



BİLECİK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

**SMS VE ANDROID TABANLI UYGULAMALARLA SINIF
İÇİ ETKİLEŞİMİN ARTTIRILMASI**

Yusuf MUŞTU
Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Enver ÇAVUŞ

BİLECİK, 2012



**BİLECİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS
JÜRİ ONAY FORMU**

Bilecik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 12.01.2012 tarih ve2./2..... sayılı kararıyla oluşturulan jüri tarafından 18.01.2012 tarihinde tez savunma sınavı yapılan Yusuf MUŞTU'nun "SMS VE ANDROID TABANLI UYGULAMALARLA SINIF İÇİ ETKİLEŞİMİN ARTTIRILMASI" başlıklı tez çalışması Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak oy birliği/oy çokluğu ile kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI): Yrd. Doç. Dr. Enver ÇAVUŞ

ÜYE: Yrd. Doç. Dr. Cihan KARAKUZU

ÜYE: Yrd. Doç. Dr. Metin KESLER

ONAY

Bilecik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun .../.../..... tarih ve/..... sayılı kararı.

İMZA MÜHÜR

ÖZET

Bu tezde, eğitimde teknolojik olanaklardan faydalanarak öğrenci-eğitimci etkileşiminin ve verimliliğinin artırılması yönünde çalışmalar yapılmıştır. Öğrenci ve eğitimcilerin, cep telefonları ve İnternet olanakları kullanılarak interaktif olarak hızlı ve verimli bir biçimde iletişim kurabilmeleri için bir sistem tasarlanmış ve uygulama ortamı geliştirilmiştir.

Bu çalışma kapsamında, sınıf içi öğrenci-eğitimci etkileşimini daha verimli kılmak için mobil iletişimle bütünleşik bir sistem geliştirilmiştir. Geliştirilen bu sisteme kullanıcılar, ihtiyaca ve olanaklara göre SMS, Web arabirimi ve Android uygulaması olmak üzere 3 farklı şekilde erişmesi sağlanmıştır. Uygulamanın esnek olabilmesi ve kurumlarda kullanılan diğer otomasyon sistemleri ile entegre çalışabilmesi de planlanmış, bunu sağlayabilmek için Web servisleri geliştirilmiştir.

Geliştirilen sistemde, SMS yorumlama sistemi, web teknolojileri ve Android uygulamaları bütünleşik olarak çalıştırılmış ve ihtiyaca en uygun ve kolay erişilebilir bir sistem tasarlanmıştır. Kullanıcıların birbirleriyle SMS üzerinden haberleşmesini sağlayan GSM modem sisteme dâhil edilerek, mesajlaşmalar modem üzerinden sağlanmıştır. Diğer taraftan sistem, sadece araştırma alanında kalmaması için farklı uygulamada da kullanılabilecek şekilde esnek olarak tasarlanmıştır. İstenirse, web servisleri sayesinde kolayca farklı sistemlerle entegre edilebilecek yapıya getirilmiştir.

Anahtar sözcükler: .NET, Android, GSM Modem, Mobil iletişim, SMS

ABSTRACT

In this thesis, studies regarding increasing efficiency and interactivity of student-educator are done by taking the advantage of technological opportunities. A system was designed for students and educators by using mobile phone and internet potentials in order to use interactively and quickly by students and educators and an application setting was developed.

Within this study, to make the in-class student-educator interactivity more productive, a system integrated with mobile communication was developed. The users can access in three different ways to this system according to the necessities and potentials: SMS, Web interface and Android application. Flexibility and ease of the application in order to be used integrated to the other automation systems in other institution is also planned and web services were developed in order to achieve this.

In this system, SMS interpretation, web technologies and Android applications are worked integrated and a system which is most suitable for the needs and easy accessible was designed. By including GSM modem which enables the users to communicate with each other over SMS messaging was done over modem. On the other hand, the system was designed flexible for also enabling to be used not only in the research but also for various applications. It has been carried to a structure to be integrated to other systems easily by web services if required.

Keywords: .NET, Android, GSM modem, Mobile communication, SMS

TEŞEKKÜR

Bu çalışmanın yürütülmesi sırasında desteğini esirgemeyen danışmanım Yrd. Doç. Dr. Enver ÇAVUŞ'a, yardımlarıyla bana destek olan Yrd. Doç. Dr. Cihan KARAKUZU ve Yrd. Doç. Dr. Metin KESLER'e, Bilecik Üniversitesi Bilgi İşlem Daire Başkanlığı çalışma arkadaşlarıma, özellikle Murat FİDAN, Musa TURKAN ve Murat ÖZALP'a, Yrd. Doç. Dr. Meral EROL FİDAN'a, beni telefonla da olsa hiç yalnız bırakmayan canım aileme ve çalışmam sırasında küçük veya büyük yardımını esirgemeyen herkese teşekkür ederim.

İÇİNDEKİLER

TEZ ONAY SAYFASI

ÖZET.....	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ÇİZELGELER DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ.....	x

1. BÖLÜM: GİRİŞ.....	1
2. BÖLÜM: MOBİL SİSTEMLER	4
2.1 GSM.....	5
2.1.1 Abone Kimlik Modülü	6
2.2 Kısa Mesaj Hizmeti	6
2.2.1 Kısa Mesaj Servis Merkezi.....	7
2.3 Android İşletim Sistemi	9
2.3.1 Android İşletim Sisteminin Tarihçesi	9
2.3.2 Android İşletim Sisteminin Yetenekleri ve Özellikleri.....	10
2.3.3 Android İşletim Sisteminin Yapısı ve Yazılım Geliştirme.....	13
2.3.4 Android Uygulamalarına Erişim	14
3. KISA MESAJ VE WEB UYGULAMALARININ GELİŞTİRİLMESİ.....	15
3.1 GSM Modem	16
3.1.1 GSM Modem ile Haberleşme	18
3.1.2 Microsoft .NET Yazılım Geliştirme Platformu	23
3.1.3 Microsoft .NET Windows Servisleri	23
3.1.4 Microsoft .NET Web Servisleri	26
3.2 Windows Servislerinin Geliştirilmesi	27
3.2.1 Modem ile Haberleşen Windows Servisinin Oluşturulması.....	28
3.2.2 Mesajları Yorumlayan Windows Servisinin Oluşturulması.....	33
3.3 Web Arabirimi Geliştirilmesi	40
3.3.1 Sınıf Bilgilendirme Ekranı	41

3.3.2	Akademisyen Not Bilgilendirme Ekranı	42
3.3.3	Akademisyen Görüşme Yönetim Ekranı	43
4.	ANDROID UYGULAMASININ GELİŞTİRİLMESİ.....	45
4.1	Aktiviteler ve Uygulama Yaşam Döngüsü.....	45
4.2	AndroidManifest Dosyası	47
4.3	Uygulama Açılış Ekranı.....	48
4.4	Kayıt Ekranı	50
4.5	Menü Ekranı	54
4.6	Gelen Mesajlar Ekranı	55
5.	BÖLÜM – SONUÇLAR VE ÖNERİLER	58
	KAYNAKLAR.....	61
	EKLER.....	64
	EK 1. Windows Servisleri Uygulama Kodları	64
	EK 2. Android Uygulama Kodları	74

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 2.1. Android sürümleri (Android, 2011).	12
Çizelge 3.1. Mesaj gösterim türü kodları.	21
Çizelge 3.2. Windows servis uygulamalarında başlama tipi ve açıklamaları.	25

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 2.1. Mobil abone sayısı ve penetrasyon oranı (BTK, 2011).	5
Şekil 2.2. Gönderici – alıcı SMS iletim şeması.	8
Şekil 2.3. Mobil işletim sistemlerinin pazar payları artış grafiği (Gartner, 2011).	12
Şekil 2.4. Android işletim sistemi mimarisi (Android, 2011).	14
Şekil 3.1. Sistemin bileşenleri.	16
Şekil 3.2. SIM900 GSM modem.	17
Şekil 3.3. SIM900 GSM modemi ve uygulama sunucu bağlantısı.	17
Şekil 3.4. Computer management (Bilgisayar yönetimi).	18
Şekil 3.5. PuTTY bağlantı arayüzü.	19
Şekil 3.6. SIM900 GSM modem bağlantısı.	20
Şekil 3.7. Hafıza numarasına göre mesaj okunması.	20
Şekil 3.8. Okunmuş mesajların listelenmesi.	21
Şekil 3.9. Mesajların gönderilmesi.	22
Şekil 3.10. Mesajın silinmesi.	22
Şekil 3.11. Windows servis yöneticisi.	24
Şekil 3.12. FTP yayınlama servisinin özellikleri.	25
Şekil 3.13. İstemciler ve web servis sunucusu arasındaki iletişim.	27
Şekil 3.14. Regex tanımı.	28

Şekil 3.15. Modem haberleşme servisi akış şeması.	29
Şekil 3.16. Seri port bağlantı tanımı.	30
Şekil 3.17. Seri port durum kontrolü.	30
Şekil 3.18. Okunmamış mesajların alınması.....	31
Şekil 3.19. Gelen kısa mesaj tablosu.	31
Şekil 3.20. Giden kısa mesaj tablosu.	32
Şekil 3.21. Mesaj yorumlama iş akışı.	34
Şekil 3.22. Kayıt parametresi iş akışı.	35
Şekil 3.23. Ders bilgi parametresi iş akışı.....	36
Şekil 3.24. Not parametresi iş akışı.	37
Şekil 3.25. Görüşme istek tablosu.	38
Şekil 3.26. Görüşme durumu tablosu.....	39
Şekil 3.27. Msj parametresi iş akışı.	40
Şekil 3.28. Kullanıcı doğrulaması ve ders bilgilerinin getirilmesi.	41
Şekil 3.29. Sınıf bilgilendirme web sayfası.	42
Şekil 3.30. Not bilgilendirme web sayfası.	43
Şekil 3.31. Akademisyen görüşme yönetimi ekranı.	43
Şekil 4.1. Android aktivite yaşam döngüsü (Anroid, 2011).....	46
Şekil 4.2. AndroidManifest dosyası.	47
Şekil 4.3. Android uygulama izinleri.	48
Şekil 4.4. Açılış ekranı.	49

Şekil 4.5. Açılış ekranı iş akışı.....	50
Şekil 4.6. Kullanıcı kayıt ekranı.	51
Şekil 4.7. Kullanıcı kayıt işlemi iş akışı.	52
Şekil 4.8. SharedPreferences tanımı.	54
Şekil 4.9. Uygulama menü ekranı.....	54
Şekil 4.10. Gelen mesajlar ekranı.	55
Şekil 4.11. Gelen mesajlar iş akışı.	56
Şekil 4.12. Mesaj gönderme ekranı.....	57

SİMGELER VE KISALTMALAR DİZİNİ

1G	: 1. Nesil ~ 1rd Generation
2G	: 2. Nesil ~ 2rd Generation
3G	: 3. Nesil ~ 3rd Generation
ASP	: Aktif Sunucu Sayfaları ~ Active Server Pages
CEPT	: Posta ve Telekomünikasyon İdareleri Avrupa Konferansı ~ European Conference of Postal and Telecommunications
ETSI	: Avrupa Telekomünikasyon Standartlar Komitesi ~ European Telecommunications and Standards Institute
FTP	: Dosya Aktarım Protokolü ~ File Transfer Protocol
GSM	: Mobil İletişim İçin Küresel Sistem ~ Global System for Mobile Communications
HTML	: Zengin Metin İşaret Dili ~ Hypertext Markup Language
HTTP	: Zengin Metin Aktarım Protokolü ~ Hypertext Transfer Protocol
IDE	: Tümüleşik Geliştirme Ortamı ~ Integrated Development Environment
IMSI	: Uluslararası Mobil Kullanıcı Kimliği ~ International Mobile Subscriber Identity
ITU	: Uluslararası Telekomünikasyon Birliği ~ International Telecommunication Union
MCC	: Mobil Ülke Kodu ~ Mobile Country Code
MNC	: Mobil Şebeke Kodu ~ Mobile Network Code
MSIC	: Mobil Kullanıcı Kimlik Kodu ~ Mobile Subscriber Identity Code
SDK	: Yazılım Geliştirme Aracı ~ Software Development Kit
SIM	: Abone Kimlik Modülü ~ Subscriber Identity Module

SMS	: Kısa Mesaj Hizmeti ~ Short Message Service
SOAP	: Basit Nesne Erişim Protokolü ~ Simple Object Access Protocol
SQL	: Yapılandırılmış Sorgu Dili ~ Structured Query Language
W3C	: Dünya Çapında Ağ Birliği ~ World Wide Web Consortium
WAP	: Kablosuz Uygulama Protokolü ~ Wireless Application Protocol
WSDL	: Web Servisi Tanımlama Dili ~ Web Service Definition Language
WWW	: Dünya Çapında Ağ ~ World Wide Web
XML	: Genişletilebilir İşaretleme Dili ~ Extensible Markup Language

1. BÖLÜM: GİRİŞ

Eğitim alanında, bilgisayar ve iletişim teknolojilerinin kullanımı 1920'li yıllara kadar dayanmaktadır. Dünyada ilk radyo yayını 1927 yılında yapılmaya başlanmıştır. Akabinde de radyonun eğitim alanında kullanılma çalışmaları başlamıştır. Türkiye'de radyonun eğitim alanında kullanılması, ilk defa 1941 yılında kırsal kesimi hedefleyen "Ziraat Takvimi" isimli program vasıtasıyla olmuştur. Televizyonun eğitim alanında kullanılması ise Amerika'da 1953'te yapılan ilk eğitsel yayınlara dayandırılmaktadır (İşman, 2008).

Radyo ve televizyonun eğitim alanında ilk uygulamalarından günümüze kadar çok farklı teknolojik araçlar, eğitim amaçlı olarak kullanılmıştır. Bilgisayar teknolojilerindeki hızlı gelişmeler, bu teknolojilerin kullanıldığı tüm alanlarda da gelişimdeki ivmeyi arttırmaktadır. Diğer taraftan, günümüzde en önemli iletişim araçlarından birisi İnternet'tir. Artık insanlar, arkadaşlarının ne ile meşgul olduğunu sosyal ağlar sayesinde İnternet üzerinden takip etmektedir. Eğitim alanında da iletişim araçlarının yaygınlaşması bu bağlamda kaçınılmaz olmuştur.

İletişim araçlarında meydana gelen gelişmeler ve bu gelişmelerin hedef kitledeki yaygınlığı, öğrencilere ulaşmanın farklı teknolojilerle sağlanmasına sebep olmuştur. Eğitimde ilk olarak radyo kullanılarak geniş kitlelere erişim sağlanması yeni teknolojinin ilerlemesiyle birlikte yerini mobil cihazlara bırakmıştır. Bu teknolojiler kullanılarak eğitim alanında kullanılabilmesi konusunda çalışmalar yapılmıştır. Yapılan çalışmalarda SMS (Short Message Service) sistemlerinde, mobil cihazlarda uyumlu çalışan web sayfalarında ve mobil işletim sistemlerinde çalışan uygulamalara yer verilmiştir. Örneğin 2002 yılında, eğitimde SMS'nin kullanılmasına yer verilmiş ve etkilerini gözlemlenmiştir (Stone vd., 2002). Daha sonraki yıllarda yapılan çalışmalarda mobil cihazlarda farklı teknolojilerin ve yöntemlerin geliştirilerek eğitim alanında kullanımı devam etmiştir.

Cep telefonlarının yaygınlaşmasıyla, öğrenci bilgi sistemiyle bütünleşik çalışan bir PHP tabanlı web sayfası geliştirilmiş ve cep telefonundan öğrencilere web sayfasına bağlanarak kendilerine ait bilgiler gösteren çalışma yapılmıştır (Gülseren, 2006). Devamında, kısa mesaj servisi kullanılarak cep telefonlarında öğretim faaliyetlerinin

yapılabileceği belirtilmiş. GSM (Global System for Mobile Communications) modem üzerinden, sınıf içerisinde beyin fırtınası, değerlendirme, bilgi öğrenme gibi faaliyetlerle öğrenmeyi destekleyici uygulamalar yapılmıştır (So, 2009). Devam eden çalışmalarda, öğrencilere hareketli ortamlarda İngilizce kelimelerin öğretilmesini sağlayan kısa mesaj tabanlı öğrenme faaliyeti geliştirilmiştir (Çavuş ve İbrahim, 2009).

Derslere ait etkinliklerin mobil cihazlar üzerinden kısa mesajlarla sunulması, öğrenime katkısının incelenmesi üzerine çalışmalar gerçekleştirilmiştir (Ramli vd., 2010). Mobil iletişim cihazlarının gelişmesi ve bunlar üzerinde çalışan işletim sistemlerine özgü uygulamalarla eğitim desteklenmiştir. Öğrenmenin hızlı bir şekilde gerçekleştirilmesi için farklı teknolojiler düşünülmüş ve Android işletim sisteminde çalışan uygulamalar gerçekleştirilmiştir. Bu uygulama ile öğrencilere sınıf ortamında kısa sınavlar yapılmıştır (Pocatilu, 2011). Bir başka çalışmada, öğrencilerin birbirlerini bir iş birliği içerisinde gruplar halinde, öğrenme faaliyetlerinin artırılmasına yönelik çalışmalar gerçekleştirilmiştir (Lee, 2011). 2009 yılında Türkiye’de başlayan 3G hizmeti ile kablosuz erişim güçlendirilmiş ve 2010 yılında bu hizmeti alan abone sayısı artış göstermiştir. 2011 yılında, uzaktan eğitim çalışmalarını mobil cihazlara farklı teknolojilerle taşıma ve değerlendirmelerin yapılması devam etmiştir. Örneğin, “Temel Bilgi Teknolojisi” dersinin eğitim içeriği mobil cihazlara aktarılarak öğrencilerin her konumdan kendilerini geliştirebilecekleri çalışmalar gerçekleştirilmiştir (Çakır, 2011).

Bu tezde ise, örgün eğitimde teknolojik olanaklardan faydalanarak öğrenci-eğitimci etkileşiminin verimliliğinin artırılması yönünde çalışmalar yapılmıştır. Öğrenci ve eğitimcilerin, cep telefonları ve İnternet olanakları kullanılarak interaktif olarak hızlı ve verimli bir biçimde iletişim kurabilmeleri için bir sistem tasarlanmış ve araştırma amaçlı uygulama ortamı geliştirilmiştir.

Öncelikle bir GSM modem sayesinde, bir bilgisayarın GSM şebekesine bağlanması sağlanmıştır. Bu bilgisayar, kullanıcılardan SMS şeklinde gönderilen mesajları almak, işlemek ve –gerekli olduğunda- geriye SMS üzerinden cevap döndürmek üzere kullanılmıştır. Bu işlemlerin sağlıklı yapılabilmesi için, Windows servisleri yazılmıştır.

Uygulaması yapılmış olan sisteme kullanıcılar, ihtiyaca ve olanaklara göre 3

farklı şekilde erişebilmektedir:

- SMS ile erişim.
- Web arabirimi üzerinden (İnternet'ten) erişim
- Cep telefonuna yüklenen Android uygulaması üzerinden (İnternet'ten) erişim

Uygulamanın esnek olabilmesi ve kurumlarda kullanılan diğer otomasyon sistemleri ile entegre çalışabilmesi de planlanmış, bunu sağlayabilmek için Web Servisleri geliştirilmiştir. Bu sayede, örneğin, Öğrenci Bilgi Sistemleri ile entegre edilerek, öğrencilerin sınav notlarını sistem üzerinden öğrenebilmesi sağlanabilmektedir.

Bu çalışmada öncekilerden farklı olarak; SMS yorumlama sistemi, web teknolojileri ve Android uygulamaları bütünleşik olarak çalıştırılmış ve ihtiyaca en uygun ve kolay erişilebilir bir sistem tasarlanmıştır. Diğer taraftan sistem, sadece araştırma alanında kalmaması için farklı uygulamada da kullanılabilecek şekilde esnek olarak tasarlanmıştır. İstenirse, web servisleri sayesinde kolayca farklı sistemlerle entegre edilebilmektedir.

Bu çalışmanın ikinci bölümde; mobil iletişim araçları açıklanmış, SMS sistemleri ve Android işletim sistemi alanındaki konular hakkında bilgi verilmiştir.

Üçüncü bölümünde ise, geliştirilen sistem ve uygulamalar hakkında detaylı bilgi verilmiştir. Uygulamada kullanılan cihazlar ve bu cihazlara bağlantı bilgileri, veritabanı yapıları, kullanılan özel kodlar, uygulama ekranlarının açıklamaları bu bölümde açıklanmıştır.

Dördüncü bölümünde ise, uygulamanın Android işletim sistemi tarafında çalışma yapısı, bağlantı bilgileri, uygulama ekranlarının açıklamalarına yer verilmiştir.

Beşinci bölümde ise uygulama çalışmasının sonucunda elde edilen bulgular paylaşılmış; eğitim alanında iletişimi arttırmak konusunda teknolojinin katkılarından bahsedilmiş, bu çalışmanın uygulamada kullanımı konusunda önerilere yer verilmiş ve ileride bu konu üzerinde çalışmak isteyenler için, öneriler paylaşılmıştır.

2. BÖLÜM: MOBİL SİSTEMLER

Ucuzlayan fiyatları, kullanım kolaylığı ve sağladığı rahatlık gibi unsurlar mobil kullanıcı sayısını hızla kullanıcı sayısı artmaktadır. Mobil araçlar yardımıyla sunulan mobil İnternet servisleri, kullanıcıların sabit bir bağlantı noktasına ihtiyaç duymasını gerektirmeden, onlara daha kişisel içerikler ve özel servisler sunmuştur (Sarısakal ve Aydın, 2003).

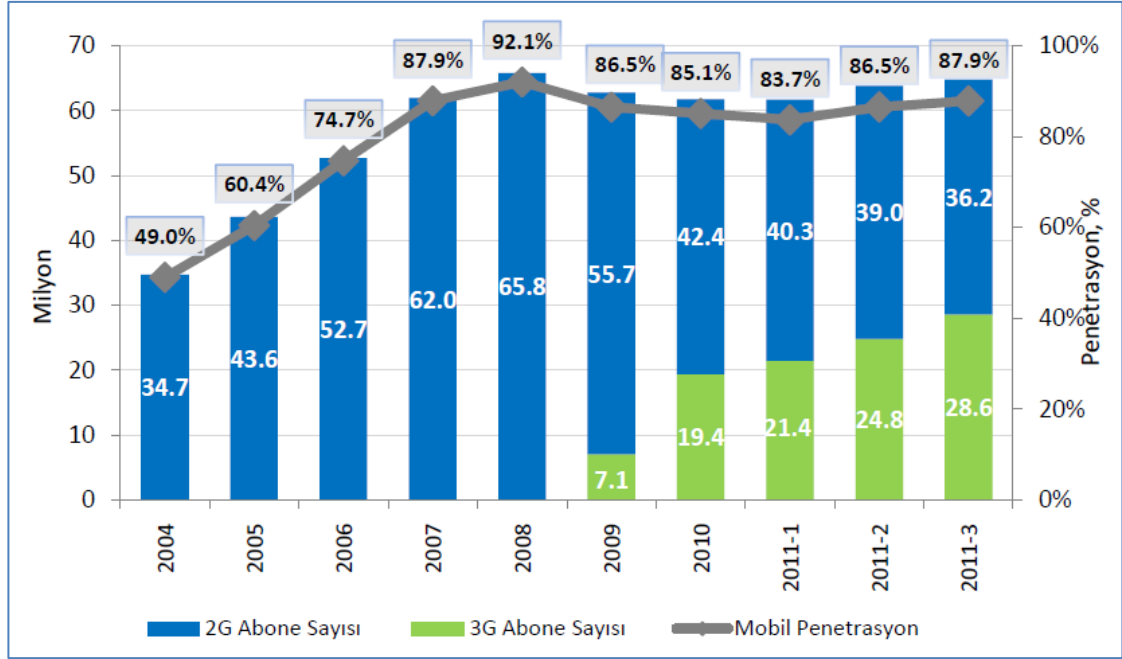
Mobil telefonların, yaygınlaşmasının en önemli nedenleri (Yuan ve Cheng, 2004);

- Mobil telefonların her zaman, her yerde kişilerin yanında taşınması,
- Her zaman iletişime açık olması,
- Bire bir iletişim kurulmasının daha fazla ilgi çekmesi,
- Gönderilen mesajların kaydedilerek daha sonra yanıtlanabilmesi,
- Bire bir sesli ve görüntülü iletişim kurma imkanı vermesi gibi sıralanabilir.

Eylül 2011 itibariyle Türkiye’de yaklaşık %87,97 penetrasyon¹ oranına karşılık gelen toplam 64,8 milyon mobil abone bulunmaktadır. Temmuz 2009’da 3G hizmet sunumunun başlamasıyla Eylül 2011 itibariyle 3G abone sayısı 28,6 milyona ulaşmıştır. Şekil 2.1’de 2G ve 3G mobil abone sayısı ile penetrasyon oranları yıllar itibariyle karşılaştırılmaktadır (BTK, 2011).

Ortalama 64 milyona yakın mobil abone için, 2009 yılında başlayan %11’lik bir orana sahip 3G abone sayısı, 2011 yılının üçüncü çeyreğine gelindiğinde ise %33’lük bir artış sağlayarak %44’lük bir orana sahip olmuştur.

¹ Ürün veya hizmetin hedef kitlesindeki yaygınlığı



Şekil 2.1. Mobil abone sayısı ve penetrasyon oranı (BTK, 2011).

2.1 GSM

GSM, mobil haberleşme için kabul edilmiş ikinci nesil, sayısal hücreli olarak tanımlanabilir. GSM, basit olarak devre anahtarlamalı sayısal ve analog veri bağlantısı hizmetleri sunan bir sistemdir (Megep, 2007).

1982 yılında CEPT (European Conference of Postal and Telecommunications) tarafından GSM sistemi kurulmaya başlanmış ve 1986 yılında GSM'in temel çalışma prensibi belirlenmiştir. 1990 ve 1991'de GSM 900 sistemi test edilmeye başlanmıştır. Çalışmalar Alman-Fransız ortaklığıyla kurulan GSM tarafından değişik bölgelerde sürdürülmüştür. Grup, hedeflerini ortaya koymuş ve hedefi gerçekleştirmek üzere protokoller belirlemiştir. 1991 yılında 18 ülke tarafından kullanılmaya başlanmıştır. 1992 senesinde ilk GSM şebekesi faaliyete geçmiştir. Amerika ve Japonya ise GSM ile uyumlu olmayan kendi mobil sistemlerini kurmuştur. Türkiye, GSM teknolojisi ile ilk defa 23 Şubat 1994'de tanışmıştır (BTK, 2011).

GSM için belirlenen standartlar, işlevleri ve erişimler için gerekli olan bağdaştırma ihtiyaçlarını da kapsamaktadır. Altyapıda kullanılacak olan donanım üzerine bir sınırlama getirilmemiştir. Böylece, GSM altyapısı hazırlayan kuruluşlar marka bağımsız çalışma şansına sahip olmuşlardır. (Megep, 2007)

2.1.1 Abone Kimlik Modülü

GSM ile ilk kez tanımlanan önemli özelliklerden biri de Abone kimlik modülü (Subscriber Identity Module ~ SIM) kartıdır. Bu kartta abone kimlik bilgisi bulunur. Bu sayede abonenin bir mobil telefona bağlılığı olmaz ve rahatlıkla değiştirebilir bir yapı sağlanmış olur. SIM kart üzerinde; uluslararası mobil abone kimliğini içeren IMSI (International Mobile Subscriber Identity) yer almaktadır. IMSI, uluslararası abone tanımlama numarası anlamına gelir. Bu numarada abonenin ve şebekenin numarası birlikte yer alır. MCC (Mobile Country Code) diye tanımlanan şebekenin uluslararası kodu ve MNC (Mobile Network Code) diye tanımlanan şebekenin ülke içerisindeki numarasıdır. MSIC (Mobile Subscriber Identity Code) abonenin kişisel numarasıdır (Imran ve Hussain, 2011).

Bir GSM abonesi için uluslararası abone tanımlama numarasını hesaplamak istersek, $IMSI = MCC + MNC + MSIC$ formülünü kullanılır. Türkiye'deki şebekeler için MCC, 286 olarak kullanılır. MNC ise Turkcell için 01, Vodafone için 02, Avea için ise 03'tür. Örneğin, Türkiye'de Vodafone abonesi olan ve telefon numarası 90 544 123 45 67 olan bir abone için IMSI numarası 28602905441234567 olacaktır.

2.2 Kısa Mesaj Hizmeti

Kısa mesaj hizmeti (Short Message Service ~ SMS), GSM şebekeleri üzerinden mobil telefonlar aracılığı ile ileti yollanması ve alınması işlemlerini kapsayan 160 karaktere kadar ulaşabilen bir mobil iletişim servisi (Haghirian vd., 2005).

SMS ETSI (European Telecommunications Standard Institute ~ Avrupa Telekomünikasyon Standartlar Komitesi) tarafından oluşturulan standart ile 7 bitlik bir alfabenin yanında 8 bit veya 16 bit şeklinde kullanılarak farklı karakter desteğini verir. Bu tarz kullanımla birlikte 8 bitlik bir yapıda 140 karakter, 16 bitlik bir yapıda ise 70 karakter gönderimi gerçekleştirilir. Bu karakterler farklı ülkelerdeki alfabenin kullanılması veya logo, melodi gibi bilgilerin gönderilmesi için kullanılır. 16 bitlik sistem ise genelde alfabeleri binlerce harf içeren uzak doğu ülkeleri tarafından kullanılır (Siang vd., 2003).

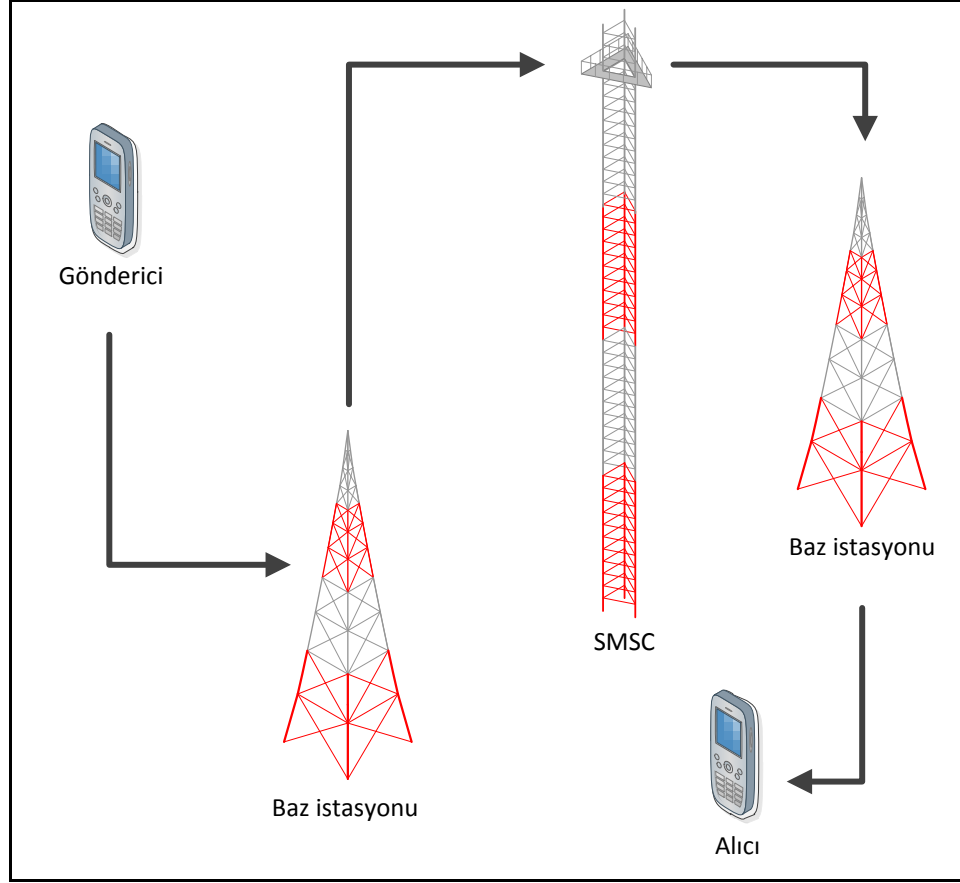
SMS gelişiminin tarihçesi (Hillebrand, vd., 2010):

- 1992: Avrupa’da çeşitli operatörler tarafından ilk kabul edilebilirlik testleri başladı.
- 1993: İlk noktadan noktaya (point-to-point) çalışan SMS alma yeteneğine sahip telefonlar piyasaya sürüldü. Bu telefonların SMS gönderme özelliği bulunmamaktadır ve operatörler tarafından, “sesli mesaj bildirimi” amacıyla kullanılmıştır.
- 1994: Artık yeni üretilen tüm telefonlar, SMS alma yeteneğine sahiptir.
- 1995: Tüm GSM ağları, SMS gönderebilmektedir. Uluslararası SMS dolaşımı başlamıştır.
- 1996: Her yeni üretilen telefon, SMS gönderme yeteneğini kazanmıştır.
- 2008: 3-4 trilyon SMS gönderilmiştir.
- 2010: Bir yıl içerisinde 6,1 trilyon SMS gönderilmiştir.(ITU, 2010)

Sesli görüşmeden farklı olarak SMS’de, alıcı ve verici arasında direkt bağlantı kuramazlar. Burada mesaj SMSC (Short Message Service Center – Kısa Mesaj Servis Merkezi) aktarılır ve buradan alıcıya teslim edilir. Alıcının cep telefonunun ulaşılabilir olduğu anda SMSC’ de saklı tutulan mesaj alıcıya gönderilir.

2.2.1 Kısa Mesaj Servis Merkezi

Kısa mesaj servis merkezi (Short Message Service Center ~ SMSC), GSM operatörlerinin kısa mesaj trafiğinin yönetilmesinden sorumludur. İletiler bir telefondan diğerine ulaştırılırken SMS servis merkezine (SMSC) gelir ve daha sonra alıcının telefonuna yönlendirilir (Yüksel ve Zaim, 2009). Şekil 2.2’de gönderici ve alıcı arasında meydana gelen kısa mesaj iletiminin şeması gösterilmiştir.



Şekil 2.2. Gönderici – alıcı SMS iletim şeması.

Cep telefonundan bir kısa mesaj gönderildiği zaman, kısa mesajların ilk olarak SMSC' ye gönderilir. Alıcıya mesaj göndermek SMSC sorumluluğundadır. Alıcı cep telefonunun durumuna bağlı olarak mesaj iletilir. Alıcının cep telefonu bir baz istasyonuna bağlı değilse mesaj SMSC üzerinde tutulur. Bir sonraki gönderme işlemi için kuyruğa aktarılır. Alıcı telefonu aktif olduğunda kuyrukta bekleyen mesaj alıcıya iletilir.

Bir SMS mesajı, alıcıya ulaşmadan, birden fazla SMSC geçmesi gerekebilir. SMSC üzerinde bekleyen mesajların bir geçerlilik süresi vardır. Bu süre içerisinde SMSC üzerinde mesaj saklanır. Alıcı uzun süre uygun konuma gelmez ise mesaj SMSC üzerinden silinir. Bu işlemden sonra mesaj gönderim kuyruğuna aktarılmaz.

SMSC üzerindeki bir iş yükü de kısa mesajların alıcılara ulaştıktan sonra mesaj göndericisi veya göndericilerine geri bildirimde bulunmaktadır. Gönderici iletmek istediği kısa mesaj için bir iletildi raporu isteyebilir. Bu tarz rapor geri bildirimlerinde, kısa

mesaj alıcısına ulaştığında SMSC mesajı gönderen kişiye durum bilgisini yolar.

2.3 Android İşletim Sistemi

Mobil cihazların donanımsal olarak gelişmesi, onların bir işletim sistemi üzerinden kontrol edilmesi ve kullanılması ihtiyacını ortaya çıkarmıştır. Önceleri sadece iletişim aracı olarak kullandığımız mobil aygıtlar şimdi birden fazla özelliği barındıran bir hale gelmiştir. Kelime olarak insansı robot anlamında da kullanılmış olan Android, Linux çekirdeğine sahip bir işletim sistemidir. Açık kaynak kodlu olan Linux işletim sistemi, özgür ve ücretsiz bir şekilde dağıtılabilir, değiştirilebilir ve kullanılabilir yapıya sahiptir.

Bu gelişim onun bir işletim sistemi seviyesinde yazılıma gereksinimini ortaya koymuştur. Mobil cihazların işletim sistemi seviyesinde bir yazılıma ihtiyaç duyması, yazılım geliştiricileri bu alanda çalışmaya yöneltmiştir. Açık kaynak kodlu olan Android işletim sistemi, bu özelliği ile farklı cihazlara uyumlu hale getirilerek, gelişmesine ve kullanım oranının artmasına sebep olmuştur. Bu yönelimle birlikte ortaya Linux çekirdeğini kullanan, açık kaynak kodlu bir işletim sistemi olan Android ortaya çıkmıştır.

2.3.1 Android İşletim Sisteminin Tarihçesi

Mobil cihaz üreticileri, geliştirmiş oldukları cihazlar için işletim sistemi seviyesinde yazılım geliştirmişlerdir. Nokia, Apple gibi büyük telefon üreticileri Symbian, iOS gibi işletim sistemlerini geliştirerek kullanmışlardır. Ancak bu işletim sistemleri yalnızca belirli cihazlarda kullanılmış, fakat piyasada yaygınlaşamamışlardır (Turan, M). Bu durum üreticileri, farklı cihazlarda çalışabilen, açık kaynak kodlu bir işletim sistemi seviyesinde yazılıma ihtiyacın olduğunu ortaya çıkarmıştır.

2003 yılında Andy Rubin, Nick Sears, Rich Miner ve Chris Whie tarafından Android Inc. adlı bir şirket kurulmuştur. Bu şirket, çalışmalar sonrasında Linux çekirdeğine sahip bir işletim sistemini cep telefonunda çalıştırmayı başarmıştır. Bu şekilde ilk Android çalışmaları başlamıştır. Google, mobil işletim sisteminde çalışmalarına Android işletim sistemini katmak istemesi nedeni ile 2005 yılında bu firmayı satın almıştır.

Google'ın Android firmasını satın almasıyla birlikte aynı zamanda bu ekipte çalışan kişilerin de Google ekibine katılması sağlanmıştır. Google 2007 yılına kadar bu projeden bahsetmemiştir. 2007 içerisinde geliştirilmesi süren Android'i duyurarak, ortak bir işletim sistemi geliştirmeye davet etmiştir. 5 Kasım 2007 tarihinde Google'ın başta olduğu 34 donanım, yazılım ve mobil operatör firması tarafından Open Handset Alliance adında bir birlik oluşturulmuştur. Bu birliğin amacı mobil cihaz sektöründe ortak, açık ve standart bir platform oluşturmaktır. Şu an bu birlikte 84 firma yer almaktadır (Open Handset Alliance, 2011).

2008 yılında Google'ın Android'e ait kaynak kodlarını sunmasıyla, telefon üreticileri bu yapıyı ürettikleri cep telefonlarına entegre etmeye çalışmıştır. Bu çalışmalar sırasında, HTC firması ilk Android işletim sistemine sahip cep telefonunu (HTC Dream) piyasaya sürmüştür(Kilgo,2009).

Android 1.0 sürümünü kullanan HTC (Dream); internet tarayıcısı, Medya yürütücüsü, Gmail hesap yöneticisi olarak kullanımının yanında Google tarafından işletilen uygulama mağazası Android Market'e de destek vermiştir.

2.3.2 Android İşletim Sisteminin Yetenekleri ve Özellikleri

Teknolojinin gelişmesiyle farklı amaçla kullanılan aygıtların tek bir aygıtta toplanması, boyutlarının küçültülmesi, taşınabilirliğin artırılması ve kullanılan işletim sistemine göre kişiselleştirmenin artırılması sayesinde, mobil cihazlarda eskisine göre çok daha farklı türlerde işlem yapılabilmektedir. Önceleri sadece görüşme yapma, kısa mesaj gönderme gibi işlemlerde kullanılan cep telefonları, günümüzde internette gezinme, konum tespiti yapma, dış ortam sensörleri ile bilgi öğrenme, multimedia uygulamaları gibi birçok alanda kullanılmaktadır.

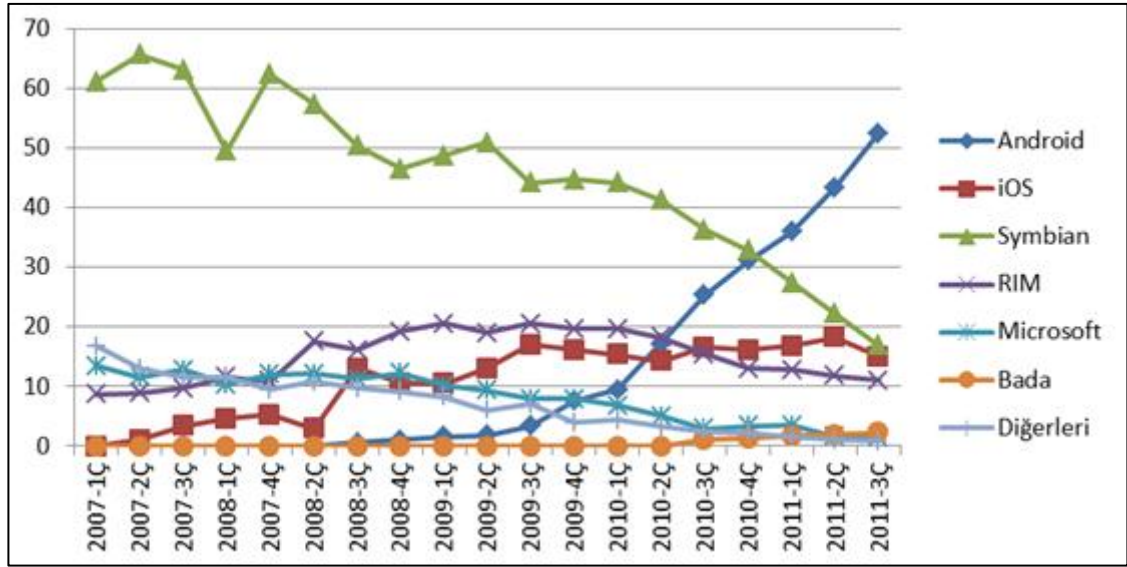
İlk başlarda cep telefonları için düşünülen Android işletim sisteminin, farklı ürün türlerinde kullanılmasıyla bu işletim sisteminin yetenekleri geliştirilmiştir. Android, günümüzde birçok farklı ürün türlerinde kullanılmaktadır. Farklı alanlardaki bu kullanım desteği, onun hızlı bir şekilde yaygınlaşmasına ve gelişmesine katkı sağlamıştır.

Android işletim sistemini kullanan cihazlar, genel olarak şu şekilde gruplandırılabilir (Taç, 2011):

- Akıllı telefonlar (Smartphone)
- Tablet bilgisayarlar
- Televizyonlar
- GPS (Global Positioning System ~Küresel Konumlama Sistemi) cihazları
- Netbooklar
- Ortam oynatıcıları (DVD, VCD, vb.. oynatıcılar)
- Oyun konsolları
- Mutfaklarda kullanılan beyaz eşyalar
- E-Kitap okuyucular

Android işletim sisteminin duyurulması ile birlikte kurulan Open Handset Alliance birliği, onun geliştirilmesinde ve kullanılmasında önemli bir role sahiptir. Bunun yanında Android işletim sisteminin Linux çekirdeğini kullanması, açık kaynağa destek vermesi, farklı cihaz türlerinde çalışması gibi nedenler, üreticiler tarafından kullanılmasına ve hızlı bir şekilde yaygınlaşmasına sebep olmuştur.

Bu yaygınlaşma onu rakiplerini kısa bir sürede geçmesine sebep olmuştur. Gartner tarafından her yıl oluşturulan raporlarda, 2007'nin birinci çeyreğinde %61.2 pazar payına sahip olan Symbian 2011'in üçüncü çeyreğinde geldiğinde %16.9'a kadar gerilemiştir. Android işletim sistemine baktığımızda ise, 2008'in üçüncü çeyreğinde yaygınlaşmaya başlamış 2011'in üçüncü çeyreğine geldiğimizde ise %52.5'lik bir Pazar payına sahip olmuştur. Mobil işletim sistemlerinin yıllara göre pazar payı değişimi Şekil 2.3'de gösterilmiştir.



Şekil 2.3. Mobil işletim sistemlerinin pazar payları artış grafiği (Gartner, 2011).

Android işletim sisteminin hızlı bir şekilde yayılması ve farklı cihaz türlerinde kullanılması farklı sürümlerinin ortaya çıkmasına sebep olmuştur. İlk zamanlarda cep telefonlarında farklı, tablet bilgisayarlarda farklı bir sürüme sahip olan Android, 4.0 sürümünün çıkartılmasıyla hem cep telefonlarında hem de tablet bilgisayarlarda kullanılan ortak bir işletim sistemi haline getirilmiştir.

Android işletim sisteminin gerek güncellemelerden, gerekse yeni platform uyumluluğunu sağlamak adına yeni sürümleri oluşturulmuştur. Oluşturulan bu her sürüme aynı zamanda bir kod adı verilmiştir. Oluşturulan bu sürümlere ait kod adı ve çıkış tarihi Çizelge 2.1’de gösterilmiştir.

Çizelge 2.1. Android sürümleri (Android, 2011).

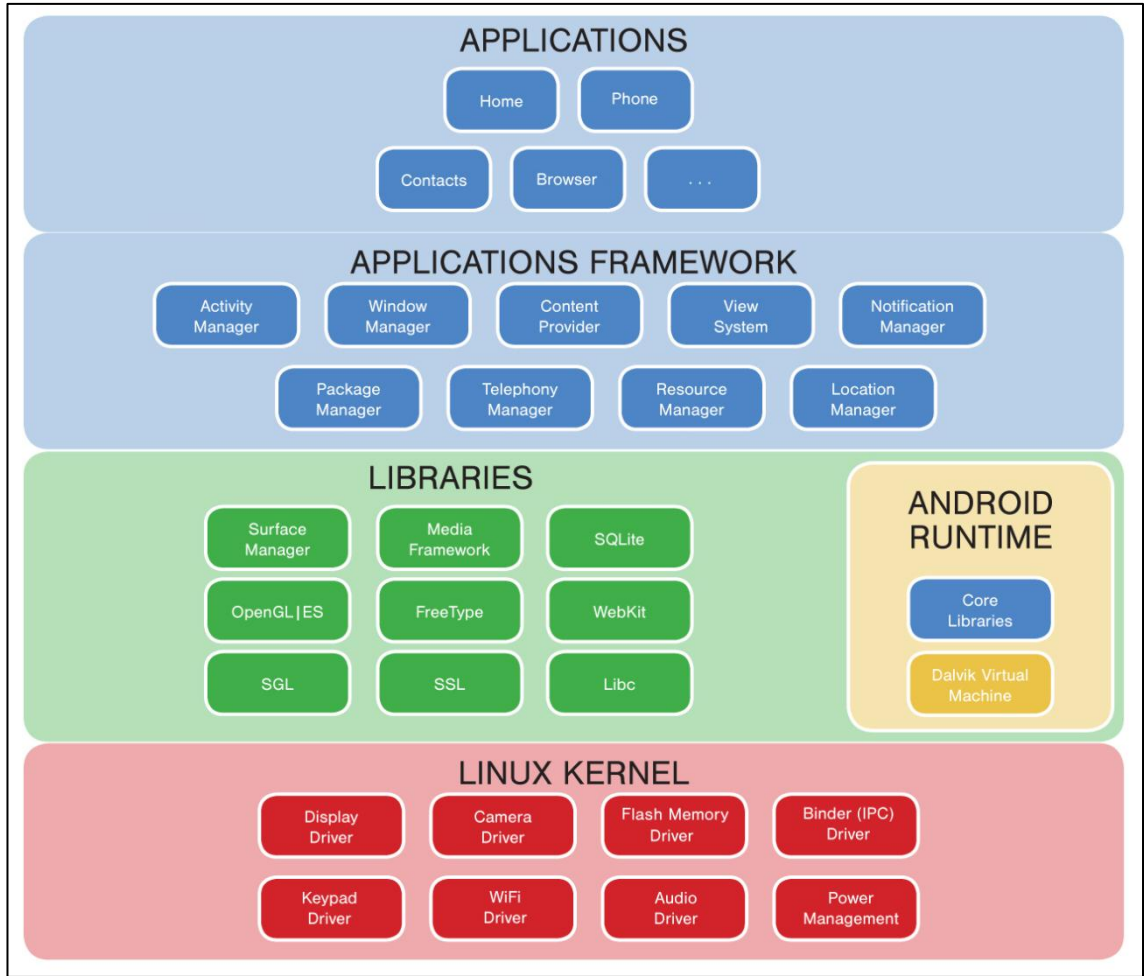
Sürüm	Kod Adı	Tarih
Android 1.0	Apple pie	Eylül 2008
Android 1.1	Banan bread	Şubat 2009
Android 1.5	Cupcake	Mayıs 2009
Android 1.6	Donut	Ekim 2009
Android 2.1	Eclair	Ocak 2010
Android 2.2	Froyo	Mayıs 2010
Android 2.3.X	Gingerbread	Aralık 2010
Android 3.X	Honeycomb	Şubat 2011
Android 4	Ice cream sandwich	Ekim 2011

2.3.3 Android İşletim Sisteminin Yapısı ve Yazılım Geliştirme

Android işletim sistemine özgü yazılım geliştirme Google tarafından derlenmiş Android SDK ile gerçekleştirilebilir. Kullanım kolaylığı sunan bir IDE (Integrated development environment ~ Tümüleşik Geliştirme Ortamı) ile geliştirilecek olan yazılımlar, hızlı ve kolay bir şekilde oluşturulabilir. Uygulama geliştirilmeye başlamadan önce, Google'ın Android yazılım geliştiricileri için oluşturduğu "<http://developer.android.com>" adresinden yararlanılabilir. Bu web adresinden, Android geliştirme kiti indirilebilir.

Android işletim sistemi, katmanlardan oluşan bir yapıya sahiptir. Bu yapı Şekil 2.4'de gösterilmiştir (Android, 2011). Bu yapıda bulunan katmaları şu şekilde açıklayabiliriz;

- **Applications:** İşletim sisteminde kullanılan uygulamalar (Telefon Rehberi, İnternet Tarayıcısı, Eposta Yönetim uygulaması, vb...)
- **Application Framework:** Android, geliştiricilerin uygulamalarına zenginlik katmak amacıyla donanım ve yazılım desteğini sağlar.
- **Libraries:** Linux çekirdeğine erişimin sağlanması için bir takım bileşenlere ihtiyaç duyulur. Bu bileşenlerin yer aldığı bölüme libraries denir.
- **Android Runtime:** Dalvik virtual machine (Dalvik sanal makinesi) ile dosyaların dex formatından dönüştürülerek çalışmasını sağlar.
- **Linux Kernel:** Linux çekirdeğinin kullanılması ile güvenlik, bellek yönetimi, süreç yönetimi, ağ yığını ve sürücü modeli gibi temel sistem hizmetleri için haberleşme yapısının gerçekleştirildiği bölümdür.



Şekil 2.4. Android işletim sistemi mimarisi (Android, 2011).

2.3.4 Android Uygulamalarına Erişim

Android işletim sisteminde uygulamalar iki kategoride toplanabilir. Bunlar;

- Yerel Uygulamalar (Native Application)
- Web Uygulamaları (Web Application)

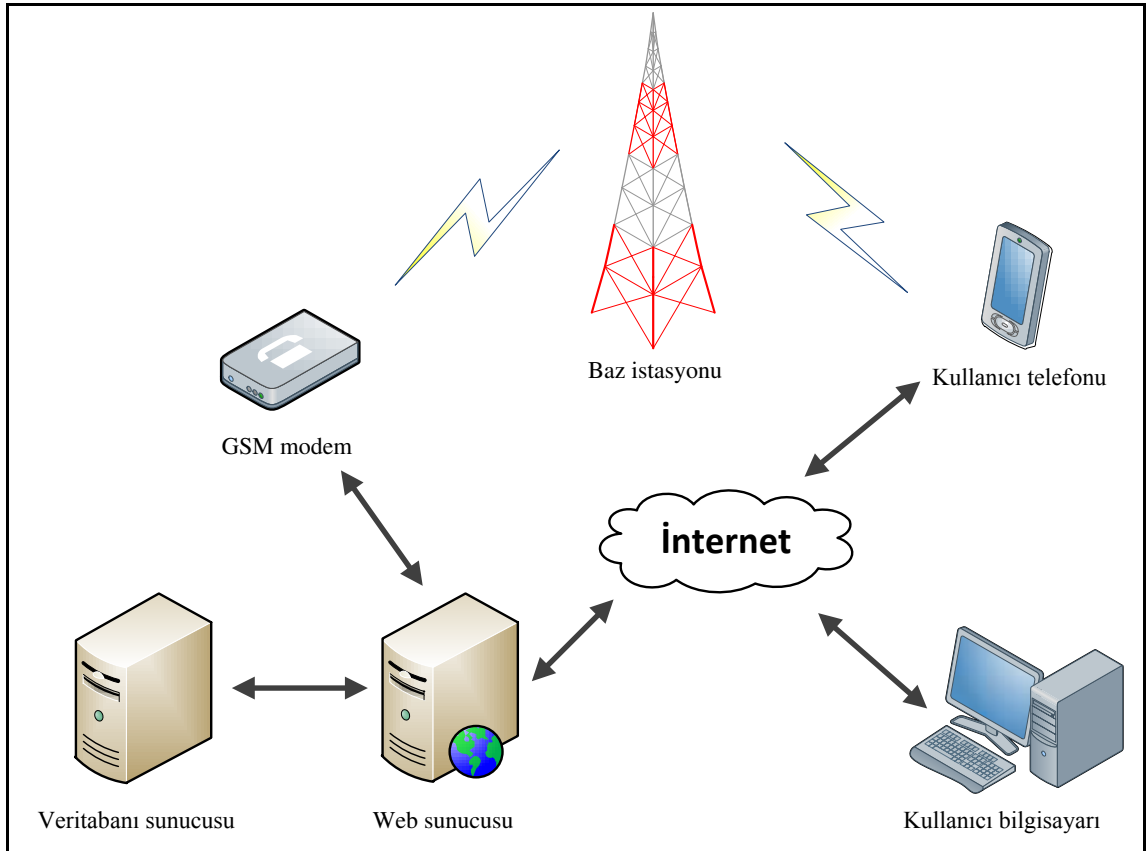
Bu iki kategoriyi açıklamak gerekirse yerel uygulamalar APK (Android Package) uzantılı olup, cihaza aktarılmasının ardından kurulması gerekir. Uygulamanın verdiği hizmetlere göre genellikle internet olmadan da çalışabilir. Web uygulamaları ise genelde internet ortamına ihtiyaç duyarak çalışır. Web uygulamalarında cihaz uyumsuzluğu ve ihtiyaca göre kişiselleştirme, yerel uygulamalara göre daha azdır.

3. KISA MESAJ VE WEB UYGULAMALARININ GELİŞTİRİLMESİ

Bu tez çalışmasında, öğrenci-eğitmen etkileşimini ve iletişimini kolaylaştırmak amacıyla bir otomasyon sistemi geliştirilmesi amaçlanmıştır. Bu sistem sayesinde, öğrenciler ve öğretmenler mekândan bağımsız olarak cep telefonları ile iletişim sağlayabilmektedir. İletişim sağlanabilmesi için, SMS ve Android üzerinde çalışan yazılım teknolojileri kullanılmıştır. Uygulamaya öğrencilerin dönem bazında ders bilgilerinin yer aldığı bir veritabanına, mobil haberleşmede ihtiyaç duyulan tabloların eklenmesi ile başlanmıştır. Web sunucusu ile mobil cihazlarla haberleşmede kullanılan GSM modem bir birleriyle haberleşecek şekil bağlanmıştır. Kullanıcılarla etkileşimi gerçekleştirmek, gelen istekleri yorumlamak adına Windows ve web servisleri oluşturulmuştur. Kullanıcıların web üzerinden ve Android işletim sistemine sahip cihazlardan sistemin kullanılmasını gerçekleştirmek için uygulama yazılımları geliştirilmiştir.

Sistem genel olarak öğrenci bilgilerinin bulunduğu bir veritabanı, kullanıcı etkileşimini kontrol edip kullanıcıları yönlendiren servislerin barındığı uygulama sunucusu, GSM üzerinden haberleşmenin sağlandığı bir GSM modem ve kullanıcılardan oluşmaktadır. Sistem mimarisinde anlatılan bu bileşenler Şekil 3.1’de yer verilmiştir.

Kullanıcılarla iletişim, 3 farklı yöntemle gerçekleşmektedir. Bunlardan ilki ve kullanım oranı en fazla olan SMS üzerinden iletişimdir. SMS üzerinden iletişimin sağlanması için GSM modem kullanılmıştır. Bu sayede ticari bir firmaya ihtiyaç duyulmadan (GSM opertörü hariç) kısa mesaj iletişimi gerçekleştirilmiştir. Diğer iletişim yöntemleri ise Web sayfası ve Android işletim sistemine göre yazılmış uygulamalardır. Bu uygulamaların bütünleşik bir şekilde kullanılması için web servisleri geliştirilmiştir. Uygulamada kullanılan veriler, ortak kullanılacak şekilde bir veritabanı sunucusunda tutulmuştur.



Şekil 3.1. Sistemin bileşenleri.

3.1 GSM Modem

GSM şebekesine, bir cep telefonu olmadan SIM kartı yardımıyla bağlanılmasında GSM modemler kullanılır. GSM modemler destekledikleri AT komut setinde, genişletilebilir arabirim bağlantılarına ve frekans bant aralığına göre farklılıklar göstermektedir. Bu uygulamada kısa mesaj gönderimine ve alımına destek verecek AT komut setine sahip GSM modemlerden SIM900 tercih edilmiştir.

RS232 arabirimine sahip GSM modem bir SIM kartı yardımıyla GSM şebekesine bağlantısı gerçekleştirilmiştir. RS-232 arabirimi ile iki cihaz arasında iletişim gerçekleştirilebilir. 15 ile 30 metre aralığında değişen iletim mesafesine sahip RS-232, bir adaptör yardımıyla farklı arabirimlere dönüştürülerek iletişim imkânı sağlanabilir (Axelson, 2000).

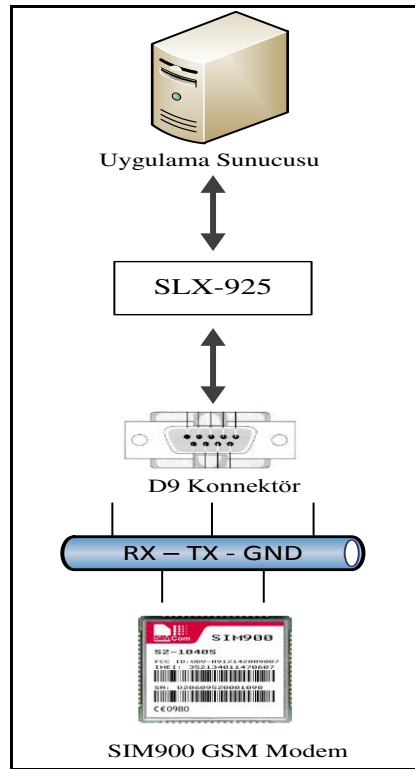
Şekil 3.2’de bu tez çalışmasında kullanılan SIM900 GSM modem gösterilmiştir. Bu GSM modeme, bilgisayar üzerinden bağlanabilmek ve gerekli işlemleri yapmak için

seri port arabirimi gerekmektedir.



Şekil 3.2. SIM900 GSM modem.

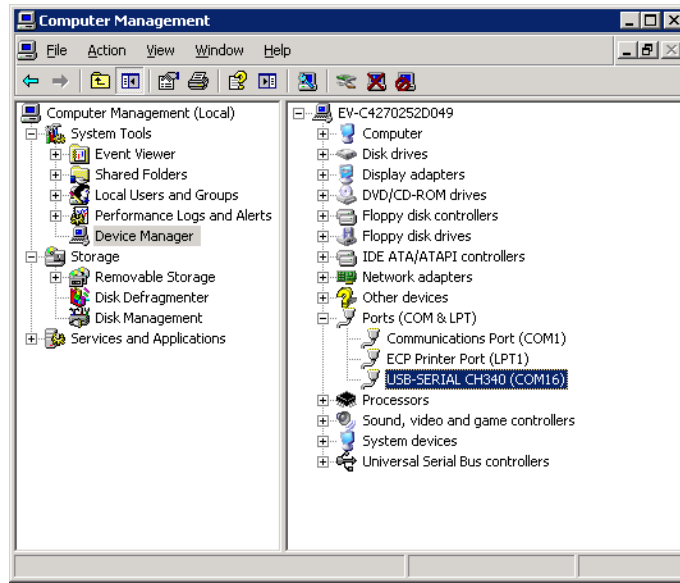
Yeni anakart modellerinde bu arabirim olmayabilir. Bu tarz durumlarda USB'den Seri arabirime dönüşüm işlemi gerçekleştiren dönüştürücüler bulunmaktadır. Bu dönüştürücüleri kullanarak SIM900 GSM modemine veya seri port arabirimine ihtiyaç duyan başka cihazlara erişim sağlanabilmektedir. Bu çalışma kapsamında, S-LINK SLX-925 çevirici kullanılmıştır. Örnek bağlantı Şekil 3.3'de gösterilmiştir.



Şekil 3.3. SIM900 GSM modemi ve uygulama sunucu bağlantısı.

Fiziksel olarak bağlantı işlemini gerçekleştirdikten sonra, seri port bağlantı desteği olan bir konsol programı vasıtasıyla, modem arabirimine erişilebilmektedir. Modem arabirimine ulaşmada hangi seri portun kullanıldığı önemlidir. Port numarasını Microsoft Windows işletim sisteminde, “Start – Programs - Administrative Tools - Computer Management” yolunu izleyerek veya bilgisayar yönetimi uygulamasının çalıştırılmasıyla öğrenilebilir. Windows Server 2003 işletim sistemi için Computer Manager (Bilgisayar yönetimi) ekranı Şekil 3.4’de gösterilmiştir.

Computer Managemet uygulamasında, “System Tools” bölümünden Device Manager’a tıklanarak, Ports (COM & LPT) altında yer alan USB-SERIAL (bu isim kullanılan dönüştürücünün modeline göre değişebilir) aygıtın hangi seri portu kullandığı öğrenilebilir. Uygulamada, COM16 seri port arabirimi kullanılmaktadır. İstenildiği takdirde COM16 arabirimi farklı bir arabirim numarası ile değiştirilebilmektedir.



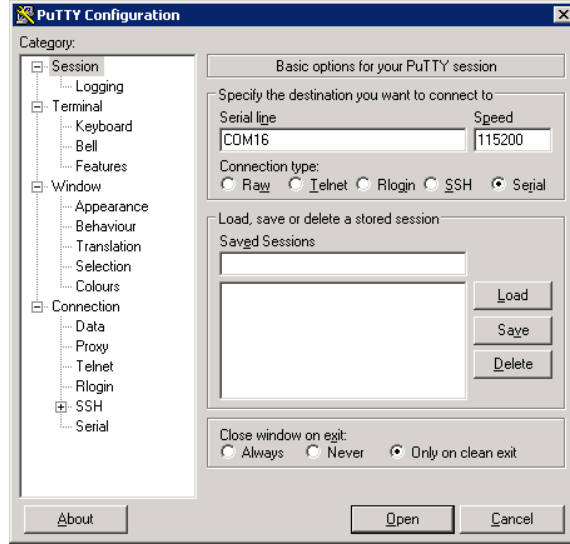
Şekil 3.4. Computer management (Bilgisayar yönetimi).

3.1.1 GSM Modem ile Haberleşme

Bilgisayardan seri port arabiriminde bulunan bir cihaz ile haberleşme konusunda birçok ücretli veya ücretsiz yazılım bulunmaktadır. Bu bağlantı örneği için ücretsiz olan

PuTTY¹ 0.61 sürümlü terminal yazılımı kullanılmıştır.

PuTTY yazılımı açıldığında seri porttan bağlanmak için port numarası ve bağlantı bilgilerine ihtiyaç duyacaktır. Bağlantı sırasında kullanılan bu yazılımın arabirimi, Şekil 3.5’ de gösterilmiştir.



Şekil 3.5. PuTTY bağlantı arayüzü

Bağlantının gerçekleşip gerçekleşmediğini anlamak için SIM900 GSM modemine ait AT komutlarından, AT veya ATI kullanılabilir. Bu komutların kısaca açıklaması şu şekildedir;

AT: Bu komut, bağlı olan cihaza ulaşıp ulaşılmadığı bilgisini geri döndürmektedir. Dönen cevap “OK” ise bağlantının sağlıklı olduğu anlamına gelmektedir. “ERROR” şeklinde bir cevap gelirse, bağlantılarda bir problem olduğu anlamına gelmektedir.

ATI: Bu komut, bağlı olan modülün adı ve sürüm bilgisini geri döndürmektedir. Şekil 3.6’da görüldüğü gibi, uygulama kapsamında SIM900 ailesinin R11.0 sürümü kullanılmıştır.

¹ Program “<http://www.chiark.greenend.org.uk/~sgtatham/putty>” adresinden ücretsiz olarak indirilebilir.

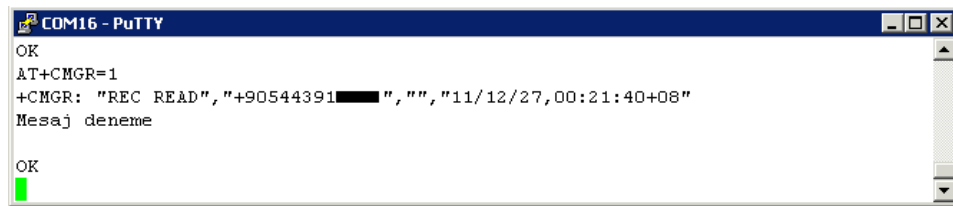


Şekil 3.6. SIM900 GSM modem bağlantısı.

3.1.1.1 GSM Modemden Mesajların Okunması

SIM900 modemden mesajların okunması iki şekilde gerçekleştirilmektedir. Bunlar, liste düzeyinde okuma veya mesajın hafızadaki numarasını (index) belirterek okuma şeklindedir. Uygulamanın bazı bölümlerinde ihtiyaca bağlı olarak, okuma işlemi listeleme şeklinde gerçekleştirilmiştir. Bazı bölümlerinde ise mesajın hafızadaki numarası yazılarak okuma işlemi gerçekleştirilmiştir.

Hafızadaki bir mesajın okunması için ilgili mesajın hafızada hangi numaralı mesaj olarak saklandığının bilinmesi gerekmektedir. Hafıza bilinmesiyle, “AT” komutlarından “AT+CMGR” kullanılarak ilgili mesaj okunabilir. Örneğin, hafızada 1 numaralı mesaj okunmak istendiğinde, konsol ekranında “AT+CMGR=1” yazılarak sonucu almak mümkündür. Sonuç Şekil 3.7’de görülmektedir. Gelen cevaptan, mesajın durumu, gönderen GSM numarası, zaman bilgisi ve mesajın içeriği okunabilmektedir.



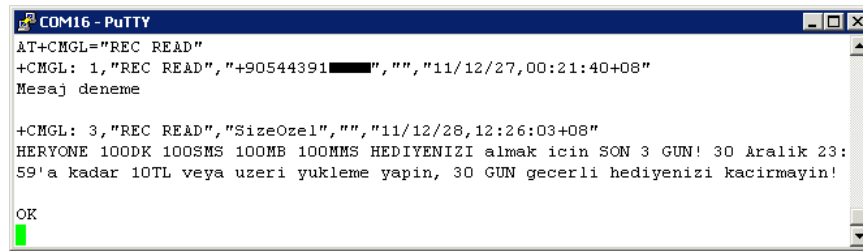
Şekil 3.7. Hafıza numarasına göre mesaj okunması.

Kayıtlı olan mesajların liste düzeyinde alınması için AT komutlarından “AT+CMGL” kullanılmaktadır. Liste düzeyinde mesajların okunması için hangi mesaj türlerin listelenmesi gerektiği belirlenmelidir. Bu türler Çizelge 3.1’de gösterilmiştir.

Çizelge 3.1. Mesaj gösterim türü kodları.

Komut	Açıklama
REC UNREAD	Alınan okunmamış mesajları listeler
REC READ	Alınan okunmuş mesajları listeler
STO UNSENT	Gönderilmemiş saklı mesajları listeler
STO SENT	Gönderilmiş saklı mesajları listeler
ALL	Tüm mesajları listeler

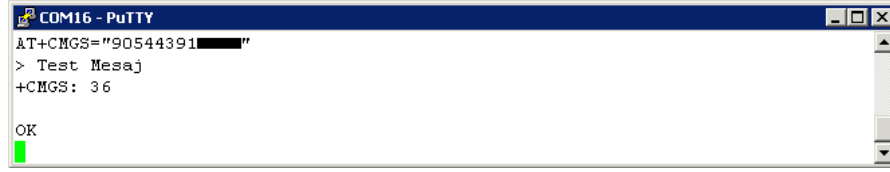
Modemde okunmuş olan mesajları listelemek ve bunlar üzerinde işlem yapılabilmesi için, öncelikle mesajların listelenmesi gerekmektedir. Bu işlem için AT komutlarından AT+CMGL="REC READ" kullanılmaktadır. Konsol ekranına bu komutu yazıldığında, Şekil 3.8’de görüldüğü gibi okunmuş mesajlar listelenecektir.

**Şekil 3.8.** Okunmuş mesajların listelenmesi.

3.1.1.2 GSM Modem Üzerinden Mesaj Gönderilmesi

“AT” komutlarından “AT+CMGS” mesaj gönderme işlemini gerçekleştirir. Aldığı parametreler mesajın gönderileceği numara, gönderilecek mesaj şeklindedir. İlk parametre olarak gönderilecek telefon numarası girilmelidir. Daha sonra, Enter tuşuna basarak gönderilecek metin girilmelidir. Metin giriş işlemi tamamlandığında “CTRL+Z” tuşlarına basarak gönderim işlemi tamamlanacaktır.

Gönderim işlemi başarılı bir şekilde tamamlanırsa, geriye cevap olarak mesajın hafızada kayıtlı olduğu numara döndürülür ve “OK” mesajı alınır. Şekil 3.9’da bir mesaj gönderimi gözükmektedir.

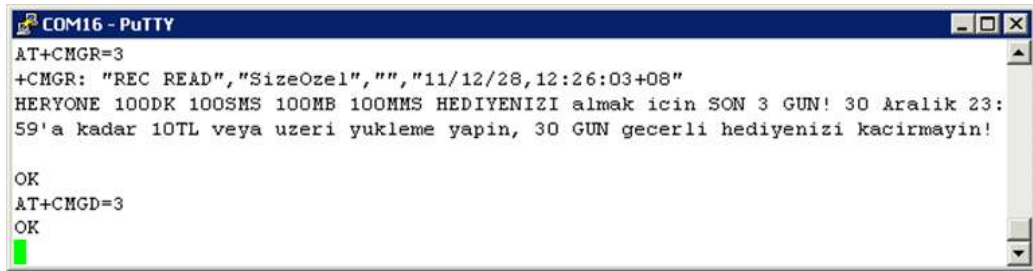


Şekil 3.9. Mesajların gönderilmesi.

Bu işlemler sırasında hatalı bir işlemin yapılması veya yapılandırma ayarlarında bir sorun olması durumunda “ERROR” mesajı cevap olarak döndürülmektedir. Geri döndürülen bu mesajlar sayesinde, geliştirilecek yazılımda meydana gelebilecek hatalara karşı önlem alınabilir.

3.1.1.3 Modem Üzerindeki Mesajların Silinmesi

Mesajların silinmesi, modemde okuma işleminde olduğu gibi tekil veya toplu olarak yapılabilir. Tek bir mesajın silinmesi için AT komutlarından “AT+CMGR” kullanılmaktadır. Parametre olarak, silinmesi istenen mesajın hafıza numarası yazılarak ilgili işlem gerçekleştirilmektedir. İşlemin başarılı bir şekilde gerçekleşmesi ile geriye “OK” mesajı döndürülmektedir. Şekil 3.10’da, modemde kayıtlı olan 3 numaralı mesajın okunduktan sonra “AT+CMGR” kodu ile silinmesi gösterilmiştir.



Şekil 3.10. Mesajın silinmesi.

Modem üzerinde, kısa mesajların okunması, gönderilmesi veya silinmesi gibi işlemleri, belirtilen zaman aralığında kontrol eden veya gerçekleştiren hizmetlere ihtiyacımız olacaktır. Bu hizmetler Windows tarafında farklı yöntemlerle yapılabilir. Bu uygulamada, Microsoft .NET yazılım geliştirme platformu üzerinde Windows servisleri geliştirilerek modem üzerinden mesaj okuma, gönderme ve silme işlemleri yapılmıştır.

3.1.2 Microsoft .NET Yazılım Geliştirme Platformu

.NET platformu, standartlar üzerine kurulmuş ortak uygulama geliştirme yapmak üzere Microsoft tarafından tasarlanmış ve 2002 yılında piyasaya sürülmüş ticari bir yazılım geliştirme platformudur. İçerisinde; çeşitli programlama dilleri, bu dillerde yazılım geliştirebilmek için gerekli olan arabirim ve kütüphaneler, veritabanı bağlantı katmanları, veritabanı sunucusu, ve benzeri bileşenler bulunmaktadır. Hizmet odaklı mimari, web uygulamaları, XML web servisleri ve smart istemci uygulama yazılımları ile ölçeklenebilir, güvenli ve güvenilir yazılım çözümleri için platform oluşturma ve temel yazılım mimarisi sağlamaktadır (Algan, 2010).

.NET, uygulama yazılımlarının çalışabilmesi için işletim sistemi üzerinde yeni bir katman sağlamaktan, yazılım geliştirme, dağıtma, veri işleme, web uygulamaları gibi bir dizi ihtiyaca standart çözümler getirir. “.NET Framework” mimarisinin farklı programlama dillerine destek sağlamak, ortak ara kod dili oluşturmak, farklı yazılımlardaki değişken tiplerini standartlaştırmak temel özellikleri olarak belirtilebilir (Algan, 2010).

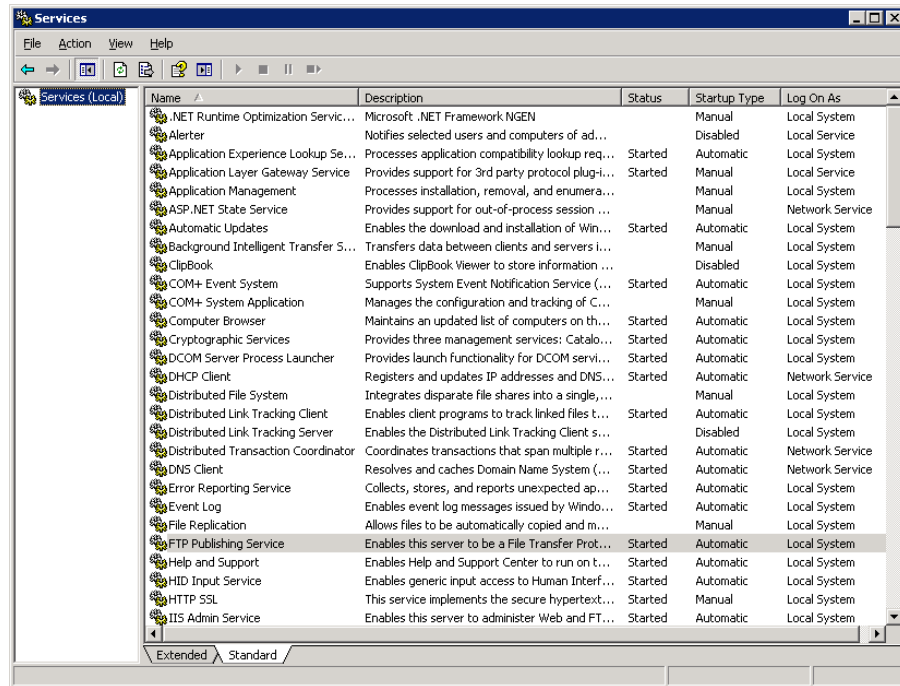
2002 yılından itibaren sürekli gelişen “.NET Framework” farklı sürümlere sahiptir. Bu çalışma yapıldığı sırada, son olarak 4.0 sürümü bulunduğundan uygulamalar bu sürüm üzerinden gerçekleştirilmiştir.

3.1.3 Microsoft .NET Windows Servisleri

Windows Servis Uygulaması (WSU), işletim sisteminin aktif olması ile birlikte arka planda çalışabilen ve Windows servis mimarisine göre geliştirilmiş uygulama türüdür (Türkoğlu, 2011). Windows servislerinin, kullanıcı tarafından çalıştırma zorunluluğuna sahip olmayan uygulamalardır. Bu tarz uygulamalar işletim sisteminin çalışması ile birlikte istenilen görevi yerine getirmeye başlar. Windows servisleri çalışma sırasında kullanıcıdan bir istek beklemediklerinden, kullanıcı etkileşimine ihtiyaç olmadan yapması gereken işlemleri gerçekleştirirler. İşleyiş bakımından Windows servisleri, belirli zamanda veya belirli bir zaman aralığında tekrarlayan işlemleri kullanıcı müdahalesi gerektirmeden arka planda gerçekleştirir. Bu çalışma kapsamındaki uygulamada belirli aralıklarda mesajları okuyan ve gönderen bir

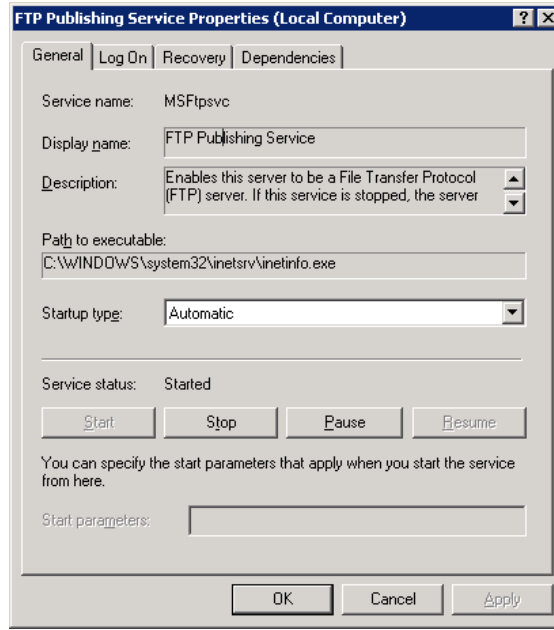
Windows servisi geliştirilmiştir.

Windows servislerinin çalıştırılması, durdurulması ve güvenlik anlamında tanımlanması gereken özelliklerin belirlenmesi gerekir. Bu servislerin yönetimi için Windows işletim sisteminin bir bileşeni olan Windows Servis Yöneticisi (WSY) uygulaması kullanılabilir. WSY, işletim sisteminde yüklü olan servislerin kontrolünün sağlanması ile görevli bir Windows uygulamasıdır. Windows Server 2003 işletim sistemine ait WSY Şekil 3.11’de görülmektedir.



Şekil 3.11. Windows servis yöneticisi.

Her bir Windows servisi için sistem tarafında tekil olarak ifade edilen bir servis ismi (Service Name) bulunur ve WSY altında ilgili servise ulaşmada kullanılır. Servisin görünüm ismi (Display Name) ise ilgili servisin hizmet adının daha açıklayıcı halde gösterimidir. Örnek olarak FTP (File Transfer Protocol ~ Dosya Aktarım Protokolü) Yayınlama Servisi kurulması ile birlikte WSY içinde FTP Publishing Service görünüm isminde gösterilirken servis ismi MSFtpsvc olarak gösterilir. Şekil 3.12’de FTP Yayınlama Servisinin özellikleri görülmektedir.



Şekil 3.12. FTP yayınlama servisinin özellikleri.

Windows servislerinin daha açıklayıcı bir tanımını isteğe bağlı olarak Description (Açıklama) özelliği altında verilebilir. Bu özellik sayesinde hizmetin yerine getirdiği görev ile ilgili daha açıklayıcı bilgiye sahip olunmuş olur. Windows servislerinin işletim sistemi veya kullanıcı tarafından müdahale edilerek başlatılması da mümkündür. Bu özellik Startup Type (Başlatma Tipi) altında yer alır. Bu özellik Çizelge 3.2’de belirtilen değerler yardımı ile tanımlanabilir.

Çizelge 3.2. Windows servis uygulamalarında başlama tipi ve açıklamaları.

Başlatma Tipi	Açıklama
Automatic (Otomatik)	Servisin işletim sistemi ile birlikte otomatik olarak başlaması gerektiğini belirtir.
Manuel (Elle)	Servisin işletim sistemi başladıktan sonra kullanıcı müdahalesiyle birlikte başlaması gerektiğini belirtir.
Disabled (Devre Dışı)	Servisin çalışmaması gerektiğini belirtir.

Her servisin, bir işletim sistemi seviyesinde kullanıcı tarafından ilişkilendirilmesi gereklidir. Servis hesabı (Log On) olarak adlandırılan bu kavramla her servis, çalıştığı bilgisayar kaynakları üzerinde, yetkileri ölçüsünde işlem yapabilmektedir. Bu hesaplardan “LocalSystem”, Windows hizmet hesabı tarafından

yönetilen özel bir kullanıcı hesabıdır. Bu hesap, herhangi bir kullanıcı adı ve parolası gerektirmeden servisin yönetici seviyesinde çalıştırılabilmesini sağlar (Türkoğlu, 2011).

Servis durumu bir servis için 3 farklı şekilde olabilir. Bunlar,

- **Started** (Başlatılmış): İlgili servisin yerine getireceği görevleri yapmak üzere çalıştığını belirtmektedir.
- **Paused** (Duraklatılmış): İlgili servisin geçici olarak duraklatıldığını belirtmektedir. Bu süre içerisinde görevler gerçekleştirilmez. Bu durumda servis çalışmaya devam ettirildiğinde kaldığı yerden işlemlerine devam eder.
- **Stopped** (Durdurulmuş): Servisin bu durumda olması, sonlandırıldığı anlamına gelir. Devam etmesi halinde servis yeniden başlatılır.

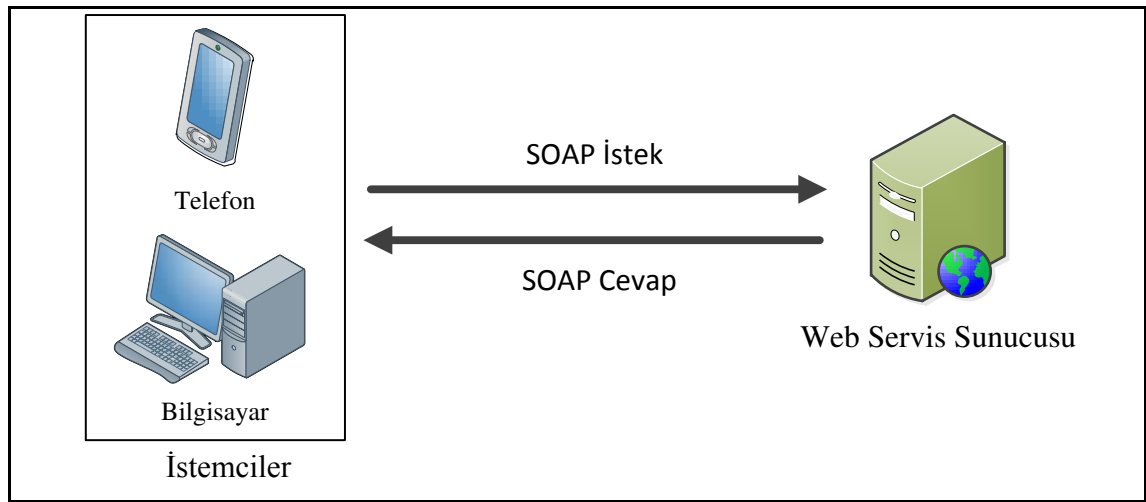
3.1.4 Microsoft .NET Web Servisleri

Web servisleri, birlikte çalışabilirlik esasları çerçevesinde bilgisayarlar arasında bir ağ üzerinde iletişimi ve uyumluluğu sağlamak için tasarlanmış yazılım bileşenleri olarak tanımlanabilir. (Nagarajan vd., 2006)

Web servisleri bir ağda bulunan bir bilgisayardaki uygulamayı temsil etmektedir. Bu uygulama diğer bilgisayarlar tarafından erişebilir bir durumda olmalıdır. Bu erişim sırasında karşılıklı anlaşabilmek için, bazı kurallara uyulmak zorundadır. Bu kurallara uymak neticesinde web servisleri kullanılabilir bir hale gelir. Bu kurallar W3C standartlar komitesi tarafından SOAP (Simple Object Access Protocol - Basit Nesne Erişim Antlaşması) adı altında birleştirilmiştir.

SOAP, XML (Extensible Markup Language ~ Genişletilebilir İşaretleme Dili) tabanlı dağıtık ortamlarda bilgi alışverişini diğer bir ifade ile bir web servisine erişimi sağlayan bir iletişim protokolüdür (Yüksek, 2011). Web servisleri, bağımsız XML ve XHTML üzerine kurulmuş ve W3C (World Wide Web Consortium ~ Dünya Çapında Ağ Birliği) standardı olarak geliştirilmiştir. Web servisleri her ne kadar HTML (Hyper Text Markup Language ~ Zengin Metin İşaret Dili) üzerinden XML kullanılarak haberleşse de web servisini kullanan istemcilerin işleyebileceği ve anlayabileceği bir hale getirmek gerekir.

İstemcilerin kullanacakları web servisindeki bilgileri önceden bilmeleri gerekir. WSDL (Web Services Description Language ~ Web Servisleri Tanımlama Dili) bu noktada devreye giren bir diğer önemli unsurdur. İstemci uygulamalar, kullanacakları web servisine ait bilgileri WSDL yardımıyla önceden tedarik ederler. Bu istemcinin web servisi üzerindeki bir web metodunun varlığından haberdar olması, onu nasıl kullanacağını bilmesi anlamına gelmektedir. İstemci ve servisin bulunduğu sunucu arasındaki iletişim görüntüsü Şekil 3.13’de gösterilmiştir.



Şekil 3.13. İstemciler ve web servis sunucusu arasındaki iletişim.

Günümüzde, web servislerinin e-ticaret, SMS ve WAP (Wireless Application Protocol ~ Kablosuz uygulama Protokolü) teknolojilerinde kullanımının yaygınlaşması ön plana çıkmıştır (Yüksek, 2011). Böylelikle web servisleri ile kurumlar arası uygulamalar birbirleriyle bütünleşik şekilde çalışarak servis geliştiricilerin işlemlerini kolaylaştırmaktadır.

3.2 Windows Servislerinin Geliştirilmesi

Oluşturulmuş olan Windows servisi, bilgisayarın çalışma süresince GSM modem ile haberleşmektedir. Bu haberleşme, mesaj okuma, mesaj gönderme ve işlem yapılmış mesajları silme şeklindedir. Modem ile haberleşme sırasında bazı işlemleri belirli süre içerisinde gerçekleştirmesi ve tekrar etmesi gerekecektir. Zamanı kontrol etmek için “.NET” kütüphanesi altında yer alan “Timer”, modem ile haberleşmek için “SerialPort”, mesajların düzenlenmesinde ise “Regex” sınıfları kullanılmıştır. Regex sınıfı, regular expression (düzenli ifadeler) ifadelerini kullanarak metin dizisinde işlem yapılmasına

olanak sağlar.

Düzenli ifadeler, özel bir metin dizisini başka bir metin dizisi içerisinde konuşlandırmak için tasarlanan ve biçimsel bir programlama dili içerisinde kullanılan özel ifadelere denmektedir. Bazı özellikleri ile tanımlanabilen bir metin tipini, kaynak metin içerisinde geçtiği tüm konumları kolayca bulunabilmesini sağlar. Örneğin; uzun bir metin dosyası içerisinde geçen tüm eposta adreslerini tespit etme işlemi, düzenli ifadeler ile bir satırlık bir kod yazılarak sağlanabilmektedir.

Düzenli ifadelerin geçmişi, 1951’de S.C.Kleene tarafından yayınlanan, “Representation of events in nerve nets and finite automata ~ Sinir ağları ve sonlu otomasyonlarda olayların temsili” isimli makaleye kadar uzanmaktadır (Kleene, 1956). Ancak, düzenli ifadelerin uygulamada ilk defa kullanılması ve Unix metin editörlerinde yaygınlaşması 1960’lı yılları bulmuştur. Günümüzde, sadece editörlerde değil, hemen hemen tüm programlama dillerinde desteklenmekte ve XML sorgulama, İnternet trafik analizi, sistem kayıt dosyalarından rapor oluşturma vb. gibi çok farklı alanlarda kullanılmaktadır (Bille ve Thorup, 2009).

Gelen mesajların telefon numarası, tarihi ve mesajın içeriğinin alınmasında kullanılan “Regex” sınıfının tanımı Şekil 3.14’de gösterilmiştir. Bu tanım, Şekil 3.7’de gösterilen bir mesajın okunması işleminden sonra gelen bilgi üzerinde “+” ile başlayan 12 haneli telefon numarası, tarih bilgisi ve mesajın içeriği alınır

```
Regex( "(\\\"\\\\+\\\\d{ 12}\\\\\"), (\\\"\\\\\"), (\\\". *\\\\\")\\\\r\\\\n(. *\\\\\" );
```

Şekil 3.14. Regex tanımı.

3.2.1 Modem ile Haberleşen Windows Servisinin Oluşturulması

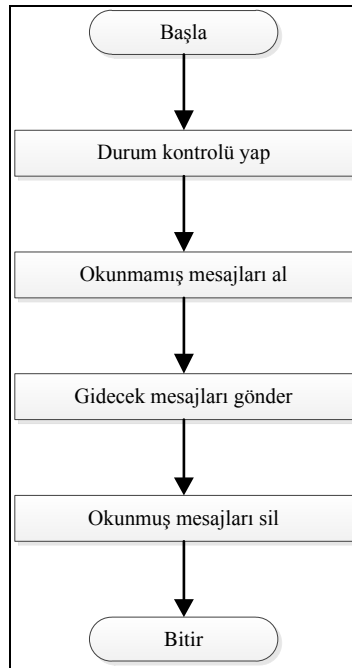
Oluşturulan bu servis, GSM modemle belirli zaman aralıklarında haberleşen ve gerekli işlemleri yerine getiren bir yapıya sahiptir. Bu haberleşme aralığı bir mesajın gönderimi, alınması ve silinmesi işlemlerinin yapılmasına göre, modemle haberleşme hızına göre, modemin özelliklerine göre değişmektedir. Bu uygulamada, kararlı bir şekilde gelen mesajların alınması ve veritabanına kaydedilmesi, gönderilecek mesaj varsa bunun alınması ve gönderilmesi son olarak modem üzerinde işlem yapılmış mesajların silinmesi işlemi beş saniyelik bir süreye düşürülmüştür. Bu süre ilave GSM

modem ya da iletişim hızı yüksek olan bir modem eklenmesiyle düşürülebilir.

Bu servis;

- Durum kontrolünü
- Okunmamış mesajların alınması
- Gidecek mesajların yollanmasını
- Okunmuş mesajların modem üzerinden silinmesi gibi işlemleri gerçekleştirmektedir. Metodun iş akış şeması Şekil 3.15’de gösterilmiştir.

Seri port arabiriminden bir cihaza bağlantıda .NET Framework kütüphanesinin System.IO.Ports isim alanının altında bulunan SerialPort sınıfı kullanılabilir. Uygulamada bu sınıf kullanılarak modem ile bağlantı gerçekleştirilmiştir.



Şekil 3.15. Modem haberleşme servisi akış şeması.

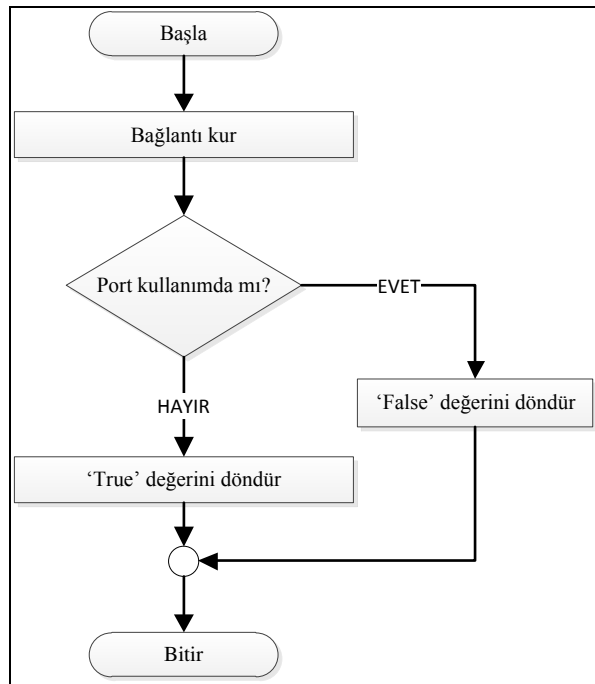
SerialPort (PortAdı, Hızı, Eşlik biti, Veri biti, Durma Biti) özelliği kullanılarak bir sınıf oluşturulabilir. Seri port olarak bilgisayarın modemle haberleşmesinde COM16 portu, hız olarak ise 115200 bit/sn verilmiştir. Veri aktarım sırasında eşlik biti kullanılmamıştır ve verilerin aktarımı 8 bit olarak tanımlanmıştır. Tekrar iletişim kurulabilmesi için gönderilen her baytın sonunda tek bir durma biti tanımlanmıştır. Bu tanımlama uygulama “Sim900SP” adında verilmiş olup Şekil 3.16’da gösterilmiştir.

```
SerialPort Sim900SP = new SerialPort("COM16", 115200, Parity.None, 8, StopBits.One);
```

Şekil 3.16. Seri port bağlantı tanımı.

Modemle bilgisayar, seri port üzerinden COM16 üzerinden haberleşmektedir. Kısa mesaj sunucusu çalışmaya başladığı zaman, birden fazla servis modem üzerinde işlem yapmaktadır. Seri port arabirimi bir uygulama tarafından kullanılırken, aynı portu farklı bir servisin kullanılmasına izin verilmemektedir. Bu tarz durumlarda portun farklı bir uygulama tarafından kullanımda olduğu hatası verilmektedir. Bunun önüne geçmek için, modem üzerinde yapılan işlemler sırasında önce kullanılacak olan portun farklı bir uygulama tarafından kullanılıp kullanılmadığını kontrol etmek gerekmektedir.

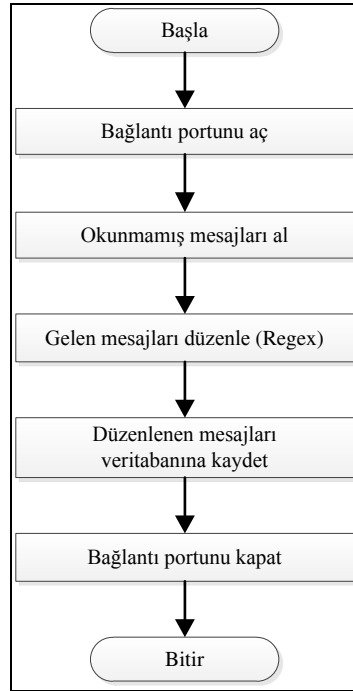
Uygulamada çakışmaları önlemek için, seri portun kullanımda olup olmadığını kontrol eden bir metot yazılmıştır. Bu metot geriye boolean türünden değer döndürmektedir. Port kullanımda ise “false”, kullanımda değilse “true” değerini geriye döndürmektedir. İş akış şeması Şekil 3.17’de gösterilmiştir.



Şekil 3.17. Seri port durum kontrolü.

Durum kontrolünden sonraki basamak, modemde bulunan okunmamış mesajların alınması işlemidir. Bu işlem kendi içerisinde iş gruplarına ayrılmaktadır. Bu iş akışı Şekil 3.18’de gösterilmiştir. Durum kontrolünden gelen ifadenin değerinin

“true” olması ile başlayan bu iş akışı, ilk olarak Sim900SP portunun açılması işlemini gerçekleştirmektedir. Daha sonra modemden okunmamış olan mesajların alınması ile devam etmektedir. Bu işlemde tek mesaj geri dönebileceği gibi, birden fazla mesaj da geri dönebilir. Okunmamış mesajlar alındıktan sonra, “Gönderen GSM numarası”, “Tarih” ve “gelen mesaj” alanlarının ayrılması için mesajlar üzerinde düzenleme işlemi yapılmaktadır.



Şekil 3.18. Okunmamış mesajların alınması.

Mesajlar üzerinde düzenleme regex sınıfı kullanılarak yapılmıştır. Mesajların düzenlenmesinin ardından, gelen mesajlar veritabanında “GelenKisaMesaj” tablosuna kayıt edilmektedir. GelenKisaMesaj tablosu Şekil 3.19’da gösterilmiştir.

GelenKisaMesaj		
	Column Name	Condensed Type
🔑	No	int
	Telefon	char(12)
	Mesaj	nvarchar(300)
	Zaman	datetime
	IslemYapildiMi	bit

Şekil 3.19. Gelen kısa mesaj tablosu.

Tüm bu işlemler tamamlandıktan sonra Sim900SP bağlantı portu kapatılmaktadır.

Mesajlar üzerinde düzenleme işlemi yapıldıktan sonra, mesajlar üzerinde yorumlama ve diğer işlemlerin yapılabilmesi için veritabanına kayıt işlemi gerçekleştirilmektedir. Veritabanına kayıt sırasında, telefon numarası ve gönderilen mesaj alanları da kaydedilir.

Mesajların ilgili kişilere gönderilmesi işlemi, veritabanında bulunan “GidenKisaMesajlar” tablosu dikkate alınarak gerçekleştirilmektedir. Tablo içerisinde bulunan alanlar ve bu alanların özellikleri Şekil 3.20’de gösterilmiştir.

GidenKisaMesaj		
	Column Name	Condensed ...
	No	int
	Telefon	char(12)
	Mesaj	nvarchar(300)
	Zaman	datetime
	IslemYapildiMi	bit

Şekil 3.20. Giden kısa mesaj tablosu.

Okunmamış mesajların alınması ile ilgili oluşturulan iş akışında, gidecek olan mesaj varsa bunun tablodan alınması ve daha sonra ilgili GSM numarasına gönderilmesi için işlemler gerçekleştirilir. Bu iş akışlarından gidecek olan mesajın alınması işlemi veritabanında oluşturulan bir prosedür yardımıyla gerçekleştirilmektedir. Bu prosedür “IslemYapidilMi” alanında false değerine sahip olan bir mesajı alır ve geriye No,Telefon, Mesaj alanlarındaki değeri döndürür.

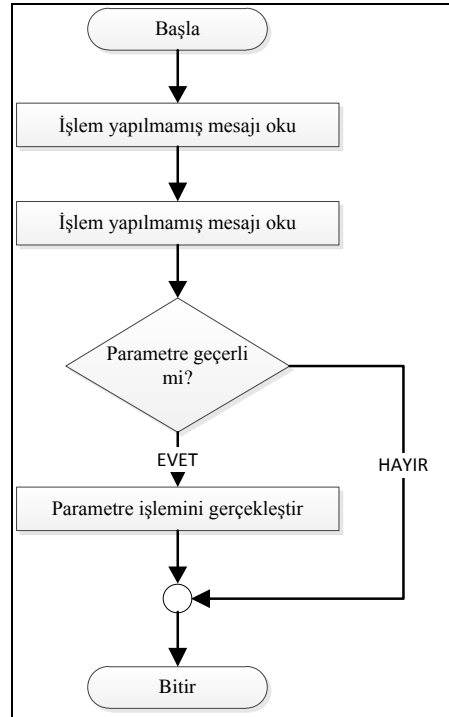
Gidecek olan mesaj olması durumunda, Sim900SP seri portu üzerinden modem ile bağlantı kurulması sağlanacaktır. Bu işlem için öncelikle seri portun farklı bir uygulama tarafından kullanılıp kullanılmadığına bakılır. Bağlantı portu kullanılmıyorsa bir sonraki komutun gönderilmesi işlemine geçilir. Modem ile bağlantı kurularak AT komut kümesinden “AT+CMGS” kullanılarak ilgili mesaj gönderilme işlemi gerçekleştirilir. Mesaj gönderme sırasında veritabanından gelen bilginin telefon ve mesaj alanları kullanılır.

Mesaj gönderme işleminin başarılı bir şekilde tamamlanmasının ardından gönderilen mesaja ait “No” alan bilgisi kullanılarak, hangi mesaj üzerinde işlem yapıldığı bilgisi, veritabanında “GidenKısaMesaj” tablosuna eklenir. Veritabanı üzerinde yapılan bu işlemler bir prosedür üzerinden gerçekleştirilmiştir.

Modem üzerinde okunmuş olan mesajlar veya kaydedilmiş olan mesajlar SIM kartının hafıza alanında saklanır. Bu alan, mesajların okunduktan sonra silinmemesi ile dolar ve bu süreçten sonra mesajlar üzerinde bir işlem yapılamaz. İşlem yapılmış mesajların, belirli aralıklarla SIM kart üzerinden silinmesi gereklidir. Bu aralık bir süreye bağlı olabileceği gibi, SIM kartında kayıtlı son mesajın numarasına göre doluluk oranı tespit edilerekte yapılabilir.

3.2.2 Mesajları Yorumlayan Windows Servisinin Oluşturulması

Uygulamada mesajların yorumlanması, bir Windows servisi üzerinden gerçekleşmektedir. Bu hizmet modeme gelen mesajların okunması ile işe başlar devamında mesajın parametresi (mesajın ilk kelimesi) alınarak geçerli bir parametre olup olmadığına bakılır. Mesaj yorumlamasında kullanılan parametreler “3.2.2.1 Uygulamada Kullanılan Kısa Mesaj Parametreleri” başlığı altında anlatılmıştır. Parametreye göre gerçekleştirilmesi gereken işler, Windows servisi üzerinden gerçekleştirilir ve kullanıcı bilgilendirilmesi gerekiyorsa bilgilendirilir. Bu sürecin iş akış şeması Şekil 3.21’ de gösterilmiştir.



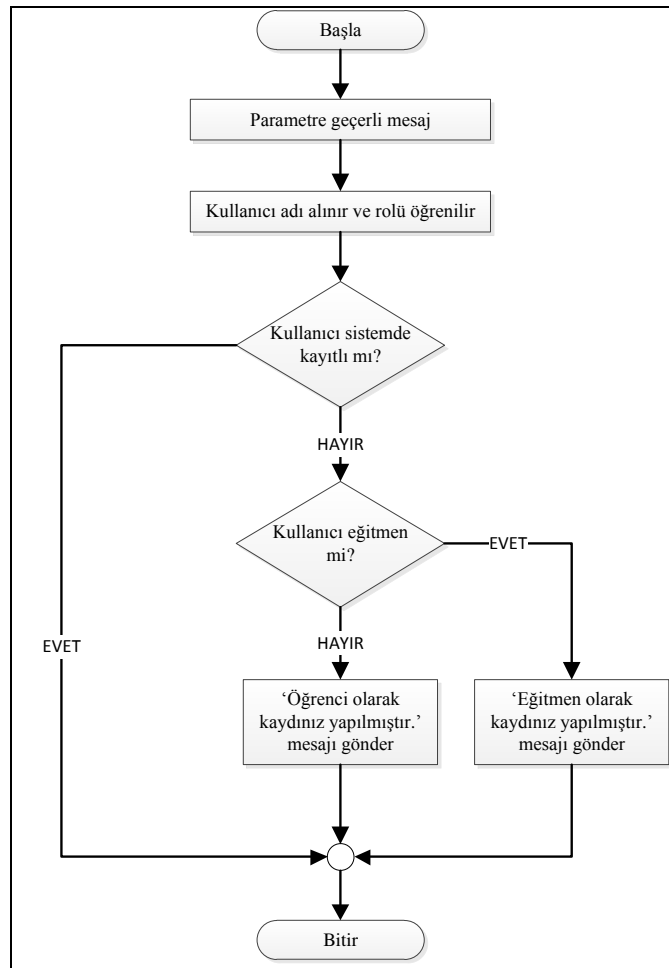
Şekil 3.21. Mesaj yorumlama iş akışı.

Bu servis, gelen mesajların veritabanına kaydedilmesinden sonra, mesajlar üzerinde işlem yapılmasını sağlamaktadır. İşlem yapılmamış mesajları okuma iş tanımı veritabanı üzerinde bulunan “GelenKisaMesaj” tablosundaki “IslemYapildiMi” alanının “false” olan değerlerine göre çağrılır. İşlem yapılmamış mesajın içeriği öncelikle “split()” metodu kullanılarak boşluk karakterine göre parçalanır ve bir dizi değişkenine aktarılır. Parçalara ayrılan kısa mesaj içeriği, “MesajIcerigi” değişkenine aktarıldıktan sonra istek parametresini öğrenmek için dizinin ilk elemanı çağrılır. Çağırılan bu birinci eleman mesajın yorumlanması için parametre niteliğini taşıyacaktır. Uygulamada kullanılan Windows Servisine ait kodlar EK 1’de “Windows Servisleri Uygulama Kodları” başlığı altında verilmiştir.

3.2.2.1 Uygulamada Kullanılan Kısa Mesaj Parametreleri

Kullanıcılar, SMS üzerinden sistemde işlem gerçekleştirmek isterse, sistem üzerinde tanımlanmış olan bazı komutları, mesaj içerisinde parametre olarak kullanabilmektedir. SMS ile sisteme gelen mesaj içeriğindeki parametreler modemin bağlı olduğu sunucu tarafından yorumlanarak, tanımlanmış görevlerden uygun olanı gerçekleştirmektedir. SMS üzerinden sisteme gönderilebilecek olan parametreler şunlardır;

Kayıt parametresi: Uygulamanın kullanılması için öncelikle kullanacak olan kişilerin kayıt işlemini gerçekleştirmeleri gerekmektedir. Kayıt işlemi için, kullanıcının cep telefonundan “kayıt” kelimesiyle başlayan ve devamında belirteceği kullanıcı adı ile sisteme kısa mesaj yollamalıdır. Buradaki sistem kavramı GSM modemde takılı olan SIM kartın numarasıdır. Mesaj, modem üzerinden sunucuya ulaştıktan sonraki süreç ile ilgili iş akışı Şekil 3.22’de belirtilmiştir. Gönderilen kullanıcı adı, öğrenci bilgi sisteminden rol bilgisi alınır. Öğrenci bilgi sisteminden kullanıcının rolünün; eğitmen gelmesi durumunda “Eğitmen olarak kaydınız yapılmıştır.”, öğrenci gelmesi durumunda “Öğrenci olarak kaydınız yapılmıştır.” bilgilendirme mesajı yollar.

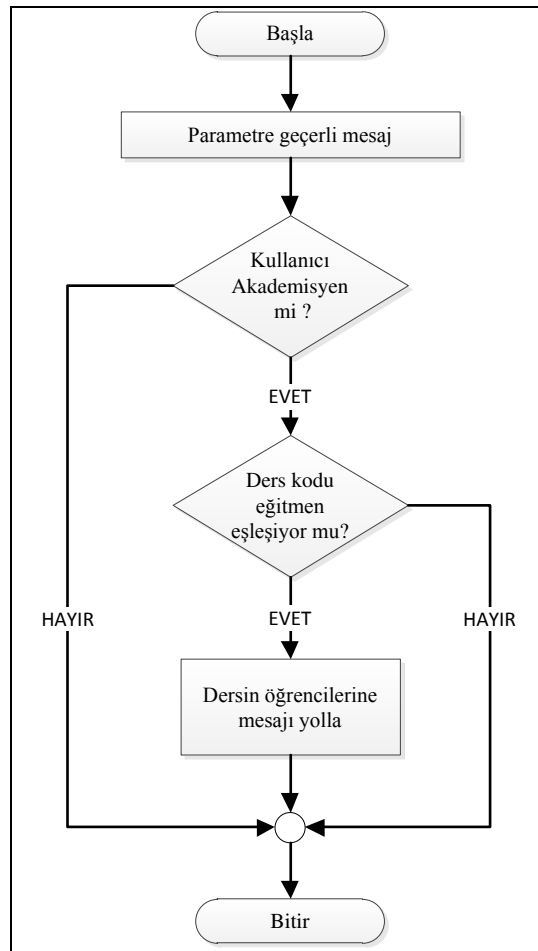


Şekil 3.22. Kayıt parametresi iş akışı.

Dersbilgi parametresi: Eğitimi tarafından, derse kayıtlı olan öğrencilere ders hakkında bilgilendirme mesajı yollamak için kullanılan parametredir. Eğitimi tarafından, SMS ile sisteme gönderilen mesajın derse kayıtlı olan tüm öğrencilere dağıtımını sunucu tarafından sağlanmaktadır.

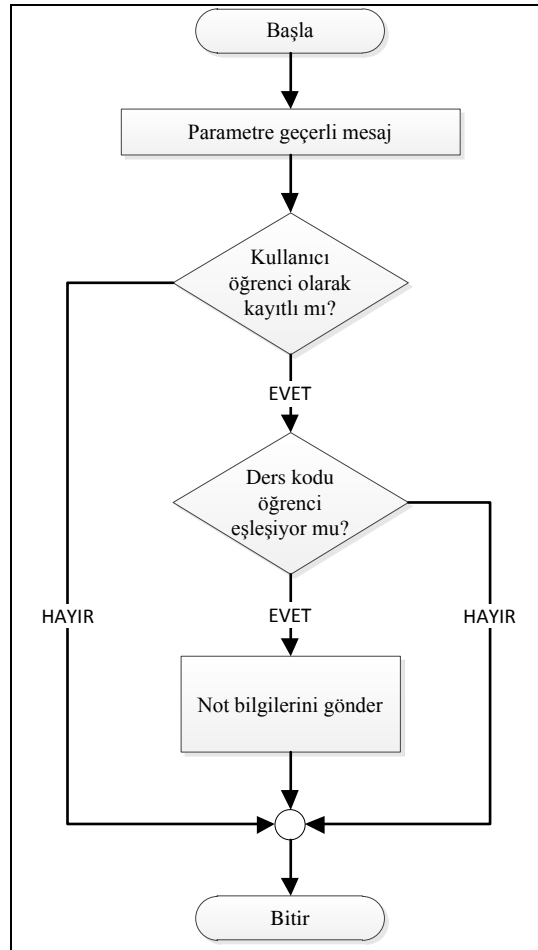
Örneğin, bir der bilgilendirme mesajı, “*Dersbilgi 144 Dersimiz bugün Lab1’de yapılacaktır.*” şeklindedir. Bu mesajda “*split()*” metodu kullanılarak boşluk karakterine göre parçalanarak bir diziye aktarılır. Daha sora parçalanmış bu bilgi dizisinin ilk elemanı parametre olarak alınır. Bir sonraki dizi elemanı, ders kodu olarak yorumlanarak sistemden karşılığına bakılır ve ilgili öğrenci gurubuna dizinin geri kalan elemanları birleştirilerek SMS ile gönderilir.

Dersbilgi parametresinin işlenmesinden önce, mesajı gönderen kişinin sistemde eğitimci olarak kayıtlı olup olmadığına bakılır. Eğitimci ise, gönderdiği ders kodunun öğrenci bilgi sisteminde kendisi üzerinde gözüküp gözükmediğine bakılarak bir sonraki kontrol işlemi gerçekleştirilir. Bu işlemde başarılı bir şekilde geçilirse ders koduna göre dersi alan öğrencilere bilgi mesajı gönderilir. Bu süreç ile ilgili iş akışı Şekil 3.23’de belirtilmiştir.



Şekil 3.23. Ders bilgi parametresi iş akışı.

Not parametresi: Öğrenciler tarafından kullanılmaktadır. Kullanıcı parametre olarak belirtmiş olduğu bu mesaj tanımında ders kodunu yazarak sistemde ders koduna ait not bilgilendirmesi bilgi mesajı olarak geri döndürülür. Ders kodu öğrenci tanımı eşleştirme kontrolü yapılarak dersin öğrenci tarafından alınıp alınmadığı kontrol edilir. Eşleşmede bir sıkıntı yoksa parametre görevini tamamlayarak kullanıcıyı bilgilendirir. Örneğin, bir dersin notlarının öğrenilmesi için “*not bil102*” şeklinde oluşturulan mesaj sisteme gönderilir. Mesajı gönderen öğrenci ders bilgilerinde “*bil102*” görünmesi durumunda aldığı not bilgileri öğrencinin cep telefonuna gönderilir. Bu iş tanımı ile ilgili iş akışı Şekil 3.24’de belirtilmiştir.



Şekil 3.24. Not parametresi iş akışı.

Msj parametresi: Bu parametre öğrenci ve öğretmen iletişimi için kullanılır. Öğrenci kendisinin kayıtlı olduğu herhangi bir dersin eğitimcisi ile iletişim haline geçmek isterse bu parametreyi kullanarak sisteme mesaj gönderir. Örnek bir mesaj gönderimi “*msj yusuf.mustu Hocam yarın ders var mı?*” şeklinde yazılabilir. Bu

mesajda, msj parametresinden sonra gelen bilginin gönderileceği eğitmenin kullanıcı adına karşılık gelmektedir. Parametre iş tanımı gerçekleşirken akademisyene öğrencinin gönderdiği mesaj iletilir ve aynı zamanda web ortamından veya Android uygulamasından görüntülenmesi için veritabanında bulunan “GorusmeIstek” tablosuna mesaj kaydedilir. “GorusmeIstek” tablosunun yapısı Şekil 3.25’de gösterilmiştir.

GorusmeIstek		
	Column Name	Condensed Type
?	No	int
	IstekYapan	char(11)
	IstekYapilan	char(11)
	IstekMesaj	nchar(200)
	CevapMesaj	nchar(200)
	IstekIslendiMi	bit
	CevapIslendiMi	bit
	IstekTarihi	datetime
	CevapTarihi	datetime

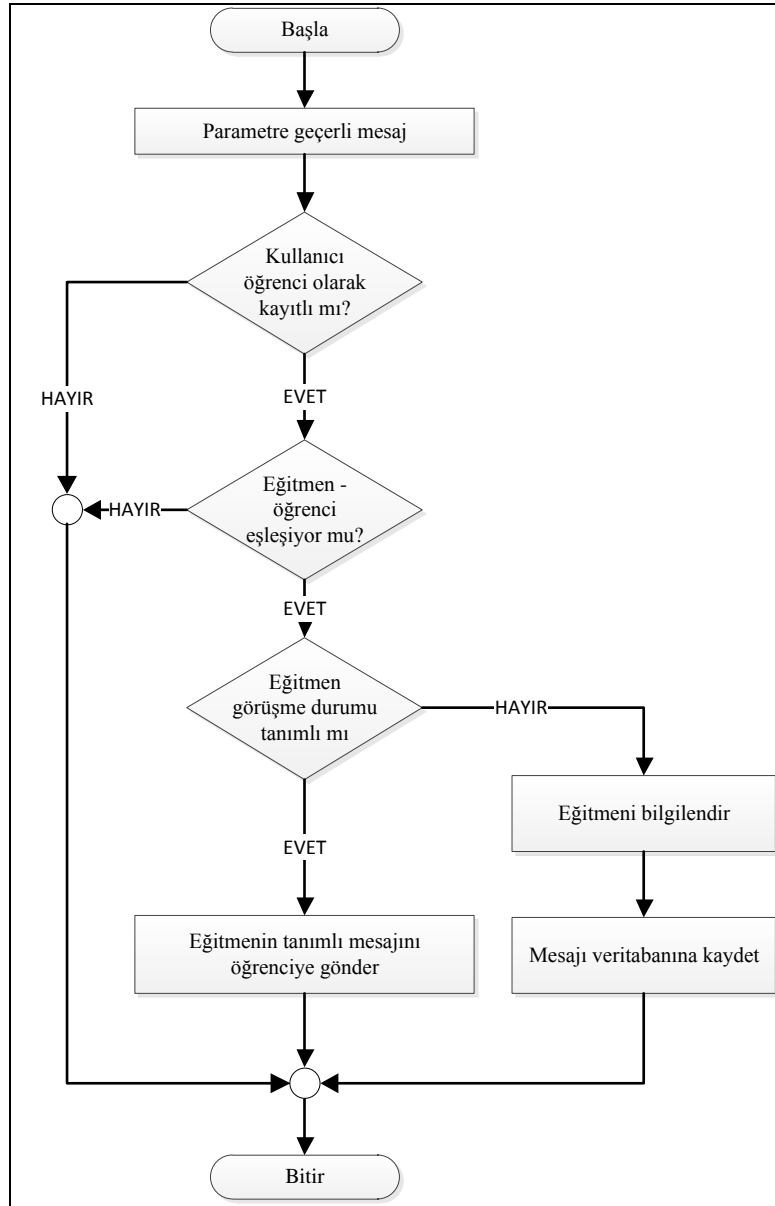
Şekil 3.25. Görüşme istek tablosu.

Öğrencilerin akademisyenlerle görüşme isteğinin olabileceği düşünülerek uygun tablo yapısı oluşturulmuştur. Akademisyenlerin bazı zaman aralıklarında görüşmeye uygun olmayacakları düşünülerek web arabiriminden uygun olmayan zamanlarının belirlenmesi sağlanmıştır. Bu sayede bu zaman dilimi içerisinde akademisyenle görüşme isteğinde bulunan öğrenciler, sistemden akademisyenin kaydetmiş olduğu uyarı mesajını otomatik olarak almış olacaklardır. Bu isteğin saklanması ve yorumlanması için veritabanında ek bir tabloya ihtiyaç duyulmuştur. Bu tablonun yapısı Şekil 3.26’da gösterilmiştir.

GorusmeDurumu		
	Column Name	Condensed Type
?	No	int
	TcKimlik	char(11)
	BaslangicTarih	datetime
	BitisTarih	datetime
	Mesaj	nvarchar(200)

Şekil 3.26. Görüşme durumu tablosu.

Akademisyenlerin web arabirimi kullanarak durumlarını ve kendisine ait gelen mesajların yönetilmesini sağladıkları yapı “3.3 Web Arabirimi Geliştirilmesi” başlığı altında anlatılmıştır. “Msj” parametresinin iş akışı Şekil 3.27’de gösterilmiştir.



Şekil 3.27. Msj parametresi iş akışı.

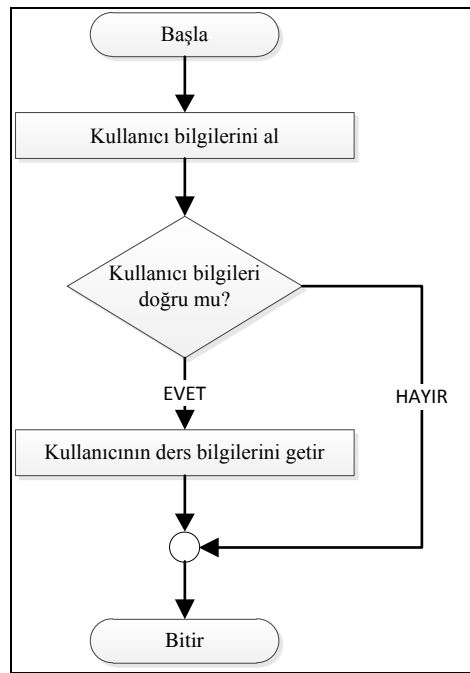
3.3 Web Arabirimi Geliştirilmesi

Eğitimcilerin, sınıf hakkında daha detaylı bilgiye sahip olmak, kendilerine gelen mesajların geçmişini görmek ve not bilgilendirmenin yapılmasını sağlamak amacıyla SMS sistemi ile bütünleşik olarak çalışan web uygulaması geliştirilmiştir. Bu sayede SMS aracılığıyla yapılamayan işlemlerin bir web sayfası üzerinden gerçekleştirilmesi sağlanmıştır.

Web ortamından gerçekleştirilen işlemlerden ilki, eğitimcinin kendisine ait ders tanımlamalarının olduğu ve o derse kayıtlı öğrencilerin listelendiği sınıfa bu sayfa

üzerinden toplu mesaj gönderiminin sağlandığı “*Sınıf Bilgilendirme*” ekranıdır.

Akademisyenlere açık olan bu sayfaya giriş için gerekli olan kullanıcı oturum açma bilgileri sunucu üzerinde oluşturulan bir kimlik yönetimi sistemi sayesinde sağlanmıştır. Bu kimlik yönetimi sistemi için, ASP.NET 2.0 ile birlikte gelen üyelik (membership) yapısı kullanılmıştır. Web arabiriminde başarılı bir şekilde oturum açılmasının ardından akademisyene ait ders bilgilerinin öğrenci bilgi sisteminden alınması gerçekleştirilmiştir. Bu iş akışı Şekil 3.28’de gösterilmiştir.



Şekil 3.28. Kullanıcı doğrulaması ve ders bilgilerinin getirilmesi.

3.3.1 Sınıf Bilgilendirme Ekranı

Sınıf bilgilendirme ekranında akademisyenin başarılı bir şekilde oturum açmasının ardından döneme ait ders bilgilerinin veritabanından çekilmesi ile birlikte ekrana getirilir. Bu aşamadan sonra akademisyen istediği sınıfı seçerek, iletmek istediği mesajı yazar ve sınıfta kayıtlı gözükten öğrencilere gönderilir. Şekil 3.29’ da bu ekran görüntüsüne yer verilmiştir. Sınıfın seçilmesi ile gelen öğrenci bilgileri sisteme daha önceden kayıt olmuş öğrenci bilgileridir.

Veritabanına eklenen mesaj, sonrasında sürekli olarak çalışan “SMSModemHaberles” Windows servisi sayesinde “GidenMesaj” tablosunda işlem

yapılmamış mesajlara bakılır ve gerekli işlemler yapılarak ilgili öğrenci bilgilendirilir.

Üzerinizde kayıtlı gözükten dersler					Mesaj Alıcı Sayısı: 5		
GunAdi	Ders	Ogr	B. Saat	No	Ad	Soyad	
Sec Perşembe	BTP203 Veritabanı Yönetim Sistemleri II	(I.Ö.)	13:00:00	10810	Asım		
Sec Perşembe	BTP209 Bilgisayar Ağ Sistemleri	(I.Ö.)	17:00:00	19288	Zeki		
Sec Perşembe	BTP203 Veritabanı Yönetim Sistemleri II	(N.Ö.)	09:00:00	18903	Ramazan		
				20095	Mine		
				18709	Canan		

Mesajınız:

[Mesajı Alıcılara Gönder](#)

Yusuf MUŞTU 2011 Tez Çalışması

Şekil 3.29. Sınıf bilgilendirme web sayfası.

3.3.2 Akademisyen Not Bilgilendirme Ekranı

Öğrenciler notlarını öğrenmek istedikleri zaman veya not girişi sırasında notlarını sistem tarafından kendilerine otomatik gönderilmesini sağlamak için geliştirilen bu ekranda, akademisyenin not girişini yaptıktan sonra ilgili öğrenciye not bilgisi kısa mesaj ile yollanmaktadır.

Sınıfa ait not girişinin bitmesi ile birlikte akademisyenin toplu bir şekilde öğrencilere aldığı not ve sınıf ortalamasının ne olduğu bilgisi de gönderilebilmektedir. Şekil 3.30’ da bu ekran görüntüsüne yer verilmiştir.

Şekil 3.30. Not bilgilendirme web sayfası.

3.3.3 Akademisyen Görüşme Yönetim Ekranı

Akademisyen ve öğrenci iletişiminin yönetildiği bu ekranda, öğrencinin akademisyene mesaj göndermesiyle birlikte veritabanında bulunan GorusmeIstek tablosuna mesajı yazılır. Akademisyen tarafından görüşme durumunda tanımlanmış mesaj varsa, otomatik olarak kullanıcıya gönderilir. Şekil 3.31’de akademisyen görüşme yönetimi ekranı gösterilmiştir.

			IstekTarihi	IstekMesaj	CevapMesaj
Mesajı Gönder	21	Yusuf MUŞTU	11/10/2011 9:30:11 PM	Hocam merhaba, yarın ders var mı?	Evet. Yarın ders var.
Mesajı Gönder	23	Yusuf MUŞTU	11/11/2011 10:59:59 AM	Hocam, ben Ahmet yarın görüşebilir miyiz?	Yarın saat 13:00 gibi olur
Mesajı Gönder	24	Yusuf MUŞTU	12/5/2011 2:17:58 PM	Web sayfası çalışıyor mu?	Çalışıyor

Şekil 3.31. Akademisyen görüşme yönetimi ekranı.

Otomatik mesaj alanı belirli bir zaman dilimi seçildikten sonra Şekil 3.31’de gösterilen “Mesaj” alanı doldurularak bu zaman diliminde akademisyenden istekte bulunan öğrenci ya da öğrencilere otomatik olarak yollanır. Akademisyene bu süre içerisinde kısa mesaj ile bilgilendirme yapılmaz. Daha sonradan gelen mesajların kontrolünü yapmak ve istenirse cevaplamak için bu mesajlar kaydedilir. Akademisyen bu web sayfasına bağlanarak kendisine gelen mesajlara cevap verebilir.

4. ANDROID UYGULAMASININ GELİŞTİRİLMESİ

Android uygulamaları, farklı yazılım geliştirme araçları kullanılarak geliştirilebilir. Ücretsiz yazılımlardan, Eclipse veya Netbeans kullanılarak yazılım geliştirme sağlanabilir. Microsoft .NET yazılım geliştirme araçları ile Android uygulaması geliştirilmek istenirse, ücretli olan “Monodroid” Visual Studio bileşeni kullanılabilir. Bu bileşen sayesinde Visual Studio ortamında Android uygulamaları geliştirilebilir (Taç, 2011).

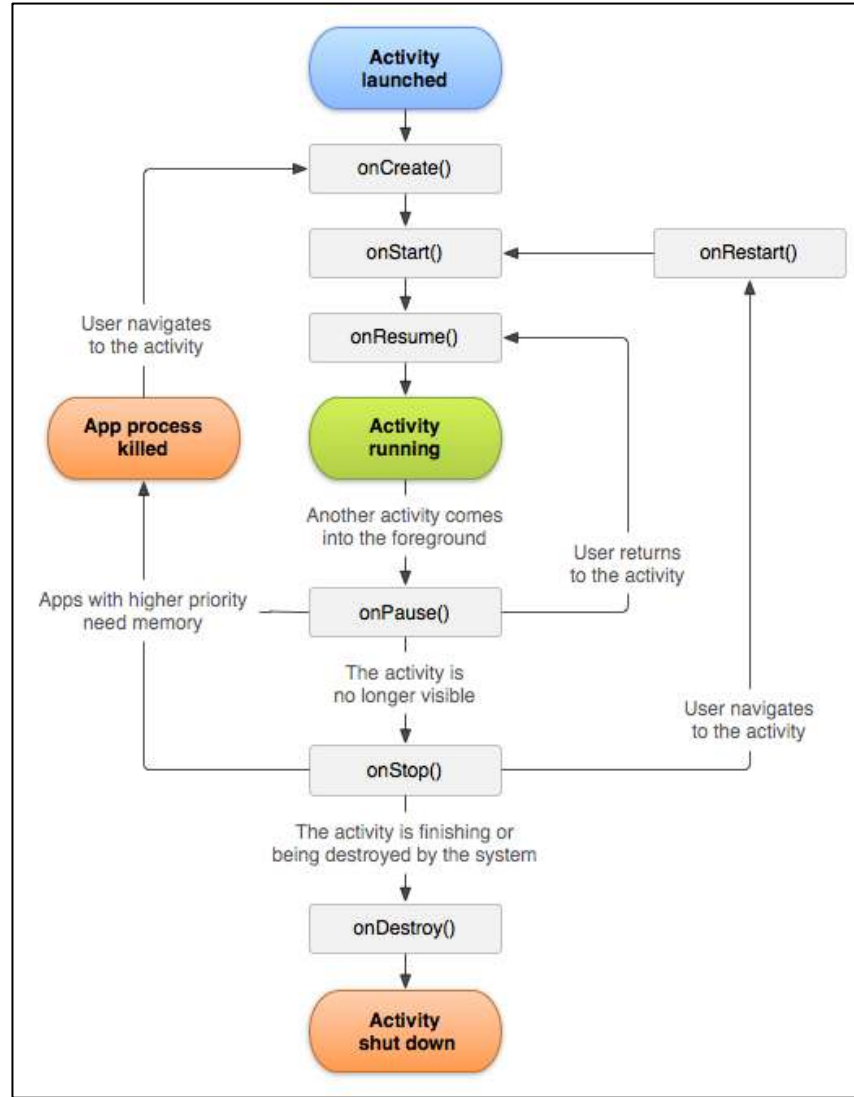
Geliştirme araçları haricinde, uygulama geliştirirken kullanılacak olan programlama diline karar verilmesi gerekmektedir. Bu programlama dilleri Java, C/C++ veya C# olabilir. Bu uygulamada, geliştirme aracı olarak Eclipse, programlama dili olarak ise Android geliştirici sitesinde destek verdiği java programlama dili kullanılmıştır. Android uygulaması geliştirmek için, “<http://developer.android.com>” adresinden AndroidSDK ve gerekli Eclipse eklentileri indirilerek başlanılmıştır.

4.1 Aktiviteler ve Uygulama Yaşam Döngüsü

Bu tez çalışmasında gerçekleştirilen Android uygulamasında, birden fazla aktiviteye (activity) sahip sayfalar kullanılmıştır. Aktivite, uygulamanın kullanıcı ile etkileşimini sağlayan, kullanıcıdan gelen bilgiler doğrultusunda işlem yapan sınıf yapısına sahip arayüzlerdir. Android uygulamalarında yer alan aktivitelerin bir yaşam döngüsü vardır. Bu yaşam döngüsü bir aktivitenin çalışma durumu kontrol eder. Birden fazla aktiviteye sahip olan uygulamalarda ön tanımlı (default) gelecek olan bir aktivite bulunur.

Android uygulamaları genel yaşam döngüsüne göre; bir aktiviteye, çalışma esnasında başlatılma, durdurulma, kapatılma gibi durumlar uygulanabilir. Aktivitenin kullanıcı tarafından görüntülenmesi için önce “onCreate()” komutundan geçmesi gerekir. Uygulama sırasında farklı bir aktiviteyi çalıştırmak için “onPause()” komutu kullanılarak aktivite duraklatılır. Aktiviteye, kaldığı yerden devam edilmek istendiği takdirde “onResume()” komutu kullanılarak ekrana getirilir. Bir aktivitenin durdurulması ise “onStop()” komutu ile gerçekleştirilir. Bu komuttan sonra aktivite “onCreate()” veya “onResume()” komutları kullanılarak tekrar görüntülenebilir.

Sonlandırılmış aktivite için “onDestroy()” komutunun çalıştırılması) yeniden görüntülenmesi için “onCreate()” komutu kullanılır. Bir aktivitenin yaşam döngüsü Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Android aktivite yaşam döngüsü (Anroid, 2011).

Aktivitelerin durum bilgileri, çalışma sırasında yığın (stack) bölgesinde tutulur. Yığınlar, “son giren ilk çıkar” mantığına göre çalışırlar. Yığına en son eklenen veri, ilk önce çekilecek olan veridir. Bir uygulamada en son başlayan aktivite bu yığın bölgesinin en üstüne yerleştirilir. Ondan önce başlamış aktiviteler ise sırasıyla bir alt bölgesinde tutulur. Bir aktivitenin kendisinden önceki aktiviteye dönmesi gerektiğinde, kendisini yok eder (“onDestroy()” komutu) ve yığın bölgesine geri dönerek önceki aktiviteyi devam ettirir. Bazen yeni bir aktivitenin çalıştırılması istendiğinde yığın

bölgesinde uygun yer yoksa bu bölgede bulunan ve yok edilmesi durumunda uygulamayı etkilemeyecek aktiviteler “onDestroy()” işlemine tabi tutulur. Bu işleme tabi tutulmadan önce “onSaveInstanceState()” komutu ile aktivitenin o an için son durumu kaydedilir. Uygulama esnasında tekrar bu aktiviteye dönmek gerekirse “onCreate()” metoduyla aktivite yeniden oluşturularak “onSaveInstanceState()” komutuyla aktiviteye ait durum bilgileri tekrar yüklenir.

4.2 AndroidManifest Dosyası

Bir Android uygulaması geliştirirken uygulamanın, hangi izinleri istediğini, hangi servisleri kullanacağı bilgisini öğrendiği ve uygulamaya ait ayarların bulunduğu uygulama projesinin ana dizininde bulunan “AndroidManifest” olarak isimlendirilmiş, xml formatında bir dosyadır. Manifest etiketinin altında application tanımlaması ile uygulamanın özellikleri tanımlanabilir. Bu tanımlardan, “icon” Android uygulamasının simgesini belirler. “label” tanımlaması ise, uygulama isminin ne olacağını belirler. Android uygulamasında yer alan aktivitelerin görüntülenmesi ve özelliklerinin belirlenmesi için “application” etiketinin altında tanımlamalar yapılır. Örneğin, bir sayfanın tanımı, çalıştıracağı kod dosyasının adı, açılış türü gibi bilgilere yer verilir. En temel haliyle bir uygulamaya ait isim, sürüm, simge ve en düşük çalışabileceği Android SDK bilgilerinin yer aldığı aynı zamanda uygulamanın çalışmasıyla ekranda bir aktivite gösterilmesini sağlayan “AndroidManifest.xml” içeriği Şekil 4.2’de gösterilmiştir.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.kbs.tez" android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="8" />
    <application android:icon="@drawable/ic_launcher"
        android:label="@string/app_name">
        <activity
            android:label="Yusuf MUŞTU - 2011"
            android:name=".acilis" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Şekil 4.2. AndroidManifest dosyası.

Manifest etiketinin altında yer alan bir diğer özellik ise güvenlik anlamında kullanıcıları bilgilendiren, uygulamanın hangi servisleri kullanacağını belirlediği etiket tanımıdır. Uses-permission etiketi ile tanımlanan bu özelliklere kısa mesaj göndermek ve internet bağlantısının kontrolünün yapılmasını sağlamak üzere kullanılmıştır. Kullanımla ilgili örnek Şekil 4.3’de gösterilmiştir.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.kbs.tez" android:versionCode="1" android:versionName="1.0" >
<uses-sdk android:minSdkVersion="8" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-
permission>
</manifest>
```

Şekil 4.3. Android uygulama izinleri.

4.3 Uygulama Açılış Ekranı

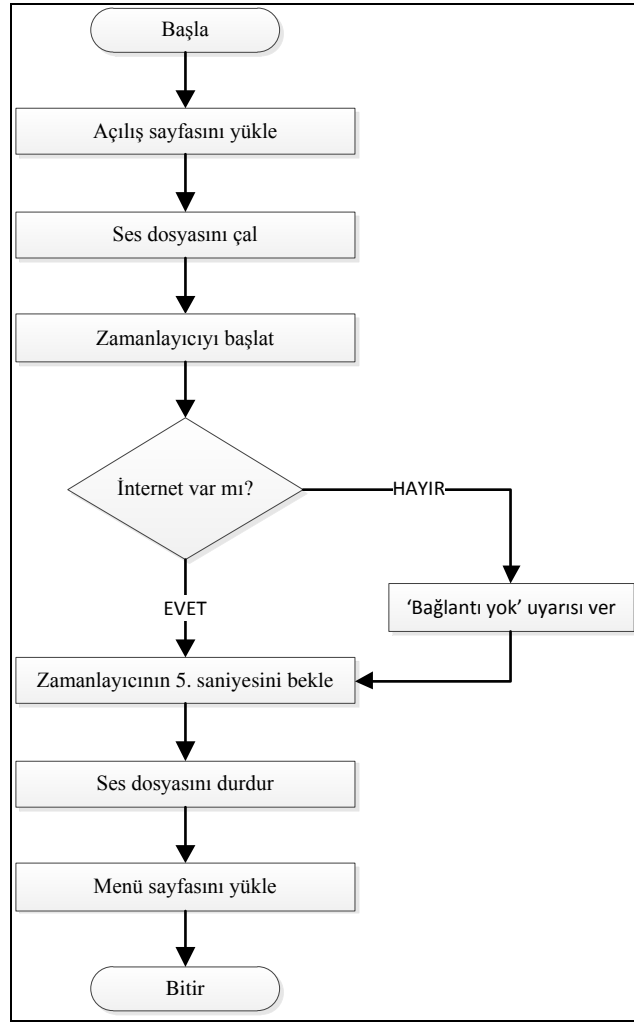
Açılış ekranı, Android uygulamasının açılması ile birlikte gelen ilk karşılama ekranıdır. Android uygulamasında bir aktivite oluşturulurken kullanıcı ve kontrol işlemlerinin yapıldığı “.java”, kontrollerin konumlandırılması ve özelliklerin belirleneceği “.xml” uzantılı arayüz dosyaları oluşturulur. Bu dosyaların etkinleştirilebilmesi için, “AndroidManifest.xml” dosyası içerisinde tanımlanması gerekmektedir.

Bu çalışmada açılış ekranında internet bağlantısının olup olmadığını denetleyen ve bağlantı durumuna göre kullanıcıyı bilgilendiren bir yapı tasarlanmıştır. Açılışta bir arka plan resmi ile birlikte ses dosyası yürütülmüştür. İnternet bağlantısının olmaması halinde kullanıcıya “İnternet bağlantısı yoktur.” şeklinde bir uyarı mesajı verilmiştir. Açılış ekranına ait görüntü Şekil 4.4’de gösterilmiştir.



Şekil 4.4. Açılış ekranı.

Uygulama çalışmaya başladığı anda açılış ekranı olarak oluşturulan “acilis.xml” dosyasında tasarımı belirtilmiş olan arayüz ekranda gösterilir. Ekranın gelmesi ile birlikte 5 saniyelik bir ses dosyası yürütülmeye başlanmaktadır. Ses dosyasının eklenmesi isteğe bağlıdır. Açılış ekranına ses dosyasının eklenmesi, uygulamada kullanılacak sesli etkileşimler için bir kontrol basamağı olarak düşünülmüştür. Beşinci saniyenin sonunda ses dosyası yürütülmesi durdurulur ve proje içerisindeki menü ekranına yönlendirme yapılır. Bu iş akışının yürütülmesi Şekil 4.5’de gösterilmiştir.



Şekil 4.5. Açılış ekranı iş akışı.

Açılış ekranında kullanılan ses oynatıcısı, “MediaPlayer” sınıfını kullanmaktadır. Bu sınıfın kullanılması için öncelikle bir ses dosyasının uygulama projesine yüklenmesi gerekir. Bir ses dosyası yüklendikten sonra “create()” komutu ile MediaPlayer’e aktarılır ve “start()” komutu ile yürütülmeye başlar. Açılış ekranına ait java ve xml kodları EK 2’de “Android Uygulama Kodları” başlığı altında yer almaktadır.

4.4 Kayıt Ekranı

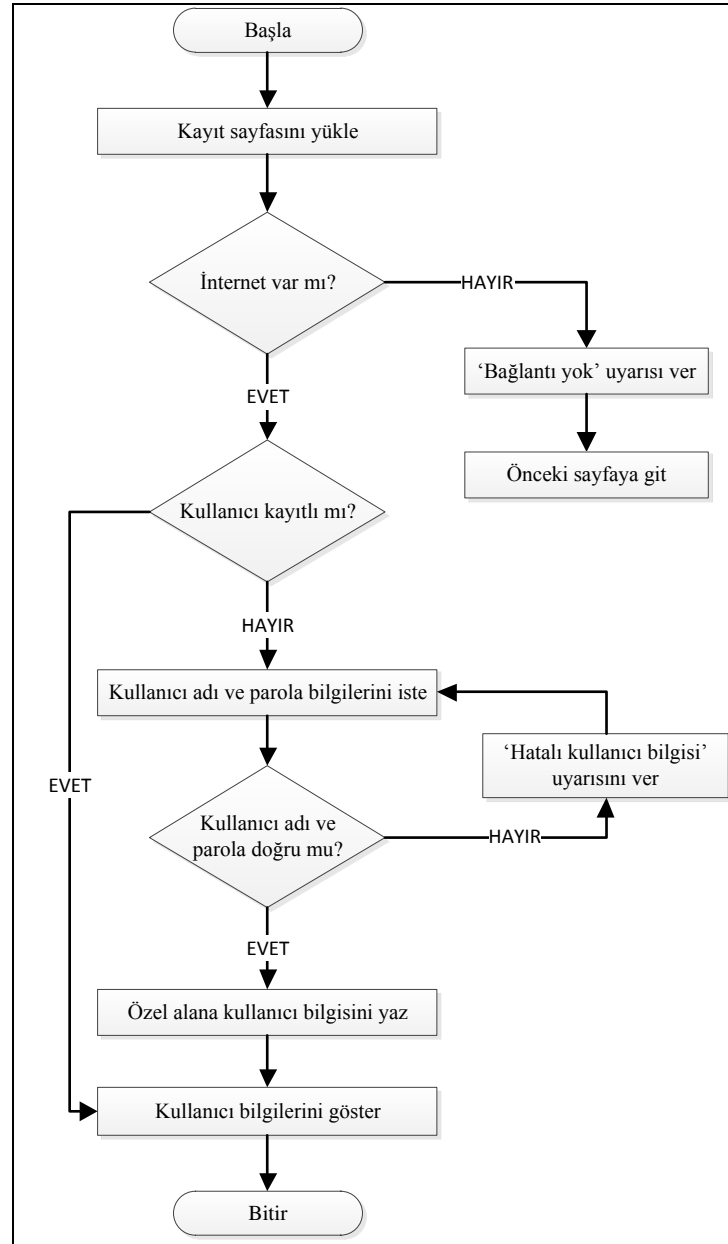
Uygulamayı kullanacak olan eğitmen ve öğrencilerin sistem üzerinde tanıtılması gerekmektedir. Tanıtma işlemi için, web hizmetinin sağlandığı sunucuda oluşturulan bir web servisi desteği ile Android uygulamasının haberleştirilmesi sağlanmıştır. Bu web servisi kayıt işlemi sırasında kullanıcının, kullanıcı adı ve parola bilgisinin doğruluğunu

kontrol eder. Kontrol işleminden başarıyla geçildikten sonra kullanıcı her uygulamayı açtığı anda kullanıcı adı ve parola bilgilerini girmemesi için kontrol esnasında sadece o kullanıcı için bir tekil değer oluşturulur. Kullanıcı için oluşturulan bu tekil değer Android uygulamasına gelmesiyle, uygulamanın her açılması sırasında kullanıcı adı ve parola girişine gerek kalmadan bu tekil değer üzerinden uygulamanın kullanılması sağlanır. Oluşturulan kullanıcı kayıt ekranı Şekil 4.6’da gösterilmiştir.

Şekil 4.6. Kullanıcı kayıt ekranı.

Kullanıcı kayıt ekranının Android ve web servisi tarafında olan iş akışı Şekil 4.7’de gösterilmiştir. Bu uygulamanın iş akışında, cihazın internet bağlantısı olup olmadığı kontrolü ve kullanıcının daha önceden kayıt olup olmadığının denetimi Android uygulaması tarafından yapılmaktadır. Kullanıcının bir ağa bağlı olup olmadığı “ConnectivityManager” sınıfı kullanılarak öğrenilebilir. “getService()” komutu ile bağlı olan ağ servisinin bilgileri alınarak “getActiveNetworkInfo()” komutunun altında yer alan “isConnected()” metodu ile geriye “true” veya “false” değerinin döndürülmesi sağlanır. “True” değeri gelmesi halinde ağ ortamına bağlı olduğu, “false” değerinin alınması ile ağ ortamında bir sıkıntı olduğu anlaşılır. Sorun durumunda, ekrana “Bağlantı yok” uyarısı ekranda gösterilir. Kayıt ekranına ait java ve xml kodları

EK 2’de “Android Uygulama Kodları” başlığı altında yer almaktadır.



Şekil 4.7. Kullanıcı kayıt işlemi iş akışı.

Uygulamanın çalışacağı cihazın ağ bilgilerini almak için “AndroidManifest.xml” dosyasından "permission.ACCESS_NETWORK_STATE" izni alınmalıdır. Bu izin sayesinde kullanıcının ağ bilgilerine ulaşılabilir.

Ağ bağlantısı kontrolünden sonra kullanıcının daha önceden uygulamayı kullanarak doğrulama işlemi yapıp yapılmadığı öğrenilir. Bunun için doğrulama işleminden sonra uygulamanın çalıştığı alanda özel bir dosya oluşturulur. Bu dosyada

kullanıcı adı ve doğrulama işlemi sonrasında web servisinden gelen kullanıcıya ait tekil bilgiyi kaydeder. Eğer bu alanda veriler mevcut ise uygulamanın diğer sayfalarında bu veriler kullanılır. Bu verilerin hatalı olması, kullanıcı tarafından bir şekilde değiştirilmesi veya kullanıcının sunucu tarafında tekil değerinin değişmesi durumunda öğrenci bilgi sisteminden kullanıcıya ait bilgi gelemeyeceğinden kullanıcının tekrar sisteme giriş yapması gerekecektir. Uygulama güvenliği açısından; Android uygulamasından kullanıcı adı ve parola bilgisi sisteme her gönderildiğinde oluşturulan tekil değer değiştirilmektedir.

Android uygulaması tarafından çok fazla verinin kayıt edilmesi ve bu veriler arasında işlem yapılması gerekiyorsa Android üzerinde çalışacak veritabanı desteğine ihtiyaç duyulabilir. Bu tarz durumlarda Android kütüphanesinde yer alan “sqlite” veritabanı kullanılabilir. Bu tez uygulamasında kaydedilmesi gereken veri miktarı az olduğu için dosya üzerinden bilgi saklama işlemi gerçekleştirilmiştir.

Verileri Android üzerinde kaydetme işlemi için SharedPreferences sınıfı kullanılmıştır. Bu sınıfın kullanılması için getSharedPreferences() komutu ile dosya tanımı ve dosyanın kullanılması ile ilgili parametre verilir. getSharedPreferences("kbs",MODE_PRIVATE) şeklinde yazılan bir komut için ilk parametre olan “kbs” verilerin kaydedileceği dosyanın ismidir. Bir sonraki parametre ise dosyanın erişim izinleri ile alakalıdır. “MODE_PRIVATE” parametresi kullanılarak oluşturulan dosyanın erişim izinleri, hangi uygulama tarafından oluşturulmuşsa ona aittir. Bu durum aynı kullanıcı kimliğini taşıyan uygulamalar içinde geçerli olabilir. Dosya tanımı yapıldıktan sonra, dosyaya yazma işlemi yapmak için edit() komutu kullanılarak, veri dosyası “SharedPreferences.Editor” sınıfından türetilen bir değişkene aktarılır. Tanımlanmak istenen bir değerin belirtilmesi Şekil 4.8’de gösterilmiştir.

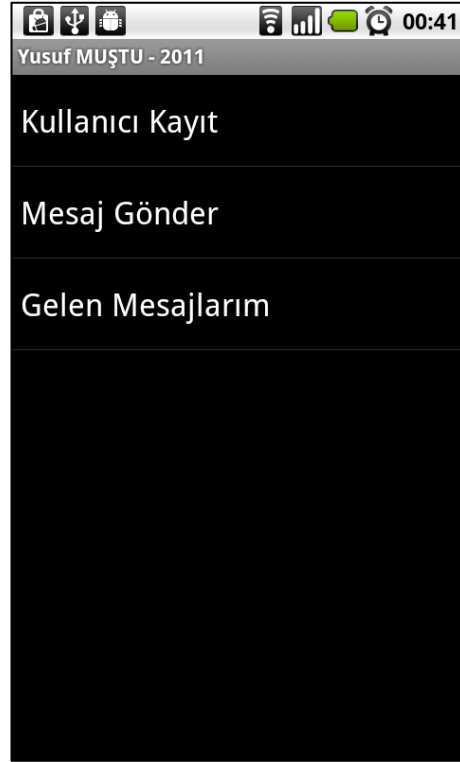
```
final SharedPreferences DosyaAyar = getSharedPreferences("DosyaAdı",MODE_PRIVATE);
final SharedPreferences.Editor DosyaDuzen = DosyaAyar.edit();
DosyaDuzen.putString("Yazılacak Değerin Tanımı", " Yazılacak Değer");
DosyaDuzen.commit();
```

Şekil 4.8. SharedPreferences tanımı.

4.5 Menü Ekranı

Şekil 4.9’da gösterilen uygulama menü ekranı, uygulamanın kullanılması için

ListActivity türünden oluşturulmuş aktivite sayfasıdır. Bu ekranda daha önceden oluşturulmuş ekranların kullanılması için yönlendirme yapılmaktadır. Bir ListActivity nesnesinin doldurulması için `setListAdapter()` komutu kullanılabilir. Bu işlem, `ArrayAdapter` türünden tanımlanan değişken üzerinden yapılır.

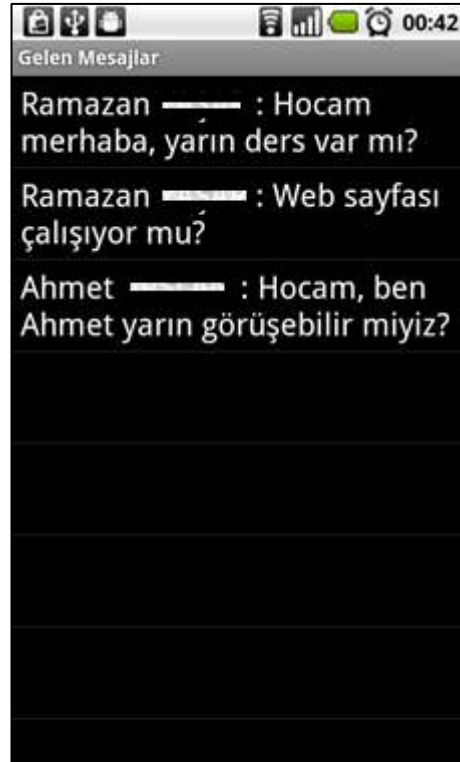


Şekil 4.9. Uygulama menü ekranı.

Açılış ekranından sonra gelen menü ekranı, kullanıcının daha önceden kayıt olup olmadığına bakar. Kullanıcının kaydı gerçekleştirmiş olması halinde gösterilir. Kullanıcı kayıt olmamışsa, sadece “Kullanıcı Kayıt” ekranının bağlantısı gösterilir. Menü ekranına ait java ve xml kodları EK 2’de “Android Uygulama Kodları” başlığı altında yer almaktadır.

4.6 Gelen Mesajlar Ekranı

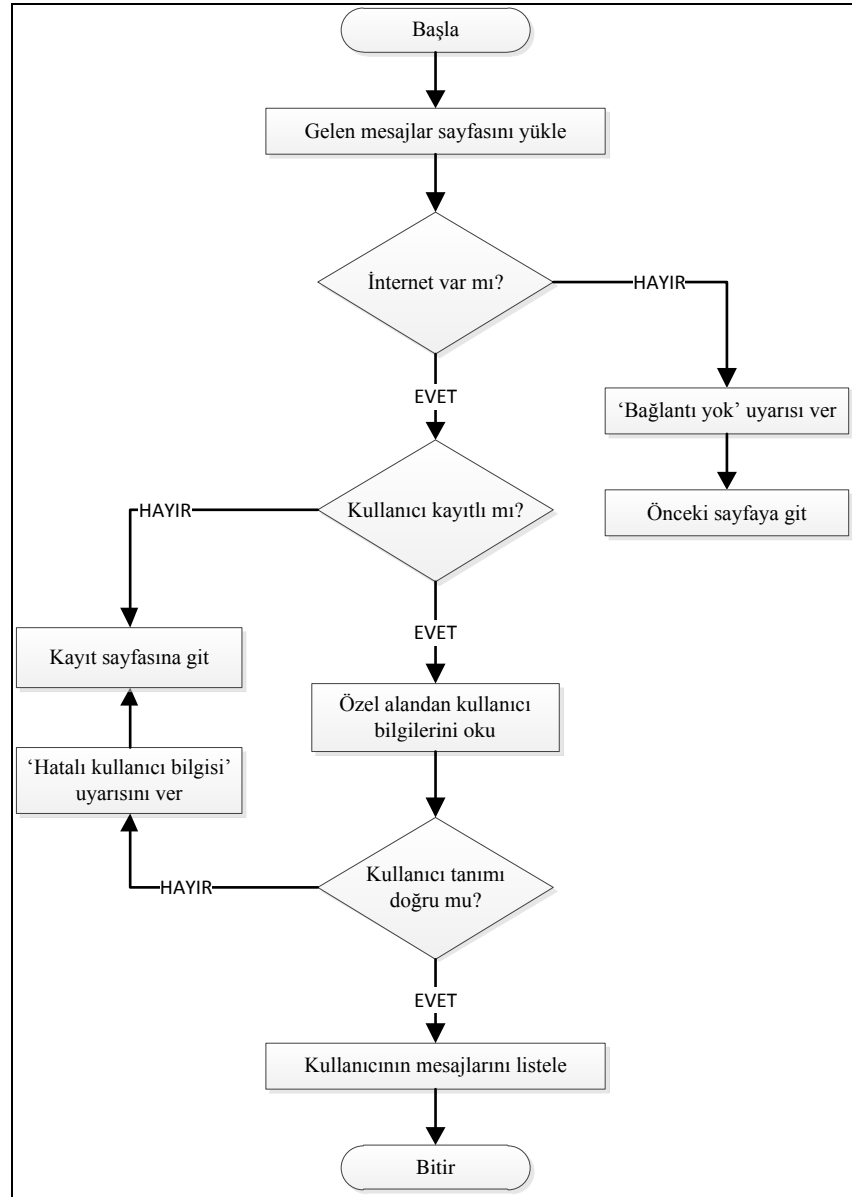
Gelen mesajların alınması işlemi için kullanıcının daha önceden sistemde kayıtlı olması gerekmektedir. Gelen mesajlar ekranı görüntüsü Şekil 4.10’da gösterilmiştir.



Şekil 4.10. Gelen mesajlar ekranı.

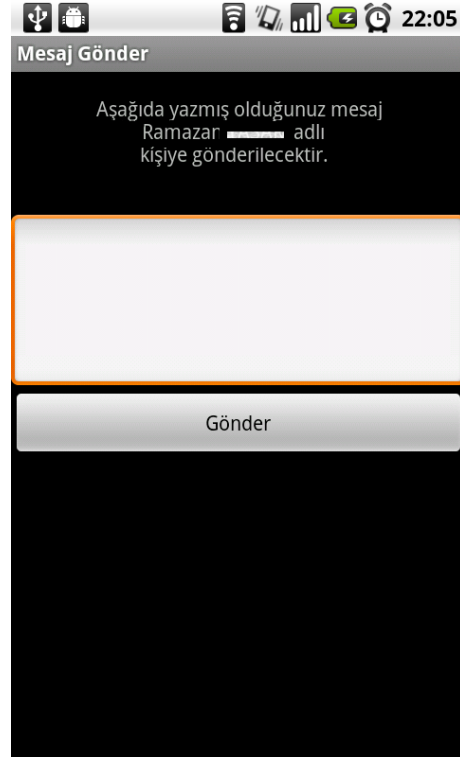
Bu ekranda mesajların gösterilmesi için kullanıcının sisteme kayıt olması gerekmektedir. Kayıt esnasında uygulamanın çalıştığı alana kaydedilen kullanıcıya ait olan tekil değeri alır. Bu alanda kullanıcı başarılı bir şekilde doğrulama işlemini gerçekleştirmesi ile oluşan Kullanıcı Adı ve Kullanıcı Id değeri bulunmaktadır. Bu değerler gelen mesajlar ekranının açılması ile okunmaya çalışılır. Böyle bir değer bu alanda yok ise kullanıcı bilgilendirilir.

Değerlerin mevcut olması durumunda ise bu değerler, sistemde oluşturulan mesajları getiren web servisi ile çekilmektedir. Bu web servisi de web uygulamasında görüşme istek sayfasında kullanılan sınıftan türetilmiştir. Gelen mesajların okunması ile ilgili işlemlerin iş akışı Şekil 4.11’de gösterilmiştir. Gelen mesaj ekranına ait java ve xml kodları EK 2’de “Android Uygulama Kodları” başlığı altında yer almaktadır.



Şekil 4.11. Gelen mesajlar iş akışı.

Gelen mesajlar ekranında listelenen mesajlara cevap vermek için farklı bir aktivite daha oluşturulmuştur. Bu aktivite “Mesaj Gönder” ekranı olarak isimlendirilmiştir. Mesaj gönderme ekranı Şekil 4.12’de gösterilmiştir. İnternet ortamından, web uygulamasında çalışan web servisini kullanarak işlemleri gerçekleştirmektedir. Bu işlemler, kısa mesaj gönderme ve web arabiriminden görüntüleme sağlayacak yapıdadır. Mesaj gönderme ekranına ait java ve xml kodları EK 2’de “Android Uygulama Kodları” başlığı altında yer almaktadır.



Şekil 4.12. Mesaj gönderme ekranı.

5. BÖLÜM – SONUÇLAR VE ÖNERİLER

Günümüzde cep telefonu kullanıcı sayılarının gittikçe artmasıyla birlikte mobil iletişime verilen önem hızla artmakta ve yeni teknolojilerin kullanımı tüm dünyada yaygınlaşmaktadır. Cep telefonları, küçük boyutlu, veri iletişim hızları yüksek, farklı teknolojilerle bütünleşebilir hale gelmiş ve günümüzün vazgeçilmez iletişim aracı olmuştur.

Cep telefonlarının eğitim-öğretim alanında kullanımı kapsamında bu çalışmada, sınıf içi öğrenci öğretmen etkileşimini daha verimli kılmak için mobil iletişimle bütünleşik bir sistem oluşturulmuştur. Android uygulaması, SMS sistemi ve Web arabirimi gibi farklı teknolojilerin bir arada çalışması sağlanmıştır. Bu şekilde, farklı ihtiyaçlar veya olanaklarla her kullanıcının sisteme rahat erişebilmesi sağlanmıştır.

Gerçekleştirilen sistemin, farklı kurumsal otomasyon sistemleri ile entegre çalışabilmesi için esnek bir yapı geliştirilmiştir. Uygulama kapsamında, öğrenci sınav notlarının bulunduğu veritabanı ile bütünleşik çalıştırılarak öğrencilere notlarının SMS ile gönderimi sağlanmış ayrıca kullanıcı bilgileri de yine kullanıcı veritabanından çekilerek kullanıcının rolleri buna göre belirlenmiştir.

SMS gönderme işlemini gerçekleştirmek için yerelde çalışan bir sunucu planlanmış, bu sunucunun bir GSM modem vasıtasıyla tüm SMS gönderme ve alma işlemlerini yapması sağlanmıştır. Bu şekilde, SMS gönderimi için ticari firmalardan hizmet satın almaya gerek kalmamıştır. Diğer taraftan; SMS sunucusunun yerelde konuşlandırılması sayesinde firma bağımlılığını ortadan kaldırdığı gibi, firmayla oluşabilecek sorunların da ortaya çıkma ihtimali bertaraf edilmiştir.

Cep telefonu, günümüzde hızlı iletişim için vazgeçilmez olmasına rağmen türlerindeki farklılık, tüm kullanıcılar için standart bir uygulama geliştirilmesinin önüne geçmektedir. Bu tez çalışması kapsamında, SMS sayesinde “hızlı ulaşım”, Android uygulaması sayesinde de “etkili iletişim” olanağı sağlanmıştır. Öğretiminin veya okul yönetiminin hızlı bir şekilde geniş bir öğrenci kitlesine ulaşmasını gerektiren durumlarda SMS sistemi, güzel bir çözüm olmaktadır.

Gerçekleştirilen uygulamayla eğitimcilerin sistemi daha rahat kullanabilmesi

için, web üzerinden çalışan yönetim arabirimi geliştirilmiştir. Bu arabirim sayesinde eğitimciler, dolu/boş zamanlarını düzenleyebilmekte, SMS üzerinden yapılmış olan geçmiş yazışmalara yeniden erişebilmekte, öğrencilere sınav not bilgilerinin SMS ile gönderilmesini sağlayabilmektedir.

Gerçeklenen sistemin maliyetine bakıldığında; uygulamaya oldukça müsait olduğu görülmektedir. Donanım olarak; SMS göndermek ve almak için kullanılan bir modem ile veritabanı, web sunucusu ve Windows servislerini çalıştırma amacı ile bir bilgisayar kullanılmıştır. İnternet'in performansı ve maliyetinin yanında GSM şebekesi üzerinden iletişim hem yavaş hem de daha pahalı olmaktadır. Ancak SMS'nin standart hale gelmesi ve çok yaygın olması nedeniyle, özellikle acil durumlar için SMS kullanımı gerekli görülmüştür. Gerçek bir kurumda bu sistemin uygulanmak istenmesi durumunda, sistemin GSM bileşeninin performansını arttırmak için, çoklu SIM kart destekleyen özel modemler kullanılabilir.

Gerçekleştirilen uygulamada sistemin tüm bileşenlerinde gerekli güvenlik yapılandırması sağlanabilmektedir. Web arabirimi ve Android uygulaması için, SSL kullanılarak ağ üzerinden taşınan trafiğin şifrelenerek gitmesi sağlanabilir. Bu şekilde, trafiği bir şekilde izlemek isteyen kötü niyetli birisinin verileri elde etmesinin önüne geçilebilmektedir. SMS sistemi zaten mevcut GSM altyapısının şifrelemesini kullanmakta ve mesajların başkaları tarafından okunabilmesi engellenmektedir.

Uygulayıcılara Öneriler:

Bu tezde uygulanan sistemin; profesyonel ve kurumsal anlamda kullanılabilmesi için, uygulamada kullanılan bilgisayar bileşenlerinin (normal bilgisayar sistemleri uzun süreli çalışmak üzere tasarlanmadığı için) sunucu mimarisinde donanım olarak planlanması gerekmektedir.

SMS mesajlarını göndermek ve almak için bu çalışmada kullanılan GSM modem cihazı tek SIM kartı desteklemektedir. Büyük kurumlarda, geniş kullanıcı kitlesi ile iletişim kurulacağı düşünüldüğünde, tek SIM kartlı modem ihtiyacı karşılamayacaktır. Bu tarz büyük uygulamalarda çok kartlı özel GSM modem cihazları tercih edilmeli ve yazılımlar bu cihazlara uygun olacak şekilde güncellenmelidir.

Araştıracılara Öneriler:

Bu tezde, ortaya konan uygulama ve sistemin daha da geliştirilmesi sağlanabilir. Yukarıda bahsedilmiş olan nedenlerden dolayı, “SMS kullanımı” uygulamanın en önemli bileşenlerinden olmuştur. Tezde kullanılan SMS sistemi, sadece metin içerikli mesajları işlemek üzere tasarlanmıştır. Sistem, daha da geliştirilerek MMS (Multimedia Messaging Service ~ Mobil Çoklu Ortam Mesajlaşma Hizmeti) destekler hale getirilebilir. Bu sayede; eğitimcinin öğrencilere çoklu ortam (resim, video, vb.) içerikli mesajlar da göndermesi sağlanabilir.

Tezde kullanılan sistemin geliştirmeye açık bir başka bileşeni, Android uygulamalarıdır. Android, günümüzün en hızlı gelişen ve yaygınlaşan mobil işletim sistemidir. Android’in hızlı yaygınlaşması, gelecekte mobil telefonların gideceği nokta konusunda ipuçları vermektedir. Bu çalışmadaki Android uygulamaları daha da geliştirilerek, mobil telefonların sunduğu bir çok özellikten faydalanılabilir. Örneğin, cep telefonundaki kamera sistemleri ile yüz tanıma sistemleri entegre edilerek, kullanıcıların Android uygulaması üzerinden sisteme kamera kullanarak oturum açması bir başka çalışma konusu olabilir. Ya da bir başka çalışma konusu olarak; Android yüklü akıllı telefonun kamera özelliği sayesinde, eğitimci-öğrenci arasında gerçek zamanlı çift yönlü iletişim sağlanabilir.

KAYNAKLAR

- Algan, S., “Her Yönüyle C# 4.0”, **Pusula**, İstanbul,2010.
- Android, <http://developer.android.com/guide/basics/what-is-android.html>, 2011.
- Axelsson,J., Dinçer, G., “Her Yönüyle Seri Port”, **Birsen Yayınevi**, 2000.
- Bilgi Teknolojileri ve İletişim Kurumu (BTK), “Türkiye Elektronik Haberleşme Sektörü- Üç Aylık Pazar Verileri Raporu”, **BTK**, Ankara, 2011.
- Bille,P., Thorup, M., “Faster Regular Expression Matching”, In Proc. 36th ICALP,171-182,2009
- Çakır, H., “Mobil Öğrenmeye İlişkin Bir Yazılım Geliştirme ve Değerlendirme”, **Çukurova Üniversitesi Eğitim Fakültesi Dergisi**, 2(40): 01-09 (2011)
- Cavuş, N., & İbrahim, “M-Learning: An experiment in using SMS to support learning new English language words” **British Journal of Educational Technology**, 40(1):78-91, 2009
- Gartner, “**Worldwide Mobile Device Sales to End Users**”, <http://www.gartner.com/it/page.jsp?id=1848514>,2011.
- Gülseren, D., “Mobil İletişim Teknolojilerinin Öğrenci Bilgi Sistemlerinde Kullanımı ve Bir Uygulama”, **Yüksek Lisans Tezi, Anadolu Üniversitesi Fen Bilimleri Enstitüsü**, Eskişehir, 2006
- Haghirian, P., Madlberger M., and Tanuskova, A., “Increasing advertising value of mobile marketing – An empirical study of antecedents”, **Presented at 38th Hawaii International Conference on System Sciences**, Hawaii, 2005.
- Hillebrand, F., Trosby, F., Holley, K., and Harris, I., "Short Message Service (SMS): The Creation of Personal Global Text Messaging", **Wiley**, 2010
- Imran, F., Hussain, M., “Advance Security aspects of Universal Mobile Telecommunication System (UMTS)”, Advance Security aspects of Universal Mobile Telecommunication System (UMTS), 33:87-93, 2011.
- ITU (International Telecommunication Union), "The World in 2010", <http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf>, 2010
- İşman, A., “Uzaktan Eğitim”, **Pegem Yayıncılık**, İstanbul,2008.
- Kleene, S., "Representation of events in nerve nets and finite automata", Princeton University Press 3-41, Princeton, 1956

KAYNAKLAR (Devam Ediyor)

- Kilgo, P.M., (n.d.). *Android OS: A robust, free, open-source operating system for mobile devices*. Retrieved October 1, 2009, from <http://pk-fire.com/etc/wtf/Android-osfinal.pdf>
- Lee, K.B., “Developing Mobile Collaborative Learning Applications for Mobile Users”, *International Journal of Interactive Mobile Technologies*, 5(4): 42-48, 2011
- Megep, “Elektrik Elektronik Teknolojisi Kablosuz Ağ Sistemleri Modülü”, *Milli Eğitim Bakanlığı*, Ankara, 2007.
- Nagarajan, M., Verma, K., Sheth, A.P., Miller, J., and Lathem J., “Semantic Interoperability of Web Services – Challenges and Experiences”, *IEEE International Conference on Web Services*, Chicago, 2006.
- Open Handset Alliance, http://www.openhandsetalliance.com/oha_members.html, 2011.
- Pocatilu, P., “Developing Mobile Learning Applications for Android using Web Services,” *Economy Informatics*, 14[106-115], (2010)
- Ramli, A., Ismail, I., and Idrus, “Mobile learning via SMS among distance learners: Does learning transfer occur?”, *International Journal of Interactive Mobile Technologies*, 4:30–35, 2010
- Sarısakal, M. N., Aydın, M. A., “E-Ticaretin Yeni Yüzü Mobil Ticaret”, *Havacılık Ve Uzak Teknolojileri Dergisi*, 1(2): 83-90 (2003).
- Siang, B.K., Bin Ramli, A.R., Prakash V., and Bin Syed Mohamed, S. A. R., “SMS gateway interface remote monitoring and controlling via GSM SMS”, *4th National Conference on Telecommunication Technology*, 84-87, 2003.
- So, S., “The development of a SMS-based teaching and learning system”, *Journal of Educational Technology Development and Exchange*, 2(1):113–124, 2009.
- Stone, A., Briggs, J., Smith, C., “SMS and Interactivity – Some Results from the Field, and its Implications on Effective Uses of Mobile Technologies in Education”, *Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education*, 2002
- Taç, M., “Android Programlama”. *Dikeyksen*, İstanbul, 2011.
- Turan, M., Baykara, M., Güler, Z., Avcı, E., “Açık Kaynak Mobil İşletim Sistemi: Android İşletim Sistemi”, *Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 349-354, (2011)

KAYNAKLAR (Devam Ediyor)

- Türkoğlu, T., “Profesyonel Programlama Teknikleri.NET ISBN:9759352714”, *Kişisel*, 2007.
- Yuan, S. T., Cheng, C., “Ontology-based personalized couple clustering for heterogeneous product recommendation in mobile marketing”, *Expert Systems with Applications*, 26(4): 461-476 (2004).
- Yüksek, Y., “Web Servislerinin Anlamsal Etkilenmesi”, *Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 362-366, Elazığ, 2011.
- Yüksel M. E., Zaim A. H., “GSM/GPRS Aygıtları Üzerinden Çok Dilli SMS Gönderme”, *XI. Akademik Bilişim Konferansı* , Harran Üniversitesi, 2009.

EKLER

EK 1. Windows Servisleri Uygulama Kodları

GSM Modem Haberleşme Servisi

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Linq;
using System.ServiceProcess;
using System.Text;
using System.Timers;
using System.IO.Ports;
using System.Text.RegularExpressions;
using System.Data.SqlClient;
using System.IO;

namespace SMSHaberlesme
{
    public partial class Service1 : ServiceBase
    {
        string SQLBaglantiCumlecigi = "Data Source=localhost; Initial
Catalog=kbs; Integrated Security=True";
        public Service1()
        {
            InitializeComponent();
        }
        protected override void OnStart(string[] args)
        {
            Timer TmrSMShaberles = new Timer(5000);
            TmrSMShaberles.Enabled = true;
            TmrSMShaberles.Elapsed += new
ElapsedEventHandler(SMSModemHaberles);
            TmrSMShaberles.Start();
        }

        void SMSModemHaberles(object sender, ElapsedEventArgs e)
        {
            SerialPort Sim900SP = new SerialPort("COM16", 115200,
Parity.None, 8, StopBits.One);
            if (!Sim900SP.IsOpen)
            {
                try
                {
                    Sim900SP.Open();
                    Sim900SP.WriteLine("AT+CMGL=\"REC UNREAD\"" +
Environment.NewLine);
                    System.Threading.Thread.Sleep(1000);
                    byte[] read = new byte[Sim900SP.BytesToRead];
                    Sim900SP.Read(read, 0, read.Length);
                    System.Text.ASCIIEncoding enc = new
System.Text.ASCIIEncoding();
                    Regex r = new
Regex("(\\\"\\+\\d{12}\\\"), (\\\"\\\"), (\\\".*\\\")\\r\\n(.*)");
                    Match m = r.Match(enc.GetString(read));
                    while (m.Success)

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

{
string[] Mesaj = { m.Groups[1].Value.ToString(),
m.Groups[3].Value.ToString(), m.Groups[4].Value.ToString() };

GelenMesajKaydet (Mesaj[0].ToString().Replace("\"", "").Substring(1,
12), Mesaj[2].ToString().Substring(0, Mesaj[2].ToString().Length -
1));

        m = m.NextMatch();
    }
    Sim900SP.Close();
}

catch (Exception Hata)
{
    SistemIslemBilgisiEkle(Hata.Message.ToString());
}
}

if (!Sim900SP.IsOpen && DateTime.Now.Minute == 20 &&
DateTime.Now.Second > 50)
{
    try
    {
        Sim900SP.Open();
        Sim900SP.WriteLine("AT+CMGDA=\"DEL READ\"" +
Environment.NewLine);
        System.Threading.Thread.Sleep(100);
        Sim900SP.WriteLine("AT+CMGDA=\"DEL SENT\"" +
Environment.NewLine);
        System.Threading.Thread.Sleep(100);
        Sim900SP.Close();
    }

    catch (Exception Hata)
    {
        SistemIslemBilgisiEkle(Hata.Message.ToString());
    }
}
DataTable GidecekMesajlar = GidecekMesajlariAl();
if (GidecekMesajlar.Rows.Count > 0)
{
    if (!Sim900SP.IsOpen)
    {
        try
        {
            Sim900SP.Open();
            Sim900SP.WriteLine("AT+CMGS=\"\" +
GidecekMesajlar.Rows[0][\"Telefon\"].ToString() + \"\" +
Environment.NewLine);

            System.Threading.Thread.Sleep(100);

            Sim900SP.WriteLine(GidecekMesajlar.Rows[0][\"Mesaj\"].ToString() +
(char)26);

            System.Threading.Thread.Sleep(100);
            Sim900SP.Close();

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

GidenMesajIslemYap(GidecekMesajlar.Rows[0]["No"].ToString());
    }
    catch (Exception Hata)
    {

SistemIslemBilgisiEkle(Hata.Message.ToString());

}

    }
}

DataTable GidecekMesajlariAl()
{
    DataTable Tablo = new DataTable(null);
    try
    {

        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        SqlCommand Komut = new
SqlCommand("WSSMSModemHaberSP_GidecekMesajlariAl", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlDataAdapter GelenVeri = new SqlDataAdapter();
        GelenVeri.SelectCommand = Komut;
        GelenVeri.Fill(Tablo);
        return Tablo;
    }
    catch (Exception Hata)
    {
        SistemIslemBilgisiEkle(Hata.Message.ToString());
        return Tablo;
    }
}

void GidenMesajIslemYap(string No)
{
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();
        SqlCommand Komut = new
SqlCommand("WSSMSModemHaberSP_GidenMesajIslemYap", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPPrmNo = Komut.Parameters.Add("@No",
SqlDbType.Int);
        SPPrmNo.Value = Convert.ToInt32(No);
        Komut.ExecuteNonQuery();
        Baglanti.Close();
    }

    catch (Exception Hata)
    {
        SistemIslemBilgisiEkle(Hata.Message.ToString());
    }
}

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

    }

}

void GelenMesajKaydet(string Telefon, string Mesaj)
{
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();

        SqlCommand Komut = new
SqlCommand("WSSMSModemHaberSP_GelenMesajKaydet", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPrmTelefon =
Komut.Parameters.Add("@Telefon", SqlDbType.Char, 12);
        SqlParameter SPrmMesaj =
Komut.Parameters.Add("@Mesaj", SqlDbType.NVarChar, 300);
        SPrmTelefon.Value = Telefon;
        SPrmMesaj.Value = Mesaj;
        Komut.ExecuteNonQuery();
        Baglanti.Close();
    }
    catch (Exception Hata)
    {
        SistemIslemBilgisiEkle(Hata.Message.ToString());
    }
}

void SistemIslemBilgisiEkle(string Bilgi)
{
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();
        SqlCommand Komut = new
SqlCommand("SP_SistemIslemBilgisiEkle", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPrmBilgi =
Komut.Parameters.Add("@Bilgi", SqlDbType.NVarChar, 300);
        SPrmBilgi.Value = Bilgi;
        Komut.ExecuteNonQuery();
        Baglanti.Close();
    }
    catch (Exception Hata)
    {
    }
}
}
}

```

Kısa Mesaj Yorumlama Servisi

```
namespace SMSIslem
```


EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```
{
    public partial class Servis1 : ServiceBase
    {
        string SQLBaglantiCumlecigi = "Data Source=localhost; Initial
Catalog=kbs; Integrated Security=True";
        public Servis1()
        {
            InitializeComponent();
        }

        protected override void OnStart(string[] args)
        {

            Timer TmrSMShaberles = new Timer(5000);
            TmrSMShaberles.Enabled = true;
            TmrSMShaberles.Elapsed += new
ElapsedEventHandler(SMSIslem);

            TmrSMShaberles.Start();
        }

        void SMSIslem(object sender, ElapsedEventArgs e)
        {
            string Telefon = "";
            string[] MesajIcerigi = null;
            DataTable IslemYapilmamisKisaMesaj =
IslemYapilmamisKisaMesajGetir();
            DataTable Tablo;
            if (IslemYapilmamisKisaMesaj.Rows.Count == 1)
            {

                Telefon =
IslemYapilmamisKisaMesaj.Rows[0]["Telefon"].ToString();
                MesajIcerigi =
IslemYapilmamisKisaMesaj.Rows[0]["Mesaj"].ToString().Split(' ');
                switch (MesajIcerigi[0].ToString())
                {
                    case "kayit":
                        Tablo =
KullaniciBilgisiniGetir(MesajIcerigi[1].ToString());

                        if (Tablo.Rows[0]["KullaniciTip"].ToString()
== "1")
                        {
                            if
(KullaniciTelefonEkle(Tablo.Rows[0]["TcKimlikNo"].ToString(),
Telefon))
                            {
                                GidecekMesajEkle(Telefon, "Akademisyen
olarak kaydiniz yapilmistir.");
                            }
                        }
                        else if
(Tablo.Rows[0]["KullaniciTip"].ToString() == "2")
                        {

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

if (KullaniciTelefonEkle(Tablo.Rows[0]["TcKimlikNo"].ToString(),
Telefon))
    {
        GidecekMesajEkle(Telefon, "Ogrenci
olarak kaydiniz yapilmistir.");
    }
    else
    {
        GidecekMesajEkle(Telefon, "Kullanici
olarak tanimli degilsiniz.");
    }
    break;

    case "dersbilgi":

        APEnYakinDersBilgilendirmesi(Telefon,
String.Join(" ", MesajIcerigi, 1, MesajIcerigi.Length - 1));

        GidecekMesajEkle(Telefon, "Mesajiniz alıcılara
gonderilmistir.");
        break;

    case "msj":
        string IstekMesaji = "";

        for (int i = 2; i < MesajIcerigi.Length; i++)
        {
            IstekMesaji = IstekMesaji + " " +
MesajIcerigi[i].ToString();
        }

        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();
        SqlCommand Komut = new
SqlCommand("SP_GorusmeIstekEkle", Baglanti);
        Komut.CommandType =
CommandType.StoredProcedure;
        SqlParameter SPrmTelefon =
Komut.Parameters.Add("@Telefon", SqlDbType.NVarChar, 12);
        SqlParameter SPrmEposta =
Komut.Parameters.Add("@Eposta", SqlDbType.NVarChar, 20);
        SqlParameter SPrmMesaj =
Komut.Parameters.Add("@Mesaj", SqlDbType.NVarChar, 160);
        SPrmEposta.Value = MesajIcerigi[1].ToString();
        SPrmTelefon.Value = Telefon;
        SPrmMesaj.Value = IstekMesaji;
        Komut.ExecuteNonQuery();
        Baglanti.Close();
        break;
    case "not":
        string IstekMesaji2 = "";
        for (int i = 2; i < MesajIcerigi.Length; i++)
        {

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

                                IstekMesaji2 = IstekMesaji2 + " " +
MesajIcerigi[i].ToString();
                                }

                                SqlConnection Baglanti2 = new
SqlConnection(SQLBaglantiCumlecigi);
                                Baglanti2.Open();
                                SqlCommand Komut2 = new
SqlCommand("SP_NotOgren", Baglanti2);
                                Komut2.CommandType =
CommandType.StoredProcedure;
                                SqlParameter SPrmTelefon2 =
Komut2.Parameters.Add("@Telefon", SqlDbType.NVarChar, 12);
                                SqlParameter SPrmDersKodu =
Komut2.Parameters.Add("@DersKodu", SqlDbType.NVarChar, 50);
                                SPrmTelefon2.Value = Telefon;
                                SPrmDersKodu.Value =
MesajIcerigi[1].ToString();
                                Komut2.ExecuteNonQuery();
                                Baglanti2.Close();
                                break;

                                default:
                                GidecekMesajEkle(Telefon, "Gecersiz bir islem
gerceklestirdiniz.");

                                break;
                                }

GelenMesajIslemYap(IslemYapilmamisKisaMesaj.Rows[0]["No"].ToString());
    }
}

DataTable KullaniciBilgisiniGetir(string TcKimlikNo)
{
    DataTable Tablo = new DataTable(null);
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        SqlCommand Komut = new
SqlCommand("SP_KullaniciBilgisiniGetir", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPrmTcKimlikNo =
Komut.Parameters.Add("@TcKimlikNo", SqlDbType.NVarChar, 500);
        SPrmTcKimlikNo.Value = TcKimlikNo;
        SqlDataAdapter GelenVeri = new SqlDataAdapter();
        GelenVeri.SelectCommand = Komut;
        GelenVeri.Fill(Tablo);
        return Tablo;
    }
    catch (Exception Hata)
    {

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

SistemIslemBilgisiEkle("Veritabanında Kullanıcı Bilgisi çekilirken
hata ile karşılaşıldı. Hata Mesajı: " + Hata.Message.ToString());
    return Tablo;
}
}

DataTable IslemYapilmamisKisaMesajGetir()
{
    DataTable Tablo = new DataTable(null);
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        SqlCommand Komut = new
SqlCommand("SP_IslemYapilmamisKisaMesajGetir", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlDataAdapter GelenVeri = new SqlDataAdapter();
        GelenVeri.SelectCommand = Komut;
        GelenVeri.Fill(Tablo);
        return Tablo;
    }
    catch (Exception Hata)
    {
        SistemIslemBilgisiEkle("Veritabanından GelenMesaj
tablosundan Islem Yapılmamış Mesaj alınırken hata meydana geldi. Hata
Mesajı: " + Hata.Message.ToString());
        return Tablo;
    }
}

bool KullaniciTelefonEkle(string TCKimlikNo, string Telefon)
{
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();
        SqlCommand Komut = new
SqlCommand("SP_KullaniciTelefonEkle", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPPrmTCKimlikNo =
Komut.Parameters.Add("@TCKimlikNo", SqlDbType.Char, 11);
        SqlParameter SPPrmTelefon =
Komut.Parameters.Add("@Telefon", SqlDbType.Char, 12);
        SPPrmTCKimlikNo.Value = TCKimlikNo;
        SPPrmTelefon.Value = Telefon;
        Komut.ExecuteNonQuery();
        Baglanti.Close();
        return true;
    }
    catch (Exception Hata)
    {
        SistemIslemBilgisiEkle("Kullanıcı telefon bilgisi
eklenirken hata meydana geldi. Hata Mesajı: " +
Hata.Message.ToString());
        return false;
    }
}

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

    }
    void APEnYakinDersBilgilendirmesi(string Telefon, string
Mesaj)
    {
        try
        {
            SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
            Baglanti.Open();
            SqlCommand Komut = new
SqlCommand("SP_APEnyakinDersBilgilendirmesi", Baglanti);
            Komut.CommandType = CommandType.StoredProcedure;
            SqlParameter SPPrmTelefon =
Komut.Parameters.Add("@Telefon", SqlDbType.Char, 12);
            SqlParameter SPPrmMesaj =
Komut.Parameters.Add("@Mesaj", SqlDbType.NVarChar, 300);
            SPPrmTelefon.Value = Telefon;
            SPPrmMesaj.Value = Mesaj;
            Komut.ExecuteNonQuery();
            Baglanti.Close();
        }
        catch (Exception Hata)
        {
            SistemIslemBilgisiEkle("SP_APEnyakinDersBilgilendirmesi sırasında hata
alındı. Hata Mesajı: " + Hata.Message.ToString());
        }
    }
    void GidecekMesajEkle(string Telefon, string Mesaj)
    {
        try
        {
            SqlConnection Baglanti = new SqlConnection(SQLBaglantiCumlecigi);
            Baglanti.Open();
            SqlCommand Komut = new
SqlCommand("SP_GidenMesajKaydet", Baglanti);
            Komut.CommandType = CommandType.StoredProcedure;
            SqlParameter SPPrmTelefon =
Komut.Parameters.Add("@Telefon", SqlDbType.Char, 12);
            SqlParameter SPPrmMesaj =
Komut.Parameters.Add("@Mesaj", SqlDbType.NVarChar, 300);
            SPPrmTelefon.Value = Telefon;
            SPPrmMesaj.Value = Mesaj;
            Komut.ExecuteNonQuery();
            Baglanti.Close();
        }
        catch (Exception Hata)
        {
        }
    }

    void GelenMesajIslemYap(string No)
    {
        try
        {

```

EK 1. Windows Servisleri Uygulama Kodları (Devam Ediyor)

```

        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();
        SqlCommand Komut = new
SqlCommand("SP_GelenMesajIslemYap", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPrmNo = Komut.Parameters.Add("@No",
SqlDbType.Int);
        SPrmNo.Value = Convert.ToInt32(No);
        Komut.ExecuteNonQuery();
        Baglanti.Close();
    }
    catch (Exception Hata)
    {}
}

void SistemIslemBilgisiEkle(string Bilgi)
{
    try
    {
        SqlConnection Baglanti = new
SqlConnection(SQLBaglantiCumlecigi);
        Baglanti.Open();
        SqlCommand Komut = new
SqlCommand("SP_SistemIslemBilgisiEkle", Baglanti);
        Komut.CommandType = CommandType.StoredProcedure;
        SqlParameter SPrmBilgi =
Komut.Parameters.Add("@Bilgi", SqlDbType.NVarChar, 500);
        SPrmBilgi.Value = Bilgi;
        Komut.ExecuteNonQuery();
        Baglanti.Close();
    }
    catch (Exception Hata)
    {}
} }

```

EK 2. Android Uygulama Kodları

Açılış Sayfası

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/acilis">
</LinearLayout>
```

```
public class acilis extends Activity
{
    MediaPlayer acilisMuzik;
    protected void onPause()
    {
        super.onPause();
        finish();
        acilisMuzik.release();
    }
    protected void onCreate(Bundle acilis)
    {
        super.onCreate(acilis);
        setContentView(R.layout.acilis);
        acilisMuzik= MediaPlayer.create(acilis.this, R.raw.acilis);
        acilisMuzik.start();
        Thread Zamanlayici = new Thread()
        {
            public void run()
            {
                try
                {
                    sleep(5000);
                }
                catch (InterruptedException e)
                {
                    e.printStackTrace();
                }
                finally
                {
                    Intent openStartingPoint =new Intent("com.kbs.tez.MENU");
                    startActivity(openStartingPoint);
                }
            }
        };
        Zamanlayici.start();
    }
}
```

Kayıt Sayfası

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

android:orientation="vertical" >
    <TextView
        android:id="@+id/tvBilgi"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="" />
    <TextView
        android:id="@+id/tvKullaniciAdi"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Kullanıcı Adı:" />
    <EditText
        android:id="@+id/etKullaniciAd"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >
        <requestFocus />
    </EditText>
    <TextView
        android:id="@+id/tvParola"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Parola:" />
    <EditText
        android:id="@+id/etparola"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword" />
    <Button
        android:id="@+id/btnkaydet"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Kullanıcı Doğrula" />
    <Button
        android:id="@+id/btnSil"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Kayıt Bilgilerini Sil" />
</LinearLayout>

```

```

public class kayit extends Activity
{
    TextView tvBilgi;
    TextView tvKullaniciAdi;
    TextView tvParola;
    EditText etKullaniciAd;
    EditText etparola;

    Button btnkaydet;
    Button btnSil;
    protected void onCreate(Bundle kayitekran)
    {
        super.onCreate(kayitekran);
        setContentView(R.layout.kayit);
    }
}

```


EK 2. Android Uygulama Kodları (Devam Ediyor)

```

final SharedPreferences DosyaAyar =
getSharedPreferences("kbsFile",MODE_PRIVATE);
    tvBilgi = (TextView) findViewById(R.id.tvBilgi);
    tvKullaniciAdi = (TextView) findViewById(R.id.tvKullaniciAdi);
    tvParola = (TextView) findViewById(R.id.tvParola);
    etKullaniciAd = (EditText) findViewById(R.id.etKullaniciAd);
    etparola = (EditText) findViewById(R.id.etparola);
    btnkaydet = (Button) findViewById(R.id.btnkaydet);
    btnSil = (Button) findViewById(R.id.btnSil);
    if(DosyaAyar.getString("KullaniciID", "").length()==36)
    {
        tvBilgi.setText("Uygulamaya daha önceden kayıt olunmuş. \n Kullanıcı:
*****"+ DosyaAyar.getString("KullaniciAdi", "").substring(8));
        btnkaydet.setVisibility(View.INVISIBLE);
        etKullaniciAd.setVisibility(View.INVISIBLE);
        etparola.setVisibility(View.INVISIBLE);
        tvKullaniciAdi.setVisibility(View.INVISIBLE);
        tvParola.setVisibility(View.INVISIBLE);
        btnSil.setVisibility(View.VISIBLE);
    }
    else
    {
        tvBilgi.setText("\nUygulamanın kullanılabilmesi için sistemde
kayıtlı olmanız gerekmektedir. Kullanıcı adı ve parola bilgilerini aşağıdaki
uygun yerlere yazarak doğrulamanın sağlanması durumunda işlemlerinize devam
edebilirsiniz.\n\nNot: Doğrulama işlemi internet üzerinden yapıldığı için
olumsuz durumlarda internet bağlantınızı kontrol ediniz. \n\n");
        btnSil.setVisibility(View.INVISIBLE);
    }
    btnkaydet.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v)
        {
            SoapObject Istek = new
SoapObject("http://tempuri.org/", "KullaniciDogrula");
            Istek.addProperty("KullaniciAdi",
etKullaniciAd.getText().toString());
            Istek.addProperty("Parola",
etparola.getText().toString());
            SoapSerializationEnvelope Paket = new
SoapSerializationEnvelope(Paket.VER11);
            Paket.dotNet = true;
            Paket.setOutputSoapObject(Istek);
            HttpTransportSE Servis = new
HttpTransportSE("http://10.9.9.9/servisler/WebService.asmx");
            try
            {

                Servis.call("http://tempuri.org/KullaniciDogrula", Paket);
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
        }
    }

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

try
    {
        if(Paket.getResponse().toString().length()>1)
        {
            DosyaDuzen.putString("KullaniciID",
            ""+(SoapPrimitive) Paket.getResponse());
            DosyaDuzen.putString("KullaniciAdi",
            etKullaniciAd.getText().toString());
            DosyaDuzen.commit();
            Toast.makeText(kayit.this,"Kullanıcı kaydınız
başarılı bir şekilde yapılmıştır.",Toast.LENGTH_SHORT).show();

            Intent menuEkran = new Intent(kayit.this,
            menu.class);
            startActivity(menuEkran);
        }
        else
        {
            etparola.setText("");
            Toast.makeText(kayit.this,"Hatalı Kullanıcı
Adı veya Parola!",Toast.LENGTH_SHORT).show();
        }
    }
    catch (SoapFault e)
    {
        e.printStackTrace();
    }
});
btnSil.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        DosyaDuzen.putString("KullaniciID", "");
        DosyaDuzen.putString("KullaniciAdi","");
        DosyaDuzen.commit();
        Toast.makeText(kayit.this,"Kullanıcı kaydınız
silinmiştir.",Toast.LENGTH_SHORT).show();
        Intent menuEkran = new Intent(kayit.this, menu.class);
        startActivity(menuEkran);
    }
});
}
}

```

Derslerim Sayfası

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

<TextView
    android:id="@+id/tvBilgi"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="" />
<TextView
    android:id="@+id/tvKullaniciAdi"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Kullanıcı Adı:" />
<EditText
    android:id="@+id/etKullaniciAd"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >
    <requestFocus />
</EditText>
<TextView
    android:id="@+id/tvParola"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Parola:" />
<EditText
    android:id="@+id/etparola"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword" />
<Button
    android:id="@+id/btnkaydet"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Kullanıcı Doğrula" />
<Button
    android:id="@+id/btnSil"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Kayıt Bilgilerini Sil" />
</LinearLayout>

public class derslerim extends ListActivity
{
    String[] Dersler;
    String[] DersAPDDKodlari;
    String[] Gunler;
    String[] Saatler;
    protected void onCreate(Bundle derslerim)
    {
        final SharedPreferences DosyaAyar =
getSharedPreferences("kbsFile",MODE_PRIVATE);
        final SharedPreferences.Editor DosyaDuzen = DosyaAyar.edit();
        super.onCreate(derslerim);

        SoapObject Istek = new SoapObject("http://tempuri.org/","Istek");

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

Istek.addProperty("KullaniciID",DosyaAyar.getString("KullaniciID",""));
    SoapSerializationEnvelope Paket = new
SoapSerializationEnvelope(Paket.VER11);
    Paket.dotNet = true;
    Paket.setOutputSoapObject(Istek);
    HttpTransportSE Servis = new
HttpTransportSE("http://192.168.1.55/servisler/WebService.asmx");
    try
    {
        Servis.call("http://tempuri.org/Istek", Paket);
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    try
    {
        SoapObject veri = (SoapObject) Paket.getResponse();
        int elementCount = veri.getPropertyCount();
        for(int i = 0;i<elementCount;i++)
        {
            StringTokenizer veriler = new
StringTokenizer(veri.getProperty(i).toString(), ";");
            DersAPDDKodlari[i] = veriler.nextToken();
            Gunler[i] = veriler.nextToken();
            Saatler[i] = veriler.nextToken();
            Dersler[i] = veriler.nextToken();
        }
    }
    catch (SoapFault e)
    {
        e.getMessage();
    }
    ArrayList<String> listDersler = new ArrayList<String>();
    for (String s : Dersler)
    {
        if (!s.equals(""))
        {

            listDersler.add(s);
        }
    }
    Dersler = listDersler.toArray(new String[listDersler.size()]);
    ArrayList<String> listDersAPDDKodlari = new ArrayList<String>();
    for (String s : DersAPDDKodlari)
    {
        if (!s.equals(""))
        {
            listDersAPDDKodlari.add(s);
        }
    }
}

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

DersAPDDKodlari = listDersAPDDKodlari.toArray(new
String[listDersAPDDKodlari.size()]);
    ArrayList<String> listDersGunler = new ArrayList<String>();
    for (String s : Gunler)
    {
        if (!s.equals(""))
        {
            listDersGunler.add(s);
        }
    }
    Gunler = listDersGunler.toArray(new
String[listDersGunler.size()]);
    ArrayList<String> listDersSaatler = new ArrayList<String>();
    for (String s : Saatler)
    {
        if (!s.equals(""))
        {
            listDersSaatler.add(s);
        }
    }
    Saatler = listDersSaatler.toArray(new
String[listDersSaatler.size()]);
    setListAdapter(new ArrayAdapter<String>(derslerim.this,
android.R.layout.simple_list_item_1, Dersler));
}

protected void onListItemClick(ListView l, View v, int position, long
id)
{
    super.onListItemClick(l, v, position, id);
    String APDD= DersAPDDKodlari[position];
    String Gun= Gunler[position];
    String Saat= Saatler[position];
    String Ders= Dersler[position];
    Intent i= new Intent(derslerim.this, sinifamesajgonder.class);

    Bundle bundle = new Bundle();
    bundle.putString("APDD", APDD);
    bundle.putString("Gun", Gun);
    bundle.putString("Saat", Saat);

    bundle.putString("Ders", Ders);
    i.putExtras(bundle);
    startActivity(i);
}
}

```

Mesajlarım Sayfası

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

</LinearLayout>

```
public class gorusmeistikleri extends ListActivity
{
    String[] Ad;
    String[] Soyad;
    String[] IstekMesaj;
    String[] IstekTarih;
    String[] IstekYapan;
    String[] GMesaj;
    protected void onCreate(Bundle gorusmeistikleri)
    {
        final SharedPreferences DosyaAyar =
getSharedPreferences("kbsFile",MODE_PRIVATE);
        final SharedPreferences.Editor DosyaDuzen = DosyaAyar.edit();
        super.onCreate(gorusmeistikleri);
        SoapObject Istek = new
SoapObject("http://tempuri.org/", "GorusmeIstekGetir");

        Istek.addProperty("KullaniciID",DosyaAyar.getString("KullaniciID",""))
;
        SoapSerializationEnvelope Paket = new
SoapSerializationEnvelope(Paket.VER11);
        Paket.dotNet = true;
        Paket.setOutputSoapObject(Istek);
        HttpTransportSE Servis = new
HttpTransportSE("http://192.168.1.55/servisler/WebService.asmx");
        try
        {
            Servis.call("http://tempuri.org/GorusmeIstekGetir",
Paket);
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        try
        {
            SoapObject object = (SoapObject) Paket.getResponse();

            int elementCount = object.getPropertyCount();
            for(int i = 0;i<elementCount;i++)
            {
                StringTokenizer veriler = new
StringTokenizer(object.getProperty(i).toString(), ";");
                Ad[i] = veriler.nextToken();
                Soyad[i] = veriler.nextToken();
                IstekMesaj[i] = veriler.nextToken();
                IstekTarih[i] = veriler.nextToken();
                IstekYapan[i] = veriler.nextToken();
            }
        }
        catch (SoapFault e)
        {
            e.getMessage();
        }
    }
}
```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

}

for(int i = 0;i<Ad.length;i++)

{
    if(Ad[i].toString().length(>0)
    {
        GMesaj[i]=Ad[i].toString()+"
"+Soyad[i].toString()+" : "+IstekMesaj[i].toString();
    }
}
ArrayList<String> listDersler = new ArrayList<String>();
for (String s : Ad)
{
    if (!s.equals(""))
    {
        listDersler.add(s);
    }
}
Ad = listDersler.toArray(new String[listDersler.size()]);
ArrayList<String> listDersAPDDKodlari = new ArrayList<String>();
for (String s : Soyad)
{
    if (!s.equals(""))
    {
        listDersAPDDKodlari.add(s);
    }
}
Soyad = listDersAPDDKodlari.toArray(new
String[listDersAPDDKodlari.size()]);
ArrayList<String> listDersGunler = new
ArrayList<String>();
for (String s : IstekMesaj)
{
    if (!s.equals(""))
    {
        listDersGunler.add(s);
    }
}
IstekMesaj = listDersGunler.toArray(new
String[listDersGunler.size()]);

ArrayList<String> listDersSaatler = new ArrayList<String>();
for (String s : IstekTarih)
{
    if (!s.equals(""))
    {
        listDersSaatler.add(s);
    }
}
IstekTarih = listDersSaatler.toArray(new
String[listDersSaatler.size()]);
ArrayList<String> listDersIstekYapan = new ArrayList<String>();
for (String s : IstekYapan)
{
    if (!s.equals(""))

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

        {
            listDersIstekYapan.add(s);
        }

    }
    IstekYapan = listDersIstekYapan.toArray(new
String[listDersIstekYapan.size()]);
    setListAdapter(new ArrayAdapter<String>(gorusmeistekleri.this,
android.R.layout.simple_list_item_1, GMesaj));
}
protected void onListItemClick(ListView l, View v, int position, long
id)
{
    super.onListItemClick(l, v, position, id);
    String Add= Ad[position];
    String Soy= Soyad[position];
    String IM= IstekMesaj[position];
    String IT= IstekTarih[position];
    String IY= IstekYapan[position];
    Intent i= new Intent(gorusmeistekleri.this,
ogrenciyemesajgonder.class);
    Bundle bundle = new Bundle();
    bundle.putString("Ad", Add);
    bundle.putString("Soyad", Soy);
    bundle.putString("Mesaj", IM);
    bundle.putString("Tarih", IT);
    bundle.putString("TcKimlikNo", IY);
    i.putExtras(bundle);
    startActivity(i);
}}

```

Menü Sayfası

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
</LinearLayout>

```

```

public class menu extends ListActivity
{
    private Toast toast;
    ConnectivityManager Baglanti = null;
    String sayfalar[]={"Kullanıcı Kayıt"};
    String sayfalar[]={"Kullanıcı Kayıt","Mesaj Gönder","Gelen
Mesajlarım"};
    String rsayfalar[]={"kayit","derslerim","gorusmeistekleri"};
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        final SharedPreferences DosyaAyar =
getSharedPreferences("kbsFile",MODE_PRIVATE);

```


EK 2. Android Uygulama Kodları (Devam Ediyor)

```

final SharedPreferences.Editor DosyaDuzen = DosyaAyar.edit();
    if(AgKontrol())
    {}

    else
    {
        Toast.makeText(getApplicationContext(), "Bağlantı Yok",
Toast.LENGTH_LONG).show();
    }
    if(DosyaAyar.getString("KullaniciID", "").length()==36)
    {
        setListAdapter(new ArrayAdapter<String>(menu.this,
android.R.layout.simple_list_item_1, sayfalar));
    }
    else
    {
        Toast.makeText(getApplicationContext(), "Tanımlı kullanıcı
bulunamadı!", Toast.LENGTH_SHORT).show();
        setListAdapter(new ArrayAdapter<String>(menu.this,
android.R.layout.simple_list_item_1, sayfalar));
    }
}
protected void onListItemClick(ListView l, View v, int position, long
id)
{
    super.onListItemClick(l, v, position, id);
    String cheses= rsayfalar[position];
    try
    {
        Class OurClass = Class.forName("com.kbs.tez."+cheses);

        Intent ourIntent = new Intent(menu.this, OurClass);
        startActivity(ourIntent);
    }
    catch (ClassNotFoundException e)
    {}
}
public boolean AgKontrol()
{
    Baglanti = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    if(Baglanti.getActiveNetworkInfo()!=null)
    {
        if(Baglanti.getActiveNetworkInfo().isConnected())
        {
            return true;
        }
    }
    return false;
}
}

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

Mesaj Gönderme Sayfası

```

public class ogrenciyemesajgonder extends Activity
{
    TextView tvBilgi;
    Button btnGonder;
    EditText etMesaj;
    public void onCreate(Bundle ogrmesaj)
    {
        super.onCreate(ogrmesaj);
        setContentView(R.layout.ogrenciyemesajgonder);
        tvBilgi = (TextView) findViewById(R.id.tvBilgi);
        btnGonder = (Button) findViewById(R.id.btnGonder);
        etMesaj = (EditText) findViewById(R.id.etMesaj);
        final Bundle extras = getIntent().getExtras();
        tvBilgi.setText("\n Aşağıda yazmış olduğunuz mesaj " +
extras.getString("TcKimlikNo") + " kimlik numaralı " + extras.getString("Ad") + "
"+extras.getString("Soyad")+" adlı kişiye gönderilecektir.\n\n");
        btnGonder.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
                SoapObject Istek = new
SoapObject("http://tempuri.org/", "KisiyeMesajGonder");

                Istek.addProperty("TcKimlikNo",
extras.getString("TcKimlikNo"));
                Istek.addProperty("Mesaj",
etMesaj.getText().toString());
                SoapSerializationEnvelope Paket = new
SoapSerializationEnvelope(Paket.VER11);
                Paket.dotNet = true;
                Paket.setOutputSoapObject(Istek);
                HttpTransportSE Servis = new
HttpTransportSE("http://192.168.1.55/servisler/WebService.asmx");
                try
                {

                    Servis.call("http://tempuri.org/KisiyeMesajGonder", Paket);
                }
                catch (IOException e)
                {
                    e.printStackTrace();
                }
                catch (XmlPullParserException e)
                {
                    e.printStackTrace();
                }
                Toast.makeText(ogrenciyemesajgonder.this, "Mesajınız
sınıfa gönderilmiştir.", Toast.LENGTH_SHORT).show();
                etMesaj.setText("");
                Intent intent = new Intent();
                setResult(RESULT_OK, intent);
                finish();
            }
        }
    }
}

```

EK 2. Android Uygulama Kodları (Devam Ediyor)

```

    });
}

}

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/tvBilgi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:id="@+id/etMesaj"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:text=""
        android:maxLength="160"
        android:lines="5"
        android:singleLine="false">
        <requestFocus />
    </EditText>
    <Button
        android:id="@+id/btnGonder"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Gönder" />
</LinearLayout>

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Yusuf MUŞTU
Doğum Yeri ve Tarihi : Samsun, 17.10.1985

Eğitim Durumu

Lisans Öğrenimi : Sakarya Üniversitesi, Teknik Eğitim Fakültesi,
Elektronik ve Bilgisayar Öğretmenliği (2009)

İş Deneyimi

Çalıştığı Kurumlar : 1) Sakarya Üniversitesi, Bilgi İşlem Dairesi Başkanlığı
(2005-2009)
2) Bilecik Üniversitesi, Bilgi İşlem Dairesi Başkanlığı
(2009-)

İletişim

Adres : Bilecik Üniversitesi, Bilgi İşlem Dairesi Başkanlığı,
Gülümbe Kampüsü, BİLECİK
Tel: : 0228.2141150
E-Posta Adresi : yusuf.mustu@bilecik.edu.tr

Tarih: / /

İmza