPINAR**

**2019-2020 Güz Dönemi
T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

ÖNSÖZ

Eđitim hayatımızda desteđini esirgemeyen aileme, yakın arkadaşlarıma, tasarım çalışmasında desteđini esirgemeyen danışman hocam Dr.Öđr.Üyesi Mustafa AKPINAR'a teşekkürlerimi sunuyorum.

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER LİSTESİ.....	vi
ÖZET.....	vii
BÖLÜM 1.	
GİRİŞ.....	1
1.1.Proje Tanıtımı.....	1
1.2.Proje Amacı Ve Kapsamı.....	1
1.3.Hedefler.....	2
BÖLÜM 2.	
KULLANILAN TEKNOLOJİLER.....	3
2.1. Visual Studio Code.....	3
2.2. React Native.....	3
2.3. Firebase Veri Tabanı.....	4
2.4. Node Paket Yöneticisi (Node Package Manager).....	4
BÖLÜM 3.	
MODEL VE YAKLAŞIM.....	5

BÖLÜM 4.

BÜTCEP UYGULAMASI.....	9
4.1. React Native Teknolojisinin Projedeki Kullanım Alanları.....	9
4.2. Node Paket Yöneticisi (Node Package Manager).....	10
4.3. Firebase Veri Tabanı.....	12

BÖLÜM 5.

SONUÇLAR VE ÖNERİLER.....	14
----------------------------------	-----------

ÖZGEÇMİŞ.....	15
----------------------	-----------

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI	
DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	16

ŞEKİLLER LİSTESİ

Şekil 3.1.	Bütçe uygulamasının arayüz tasarımının zaman içerisindeki gelişimi	5
Şekil 3.2.	Veri tabanındaki kullanıcı hesapları (solda) ve örnek harcama veri seti (sağda).....	5
Şekil 3.3.	Veri görüntüleme ve giriş sayfası tasarım fikirleri	6
Şekil 3.4.	Bütçe uygulaması ana sayfa ve kullanıcı girişi sayfaları ...	7
Şekil 3.5.	Bütçe uygulaması harcamalar sayfası animasyon ve kullanım detayları	7
Şekil 4.1.	Bütçe uygulaması harcama oluştur sayfasındaki Date-Time Picker kütüphanesi ile eklenen tarih girdisi	9
Şekil 4.2.1	NPM kullanılarak timeseries-analysis kütüphanesinin uygulamaya dahil edilmesi	10
Şekil 4.2.2	Timeseries-analysis kütüphanesi ile veri tabanından veri çekilmesi.....	11
Şekil 4.2.3	Timeseries-analysis kütüphanesi ile veri tabanından çekilen veri ile tahmin işlemi yapılması	11
Şekil 4.3.1	Harcama oluşturma kısmında Firebase'in çalışması ve denetlemeler.....	12
Şekil 4.3.2	Firestore konsol görünümü	13

ÖZET

Anahtar kelimeler: Mobil Uygulama, Veri Tahmini, Veri Analizi

Teknolojinin gelişmesiyle birlikte tüm sistemler web, mobil, bulut tabanlı uygulamalara dönüşmektedir. İnternete erişimin kolaylaşması ve erişim hızının artması, bilgiye basit ve kolay bir şekilde ulaşma, zamandan tasarruf etme isteği, web, mobil ve ya bulut tabanlı uygulamaların güvenli olması bunların başlıca sebeplerindendir.

Bununla birlikte mobil cihazlar gün geçtikçe hayatımızda daha geniş yer kaplamakta ve ihtiyaçları mobil çözümlere yönlendirme isteği doğmaktadır.

Bundan dolayı klasik hesap yöntemlerini kullanmak yerine harcamaları takip edebilmek adına mobil uygulama üzerinde basit ve kullanışlı çözüm arayışları artmaktadır. Bütçep Mobil Uygulaması bu ihtiyacı karşılayan ve eldeki girdilerle tahmini sonuçlar veren bir uygulamadır.

Sistemi kullanan kullanıcı, girmiş olduğu harcama verileri üzerinden, yeterli veri bulunması durumunda, belirli dönemler için (günlük, haftalık, aylık vb.) gerçekçi tahmin tutarlarına ulaşabilmektedir. Aynı zamanda geliştirdiğimiz uygulama ile geçmiş dönemdeki harcamaları kolayca kategorize edebilir, güncelleyebilir ya da silebilir.

BÖLÜM 1. GİRİŞ

Gelişen teknolojiyle birlikte tüm sistemler web, mobil tabanlı uygulamalara dönüşmektedir. Bunun başlıca sebepleri internet erişim hızındaki artış, zaman tasarrufu, web veya mobil tabanlı uygulamaların güvenli olmasındandır.

Bununla birlikte günümüzde insanların yaptığı harcama miktarı ve çeşitliliği, ürün pazarlarının genişlemesi ve temel ihtiyaçların günümüze göre şekillenmesine bağlı olarak artış göstermiştir. Bu harcamaların kaydını tutmak artık neredeyse ihtiyaç haline gelmiş durumdadır.

Bu durumda hem hızlı hem de istekleri karşılayacak bir mobil uygulama insanlara yardımcı olacaktır. Uygulamamız harcama verilerinin kaydını kişisel olarak tutacak ve istek doğrultusunda içerisindeki tahmin algoritması ile gelecek harcamalara dair ipuçları verecektir.

1.1. Proje Tanıtımı

Bütcep, React Native tabanlı, IOS ve Android platformlarıyla uyumlu bir mobil uygulamadır.

1.2. Proje Amacı Ve Kapsamı

Bütcep'in amacı internet erişiminin olduğu her yerde kullanıcı hesaplarına giriş yaparak, harcamaların kategori, firma ve fiyat bilgilerinin saklanmasıdır. Bununla beraber girilen fiyat verileri ışığında bir sonraki döneme (gün, hafta, ay, 3ay) ait harcama miktarını tahmin edebilmektedir.

1.3. Hedefler

Harcama verilerini detaylarıyla veri tabanında tutmak, tutulan verileri kullanıcıya sunmak ve veri üstünde deęişikliğe izin vermek. Kullanıcılara harcamaların fiyat verilerinden –girilen veri seti yeterli ise- günlük, haftalık, aylık veya üç aylık süreçler için tahmin çıkartmak.

BÖLÜM 2. KULLANILAN TEKNOLOJİLER

2.1. Visual Studio Code

Visual Studio Code, Microsoft tarafından üretilen, Windows, Linux ve Mac işletim sistemlerinde çalışabilen bir text editördür. Eklentiler ile birlikte Visual Basic.NET, Visual C# ve Visual C++ gibi dillerin yanı sıra Node JS, Ruby, Python gibi bir çok programlama dilini desteklemektedir.

Projede Visual Studio Code Windows sürümünü kullanılmıştır. Programı Microsoft'un kendi sitesinden ücretsiz olarak indirdik. Kullanım açısından gayet basit olan Visual Studio Code, sade arayüzüyle de kullanıcıya kolaylık sağlamakta. Kendi terminaline sahip olması ve yapılan hataları gösterme özellikleri sayesinde nerede hata yaptığınızı rahat bir şekilde bulmanızı sağlıyor. Akıcı ve sade tasarımı ile yazılımcının birçok işi aynı anda yürütmesine de olanak sağlıyor.

2.2. React Native

React Native, Facebook tarafından geliştirilen, çoklu mobil platform destekli uygulama geliştirme olanağı sağlayan bir teknolojidir. Kendine has bir dil formatı olan React Native bizlere tek bir dil üzerinden kodlama yapabilme olanağı sağlıyor.

Proje hazırlarken React Native kullanılarak uygulama çoklu platform seçeneği ile geliştirilmiştir. Çoklu platform desteği ise Javascript kodlarını native dile (Java, Objective-C, ..) çevirirken "Babel" adı verilen bir yapı ile sağlanıyor.

Yazılım geliştirme açısından halen geliştirilmekte olan bir teknoloji olduğundan kaynak eksikliği barındırsa da topluluk desteği ve oluşturulan topluluk kütüphaneleri sayesinde gayet başarılı bir şekilde amacına hizmet eden bir teknoloji olarak göze çarpmakta.

2.3. Firebase Veri Tabanı

Firebase, Firebase Inc. tarafından geliştirilen, daha sonra Google çatısı altına geçen bir uygulama geliştirme platformudur.

Bu uygulamada veri tabanı seçimi olarak Firebase kullanılmaktadır. Firebase kullanımı ile birlikte, standart bir veri tabanında oluşturulması ve bağlanması gereken tablolara gerek kalmadan veri tabanı işlemleri gerçekleştirilmektedir.

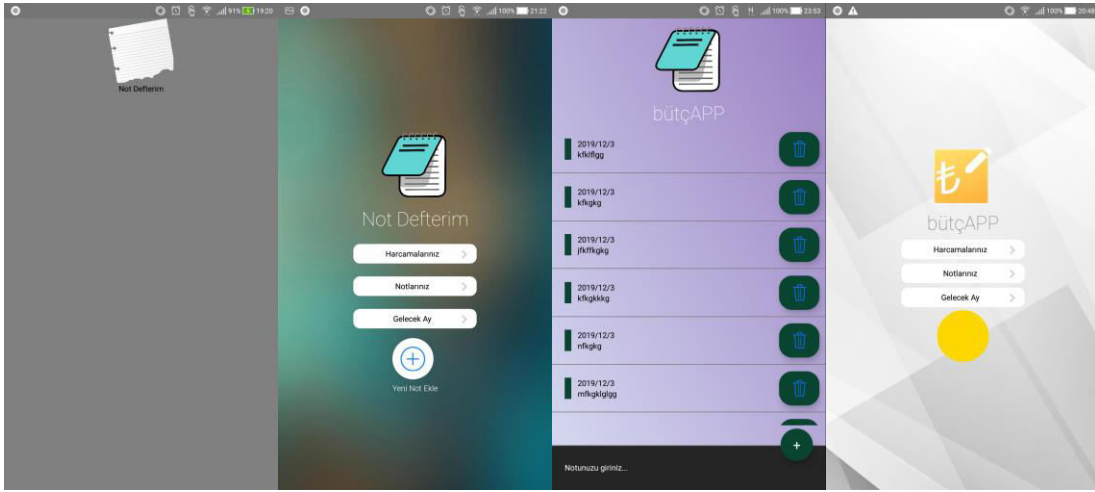
2.4. Node Paket Yöneticisi (Node Package Manager)

Node Paket Yöneticisi ya da kısaca Npm Javascript paket yöneticisidir. Npm Isaac Z. Schlueter tarafından tamamı Javascript kullanılarak geliştirilmiştir. İlk sürümü 2010 yılında yayınlanan Npm, özünde 3. parti yazılımları yüklemeyi sağlar. Temel işlevini daha iyi anlamak amacıyla Npm'i bir uygulama marketi olarak düşünürsek, Npm üzerinden yüklediğimiz paketler de uygulama marketinden yüklediğimiz uygulamalar olarak düşünülebilir.

Npm ile temel otomatik veya manuel paket yükleme, paket silme, paket listeleme, paket güncelleme gibi işlemler yapılabilir. Npm'i kurmak için öncesinde nodejs kurulumunun yapılmış olması gerekir. Node yüklendiğinde npm otomatik olarak yüklenir. Npm'in oluşturduğu Package.json dosyası ile proje hakkında bilgilere ulaşılabilir.

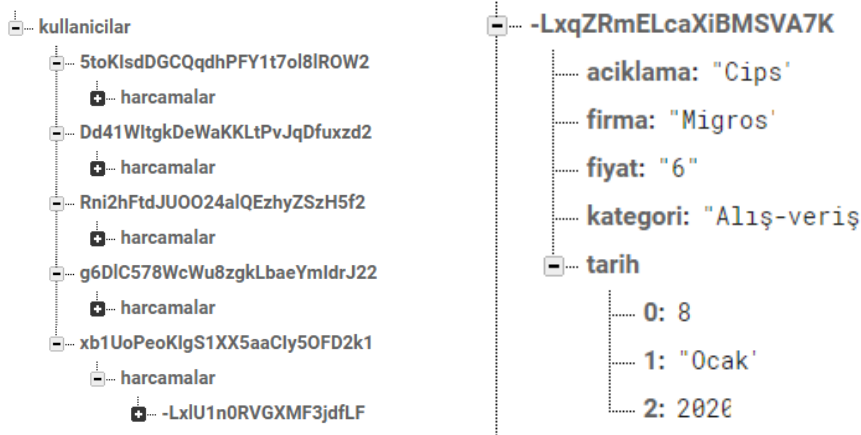
BÖLÜM 3. MODEL VE YAKLAŞIM

Bu çalışmada sade ve minimal bir arayüz kullanımı uygun görülmüştür. Gereksiz ve karışık ekranlar yerine, amacına hizmet eden ve kullanım rahatlığı açısından basit ekranlar ile kullanıcıyı yormamak amaçlanmıştır. Bu amaçla yola çıkılarak öncelikle arayüz tasarımı, deneysel bir uygulamada gerçekleştirildi. Aşağıda bazı arayüz tasarımı fikirleri gösterilmektedir.



Şekil 3.1. Bütçe uygulamasının arayüz tasarımının zaman içerisindeki gelişimi

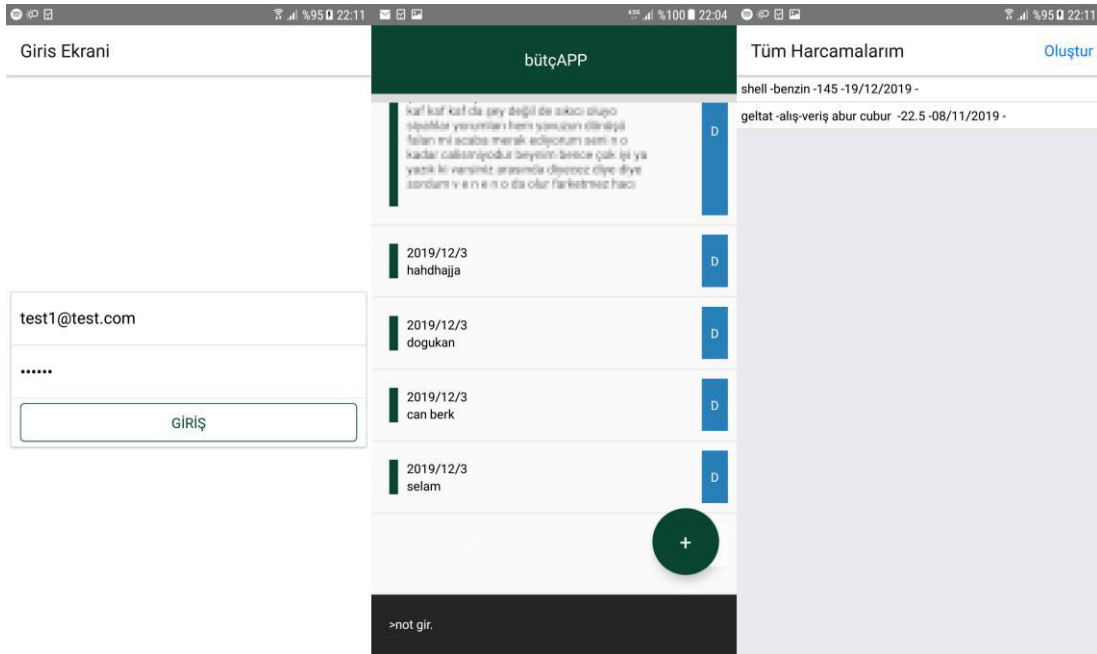
Tasarım üzerinde yapılan çalışmalarda güdülen amaç sadelik ve kullanılabilirlik esaslarından sapmadan şık bir tasarım elde etmek ve kullanıcıyı yormamaktır. Bu arayüzlere uygun olarak ihtiyacı karşılayacak bir veri tabanı tasarımı yapıldı. Aşağıda veri tabanından örnek bir parça gösterilmektedir.



Şekil 3.2. Veri tabanındaki kullanıcı hesapları (solda) ve örnek harcama veri seti (sağda)

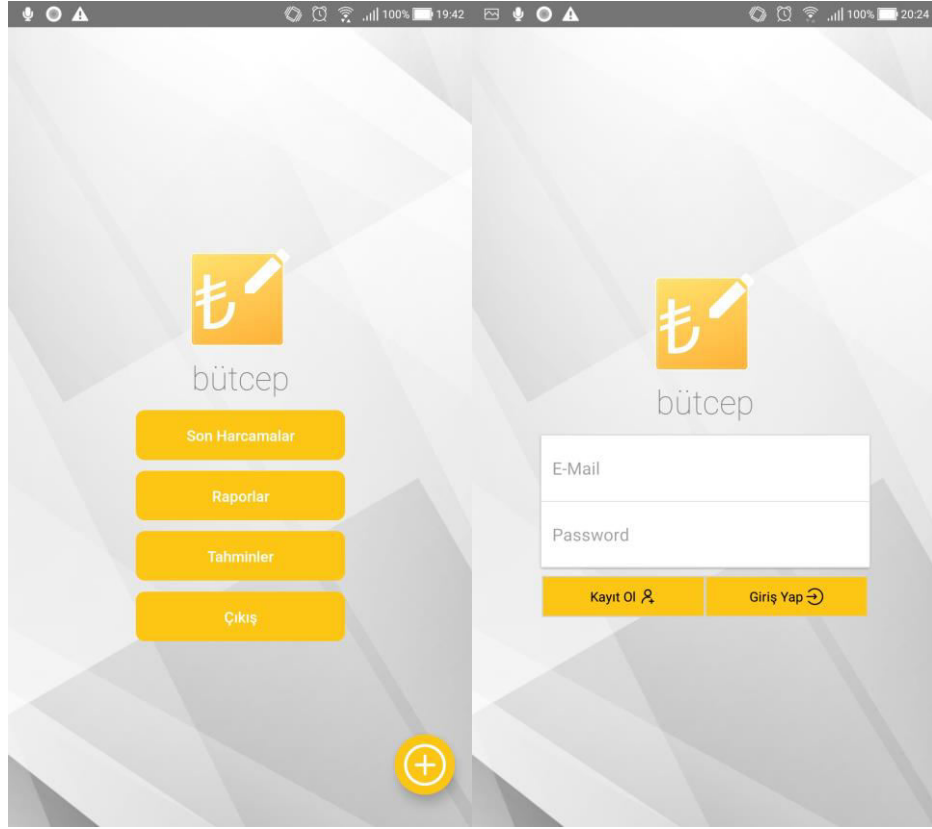
Hem arayüz hem de veri tabanı çalışmaları paralel olarak sürdürülerek projenin parçaları birbiriyle senkronize bir şekilde geliştirildi. Bu şekilde hem uygulama geliştirirken oluşacak kopukluklar giderildi hem de tam olarak ihtiyaç olan kısımlardaki problemler daha hızlı çözüldü.

Daha sonra uygulama ana sayfasından dallanan sayfalar hazırlandı. İlk tasarlandığında sadece dallanan iki sayfa bulunurken daha sonrasında hesaplama algoritmasının, kullanıcı kayıt ve giriş sayfalarının, veri görüntüleme ve düzenleme sayfalarının ihtiyaçları doğrultusunda oluşturulan sayfalarla uygulama son şeklini aldı. Aşağıda bazı veri görüntüleme ve giriş sayfası gösterilmektedir.

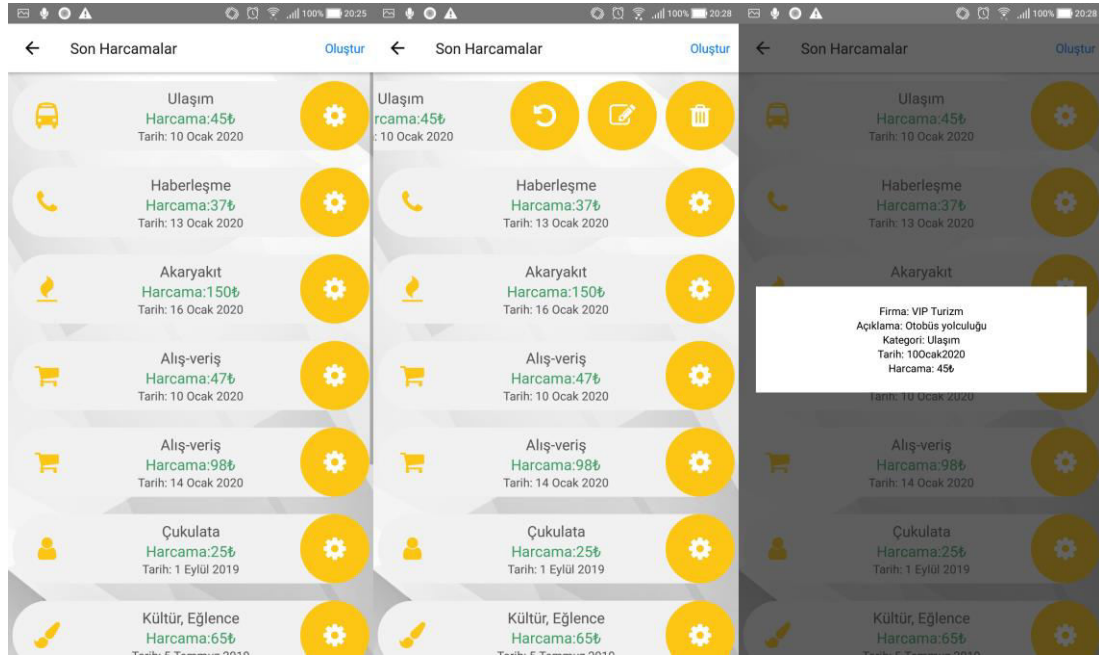


Şekil 3.3. Veri görüntüleme ve giriş sayfası tasarım fikirleri

Mevcut sayfaların veri tabanı ile bağlantısı sağlandığında sağlıklı bir şekilde veri alımı gerçekleşti. Bunun sonucunda uygulama son halini alırken göze hoş gelmesi açısından animasyon detayları gibi ufak çaplı değişiklikler de eklenerek tasarım kısmını tamamlandı. Aşağıda uygulamanın arayüzünün son hali detaylıca gösterilmektedir.



Şekil 3.4. Bütçe uygulaması ana sayfa ve kullanıcı girişi sayfaları



Şekil 3.5. Bütçe uygulaması harcamalar sayfası animasyon ve kullanım detayları

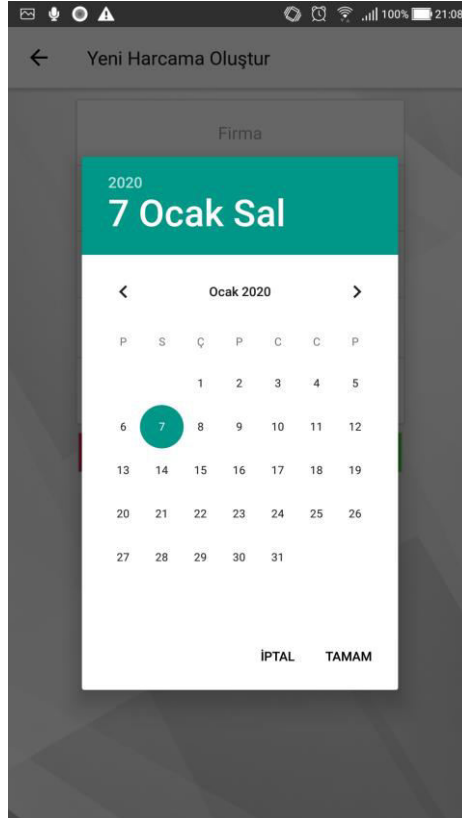
Tasarım ve veri tabanı bağlantısı kısmı tamamlanarak bir çoklu platform destekli mobil uygulama geliştirildi.

BÖLÜM 4. BÜTCEP UYGULAMASI

4.1. React Native Teknolojisinin Projedeki Kullanım Alanları

Bu uygulamada tasarım ve yapım kısmı başlamadan React Native teknolojisi kullanılmasına karar kılındı. React Native teknolojisi sayesinde çoklu platform desteği sağlanacağından yazılımcının farklı diller öğrenmesine gerek kalmaması amaçlandı. Yeni ve gelişmekte olan React Native teknolojisini proje genelinde tasarım kısımlarında kullanarak tek bir seferde, tek bir dille geliştirme sağlandı. Örneğin uygulamanın her bir parçası her platformda her cihazda aynı yerde görünmekte ve aynı işlevi yerine getirmektedir.

Ayrıca React Native teknolojisinin barındırdığı topluluk desteği sayesinde erişim sağlanan kütüphaneler ile görsel ve kullanım bakımından kolaylık sağlayan araçlar kullanıldı. Aşağıda kullanılan React Native teknolojisinin kütüphanelerinden bir örnek gösterilmektedir.



Şekil 4.1. Bütçe uygulaması harcama oluştur sayfasındaki Date-Time Picker kütüphanesi ile eklenen tarih girdisi

Ayrıca React Native teknolojisi sayesinde uygulama içerisine eklenecek birçok görsel ve hareketli kısmın tıpkı web uygulaması geliştirir gibi olması yazılımcının işini kolaylaştırmaktadır.

Yeni eklenti veya kütüphane kurulumlarında ise React Native teknolojisi, NPM sayesinde gayet hızlı ve erişimi kolay bir şekilde paketlerini kurmaktadır.

4.2. Node Paket Yöneticisi (Node Package Manager)

Kütüphanelerde özellikle sıfırdan geliştirilmesi çok uzun sürecek veya yazılım kısmı zor olacak lakin uygulamalarda kullanılması gerekli olan bazı eklentiler bulunmaktadır. Örneğin tarih seçimi, tablo veya grafik çıkartma gibi konularda bu tarz kütüphaneler kullanılmaktadır.

Bütcep uygulamasında ise bazı noktalarda bu kütüphaneler sıkça kullanılmıştır. Kullanılan bu kütüphanelerin topluluk tarafından oluşturulması ve dokümantasyonunun yeterliliği, NPM sayesinde kolayca kurulması ise yazılımcının işini oldukça kolaylaştırmaktadır. Aşağıda bir kütüphanenin NPM kurulum talimatları gösterilmektedir.

installation

```
npm install timeseries-analysis
```

Şekil 4.2.1. NPM kullanılarak timeseries-analysis kütüphanesinin uygulamaya dahil edilmesi

Bütcep uygulamasında ise bu kütüphanenin kullanım şekli, veri setlerindeki fiyat bilgisinin çekilmesiyle, -yeterli veri varsa- kullanıcının belirttiği şekilde günlük, haftalık, aylık veya üç aylık harcama tahminini kullanıcıya yansıtmaktır. Aşağıda bu kütüphanenin uygulamadaki kullanımına dair kod parçacığı gösterilmektedir.


```

var dailyresult=[];
var t = new timeseries.main(timeseries.adapter.fromDB(this.props.spendingArray, {
  date: 'tarih',
  value: 'fiyat',
}));
var date = new timeseries.main(timeseries.adapter.fromDB(this.props.spendingArray, {
  value: 'tarih',
}));
var myJSON = JSON.stringify(this.props.spendingArray);
var dforecast = 0;
var wforecast = 0;
var mforecast = 0;
var thforecast = 0;

```

Şekil 4.2.2 timeseries-analysis kütüphanesi ile veri tabanından veri çekilmesi

```

var d= new timeseries.main(timeseries.adapter.fromArray(dailyfinalstr));
var dcoeffs = d.ARMaxEntropy({degree: 5});
for (var i=0;i<dcoeffs.length;i++) {
  dforecast -= d.data[(dailyfinalstr.length-1)-i][1]*dcoeffs[i];
}
if(dforecast<0){
  dforecast=0;
}
var w= new timeseries.main(timeseries.adapter.fromArray(weeklyfinalstr));
var wcoeffs = w.ARMaxEntropy({degree: 3});
for (var i=0;i<w.length;i++) {
  wforecast -= w.data[(weeklyfinalstr.length-1)-i][1]*wcoeffs[i];
}
if(wforecast<0){
  wforecast=0;
}
var m= new timeseries.main(timeseries.adapter.fromArray(monthlyfinalstr));
var mcoeffs = m.ARMaxEntropy({degree: 3});
for (var i=0;i<mcoeffs.length;i++) {
  mforecast -= m.data[(monthlyfinalstr.length-1)-i][1]*mcoeffs[i];
}
if(mforecast<0){
  mforecast=0;
}
var th= new timeseries.main(timeseries.adapter.fromArray(thirdmonthfinalstr));
var thcoeffs = th.ARMaxEntropy({degree: 2});
for (var i=0;i<thcoeffs.length;i++) {
  thforecast -= th.data[(thirdmonthfinalstr.length-1)-i][1]*thcoeffs[i];
}
if(thforecast<0){
  thforecast=0;
}

```

Şekil 4.2.3. timeseries-analysis kütüphanesi ile veri tabanından çekilen veri ile tahmin işlemi yapılması

4.3. Firebase Veri Tabanı

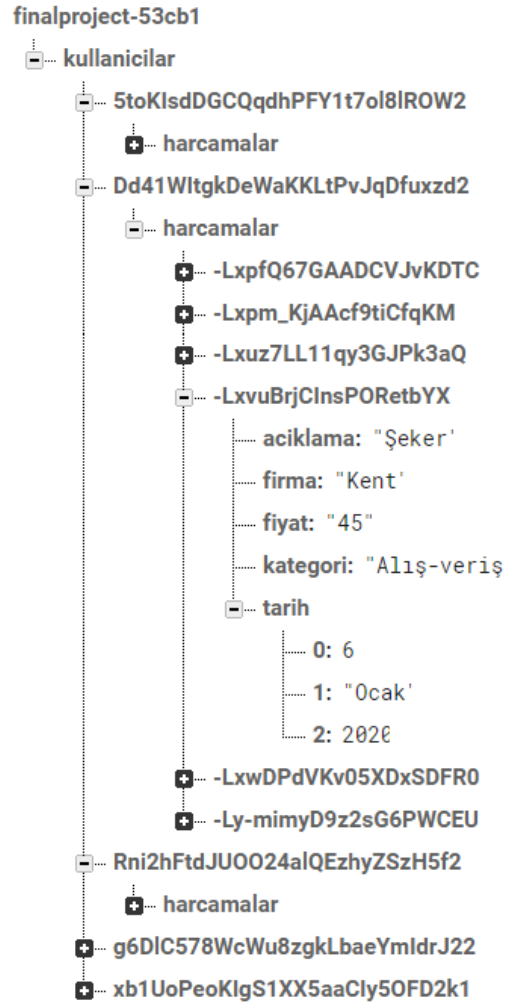
Bu uygulamada veri tabanı seçimi olarak Firebase kullanılmaktadır. Firebase kullanımı ile birlikte, standart bir veri tabanında oluşturulması ve bağlanması gereken tablolara gerek kalmadan veri tabanı işlemleri gerçekleştirilmektedir. Uygulama genelinde bakıldığında kullanıcı girişi yapılmadan uygulama içeriğine veya verilere erişim mümkün olmamaktadır. Bunun için öncelikle kullanıcı kendine düzenlenmiş veri tabanı üzerinden bir hesap oluşturmalıdır.

Oluşan hesaplar neticesinde Firebase, kullanıcılar klasörü altında tüm kullanıcıları depolamaktadır. Depolanan bu kullanıcıların her birinde kendine has harcama verileri tutulmaktadır. Veri tabanında tutulması ve tahmin işlemlerinin doğruluğu açısından oluşturulan harcama verileri fiyat, kategori ve tarih verileri olmadan gönderilmemektedir. Bu bilgiler eksiksiz girildiği takdirde veri tabanı işlem yaparak bilgileri veri seti olarak depolamaktadır. Aşağıda harcama oluşturma kısmının harcamaları veri tabanına gönderdiği kod parçacığı yer almaktadır.

```
export const harcamaOlustur = ({firma, aciklama, fiyat, tarih, kategori}) => {
  const {currentUser} = firebase.auth();
  return dispatch => {
    dispatch({type: HARCAMA_OLUSTUR_REQUEST});
    if (tarih === '' || kategori === '' || fiyat === '') {
      Alert.alert(
        'Mesaj: ',
        'Kategori, Tarih ve Fiyat alanları boş bırakılamaz!',
        [{text: 'Tamam', onPress: () => null}],
      );
      dispatch({type: HARCAMA_OLUSTUR_REQUEST_FAIL});
    } else {
      firebase
        .database()
        .ref(`/kullanicilar/${currentUser.uid}/harcamalar`)
        .push({firma, aciklama, fiyat, tarih, kategori})
        .then(() => {
          dispatch({type: HARCAMA_OLUSTUR_REQUEST_SUCCESS});
          Actions.harcamaList();
        });
    }
  };
};
```

Şekil 4.3.1. Harcama oluşturma kısmında Firebase'in çalışması ve denetlemeler

Firestore kısmında ise yapılan düzenlemeler ile uygulamanın gerektirdiği şekilde biçimlenen veri tabanını kullanılmaktadır. Firestore'in konsol sayfasından takip avantajı ile birlikte girilen verilerin takip edilmesi kolay bir hal almaktadır. Aşağıda Firestore'in konsol görünümü gösterilmektedir.



Şekil 4.3.2. Firestore konsol görünümü

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Bütçe analiz uygulaması olan Bütcep, kullanıcılara basit ve sade bir arayüz sunarak harcamalarını takip edebileceği ve girilen verilerin ışığında dönemlik tahminler sunan bir uygulama olarak hedeflenmiştir. Ayrıca kullanılan teknoloji ile bir proje ile iki farklı mobil platform kullanıcılarına ulaşmak amaçlanmıştır. Projenin devam versiyonlarında yalnızca harcamaların değil ayrıca gelirlerin de girilebilmesi ve bu şekilde aylık gelir-gider durumunun gösterilmesi, kategorilere göre harcamaların görsel olarak zenginleştirilerek sunulması, sürekli dönemlik giderlerin (kira, taksit vb.) otomatik olarak eklenmesi hedeflenmektedir.

ÖZGEÇMİŞ

Can Berk ÖZDEMİR, 09.05.1997'de Sakarya'da doğdu. İlk ve ortaokul eğitimini Atatürk İlköğretim Okulu'nda tamamladı. Lise eğitimini Tes-İş Adapazarı Anadolu Lisesi'nde aldı ve 2015 yılında mezun oldu. Aynı yıl Sakarya Üniversitesi Bilgisayar Mühendisliği'ni kazandı. Üniversite eğitimi sırasında yazılım ve donanım stajlarını sırasıyla İzmit Goodyear Lastikleri T.A.Ş. ve Sakarya Üniversitesi Uzaktan Eğitim ve Araştırma Merkezi'nde yapmıştır.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU :

ÖĞRENCİLER (Öğrenci No/AD/SOYAD):

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuzuna uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN:

DANIŞMAN İMZASI: