



**T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



YÜKSEK LİSANS TEZİ

**MOBİL FİNANS SERVİSLERİ İÇİN GÜVENLİ YAKIN SAHA
İLETİŞİMİ DESTEKLİ TELEFON İSTEMCİSİ TASARIMI VE
GELİŞTİRİLMESİ**

Halil İbrahim İBALI

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN

Prof. Dr. Abdül Halim ZAIM

II. DANIŞMAN

Yrd. Doç. Dr. Muhammed Ali AYDIN

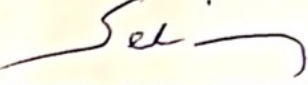
Haziran, 2013

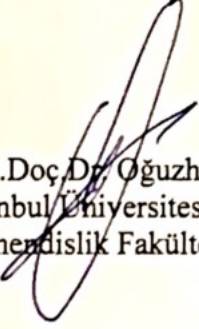
İSTANBUL

2601100002 Öğrenci numaralı Halil İbrahim İBALI tarafından hazırlanan bu çalışma 14/06/2013 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi


Prof. Dr. A. Halim ZAIM (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Selim AKYOKUŞ
Doğuş Üniversitesi
Mühendislik Fakültesi


Yrd. Doç. Dr. Oğuzhan ÖZTAŞ
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Ahmet SERTBAŞ
İstanbul Üniversitesi
Mühendislik Fakültesi


Doç. Dr. Hakan DOĞAN
İstanbul Üniversitesi
Mühendislik Fakültesi



20.04.2016 tarihli resmi gazetede yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin aboneli olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Bu tez, Üniversite-Sanayi işbirliği kapsamında T.C. Bilim, Sanayi ve Teknoloji Bakanlığı San-Tez programının 00810.STZ.2011-1 numaralı projesi ile desteklenmiştir.

ÖNSÖZ

Bu çalışma, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında yapılan “Mobil Finans Servisleri İçin Güvenli Yakın Saha İletişimi Destekli Telefon İstemcisi Tasarımı ve Geliştirilmesi” adlı yüksek lisans tez çalışmasını içermektedir.

Tez konusu, İstanbul Üniversitesi ve Alcatel-Lucent Teletaş Telekomünikasyon A.Ş. ortaklığında, SAN-TEZ kapsamında proje olarak tanımlanıp desteklenmiş ve bu kapsamda tamamlanmıştır.

Lisans ve yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocalarım Prof. Dr. Abdül Halim ZAIM’e ve Yar. Doç. Dr. Muhammet Ali AYDIN’a en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca her türlü desteği veren çalışma arkadaşım Mehmet Ali ERTÜRK’e ve çalışmamın uygulama kısmını destekleyen Alcatel-Lucent Türkiye ekibine teşekkürü borç bilirim.

Haziran 2013

Halil İbrahim İBALI

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	vii
TABLO LİSTESİ.....	x
SİMGE VE KISALTMA LİSTESİ	xi
ÖZET	xii
SUMMARY	xiii
1. GİRİŞ	1
2. GENEL KISIMLAR.....	4
2.1. TEZİN GEREKÇESİ VE HEDEFLERİ	4
2.2. TEZİN AR-GE NİTELİĞİ VE TEKNOLOJİK YÖNÜ	7
3. MALZEME VE YÖNTEM.....	10
3.1. MOBİL UYGULAMA MİMARİLERİ	10
3.1.1. Blackberry Mimarisi.....	10
3.1.2. Android İşletim Sistemi Mimarisi	15
3.2. NFC (YAKIN SAHA İLETİŞİMİ) SİSTEMİ.....	22
3.2.1. NFC Uygulamaları	25
3.2.2. NFC Uygulamalarının Faydaları	27
3.2.3. NFC Teknolojisini Uygulama Şekilleri.....	28
3.2.4. NFC Mobil Ödeme Uygulamaları	30
3.2.5. UICC (Küresel Bütünleşik Devre Kartı) Kart Mimarisi	33
3.2.5.1. NFC Standartları	33
3.2.5.2. UICC – NFC Modül Arayüzü	34
3.2.5.3. Application Processor (Uygulama İşlemcisi) – UICC Interface (Arayüzü).....	34
3.2.5.4. Application Processor (Uygulama İşlemcisi) – NFC Modül Arayüzü	35
3.2.5.5. OTA (Hava Üzerinden İletişim) Teknolojisi Standartları ve Teknolojileri	35
3.2.6. NFC (Yakın Saha İletişimi) Framework (Çerçevesi)	36
3.2.7. Java Kart Teknolojisi.....	36

4. BULGULAR.....	40
4.1. MWS- GÜVENLİ ARABİRİM UYGULAMASI TASARIMI.....	40
4.2. G&D- GÜVENLİ ARABİRİM UYGULAMASI TASARIMI.....	44
4.3. ALCATEL LUCENT MWS KULLANAN MOBİL İSTEMCİ TASARIMI.....	46
4.3.1. Mobil Telefonda Uygulamanın Kullanıcı Arayüzlerinin Geliştirilmesi ve Test Edilmesi (Blackberry).....	52
4.3.2. Mobil Telefonda Uygulamanın Kullanıcı Arayüzlerinin Geliştirilmesi ve Test Edilmesi (Google Nexus S (Android))	59
4.3.3. Sanal POS Uygulaması.....	64
5. TARTIŞMA VE SONUÇ	70
KAYNAKLAR.....	73
EKLER	76
EK 1. Güvenli Arabirim Kaynak Kodları.....	76
EK 2. Güvenli Arabirim Test Uygulaması Kaynak Kodları	82
EK 3. Blackberry Mobil Cüzdan Uygulaması Kod Dökümantasyonu.....	87
EK 4. Android Mobil Cüzdan Uygulaması Kod Dökümantasyonu	95
ÖZGEÇMİŞ	99

ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1: Alcatel-Lucent Mobil Cüzdan Servisi Altyapısı.....	8
Şekil 3.1: Blackberry Platformu Mimarisi.....	11
Şekil 3.2: Android İşletim Sistemi Mimarisi.....	17
Şekil 3.3: Okuyucu Kipi.....	24
Şekil 3.4: Yazıcı Kipi.....	24
Şekil 3.5: Kart Öykünüm Kipi.....	25
Şekil 3.6: Birebir İletişim Kipi.....	25
Şekil 3.7: NFC Sim Kart.....	28
Şekil 3.8: NFC Mikro SD Kart.....	29
Şekil 3.9: Pasif NFC Kart.....	30
Şekil 3.10: Pasif NFC Kart Uygulaması.....	30
Şekil 3.11: Mobil Telefon Üreticisi Merkezli NFC Uygulaması Yaklaşımı.....	31
Şekil 3.12: GSM Servis Sağlayıcısı Merkezli NFC Uygulaması Yaklaşımı.....	32
Şekil 3.13: UICC Kart Mimarisi [33].....	33
Şekil 3.14: NFC Standartları [33].....	34
Şekil 3.15: Yakın Saha İletişimi Çerçevesi [33].....	36
Şekil 3.16: Java Kart Mimarisi [35].....	37
Şekil 3.17: Java Kart APDU İletişimi [35].....	38
Şekil 3.18: APDU Yapısı [35].....	39
Şekil 4.1: Güvenli Arabirim İşlemleri.....	40
Şekil 4.2: Güvenli Arabirim Sınıf Yapısı.....	41
Şekil 4.3: PIN Güncelleme İşlemi Akış Şeması.....	42
Şekil 4.4: Kimlik Bilgisi Güncelleme İşlemi Akış Şeması.....	43

Şekil 4.5: Ödeme İşlemi Akış Şeması.....	44
Şekil 4.6: Kullanıcı İşlemleri.	47
Şekil 4.7: Login İşlemi.....	48
Şekil 4.8: Çıkış İşlemi.....	48
Şekil 4.9: Kullanıcı Hesap Bilgileri İşlemi.	49
Şekil 4.10: Hesap Bilgileri İsteme İşlemi.	49
Şekil 4.11: Hesap Hareketleri İsteme İşlemi.....	50
Şekil 4.12: Para Transferi Akış Şeması.	50
Şekil 4.13: Kullanıcıdan Kullanıcıya Para Transferi Şeması.....	51
Şekil 4.14: Sınıf Yapısı.	52
Şekil 4.15: Para Kartlar Arayüzü.	53
Şekil 4.16: Kuponlar ve Sadakat Kartlar Arayüzü.....	54
Şekil 4.17: Kullanıcı Profili Arayüzü.	54
Şekil 4.18: NFC Durum Arayüzü.....	55
Şekil 4.19: İşlemler Arayüzü.....	56
Şekil 4.20: Para Kart Geçmiş İşlemler Arayüzü.	56
Şekil 4.21: Para Kart Geçmiş İşlemler Arayüzü.	57
Şekil 4.22: Menü Kullanım Arayüzü.	57
Şekil 4.23: Para Kart Geçmiş İşlemler Arayüzü.	58
Şekil 4.24: Hesap Durum Arayüzü.	58
Şekil 4.25: Hesaba Para Ekleme Arayüzü.	59
Şekil 4.26: Kullanıcı Giriş Sayfası Arayüzü.	60
Şekil 4.27: Kullanıcı Profil Arayüzü.....	60
Şekil 4.28: Mobil Cüzdan Ana Menü Arayüzü.....	61
Şekil 4.29: Para Kart Arayüzü.	61
Şekil 4.30: Para Kart Geçmiş İşlemler Arayüzü.	62
Şekil 4.31: Başka Kullanıcıya Para Transfer Arayüzü.....	62

Şekil 4.32: Hesaba Para Ekleme Arayüzü.	63
Şekil 4.33: Opet Sadakat Kart Arayüzü.	63
Şekil 4.34: Hesap Ödeme Arayüzü.	64
Şekil 4.35: NFC Okuyucu Ödeme Tutarı Görüntüsü.	65
Şekil 4.36: NFC Okuyucu Ödeme İşlemi Görüntüsü.	66
Şekil 4.37: NFC Okuyucu Ödeme İşlemi Sonucu Görüntüsü.	67
Şekil 4.38: NFC Okuyucu Ödeme İşlemi Hatası Görüntüsü.	68
Şekil 4.39: Para Kart Geçmiş İşlemler Görüntüleme Arayüzü (Blackberry).....	69



TABLO LİSTESİ

Sayfa No

Tablo 2.1: NFC'nin Kısa Mesafe İletişim Teknolojileriyle Karşılaştırılması5

Tablo 4.1: APDU Mesaj Yapısı.....45



SİMGE VE KISALTMA LİSTESİ

Kisaltmalar	Açıklama
AL	: Alcatel Lucent
APDU	: Application Protocol Data Unit (Uygulama Protokol Veri Birimi)
API	: Application Programming Interface (Uygulama Programlama Arayüzü)
CA	: Certificate Authorities (Sertifika Sağlayıcıları)
ECMA	: European Computer Manufacturers Association (Avrupa Bilgisayar Üreticileri Birliği)
ETSI	: European Telecommunications Standards Institute (Avrupa Telekomünikasyon Standartları Enstitüsü)
G&D	: Giesecke & Devrient
GSM	: Global System for Mobile Communications (Mobil İletişim için Küresel Sistem)
HCI	: Host Controller Interface (Ana Bilgisayar Kontrol Arayüzü)
ISO	: International Organization for Standardization (Uluslararası Standartlar Teşkilatı)
IT	: Information Technologies (Bilgi Teknolojileri)
JCVM	: Java Card Virtual Machine (Java Kart Sanal Makinesi)
ME	: Mobile Edition (Mobil Versiyon)
MNO	: Mobile Network Operators (Mobil Ağ Operatörleri)
MWS	: Mobile Wallet Services (Mobil Cüzdan Servisleri)
NFC	: Near Field Communication (Yakın Alan-Saha İletişimi)
OS	: Operating System (İşletim Sistemi)
OTA	: Over The Air (Hava Üzerinden İletişim)
PDA	: Personal Digital Assistant (Kişisel Dijital Yardımcı)
PIN	: Personal Identification Number (Kişisel Kimlik Numarası)
POS	: Point Of Sale (Ödeme Noktası)
RFID	: Radio Frequency Identification (Radyo Frekansı ile Tanımlama)
SCWS	: Smart Card Web Server (Akıllı Kart Örün Sunucusu)
SD	: Secure Digital (Güvenli Dijital)
SDK	: Software Development Kit (Yazılım Geliştirme Aracı)
SE	: Secure Element (Güvenli Arabirim)
SIM	: Subscriber Identity Module (Abone Kimlik Modülü)
SMS	: Short Message Service (Kısa Mesaj Servisi)
SVA	: Stored Value Account (Saklanan Değer Hesabı)
TTP	: Trusted Third Parties (Güvenilir Üçüncü Kuruluşlar)
UICC	: Universal Integrated Circuit Card (Küresel Bütünleşik Devre Kartı)
VM	: Virtual Machine (Sanal Makine)
WiFi	: Wireless Fidelity (Kablosuz Bağlantı Alanı)

ÖZET

YÜKSEK LİSANS TEZİ

MOBİL FİNANS SERVİSLERİ İÇİN GÜVENLİ YAKIN SAHA İLETİŞİMİ DESTEKLİ TELEFON İSTEMCİSİ TASARIMI VE GELİŞTİRİLMESİ

Halil İbrahim İBALI

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Abdül Halim ZAİM

II. Danışman : Yrd. Doç. Dr. Muhammet Ali AYDIN

Bu tez çalışmasında, cep telefonlarının mobil cüzdan (mobile wallet) ve mobil finans servisleri sunan araç olarak kullanılabilmesi ve NFC teknolojisine özgü güvenli iletişim ve bilgi saklama problemlerinin çözülebilmesi için güvenli telefon istemcisi tasarlanmış ve geliştirilmiştir. Bu çalışmada mobil cüzdan servisi, istenen şekilde yapılandırılabilme ve genişletilebilme imkânına sahip bir ödeme altyapısı üzerinde bulunan çeşitli ödeme uygulamalarından meydana gelen bir sistemdir. Bir başka ifadeyle, mobil cüzdan servisi, GSM/3G servis sağlayıcıları, bankalar, kredi kartı firmaları (örnek; Mastercard, Visa vs.) ve kurumlar için sanal kredi kartı ve nakit para kullanımına izin veren önödemeli Saklanan Değer Hesabı (SVA-Stored Value Account) modelini kullanan mobil ticaret ve pazarlama uygulamaları, para transferi, hesap durumu ve yapılan işlem geçmişinin görüntülenmesi gibi kullanıcı ve satıcı için arayüzler ve yönetim uygulamaları sunan yüksek teknoloji uygulamasıdır. Bu uygulama ile kullanıcı hesaplarına, kredi kartları ve banka hesapları kullanılarak yükleme yapılabilir. Bu şekilde GSM/3G servis sağlayıcıları ve kurumlar, müşterilerine gerçek paranın kullanıldığı bir mobil cüzdan servisi sunabilme imkânına sahip olurlar.

Haziran 2013, 112 sayfa.

Anahtar kelimeler: NFC, mobil cüzdan, yakın saha iletişimi.

SUMMARY

M.Sc. THESIS

SECURE NEAR FIELD COMMUNICATION ENABLED PHONE CLIENT DESIGN AND DEVELOPMENT FOR MOBILE FINANCIAL SERVICES

Halil İbrahim İBALI

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor : Prof. Dr. Andül Halim ZAİM

Co-Supervisor : Assist. Prof. Dr. Muhammet Ali AYDIN

In this thesis, a secure NFC-enabled phone client has been designed and developed to solve the intrinsic communication and information storage problems of NFC technology and thus, to use cellular phones as mobile wallet. Here, the mobile wallet service (MWS) refers to a system, which consists out of a configurable and extensible payment framework and various applications. In other words, the MWS and MFS are high-tech applications to provide proximity payment services to GSM/3G operators and enterprises using a prepaid SVA (Stored Value Account) model, which offers several client/supplier interfaces and management applications, such as mobile e-commerce and marketing applications, money transfer, balance and transaction history view, etc. This enables GSM operators and enterprises to deliver proximity payment to their customers via the mobile wallet service offering.

June 2013, 112 pages.

Keywords: NFC, mobile wallet, near field communication.

1. GİRİŞ

Yakın Alan İletişimi, (NFC) Kimlik Kartları- Temassız Akıllı Kartlar standartlarına uygun olarak, iki akıllı cihazın kolay ve güvenilir bir şekilde iletişim kurmasını sağlayan kısa mesafe kablosuz iletişim teknolojisidir. Radyo Frekansı ile Tanımlama (Radio Frequency Identification, RFID) tabanlı olan NFC teknolojisi, yüksek frekansta (13.56 MHz) ve düşük bant genişliğinde (en çok 424Kbit/s) güvenli bir ara yüz üzerinden veri haberleşmesini sağlamaktadır [1].

NFC teknolojisi, RFID üzerinde çalışan diğer kablosuz iletişim teknolojilerine (Bluetooth, WiFi) nazaran daha kısa mesafede ve daha düşük bir bant genişliğinde veri iletişimi sağlamaktadır. NFC iletişimi, iki NFC uyumlu akıllı aygıtın birbirine birkaç santimetre (0–4 cm) yaklaştırılması ile oluşur. NFC teknolojisinin çok yakın mesafede veri iletişimi sağlıyor olması, güvenlik konusunda önemli avantaj sağlamaktadır [2].

NFC teknolojisi 2002 yılının son çeyreğinde Sony ve Philips ortaklığında geliştirilmiş ve 2002 yılının son ayında Avrupa'nın ECMA Standartlar Birliği tarafından kabul edilmiştir. 8 Aralık 2003 tarihinde de ISO/IEC tarafından bir standart olarak kabul edilmiştir. ISO/IEC ve ECMA tarafından NFC için ISO/IEC 18092/ECMA–340 NFCIP–1 ve ISO/IEC 21481/ECMA–352 NFCIP–2 olmak üzere iki ayrı standart geliştirilmiştir [3]. Bu yeni nesil teknolojinin geliştirilmesi ve yaygınlaştırılması amacı ile de 18 Mart 2004 tarihinde Nokia, Sony ve NXP tarafından NFC Forum kurulmuştur [1].

Günümüzde NFC teknolojisinin temassız ödeme, uzaktan bilet satın alma, kişiden kişiye para transferi, müşteri sadakat ve hediye uygulamaları, mobil ticaret ve pazarlama alanlarında çeşitli uygulamaları vardır [4]. Bununla birlikte, şu anda mevcut veya tasarlanmakta olan NFC uygulamalarının gerçekleştirilebilmesi mobil telefon mimarisinde güvenilirliğine bağlıdır.

Cep telefonları şu anda en yaygın mobil bilgi işlem cihazlarıdır ve toplumsal hayatta büyük bir yer edinmiştir. Radyo Frekansı Kimlik Doğrulama (RFID – Radio Frequency Identification)'nın bir uzantısı olan NFC teknolojisinin mobil kaynaklarla fiziksel objeleri bir araya getirme

potansiyeli vardır [5]. NFC uygulamalarında ortaya çıkan en önemli problem alışveriş bilgilerinin tutulduğu güvenlik biriminin (secure element'in) nerede saklanacağıdır. Genel olarak, literatürde bu sorun iki farklı şekilde çözülmeye çalışılmıştır:

- i) mobil cihazın içine harici bir çip olarak yerleştirilmesi (SE-NFC);
- ii) güvenlik biriminin sim kart üzerinde saklanması (SIM-NFC) [1-6].

Cep telefonlarında NFC teknolojisinin kullanımı, genelde bir çip üzerinden (SE-NFC) sağlanmaktadır. Bu kullanım şeklinin getirdiği bazı dezavantajlar bulunmaktadır. Bunlar; telefon değiştirildiğinde her kredi kartı için yapılan ayarların tekrar tekrar yapılmak zorunda kalınması, çoğu standardizasyona tabi tutulmaması vb. sorunlar son kullanıcıyı rahatsız edebilecek kullanım zorlukları getirmektedir. Ancak, sim kart üzerinden sağlanan NFC hizmeti (SIM-NFC), çok daha sağlıklı ve güvenilir çalışmaktadır. Gemalto şirketinin bulduğu ve ETSI tarafından standardize edilmiş olan SWP (Single Wire Protocol) protokolü sayesinde, NFC çip ve sim kart arasındaki iletişim güvenilirliği artmaktadır [7]. Ayrıca, mobil cüzdan hizmetinde kullanılacak olan üçüncül kişilerin, büyük GSM operatörleri olacak olması, ileride ortaya çıkabilecek olan sorunları minimuma indirirken, herhangi bir problem çıktığında, son kullanıcının karşısında büyük ve deneyimli bir şirket bulacak olması, son kullanıcı açısından büyük önem teşkil etmektedir.

Günümüzde yakın saha iletişimi (NFC- Near Field Communication) teknolojisinin ticari alanlarda yukarıda bahsedildiği şekilde çeşitli uygulamalarının bulunmasıyla birlikte, NFC teknolojisinin pratik uygulamalarını gerçekleştirmek için hatasız ve güvenilir iletişimi engelleyen sorunların çözülmesi gerekmektedir.

Bu tezde, cep telefonlarının mobil cüzdan (mobile e-wallet) olarak kullanılabilmesi ve NFC teknolojisine özgü güvenli iletişim ve bilgi saklama problemlerinin çözülebilmesi için güvenli telefon istemcisi tasarlanacak ve geliştirilecektir. Bu çalışmanın genel amaçları ve erişilmek istenen sonuçlar aşağıda kısaca belirtilmiştir:

- SIM kart üzerindeki güvenli birimle veya telefon üzerindeki güvenli birimle çalışmayı destekleyen Java ME uygulamasının geliştirilerek cep telefonları üzerinden etkin ve güvenli mobil cüzdan uygulaması tasarlanması ve geliştirilmesi,

- Alcatel-Lucent Touchatag veya başka mobil cüzdan servisleri ve mobil finansal servisler kapsamında sunulmuş olan tüm servislerin güvenli ve hatasız bir şekilde kullanılmasını sağlayan, yaygın kullanılabilirlik yönünden performans ve fonksiyonel testlerinin tamamlanarak tez çalışması sonuçlarının ülkemize katma değer katacak bir ürüne dönüştürülmesi,
- Tez çalışmaları süresince elde edilen bilgi birikiminin ve çalışma sonuçlarının uluslararası hakemli dergilerde ve konferanslarda yayınlanması.



2. GENEL KISIMLAR

2.1. TEZİN GEREKÇESİ VE HEDEFLERİ

Günümüzdeki cep telefonları, NFC teknolojisinin ticari uygulamalarını destekleyecek hale getirebilmek için, NFC çipleriyle donatılmaya başlanmıştır. Bu çiplerin telefonlar içine yerleştirilmesinin sebebi ise, telefonların artık vazgeçemeyeceğimiz aksesuarlar olması ve neredeyse 7’den 70’e herkeste bulunmasıdır. Bu sayede, NFC uygulamaları cep telefonu sahibi olan herkes tarafından kullanılabilir seviyeye inmekte ve yaygın bir kullanıma sahip ticari uygulamalar geliştirilmesine olanak sağlamaktadır.

Mobil cüzdan uygulamasının NFC uygulamaları kullanılarak yapılmak istenmesinin nedeni ise, herkesin kullandığı cep telefonlarının bir kredi kartı veya nakit para taşıyan cüzdan gibi kullanılarak insanlara kolaylık sağlanması ve son kullanıcıların NFC uygulamalarının getirdiği potansiyel avantajlardan yararlanarak hayatlarını daha kolay bir hale getirmektir. Bu uygulamalar sayesinde, kullanıcılar kolayca nakit veya kredi kartı ile alışveriş yapabilecekler ve para taşımak veya uzun kredi kartı onay işlemlerini beklemek zorunda kalmayacaklardır. NFC teknolojisi kısa mesafeye sahip olan kablosuz iletişim teknolojileri arasında önemli bir yere sahiptir. Aşağıdaki tabloda görüldüğü üzere, hızlı kurulum zamanı olması, kullanım alanlarının çokluğu ve kullanım kolaylığından dolayı yaygınlaşma potansiyeli yüksek olan bir teknolojiye sahiptir [1].

Tablo 2.1: NFC'nin Kısa Mesafe İletişim Teknolojileriyle Karşılaştırılması.

	NFC	RFID	IrDa	Bluetooth
Kurulum Zamanı	<0.1 ms	<0.1 ms	~0.5 ms	~6 sec
İletişim Mesafesi	10 cm'ye kadar	3 m'ye kadar	5 m'ye kadar	30 m'ye kadar
Kullanılabilirlik	Kişi merkezli, Basit, Hızlı	Ürün merkezli, Basit	Veri merkezli, Basit	Veri merkezli, Orta zorlukta
Kullanım Alanları	Ödeme, Giriş izni, Paylaşım, Kolay kurulum	Ürün takibi	Kontrol ve Veri değişimi	Veri değişimi için ağ kurulumu, Mikrofon-Kulaklık takımı
Kullanım Şekli	Dokunarak, Geçiş yaparak, Kolay bağlantı	Bilgi alışverişi	Basit	Konfigürasyon gereklidir

NFC teknolojisi 2004'ten bu yana NFC topluluğunun (NFC Forum) kurulmasıyla birlikte aktif bir şekilde geliştirilmektedir. İlk çıktığında, sadece basit RFID (Radio Frequency Identification – Radyo Frekans ile Tanımlama) uygulamalarını biraz daha geliştirmek amacıyla çeşitli NFC çalışmaları yapılmıştır. Fakat daha sonra ticari ürün potansiyeli fark edilince, daha kapsamlı uygulamalarda kullanılmaya başlanılmıştır. Bunlardan en önemlisi, cep telefonlarının mobil cüzdan olarak kullanılmasının amaçlanmasıdır.

Günümüzde, NFC destekli telefonların içinde bulunan NFC çipleri programlanarak temassız (contactless) uygulamaların çalışması sağlanabilmektedir. Bu tip uygulamaların avantajı ise,

diğer tipteki kartlar gibi uzun süren işlem zamanına sahip olmamalarıdır. Mesela; bir kredi kartıyla ödeme yapmak istendiğinde, yaklaşık olarak işlem en az 2 dakika sürmektedir. Fakat bu işlem temassız uygulamaya sahip olan bir kart ya da cihazla yapıldığında, işlem süresi yaklaşık olarak 1 saniyedir. Temassız kartlar farklı formlarda olabilirler, fakat bu uygulamalara en uygun olanı NFC çipleri ve bunları kullanan cihazlardır.

NFC uygulamalarını gerçekleştirmenin iki farklı yolu vardır. Bunlar; sadece telefon içindeki NFC çipinin kullanılması veya sim kartını kullanarak GSM operatörleri aracılığı ile banka işlemlerinin yapılmasını sağlamaktır. Sim kart üzerinden NFC uygulamalarının gerçekleştirilmesi birçok avantaj sağlamaktadır. Özellikle, telefon cihazına bağımlılığı ortadan kaldırması en büyük getirisidir. Ayrıca, GSM operatörlerinin dâhil olması ve bankalarla iş birliği içerisinde çalışmaları, son kullanıcılar açısından büyük yarar ve güven sağlamaktadır.

Sonuç olarak; bu çalışmada Alcatel-Lucent mobil cüzdan servisi kapsamında sunulmuş olan tüm servislerin güvenli ve hatasız bir şekilde kullanılmasını sağlayan, yaygın kullanılabilirlik yönünden performans ve fonksiyonel testlerinin tamamlanarak çalışma sonuçlarının ülkemize katma değer katacak bir ürüne dönüştürülmesi hedeflenmiştir.

Kısaca, bu çalışma aşağıdaki aşamaları kapsamaktadır:

- Java ME teknolojisi kullanılarak güvenli birimi SIM kart (UICC- Universal Integrated Circuit Card) üzerinde veya telefon cihazı üzerinde bulunan NFC destekli telefonlar için istemci uygulama geliştirilmesi.
- Para yükleme, para transferi, temassız ve mobil ödeme, işlem geçmişi görüntülenmesi, kupon ve puan gibi müşteri sadakat uygulamalarının görüntülenmesi ve kullanılması gibi Mobil Cüzdan Servisi tarafından sağlanan tüm uygulamalara güvenli ve sorunsuz erişimin geliştirilmesi.
- Satış noktası cihazı üzerinden mobil cüzdan servisi ile etkileşimli ödeme modeli ve senaryosunun (ödeme şeklinin seçilmesi gibi) ileri analizi ve gerçekleşmesi.

- Yüksek tutarlı ödemeler için şifre tabanlı ödeme. Satış noktası cihazı veya telefon tabanlı olarak telefon veya SIM kart (UICC) üzerindeki güvenli birimde saklanan şifre üzerinden gerçekleştirilecek ödeme mimarisinin ve senaryosunun gerçekleşmesi.

Çalışma bitiminde ortaya çıkacak ürünle beraber, insanların hayatı daha da kolaylaşacak ve bu uygulamanın sağladığı kolaylık, güvenlik ve esneklik sayesinde insanların alışveriş alışkanlıkları değişecektir. Yaygın kullanım potansiyeline ilaveten, dünyada yüksek güvenli mobil cüzdan uygulamalarının henüz geliştirilme aşamasında olması nedeniyle, bu ürünün yurtdışına ihraç edilebilme potansiyeli vardır. Dolayısıyla, bu tez çalışmaları ile gerçekleştirilecek ürünün Türkiye ekonomisine önemli bir ihraç ürünü olarak katkıda bulunması beklenmektedir.

2.2. TEZİN AR-GE NİTELİĞİ VE TEKNOLOJİK YÖNÜ

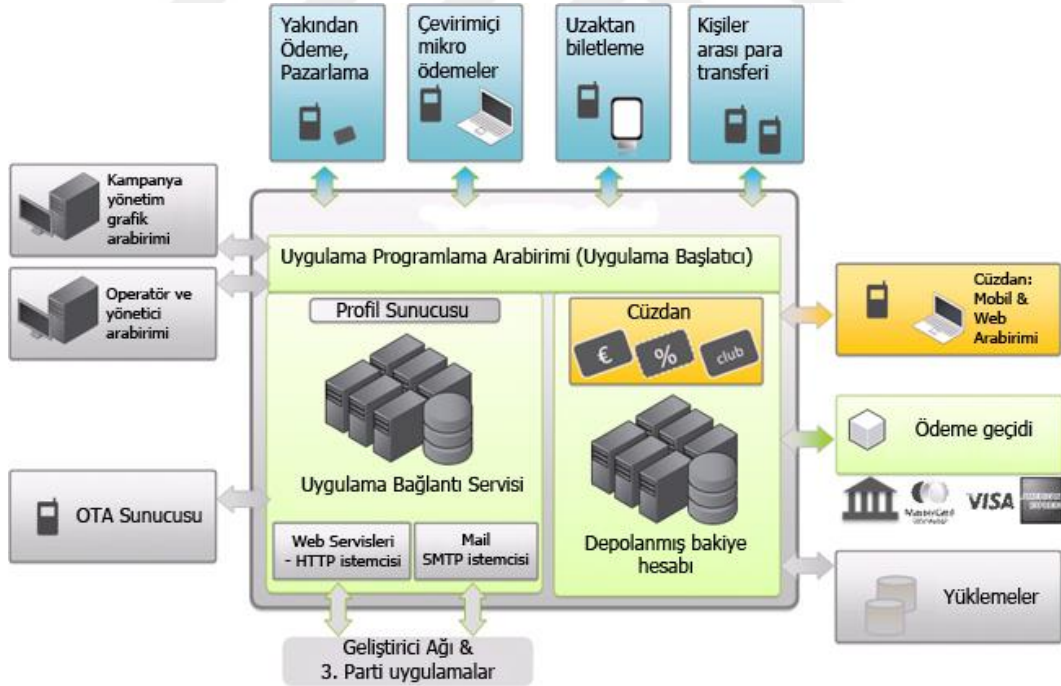
Son yıllarda, ticarete, ürün kontrolünde, şehir ve sağlık uygulamalarında kullanılmak üzere NFC (Near Field Communication – Yakın Saha İletişimi) teknolojisinin geliştirilmesi ve gerçekleştirilmesi konusunda endüstride ve akademide olağanüstü hızla artan bir ilgi olmuştur. Bununla beraber bu teknolojiye önemli bir yere sahip olan mobil cüzdan uygulaması her ne kadar teoride kullanılabilirliği gösterilse de pratikte sadece belli yerlerde kullanılmıştır. Mobil cüzdan uygulamasıyla ilgili yayınlanmış yayınlarda (makale, bildiri, vb.) güvenlik biriminin (secure element-SE) nerede saklanmasından, iletişim standartlarından, hangi telefon üreticilerinin bu teknolojiyi destekleyen telefonları ürettiği veya geliştirme aşamasında olduğu gibi konulardan bahsedilmiştir.

Genel olarak, mobil cüzdan servisi, istenen şekilde yapılandırılabilme ve genişletilebilme imkânına sahip bir ödeme altyapısı üzerinde bulunan çeşitli ödeme uygulamalarından meydana gelen bir sistemdir. Mobil cüzdan uygulamaları geliştirilirken, pek çok farklı modülün aynı anda ve farklı zamanlarda, hiçbir problem olmadan uyum içinde çalışması gerekmektedir. Mobil cüzdan uygulamaları, mantık olarak her ne kadar basit bir yapı olarak düşünülebilecek yapıda olsalar da, dikkat edilmesi gereken pek çok husus vardır. Özellikle, bu uygulamalar sim kart üzerinden gerçekleştirildiğinde, durum daha karmaşık bir hal almaktadır.

Alcatel-Lucent Mobil Cüzdan Servisi (MWS); operatörlere ve işletmelere, esnek ödeme ve

sadakat, kupon gibi pazarlama hizmetleri sunan servislerden oluşan bir birimdir. Bu servis, mobil cüzdan gibi genel uygulama etkinleştiricileriyle mobil ödeme çerçevesinden ve platform tarafından etkinleştirilmiş çeşitli mobil ödeme uygulamalarından oluşur. Ödeme çerçevesi, temassız, SMS, mobil uygulama ve web gibi çeşitli kaynaklardan tetikleyicileri bulundurarak, iş akışına ve görev dağıtıcısına bağlı olarak doğru hareketi başlatır. Ödeme çerçevesi tarafından etkinleştirilen ödeme uygulamaları, sadakat programlarını ve pazarlamayı içeren yakın ödeme, kişiden kişiye para transferini, uzaktan biletlemeyi ve online ticareti içerir. Bu alt yapıda merkezde bulunan Uygulama Programlama Arabirimi (API – Application Programming Interface), sahip olunan hesapları, mobil cüzdan uygulamasını ve çevresiyle etkileşimi sağlayan bileşenleri içinde barındırır. Bu arabirim sayesinde, yeni uygulamalar çalıştırılabilir, geliştirici ağını (developer network) kullanarak ticari ürünler/sistemler oluşturulabilir. Yüklemler ve ödemeler ise, güvenli bağlantılar üzerinden gerçekleştirilebilir.

Aşağıdaki şekilde Alcatel-Lucent Mobil Cüzdan Servisi altyapısı gösterilmektedir.



Şekil 2.1: Alcatel-Lucent Mobil Cüzdan Servisi Altyapısı.

Alcatel-Lucent Mobil Cüzdan Servisi (MWS); mobil cüzdan ile yapılan ödeme uygulamalarıyla kullanıcıya kolay ve uzun vadede ödeme imkanı sağlar. Bir mobil ağ operatörü kullanıcısı, bir bilet satın aldığı anda veya bir kupon kazandığında, operatör işlem ücreti alacaktır. Mobil cüzdan

fonksiyonlarıyla tüketici operatöre ekstra bağlantılarla daha az ücret ödeyebilecektir. Mobil cüzdan arayüzü ve ödeme uygulamaları, az özellikli mobil telefonlardan çok özellikli akıllı telefonlara kadar tüm telefonlarda kullanılabilir. Bir hizmet modeli olarak beyaz etiketli yazılım, bir operatör için kendi markalı cüzdanıyla başlamak için en düşük seviyede yatırım yapmasını ve markete hızlı erişmesini sağlar.

Genel olarak, bu tezde, mobil cüzdan servisinden hizmet alınmasını sağlayan SIM kart üzerindeki güvenlik biriminden veya telefon üzerindeki güvenlik biriminden çalışmayı destekleyen Java ME uygulaması geliştirilmiştir. Bu sistem, mobil cüzdan uygulamasının son kullanıcı tarafından daha kolay, güvenli ve hatasız kullanımını sağlamaktadır. Alcatel MWS (Mobil Cüzdan Servisi), GSM servis sağlayıcıları ve kurumlar için ön ödemeli Saklanan Değer Hesabı (SVA- Stored Value Account) modelini kullanan temassız ödeme sistemi, müşteri sadakat programları, para transferi, hesap durumu ve yapılan işlem geçmişinin görüntülenmesi gibi kullanıcı ve satıcı için ara yüzler ve yönetim uygulamaları sunan yüksek teknoloji uygulaması gerçekleştirmektedir. Bu uygulama ile kullanıcı hesaplarına, kredi kartları, banka kartları ve banka hesapları kullanılarak yükleme yapılabilir, yükleme işlemi otomatikleştirilebilir. Bu şekilde GSM servis sağlayıcıları ve kurumlar, müşterilerine gerçek paranın kullanıldığı bir mobil cüzdan servisi sunabilme imkânına sahip olurlar.

3. MALZEME VE YÖNTEM

3.1. MOBİL UYGULAMA MİMARİLERİ

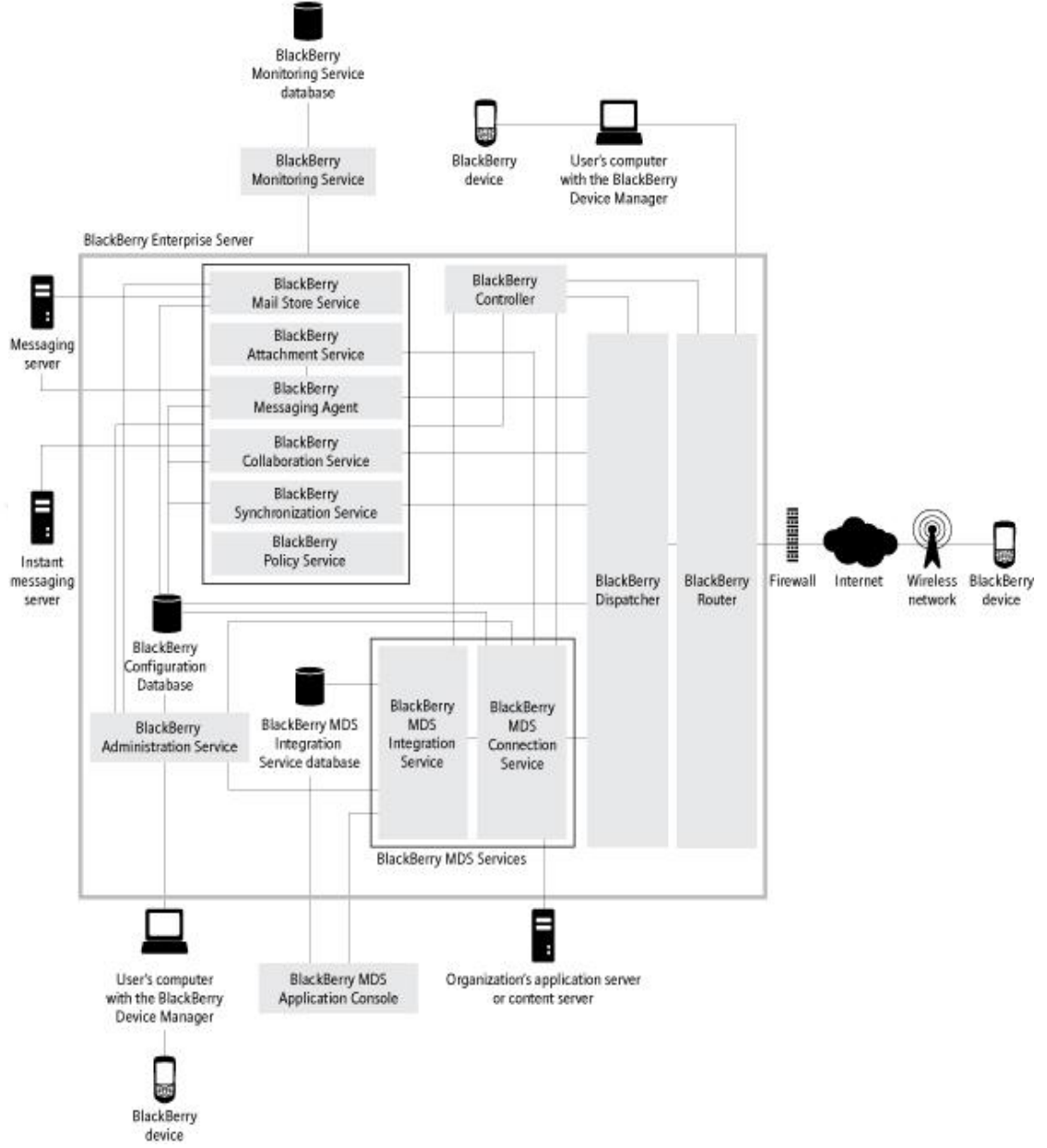
Günümüzde birçok mobil telefon üretici ve bu üreticilere bağlı olarak mobil yazılım geliştirme mimarileri mevcuttur. NFC yeni bir teknoloji olması dolayısıyla bu teknolojiye en büyük yatırımı yapan ve yazılım geliştiricilere API (Application Programming Interface – Uygulama Programlama Arayüzü) sunarak NFC çipini kullanılabilir bir hale getiren iki büyük platform; Blackberry ve Android mimarileri detaylı olarak bu çalışmada incelenmiştir.

3.1.1. Blackberry Mimarisi

BlackBerry ® platformu RIM (Research In Motion) firması tarafından Java yazılım geliştirme platformu kullanılarak geliştirilen, kendi ismini aldığı BlackBerry ® OS (Operation Sytem) işletim sistemi kullanmaktadır. BlackBerry ® platformu için yazılım geliştirmek isteyen geliştiriciler, J2ME ve BlackBerry ® hazırlamış olduğu yazılım geliştirme kütüphanelerini kullanarak akıllı telefon uygulamaları geliştirebilmektedirler. BlackBerry ® platformunun en güncel işletim sistemi BlackBerry ® OS 7.1 olup yapmış olduğumuz uygulamalar bu platform üzerinde çalışmaktadır.

BlackBerry ® Enterprise Server aşağıdaki eylemleri gerçekleştirmek üzere tasarlanmış çeşitli bileşenlerden oluşur:

- BlackBerry cihazı kullanıcıları için organizasyon uygulamaları verimlilik araçları ve veri sağlamak,
- Diğer BlackBerry Enterprise Server bileşenleri izlemek,
- Süreç, yönlendirme, sıkıştırma ve veri şifreleme,
- Kablosuz ağ ile iletişim [8].



Şekil 3.1: Blackberry Platformu Mimarisi.

BlackBerry mimarisini oluşturan bileşenleri ayrıntılı inceleyelim.

BlackBerry Yönetim Servisi, BlackBerry Yapılandırma Veri tabanına bağlanır. BlackBerry Enterprise Server bileşenleri, kullanıcı hesapları ve BlackBerry cihaz yönetimi için bu özellikleri içeren BlackBerry Domain, yönetimi için BlackBerry Yönetim Hizmeti kullanabilirsiniz.

BlackBerry Mail Deposu Hizmeti, kuruluşun ortamda mesajlaşma sunucularına bağlanır ve BlackBerry Yönetim Hizmeti mesajlaşma sunucuları üzerinde kullanıcı hesapları aramak için ihtiyaç duyduğu iletişim bilgilerini alır.

BlackBerry Enterprise Server yüklediğinizde, aynı zamanda bir BlackBerry Mail Deposu Hizmeti yüklemiş olursunuz. BlackBerry Mail Store Service, BlackBerry Enterprise Server kullanan aynı bağlantı bilgilerini kullanarak mesajlaşma sunucusuna bağlanır. BlackBerry Yönetim Hizmeti RPC kullanarak BlackBerry Mail Store Servisi ile iletişim kurmak için tasarlanmıştır.

BlackBerry Eklenti Servisi, desteklenen mesaj eklentilerini kullanıcıların BlackBerry cihazlarında görüntüleyebileceği formata dönüştürür.

BlackBerry İş birliği Servisi, kuruluşun anlık mesajlaşma sunucusu ve BlackBerry cihazları işbirliği istemcisi arasında bir bağlantı sağlar.

BlackBerry Yapılandırma Veri tabanı, BlackBerry Enterprise Server bileşenlerinin kullandığı yapılandırma bilgilerini içeren bir ilişkisel veri tabanıdır.

Örneğin, BlackBerry Yapılandırma Veri tabanı aşağıdaki bilgileri içerir:

- BlackBerry Enterprise Servera kablosuz ağ bağlantısı ile ilgili ayrıntılar,
- Kullanıcı listesi,
- BlackBerry MDS bağlantı hizmeti push özellikleri için PIN ve e-posta adresleri arasında adres eşleşmeleri.

BlackBerry Kontrolörü, BlackBerry Enterprise Server bileşenlerini görüntüler ve yanıt alamadığında onları tekrar başlatır.

BlackBerry İletişim Hizmeti, BlackBerry cihazlarından gönderdiğiniz ve aldığınız tüm verileri sıkıştırır ve şifreler. BlackBerry İletişim Hizmeti kablosuz ağ ve BlackBerry Yönlendiricileri üzerinden veri gönderir.

BlackBerry MDS Uygulama Konsolu, BlackBerry MDS Süreç uygulamalarını ve BlackBerry® Tarayıcı uygulamalarını yönetmek için kullanılan web tabanlı yönetim konsoludur. BlackBerry cihazları üzerinde BlackBerry MDS Süreç uygulamaları ve BlackBerry Tarayıcı uygulamaları kurmak, güncellemek ve yönetmek için ayrıca bir BlackBerry MDS Entegrasyon Servisi isteği göndermek için BlackBerry MDS Uygulama Konsolu'nu kullanabilirsiniz.

BlackBerry MDS Bağlantı Hizmeti, web içeriği, internet veya intranet erişim için kullanıcılara izin verir ve ayrıca uygulama verileri ve güncellemeleri için kuruluşunuzun uygulama sunucularına veya içerik sunucularına bağlanmak için BlackBerry cihazları üzerinde uygulamalara izin verir.

BlackBerry MDS Entegrasyon Hizmeti, BlackBerry cihazları BlackBerry MDS Süreç uygulamaları ve BlackBerry Tarayıcı uygulamaları için uygulama düzeyinde entegrasyon sağlar. BlackBerry cihazları üzerinde BlackBerry MDS Süreç uygulamalarını ve BlackBerry Tarayıcı uygulamalarını yüklemek için BlackBerry MDS Entegrasyon Servisini kullanabilirsiniz.

BlackBerry MDS Uygulama Deposu, BlackBerry MDS Entegrasyon Servisi tarafından barındırılan bir hizmettir. BlackBerry MDS Süre Uygulamalarını ve BlackBerry Tarayıcı Uygulamalarını, BlackBerry MDS Uygulama Deposu saklar. Yazılımcılar BlackBerry MDS Süreç Uygulamaları oluşturmak ve yayınlamak için BlackBerry® MDS Studio veya BlackBerry® Plug-in for Microsoft® Visual Studio® geliştirici araçlarını kullanabilirler. Aynı zamanda BlackBerry Tarayıcı Uygulamaları oluşturmak içinde standart text editörlerini kullanabilirler. BlackBerry MDS Uygulama Deposunda ki bu uygulamaları yayınlamak içinde BlackBerry MDS Uygulama Konsolu'nu kullanabilirler.

BlackBerry Mesajlaşma Distribütörü, mesajlaşma hizmetleri, takvim yönetimi, adresi aramaları, mail eklentileri görüntüleme ve indirme ve şifreleme anahtarı üretimi sağlamak için mobil kuruluşların mesajlaşma sunucusuna bağlanır. BlackBerry Mesajlaşma Distribütörü mesajlaşma sunucusunda ajanda verilerine erişmek için BlackBerry Senkronizasyon Hizmeti için bir ağ geçidi gibi davranır. BlackBerry Mesajlaşma Distribütörü, BlackBerry Yapılandırma Veri tabanı ile kullanıcı posta kutularını arasındaki senkronizasyonu sağlar.

BlackBerry Görüntüleme Servisi, kuruluşunuzun BlackBerry Domainini izlemenize yardımcı olmak için tasarlanmış bir web tabanlı uygulamadır. Yöneticiler, sorunlarını gidermek ve Kuruluşlarının BlackBerry Domain sağlığını izlemek için BlackBerry Görüntüleme Servisini kullanırlar.

BlackBerry İzleme Servisi, veri tabanı bir Microsoft ® SQL Server içinde, kuruluşunuzun BlackBerry Enterprise Server ortamında ® 57 hafta boyunca veri tabanı hakkında topladığı bilgileri depolar. Standart SQL arama işlemleri kullanarak veri tabanı bilgilerine erişebilirsiniz.

BlackBerry Poliçe Servisi, kablosuz ağ üzerinden yönetim hizmetleri gerçekleştirir. BlackBerry Poliçe Servisi, IT poliçeleri ve IT yönetim komutları ve hükümleri servis kitaplarını gönderir. IT poliçeleri ve IT yönetim komutları BlackBerry cihaz güvenlik, kablosuz ağ üzerinden senkronize veri ve BlackBerry cihazları diğer yapılandırma ayarları için ayarları tanımlar. BlackBerry Poliçe Servisi ayrıca, BlackBerry cihazları özellikleri ve bileşenleri için ayarları yapılandırmak üzere servis kitapları gönderir.

BlackBerry Yönlendiricileri, BlackBerry cihazlarına ve veri göndermek için kablosuz ağa bağlanır. Aynı zamanda BlackBerry ® Aygıt Yöneticisi ana bilgisayarına bağlı olan BlackBerry cihazlarına kendi kuruluşunuzun ağı üzerinden veri gönderir.

BlackBerry Senkronizasyon Servisi, ajanda verilerini, BlackBerry cihazları ve kablosuz ağ üzerinden mesajlaşma sunucusu arasında senkronize eder.

BlackBerry Web Desktop Manager (Örün Masaüstü Yönetici), kullanıcıların, BlackBerry cihazları yönetmek için izin veren bir web tabanlı uygulamadır. Örneğin; kullanıcılar BlackBerry cihazlarını aktive edebilir, bilgileri yedekleyebilir ya da saklayabilir, mesajlaşma seçeneğini değiştirebilir, bilgileri senkronize edebilir ve uygulamalar yükleyebilir.

Kuruluşunuzun uygulama sunucusuna veya içerik sunucusuna, BlackBerry MDS Hizmetleri itme uygulamaları ve intranet içeriği sağlar.

Anlık Sohbet Sunucusu, Anlık mesajlaşma hesaplarını saklar.

Mesajlaşma Sunucusu e-mail hesaplarını saklar.

BlackBerry Araç Yönetimi barındıran kullanıcının bilgisayarına bir seri veya USB bağlantısı kullanarak, bilgisayarlara BlackBerry cihazlarını bağlamak için izin verir. BlackBerry Enterprise Server ve BlackBerry cihazları arasında veri göndermek için bağlantıyı kullanabilirsiniz. BlackBerry cihazlarında ki veri trafiği, kullanıcıların bilgisayarlarına bağlıyken kablosuz ağı atlar. BlackBerry Araç Yönetimi, BlackBerry cihazları için doğrudan veri gönderen yönlendiricilere bağlanır.

3.1.2. Android İşletim Sistemi Mimarisi

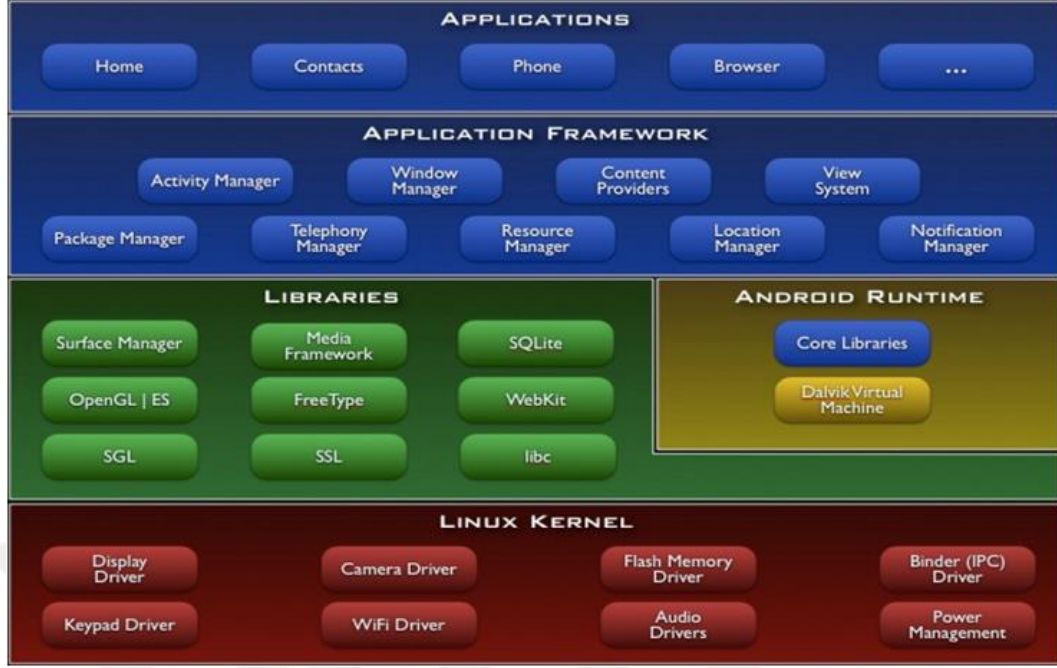
Android, Google ve Open Handset Alliance tarafından kodlanmış, Linux İşletim Sistemi tabanlı, bir mobil cihaz (PDA ve cep telefonları) için geliştirilmiş, açık kaynak kodlu bir işletim sistemidir. Android işletim sistemi, 2005 yılında Google Inc. tarafından Android Inc.'in satın alınmasıyla adını duyurmuştur. Bundan 2 yıl sonra Google'ın öncülüğünde, mobil teknoloji alanında kendini kanıtlamış şirketler, mobil teknolojilere açık standartlar getirmek için Open Handset Alliance'ı kurdular. Bunun ardından 34 farklı donanım, yazılım ve telekom şirketi, mobil cihazlar için telif hakkı olmayan bir işletim sisteminin teknolojinin gelişimi için yararlı olduğu konusunda hemfikir olmuşlardır. 2008'de piyasaya sürüldüğünde, birçok Android İşletim Sistemi, Apache Free-Software ve Açık Kaynak Kodu Lisansıyla geliştirilmeye açık hale gelmiştir [9].

Android, Linux kernel üzerine inşa edilmiş bir mobil işletim sistemidir. Mobil cihazlarla birlikte gelen yeni teknolojik donanımlara (denge sensörleri, GPS vb.) iyi adapte olan Linux Kernel, Android'in kısa sürede bu kadar çok tutulmasında önemli rol oynamaktadır. Android, aygıtların fonksiyonelliğini genişleten, uygulamalar yazan geniş bir geliştirici grubuna sahiptir. Geliştiriciler, ilk olarak aygıtı, Google'ın Java kütüphanesi aracılığıyla kontrol ederek Java dilinde yazmışlardır. Bu sistemde ara katman yazılımı, kütüphaneler ve API C diliyle yazılmıştır. Uygulama yazılımları ise, Apache harmony üzerine kurulu Java uyumlu kütüphaneler içeren uygulama iskeleti üzerinden çalışır. Android, derlenmiş Java kodunu çalıştırmak için dinamik çevirmeli Dalvik sanal makinesini kullanır [8]. Android Market ise, Android İşletim Sistemi uygulamalarının çeşitli sitelerden indirilebilmesinin yanı sıra, Google tarafından işletilen kurumsal uygulama mağazasıdır [9].

Android mimarisi, diğer işletim sistemleri mimarisine kıyasla daha gelişmiş ve daha fazla görev katmanına sahip bir mimaridir [9]. Örneğin; Win. NT mimarisinde user ve kernel olarak iki seviye vardır ve bu iki seviye net bir çizgiyle ayrılmıştır. Fakat Android mimarisinde beş katmanlı bir yapı kullanılmıştır. Bu katmanlar Dalvik adı verilen bir sanal makine üzerinde çalışarak daha üst katmanlarda daha kolay uygulama geliştirmeye olanak sağlıyor [10]. İçerisinde entegre edilmiş bir browser da barındıran android, verilerin tutulmasına olanak sunacak SQLite içeriyor. MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF gibi medya formatlarını sorunsuz destekleyen android, üzerinde çalıştığı cihaza bağlı olarak NFC, 3G, WiFi, Bluetooth, EDGE gibi özellikleri de sorunsuz şekilde destekliyor. En çok dikkat çeken özelliği ise belki de Eclipse için geliştirilmiş eklentidir. Bu sayede eclipse üzerinde zengin içerikli uygulama geliştirme çok kolaylaşmış durumdadır [9].

Şekil 3.2 Android mimarisini temel bileşenleriyle göstermektedir.

- Applications (Uygulama Katmanı),
- Application Framework (Uygulama Çatısı),
- Libraries (Kütüphaneler),
- Android Runtime,
- Linux Kernel (Kernel Seviyesi) şeklinde ayırabiliriz [11].



Şekil 3.2: Android İşletim Sistemi Mimarisi.

Uygulama katmanı, kullanıcıyla birebir ilişkide olan katmandır. Bu katmanda web tarayıcısı erişimi, e-mail istemcisi, sms programı, takvim, Google maps, telefon rehberi ve harita gibi temel ve sonradan yüklediğiniz 3. parti yazılımlar gibi kullanıcı tanımlı uygulamalar bulunur. Android uygulamalarının tamamına yakını Java programlama diliyle geliştirilir.

Android'in tercih edilmesinde uygulamalar büyük rol oynamaktadır. Bu yüzden bu tercih sebebinin, yani uygulamaların işletim sistemi tarafından kusursuz yönetilmesi gerekir. Android'de uygulamaların yönetimi Application Framework (Uygulama Çatısı) tarafında gerçekleştirilir. Bu katmanın görevleri arasında; uygulamaya ait kaynakların yönetimi, uygulamalar arası veri paylaşımının yönetimi, uygulamaların yaşam döngülerinin yönetilmesi ve donanım isteklerinin yönetilmesi gibi önemli uygulamalar vardır. Android yazılım geliştiriciler için oldukça zengin bir platform sunar. Geliştiriciler yazılımlarında donanım bilgileri, bulunduğu yer bilgisi, arka plan servisleri gibi birçok bilgiye hiçbir kısıt olmadan rahatlıkla erişebilirler. Bunun dışında uygulamalar kendi servislerini diğer uygulamalarla paylaşabilirler ve diğer uygulamaların sundukları servislere erişebilirler.

Libraries (Kütüphaneler) katmanı, Android'in çeşitli sistem bileşenlerinin kullanılabilmesi için gerekli olan, Android uygulamaları çerçevesinde Android geliştiricileri tarafından kullanılan ve C ve C++ dilleriyle yazılmış olan kütüphanelerin bulunduğu katmandır. Örneğin; veri tabanı

uygulamaları için kullanılan SQLite veri tabanının kütüphanesi, medya, grafik, 3D ve web tarayıcısı kütüphaneleri de bu katman içindedir.

Android Runtime katmanı, Java kütüphaneleri için gerekli olan temel kütüphaneler kümesini içerir.

Android uygulamalarının tamamına yakını Java diliyle yazıldığından bu uygulamaların çalıştırılmasında bir Java Sanal Makinesi'ne ihtiyaç vardır. Mobil cihazlardaki kaynakların oldukça sınırlı olması, Android için yeni bir sanal makineye ihtiyaç duyulmasına sebep olmuştur. Google bu sebeple Android için Dalvik Sanal Makinesini geliştirmiştir. Androidde her uygulama kendi sürecinin Dalvik VM'den aldığı bir oturumda koşar. Dalvik VM aynı anda çok sayıda sürece oturum hakkı verebilir. Yani birden fazla process zaman paylaşımı olarak çalışabilmektedir [10].

Sun Microsystems'ın JVM'i Java kodunu derlerken main class ve ne kadar harici class varsa bunları ayrı .class dosyaları şeklinde oluşturur. İcraa sırasında bu .class dosyaları sırasıyla koşulur. Dalvik VM'de ise çalıştırılabilir dosya .dex dosyasıdır ve her uygulama için tek bir .dex dosyası oluşturulur. Bunu yapmakta amaç sınırlı olan belleğe yüklenecek uygulama kaynak dosyalarının sadece gerekli olanlarını seçmektir. Dalvik VM'ın iş parçacıklarını ve belleği yönetme fonksiyonları Linux Kernel'a dayanır [10].

Android OS, bellek yönetimi, süreç yönetimi, güvenlik ayarları, sürücü hizmetleri, dosya sistemine erişimi, güç yönetimi yazılımı ve donanımı gibi temel işletim sistemi görevlerini Linux Kernel 2.6 ile gerçekleştirmektedir. Android, bahsi geçen özellikleri linux kernel'dan almış olsa bile bunları mobil cihazlara uyarlamak için bazı iyileştirmelere ve geliştirmelere de gitmiştir. Bu özellikler sırasıyla açıklanmıştır [12].

Android uygulamalarında çoğunlukla farklı işlerin aynı anda yapılması ve farklı sonuçların eş zamanda üretilmesi gereklidir. Örneğin; bir lokasyon uygulamasında kullanıcı uygulamanın görüntü bileşeniyle ilgilenirken, uygulamanın bir farklı bileşeni de anlık olarak sürekli GPS verisini işlemeli ve uygun değerleri kullanıcının anlayabileceği şekilde oluşturmalıdır. Aynı zamanda bu bileşenler gerekli durumlarda çalışmasını duraklatabilmeli ve devam

ettirebilmelidir. Bu gereksinimlere uygun olarak bir Android uygulaması çalıştırıldığında varsayılan tek bir linux process'i başlatılır. Burada geliştiricinin seçimine bağlı olarak aynı uygulamanın bir bileşeni için farklı bir process de oluşturulabilir. Hatta iki farklı uygulamanın iki farklı bileşeni tek bir process oluşturarak çalışmalarına devam edebilirler [13].

Android'de kaynakların sınırlı olması, gereken durumlarda bazı süreçlerin işletim sistemi tarafından öldürülmesini gerektirebilir. Örneğin, boş bellek alanının çok daralması durumunda öldürülecek süreçlerin seçiminde, kullanıcıya en az etkisi olacak şekilde bir seçim yapılır. Bunun kararı da sürecin evresine bakılarak verilir.

Android bir süreci olabildiğince uzun süre aktif tutmak istese bile kaynak yetersizliği sebebiyle zamanla biriken süreçlerden uygun olanları öldürmesi gerekecektir. Bu durumda öldürülecek olan sürecin seçilmesi kritik bir karardır. Android bu süreçlerin seçiminde "önem hiyerarşisi" yöntemini kullanır. Bu hiyerarşide her süreç önem sırasına göre kendine bir yer edinir. Sürecin önemi, o an çalışan bileşenlerinin sayısına ve durumuna bağlı olarak belirlenir. Bu hiyerarşinin en altındaki süreç ilk öldürülecek süreçtir. Önem hiyerarşisinde 5 çeşit process için önem seviyeleri bulunmaktadır. Bunlar şöyle sıralanmıştır:

Foreground Process (Öncelikli İşlem), kullanıcının o an yaptığı iş veya işlerde rolü olan süreçlerdir. Eğer süreç kullanıcıya o an görünür bir hizmet sunuyorsa, görünür değil fakat servis sunuyorsa (müzik çalar vb.) veya kullanıcıya görünür servis sunacak fakat başlama evresindeyse Foreground Process (Öncelikli İşlem) olarak nitelendirilir ve hiyerarşik yapının en üstünde yer alırlar. Fakat aynı anda çok fazla Foreground process çalıştırılmasına da izin verilmez.

Visible Process (Görünür İşlem), daha önce bir parçası koşmuş ve duraklatılmış, fakat istenildiği anda hizmet vermeye hazır olan süreçlerdir. Örneğin; mesaj yazarken kısa süreliğine başka bir uygulamaya göz atıldığında mesaj uygulaması duraklatılır fakat önemini kaybetmez.

Background Process (Önceliksiz İşlem), kullanıcı tarafından varlığı fark edilmeyen, aktivitesi durdurulmuş süreçlerdir. Bu süreçlerin silinmesi durumunda dahi kullanıcı bu durumu fark etmez. Bu yüzden herhangi bir zamanda öldürülebilirler. Bu tür süreçlerin halen bellekte

tutulmalarının sebebi Least Recently Used, yani son zamanlarda kullanılmış ve tekrar kullanılması mümkün olmasıdır. Bu sayede tekrar çağırıldığında tüm .dex dosyasını yüklemek yerine sürecin istenen aktivitesi çağırılarak süreç aktif hale getirilir.

Empty Process (Boş İşlem), tüm işini tamamlamış ve hiçbir aktif veya deaktif bileşeni olmayan süreçlerdir. Bellekte tutulmalarının amacı cache mantığına dayalıdır.

Bir android uygulaması çalıştırıldığında oluşturulan ilk thread (bileşen) ‘main thread (ana bileşen)’ olarak adlandırılır. Bu thread’in amacı uygulama arayüzünü oluşturacak bileşenleri uygulamaya dâhil etmektir. Bu bileşenler Android’in widgets (küçük araçlar) kütüphanesinden edinilir. Temel görevi bu olduğu için User Interface Thread (Kullanıcı Arayüzü Bileşeni) olarak da anılır [13].

Android işletim sistemi bir bileşen için farklı bileşenler (thread’ler) oluşturmaz. Varsayılan durumda her bileşen kendi uygulamasının süreci içindeki main thread (ana bileşen) tarafından idame ettirilir. Bu tek bileşenli (thread’li) yapı bazen ciddi sorunlara sebep olabilmektedir. Örneğin, ekranda oluşturulacak bir çizim bir uzak kaynaktan alınacaksa (örn. İnternet) ve bu alma işlemi gereğinden uzun sürerse bu thread bloklanır. Bloklanan threadin 5 saniye içinde bloğu kaldırılmazsa uygulama Not Responding (Cevap Vermiyor) gerekçesiyle sistem tarafından sonlandırılır. Bu tek thread’li yapının neden olduğu bu tür sorunları gidermek için ‘worker thread’ gibi çözümler sunulmuştur. Worker thread (İşçi Bileşen) uygulamanın geliştirilmesi aşamasında geliştirici tarafından bazı işleri yapması için görevlendirilmiş threadlerdir. Fakat burada oluşturulan worker threadlerin yanlış bir sonuç üretmesi veya sonuç üretememesi main thread’in yine bloklanmasına sebep olabileceğinden tam çözüm olmamıştır. Çözüm için geliştirme aşamasına ait farklı yaklaşım ve yöntemler mevcuttur. Bunlardan en etkili AsyncTask denilen çözüm yoludur.

Android süreçler arası iletişim (IPC) tekniği olarak Remote Procedure Calls (Uzak Yordam Çağrılar) tekniğini seçmiştir. Bu teknikte aynı kaynağı kullanan işlemler arasında Sunucu-İstemci ilişkisi vardır. Kaynağı elinde tutan işlem sunucusu, ele geçirmek isteyen istemci durumundadır ve ele geçirmek için prosedür çağrılarını yapmaktadır. Bu teknikte en bilinen uygulama Mesaj Geçme (Message Passing)’dir. Bu karşılıklı dışlama yönteminde ortak

kullanılacak kaynak işini bitirip buna dair mesajı bekleyen sürece (istemci) gönderene dek kaynak sadece sunucunun erişimine açıktır. Böylelikle Mesaj Geçme yöntemiyle Mutual Exclusion (Karşılıklı Dışlama) sağlanmış olur.

Android bellek yönetimiyle doğrudan ilgilenmez. Bu iş çekirdeği olan linux kernel 2.6 tarafından gerçekleştirilir. Ayrıca Dalvik Virtual Machine (Sanal Makina) süreçlerin koşulmasından sorumlu olduğu için dolaylı olarak bellek yönetiminde rol alır [8]. Android belleği yazılımsal olarak yönetmeyi amaçlar. Bunu da daha önce bahsedildiği üzere belleği verimli kullanarak, yani süreçleri önem hiyerarşisine göre değerlendirerek yapar.

PDA, tablet gibi akıllı cihazlar kişisel kullanım için çok uygun olduklarından birçok kişisel bilgiyi içinde barındırırlar. Android işletim sisteminde bu sebeple güvenlik önemli bir noktadır. Sistem güvenliği Linux çekirdeğiyle gelen güvenlik uygulamalarıyla sağlanabilmektedir. Başta donanım ve dosya güvenliği olmak üzere cihaza yönelik güvenlik unsurları linux kernel 2.6 ile gerçekleştirilir.

Kişisel bilgilerin güvenliğine yönelik uygulama ise izinlere dayalıdır. Android uygulamaları kapalı kutu (sandbox) modelinde çalışırlar ve kendilerine verilen izinlerin dışına çıkamazlar. Telefon, SMS, İnternet gibi farklı izinler mevcuttur. Bir uygulamaya gerekli olan izinler geliştirilme aşamasında belirlenir ve uygulamaya izin olarak eklenir. Kullanıcılar yeni bir uygulama yükleyecekleri zaman bu uygulamanın talep ettiği izinler ekranda listelenir. Böylelikle kullanıcı uygulamanın amacıyla ters düşen bir izin istemesi durumunda kendi güvenliği için uygulamaya izin vermeyebilir. Bu şekilde kötü amaçlı yazılımların da önüne geçmek mümkündür. Örneğin; işi, çekilen fotoğraflara rötuş yapmak olan bir uygulama kamera donanım izninin yanı sıra internet izni de isterse uygulamanın iyi niyetinden şüphe edilip güvenlik için karar verebilme şansı kullanıcıdadır.

Sonuç olarak, Android yeni çağın cihazları için yeni çağın işletim sistemi olmaya adaydır. 5 yıllık bir süre zarfında bu kadar kullanım oranına ulaşması da bu tezi destekler niteliktedir.

3.2. NFC (YAKIN SAHA İLETİŞİMİ) SİSTEMİ

Çalışmanın bu kısmında ise, NFC sistemi, NFC uygulamaları, NFC destekli telefon mimarisi, güvenli birim yapısı, güvenli birimi cihaz ve SIM kart (UICC - Universal Integrated Circuit Card - Küresel Bütünleşik Devre Kartı) üzerinde destekleyen uygulama mimarileri ve Java Kart teknolojisi incelenmiş, uygulamanın mimarisinin ve bileşenlerinin belirlenmesi için çaba sarfedilmiş ve genel olarak devam eden kısımda detaylı bilgi verilmiştir.

NFC teknolojisi güvenlik, kullanım kolaylığı, rahatlığı ve esnekliği ile sosyal yaşamı kolaylaştıracak bir çözüm üretmektedir. NFC teknolojisinin hizmet alanlarının genişlemesini sağlayan diğer önemli bir husus ise NFC teknolojisinin cep telefonlarıyla bütünleşik geliştirilmesidir [14, 15]. Mobil ödeme, akıllı poster ve sosyal ağ uygulamaları bunlara örnek olarak gösterilebilir. Mevcut teknolojilerin sağladığı hizmetlerin devamı ve yaygınlaştırılması için birlikte uyumlu çalışabilen bir sisteme ihtiyaç duyulmaktadır. Bu sebepten ötürü NFC sisteminde bir çok partner bulunmaktadır. Bu partnerler tüketiciler, NFC destekli cihaz üreten firmalar, güvenlik açısından sertifika sağlayıcıları (Certificate Authorities, CA), güvenilir üçüncü kuruluş (Trusted Third Parties, TTP) ve mobil ağ operatörleridir (Mobile Network Operators, MNO) [16, 14, 17]. Bu sistemin karışıklığı teknik ve yönetsel zorluklara göre değişiklik göstermektedir. NFC uygulamasının teknik zorluğu ve güvenlik yapısına göre sistemde yer alan partnerlerin üstleneceği sorumluluk farklılık göstermektedir. Hatta yeni partnerler dahi eklenebilmektedir. Bu anlamda sistemdeki tüm aktörlerin asıl amacı çıkarlarını göz ederek pazarda söz sahibi olmak ve karlarını maksimize etmektir. Kullanıcıların NFC teknolojisini daha çok kullanmaya özendirmek amacı ile yenilikçi uygulamaların ve servis alanlarının artmasıyla birlikte, hizmet sağlayıcılar arasındaki rekabetin artması da muhtemeldir [15].

NFC sisteminde güvenlikte çok önemli bir yer almaktadır. Sistem güvenliği genel olarak sertifika sağlayıcıları (CA) ve güvenilir üçüncü kuruluşlardır (TTP). Güvenlik sistemde bulunan tüm aktörler için önemlidir. Güvenliği sağlamak adına sertifika sağlayıcılarının rolü sistemdeki aktörlerin birbirleri ile ilişkilerinde kullanmak adına ıslak imza tahsis etmektir. Böylece oluşabilecek herhangi bir tanımama ya da inkâr etme durumu ortadan kalkmış olacaktır. Güvenilir üçüncü kuruluşlar ise NFC teknolojisinden faydalanan uygulamalarının yaşam döngülerini güvenilir olarak yönetmeyi hedefler. Bunların yanında NFC teknolojisinde

uzaktan yükleme sisteminde mümkündür. Buna OTA adı verilmektedir. OTA sayesinde yakın saha iletişimini destekleyen uygulamaların indirilmesi ve güncellenmesi mümkün kılacaktır. Bu anlamdaki güvenliği de Güvenilir üçüncü kuruluşlar tarafından sağlanmaktadır [18]. OTA, NFC teknolojisinde önemli bir yer tutmaktadır.

Mobil ağ operatörlerin genel olarak OTA alt yapısına tamamen sahip olduğu görülmektedir [15, 17]. NFC pazarında mobil operatörlerin kazanacağı yararlar NFC uygulamalarının dizaynına göre farklılık sergilemektedir. Mobil operatörlerin bir diğer görevi de, UICC kartlarda güvenli elemanı (Secure Element, SE) barındırarak, hizmet sağlayıcıların uygulamalarının depolanmasını ve yönetimini sağlamaktadır [19, 15]. Güvenlik adına bir diğer önemli kavramda SD (Secure Digital, Güvenli Dijital) kartlardır. SD kartlar NFC destekli mobil aygıtlara yüklenerek NFC uygulamalarını OTA üzerinden elde edilmesi ya da güncellenmesi için güvenli bir ortam sağlar [18, 20]. Bu sistemde üçüncü kuruluşların kim olacağı önemli bir konu teşkil etmekle birlikte henüz çözülememiş bir konudur.

Yakın Alan İletişimi teknolojisi veri iletişimini üç farklı çalışma kipinde gerçekleştirmektedir:

1. Okuyucu/Yazıcı kipi
2. Kart öykünüm kipi
3. Bire-bir iletişim kipi

Okuyucu yazıcı kipi, ISO14443 standardıyla ve RF ara yüzüyle etkileşimli olarak işlem yapmaktadır. Okuyucu yazıcı kipindeki işlemler NFC özelliğini destekleyen mobil aygıtın yaklaşık 4 cm yakınlıkta manyetik alan oluşturmak sureti ile NFC uyumlu pasif etikete dokundurulması işlemi ile başlar. Bu şekilde NFC destekli mobil aygıt etiketteki veriyi okuyabilir (Şekil 3.3) ya da veri üzerinde değişiklik yapabilir (Şekil 3.4) [21, 22].



Şekil 3.3: Okuyucu Kipi.



Şekil 3.4: Yazıcı Kipi.

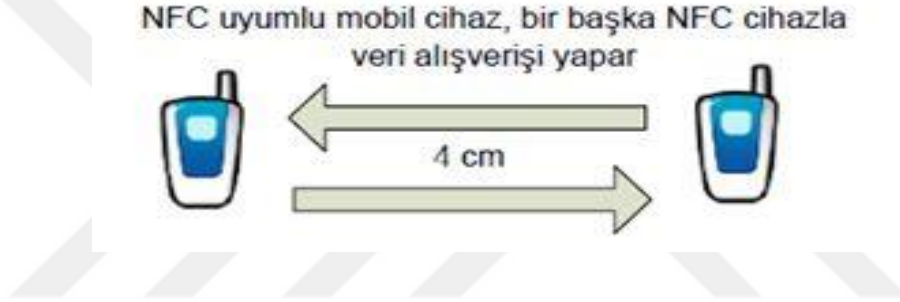
Okuyucu yazıcı kiplerine örnek vermek gerekirse akıllı poster uygulamaları buna uygun olacaktır [23].

Kart öykünüm kipinde ise NFC destekli mobil aygıt içinde birçok bilgi barındırabilir ve bu bilgi yakın alan iletişim teknolojisi ile kart okuyucuya geçirilebilir. Yakın saha iletişimli okuyucu bu çalışma esnasında yaklaşık 13.56 MHz manyetik alan oluşturur. Bu kipe ise temassız ödeme ve elektronik bilgilendirme örnek olarak verilebilir [23].



Şekil 3.5: Kart Öykünüm Kipi.

ISO/IEC 18092 ile standardize edilmiş olan birebir iletişim kipinde, okuyucu kart olmaksızın NFC destekli iki mobil aygıt birbirlerine yaklaştırılmak sureti ile iletişime geçebilirler. Örnek olarak, iki kişi arasında bilgi alışverişi kartvizit alışverişi olabilir.



Şekil 3.6: Birebir İletişim Kipi.

Yaptığımız literatür çalışması esnasında, yakın alan iletişim teknolojisi ile gerçekleştirilen ve ticari olarak kullanılan 4 tane uygulama vardır. Bunlar: akıllı ödeme uygulaması, akıllı bilet uygulaması, akıllı anahtar uygulaması ve akıllı kupon uygulamasıdır. Bu uygulamaların hepsi kart öykünüm kipinde yapılmıştır. Bu sebeple bu kip yakın saha iletişim teknolojisinin geleceğini oluşturacağını ön görebiliriz.

3.2.1. NFC Uygulamaları

NFC teknolojisi kullanılarak geliştirilen uygulamalar kullanıcılara birçok faydalar sağlamaktadır. Aynı zamanda şu ana kadar NFC literatüründe yapılan çalışmaların yaklaşık %40'ı, NFC uygulama geliştirmeleri üzerinde yoğunlaşmıştır [24]. Ekosistem ve güvenlik konuları önemli konular olsa da NFC uygulamaları bu teknolojinin omurgasını oluşturmaktadır. NFC uygulamaları temelde üç farklı gruba ayrılır. Bunlar: akıllı ortam uygulamaları, akıllı cüzdan uygulamaları ve veri değişim uygulamaları.

Akıllı ortam uygulamaları, genelde okuyucu/yazıcı kipinde çalışmaktadır ve oldukça geniş bir kullanım alanı vardır. Bu alanın en temel uygulaması ise akıllı poster uygulamasıdır. Bu uygulamada poster üzerine yerleştirilen NFC etikete cep ile dokunulduğunda etiketteki veri cep telefonuna taşınır ve daha önceden tasarlanmış olan amaçlar (telefona etkinlik kaydetme, SMS ile bilgi gönderimi, bir web hizmeti başlatma vs.) için kullanılır. Uzaktan hasta takibi [25], uzaktan alışveriş [26], yaşlı insanlar için uzaktan yemek siparişi [27] gibi uygulamalarda bu alandaki diğer örnekler olarak sıralanabilir.

Akıllı cüzdan uygulamaları, genel olarak kart öykünüm kipinde çalışmaktadır. Bu gruba akıllı ödeme, akıllı bilet ve akıllı anahtar gibi uygulamalar girmektedir. Bu uygulamanın temelinde esas alınan NFC uyumlu cep telefonun güvenli parçasında şahsa ait olan gizli bilginin (kartvizit gibi) saklanmasıdır. NFC okuyucu güvenli parçasındaki bu gizli bilgiyi okuma işleminin ardından gerekli diğer işlemleri gerçekleştirmektedir. Bu uygulamalar NFC servislerinin en önemli uygulamaları olarak görülmektedir. NFC akıllı ödeme uygulamalarında kredi ya da banka kartı bilgileri cep telefonunda saklanır ve mobil ödeme işlemi yapılacağı zaman, söz konusu olan cep telefonu NFC okuyucuya dokundurularak ödeme gerçekleştirilir [28, 29]. NFC akıllı bilet uygulamalarında önceden alınan bilet cep telefonunda saklanır ve turnikeden geçiş sırasında cep telefonu okuyucuya dokundurularak, bilet doğrulaması yapılır [30].

Veri değişimi uygulamaları, birebir iletişim kipinde icra edilmektedir. Bu uygulamalarda esas alınan husus akıllı mobil cihazları kullanarak karşılıklı bilgi alışverişinde bulunmaktır. Dosya transferi [30] ve kartvizit değişimi [31] bu alanda yapılan uygulamalara örnek olarak verilebilir.

NFC'nin üç farklı çalışma kipinin geliştirilmiş olması ve bu üç kipin veri iletişimde farklılaşması uygulamaları üç farklı gruba ayırmıştır.

NFC teknolojisinin en önemli servis alanlarından biri de mobil ödeme sistemidir [32]. Mobil ödeme sistemi sayesinde tüketiciler ceplerinde birçok banka kartı taşımak yerine, tüm kartların bilgilerini NFC destekli mobil cihazlarında saklayıp alış veriş sırasında ya da diğer işlemler esnasında cihazını ödeme noktasına (Point of Sale, POS) dokundurma yolu ile gerçekleştirebilir. Bilet sisteminde biletlerinin ödemelerini yapabilir ya da e-biletlerini görüntüleyebilirler.

Kullanıcıların mobil ödeme sistemini kullanabilmesi için en başta NFC destekli bir mobil cihaza sahip olmaları gerekir. Bu koşul sağlanırsa, kullanıcılara bankaları tarafından sağlanacak NFC mobil ödeme uygulamasının OTA sistemi üzerinden elde edilmesi gerekir. İşlemler sürecinde güvenlik ise tercihe bağlı olarak (UICC, SD kart vs.) barındırılabilir [19].

Mobil ödeme sistemi her ne kadar kolay gözükse de, arka planda çalışan karmaşık bir sistem bulunmaktadır. Bu sistem bütün aktörlere kazançlı bir model yaratmayı amaçlamalıdır [23].

3.2.2. NFC Uygulamalarının Faydaları

Önceki bölümlerde ele alındığı gibi okuyucu/yazıcı kipinde, bir NFC etiketinde bulunan veri (bir metin, telefon numarası veya herhangi bir veri) mobil cihaza iletilir ve mobil cihaz bu veriyi herhangi bir amaç için işleyebilir. En basit olarak, bu veri ekranda kullanıcıya gösterilir ve kullanıcı hareket halindeyken bu veriyi kullanmaya devam edebilir. Bu nedenle bu çalışma kipindeki uygulamalar mobil cihazların ana amacı olan kullanıcı hareketliliğini yükseltmeyi amaçlar. Bu çalışma kipindeki uygulamaların bize kazandırdığı bir diğer yarar ise kullanıcıların harcaması gereken fiziksel gücü düşürmesidir. Bu yarara örnek vermek gerekirse NFC ile uzaktan hasta kontrolü, uzaktan yemek siparişi gibi uygulamalar verilebilir. Günümüzde bulunan NFC uygulamalarının çoğunluğu okuyucu/yazıcı kipindedir. Bunun sebebi ise uygulama geliştirmenin daha kolay olması ve birçok gerçek senaryonun bu kipte kolaylıkla uygulanabilmesidir.

Kart öykünüm kipinin temel amacı ise kullanıcı hareketliliğinden çok, cep telefonlarını kullanıcıya daha bağlı hale getirmektir. Örnek vermek gerekirse, kredi kartı ve akıllı anahtar gibi veriler mobil cihazlar ile birleştiğinde, mobil cihazları artık mobil cüzdan olarak kullanılabilir. Bu kipte geliştirilen uygulamaların genel özellikleri araştırıldığında ise şahsa özel veriyi veya nesneyi mobil cihazlar ile birleştirdiğini görmekteyiz. Akıllı kuponların mobil cihazlarda korunması ise buna örnek olarak gösterebilir. Bu kipi farklı bir diğer yararı ise ulaşım kontrolü desteklemesidir. Ana amaç ise fiziksel bir nesnenin bağımlılığını ortadan kaldırmak ve ulaşım kontrolünü sağlamak şeklinde kısaca açıklanabilir. Bu çalışma kipinde günümüzde geliştirilen uygulamalara bakarak, gelecekte uygulanabilecek potansiyel servislerde, insanların cep telefonlarını artık bir dijital veri saklama yeri olarak kullanabileceğini öngörmekteyiz. İlerleyen günlerde gerçekleştirilebilecek kart öykünüm hizmetlerinden birkaçı, nüfus cüzdanlarının, pasaportların, parmak izlerinin ehliyet belgelerinin

ve diğ er kullanıcıya ait verilerin mobil cihazlar yardımıyla birleştirilmesidir. Ayrıca bazı önemli verilerin mobil cihazlarda saklanarak, o verilere ulaşması gereken insanlara izin verilerek ve gerekli engellemeler yapılarak bu verilere ulaşılmasının sağlanacağı ve kontrol edileceğini ön görüyoruz.

Birebir iletişim kipinde ise temel amaç iki cep telefonunun birbirlerine veri transferi yapabilmesidir. Geliştirilen uygulamalardan bazıları dosya transferi, kartvizit değişimidir. Ş u ana kadar bu çalışma kipinde geliştirilen uygulamalar, diğ er kiplerde geliştirilen uygulamalara göre çok azdır. Bu kipi getirdiğı en önemli yarar ise iki mobil cihaz arasında kolay veri aktarımıdır. Fakat bu kipte yeteri kadar çalışma yapılmamıştır. Bu çalışma kipinde gelecekte gelebilecek potansiyel hizmetlere baktığımızda ise güvenlik açısından gizli olması gereken bilgilerin güvenli olarak paylaşımını öngörebiliriz. Zaten NFC cihazları 4 cm içinde veri transferi yaptığı için fiziksel güvenliği sağlamak kolaydır.

3.2.3. NFC Teknolojisini Uygulama Ş ekilleri

Halen geliştirilme süreci devam eden NFC teknolojisi ile ilgili olarak üç yapı bulunmaktadır [1]. Bunlar telefonumuza harici olarak yerleştireceğimiz;

1. NFC Sim Kartlar,
2. NFC SD/mikro SD Kartlar,
3. Mifare pasif etiketler.



Ş ekil 3.7: NFC Sim Kart.

Şekil 3.7’de görüldüğü üzere NFC sim kart yapısında; RFID etiket yapısına benzer bir şekilde (çip artı anten) kısmı sim kartın üzerinde bulunmaktadır. Kartın etrafını çevrelemiş olan anten yapısına ilaveten ID bilgisini taşımakta olan çip de yine kartın üst kısmında bulunmaktadır. NFC sistemi için ayrılmış olan bu hafıza alanına erişim izni ise mobil sağlayıcı ve bankanın yanı sıra üçüncü bir firma tarafından sağlanılmaktadır. Aynı günümüzde kullanılmakta olan sim kartlara mobil sağlayıcıların kendi uygulamalarını eklemekte kullandıkları sistem olan SMS ile kod yollama tekniği ile NFC için ayrılmış olan hafıza alanına erişilebilmektedir. Örneğin, mobil cihazımızdan yazdığımız kısa mesajın başına cep telefonumuz mesajı yollarken text uzantılı bir mesaj attığına dair karşı kullanıcı cihazının anlayabileceği standart bir kod ekler. Mobil şebekeler de sim kartın içine bir uygulama yollarken kısa mesaj yolu ile yollarlar. Fakat yollanan mesajın başındaki kod farklı olacağından gelen kısa mesajı açsak dahi text olarak göremeyiz.

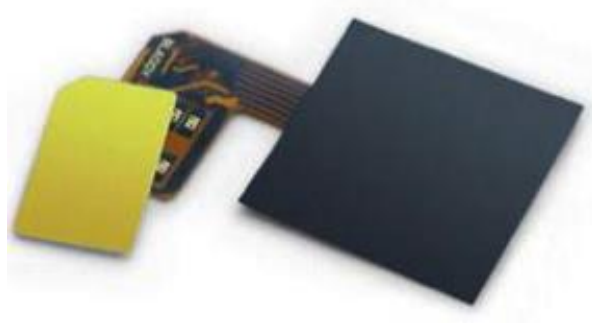
NFC için ayrılmış olan hafıza alanına erişim de aynen yukarıda anlatıldığı şekilde olmaktadır. Fakat yasal olarak mobil şebeke operatörünün bu alana erişmesine izin verilmediği için, izne sahip olan üçüncü bir firma bu özel olarak ayrılmış hafıza alanına erişebilmektedir.



Şekil 3.8: NFC Mikro SD Kart.

Şekil 3.8’de görüldüğü üzere diğer bir NFC’ye geçişte kullanılan ara form da NFC’li mikro SD kart uygulamalarıdır. Yine bu uygulamada da mevcut cep telefonumuza haricen takacağımız bir hafıza kartı ile sisteme dâhil olup NFC mobil ödeme sistemini aktif şekilde kullanabilmemiz sağlanmaktadır. NFC mikro SD kartlarda da yine sim kart uygulamalarında olduğu gibi NFC

sistemi için ayrılmış bir hafıza alanı ile birlikte haberleşmenin sağlanabilmesi için sarmal anten yapısı da bulunmaktadır.



Şekil 3.9: Pasif NFC Kart.

Son olarak bahsedeceğimiz yapı ise neredeyse tamamıyla NFC mobil ödeme sisteminde kullanılmakta olunan pasif 1 kbit’lik RFID etiket kullanılarak gerçekleştirilen yapıdır [16]. Bu yapıda mifare şifreleme tekniklerini destekleyen ya da Des-wire etiket denilen şifrelemeye uygun etiketlerden birisi kullanılmaktadır (Şekil 3.9). Sim kart ile bağlantılı şekilde yerleştirilen kart batarya konulduktan sonra rahatça bataryanın üzerine bükülerek hem fazla yer kaplamamakta hem de görüntü olarak herhangi bir kabalık oluşturmamaktadır (Şekil 3.10). Bu yapıya baktığımızda ise, mobil abonelere telefonlarına sadece bu etiketi yerleştirerek mobil ödeme sistemine dâhil olmaları imkânı sağlanmaktadır.



Şekil 3.10: Pasif NFC Kart Uygulaması.

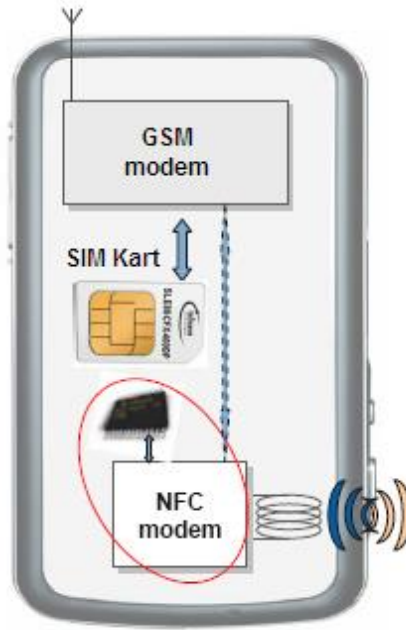
3.2.4. NFC Mobil Ödeme Uygulamaları

Literatür taramasında belirtildiği gibi, NFC destekli cihaz, NFC modem ve güvenlik birimi (SE-Secure Element) olarak iki bileşenden meydana gelmektedir. Güvenli birim (Secure Element); donanımlardan, yazılımlardan, arayüzlerden ve güvenli depolama olanağı sağlayan bir cep

telefonundaki gömülü protokollerden oluşan bir birimdir. Güvenli birim; kart öyküleme modu sırasında kullanıcı tarafından seçilerek, ödeme verileri, şifreler, ödeme uygulama kodu gibi konularda ödeme sistemlerinin korunması ve uygulamaların gerçekleşmesi için güvenli bir alan sağlar [14]. Ödeme uygulamalarında kullanılmasının yanında, kimlik doğrulama işlemlerinde ve güvenlik işlemleri gerektiren ödeme ile ilgili olmayan depolama uygulamaları için de kullanılabilir.

NFC tabanlı uygulamaların geliştirilmesi için iki yaklaşımdan bahsedilmektedir:

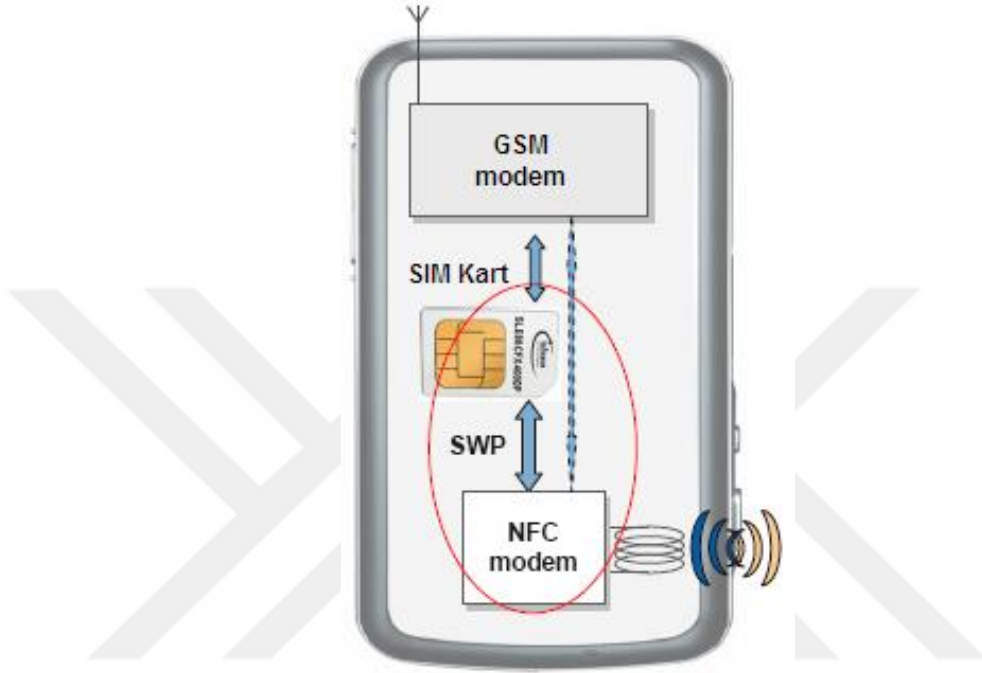
Mobil telefon üreticisi merkezli NFC uygulaması yaklaşımında, güvenlik birimi mobil cihazın içine harici bir çip olarak yerleştirmekte ve cihaz tabanlı bir hale gelmektedir. Geliştirilecek NFC uygulamaları bu güvenli birimle beraber çalışacak altyapıda olmalıdırlar. Bu yaklaşım “SE-NFC” olarak isimlendirilmektedir.



Şekil 3.11: Mobil Telefon Üreticisi Merkezli NFC Uygulaması Yaklaşımı.

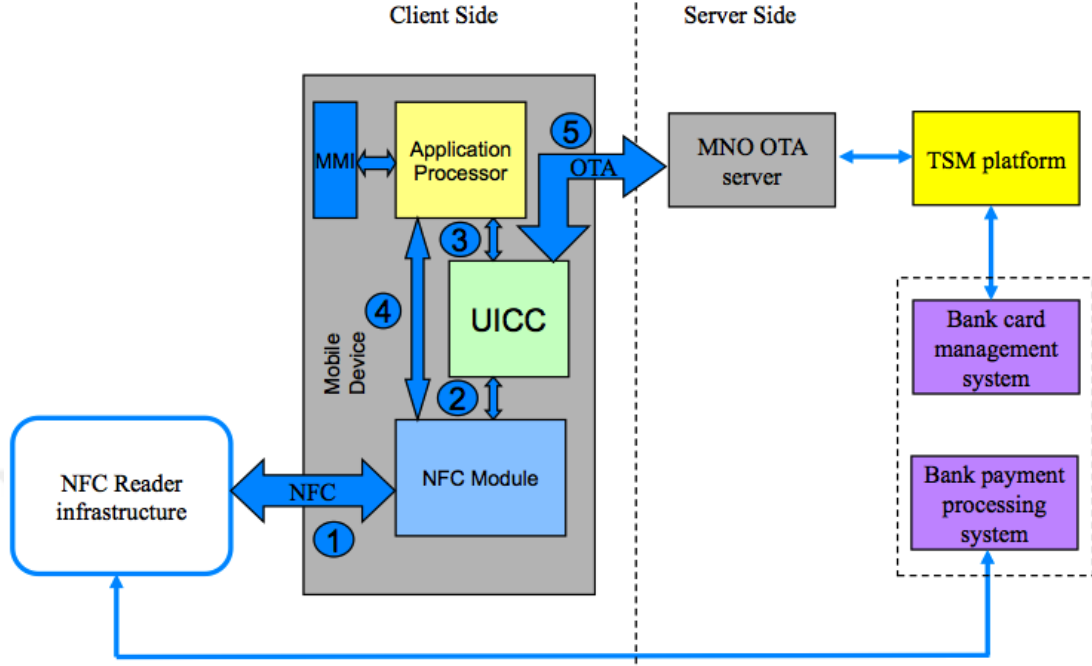
GSM servis sağlayıcısı merkezli NFC uygulaması yaklaşımında, güvenlik birimi mobil cihazın içine harici bir çip olarak yerleştirilmeyip sim kart üzerindeki bir bölümde tutulmaktadır. NFC modem ile SIM üzerindeki güvenli birim (SE) kendi aralarında Avrupa Telekomünikasyon Standartları Enstitüsü (ETSI) tarafından onaylanması sürecinde olan SWP (Single Wire Protocol – Tek Kablo Protokolü) protokolü üzerinden iletişim sağlarlar. Bu mimaride NFC

uygulamaları cihaz tabanlı bir durumdan çıkıp, SIM kartı sağlayan GSM servis sağlayıcı tabanlı bir duruma gelmektedir. Geliştirilecek NFC uygulamaları bu SIM üzerindeki güvenli birimle beraber çalışacak altyapıda olmalıdırlar. Bu yaklaşım “USIM / UICC / SE-SIM” olarak isimlendirilmektedir.



Şekil 3.12: GSM Servis Sağlayıcısı Merkezli NFC Uygulaması Yaklaşımı.

3.2.5. UICC (Küresel Bütünleşik Devre Kartı) Kart Mimarisi



Şekil 3.13: UICC Kart Mimarisi [33].

3.2.5.1. NFC Standartları

— ISO 14443 (Temassız Bütünleşik Devre Kartları)

Fiziksel Özellikleri, radyo arayüzü, başlatma ve iletişim protokollerini standartlaştırır.

— ISO/IEC 15693 (Temassız Bütünleşik Devre Kartları)

RFID birimini inceler.

— ISO/IEC 18092 or ECMA 340 (Yakın Saha İletişimi Arayüzü ve Protokolü, NFCIP-1)

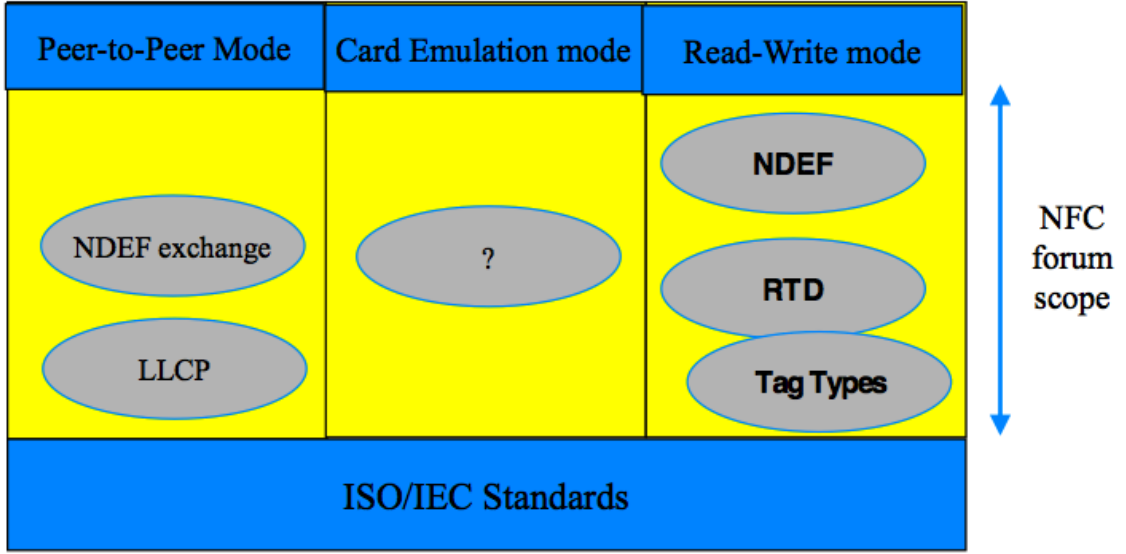
RF alan/sinyal arayüzleri, başlatma ve taşıma protokolleri standartlaştırır.

— ECMA 352 or ISO 21481 (NFCIP-2)

Çalışma modunu belirler. (ISO 14443, 15693 or NFCIP-1)

— NFC Forum

ISO/IEC standartlarını belirler.



Şekil 3.14: NFC Standartları [33].

3.2.5.2. UICC – NFC Modül Arayüzü

— ETSI TS 102 613: Tek Kablo Protokolü (SWP)

2007 Kasım’da onaylanan yeni bir standarttır. Uygulamalar oluşmaya başlar. Donanım katmanını ve yazılımı belirleyen SWP gereklidir.

— ETSI TS 102 622: UICC- Temassız Kullanıcı Arayüzü, Ana Bilgisayar Kontrol Arayüzü (HCI)

SWP’un üstünde API (Uygulama Programlama Arayüzü) standartını belirler. 2008 Eylül’de onaylanan yeni bir standarttır. Uygulamalar yeni oluşmaktadır.

3.2.5.3. Application Processor (Uygulama İşlemcisi) – UICC Interface (Arayüzü)

— 3GPP TS 11.11 SIM-ME Arayüzü / ETSI TS 102.221 UICC- Terminal Arayüzü

ISO 7816 tabanlı haberleşmedir. Seri düşük hızlıdır (9600 bps). Yaygın uygulamalardır.

— ETSI TS 102 600 Akıllı Kartlar: UICC- Terminal Arayüzü, USB Arayüz Karakterleri

12 Mbps ‘a kadar hızlanır. Uygulamalar yeni oluşmaktadır.

— 3GPP TS 11.14 SIM Uygulama Araç Takımı /ETSI TS 122.038 USIM Uygulama Araç Takımı / ETSI TS 102.223 (Kart Uygulama Araç Takımı)

Yaygın uygulamalardır; fakat terminal tarafında tüm uygulama nadir bulunmaktadır.

— **OMA Akıllı Kart Örün Sunucusu (SCWS)**

Telefonda tarayıcıdan başlatılır. Uygulama yaygın değildir. Zengin grafik MMI mümkündür.

3.2.5.4. Application Processor (Uygulama İşlemcisi) – NFC Modül Arayüzü

Yaygın kabul görmeyen standarttır. Terminal üreticisi NFC modülü telefonda bulundurduğu için servis sağlayıcıları için çok önemli değildir. NFC Forum API standartları belirlemek için NCI (NFC Kontrol Arayüzü) üzerinde çalışıyor [33].

3.2.5.5. OTA (Hava Üzerinden İletişim) Teknolojisi Standartları ve Teknolojileri

— **GlobalPlatform**

Finans sektöründen başlar. UICC üzerinde güvenlik alanı oluşturur. Güvenli Kanal Protokolleri (SCP) çalışır. Güvenli uygulama indirilmesini, yüklenmesini, silinmesini ve kilitlenmesini yönetir.

— **3GPP TS 03.48/ ETSI TS 102.225/226**

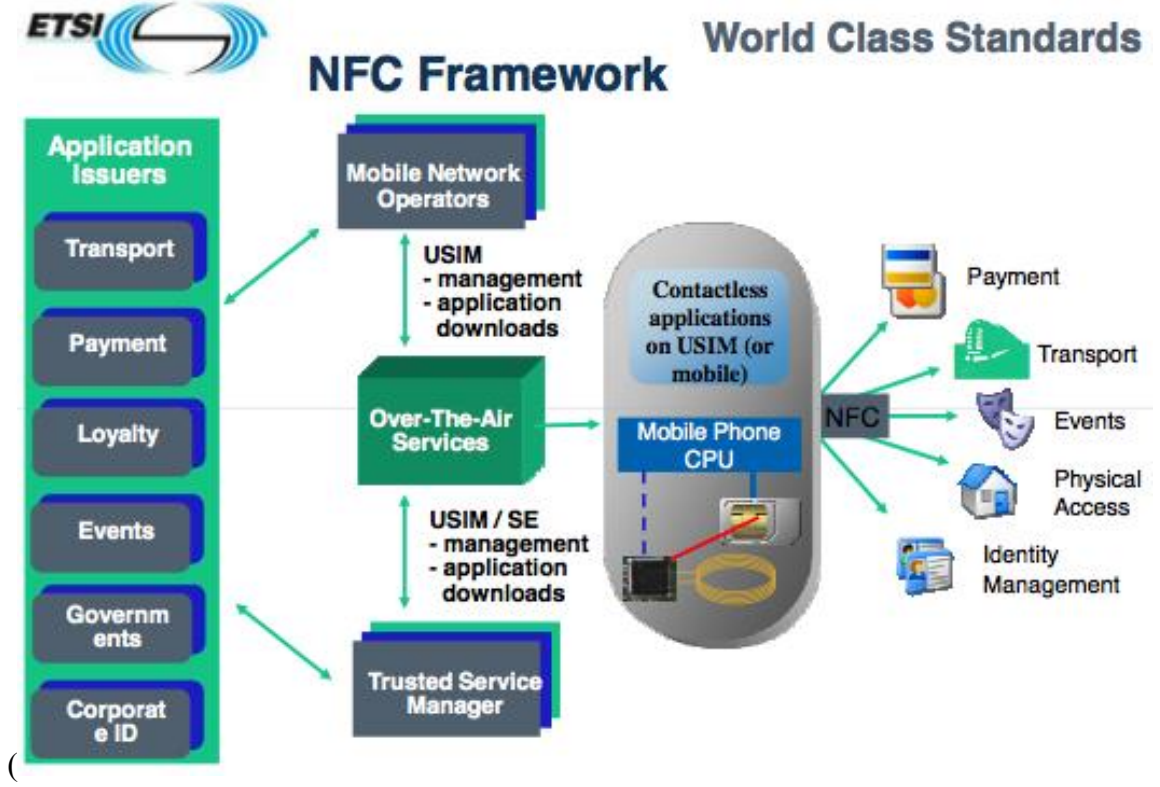
Remote server'dan UICC'ye direkt olarak veri gönderir. Global platform komutlarıyla uyumludur.

— **Midlet proxy method**

Server ve UICC arasında telefon proxy haberleşmesinde midlet kullanır.

3.2.6. NFC (Yakın Saha İletişimi) Framework (Çerçevesi)

Şekil 3.15’de NFC (Yakın Saha İletişimi) Framework’u (Çerçevesi) gösterilmektedir:



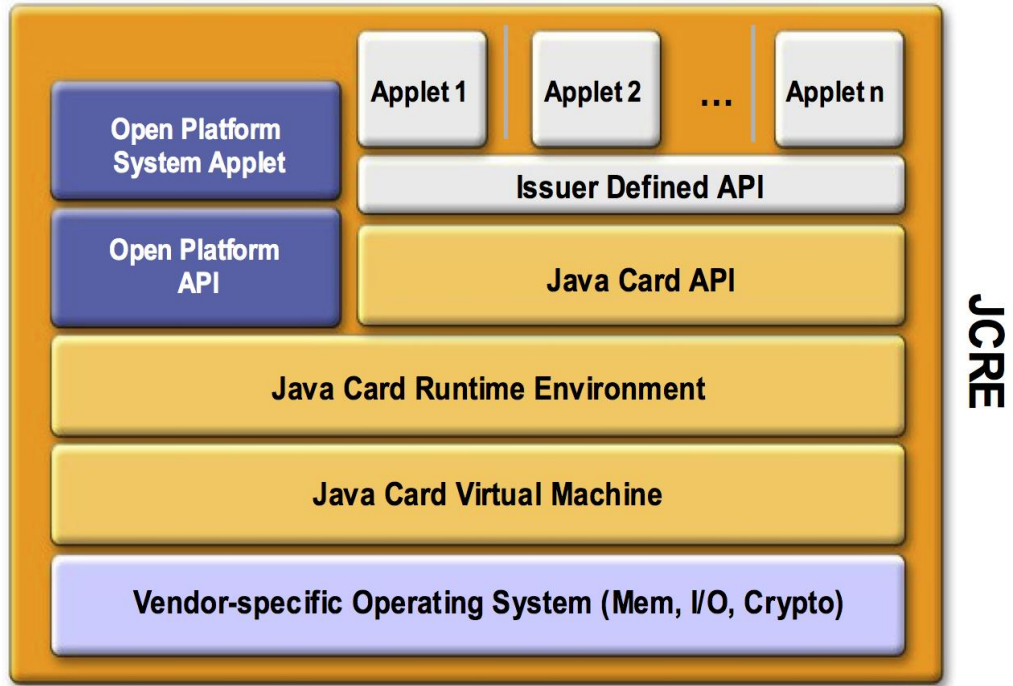
Şekil 3.15: Yakın Saha İletişimi Çerçevesi [33].

3.2.7. Java Kart Teknolojisi

Java Card teknolojisini applet olarak tanımlanan Java tabanlı uygulamaların akıllı kartlar (smart card) üzerinde çalışmasını sağlayan bir teknolojidir. SIM ve ATM kartlar üzerinde yaygın olarak kullanılan, Oracle tarafından geliştirilen bir uygulama geliştirme platformudur. Java Card platformunun sağladığı güvenlik sayesinde kullanıcı bilgileri güvenli olarak akıllı kartlar veya SIM kartlar üzerinde tutulmaktadır. Herhangi bir darbe ya da zorlamaya sonunda kart geçersiz olup, içindeki bilgiler kullanılamaz hale gelmektedir. Sağlamış olduğu standart arayüzler ve güvenlik sayesinde çok yaygın olarak akıllı kart üreticileri tarafından kullanılmaktadır [34].

Java Card Platform 2 ve Java Card Platform 3 olmak üzere iki ana platforma ayrılmaktadır. Java Card 3 Platformu daha yeni bir platform olup, çok hafif bir sunucu barındırmaktadır. Böylelikle yapılan her türlü işlem, sunucuya gelen istemlerle olmaktadır. Bunun yanı sıra Java

Card 2 daha eski bir platform olup çok yaygın olarak kullanılmaktadır. Akıllı kart ile yapılan iletişim APDU komutları üzerinden olup, client ve kart üzerinde komut değişimi yapılmaktadır. Java Card'lar Java teknolojisinin bir ürünü olup, ilk olarak Sun Microsystems tarafından geliştirilmiştir. Günümüzde bu teknoloji Oracle tarafından geliştirilmektedir. Java Card'lar kısaca, Java uygulamasının bir alt seti olup çok sınırlı bir sanal makine üzerinde çalışmaktadır. Bir ve daha fazla uygulama güvenli olarak bir kart üzerinde çalışabilir. Kripto kütüphaneleri sayesinde güvenli bir teknolojidir [35].



Şekil 3.16: Java Kart Mimarisi [35].

Java'nın bir alt kümesi olan Java Card dili aşağıdaki özelliklere sahiptir:

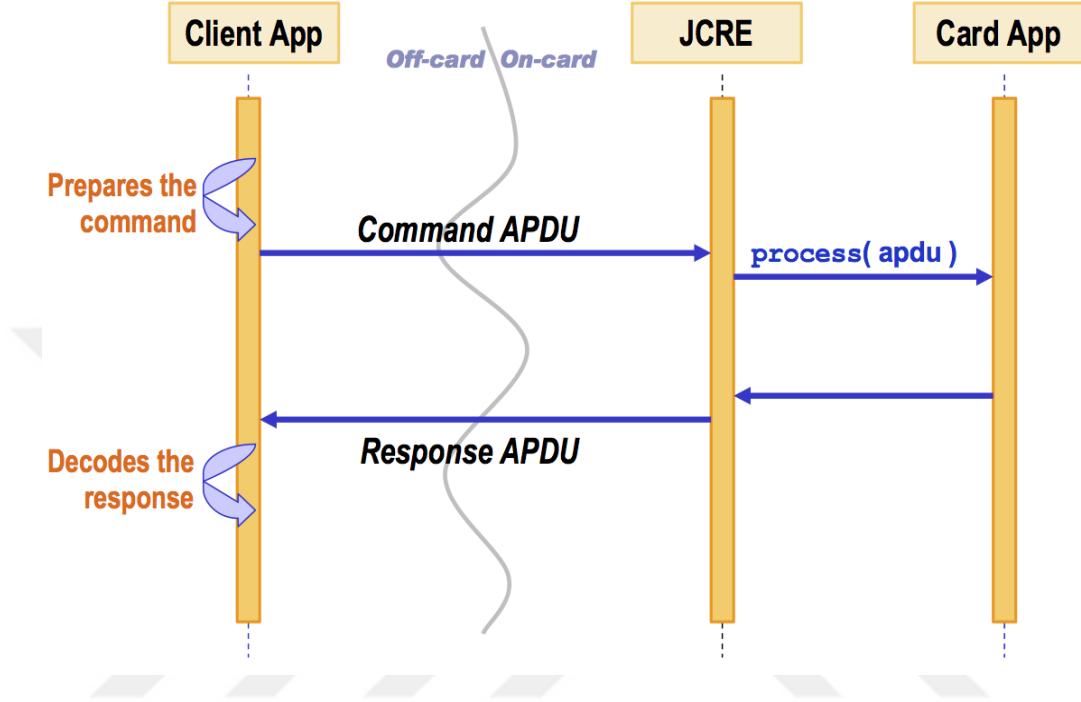
- Küçük veri tipleri, boolean, byte, short,
- Tek boyutlu diziler,
- Paketler, sınıflar, arayüzler ve hatalar(exceptions),
- Miras alma, sanal metodlar, dinamik nesne oluşturma.

Java Card sanal makinesinin ana özellikleri;

- Kart(Card) kaynak yönetimi,
- İletişim metodları (APDU),
- Applet yürütme (execution),

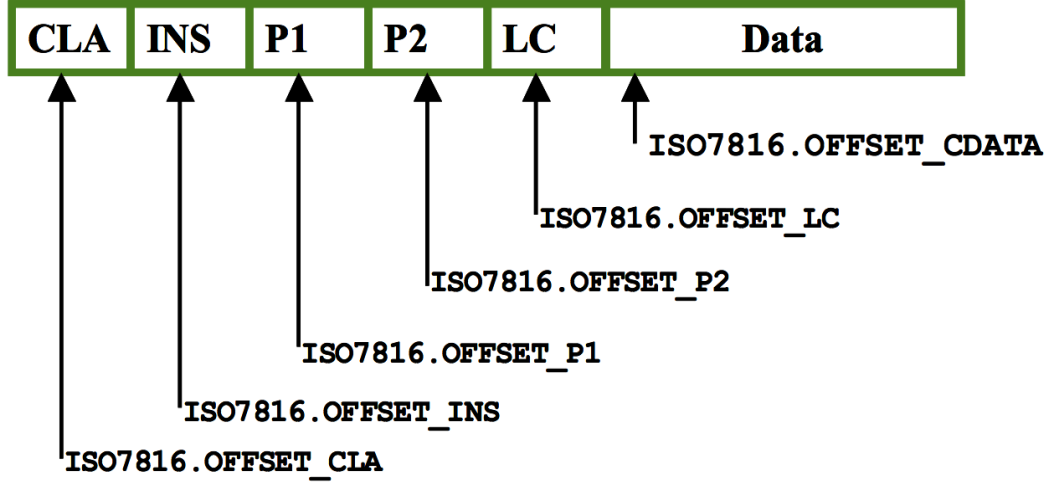
- Applet güvenliği (güvenlik duvarı).

Java Card uygulaması veri akış diyagramı aşağıdaki Şekil 3.17’de gösterilmiştir.



Şekil 3.17: Java Kart APDU İletişimi [35].

Şekil 3.17’de gösterilen iletişim şu şekilde oluşmaktadır. İstemci uygulama APDU komutu hazırlar ve bu komuta JCVM (Java Card Sanal Makine) vasıtası ile Java Card uygulamasına (Applet)’e gönderir. Applet komutu işler ve sonucu JCVM aracı ile istemci uygulamaya gönderir. Bir APDU komutunun yapısı Şekil 3.18’de gösterilmiştir.



Şekil 3.18: APDU Yapısı [35].

Bir APDU komutunun yapısı aşağıdaki şekildedir:

- CLA : Komutun sınıfını (türünü)
- INS : Komutun ne olduğunu
- P1 / P2 : Parametreler
- LC : Veri boyutunu
- Data : Veri

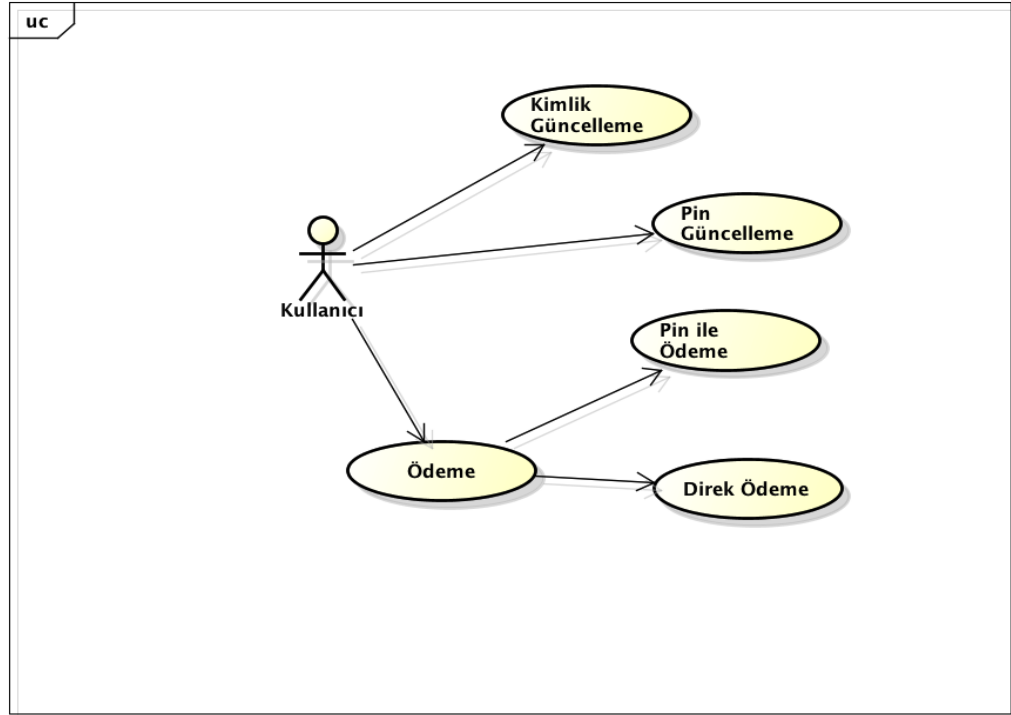
SIM ve telefon üzerinde bulunan güvenli arabirimler Java Card uygulamasını çalıştırma yeteneğine sahip bir Java Card Sanal Makinesi ihtiva ederler. Tez kapsamında tasarlanıp geliştirilecek olan güvenli arabirim uygulaması Java Card teknolojsini temel almaktadır.

4. BULGULAR

4.1. MWS- GÜVENLİ ARABİRİM UYGULAMASI TASARIMI

Güvenli arabirim uygulaması tasarlanırken gereken bileşenler, kullanıcı senaryoları ve akış diyagramları belirlenerek uygulamanın mimari tasarımı ortaya çıkartılmıştır. Bu tasarım sonrasında Java Card teknolojisi kullanılarak geliştirilen güvenli arabirim uygulaması ile Alcatel MWS (Mobile Wallet Service – Mobil Cüzdan Servisi) tarafından sağlanan kullanıcı kimlik bilgileri güvenli olarak tutulmaktadır. Ayrıca NFC destekli sanal pos uygulamalarında kullanıcı bilgileri terminal ile güvenli bir biçimde iletilerek, terminal üzerinden ödeme işlemleri yapılmaktadır.

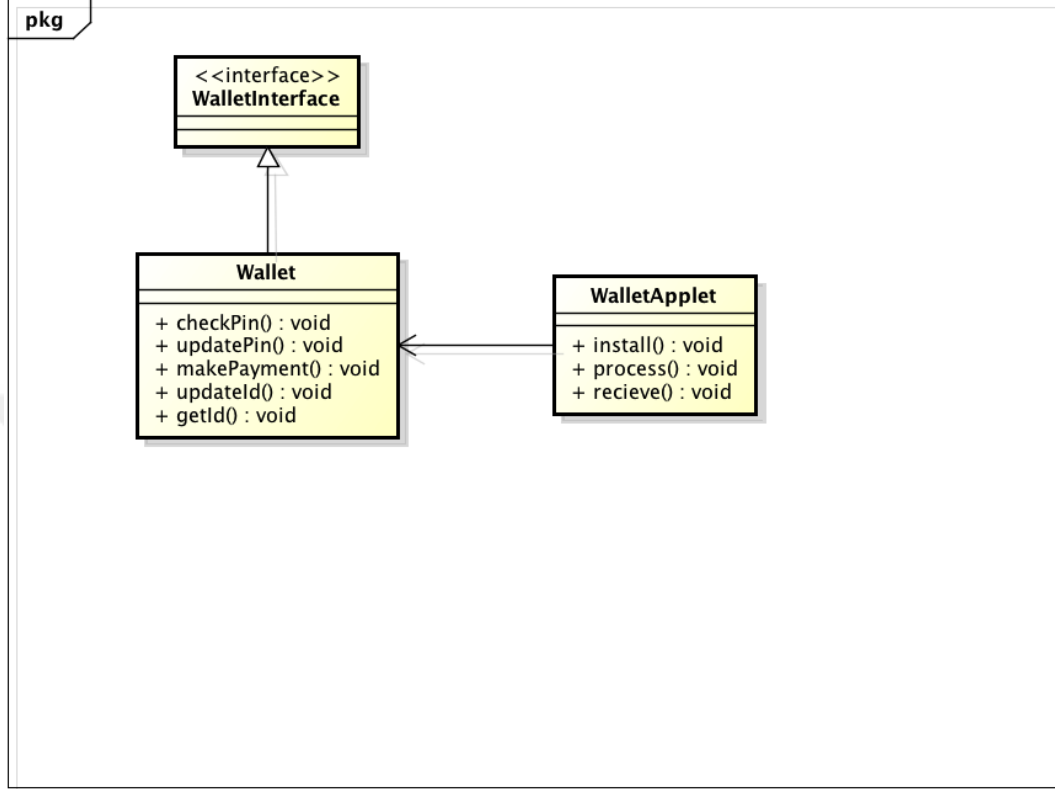
Şekil 4.1’de mobil cüzdan servislerini kullanan bir kullanıcının güvenli arabirim ile yapabileceği işlemler gösterilmiştir.



Şekil 4.1: Güvenli Arabirim İşlemleri.

Alcatel-Lucent MWS hizmeti tarafından hizmet alan bir kullanıcının yapacağı işlemler; güvenli arabirime kimlik bilgilerini ekleyebilir ya da silebilir, ödeme yapabilir veya pin değiştirebilir.

Bu işlemler doğrultusunda gerekli olan uygulama sınıfları ve arabirimler yapısı Şekil 4.2 üzerinde gösterilmiştir.

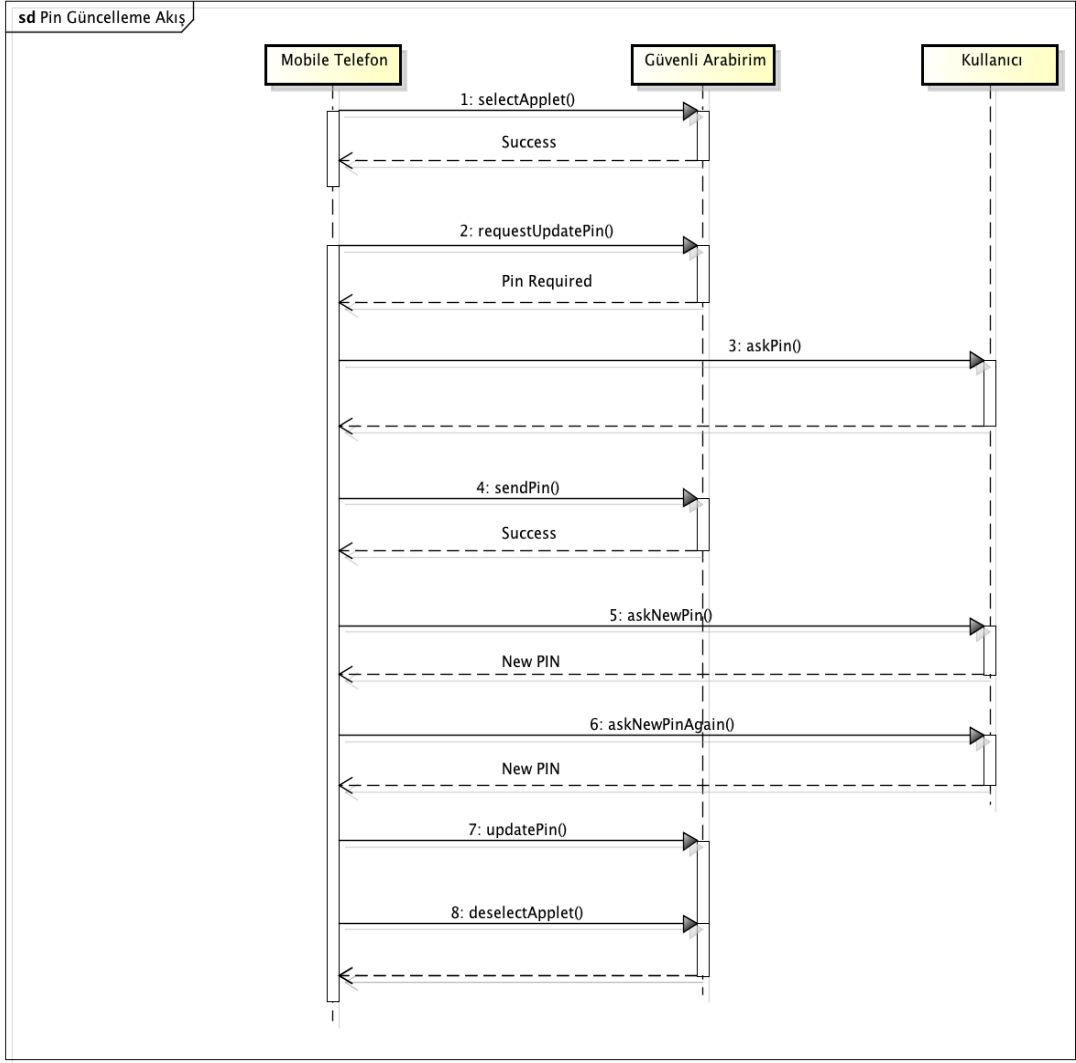


Şekil 4.2: Güvenli Arabirim Sınıf Yapısı.

Şekil 4.2 üzerinde de görüldüğü üzere, Java Card üzerinde çalışacak olan applet, kullanıcı cüzdanından referans almaktadır. Kullanıcı cüzdanı ise cüzdan arabiriminden faydalanmaktadır.

Applet ilk çalıştığı anda, Java Card mimarisinin yapısından dolayı, uygulama karta yüklenir ve başlangıç parametereleri ayarlanır. Bu işlemler bitiminde Applet hazır bir duruma gelmiştir. Gelen her komut Java Card Sanal Sunucu tarafından mobil cüzdan uygulamasına yönlendirilir. *process* metodu ise gelen APDU parametrelerine göre yapacağı işlemleri belirler (pin kontrolü, pin güncelleme, ödeme yapma, kimlik okuma ve kimlik güncelleme).

Tasarımda belirtilen işlemleri tek tek inceleyelim. Pin bilgisini güncelleme adımlarının akış şeması Şekil 4.3’de gösterilmiştir.

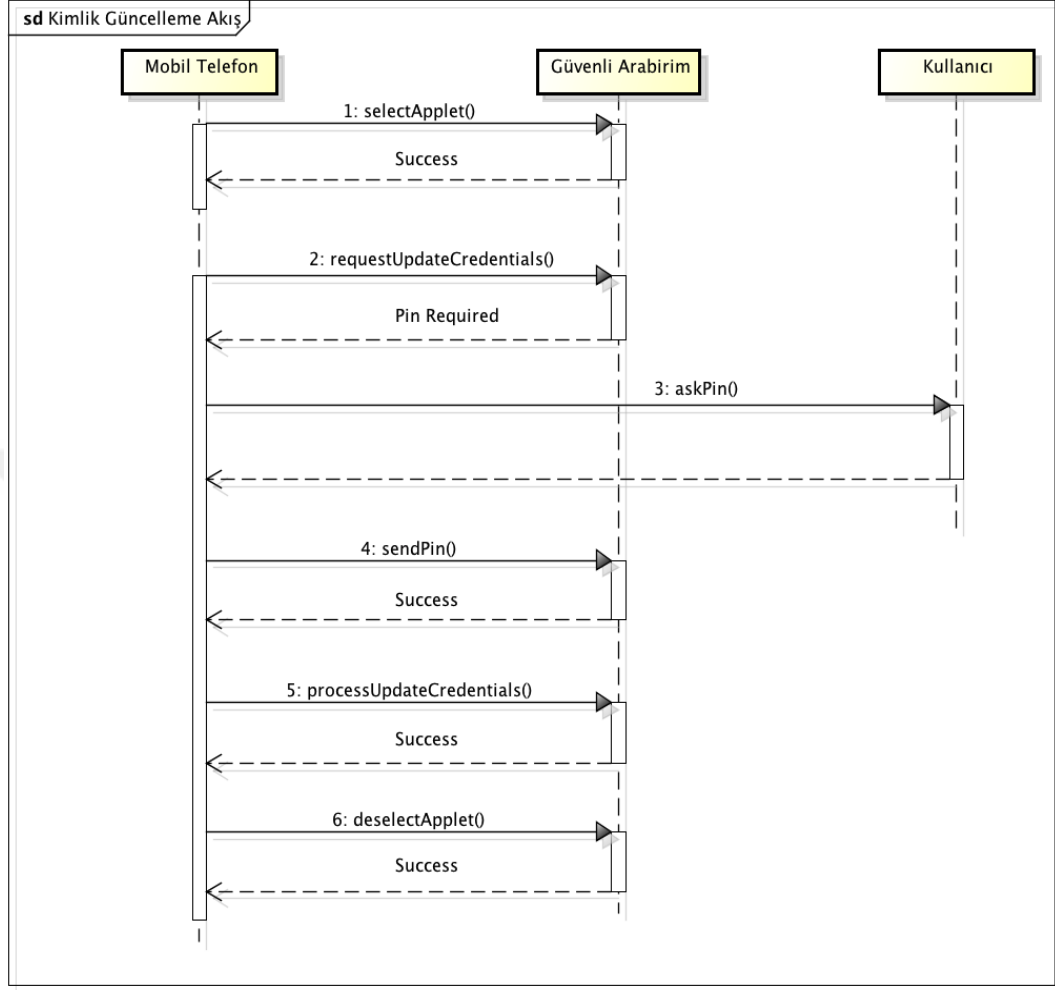


Şekil 4.3: PIN Güncelleme İşlemi Akış Şeması.

Buradaki akışlara göre aşağıdaki adımlar izlenmektedir:

1. Güvenli arabirim üzerinden Alcatel Mobil Cüzdan Applet'i seçiliyor.
2. Pin güncelleme komutu mesajı güvenli arabirime gönderiliyor. Güvenli arabirim kullanılmakta olan pini istiyor.
3. Telefon üzerinden kullanıcıya pin soruluyor.
4. Pin telefon tarafından güvenli arabirime gönderiliyor.
5. Yeni pin kullanıcıya soruluyor.
6. Yeni pin kullanıcıya tekrar soruluyor.
7. Pinler aynı ise pin güncelleniyor.
8. Güvenli arabirim bağlantısı kapatılıyor.

Benzer bir biçimde Şekil 4.4’de kimlik güncelleme işleminin akış şeması gösterilmiştir.

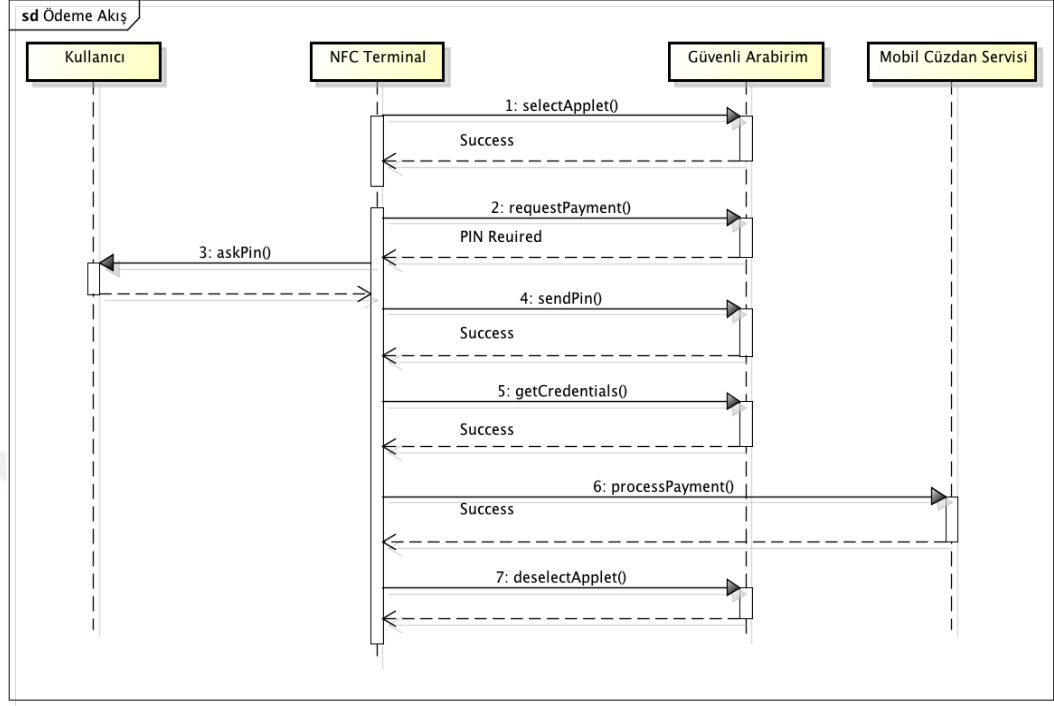


Şekil 4.4: Kimlik Bilgisi Güncelleme İşlemi Akış Şeması.

Şekil 4.5’te gösterilen ödeme işleminin adımları aşağıdaki gibidir:

1. Güvenli arabirim üzerinden Alcatel Mobil Cüzdan Applet’i seçiliyor.
2. Terminal tutar göndererek ödeme işlemini istiyor. Güvenli arabirim, ödenecek tutar yüksek olduğu için PIN girilmesini istiyor.
3. Kullanıcıya PIN soruluyor.
4. Kullanıcı tarafından girilen PIN güvenli arabirime gönderiliyor. Doğru ise güvenli arabirim tarafından başarılı mesajı dönüyor.
5. Terminal ödeme işlemini gerçekleştirebilmek için, güvenli arabirimden kullanıcı kimlik bilgisini istiyor. Daha önce PIN girildiği için güvenli arabirim, kimlik bilgisini terminale gönderiyor.
6. Terminal Mobil Cüzdan Servislerini kullanarak ödeme yapıyor.

7. Güvenli arabirim kapatılıyor.



Şekil 4.5: Ödeme İşlemi Akış Şeması.

4.2. G&D- GÜVENLİ ARABİRİM UYGULAMASI TASARIMI

Güvenli arabirim uygulamasının gerekli bileşenleri Giesecke & Devrient (G&D) SDK kullanılarak geliştirilmiştir. G&D SDK ayrıca geliştirilen uygulamaların test edilebilmesi için test araçlarını ve geliştirilen ürünün SIM üzerine aktarılması için gerekli araçları içermektedir [36].

Bir önceki kısımda belirtilen işlevler G&D Sky SIM ortamı kullanılarak geliştirilmiştir. Geliştirdiğimiz uygulama öncelikli olarak SIM üzerinde test edilmiştir. Bu test senaryoları aynı şekilde SIM üzerinde uygulanmıştır [36].

Geliştirilen Java Card Mobile Wallet uygulamasına APDU komutları ile erişilmektedir. Mesajlaşma yapısı aşağıdaki şekildedir [35]:

Tablo 4.1: APDU Mesaj Yapısı.

CLA	INS	P1	P2	Len	Data
Komut Sınıfı	Komut Kodu	Parametre	Parametre	Veri Boyutu	Veri

Örnek komut;

80 (CLA) 20(INS) 00(P1) 00(P2) 04(Len) 00 00 00 00 (Data)

Geliştirilen uygulamanın test senaryoları aşağıdaki şekilde listelenmiştir.

Test 1 - PIN Doğrulama

Host > Applet'i seç

apdu:0;target=A0.00.00.00.90.00.00.01.02.03.04.05

Host > PIN doğrulama komutu

802000000400000000

SE < 9000

İşlem başarılı bir şekilde tamamlandı

Test 2 - 30 TL altında ödeme yapma (hızlı ödeme)

Host > Applet'i seç

apdu:0;target=A0.00.00.00.90.00.00.01.02.03.04.05

Host > Ödeme komutu

80400000020001

SE < 00019000

İşlem başarılı

Test 3 - 30 TL üzerinde ödeme yapma

Host > Applet'i seç

apdu:0;target=A0.00.00.00.90.00.00.01.02.03.04.05

Host > Ödeme komutu

80400000020022

SE < 6E02

PIN girişi gerekli

Host > PIN doğrulama komutu

802000000400000000

Host > Ödeme komutu

80400000020022

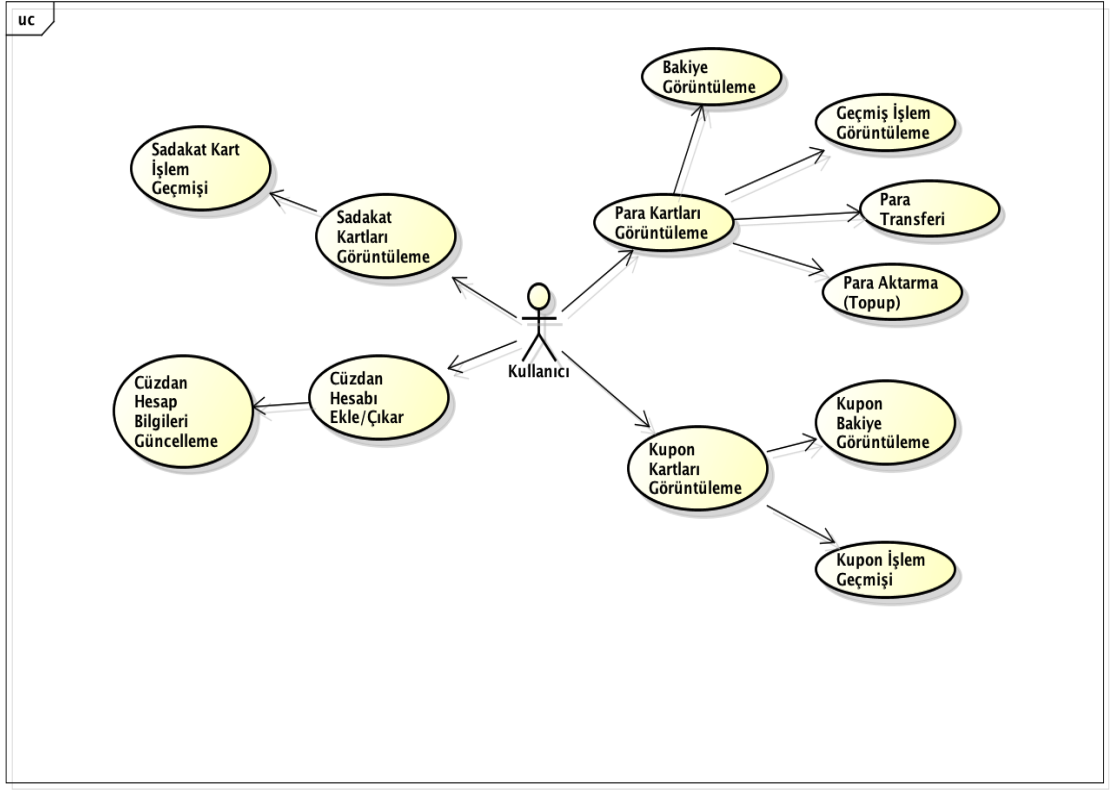
SE < 00229000

İşlem başarılı

Uygulama testleri simulator ortamında, başarı ile gerçekleştirildikten sonra, G&D kit kullanılarak SIM kart üzerinde ve geliştirilen test uygulaması ile BlackBerry cep telefonunda tekrar gerçek ortam olarak gerçekleştirilmiştir.

4.3. ALCATEL LUCENT MWS KULLANAN MOBİL İSTEMCİ TASARIMI

Alcatel-Lucent MWS hizmeti detaylı olarak incelendikten sonra, son kullanıcı tarafından kullanılacak bu hizmetlerin mobil cihaz istemcisi tasarlanmıştır. Bu tasarım doğrultusunda, MWS hizmeti ile veri akışını sağlayacak olan ağ katmanı, MWS tarafından sağlanan RESTful web servisleri kullanarak geliştirilmiştir. Öncelikli olarak, yapılacak işlemler belirlenmiş ve bu işlemler Şekil 4.6'da gösterilmiştir.

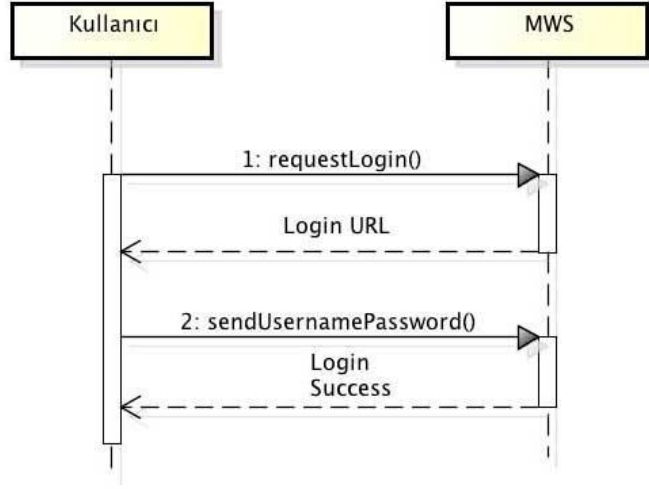


Şekil 4.6: Kullanıcı İşlemleri.

Kullanıcı güvenli ara birime hesap ekleyip çıkartabilir, hesap bilgilerini güncelleyebilir, sadakat kartlarını, sadakat kartların işlem geçmişlerini; para kartları, para kartların bakiyesini, geçmiş işlemlerini; kupon kartlarını, kupon kartlarının bakiyesini, işlem geçmişlerini görüntüleyebilir, başka hesaba para transferi ve aktarması yapabilir.

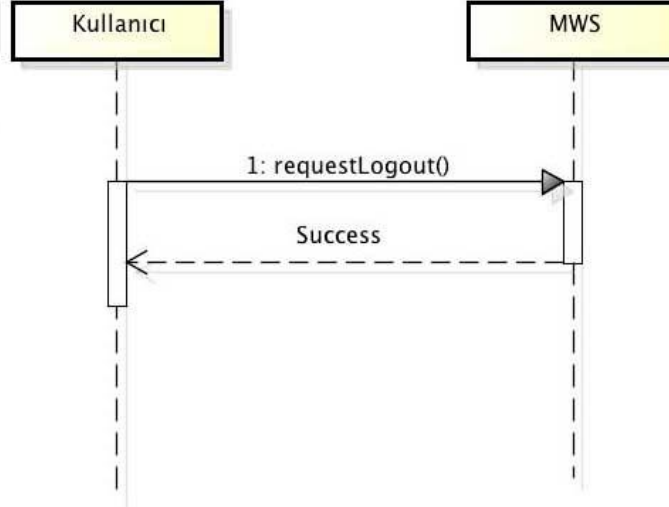
Kullanıcı login işlemi Şekil 4.7’de gösterilmiştir. Buna göre kullanıcı;

1. MWS sunucusundan login istemi gönderir.
2. Sunucu login için adres gönderir.
3. Login adresine kullanıcı bilgileri gönderilir.
4. Login işlemi başarılıdır.



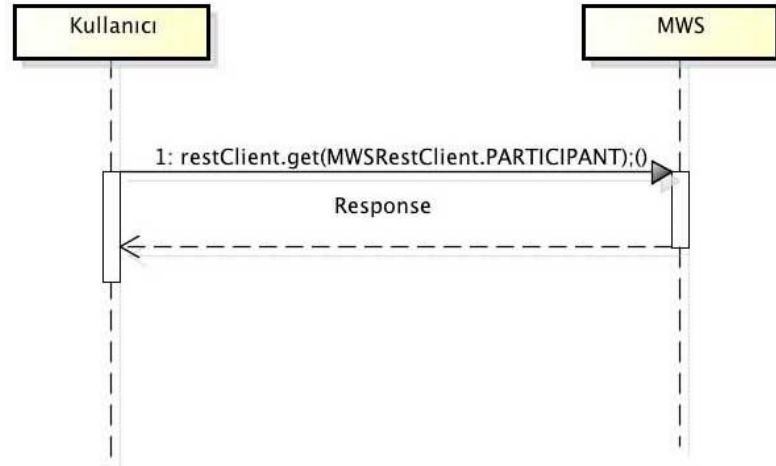
Şekil 4.7: Login İşlemi.

Şekil 4.8’de çıkış (logout) işlemi gösterilmiştir. Buna göre, kullanıcı sunucuya çıkış isteği gönderir ve sunucu çıkış yapar.



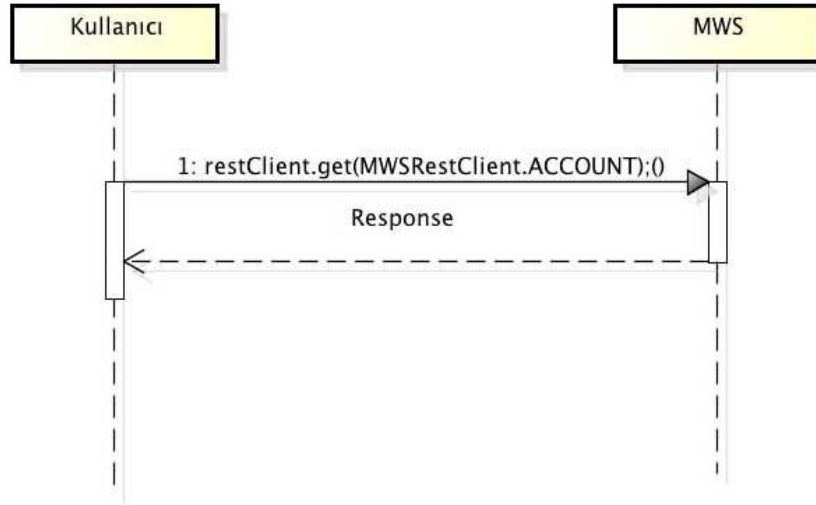
Şekil 4.8: Çıkış İşlemi.

Kullanıcı hesap bilgilerini isteme istemi Şekil 4.9’da gösterilmiştir.

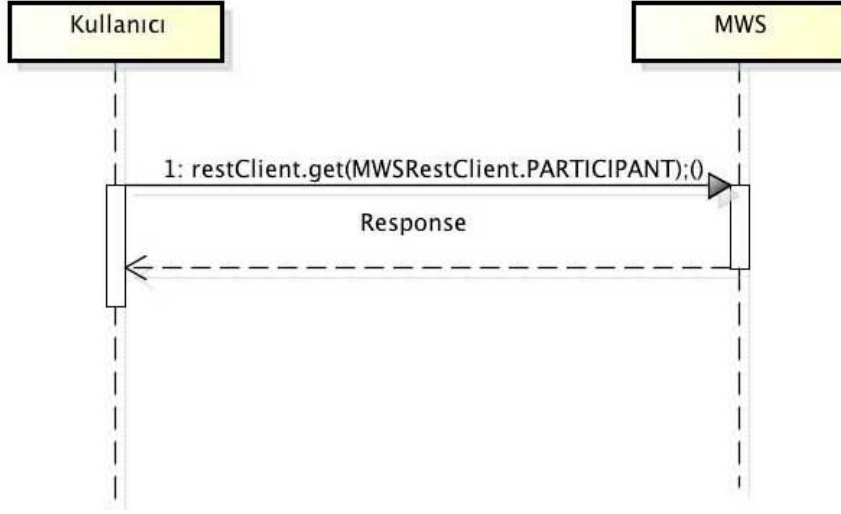


Şekil 4.9: Kullanıcı Hesap Bilgileri İşlemi.

Benzer bir şekilde para kart/kupon/sadakat kart hesap bilgilerini isteme işlemi Şekil 4.10'da, bu hesapların işlem geçmişi ise Şekil 4.11'de gösterilmiştir.

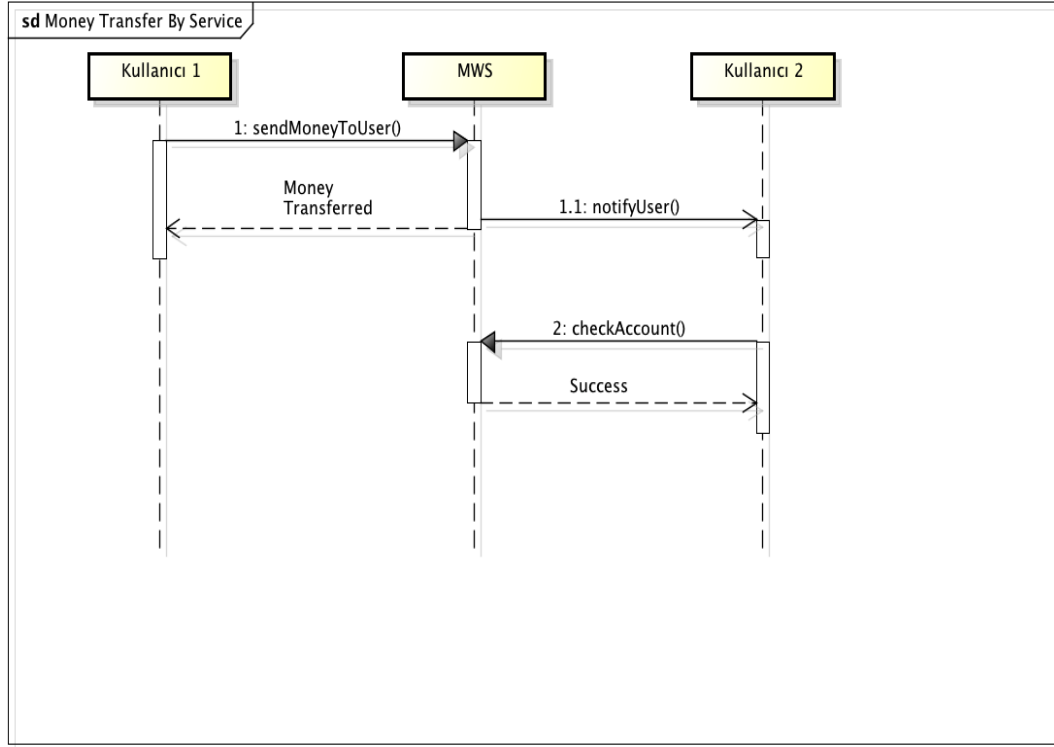


Şekil 4.10: Hesap Bilgileri İsteme İşlemi.



Şekil 4.11: Hesap Hareketleri İsteme İşlemi.

Şekil 4.12’de Alcatel Mobil Cüzdan Servislerini kullanarak bir kullanıcının bir kullanıcı ile yapacağı para transferi akış şeması gösterilmiştir.



Şekil 4.12: Para Transferi Akış Şeması.

Şekil 4.12’de görüldüğü üzere kullanıcı aşağıdaki şekilde para transfer işlemini yapar:

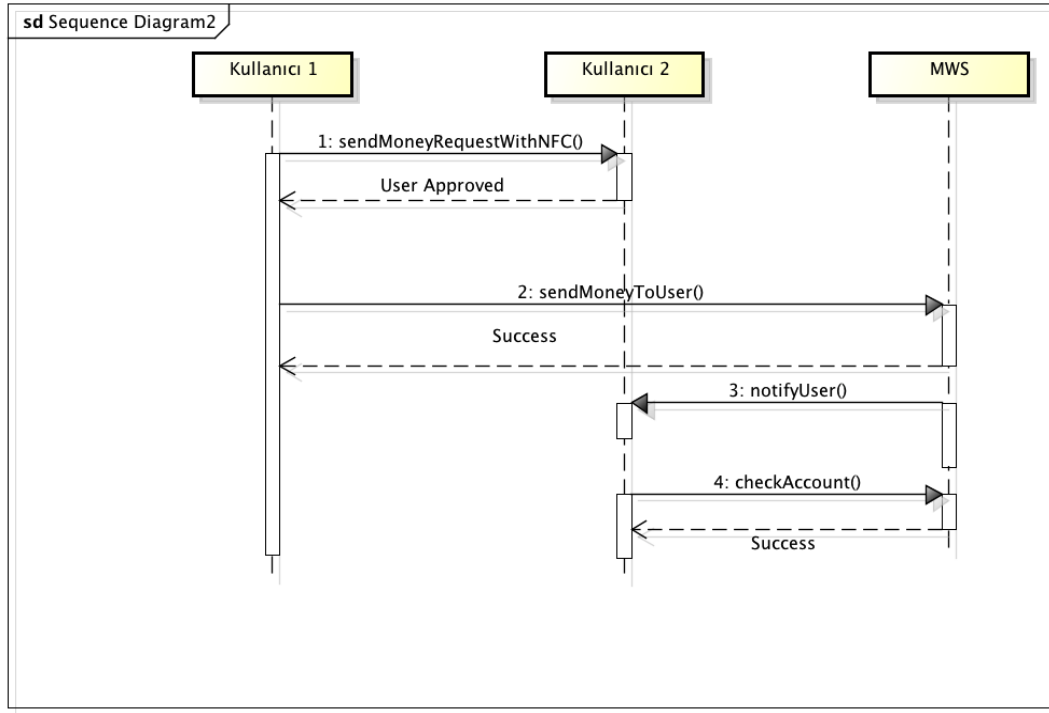
- 1-Para göndermek istediği kullanıcıyı ve para miktarını belirler.

2-Belirlenen parametreler ile MWS servisine para gönderme istemi gider.

3-MWS bu işlemi onaylar ve gerekli bildirimleri yapar.

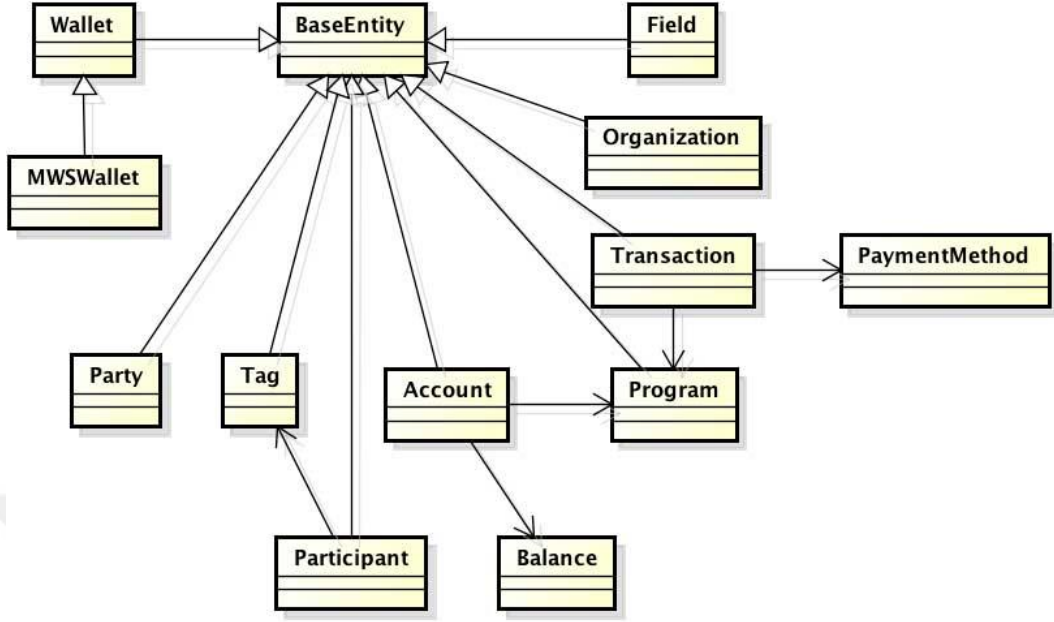
Şekil 4.13'te Alcatel Mobil Cüzdan Servislerini kullanarak bir kullanıcının başka bir kullanıcıya yapacağı para transferi akış şeması gösterilmiştir. Buna göre:

- 1- Para transferi yapmak isteyen kişi diğer telefona cihazını dokundurur.
- 2- Karşı taraftaki kullanıcı para transfer uyarısını onaylar.
- 3- Gönderilmek istenen para MWS'e gönderilir.
- 4- Başarılı sonuç döner.
- 5- MWS sunucu gerekli bildirimleri yapar.



Şekil 4.13: Kullanıcıdan Kullanıcıya Para Transferi Şeması.

Şekil 4.14'te mobil istemcilerde tutulacak olan verinin sınıf diyagramı gösterilmiştir.

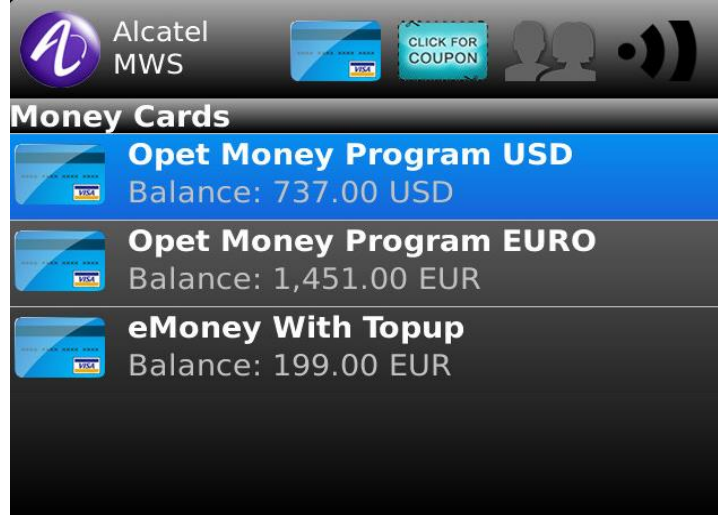


Şekil 4.14: Sınıf Yapısı.

4.3.1. Mobil Telefonda Uygulamanın Kullanıcı Arayüzlerinin Geliştirilmesi ve Test Edilmesi (Blackberry)

Mobil uygulama arayüzleri geliştirilirken platformun sağlamış olduğu arayüz kütüphaneleri kullanılarak tasarlanmıştır. Her platform kendine özgü kullanıcı deneyimine sahip olduğu için, tasarımlarda kullanıcı deneyimi, kullanılabilirlik, performans göz önüne alınmıştır. Bu kısımda BlackBerry platformu için kullanıcı arayüzleri ve işlevlerinden bahsedilecektir.

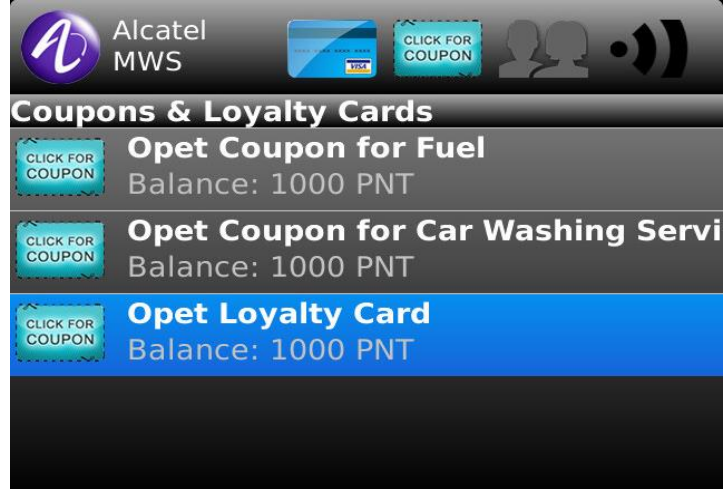
Bu uygulama kapsamında uygulamayı test edebilmek için Alcatel-Lucent MWS hizmetlerini kullanarak test amaçlı hesaplar, para kartları, kuponlar ve sadakat kartları tanımlanmıştır. Bu tanımlamaların hiç birinin gerçek firma ile bir ilgisi olmayıp, sadece test amaçlıdır.



Şekil 4.15: Para Kartlar Arayüzü.

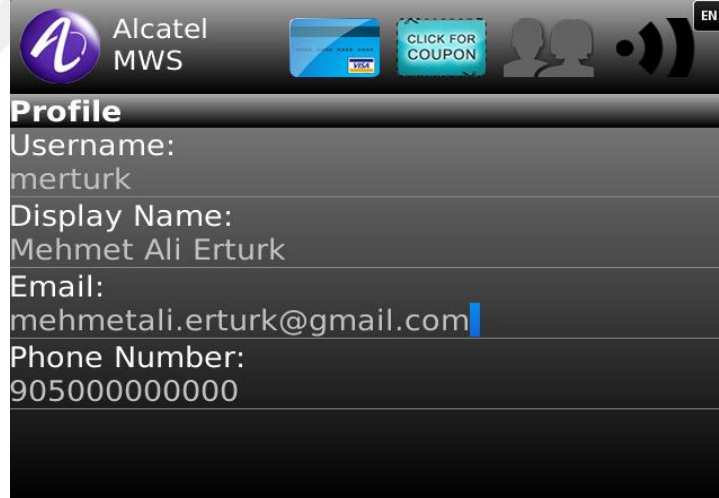
Sisteme giriş yapmış bir kullanıcının görmüş olduğu ekran Şekil 4.15’de gösterilmiştir. Şekil 4.15’de gösterilen ekran görüntüsünde ana menü işlemleri ikonlar şeklinde üst tarafta ve o ikona bağlı içerik ekranın orta kısmında gözükmektedir. Uygulama açıldığında internet bağlantısı ile Alcatel-Lucent MWS hizmetine bağlanılarak, kullanıcının para kartları, sadakat kartları ve kuponları sorgulanmaktadır. Kullanıcının ilk görmüş olduğu ekran para kartlarıdır. Ayrıca ekranda gözüken herhangi bir hesap seçildiğinde ek menü ile kullanıcının yapabileceği işlemler gözükmektedir.

Şekil 4.15’de Alcatel Mobil Cüzdan Servisleri kullanılarak Para Kartlar Arayüzü gösterilmiştir. Buradan Opet Dolar Para Programı, Opet Euro Para Programı veya eMoney With Topup para kartları seçilebilir. Üst menüden kupon ve sadakat kartları seçilirse, kullanıcıya tanımlı sadakat kartları ve bu kartların bakiyeleri görünmektedir.



Şekil 4.16: Kuponlar ve Sadakat Kartlar Arayüzü.

Şekil 4.16’da Alcatel Mobil Cüzdan Servisleri kullanılarak Kuponlar ve Sadakat Kartlar Arayüzü gösterilmiştir. Buradan Opet Yakın Kuponu, Opet Yıkama Kuponu veya Opet Sadakat Kartı seçilerek işlem yapılabilir.



Şekil 4.17: Kullanıcı Profili Arayüzü.

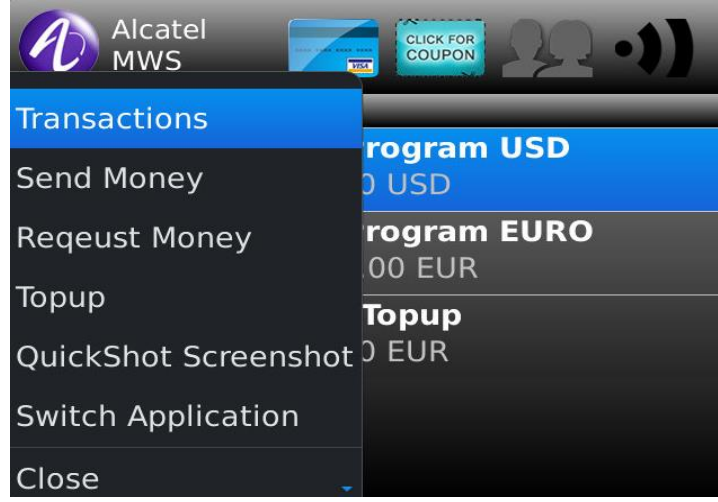
Üst menüden profil ikonu olan 3. adım seçildiğinde, kullanıcı hesap bilgileri gözükmetedir. Şekil 4.17’de Alcatel Mobil Cüzdan Servisleri kullanılarak, kullanıcı hesap bilgilerini güncelleyebilir veya uygulamadan çıkış yapılabilir.



Şekil 4.18: NFC Durum Arayüzü

Şekil 4.18’de Alcatel Mobil Cüzdan Servisleri kullanılarak Yakın Alan İletişimi durum ekranı gösterilmiştir. Bu menü ise kullanıcı NFC terminal ile para ödemek istediği zaman, güvenli arabirim aktif hale gelerek NFC terminali ile iletişim başlar. Güvenli arabirimde tanımlı olan ödeme işlemi buradan yapılmaktadır.

Ana menünün dışında kullanıcının yaptığı ek işlemler, BlackBerry telefonlarında bulunan fonksiyon tuşu ile gerçekleştirilmektedir. Para kartları ekranından herhangi bir hesap seçiliyken, kullanıcı bu fonksiyon tuşuna basarsa Şekil 4.19’da gözüken işlem listesi gelmektedir. Bu işlemler incelenirse; kullanıcının bir başka kullanıcıya para gönderme, başka kullanıcıdan para isteme, geçmiş işlemleri görüntüleme, hesaba para ekleme ve uygulamayı kapatma gibi işlemleri gerçekleştirir. Eğer kullanıcı işlem geçmişini görmek isterse *Transactions* seçeneğini seçip Şekil 4.20’de gözüken ekrana gelmektedir.



Şekil 4.19: İşlemler Arayüzü.



Şekil 4.20: Para Kart Geçmiş İşlemler Arayüzü.

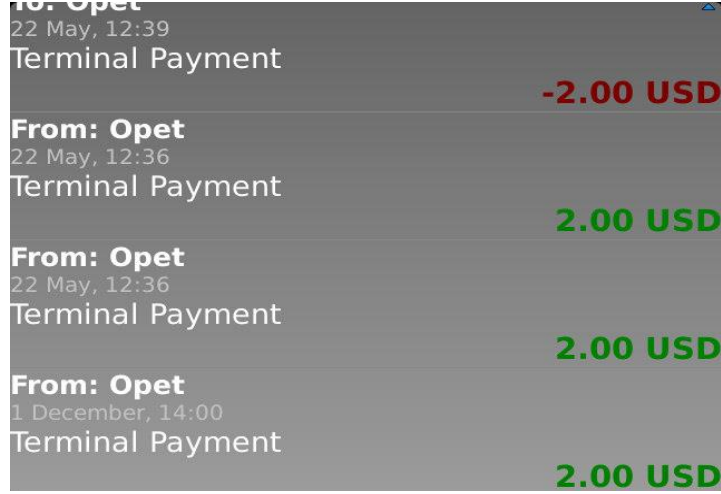
Şekil 4.20’de Alcatel Mobil Cüzdan Servisleri kullanılarak Opet Dolar Para Program’ının geçmiş işlemleri gösterilmiştir. Ödeme tarihleri ve saatlariyle 3 farklı terminal ödeme işlemi görülmektedir. Bu ekranda:

To: Opet : Paranın kime gittiği

22 May, 13:56 : İşlem tarihi

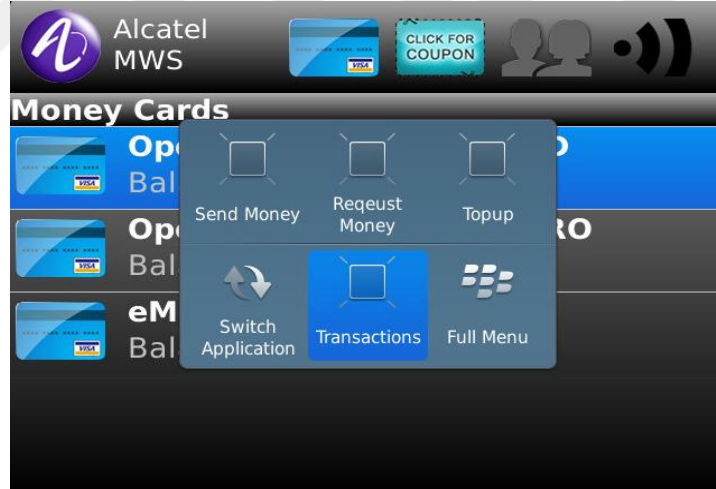
Terminal Payment : Yapılan işlemin açıklamasını,

-6.00 USD : Yapılan işlemin tutarı gözükmemektedir.



Şekil 4.21: Para Kart Geçmiş İşlemler Arayüzü.

Şekil 4.21’de Alcatel Mobil Cüzdan Servisleri kullanılarak Opet Dolar Para Program’ının diğer geçmiş işlemleri gösterilmiştir. Şekilde görüldüğü üzere kırmızı renkle belirtilen rakamlar hesaptan ödenen işlemler, yeşil renkle belirtilen rakamlar ise hesaba gelen işlemlerdir.



Şekil 4.22: Menü Kullanım Arayüzü.

Şekil 4.22’de Alcatel Mobil Cüzdan Servisleri kullanılarak Şekil 4.19’daki görüntüden başka bir şekilde, bir kullanıcının yapabileceği işlemler gösterilmiştir. Kullanıcının bir başka kullanıcıya para gönderme, başka kullanıcıdan para isteme, geçmiş işlemleri görüntüleme, hesaba para ekleme, ekran görüntüsü alma ve uygulamayı kapatma gibi işlem seçenekleri bulunmaktadır.



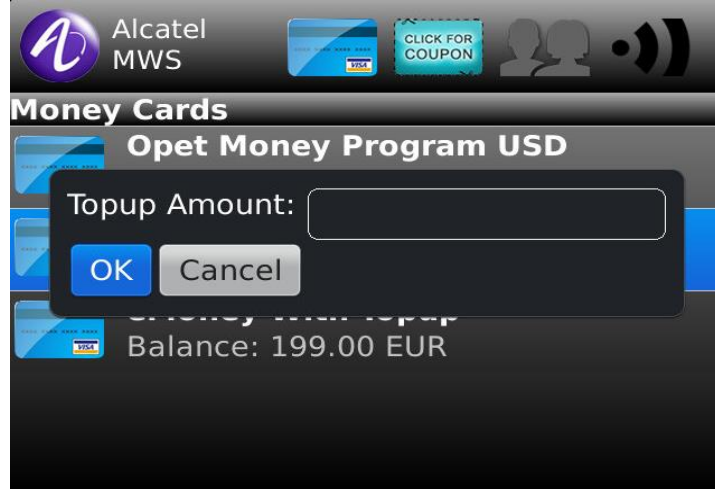
Şekil 4.23: Para Kart Geçmiş İşlemler Arayüzü.

Şekil 4.23’de Alcatel Mobil Cüzdan Servisleri kullanılarak bu sefer Opet Euro Para Program’ının geçmiş işlemleri gösterilmiştir. Ödeme tarihleri ve saatlariyle 3 farklı terminal işlemi görülmektedir.



Şekil 4.24: Hesap Durum Arayüzü.

Şekil 4.24’de Alcatel Mobil Cüzdan Servisleri kullanılarak bir kullanıcının hesap durumu gösterilmiştir.



Şekil 4.25: Hesaba Para Ekleme Arayüzü.

Şekil 4.25’de Alcatel Mobil Cüzdan Servisleri kullanılarak bir kullanıcının hesabına yapacağı para ekleme işlemi gösterilmiştir. Para ekleme arayüzü ile belirtilen para, hesabın bağlı olduğu kurumla birlikte bir ödeme sayfasına yönlendirilmektedir. Bu servisler ile kredi kartından cüzdanına para aktarabilmektedir.

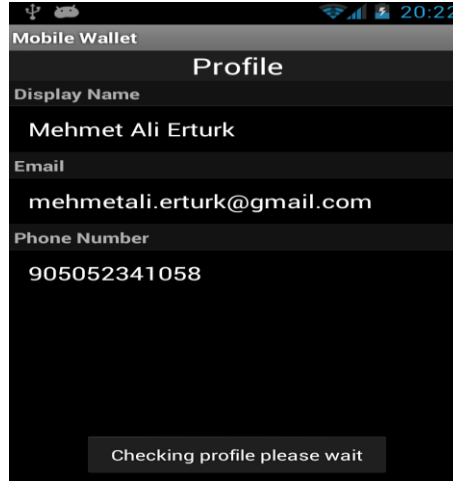
4.3.2. Mobil Telefonda Uygulamanın Kullanıcı Arayüzlerinin Geliştirilmesi ve Test Edilmesi (Google Nexus S (Android))

Bu başlık altında Android uygulaması için geliştirilen arayüzler bulunmaktadır. Arayüzler tasarlanıp geliştirilirken, Android platformunun kullanıcı deneyimi ele alınarak bir tasarım yapılmıştır.



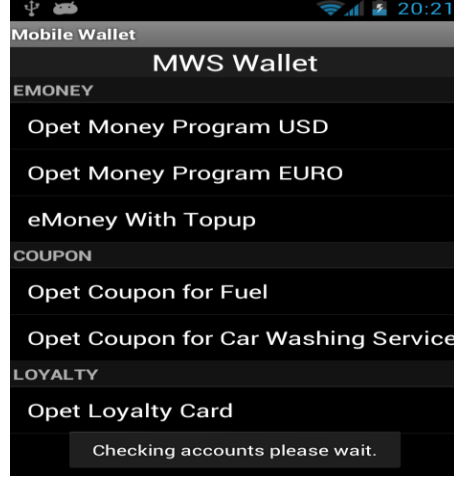
Şekil 4.26: Kullanıcı Giriş Sayfası Arayüzü.

Şekil 4.26’da Alcatel Mobil Cüzdan Servisleri kullanılarak kullanıcı giriş sayfası ekranı gösterilmiştir. Kullanıcı oturup açmak istediğinde Alcatel-Lucent MWS tarafından sağlanan login ekranına gönderilmektedir. Kullanıcı bilgilerini girerek oturum açabilir.



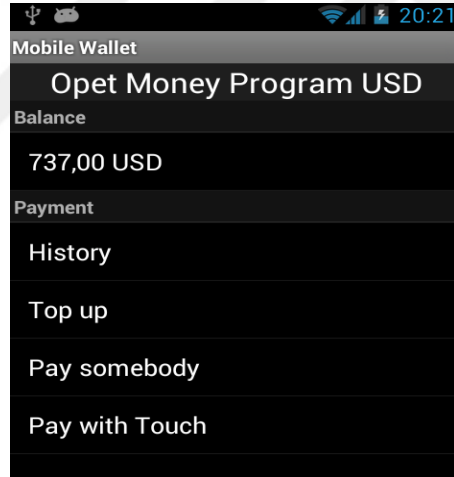
Şekil 4.27: Kullanıcı Profil Arayüzü.

Şekil 4.27’de Alcatel Mobil Cüzdan Servisleri kullanılarak bir kullanıcının profil bilgileri ekranı gösterilmiştir. Kullanıcı bu arayüzü kullanarak hesap bilgilerini güncelleyebilir, çıkış yapabilir.



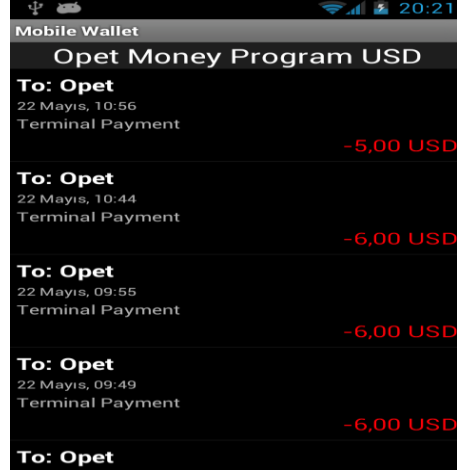
Şekil 4.28: Mobil Cüzdan Ana Menü Arayüzü.

Şekil 4.28’de Alcatel Mobil Cüzdan Servisleri kullanılarak Mobil Cüzdan ana menüsü gösterilmiştir. Buradan para kartlar, kuponlar ve sadakat kartları seçilebilmektedir.



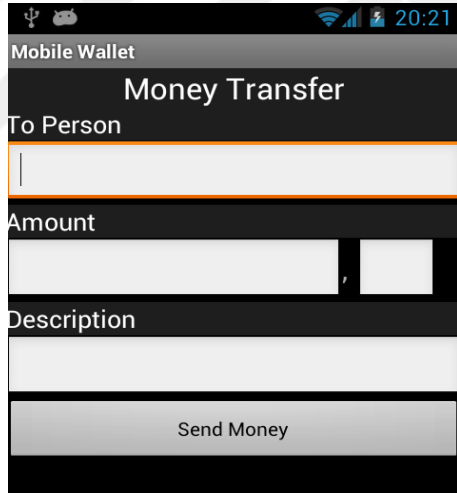
Şekil 4.29: Para Kart Arayüzü.

Herhangi bir para türü seçildiğinde Şekil 4.29’da gözüken Alcatel Mobil Cüzdan Servisleri kullanılarak yapılabilecek olan işlemler gözükmemektedir. Buradan bu programla ilgili geçmiş işlemler, para ekleme ve ödeme gibi işlemleri seçilebilir. Kullanıcı para geçmişini seçtiği zaman Şekil 4.30’da gözüken işlem geçmişi gözükmemektedir.



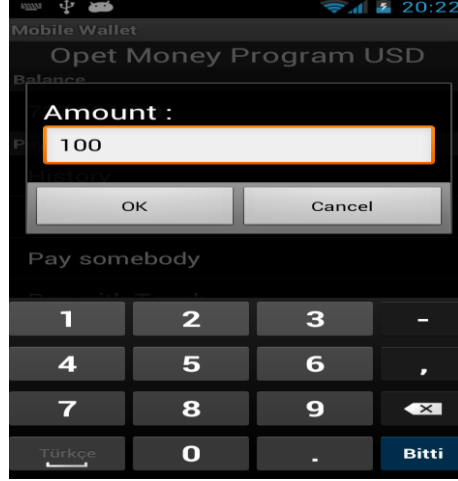
Şekil 4.30: Para Kart Geçmiş İşlemler Arayüzü.

Şekil 4.30’da Alcatel Mobil Cüzdan Servisleri kullanılarak Opet Dolar Para Program’ının 4 farklı zamandaki geçmiş işlemleri gösterilmiştir.



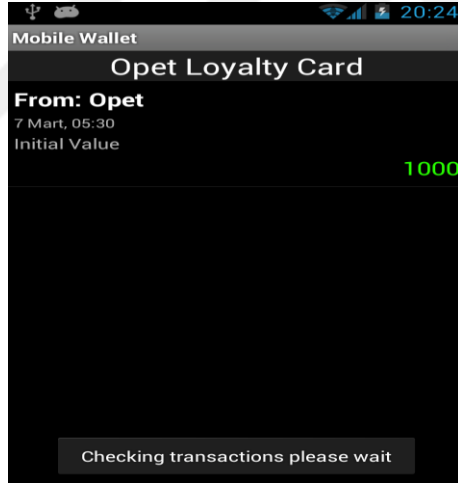
Şekil 4.31: Başka Kullanıcıya Para Transfer Arayüzü.

Şekil 4.31’de Alcatel Mobil Cüzdan Servisleri kullanılarak bir kullanıcının başka bir kullanıcıya yapacağı para transferi ekranı gösterilmiştir. Buradan para gönderilecek kişinin bilgileri girilerek işlem gerçekleştirilebilir.



Şekil 4.32: Hesaba Para Ekleme Arayüzü.

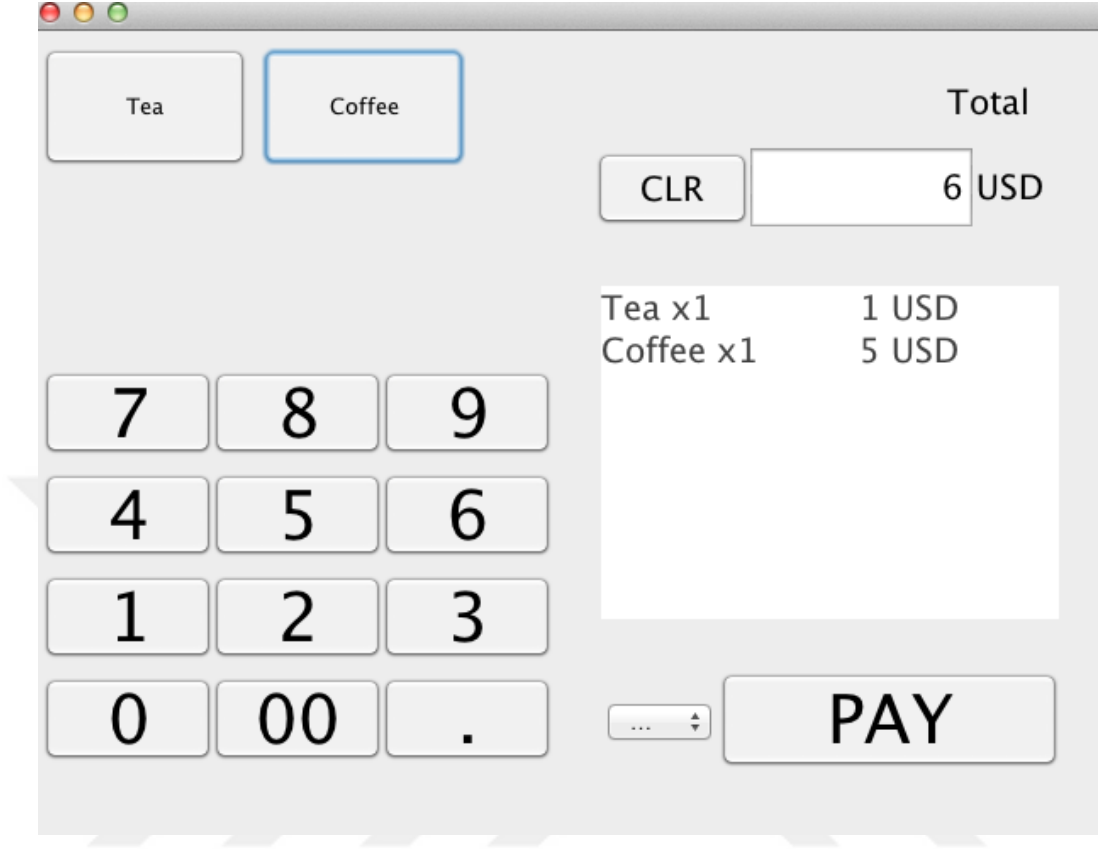
Şekil 4.32’de Alcatel Mobil Cüzdan Servisleri kullanılarak hesaba para ekleme ekranı gösterilmiştir.



Şekil 4.33: Opet Sadakat Kart Arayüzü.

Şekil 4.33’de Alcatel Mobil Cüzdan Servisleri kullanılarak Opet Sadakat Kart işlemleri gösterilmiştir.

4.3.3. Sanal POS Uygulaması



Şekil 4.34: Hesap Ödeme Arayüzü.

Geliştirilen prototipi test etmek için Şekil 4.34’de gösterilen bir sanal pos uygulaması gerçekleştirilmiştir. Sanal pos ekranındaki içecek hesap uygulama arayüzü gösterilmiştir. Bu uygulama ACS NFC Reader (Okuyucu) terminali ile bağlantı kurarak ödeme işleminin gerçekleştirilmesini sağlamaktadır. Ekrandan seçilen tutar ile birlikte ödeme başlatıldığında, NFC okuyucu ekranında ödeme yapılacak tutar gözükmemektedir. Şekil 4.35’de Yakın Alan İletişimi (NFC) Okuyucusu ödenmesi gereken tutarı göstermektedir.



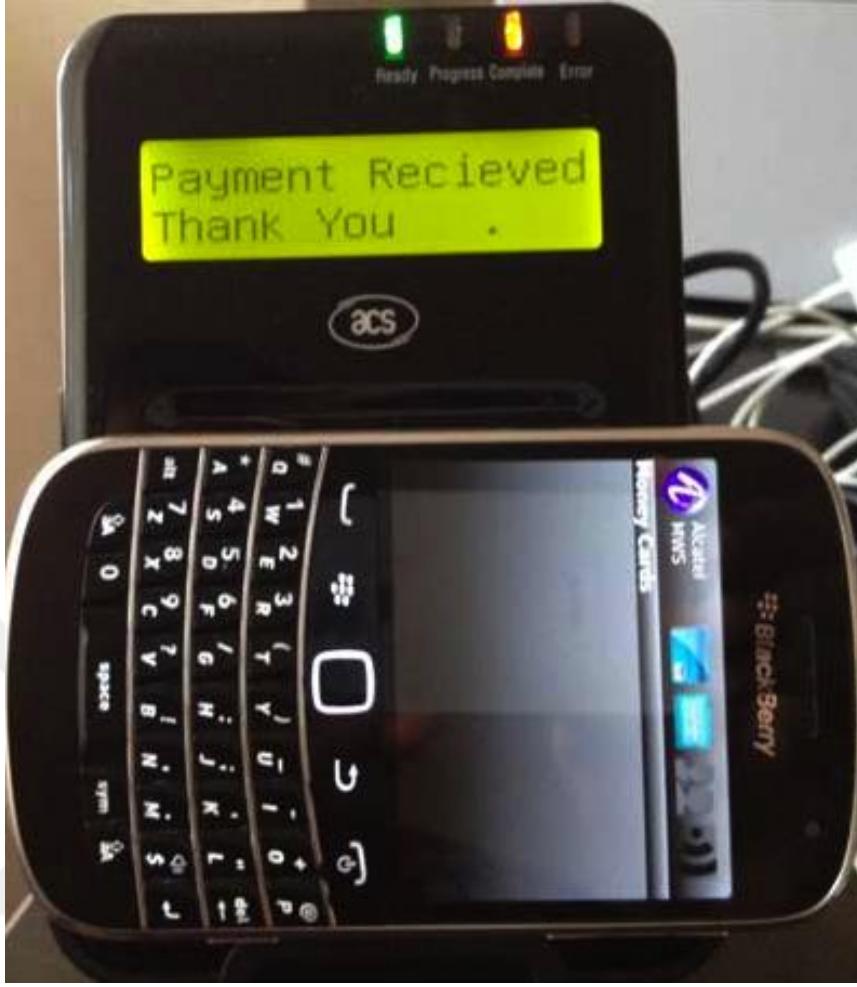
Şekil 4.35: NFC Okuyucu Ödeme Tutarı Görüntüsü.

NFC destekli BlackBerry Bold 9900 üzerinde çalışan Alcatel-Lucent MWS mobil istemcisi, terminale yaklaştırıldığında, mobil cep telefonu üzerinde bulunan SIM kartın güvenli arabirimi ile terminal arasında bir iletişim başlar ve terminal kullanıcı bilgilerini alarak mobil cüzdan hizmeti ile ödeme işlemini gerçekleştirir.

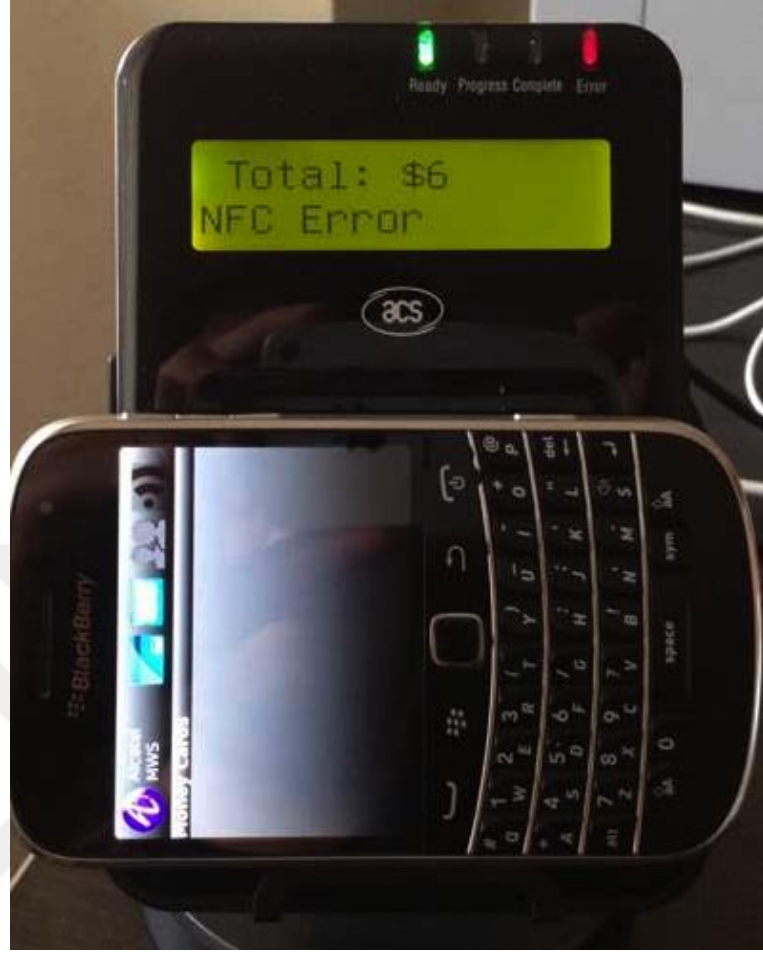


Şekil 4.36: NFC Okuyucu Ödeme İşlemi Görüntüsü.

Şekil 4.36’da ödeme işleminin yapıldığını terminal ekranından görebiliriz. Eğer ödeme işlemi başarılı ise Şekil 4.37’de gözüken ödemenin tamamlandığını gösteren bir teşekkür mesajı gösterilmektedir.



Şekil 4.37: NFC Okuyucu Ödeme İşlemi Sonucu Görüntüsü.



Şekil 4.38: NFC Okuyucu Ödeme İşlemi Hatası Görüntüsü.

Eğer NFC ile ya da güvenli arabirim arasında bir problem oluştuğunda Şekil 4.38’de gözüken NFC hatası görünecektir.



Şekil 4.39: Para Kart Geçmiş İşlemler Görüntüleme Arayüzü (Blackberry).

Başarılı ödeme işlemini Şekil 4.39’da gözüken BlackBerry ekran görüntüsünde, yapılan ödeme işleminin Opet Dolar Para Program’ının geçmiş işlemlerinde gösterilmiştir. 6 \$ tutarındaki terminal ödeme işlemi görülmektedir.

5. TARTIŞMA VE SONUÇ

Yeniliklerin, katma değer yaratan servislerin ve rekabetçi ortamın sürekli olarak arttığı telekomünikasyon pazarında, servis sağlayıcılar ve ekipman sağlayıcıları son müşteriyi kendi şebekesine ve hizmet firmasına çekebilecek düzeyde, altyapıya katma değer sağlayacak uygulamalara ihtiyaç duymaktadırlar. Bu nedenle özellikle büyüyen mobil pazarda uygulama tasarımı ve geliştirilmesi, bunun yanı sıra Blackberry ve Android gibi mobil işletim sistemleri üzerinde koşabilecek geniş kapsamlı uygulama geliştirme faaliyetleri çok önemli hale gelmiştir.

Son yıllarda, insanların günlük ihtiyaçlarını karşılamaya yönelik uygulamalarda kullanılmak üzere, temassız (contactless) kartların geliştirilmesi için, pek çok araştırma ve geliştirme yapılmıştır. Bu temassız kartlar, normal kartlardan çok daha kullanışlı, hızlı ve farklı tipte işlerin (mobil cüzdan uygulamaları, bilgi edinme, yapılarda güvenlik uygulamaları vs.) tek bir kartla yapılmasına olanak sağlayan yapıdadırlar. NFC (Near Field Communication – Yakın Saha İletişimi) çipler ise, bu tip uygulamaların en çok kullanıldığı cihaz olarak dikkat çekmektedirler. 2004'ten bu yana, bu konuda bilimsel ve ticari alanlarda yapılan araştırma ve uygulama sayısı olağanüstü bir hızla artmaktadır.

Bu tez çalışmasında bu aşamaya gelene kadar, neler yaptığımızı kısaca özetleyecek olursak, öncelikle Java ME teknolojisiyle mobil telefonlar üzerinde Android ve Blackberry gibi mobil uygulama mimarileri ve çalışma şekilleri incelenmiştir. Daha sonra, NFC destekli telefon mimarisi, güvenli birim yapısı, güvenli birimi cihaz ve SIM kart (UICC - Universal Integrated Circuit Card - Küresel Bütünleşik Devre Kartı) üzerinde destekleyen uygulama mimarileri incelenmiştir.

Uygulama geliştirmeye başlanarak, Alcatel-Lucent'in mobil cüzdan servisleri ile entegre çalışacak olan SIM (UICC) kart üzerindeki güvenli birimle ve telefon üzerindeki güvenli birim üzerinde çalışan Java Card uygulaması tasarlanmış ve geliştirilmiştir.

Java Card uygulamasına ek olarak güvenli arabirim uygulamasını test etmek için bir Java ME uygulaması geliştirilmiştir. Uygulanan test senaryoları sonunda güvenli arabirimin sorunsuz çalıştığı tespit edilmiştir. Ayrıca, yüksek tutarlı ödemeler için şifre tabanlı ödeme üzerinde çalışılmış, satış noktası cihazı veya telefon tabanlı olarak telefon veya SIM kart (UICC)

üzerindeki güvenli birimde saklanan şifre üzerinden gerçekleşecek ödeme mimarisi ve senaryosu gerçekleştirilmiştir.

Android ve Blackberry platformlarında Alcatel-Lucent'in mobil cüzdan servisleri ile entegre, güvenlik birimini cihaz üzerinde bir çip olarak tutan NFC destekli Blackberry ve Google Nexus S telefonları üzerinde çalışacak olan Java Card uygulaması cihaz yerine simülasyon üzerinde geliştirilip aynı simülasyonda test edilmiştir. Bu uygulamanın geliştirilmesi Java ME teknolojisi ile gerçekleştirilmiştir.

Para yüklemesi, temassız ödeme, kişiden kişiye para transferi, işlem geçmişi görüntülenmesi, kupon ve puan gibi müşteri sadakat uygulamaları geliştirilmiştir. Daha sonra, uygulamanın kullanıcı ile etkileşimi sağlayacak arayüzleri, hazırlanan senaryolarla geliştirilip kullanılabilirlik ve fonksiyonellik testlerinden başarıyla geçmiştir.

Son olarak da, tüm bileşenler bir araya getirilerek prototip olarak uygulama oluşturulup, bir bütün olarak farklı senaryolarla hem fonksiyonellik hem de kullanılabilirlik testlerine tabi tutulup Blackberry ve Android tabanlı Google Nexus S cihazlarında başarılı sonuçlar alınmıştır.

Sonuç olarak, bu yüksek lisans tez çalışmasında, üniversite-sanayi işbirliğinin güzel bir örneği olan San-Tez programı sayesinde, Alcatel-Lucent Türkiye ve İstanbul Üniversitesi Bilgisayar Mühendisliği Bölümü ortaklığında, cep telefonlarının mobil cüzdan (mobile e-wallet) olarak kullanılabilmesi sağlanmış ve NFC teknolojisini kullanan Blackberry ve Android işletim sistemi üzerine telefon istemcisi ile GSM SIM kart üzerine entegre güvenli birimi kullanan SIM kart uygulaması geliştirilmiştir. Bu bağlamda, Alcatel-Lucent Touchatag mobil cüzdan servisi kapsamında sunulmuş olan tüm servislerin güvenli ve hatasız bir şekilde kullanılmasını sağlayan, yaygın kullanılabilirlik yönünden performans ve fonksiyonel testlerinin tamamlanarak insanların hayatını daha da kolaylaştıracak bir uygulama geliştirilmeye çalışılmış, tez çalışması sonuçlarının ülkemize katma değer katacak bir ürüne dönüştürülmesi sağlanmıştır.

Araştırma ve geliştirme aktivitelerine ilaveten, oluşturulan uygulama kapsamında kullanılacak teknoloji ve protokollerle ilgili, çeşitli sunumlar gerçekleştirilmiştir. Alcatel-Lucent Teletaş

laboratuvarında yer alan ve çalışmanın son aşamasında entegrasyon için kullanılan NFC okuyucu, NFC etiketler ve servis sunucusu gibi Touchatag ürünü cihazlar firma tarafından sağlanmıştır. Bunun yanında, firmanın AR-GE içindeki mühendisleri teknik düzeyde (örneğin; mobil teknolojiyi yakından tanıma, Java programlama dilini daha iyi ve rahat kullanabilme, Touchatag ürününün tanıtılması ve servislerinin bizzat gösterilmesi vs...) yardımcı olmuşlardır.

Gelecekte, bu tez çalışmasında yapılan araştırmalardan ve uygulamalardan elde edilen bilgi ve tecrübe birikimi kullanılarak çok sayıda yayın hazırlanması ve farklı projeler geliştirilmesi mümkündür. Kısa zamanda, NFC çipli mobil cihazların yaygınlaşmasıyla, dünyada yüksek güvenli mobil cüzdan uygulamalarına olan ihtiyaç artacak ve bu uygulamaların henüz geliştirilme aşamasında olması, geniş bir araştırma alanı ortaya çıkmasına ve geliştirilen ürünlerin yurtdışına ihraç edilebilme potansiyelini artırmaktadır.

Ayrıca bu tez kapsamında ortaya çıkan uygulama kullanılarak gerçekleştirilen uygulamanın performansı ve güvenilirliği ile ilgili gerçek veriler elde edilebilir. Bu veriler kullanılarak nelerin optimize edilebileceği ile ilgili detaylı bir araştırma yapmak mümkün olabilir.

KAYNAKLAR

- [1]. 2011, *NFC Forum*, <http://www.nfcforum.org/>, [Ziyaret Tarihi: 1 Ocak 2012].
- [2]. 2012, *Near Field Communication*, https://en.wikipedia.org/wiki/Near_field_communication, [Ziyaret Tarihi: 1 Ocak 2012].
- [3]. Madlmayr, G., Langer, J., Kantner, C. and Scharinger, J., 2008, *NFC Devices: Security and Privacy, Proceedings of the 3rd International Conference on Availability, Reliability and Security*, 4-7 Mart 2008 Barcelona İspanya, ISBN: 978-0-7695-3102-1, 642-647.
- [4]. 2009, *Orange*, http://www.orange.com/en_EN/innovation/latest_news/hot_topics/tous_les_dossiers/att0009564/ddm_200512uk.pdf, [Ziyaret Tarihi: 10 Ocak 2012].
- [5]. Falke, O., 2007, Technical Report, *Mobile Services for Near Field Communications*, ISSN 1862-5207.
- [6]. Cheng, H. C., 2009, A Generic Model for NFC-based Mobile Commerce in Advanced Communication Technology, *ICACT 2009 11th International Conference*, 2009 Phoenix Park, 1-2.
- [7]. 2012, *Gemalto*, http://www.gemalto.com/nfc/mobile_payment_ecosystem.html, [Ziyaret Tarihi: 10 Ocak 2012].
- [8]. 2012, *Blackberry Enterprise Server Architecture*, http://docs.blackberry.com/en/admin/deliverables/7335/BlackBerry_Enterprise_Server_architecture_305806_11.jsp, [Ziyaret Tarihi: 15 Ocak 2012].
- [9]. 2012, *Android (operating system)*, [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)), [Ziyaret Tarihi: 1 Şubat 2012].
- [10]. 2012, *Dalvik (software)*, [http://en.wikipedia.org/wiki/Dalvik_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software)), [Ziyaret Tarihi: 1 Şubat 2012].
- [11]. 2011, *Introduction to Android*, <http://www.slideshare.net/deepakshare/android-arch-presentation>, [Ziyaret Tarihi: 1 Şubat 2012].
- [12]. 2012, *Linux Kernel*, http://en.wikipedia.org/wiki/Linux_kernel, [Ziyaret Tarihi: 1 Şubat 2012].
- [13]. 2012, *Processes and Threads*, <http://developer.android.com/guide/components/processes-and-threads.html>, [Ziyaret Tarihi: 1 Mart 2012].
- [14]. Benyó, B., Vilmos, A., Förd_s, G., Sódor, B. and Kovács, K., 2009, The StolPan View of the NFC Ecosystem, *Proceedings of the Conference on Wireless Telecommunications Symposium*, 22-24 Nisan 2009 Prag Çek Cumhuriyeti, ISBN: 978-1-4244-2588-4, 1-5.

- [15]. Benyó, B., Vilmos, A., Kovacs, K. and Kutor, L., 2007, NFC Applications and Business Model of the Ecosystem, *Proceedings of the 16th IST Mobile and Wireless Communications Summit*, 1-5 Temmuz 2007 Budapeşte Macaristan, ISBN: 963-8111-66-6, 1-5.
- [16]. Madlmayr, G., Langer, J. and Scharinger, J., 2008, Managing an NFC Ecosystem, *Proceedings of the 7th International Conference on Mobile Business*, 7-8 Temmuz 2008 Barcelona İspanya, ISBN: 978-0-7695-3260-8, 95-101.
- [17]. Benyó, B., 2009, Business Process Analysis of NFC-based Services, *Proceedings of the IEEE 7th International Conference on Computational Cybernetics*, 26-29 Kasım 2009 Mallorca İspanya, ISBN: 9781424453108, 75-79.
- [18]. Madlmayr, G., Langer, J., Kantner, C., Scharinger, J. and Schaumüller-Bichl, I., 2009, Risk Analysis of Over-the-Air Transactions in an NFC Ecosystem, *Proceedings of the 1st International Workshop on Near Field Communication*, 24-26 Şubat 2009 Hagenberg Avusturya, ISBN: 978-0-7695-3577-7, 87-92.
- [19]. Reveilhac, M. and Pasquet, M., 2009, Promising Secure Element Alternatives for NFC Technology, *Proceedings of the 1st International Workshop on Near Field Communication*, 24-26 Şubat 2009 Hagenberg Avusturya, ISBN: 978-0-7695-3577-7, 75-80.
- [20]. 2008, *Dynamic management of multi-application secure elements, StoLPan*, http://www.nfc-forum.org/resources/white_papers/Stolpan_White_Paper_08.pdf, [Ziyaret Tarihi: 1 Nisan 2012].
- [21]. Ruiz, I. L., Gomez-Nieto, M. A., 2008, University Smart Poster: Study of NFC Technology Applications for University Ambient, *Proc. 3rd Symposium of Ubiquitous Computing and Ambient Intelligence*, 2008 Salamanca İspanya, Springer, 112-116.
- [22]. Miraz, G. M., Ruiz, I. L. and Gomez-Nieto, M. A., 2009, University of Things: Applications of Near Field Communication Technology in University Environments, *The Journal of Eworking*, 2009, 52-64.
- [23]. Ozdenizci, B., Coskun, V., Aydin, M. N. and Ok, K., 2010, NFC Loyal: A Beneficial Model to Promote Loyalty on Smart Cards of Mobile Devices, *International Conf. for Internet Technology and Secured Transactions (ICITST-2010)*, 8-11 Kasım 2010 Londra İngiltere, 134-139.
- [24]. Ghiron, S. L., Sposato, S. and Medaglia, C. M., 2009, NFC Ticketing: A Prototype and Usability Test of an NFC-Based Virtual Ticketing Application, *Proc. 1st International Workshop on Near Field Communication*, 2009 Hagenberg Avusturya, IEEE, 45-50.

- [25]. Morak, J., Schwetz, V., Hayn, D., Fruhwald, F. and Schreier, G., 2008, Electronic Data Capture Platform for Clinical Research based on Mobile Phones and Near Field Communication Technology, *Proc. 30th Annual International Conf. of the IEEE Engineering in Medicine and Biology Society*, 2008 Vancouver Kanada, IEEE, 5334-5337.
- [26]. Cappiello, I., Puglia, S. and Vitaletti, A., 2009, Design and Initial Evaluation of a Ubiquitous Touch-Based Remote Grocery Shopping Process, *Proc. 1st International Workshop on Near Field Communication*, 2009 Hagenberg Avusturya, IEEE, 9-14.
- [27]. Siira, E. and Haikio, J., 2007, Experiences from Near-field Communication (NFC) in a Meal Service System, *Proc. 1st RFID Eurasia Conf.*, 2007 Istanbul Türkiye, IEEE, 273-278.
- [28]. Pasquet, M., Reynaud, J., Rosenberger, C., 2008, Secure Payment with NFC Mobile Phone in the SmartTouch Project, *Proc. International Symposium on Collaborative Technologies and Systems*, 2008 California A.B.D., IEEE, 121-126.
- [29]. Massoth, M. and Bingel, T., 2009, Performance of different mobile payment service concepts compared with a NFC-Based Solution, *Proc. 4th International Conf. on Internet and Web Applications and Services*, 2009 Venice İtalya, IEEE, 205-210.
- [30]. 2011, *Near Field Communication and the NFC Forum: The Keys to Truly Interoperable Communications*, http://www.nfc-forum.org/resources/white_papers/nfc_forum_marketing_white_paper.pdf, [Ziyaret Tarihi: 20 Nisan 2012].
- [31]. 2005, *Near Field Communication- White Paper Ecma/TC32-TG19/2005/012*, <http://www.ecmainternational.org>, [Ziyaret Tarihi: 1 Mayıs 2012].
- [32]. Ondrus, J. and Pigneur, Y., 2009, Near Field Communication: An Assessment for Future Payment Systems, *Information Systems and E-Business Management*, 2009, 347-361.
- [33]. Warakagoda, N., 2012, *Near Field Communication Opportunities & Standards*, http://www.umts.no/files/081028_nfc_standards_payments_Narada.pdf, [Ziyaret Tarihi: 10 Haziran 2012].
- [34]. 2011, *Java Card Technology Overview*, <http://www.oracle.com/technetwork/java/javame/javacard/overview/about/index.html>, [Ziyaret Tarihi: 10 Haziran 2012].
- [35]. 2012, *Gemalto Java Card Platform Overview*, http://www.gemalto.com/simagine/2009/download/java_card_platform.pdf, [Ziyaret Tarihi: 10 Haziran 2012].
- [36]. 2012, *Giesecke & Devrient*, <http://www.gi-de.com/en/index.jsp>, [Ziyaret Tarihi: 1 Mayıs 2012].

EKLER

EK 1. Güvenli Arabirim Kaynak Kodları

Wallet.java

package com.alcatel.se;

import javacard.framework.ISOException;

import javacard.framework.OwnerPIN;

import javacard.framework.Util;

/**

* Bu sınıf WalletInterface arayüzünü gerçekleyerek, güvenli arabirim
* operasyonlarını sağlamaktadır.

*/

public class Wallet **implements** WalletInterface {
 static byte initialPin[] = { 0x30, 0x30, 0x30, 0x30 };

static short paymentLimitWithoutPin = 30;

public boolean authenticated = **false**;

private byte[] participantId = {};

private OwnerPIN walletPin;

/**

* Varsayılan sınıf kurucusudur, cüzdan ile birlikte varsayılan pin atanır.

*/

public Wallet() {
 walletPin = **new** OwnerPIN((**byte**) 3, (**byte**) 4);
 walletPin.update(initialPin, (**short**) 0, (**byte**) 4);
 participantId = **new byte**[]{};

 }

/**

* Kullanıcı tarafından girilen PIN kontrol edilir, eğer doğru ise oturum başlar.

* PIN hatalı ise hata mesajı gönderir.

* *@param buffer*

* kullanıcıdan gelen veri

* *@param offset*

* tampon bellekteki ofset

* *@param length*

* tampon bellekteki veri uzunluğu

*

*/

public void checkPin(**byte**[] buffer, **short** offset, **byte** length) {

if (walletPin.check(buffer, offset, length)) {
 authenticated = **true**;

return;

 }

 ISOException.throwIt(SW_PIN_FAILED);

 }

```

/**
 * Girilen para ile ödeme yapar.
 * @param paymentAmount
 *     ödenecek tutar
 *
 */
public byte[] makePayment(short paymentAmount) {

    if (paymentAmount == 0)
        ISOException.throwIt(SW_PAYMENT_NOT_VALID);

    if (paymentAmount <= paymentLimitWithoutPin) {
        return this.participantId;
    } else {
        if (authenticated) {
            return this.participantId;
        } else {
            ISOException.throwIt(SW_AUTHENTICATION_REQUIRED);
        }
    }
    return new byte[] {0x00};
    // ISOException.throwIt(SW_INSTRUCTION_NOT_SUPPORTED);
}
/**
 * Güvenli arabirim üzerinde kullanıcı bilgisini günceller.
 * @param id
 *     Güncellenecek kullanıcı bilgisi
 *
 */
public void updateID(byte[] id) {
    if (authenticated) {
        participantId = new byte[(short) id.length];
        Util.arrayCopy(id, (short) 0, participantId, (short) 0,
            (short) id.length);
        return;
    }
    ISOException.throwIt(SW_AUTHENTICATION_REQUIRED);
}
/**
 * Güvenli arabirim üzerindeki PIN'i günceller
 * @param id
 *     Güncellenecek PIN
 *
 */
public void updatePIN(byte[] pin) {
    if (authenticated) {
        walletPin.update(pin, (short) 0, (byte) 4);
        return;
    }
}

```

```

        ISOException.throwIt(SW_AUTHENTICATION_REQUIRED);
    }

    /**
     * Güvenli arabirimden kullanıcı kimliğini gönderir.
     * @return byte[]
     *      Gönderilen kimlik
     */
    public byte[] getParticipantId() {
        if (!authenticated) {
            ISOException.throwIt(SW_AUTHENTICATION_REQUIRED);
        }
        return this.participantId;
    }
}

```

WalletApplet.java

```

package com.alcatel.se;

```

```

import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.ISO7816;
import javacard.framework.ISOException;
import javacard.framework.Util;

```

```

/**
 * Bu sınıf JavaCard Applet'i miras alarak güvenli arabirim işlemlerini sunar.
 */
public class WalletApplet extends Applet implements WalletInterface {

```

```

    private Wallet wallet;

```

```

    /**
     * JCRE'ye uygulamayı kayıt eden kurucu metoddur.
     * Install fonksiyonu ile uygulama kurulur.
     * @param bArray
     *      kurulum parametrelerini tutan dizi
     * @param bOffset
     *      bArray içinde başlangıç offseti
     * @param bLength
     *      bArray içinde gelen verinin uzunluğu
     */

```

```

    public WalletApplet(byte[] bArray, short bOffset, byte bLength) {
        wallet = new Wallet();
        register(bArray, (short) (bOffset + 1), bArray[bOffset]);
    }

```

```

    /**

```

```

* JavaCard ortamı bu metodu ilk olarak çağırılmaktadır.
* @param bArray
*     kurulum parametreleri
* @param bOffset
*     dizi içindeki başlangıç ofseti
* @param bLength
*     dizi içindeki parametrelerin uzunluğu
* @throws ISOException
*     hata olursa ISOException oluşur
* @see javacard.framework.Applet#install(byte[], short, byte)
*/
public static void install(byte[] bArray, short bOffset, byte bLength) {
    new WalletApplet(bArray, bOffset, bLength);
}

/*
* APDU mesajı içinde gelen parametreleri alır ve buna göre işlem kararı verir.
* @param APDU
*     alınan APDU mesajı
*/
public void process(APDU apdu) throws ISOException {

    if (selectingApplet()) {
        wallet.authenticated = false;
        return;
    }

    byte[] buffer = apdu.getBuffer();

    if ((buffer[ISO7816.OFFSET_CLA] == 0)
        && (buffer[ISO7816.OFFSET_INS] == (byte) (0xA4)))
        return;

    switch (buffer[ISO7816.OFFSET_INS]) {
    case VERIFY: {
        receive(apdu);
        wallet.checkPin(buffer, (short) ISO7816.OFFSET_CDATA,
            buffer[ISO7816.OFFSET_LC]);
    }
    break;

    case UPDATE_PIN: {
        receive(apdu);
        byte[] pin = { 0x00, 0x00, 0x00, 0x00 };
        Util.arrayCopy(buffer, ISO7816.OFFSET_CDATA, pin, (short) 0,
            (short) 4);
        wallet.updatePIN(pin);
    }
}

```

```

    break;

case MAKE_PAYMENT: {
    receive(apdu);
    short paymentAmount = (short) Util.getShort(buffer,
        ISO7816.OFFSET_CDATA);
    byte[] id = wallet.makePayment(paymentAmount);

    short len = (short) id.length;
    Util.arrayCopy(id, (short) 0, buffer, (short) 0, (short) len);
    apdu.setOutgoing();
    apdu.setOutgoingLength((short) len);
    apdu.sendBytesLong(buffer, (short) 0, len);
}
    break;

case UPDATE_ID: {
    receive(apdu);
    short len = buffer[ISO7816.OFFSET_LC];
    byte[] id = new byte[(short) len];
    Util.arrayFillNonAtomic(id, (short) 0, len, (byte) 0x00);
    Util.arrayCopy(buffer, ISO7816.OFFSET_CDATA, id, (short) 0,
        (short) len);

    wallet.updateID(id);
    Util.arrayCopy(wallet.getParticipantId(), (short) 0, buffer,
        (short) 0, (short) buffer[ISO7816.OFFSET_LC]);

    apdu.setOutgoing();
    apdu.setOutgoingLength(len);
    apdu.sendBytesLong(wallet.getParticipantId(), (short) 0, len);
}
    break;

case GET_ID: {
    receive(apdu);
    short len = (short) wallet.getParticipantId().length;
    Util.arrayCopy(wallet.getParticipantId(), (short) 0, buffer,
        (short) 0, (short) len);
    apdu.setOutgoing();
    apdu.setOutgoingLength((short) len);
    apdu.sendBytesLong(buffer, (short) 0, len);
}
    break;

default:
    //ISOException.throwIt(SW_INSTRUCTION_NOT_SUPPORTED);

```

```

        receive(apdu);
        apdu.setOutgoing();
        apdu.setOutgoingLength((short) 10);
        apdu.sendBytesLong(buffer, (short) 0, (short) 10);

        break;
    }
}

/**
 * APDU receive operation takes place
 * @param APDU
 *     APDU to receive
 */
public short receive(APDU apdu) throws ISOException {
    byte[] buffer = apdu.getBuffer();

    short readCount = (short) (buffer[ISO7816.OFFSET_LC] & 0x00FF);

    if (readCount != apdu.setIncomingAndReceive())
        ISOException.throwIt(ISO7816.SW_WRONG_LENGTH);

    return readCount;
}
}

```

WalletInterface.java

```

package com.alcatel.se;

/**
 * Bu arabirim, güvenli arabirimde kullanılacak kodları tutmaktadır.
 */
public interface WalletInterface {
    final static byte ALU_CLA_1 = (byte) 0x80;
    final static byte ALU_CLA_2 = (byte) 0x81;
    final static byte ISO_CLA = (byte) 0x00;
    final static byte VERIFY = (byte) 0x20;
    final static byte UPDATE_PIN = (byte) 0x22;
    final static byte MAKE_PAYMENT = (byte) 0x40;
    final static byte UPDATE_ID = (byte) 0x50;
    final static byte GET_ID = (byte) 0x55;
    final static short SW_PIN_FAILED = (short) 0x63C0;
    final static short SW_INSTRUCTION_NOT_SUPPORTED = (short) 0x6D00;
    final static short SW_PAYMENT_NOT_VALID = (short) 0x6E01;
    final static short SW_AUTHENTICATION_REQUIRED = (short) 0x6E02;
    final static short SW_INSTRUCTION_NOT_ALU = (short) 0x6D01; }

```

EK 2. Güvenli Arabirim Test Uygulaması Kaynak Kodları

MyApp.java

```

package com.alcatel.sim;

import net.rim.device.api.ui.UiApplication;

/*Bu sınıf UiApplication sınıfını miras alarak grafik arayüz işlemleri sunmaktadır.
*/
public class MyApp extends UiApplication
{
    /**
     * Uygulama başlangıç noktası
     * @param args Komut satırı parametreleri
     */
    public static void main(String[] args)
    {
        MyApp theApp = new MyApp();
        theApp.enterEventDispatcher();
    }

    public MyApp()
    {
        pushScreen(new MyScreen());
    }
}

```

MyScreen.java

```

package com.alcatel.sim;

import javax.microedition.apdu.APDUConnection;
import javax.microedition.io.Connector;
import net.rim.device.api.ui.Field;
import net.rim.device.api.ui.FieldChangeListener;
import net.rim.device.api.ui.Font;
import net.rim.device.api.ui.MenuItem;
import net.rim.device.api.ui.component.BasicEditField;
import net.rim.device.api.ui.component.ButtonField;
import net.rim.device.api.ui.component.Menu;
import net.rim.device.api.ui.component.TextField;
import net.rim.device.api.ui.container.MainScreen;
import net.rim.device.api.util.ByteArrayUtilities;
import net.rim.device.api.util.StringProvider;

/**Blackberry'nin standart arayüzlerini sunan bir ekrandır.
*/
public final class MyScreen extends MainScreen {

```

```

private BasicEditField basicEditField, commandEditField;
private TextField textField;
private APDUConnection connection;

/** Yeni bir SETestScreen oluřturur.
 */
public MyScreen() {
    super(MainScreen.VERTICAL_SCROLL | MainScreen.VERTICAL_SCROLLBAR);
    setTitle("Sim Test");

    Font font = Font.getDefault().derive(Font.ANTIALIAS_DEFAULT, 24);

    basicEditField = new BasicEditField("AID:",
"apdu:0;target=A0.00.00.00.90.00.00.01.02.03.04.05", 100, BasicEditField.EDITABLE);
    basicEditField.setFont(font);
    add(basicEditField);

    commandEditField = new BasicEditField("Command:", "", 100,
BasicEditField.EDITABLE);
    commandEditField.setFont(font);
    add(commandEditField);

    ButtonField buttonFieldSendAPDU = new ButtonField("Send APDU",
ButtonField.CONSUME_CLICK | ButtonField.FIELD_RIGHT);
    add(buttonFieldSendAPDU);
    buttonFieldSendAPDU.setChangeListener(new FieldChangeListener() {
        public void fieldChanged(Field arg0, int arg1) {
            sendData(null);
        }
    });

    textField = new TextField();
    textField.setFont(font);
    add(textField);
}

protected void makeMenu(Menu menu, int instance) {
    super.makeMenu(menu, instance);

    MenuItem mCardEmulation = new CommandCardEMenuItem();
    menu.add(mCardEmulation);

    MenuItem mAppletSelect = new CommandSelectAppletMenuItem();
    menu.add(mAppletSelect);

    MenuItem m1 = new Command1MenuItem();
    menu.add(m1);
}

```



```
MenuItem m2 = new Command2MenuItem();
menu.add(m2);
```

```
MenuItem m3 = new Command3MenuItem();
menu.add(m3);
```

```
MenuItem m4 = new Command4MenuItem();
menu.add(m4);
```

```
MenuItem m5 = new Command5MenuItem();
menu.add(m5);
```

```
}
```

```
private class CommandCardEMenuItem extends MenuItem {
    public CommandCardEMenuItem() {
        super(new StringProvider("Card Emulation"), 0, 0);
    }
    public void run() {
        setCardEmulation();
    }
}
```

```
private class CommandSelectAppletMenuItem extends MenuItem {
    public CommandSelectAppletMenuItem() {
        super(new StringProvider("Select Applet"), 0, 0);
    }

    public void run() {
        selectApplet();
    }
}
```

```
private class Command1MenuItem extends MenuItem {
    public Command1MenuItem() {
        super(new StringProvider("Command Check Pin"), 0, 0);
    }
    public void run() {
        byte[] command = { (byte) 0x80, (byte) 0x20, (byte) 0x00, (byte) 0x00, (byte) 0x04,
(byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x00 };
        sendData(command);
    }
}
```

```
private class Command2MenuItem extends MenuItem {
    public Command2MenuItem() {
        super(new StringProvider("Command Make Payment"), 0, 0);
    }
}
```

```

    public void run() {
        byte[] command = { (byte) 0x80, (byte) 0x40, (byte) 0x00, (byte) 0x00, (byte) 0x02,
        (byte) 0x00, (byte) 0x22 };
        sendData(command);
    }
}

```

```

private class Command3MenuItem extends MenuItem {
    public Command3MenuItem() {
        super(new StringProvider("Command Update ID"), 0, 0);
    }
}

```

```

    public void run() {
        byte[] command = { (byte) 0x80, (byte) 0x50, (byte) 0x00, (byte) 0x00, (byte) 0x04,
        (byte) 0x10, (byte) 0x10, (byte) 0x10, (byte) 0x10 };
        sendData(command);
    }
}

```

```

private class Command4MenuItem extends MenuItem {
    public Command4MenuItem() {
        super(new StringProvider("Command Get ID"), 0, 0);
    }
}

```

```

    public void run() {
        byte[] command = { (byte) 0x80, (byte) 0x55, (byte) 0x00, (byte) 0x00 };
        sendData(command);
    }
}

```

```

private class Command5MenuItem extends MenuItem {
    public Command5MenuItem() {
        super(new StringProvider("Command Check Pin With Error"), 0, 0);
    }
}

```

```

    public void run() {
        byte[] command = { (byte) 0x80, (byte) 0x20, (byte) 0x00, (byte) 0x00, (byte) 0x04,
        (byte) 0x00, (byte) 0x00, (byte) 0x00, (byte) 0x10 };
        sendData(command);
    }
}

```

```

private void sendData(byte[] command) {

```

```

    try {
        byte[] result = null;
        if (command == null) {
            result

```

=

```

connection.exchangeAPDU(ByteArrayUtilities.hexToByteArray(commandEditField.getText(
)));
    log("S: " + ByteArrayUtilities.hexToByteArray(commandEditField.getText()));
}
else {
    result = connection.exchangeAPDU(command);
    log("Send: " + ByteArrayUtilities.byteArrayToHex(command));
}
if (result != null)
    log("Result: " + ByteArrayUtilities.byteArrayToHex(result));
else
    log("Result is Null");
} catch (Exception e) {
    log(e.getMessage());
    e.printStackTrace();
}
}

private void setCardEmulation() {
    try {
} catch (Exception e) {
    log(e.getMessage());
    e.printStackTrace();
}
}

private void selectApplet() {
    try {
        log("Trying to select applet");
        String uri = basicEditField.getText();
        log("Applet URI: " + uri);
        connection = (APDUConnection) Connector.open(uri);
        log("Applet selected.");
    } catch (Exception e) {
        log(e.getMessage());
        e.printStackTrace();
    }
}

private void log(String l) {
    String log = textField.getText();
    log += "\n";
    log += l;

    textField.setText(log);
}
}

```

EK 3. Blackberry Mobil Cüzdan Uygulaması Kod Dökümantasyonu

package com.alcatel.mobilewallet.entity;

/**

* Veri tutan sınıflar için bir temel sınıftır.

* Genel işlemleri sağlamaktadır.

*/

public class BaseEntity;

/**

* Account sınıf hesap bilgisini tutmaktadır.

*/

public class Account **extends** BaseEntity;

/**

*Bakiye bilgisini tutan sınıftır.

*/

public class Balance;

/**

* Field alanı basit verileri tutmaktadır.

*/

public class Field **extends** BaseEntity;

/**

* Organizasyon bilgisini tutmaktadır.

*/

public class Organization **extends** BaseEntity;

/**

* Mobil Cüzdan Hizmetini alan kullanıcı bilgileri tutulmaktadır.

*/

public class Participant **extends** BaseEntity;

/**

* Parti bilgisini tutan sınıftır.

*/

public class Party **extends** BaseEntity;

/**

* Mobil Cüzdan Hizmetinde tanımlı ödeme metodlarını tutan sınıftır.

*/

public class PaymentMethod;

/**

* Program bilgisini tutmaktadır.

*/

public class Program **extends** BaseEntity;

/**

* Etiketler için temel sınıftır.

*/

public class Tag **extends** BaseEntity;

/**

* Mobil Cüzdan Hizmetlerinin işlem geçmişi için kullanılmaktadır.

*/

public class Transaction **extends** BaseEntity;

/**

* Temel cüzdan sınıfıdır.

*/

public class Wallet **extends** BaseEntity;

/**

* Wallet sınıfını miras alan bir sınıftır.

* Kullanıcı bilgisini ve gerekli bilgileri tutmaktadır.

*/

public class MWSWallet **extends** Wallet;

/**

* Cüzdan işlemleri için kullanılan bir sınıftır.

*/

public class WalletManager;

package com.alcatel.mobilewallet.lib.http;

/**

* Alcatel-Lucent Mobil Cüzdan hizmetleri ile iletişimi sağlayan bir sınıftır.

*/

public class MWSRestClient

/**

* MWSRestClient statik nesnesini gönderir.

* *@return Statik nesne*

*/

public static MWSRestClient getInstance();

/**

* Sınıf kurucusudur, Restful servisler için gerekli olan parametreleri alır.

* *@param key Restful servis anahtarı*

* *@param secret Restful servis gizli anahtarı*

*/

private MWSRestClient(String key, String secret);

/**

* Varsayılan sınıf kurucusudur.

*/

private MWSRestClient() ;

/**

* Servisin adresini gönderir

* *@param relativeUrl Servis adresi*

* *@return*

*/

private static String getAbsoluteUrl(String relativeUrl);

/**

* Restful servis bağlantısını sağlar.

* *@return Giriş yapabilmek için gerekli adresi dönderir.*

*/

public String getAuthorizeUrl();

/**

* Giriş tokenini gönderir.

* *@return*

*/

```

public AccessToken getAccessToken();

/**
 * Verilen parametrelerle sunucuya Get isteği yapar.
 * @param url İstek yapılacak adres
 * @param params İstek parametreleri
 * @param callback İstem bitince dönüş yapıcak metod
 */
public void get(final String url, final Hashtable params, final MWSRestClientListener
callback);
/**
 * Verilen parametrelerle sunucuya Put isteği yapar.
 * @param url İstek yapılacak adres
 * @param params İstek parametreleri
 * @param callback İstem bitince dönüş yapıcak metod
 */
public void put(final String url, final Hashtable params, final MWSRestClientListener
callback);
/**
 * Verilen parametrelerle sunucuya Post isteği yapar.
 * @param url İstek yapılacak adres
 * @param params İstek parametreleri
 * @param callback İstem bitince dönüş yapıcak metod
 */
public void post(final String url, final Hashtable params, final MWSRestClientListener
callback);
/**
 * Restful client listener that is handled for requests.
 * If request is successful requestSuccess is called.
 * Else requestFailed is called.
 * Restful istem dinleyicisidir. Eğer istem başarılı ise requestSuccess çağrılır
 * başarısız ise requestFailed çağrılır.
 */
public interface MWSRestClientListener;

package com.alcatel.mws.lib.parser;
/**
 * Restful servisleri parse eden sınıf.
 */
public class Parser {
/**
 * Parser sınıfı için tek bir nesne sağlar.
 * @return Parser sınıfı nesnesi gönderir.
 */
public static synchronized Parser getInstance();
/**
 * Parser sınıfı başlatılır.
 * @param xml Parse edilecek XML

```

```

    * @return Document türünde veri gönderilir.
    */
    private static Document initializeParser(String xml);
    /**
    * Tüm işlem geçmişini parse eder.
    * @param xml İşlem verisi
    * @return Vector
    */
    public Vector parseTransactionXML(String xml);
    /**
    * Tek bir işlemi parse eder.
    * @param element
    * @return İşlem element.
    */
    private Transaction parseTransaction(Element element);
    /**
    * Tüm Organizasyonları parser eder.
    * @param xml Organizasyon verisi
    * @return Vector
    */
    public Vector parseOrganizationXML(String xml);
    /**
    * Tek bir organizasyonu parse eder.
    * @param element
    * @return Organizasyon element.
    */
    private Organization parseOrganization(Element element);
    /**
    * Tüm hesap bilgisini parse eder.
    * @param xml Hesap bilgisi.
    * @return Vector
    */
    public Vector parseAccountXML(String xml);
    /**
    * Bir hesap bilgisini parse eder.
    * @param element
    * @return Hesap bilgisi gönderir.
    */
    private Account parseAccount(Element element);
    /**
    * Tüm program bilgisini parse eder.
    * @param xml XML verisi
    * @return Vector
    */
    public Vector parseProgramXML(String xml);
    /**
    * Sadece bir program parse eder.
    * @param element

```

```

    * @return Program
    */
    public Program parseProgram(Element subElement);
    /**
    * Bir düğümdeki profilleri parse eder.
    * @param element
    * @return Vector
    */
    private Vector parseProfile(Element element);
    /**
    * Kullanıcı verisini parse eder.
    * @param xml XML veri
    * @return Vector
    */
    public Vector parseParticipantXML(String xml);
    /**
    * Bir düğümdeki profil bilgisini parse eder.
    * @param element
    * @return Profil bilgisi
    */
    private Participant parseParticipant(Element element);
    /**
    * Uygulamada kullanılan yardımcı sınıftır.
    */
    public class Util;
    /**
    * Serviste kullanılacak XML üreten sınıftır.
    */
    public class XMLGenerator
    /**
    * Para transferi için XML üretir.
    * @param fromProfile Kimden bilgisi
    * @param toProfile Kime bilgisi
    * @param programCode Program kodu
    * @param currencyCode Para kur bilgisi
    * @param value Tutar
    * @param description Transfer açıklaması
    * @return
    */
    public String storedValueTransferXML(String fromProfile, String toProfile, String
    programCode, String currencyCode, int value, String description);
    /**
    * Profil güncellemesi için gerekli XML üretir.
    * @param userName Kullanıcı bilgisi
    * @param displayName Gözüken isim
    * @param phoneNumber Telefon bilgisi
    * @param email E-mail bilgisi
    * @param items Etiketler

```



```

    * @return
    */
    public String profileUpdateXML(String userName, String displayName, String
phoneNumber, String email, Hashtable items);

    /**
     * Topup için XML oluşturur.
     * @param Hesap bilgisi
     * @param Tutar
     * @return
     */
    public String topupXml(Account a, int value);

package com.alcatel.mws.nfc.se;
    /**
     * Güvenli arabirim ile ilgili iletişim metodlarını ve yöntemlerini tanımlayan sınıftır.
     */
    public class SecureElementConnection
    /**
     * Applet'e bağlanmak için kullanılan adrestir.
     */
    private static String connectionStr =
"apdu:0;target=A0.00.00.00.90.00.00.01.02.03.04.05";
    /**
     * Güvenli arabirim bağlantısı gönderir.
     * @return SecureElementConnection
     */
    public SecureElementConnection getInstance();
    /**
     * Applet'e bağlayan metod.
     * @return true başarılı ise dönen değer
     */
    public boolean connectToSecureElement();
    /**
     * Güvenli arabirime byte veri gönderir.
     * @param cmd Gönderilecek veriler
     * @return byte[] Güvenli arabirim sonucu
     */
    public byte[] sendData(byte[] cmd);
    /**
     * PIN kontrolü yapar.
     * @param pin Pin
     * @return byte[] Güvenli arabirim sonucu
     */
    public byte[] checkPin(String pin) ;
    /**
     * Güvenli arabirim üzerinden ödeme yapar.
     * @param total Ödeme tutarı

```

```

    * @return byte[] Güvenli arabirim sonucu
    */
    public byte[] makePayment(String total);
    /**
     * Güvenli arabirimden kullanıcı bilgisini alır.
     * @return byte[] Güvenli arabirim sonucu
     */
    public byte[] getID();
    /**
     * Güvenli arabirim üzerindeki bilgileri günceller.
     * @param id Kullanıcı bilgileri
     * @return byte[] Güvenli arabirim sonucu
     */
    public byte[] updateID(String id);
    /**
     * PIN güncellemesi yapar.
     * @param pin Güncellenecek PIN
     * @return byte[] Güvenli arabirim sonucu
     */
    public byte[] updatePIN(String pin);
    /**
     * Güvenli arabirim bağlantısını kapatır.
     */
    public void close();

package com.alcatel.mws.ui;

/**
 * UiApplication sınıfından miras alarak arayüz sağlar.
 */
public class MobileWallet extends UiApplication;
/**
 * Tüm arayüz sınıfları için temel sınıftır.
 */
public class BaseScreen extends MainScreen;
/**
 * Hesap detaylarının gözüktüğü arayüzdür.
 */
public class MWSAccountDetailScreen extends BaseScreen;
/**
 * Hesap listeleyen arayüzdür.
 */
public class MWSAccountListScreen extends BaseScreen;
/**
 * Kullanıcı giriş arayüzüdür.
 */
public class MWSLoginScreen extends MainScreen;

```

```

* Sadakat ve kupon kartlarını listeleyen arayüzdür.
*/
public class MWSLoyaltyListScreen extends BaseScreen;
/**
* Para kartlarını listeleyen arayüzdür.
*/
public class MWSMoneyListScreen extends BaseScreen;
/**
* NFC işlemleri arayüz sınıfıdır.
*/
public class MWSNfcScreen extends BaseScreen;
/**
* Kullanıcı ayarları ile ilgili arayüzdür.
*/
public class MWSSettingsScreen extends BaseScreen;
/**
* Topup ile para yükleme arayüzüdür.
*/
public class MWSTopupScreen extends BaseScreen;
/**
* Hesap geçmişini listeleyen arayüz sınıfıdır.
*/
public class MWSTransactionListScreen extends BaseScreen

```

EK 4. Android Mobil Cüzdan Uygulaması Kod Dökümantasyonu

package com.alcatel.mobilewallet.entity;

/**

- * Veri tutan sınıflar için bir temel sınıftır.
- * Genel işlemleri sağlamaktadır.

*/

public class BaseEntity **implements** Serializable

/**

- * Account sınıf hesap bilgisini tutmaktadır.

*/

public class Account **extends** BaseEntity;

/**

- *Bakiye bilgisini tutan sınıftır.

*/

public class Balance;

/**

- * Field alanı basit verileri tutmaktadır.

*/

public class Field **extends** BaseEntity;

/**

- * Organizasyon bilgisini tutmaktadır.

*/

public class Organization **extends** BaseEntity;

/**

- * Mobil Cüzdan Hizmetini alan kullanıcı bilgileri tutulmaktadır.

*/

public class Participant **extends** BaseEntity;

/**

- * Parti bilgisini tutan sınıftır.

*/

public class Party **extends** BaseEntity;

/**

- * Mobil Cüzdan Hizmetinde tanımlı ödeme metodlarını tutan sınıftır.

*/

public class PaymentMethod;

/**

- * Program bilgisini tutmaktadır.

*/

public class Program **extends** BaseEntity;

/**

- * Etiketler için temel sınıftır.

*/

public class Tag **extends** BaseEntity;

/**

- * Mobil Cüzdan Hizmetlerinin işlem geçmişi için kullanılmaktadır.

*/

public class Transaction **extends** BaseEntity;

/**

```

* Temel cüzdan sınıfıdır.
*/
public class Wallet extends BaseEntity;
/**
* Wallet sınıfını miras alan bir sınıftır.
* Kullanıcı bilgisini ve gerekli bilgileri tutmaktadır.
*/
public class MWSWallet extends Wallet;
/**
* Cüzdan işlemleri için kullanılan bir sınıftır.
*/
public class WalletManager;
/**
* Restful servisler için bilgi tutar.
*/
public class Config;

package com.alcatel.mobilewallet.activity;

/**
* NFC işlemleri için kullanılan arayüz sınıfıdır.
*/
public class Beam extends Activity;
/**
* Liste türleri için temel arayüz sınıfıdır.
*/
public class MWSBaseListView extends ListActivity;
/**
* Hesap bilgileri için arayüz sınıfıdır.
*/
public class MWSAccountList extends MWSBaseListView;
/**
* Bakiye görüntüleme için sınıftır.
*/
public class MWSBalanceView extends MWSBaseListView;
/**
* Para transfer uyarısı için arayüz sınıfıdır.
*/
public class MWSConfirmMoneyRequest extends Activity;
/**
* İşlem geçmişi için veri sınıfıdır.
*/
public class MWSHistoryAdapter extends ArrayAdapter<Transaction>;
/**
* Hesap geçmişi arayüz sınıfıdır.
*/
public class MWSHistoryList extends MWSBaseListView;

```

```

/**
 * Giriş yapmak için arayüz sınıfıdır.
 */
public class MWSLoginView extends Activity;
/**
 * Sadakat ve kupon kartlar için arayüz sınıfıdır.
 */
public class MWSLoyaltyList extends ListActivity;
/**
 * NFC için arayüz sınıfıdır.
 */
public class MWSNFCSend extends Activity;
/**
 * Para transferi için arayüz sınıfıdır.
 */
public class MWSPeerToPeerView extends Activity;
/**
 * Profil bilgileri için veri sınıfıdır.
 */
public class MWSProfileAdapter extends ArrayAdapter<HashMap<String, String>>;
/**
 * Profil için arayüz sınıfıdır.
 */
public class MWSProfileList extends MWSBaseListView;
/**
 * Genel veri sınıfıdır.
 */
public class SeparatedListAdapter extends BaseAdapter;
/**
 * Mobil cüzdan için veri sınıfıdır.
 */
public class WalletAdapter extends ArrayAdapter<Wallet>;

package com.alcatel.mobilewallet.lib.http;

/**
 * Alcatel-Lucent Mobil Cüzdan Hizmetlerinde iletişimi sağlayacak olan sınıftır.
 */
public class MWSRestClient
/**
 * Sunucudan Get istemi yapar.
 *
 * @param url İstem için URL
 * @param params İstem parametreleri
 * @param responseHandler İstem bitince çalıştırılacak metod
 */
public static void get(String url, RequestParams params,

```

```

        AsyncHttpResponderHandler responseHandler);

/**
 * Sunucudan Post istemi yapar.
 *
 * @param url Istem için URL
 * @param params Istem parametreleri
 * @param responseHandler Istem bitince çalıştırılacak metod
 */
public static void post(String url, RequestParams params,
        AsyncHttpResponderHandler responseHandler);

/**
 * Sunucudan Put istemi yapar.
 *
 * @param url Istem için URL
 * @param params Istem parametreleri
 * @param responseHandler Istem bitince çalıştırılacak metod
 */
public static void put(String url, RequestParams params,
        AsyncHttpResponderHandler responseHandler);

/**
 * Servis url'ini gönderir.
 *
 * @param relativeUrl Servis url
 * @return
 */
private static String getAbsoluteUrl(String relativeUrl);

/**
 * Restful servisi başlatır ve giriş adresi gönderir.
 *
 * @return Authorize url
 */
public static String getAuthorizeUrl();

```

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Halil İbrahim İBALI
Doğum Yeri	Konya
Doğum Tarihi	25.07.1985
Uyruğu	☒ T.C. ☐ Diğer:
Telefon	05552663605
E-Posta Adresi	ibrahimibali@yahoo.com
Web Adresi	-



Eğitim Bilgileri	
Lisans	
Üniversite	Doğuş Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	Bilgisayar Mühendisliği
Mezuniyet Yılı	30.06.2008

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Bilgisayar Mühendisliği Anabilim Dalı
Programı	Bilgisayar Mühendisliği Anabilim Dalı
Mezuniyet Tarihi	14.06.2013