

**VIII Inetr'02**

**Türkiye İnternet Konferansı  
19-21 Aralık 2002  
Harbiye Askeri Müze ve Kültür Sitesi  
İstanbul**

**Java ve SOAP kullanılarak Mobil Cihazlardan Hisse Senedi Alımı  
Uygulaması**

Özgür Toprak, Seyhun Altunbay  
Bilgisayar Mühendisliği Bölümü, Işık Üniversitesi  
altunbay@isikun.edu.tr, toprakoz@yahoo.com

# Java ve SOAP kullanılarak Mobil Cihazlardan Hisse Senedi Alımı Uygulaması

Özgür Toprak, Seyhun Altunbay  
Bilgisayar Mühendisliği Bölümü, Işık Üniversitesi  
altunbay@isikun.edu.tr, toprakoz@yahoo.com

## Özet

*Günümüzde mobil tabanlı teknolojileri kullanan çözümler hızla yaygınlaşmaktadır. Bu bağlamda mobil teknolojilerin diğer teknolojilerle bütünleşmesini sağlamak büyük bir önem kazanmıştır. Müşterilerine mobil çözümler sunmak isteyen şirketlerin bu ihtiyacı, WAP ve Java destekli telefonların sayısının artması ve GPRS teknolojisinin cep telefonu operatörleri tarafından desteklenmesiyle birlikte gün geçtikçe bir zorunluluk haline gelmektedir.*

*Burada sunulan uygulama, bir borsa aracı kurumunun müşterilerine sunduğu "mobil hizmet" için gerek kullanıcı arayüzü gerekse genel sistem mimarisi açısından bir örnek teşkil edebilir. Java tabanlı telefonlar ve Palm avuç içi bilgisayarlar için geliştirilen bu uygulama Web servislerinden de faydalanan esnek ve genişletilebilir bir yapıya sahiptir.*

## 1. Giriş

Java 2 platformunun elektronik tüketici ürünlerine (Akıllı kartlar, cep telefonları, avuç içi bilgisayarlar, çağrı cihazları) yönelik sürümü J2ME (Java 2 Micro Edition), şirketlerin sunabilecekleri çözümleri çeşitlendirmiştir.

Şirketlerin sistemlerini mobil çözümleri sunmak için gereken teknolojilerle bütünleştirebilmeleri gerekir. Bunu kısa zamanda başarabilmek için gereken, halihazırdaki sistemlerin ölçeklenebilir yapıya sahip olmalarıdır. Eğer bu sistem çok katmanlı J2EE (Java 2 Enterprise Edition) uygulamalarıyla oluşturulmuşsa, ölçeklenebilir bir sistem yapılandırılmış demektir.

Varsayılan, şirketin J2EE uygulamaları altyapısına sahip olmasıdır.

Bu uygulamadaki amaç:

1. Halihazırdaki J2EE uygulamalarındaki iş mantığını (Business Logic) kullanarak ve sisteme kalın (akıllı) istemci kısmını ekleyerek Java tabanlı mobil uygulamalara destek veren bir yapı oluşturabilmek,
2. Web Servisleri kullanılarak mobil uygulamadaki esneklik ve hizmet çeşitliliğini arttırmak.

3. Uygulamanın ihtiyaç duyduğu yüksek güvenli yapıyı sağlamak.
4. Bir mobil uygulamanın gerektirdiği kullanıcı dostu arayüz özelliğini sağlamak.

## 1.1 Kısaca Web Servisleri

Web servisi uzaktan çağırılıp çalıştırılabilen ve belli görevleri yerine getiren bir modüldür. İstemci, sunucudan RPC (Remote Procedure Calling) kullanarak belli web servislerini çağırabilir. İstemci bu web servislerinden gelen verilerle uygulamanın gerektirdiği bazı işlemleri yerine getirmiş olur [1]. Örneğin, uygulama o anda İstanbul'un hava sıcaklığına ihtiyaç duyarsa, bunu bir web servisi kullanarak başarabilir. Aynı uygulama belli bir noktadaki araç trafik yoğunluğunu başka bir web servisinden elde edebilir.

Bugün web servisleri geliştirme sürecinde üç standart bulunmaktadır: SOAP, WSDL, UDDI.

### a. Simple Object Access Protocol (SOAP)

Web servisleri, taşıma katmanı olarak **SOAP** (Simple Object Access Protocol) ve HTTP kullanırlar. SOAP mesajları XML belgeleridir ve bu belgeler web servisiyle, bu web servisi çağırılan uygulama arasında gidip gelir.

Gerek web servislerinin gerekse bu web servisleri kullanan uygulamaların her dilde yazılabilmeleri ve her platformda çalışabilmeleri bu teknolojinin en önemli özelliklerinden biridir.

Aşağıda örnek bir SOAP iletisi bulunmaktadır. Her SOAP iletisi bir kök "Envelope" elemanına sahiptir. Envelope elemanın içinde Header ve Body olmak üzere iki parça vardır. Header elemanı yönlendirme (routing) veya kaynak (context) bilgilerini içerir. Body kısmıysa iletinin kendisini içerir. Gerek Header, gerekse Body kısımları boş olabilir.

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "urn:stock-quote-services"
```

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>BORL</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Şekil 1 – Örnek bir SOAP mesajı (Kaynak: BORLAND)

## **b. Web Services Description Language (WSDL)**

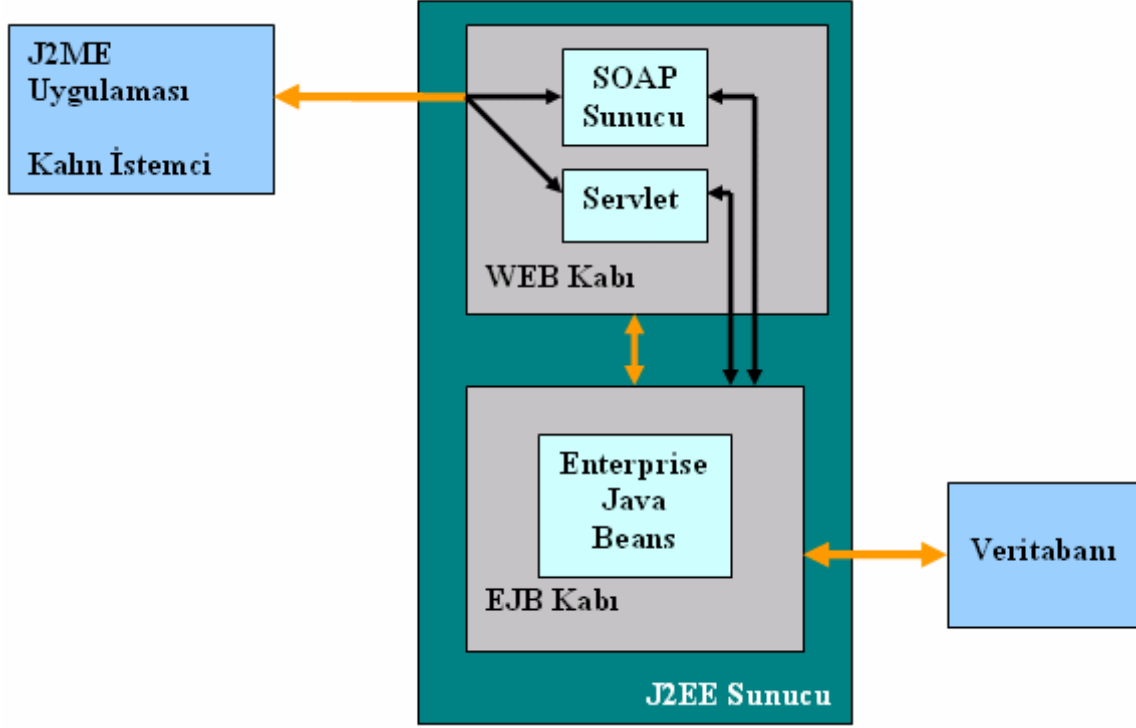
Bir web servisin hangi işlemleri yaptığını ve nasıl çağrılacağını belirten XML tabanlı bir dildir. Bir istemci, kullanmak istediği web servisini WSDL belgesi sayesinde tanır, nasıl kullanılacağını bu belge aracılığıyla anlayabilir. WSDL belgesi, hangi metodların hangi parametrelerle çağrılabileceğini içerir.

## **c. Universal Description, Discovery and Integration (UDDI)**

UDDI; web servislerini tanımlama, yayınlama ve bulma işlemleri konusunda bir standarttır. WSDL belgelerine sahip web servisleri UDDI sistemine kayıt olurlarsa, bu web servisleri istemciler tarafından kolayca bulunabilir. [1]

## 2. J2EE – J2ME Bütünleşmesi

Şekil 2’de uygulamanın genel yapısı görülmektedir.

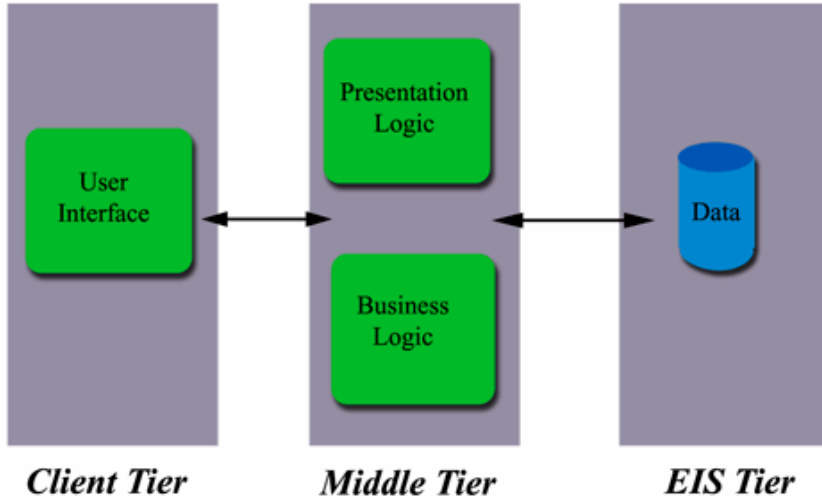


Şekil 2 – Uygulamanın genel yapısı.

### 2.1 J2EE (Java 2 Enterprise Edition)

J2EE uygulamaları çok katmanlı ve dağıtık mimariye sahiptirler. Burada modüler yapı ön plana çıkmaktadır. Sistemin karmaşıklığının, ölçeklenebilirliğinin ve diğer sistemlerle bütünleşmesinin kolaylaştırılması açısından çok katmanlı mimari büyük avantaj sağlamaktadır. Bugün J2EE, çok katmanlı dağıtık mimarinin uygulamalarının en önemlilerinden biridir.

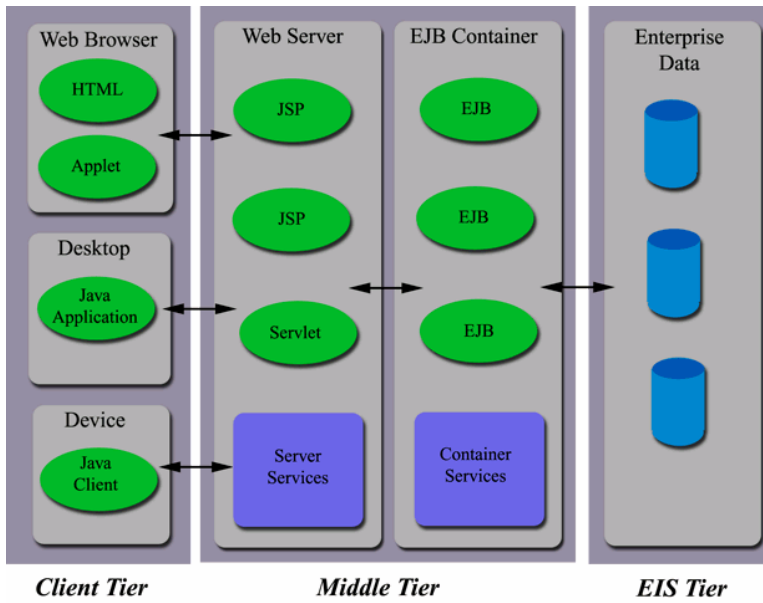
J2EE uygulamalarında (Şekil 3), “iş mantığı” (Business Logic) istemci aracılığıyla sunulan arayüze yönelik öğeler içermemelidir. “Sunum mantığı” (Presentation Logic) kullanıcı arayüzüyle iş mantığı arasında yer alan bir katmandır [2]. Böylece farklı kullanıcı arayüzleri için gereken destek farklı sunum mantıkları geliştirerek sağlanabilir. Örnek olarak, WAP destekli mobil cihazlara yönelik ince istemci yapısı sağlanarak WML (Wireless Markup Language) sunum mantığı geliştirilebilir.



Şekil 3 – Çok katmanlı model (Kaynak: Borland).

**EJB (Enterprise Java Beans)** bir sunucu bileşen modelidir. Uygulama sunucularında çalışır. Bileşen model, bütün sunucu taraflı nesnelerin uyması gereken, uygulama sunucuları ve uygulama geliştiricilerin üzerinde anlaştıkları kurallar üzerine oturtulmuş bir standarttır. EJB; dağıtık (RMI aracılığıyla), ölçeklenebilir ve güvenli Java uygulamalarının geliştirilmesini sağlar. [3]

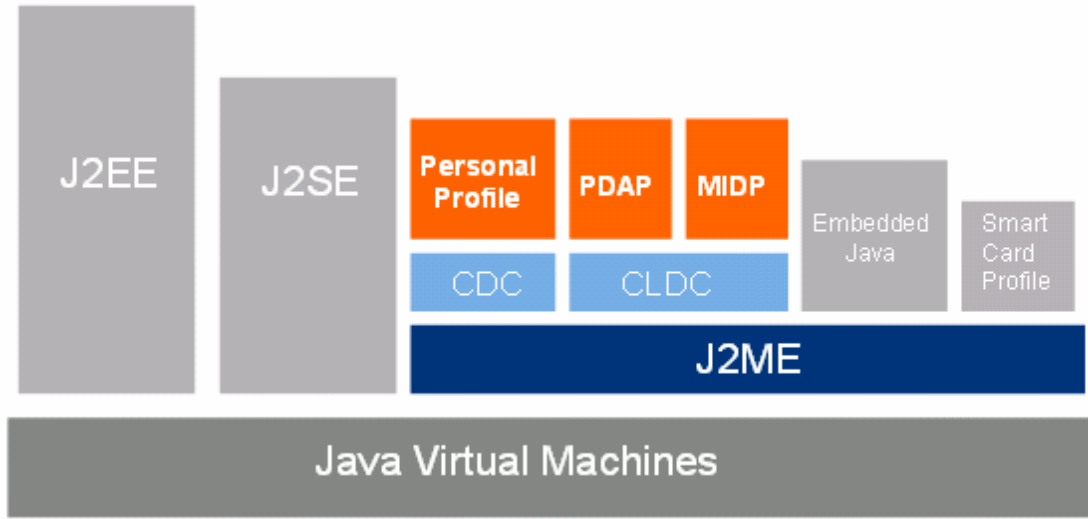
Şekil 4’te örnek bir J2EE yapısı görülüyor.



Şekil 4 – Örnek J2EE uygulaması (Kaynak: Borland).

## 2.2 J2ME (Java 2 Micro Edition)

Ağa bağılı J2ME uygulamaları mobil cihazlar üzerinde özerk olarak çalışırlar. Gerektiğinde sunucuya bağlanabilir ve gerekli verileri sunucudan getirerek kullanıcıya sunar. J2ME uygulamasının özerk yapısı, Şekil 3'te bulunan sunucu tarafındaki sunum mantığının tamamıyla istemci üzerinde olmasını gerektirir; sunucudan (Servlet'ler aracılığıyla) sunum mantığının kullanacağı veriler sağlanır.



Şekil 5 – Java platformu [4]

Bu uygulamada Şekil 5'te görülen J2ME profillerinden MIDP (Mobile Information Device Profile) kullanılmıştır. Bu profil özellikle Java tabanlı cep telefonları ve avuçiçi bilgisayarlar için kullanılmak üzere geliştirilmiştir.

Bu cihazların sahip olması gereken en düşük özellikler:

- 96x54 piksel ekran
- 1-bit renk derinliği
- Klavye veya dokunmatik ekran
- 128 KB uçucu olmayan (non-volatile) bellek
- 8 KB uçucu olmayan, uygulama tarafından yaratılabilen kalıcı veri
- 32 KB uçucu hafıza (Java runtime environment için)
- 2 yönlü ağ bağlantısı
- Düşük bant genişliği [5]

### 3. Uygulamanın Özeti

Bir borsa aracı kurumu müşterilerine mobil hizmet sunmaktadır. Müşteriler varolan hesaplarını Java tabanlı cep telefonları veya avuç içi bilgisayarları aracılığıyla yönetebilmektedir.

Uygulamanın özellikleri:

- a. Hisse senedi işlemleri
- b. Yatırım Fonu işlemleri
- c. Portfoylar bilgileri
- d. Hisse senedi grafik çizme
- e. Piyasayla ilgili anlık veriler ve haberler
- f. Kişiselleştirilebilir özellikler
- g. Hisse senedi bandı

Uygulamanın iki ana kısmı bulunmaktadır:

- a. Tüm Java tabanlı mobil cihaz sahibi olanlara açık kısım
- b. Müşterilerin işlem yapabilmelerine olanak sağlayan kısım

İlk kısımda, kullanıcı piyasa hakkında haber ve rakamsal verilere ulaşabilir. Bu kısımda güvenli veri transferi sağlanmasına gerek yoktur. Verilerin iletimi web servisleri kullanılarak sağlanır.

İkinci kısım sadece şirketin müşterilerine yöneliktir. Müşteriler yatırımlarına her an ve her yerden yön verebilme imkanına sahiptirler. Hisse senedi ve yatırım fonu gibi menkuller alınıp satılabilir. Burada güvenli bir bağlantı sağlanmıştır.

#### 3.1 Güvenlik

Güvenlik konusunda alınan önlemler şöyle özetlenebilir:

- a. *SSL (Secure Socket Layer)*: Sunucu, sertifikasını istemciye gönderir. İstemci sahip olduğu kök sertifikalarıyla sunucudan aldığı sertifikayı karşılaştırır. İstemcinin sertifikayı doğrulaması için, istemcinin sunucunun sertifikasını gösteren anahtara karşılık gelen kök sertifikasına sahip olması gerekir.
- b. *Oturum (Session)*: Servlet istemciden gelen şube kodu, müşteri kodu ve şifreyi aldıktan sonra EJB içerisinde ürettiği bir eşsiz (unique) numarayı oturum tabanlı çerez (Session based cookie) aracılığıyla istemciye gönderir. İstemci çerezi başlık alanlarından (Header Field) alır ve her bağlantı sırasında bu özel numarayı sunucuya göndermek üzere saklar. Sunucu, istemcinin her isteğinde “şube kodu” ve “hesap no” ikilisinin “özel numara” ile örtüşüp örtüşmediğini kontrol eder.

c. Üç aşamalı müşteri doğrulama: Hesap No + Şifre + [Kişisel Sorular] (Şekil 6,7).



Şekil 6 – Müşteri şube no, Hesap no ve Şifre girişi.

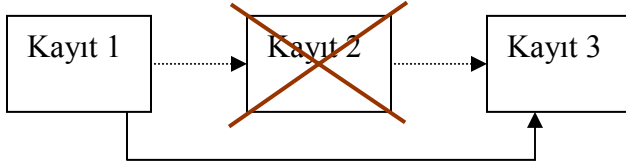


Şekil 7 – Özel soru.

### 3.2 RMS (Kayıt Yönetim Sistemi - Record Management System)

MIDP, MIDlet'lerin cihazlarda kalıcı olarak veri depolamasını ve çağırabilmesini sağlar. Bu sistem, basit bir kayıt yönelimli veritabanıyla sağlanır. Bir MIDP veritabanınındaki veri MIDlet'ten çıkış yapıldığında da varlığını sürdürür. Uygulamaya RMS desteğini sağlamak için *javax.microedition.rms* paketini kullanmak gerekmektedir.

RMS'de kayıtlar birbirine bağlı olarak saklanırlar. Bir kayıt eklendiğinde, alacağı kayıt numarası bir önceki kaydın bir fazlasıdır. 1, 2 ve 3 numaralı kayıtlar mevcutsa ve 2 numaralı kayıt silinirse bir daha 2 numaralı kayıt kullanılamaz ve artık 1 numaralı kayıttan sonra 3 numaralı kayıt gelir (Şekil 8).



Şekil 8 – RMS'de kayıt silme işlemi.

Bu uygulamada RMS,

1. Sisteme giriş yaparken müşterinin “özel soru” nun sorulup sorulmamasına ait tercihini hatırlamak,
2. Mobil şubeye giriş yapılırken müşterinin şube kodunu ve hesap numarasını saklayabilmek,
3. Hisse bandını oluşturan hisselerin içeriğini uygulama kapandıktan sonra da saklayabilmek,

amacıyla kullanılmıştır.

### 3.3 kSOAP ile Web Servislerine Ulaşmak

MIDP, SOAP için doğal bir destek vermez. Fakat açık kaynak kodlu bir proje olan kSOAP (<http://www.ksoap.org>) bu ihtiyaca cevap vermektedir. kSOAP 1.0 JAR dosyası yaklaşık 40 KB büyüklüğündedir.

Sunucu tarafında Apache SOAP 2 kullanılmaktadır. Servlet'lere SOAP desteği kazandırarak istemcinin SOAP sunucusundaki bir uzaktan yordam çağrısını (Remote Procedure Calling) çalıştırması sağlanmaktadır. (Şekil 9)

```
HttpTransport transport =  
new HttpTransport("http://test:8082/Mobil/servlet/Hisse", "urn:Hisse# hisseAdiGetir ");  
SoapObject istem = new SoapObject("urn:Hisse ", "hisseAdiGetir");  
istem.addProperty("sube", "78");  
istem.addProperty("hesapNo", "3453259800");  
istem.addProperty("hisseKodu", "21");  
String gelen = transport.call(istem).toString();
```

Şekil 9 - kSOAP kullanarak uzaktan yordam çağırılması.

SOAP çağrılarını HTTP üzerinden J2ME bağlantı yapısını kullanarak gerçekleştirmek için *HttpTransport* sınıfı kullanılmaktadır.

*SoapObject* sınıfı, bir SOAP nesnesinin SOAP istemi veya yanıtını belirtmek için kullanılmaktadır.

```
public class Hisse {  
    public String hisseAdiGetir (String sube, String hesapNo, String hisseKodu) {  
        return hisseAdi(sube, hesapNo, hisseKodu);  
    }  
    ...  
    ..  
}
```

Şekil 10 – Servlet tarafındaki kaynak kodu

### 3.4 Uygulamanın Boyutu

Uygulamanın boyutunu standartlar ölçüsünde tutabilmek amacıyla çeşitli yöntemler kullanılmıştır:

1. Uygulamanın boyutunun Nesne Yönelimli Programlama'nın getirdiği yararlardan öncelikli hale geldiği durumlarda sınıfların (class) sayısı en aza indirgenmelidir.

2. Dinleyiciler (CommandListener, ItemStateListener) tek bir sınıfta tanımlanmıştır. Bu sınıf, uygulamada mutlaka olması gereken MIDlet sınıfında bulunmaktadır. Böylece uygulamada fazla sayıda dinleyicinin getireceği kod miktarı artışından kaçınılmıştır.

3. Paket, sınıf, veri ve metot gibi elemanların isimlerini kısaltmak amacıyla, aynı zamanda da sınıfların kaynak koda dönüştürülerek okunmasını zorlaştırmak için bir “obfuscator” programı olan RetroGuard kullanılmıştır.

### 3.5 Yazılım Geliştirme Araçları

*Java IDE*: JBuilder 6 Enterprise Edition.

*J2ME Emülatörü*: JBuilder 6 ile bütünleşik çalışan Nokia 9310i, Motorola i85, PalmOS.

*Uygulama Sunucusu (Application Server)*: Borland Enterprise Server 5.

*Servlet Container*: Tomcat 3.3.

*SOAP Sunucu*: Apache SOAP 2.0

## 4. Sonuç

Bu uygulamayla, mobil cihazlarda çalışan uygulamaların ihtiyaç duyduğu

- Güvenli ortam
- Kişiselleştirilebilme
- Esneklik ve genişletilebilirlik
- Arayüzün kullanım kolaylığı

özellikleri karşılanmaya çalışılmıştır.

İstemci web servisleri aracılığıyla piyasa verileri, haberleri, yorumları getirmektedir. İhtiyaç duyulduğunda farklı verilere farklı web servisleri aracılığıyla erişilebilir. Bu, uygulamaya genişleyebilir bir yapı da sunmaktadır.

J2ME uygulamalarına kalın istemci yapısının getirdiği bir özellik grafik çizme özelliklerinin gelişmişliğidir. İnce istemci modellerinin aksine MIDP cihazlardaki uygulamalarda istemci ekranında karmaşık grafiklerin çizilebilmesi mümkündür.

RMS aracılığıyla cihazda türlü kullanıcı tercihleri kalıcı olarak saklanarak daha kullanıcı dostu bir uygulama yaratılmıştır.

Uygulamanın boyutu yaygın MIDP cihazlarına sığabilecek kadar küçültülerek verilen hizmetin sınırlanması engellenmeye çalışılmıştır.

Bu uygulama, J2ME tabanlı bir istemci kullanarak Java'nın halihazırda sahip olduğu platform bağımsızlık, endüstri standartlarına olan bağlılık gibi özelliklerini de elde etmiş olmaktadır.

## **Kaynaklar**

[1] - World Wide Web Consortium, <http://www.w3.org/TR/2002/WD-wsdl12-20020709/>

[2] - Monica Pawlan, "Introduction to the J2EE Platform";  
<http://developer.java.sun.com/developer/technicalArticles/J2EE/Intro/>; Mart 2001

[3] - Sun Microsystems; "What Is the Java 2 Platform, Enterprise Edition";  
[http://java.sun.com/j2ee/sdk\\_1.2.1/techdocs/guides/j2ee-overview/OverviewTOC.fm.html](http://java.sun.com/j2ee/sdk_1.2.1/techdocs/guides/j2ee-overview/OverviewTOC.fm.html)

[4] - Jesus David Rivas, Roger Riggs; "Java 2 Platform, Micro Edition (J2ME) Technical Overview"; <http://java.sun.com/javaone/javaone2001/pdfs/2267.pdf>

[5] – Bill Buzbee, "J2ME Overview", 17 Mayıs 2002.