

İSTANBUL TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ

ENPARA FİYATLAR VE ORANLAR TAKİP
UYGULAMASI

Bitirme Projesi

İlhan KAYA
040080212

Bölüm: Bilgisayar Mühendisliği
Anabilim Dalı: Bilgisayar Bilimleri

Danışman: Prof.Dr. Uluğ BAYAZIT

Haziran 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ

ENPARA FİYATLAR VE ORANLAR TAKİP
UYGULAMASI

Bitirme Projesi

İlhan KAYA

040080212

Bölüm: Bilgisayar Mühendisliği
Anabilim Dalı: Bilgisayar Bilimleri

Danışman: Prof.Dr. Uluğ BAYAZIT

Haziran 2017

Özgünlük Bildirisi

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntıların, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

İstanbul, 2017

İlhan Kaya

Teşekkür

Bu çalışmanın gerçekleştirilmesine öncülük eden, proje süreci boyunca değerli bilgilerini paylaşan, saygıdeğer danışman hocam; Prof. Dr. Uluğ BAYAZIT’a, proje süreci boyunca sorularımı yanıtlayan ve projenin ilerleyişine katkıda bulunan IBTECH Uluslararası Bilişim ve İletişim Teknolojileri firması Enpara (QNB Finansbank) Ekibi’ne, ve 2013’ten beri her zaman yanımda olduğunu hissettiren güzel insan Emeti ZENGİN’e ve her zaman zorlukları benimle göğüsleyen ve hayatımın her evresinde bana destek olan değerli aileme sonsuz teşekkürlerimi sunarım.

İstanbul, 2017

İlhan Kaya

ENPARA FİYATLAR VE ORANLAR TAKİP UYGULAMASI

(ÖZET)

21. yüzyılın başlarından itibaren hızlı bir şekilde gelişen teknoloji ve internet ağı sayesinde akıllı telefonların kullanım alanları ve işlem kapasiteleri önemli derecede artmış oldu. Dolayısı ile bu durum akıllı telefonların farklı alanlarda kullanılmasını sağlarken, birçok yeni akıllı telefon uygulamasının (mobil uygulama) da ortaya çıkmasına olanak sağladı. Özellikle 2010 yılından sonra akıllı telefon uygulamalarında önemli bir artış yaşandı ve bir çok alanda bu uygulamalardan faydalanıldı.

Bu gelişmeleri takiben bankacılık ve finans sektörü bir çok mobil uygulama ile bu alandaki yerini aldı. Günümüzde hemen hemen her bankanın internet bankacılığının yanında mobil uygulamaları mevcuttur. Bankacılığın bu kadar sanallaşmasının sonucunda QNB Finansbank Ekim 2012’de Türkiye’nin ilk şubesiz dijital bankası olan ve tüm hizmetlerini sadece dijital kanallar üzerinden sunan Enpara’yı kurdu. Enpara bünyesinde; EFT/Havale, Kredi İşlemleri, Döviz İşlemleri, Mevduat İşlemleri ve Fatura Ödeme gibi temel bankacılık işlemleri yapılabilmektedir [1].

Bu projenin amacı, Enpara mobil uygulamasındaki Fiyatlar ve Oranlar bölümü içindeki Döviz ve Altın Fiyatları kısmını yeni bir uygulama içerisinde, müşterilere ve kullanıcılara sunmak, anlık döviz ve altın fiyat değişimleri hakkında kullanıcıları bilgilendirmek ve döviz ve altın alım-satımı için uyarı emirleri oluşturulmasına imkan sağlamaktır.

ENPARA PRICING AND RATES TRACKING APPLICATION (SUMMARY)

Rapidly evolving technology and internet network since the turn of the 21st century, the usage areas and processing capacities of smartphones have increased significantly. This is enabled smartphones to be used in different areas, allowing many new smartphone applications (mobile applications) to emerge. Especially after 2010, there was a significant increase in smartphone applications and many applications were made available to them.

Following these developments, the banking and finance sector took its place in this area with many mobile applications. In today, almost every bank has mobile banking besides internet banking. As a result of this virtualization of banking, QNB Finansbank established Enpara, which is Turkey's first digital bank in October 2012 and offers all its services through digital channels only. With Enpara; basic banking transactions such as Money Transfer, Credit Transactions, Foreign Currency Transactions, Deposit Transactions and Billing Payment can be done [1].

The aim of this project is to present the Exchange and Gold Prices section of the Enlargement Mobile Applications Pricing and Ratios section in a new application to customers and users, to inform users about instant currency and gold price changes and to create warning orders for currency and gold purchases and sales.

İÇİNDEKİLER

İÇİNDEKİLER	vi
1. GİRİŞ	1
1.1 Sanal Bankacılık ve Enpara	1
1.1.1 Sanal Bankacılık Nedir?.....	1
1.1.2 Enpara Nedir?	1
1.2 Projenin Amacı	1
2. PROJE TANIMI VE PLANI	3
2.1 Proje Tanımı	3
2.2 Proje Planı	3
2.2.1 Proje Zaman Diyagramları	4
2.2.2 Kuramsal Araştırma	5
2.3 Projenin Kaynakları.....	5
3. TEORİK BİLGİ.....	6
3.1 Uygulama ve Veri Tabanı Örnek Haberleşme Senaryosu	6
4. PROJENİN ANALİZİ VE MODELLENMESİ.....	9
4.1 Proje Analizi.....	9
4.1.1 Native Platformlar İçin.....	9
4.1.2 Android Platformlar İçin	9
4.2 Projenin Modellenmesi	10
4.2.1 JSON Yapısı	10
4.2.2 Tablo Yapısı.....	11
4.2.3 Model Yapısı	13
5. UYGULAMA VE TEST	16
5.1 Uygulama.....	16
5.1.1 Ana Sayfa (Home)	16
5.1.2 Alarm Sayfası (Alert)	17
5.1.3 Alarm Ekleme ve Güncelleme (Add Alert, Update Alert).....	17
5.2 Test.....	18
6. DENEYSEL SONUÇLAR.....	19
7. SONUÇ VE ÖNERİLER	21
8. KAYNAKLAR	22

1. GİRİŞ

1.1 Sanal Bankacılık ve Enpara

1.1.1 Sanal Bankacılık Nedir?

Günümüzde (özellikle büyükşehirlerde) bankada işlem yaptırmak büyük bir zaman kaybına sebep olabilmektedir. Kalabalık saatlerde gittiğinizde 30 dakika ile 1 saat arası beklemeyi göze almanız, sabırlı olmanız ve bu işe vakit ayırmanız gerekir. İnternet bankacılığı da denilen sanal bankacılıkta ise oturduğunuz yerden evinizin kirasını ödemeniz, mali tablonuza göz atmanız, faturalarınızı ödemeniz ve diğer tüm işlemlerinizi yapmanız mümkün. Şimdilik sadece fiziksel olarak para alamayan sanal bankalar ile bütün işlemlerinizi halledebiliyorsunuz. Üstelik hem çok daha ucuza hem de zaman kaybetmeden.

1.1.2 Enpara Nedir?

Enpara, Ekim 2012’de Finansbank tarafından kurulan, tüm hizmetlerini sadece dijital kanallar üzerinden sunan şubesiz dijital bir bankadır [1]. Enpara üzerinden şu işlemler gerçekleştirilebilir:

- Vadesiz Hesaplar
- Vadeli Hesaplar
- Para Transferi İşlemleri (EFT/Havale)
- Kredi İşlemleri
- Fatura Ödeme
- Döviz Alım ve Satımı
- Altın Alım ve Satımı

1.2 Projenin Amacı

Bu projenin amacı Enpara müşterilerine ve kullanıcılarına bir ürünün (Döviz ve Altın) istenilen fiyat oranına geldiğinde müşterinin uygulama tarafından uyarılması ve müşterinin bu doğrultuda işlem yapmasını sağlamaktır. Müşteri istediği ürün için oran uyarısını sisteme girerek, istediği ürün için giriş yaptığı orana ulaştığında uyarılması sağlanacaktır. Bu şekilde

müşteri sürekli ürün oranındaki değişimi takip etmek zorunda olmayacaktır, ürün istenen orana gelince sistem tarafından uyarılacaktır ve istediği ürüne istediği oranda işlem yapabilecektir.

Döviz ve Altın için fiyat ve oran değişimlerini takip edebilecek ve alım-satım işlemlerinin uygulandığı ürünler için müşteri belirlediği oranlarda (ürünün istenilen kara ulaşması) sistemin uyarıları sayesinde istediği oranda kar yaparak alım ve satım yapacaktır [1].

2. PROJE TANIMI VE PLANI

2.1 Proje Tanımı

Proje temel olarak bir mobil uygulamaya dayanmaktadır. Bu mobil uygulamaya veri giriř ve çıkışı ile müşteriye ürün oranı ve bu orandaki deęişimler aktarılacaktır. Müşteri bu şekilde istedięi ürünler için uygulamaya takip uyarısı girebilecek ve istedięi ürün oranındaki deęişim hakkında uyarı olarak bilgi sahibi olacaktır.

Uygulamadaki ürün oranı için uyarı giriř bölümü řu şekilde olacaktır:

- Uyarı Oluřturma: Yeni bir oran uyarısı oluřturma
- Uyarı Silme: Var olan bir oran uyarısını silme
- Uyarı Güncelleme: Var olan bir oran uyarısını güncelleme

Müşteriler veya kullanıcılar uygulamayı mobil uygulama marketlerinden cihazlarına indirip kullanabileceklerdir.

2.2 Proje Planı

Proje planlaması řu adımlardan oluřmaktadır:

- Planlama: Proje için gerekli ön araştırma yapıldı, proje planlanması gerekleřtirildi.
- Analiz ve Modelleme: Proje için gerekli analiz alıřması yapıldı, ve bu alıřma sonucu bir modelleme yapıldı. Projede kullanılacak veri akıř yöntemleri hakkında araştırma yapıldı.
- Tasarım: Projede kullanılacak ekran ve veri tablo yapıları belirlendi. Tablolar oluřturuldu.
- Geliřtirme: Proje geliřtirmesinin yapılacağı (kaynak kodunun yazılacağı) platformlar hakkında araştırma yapıldı. Kullanılacak programlama dillerine karar verildi.

- Test ve Dökümantasyon: Proje için gerekli test senaryoları gerçekleştirildi ve doküman edildi.

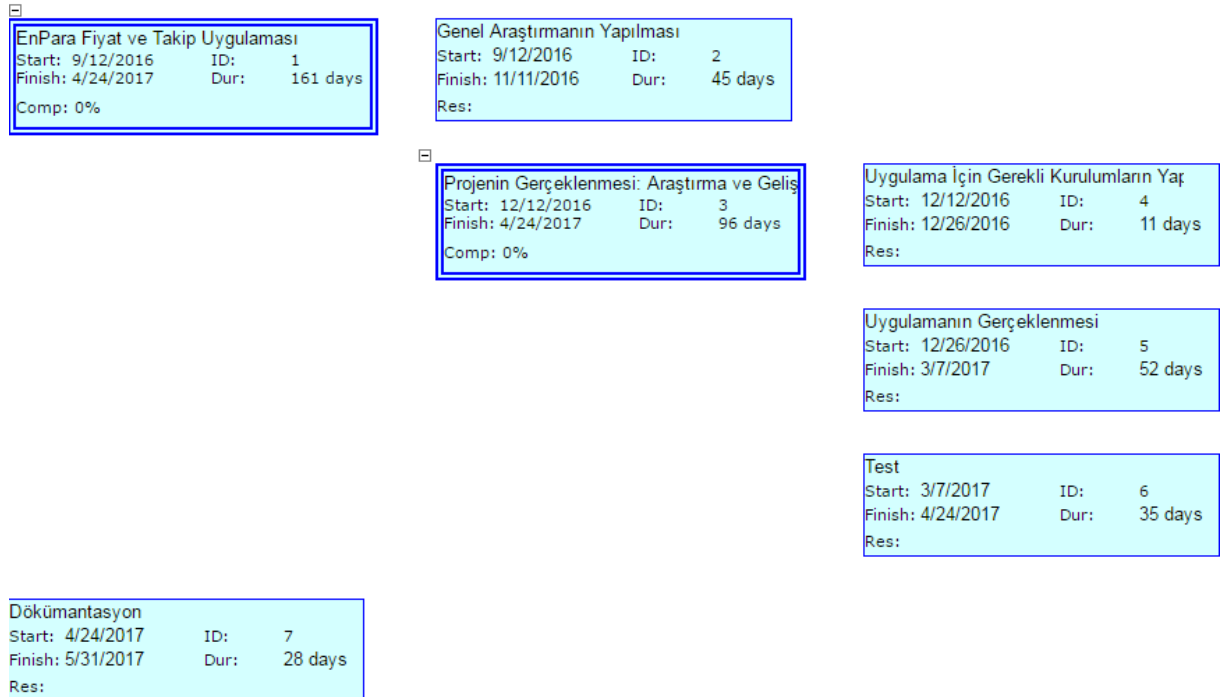
2.2.1 Proje Zaman Diyagramları

Proje planına ait zaman diyagramları şu şekildedir:

- Kıstaslar ile zaman süreci Tablo-2.1’deki ve Şekil-2.1’deki gibidir.

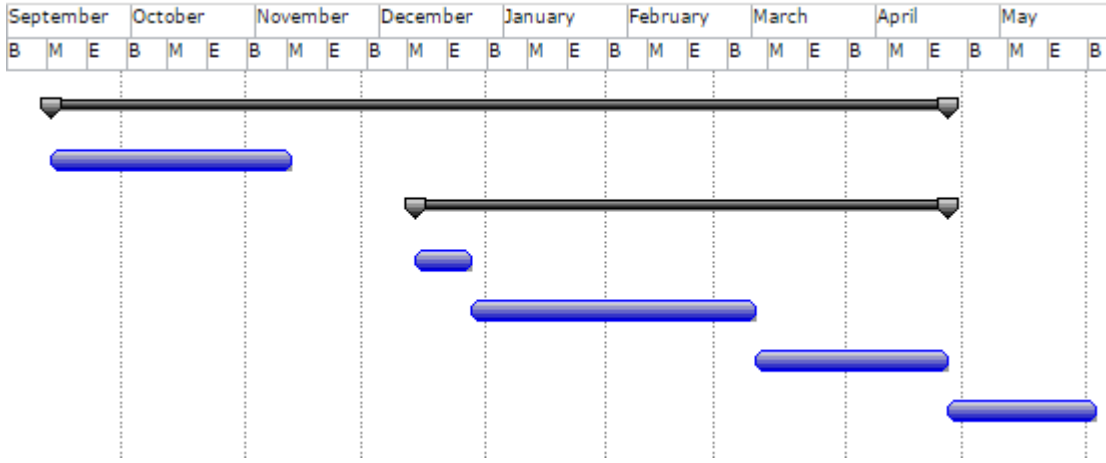
	i	Task Name	Duration	Work	Start	Finish
1		EnPara Fiyat ve Takip Uygulaması	141 days	0 hrs	10/10/2016	4/24/2017
2		Genel Araştırmanın Yapılması	45 days	0 hrs	10/10/2016	12/9/2016
3		Projenin Gerçeklenmesi: Araştırma ve Geliştirme	96 days	0 hrs	12/12/2016	4/24/2017
4		Uygulama İçin Gerekli Kurulumların Yapılması	11 days	0 hrs	12/12/2016	12/26/2016
5		Uygulamanın Gerçeklenmesi	52 days	0 hrs	12/26/2016	3/7/2017
6		Test	35 days	0 hrs	3/7/2017	4/24/2017
7		Dökümantasyon	28 days	0 hrs	4/24/2017	5/31/2017

Tablo-2.1: Kıstaslar ile zaman süreci



Şekil-2.1: Kıstaslar ile zaman süreci

- Gantt Chart diyagram yapısı Şekil-2.2'deki gibidir.



Şekil-2.2: Gantt Chart diyagramı

2.2.2 Kuramsal Araştırma

Projenin ilk adımı olarak, yaklaşık 1,5 ay proje ile ilgili araştırmaların yapılması ile geçti. Bu süreçte proje ile ilgili gerekli verinin nasıl sağlanacağı, veri akışının ne şekilde olacağı, kullanıcı arayüzünün ne şekilde olacağı hakkında analizler yapıldı. Veri akışı dosya formatı ile ilgili araştırmalar yapıldı. JSON (JavaScript Object Notation) dosya yapısı ile döviz ve altın bilgisinin çekilmesine karar verildi [8].

Projeyi gerçekleştirmek için hangi IDE, görsellik için hangi dizayn araçları dilleri ve uygulama arka tarafının gerçekleştirilmesi için hangi yazılım dillerinin kullanılacağı hakkında araştırmalar yapıldı. Başta native platformlar için mobil uygulama gerçekleştirilmek istenirse orata çıkan mevcut sorunlar yüzünden Adroid platformları için uygulama tam olarak gerçekleştirildi [2].

2.3 Projenin Kaynakları

Proje için;

- Bir bilgisayar gerekmektedir, bu kaynak öğrenci tarafından sağlanmıştır.
- Gerekli IDE ve kütüphaneler ücretsiz (özgür) kaynaklardan sağlanacaktır.

3. TEORİK BİLGİ

Satın almalar, yatırımlar ve büyük girişimler ile milyarlarca dolarlık bir sektöre dönüşen mobil uygulama pazarı yaşamımızı bambaşka bir boyuta taşıdı. IoT, cloud gibi bütün mega trendlerin alt yapısını oluşturan mobil teknolojilerdeki gelişim, finanstan otomative; perakendecilikten sağlığa pek çok sektörde dijital dönüşümü sağladı. Günümüzde artık şirketler, mobil uyumlu değil; mobil öncelikli olmak zorundadır. Rekabetçi piyasa koşullarında inovasyonun önemini bilen ve teknolojiye erken adapte olan firmalar, mobil konusunda gerekli yatırımlarını yaptılar ve yapmaya devam ediyorlar. Değişen ihtiyaçlara göre mobil uygulamalarını yeniliyorlar veya yeni uygulamalar geliştiriyorlar. Mobil uygulama konusunda yatırım yapmayı planlayan firmalar için dikkat edilmesi gereken en önemli konu, doğru mobil uygulama platformunu seçebilmek.

Mobil uygulamaları; native, cross platform ve hibrit olmak üzere 3 farklı başlıkta inceleyebiliriz. Her birinin bir birine göre avantajları ve dezavantajları vardır. Genel hatları ile inceleyecek olursak :

- Native: Her mobil uygulama için kendi platformunda özel geliştirme
- Hybrid: HTML5 ve JavaScript ile birlikte telefona özel uygulama
- Cross Platform I Xamarin : C# ile yazılan tek kod ile çoklu platformda native uygulama.

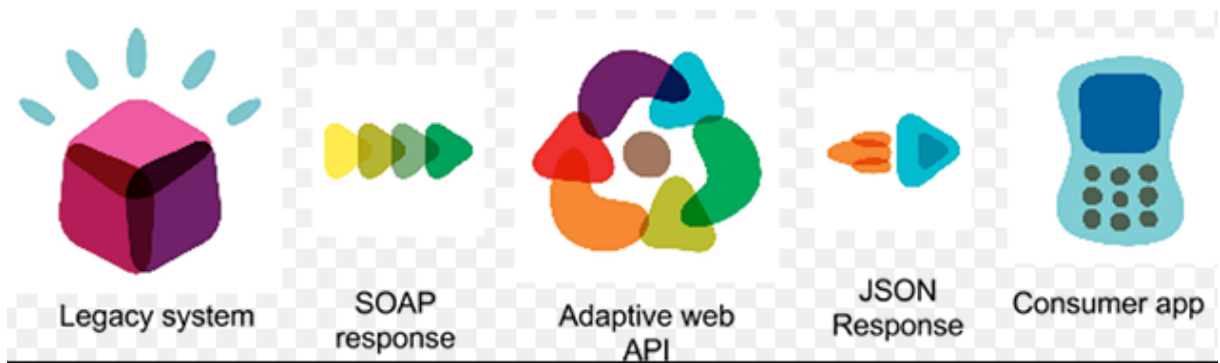
Bu 3 şekilde geliştirmenin kolaylıkları ve zorlukları yapmak istediğiniz işlere göre önem kazanıyor [3].

Enpara Fiyatlar ve Oranlar Takip Uygulaması temel olarak banka veri tabanından bir web servis (API) aracılığı ile döviz ve altın alım-satım bilgilerini Şekil-3.1'deki gibi uygulamaya aktaracaktır [6]. Daha sonra gelen bu veriler uygulama aracılığıyla kullanıcı ile paylaşılacaktır. Kullanıcı bu veriler için uyarı oluşturabilir, ve ürün istediği orana geldiğinde uyarılır. Web servisin oranları güncellemesi ile uygulama altındaki tüm alanlar yeni oranlar ile güncellenir ve kullanıcıya sunulur [10].

3.1 Uygulama ve Veri Tabanı Örnek Haberleşme Senaryosu

Uygulama, API ve veri tabanı arasındaki veri transferi şu şekilde olacaktır:

- İlk olarak uygulama API'den verileri güncellemesi için istekte bulunacak [9].
- Dahası sonra güncelleme isteği alan API verileri çekeceği veri tabanına bağlanıp verilerin güncel halini alacaktır [9].
- Verilerin güncel halini alan API, bu güncel verileri JSON dosya formatı ile uygulamaya aktaracaktır [7].
- Uygulama güncel verileri alıp kullanıcı ile paylaşacaktır.



Şekil-3.1: Mobil uygulama, web servis ve banka database haberleşmesi

Temel olarak uygulama web servisten oranların güncel halini ister, web servis uygulamaya bir JSON dosya ile cevap verir ve uygulama oranların güncel hallerini elde etmiş olur [6].

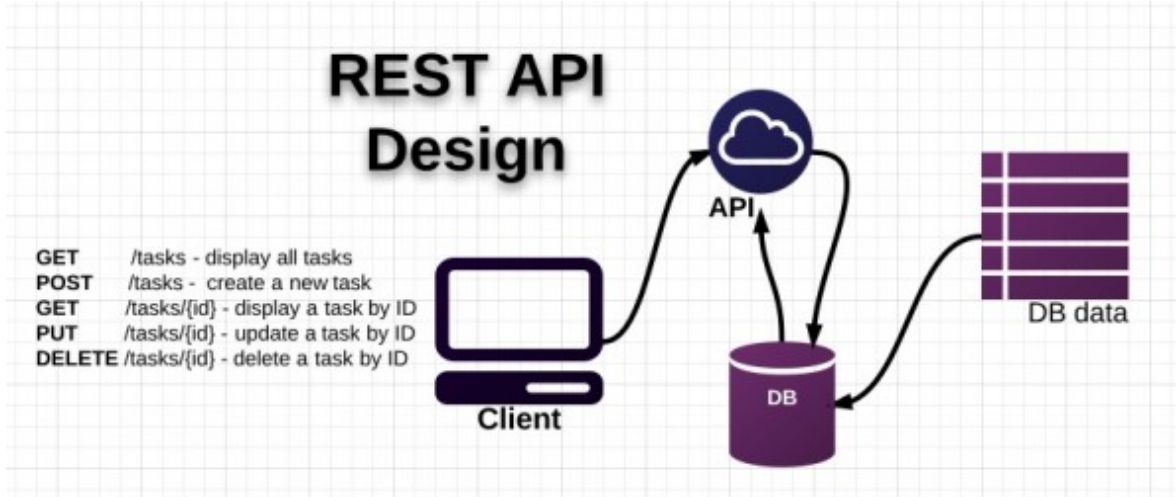
Şekil-3.2’de örnek JSON dosya formatı gösterilmiştir.

- “Ad” değişkeninin değeri “ilhan” stringidir.
- “Soyad” değişkeninin değeri “kaya” stringidir.
- “Bolum” değişkeninin değeri “bilgisayar mühendisliği” stringidir.
- “Sehir” değişkeninin değeri “istanbul ” stringidir.

```

1  {
2      "Ad": "ilhan",
3      "Soyad": "kaya",
4      "Bolum": "bilgisayar mühendisliği",
5      "Sehir": "istanbul"
6  }
```

Şekil-3.2: Örnek JSON dosya formatı



Şekil-3.3: REST API Yapısı

Şekil-3.3'teki akış şemasındaki elemanları şu şekilde açıklayabiliriz:

- **DB data:** Enpara veri tabanı, güncel döviz ve altın kur bilgisini barındırır.
- **DB:** Uygulama veri bilgisini tutar, Firebase altında mevcut yapı oluşturuldu.
- **API:** Uygulama ve veritabanı arasındaki akışı sağlıyor.
- **Client:** Uygulama kullanıcısının, veri takibi ve veri girişini temsil ediyor.

4. PROJENİN ANALİZİ VE MODELLENMESİ

4.1 Proje Analizi

Enpara Fiyatlar ve Oranlar Takip Uygulaması için analiz yapılırken uygulamanın hangi ortamda (IDE) gerçekleştirileceği, hangi yazılım dillerinin ve hangi kütüphanelerin (framework) kullanılacağına karar verilmiştir.

Uygulama dışardan gelecek veriler JSON dosya formatında gelecektir.

4.1.1 Native Platformlar İçin

Native (IOS, Android, Windows vs) platformlar için aşağıdaki IDE, yazılım dilleri ve kütüphaneler kullanılmıştır [4].

- Framework
 - [Ionic framework 3.3.0](#)
 - [AngularFire2 v4.0-rc.0](#)
 - [Firebase Javascript SDK 3.9.0](#)
 - [Angular 4](#)
 - Apache Cordova
- IDE
 - [Visual Studio Code](#)
 - [Ionic CLI](#)
 - [Nodejs 6.10.2 or higher\(Npm\)](#)
- Yazılım Dili
 - [Typescript](#)
- [Çalıştırma](#)

4.1.2 Android Platformlar İçin

Android platformlar için geliştirme yapılırken aşağıdaki IDE, yazılım dilleri ve kütüphaneler kullanıldı.

- Üçüncü Parti Kütüphaneler ve Kaynaklar
 - [Google Firebase](#)

- [OneSignal Push Notificationlar için](#)
- [OkHttp](#)
- [Material Dialogs](#)
- IDE
 - Android Studio 2.3.2
- Yazılım Dili
 - Java for Android

Gerekli program kurulumlarından ve kütüphane kurulumlarından sonra proje modellenmesine geçilmiştir.

4.2 Projenin Modellenmesi

Proje modellenirken gerekli tablo yapıları oluşturuldu, bu tablolara web servisin nasıl etki edeceği, bu etkinin kullanıcıya nasıl gösterileceği hakkında gerekli çalışmalar yapıldı.

Uygulamanın web servis ile yaptığı haberleşme sonucunda tablolardaki bilgiler güncellenecek, gerekli hesaplamalar yapıp bu bilgiler kullanıcı ara yüzüne aktarılacaktır.

4.2.1 JSON Yapısı

Uygulamaya veri akışı JSON formatı ile yapılacaktır, aşağıda uygulamayı besleyecek ve sürekli güncelleyecek JSON formatları mevcuttur.

- Dolar (USD) Kur Bilgisi için JSON yapısı Şekil-4.1’de görülmektedir.
 - “selling” değişkeninin değeri Dolar kuru için satış değerini getirmektedir.
 - “buying” değişkeninin değeri Dolar kuru için alış değerini getirmektedir.

```

1  {   "selling":3.5754,
2      "update_date":1496132277,
3      "currency":1,
4      "buying":3.5751,
5      "change_rate":-0.0260043116826,
6      "name":"amerikan-dolari",
7      "full_name":"Amerikan Dolar\u0131",
8      "code":"USD"
9  }
```

Şekil-4.1: Dolar kur bilgisi JSON formatı

- Euro (EUR) Kur Bilgisi için JSON yapısı Şekil-4.2’de görülmektedir.
 - “selling” değişkeninin değeri Euro kuru için satış değerini getirmektedir.
 - “buying” değişkeninin değeri Euro kuru için alış değerini getirmektedir

```

1 {  "selling":3.9858,
2    "update_date":1496132419,
3    "currency":2,
4    "buying":3.9847,
5    "change_rate":-0.212803244624,
6    "name":"euro",
7    "full_name":"Euro",
8    "code":"EUR"
9  }

```

Şekil-4.2: Euro kur bilgisi JSON formatı

- Altın (GOLD) Bilgisi için JSON yapısı Şekil-4.3’te görülmektedir [8].
 - “selling” değişkeninin değeri Altın için satış değerini getirmektedir.
 - “buying” değişkeninin değeri Altın için alış değerini getirmektedir.

```

1 {  "selling":145.518174997,
2    "update_date":1496132481,
3    "gold":2,
4    "source":99,
5    "buying":145.453349447,
6    "change_rate":-0.121558474362,
7    "name":"gram-altın",
8    "full_name":"Gram Alt\u0131n",
9    "short_name":"Gram",
10   "source_name":"serbest-piyasa",
11   "source_full_name":"Serbest Piyasa"
12 }

```

Şekil-4.3: Altın bilgisi JSON formatı

4.2.2 Tablo Yapısı

Uygulamanın veri tabanı döviz ve altın için alım ve satım bilgisini tutacak, kullanıcının oluşturduğu alarmları içerecek, kullanıcı bilgisini içerecek (push notificationlar için) şekilde Şekil-4.4’teki gibi tasarlanmıştır.

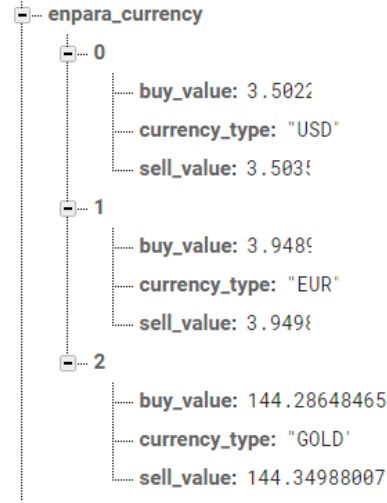
```

enpara-mobile
├── enpara_alert
├── enpara_currency
├── enpara_currency1
└── enpara_users

```

Şekil-4.4: Uygulama tablo yapısı

- Döviz Kuru Tablosu (Enpara Currency) Şekil-4.5'teki gibidir.



id	buy_value	currency_type	sell_value
0	3.5022	"USD"	3.5031
1	3.9485	"EUR"	3.9491
2	144.28648465	"GOLD"	144.34988007

Şekil-4.5: Döviz Kuru Tablosu

Döviz kuru tablosunda döviz tipi bilgisi (currency_type), alış değeri bilgisi (buy_value) ve satış değeri bilgisi (sell_value) tutulmaktadır.

- Alarm Tablosu (Enpara Alert) Şekil-4.6'deki gibidir.



id	alert_type	currency_type	timestamp	value
-KlyJGqfiKA7ESUTy-ab	"SELL"	"USD"	149676887461	"3.505455"
-KlyMlCXz8WZx9KE78CT	"BUY"	"USD"	149676877926	3.5022
-KlyMp4Y2FQouTuBPyaK				

Şekil-4.6: Alarm Tablosu

Enpara alarm tablosu içinde alarmin hangi döviz kuru için tutulduğunu belirlemek için döviz tipi (currency_type), alım işlemi veya satım işlemi olduğunu belirlemek için alarm tipi (alert_type), alarmin değerini tutmak değeri (value) ve alarmin ne zaman oluşturulduğunu tutmak için tarih bilgisi (timestamp) bilgisi bulunmaktadır.

- Kullanıcı Tablosu (Enpara Users) Şekil-4.7’deki gibidir.

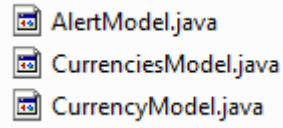


Şekil-4.7: Kullanıcı Tablosu

Enpara kullanıcı tablosunda her kullanıcıya tekil bir id değeri verilmektedir (`userId`). Her kullanıcının alacağı push notifications için kullanıcılara tekil token değeri verilmektedir (`pushToken`).

4.2.3 Model Yapısı

Uygulamada aşağıdaki gibi üç çeşit model bulunmaktadır. Bu model yapıları Şekil-4.8’deki gibidir.



Şekil-4.8: Model yapısı

- Döviz Kuru Modeli Şekil-4.9'daki gibidir.

```

/**
 * Created by ilhan.kaya on 6/3/2017.
 */

@IgnoreExtraProperties
public class CurrencyModel {
    private Double buy = 0.0;
    private Double sell = 0.0;

    public CurrencyModel() {
    }

    public CurrencyModel(Double buy, Double sell) {
        this.buy = buy;
        this.sell = sell;
    }

    public Double getBuy() {
        return buy;
    }

    public Double getSell() {
        return sell;
    }

    public void setBuy(Double buy) {
        this.buy = buy;
    }

    public void setSell(Double sell) {
        this.sell = sell;
    }

    @Exclude
    public Map<String, Object> toMap() {
        HashMap<String, Object> result = new HashMap<>();
        result.put("buy", buy);
        result.put("sell", sell);
        return result;
    }
}

```

Şekil-4.9: Döviz kuru modeli

Döviz kuru modeli ile kur bilgileri üzerinde işlemler yapılacaktır. Döviz kur modeli maplanerek model içeriğine daha rahat erişilmektedir.

- Döviz Kurları Modeli Şekil-4.10'daki gibidir.

```

/**
 * Created by ilhan.kaya on 6/3/2017.
 */

@IgnoreExtraProperties
public class CurrenciesModel {
    private CurrencyModel usd;
    private CurrencyModel eur;
    private CurrencyModel gold;

    public CurrenciesModel() {
    }

    public CurrenciesModel(CurrencyModel usd, CurrencyModel eur, CurrencyModel gold) {
        this.usd = usd;
        this.eur = eur;
        this.gold = gold;
    }

    public CurrencyModel getUsd() {
        return usd;
    }

    public CurrencyModel getEur() {
        return eur;
    }

    public CurrencyModel getGold() {
        return gold;
    }
}

```

Şekil-4.10: Döviz kurları modeli

Döviz kurları modeli, döviz kuru modelinden beslenmektedir ve kur için bir kur modeli eklenmesi işe kurlar modeli ortaya çıkmış olmaktadır. Bu aşamada Nesne Tabanlı Programlama yapısı kullanıldı.

- Alarm Modeli Map Yapısı Şekil-4.11'deki gibidir.

```

public Map<String, Object> toMap() {
    HashMap<String, Object> result = new HashMap<>();
    result.put("currency", currency);
    result.put("alertType", alertType);
    result.put("value", value);
    result.put("time", time);
    return result;
}

```

Şekil-4.11: Alarm modeli mapleme

Alarm modeli içerisinde kur bilgisi, alarm tipi, alarm değeri ve alarm oluşturma tarih bilgisi barındırılmaktadır. Alarm modeli maplenerek, bilgilerine rahatça erişilmektedir.

5. UYGULAMA VE TEST

5.1 Uygulama

Proje uygulaması ilk etapta Native (çoklu) platformlar için yapılmaya başlandı, fakat çoklu ortam uygulamalarında karşılaşılan push notifications (bildirimler) sorunları sebebi ile uygulama Android platformları için gerçekleştirildi [3]. Uygulama temel olarak iki ana ekrandan oluşmaktadır, bunun yanında alarm girişi ve güncellenmesi içinde iki girdi ekranı mevcuttur [2].

Uygulama gerçekleştirirken Nesne Tabanlı Programlama'dan (OOP) faydalandı, kur ve kurlar sınıf yapıları türetildi.

Kullanıcı alarm girişi yaptıktan sonra her saniyede bir döviz kur bilgileri güncellenmekte ve kullanıcının alarmı kontrol edilmektedir. Kullanıcının kurduğu alarmlar şu durumlarda gerçekleşebilir:

- Alım değeri \leq Alarm alım değeri
- Satım değeri \geq Alarm satım değeri

Yukarıdaki durumlar için değerine ulaşan veya değerini aşan alarmlar için kullanıcı uygulama tarafından uyarılır. Uyarılan kullanıcı kurduğu alarm değerine ulaşıldığını veya alarm değerinin aşıldığını görür. Bu aşamadan sonra kullanıcı alarmı daha yüksek bir değere çekerek güncelleyebilir veya silebilir.

5.1.1 Ana Sayfa (Home)

Uygulama açılınca ilk olarak kullanıcının karşısına çıkan sayfadır. Bu sayfada döviz kurları bilgileri bulunmaktadır ve sürekli bu kur bilgisi güncellenmektedir. Ana Sayfa Şekil-5.1'deki gibidir.

EnPara Fiyatlar ve Oranlar		
Home		Alert
CURRENCY	BUY	SELL
USD	3.5022 TL	3.5035 TL
EUR	3.9489 TL	3.9498 TL
GOLD	144.286484654 TL	144.349880075 TL

Şekil-5.1: Uygulama ana sayfası

5.1.2 Alarm Sayfası (Alert)

Alarm oluşturulan, silinen veya güncellenen sayfadır, bu sayfada alım ve satım alarmları oluşturulup, alarm takipleri yapılmaktadır. Alarm sayfası Şekil-5.2'deki gibidir.

EnPara Fiyatlar ve Oranlar

Home

Alert

+

CURRENCY	ALERT TYPE	VALUE	TIME
USD	SELL	3.505455 TL	06.06.2017 20:07
USD	BUY	3.5022 TL	06.06.2017 20:06
EUR	BUY	3.9489 TL	06.06.2017 20:06

Şekil-5.2: Alarm Sayfası

5.1.3 Alarm Ekleme ve Güncelleme (Add Alert, Update Alert)

Alarm ekleme ekranı, kullanıcının alarm girişi yaptığı yani alarm oluşturduğu ekrandır. Bu ekranda kullanıcıdan alarm kurulacak döviz kuru bilgisi, alarm tipi (alım yada satım) ve alarm değeri girişi yapması beklenmektedir. Alarm Ekleme sayfası Şekil-5.3'teki gibidir.

Add Alert	
Currency Type	USD ▾
Alert Type	BUY ▾
Value	3.5022
<div>SAVE</div>	

Şekil-5.3: Alarm Ekleme Sayfası (Add Alert)

Alarm güncelleme ekranı kullanıcının mevcut alarmları değiştirmesine olanak sağlamaktadır.

5.2 Test

Uygulama için test senaryoları şu şekildedir:

- Data güncellenmesi: Döviz kurlarının mevcut değerleri kaydedilir, bir-iki saniye içinde güncellenen değerler ile kıyaslanır.
- Alarm oluşturma: Add Alert sayfasından döviz kur bilgisi, alarm bilgisi, alarm değeri girişi yapılır ve Save butonuna basılır. Girişi yapılan alarmın oluştuğu görülür.
- Alarm güncelleme: Daha önce oluşturulan bir alarmın değeri, Update Alert sayfası ile güncellenecek alarmın seçilmesi sonucu alarm değerinin değiştirilmesi ile güncellenir.
- Alarm silme: Daha önce oluşturulmuş bir alarm değeri, Delet alert butonu ile silinir.
- Alarm değerine ulaşılması: Kurulu alarmı olan bir döviz kuru fiyat değişimleri sonucu, alarm değerine ulaşırsa kullanıcı uyarılır.

6. DENEYSEL SONUÇLAR

Alarm eklme sayfası aracılığı ile yeni alarm girişi yapılır, ve bu yeni alarm Alarm sayfasından takip edilebilir, güncellenip değeri değiştirilebilir veya silinebilir. Şekil-6.1’deki gibi yeni alarm girişi yapılabilir.

EnPara Fiyatlar ve Oranlar

HOME ALERT

CURRENCY	ALERT TYPE	VALUE	TIME
USD	BUY	3.5418	

Long press to edit or delete a alert

Şekil-6.1: Yeni alarm girişi

Alarm girişi yapıldıktan sonra, girişi yapılan alarm bilgisi Şekil-6.2’deki gibi olur.

EnPara Fiyatlar ve Oranlar

HOME ALERT

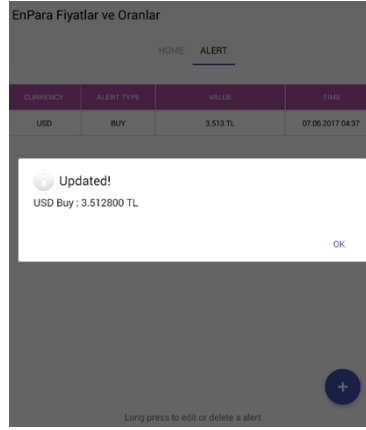
CURRENCY	ALERT TYPE	VALUE	TIME
USD	BUY	3.5418 TL	07.06.2017 05:59

Long press to edit or delete a alert

Şekil-6.2: Alarm görüntüsü

Uygulama kullnıcısının oluşturduğu alarmlar için döviz kuru değerinin değişmesi sonucu, kur değeri alarm değerine eşit olursa veya alarm değerini aşarsa alarm verilir ve kullanıcı uyarılır.

Şekil-6.1’de görüldüğü üzere değerine ulaşan bir alarm için kullanıcı alarm mesajı ile uyarılmaktadır. Bunun sonucunda kullanıcı oluşturduğu alarmin değerine ulaştığını görür, ve bu doğrultuda işlemler yapar.



Şekil-6.3: Değerine ulaşan alarmin kullanıcıya bildirilmesi

7. SONUÇ VE ÖNERİLER

Oldukça popüler mobil geliştirme araçları olan Visual Studio ve Android Studio platformlarına karşı olan ilgi gün geçtikçe katlanarak artmaktadır. Bu platformlar, bu alanda en kapsamlı ve detaylı içeriğe sahip platformlardır. Gerçeklenen projede Ionic framework, Angular, Nodejs gibi kütüphanelerden faydalanılmıştır [4]. Bu kütüphaneler mevcut platformlar ile kullanılarak daha verimli bir süreç gerçekleştirildi. Google Firebase ile relatime database yapısından faydalanıldı. Server-Client bağlantı yapısı, veri akışı hakkında araştırmalar yapılarak projeye eklendi. Ve nesne tabanlı programlamadan faydalanılarak proje modellendi [5].

İlk olarak Native platformlar için çalışma gerçekleştirilmeye çalışılsa da, uygulama için gerekli uyarı sisteminin native platform araçları ile gerçekleştirilemediği, ve düzgün bir yapının ortaya çıkmadığı gözlemlendi. Daha sonra uygulama Android platformlar için gerçekleştirildi ve daha düzgün bir yapı gözlemlendi [3].

Bu raporda mobil uygulama geliştirme sürecindeki adımlar, Native platform ve Android platformlar için uygulama geliştirme sürecine dikkat çekilerek, bu platformların sahip olduğu diğer araçlar tanıtılmaya çalışılmıştır [5]. Proje QNB Finansbank Enpara ekibi tarafından prototip olarak kullanılacak, projeye tam anlamıyla düzgün bir push notification yapısı eklenecektir. Proje daha da geliştirilerek; sesli uyarı, watch özelliği gibi yeni özellikler eklenebilecek şekilde geliştirilmiştir. Dolayısı ile gerçekleştirilen bu çalışma, gerekli çaba sarf edilerek geliştirilmeye devam edilirse, mobil uygulama geliştirme için gerekli olan temel bilgi ve beceriler kazanılmış olacaktır [3].

8. KAYNAKLAR

- [1] QNB Finansbank, “Enpara.com nedir?”, www.qnbfinansbank.enpara.com, (Eriřim: 30 Ekim 2016)
- [2] Kırca, T., “Native-Html5-Hybrid Mobil Uygulama Geliřtirme Diline Karar Vermek”, Şubat 2015, www.mobilhanem.com, (Eriřim: 06 Mart 2017)
- [3] Smyth, N., Android Studio 2.3 Development Essentials, Android 7 Edition Paperback, March 17, 2017
- [4] Ionic Freamwork, ionicframework.com, (Eriřim: 04 Mart 2017)
- [5] Mobilist Biliřim Sistemleri, "Mobile Uygulama", Ocak 2017, www.mobilist.com.tr/tr/mobil-uygulama.html, (Eriřim: 20 Mayıs 2017)
- [6] Iřık, O., "REST API Nedir? Nasıl Kullanılır?", Ocak 2015, www.obenplus.com/blog/rest-api-nedir-nasil-kullanilir/, (Eriřim: 03 Mart 2017)
- [7] Köse, Ü., "JSON nedir ? Ne amaçla Kullanılır?", 19 Mart 2015, umiitkose.com/2015/03/json-nedir-ne-amacla-kullanilir/, (Eriřim: 01 Kasım 2017)
- [8] Lengstorf, J., "JSON: What It Is, How It Works, & How to Use It", www.copterlabs.com/json-what-it-is-how-it-works-how-to-use-it/, (Eriřim: 05 Aralık 2017)
- [9] Wasson, M., Pasic, A., "Calling a Web API From a .NET Client (C#)", 20 Ocak 2014, docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client, (Eriřim: 10 Aralık 2016)
- [10] Jscrambler, "Create your own client", jscrambler.com/en/help/webapi/create_your_own_client, (Eriřim: 15 Ocak 2017)