# A CONTROL FRAMEWORK FOR MOBILE PAYMENT SYSTEMS

by

Kemal ELÇİ

B.S., Electronics & Communication Engineering, Yıldız Technical University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in System & Control Engineering
Boğaziçi University
2007

# ACKNOWLEDGEMENTS

First, I would like to thank my supervisor Dr. M. Tamer Şıkoğlu for the invaluable guidance he provide and endless inspiration he gave to me.

Secondly, I wish to express my sincere gratitude to my close friend, Ergun Bağcı who gave his valuable support throughout the project period.

I would like to thank my employers for the tolerance they show during my thesis work. I present my sincere thanks to Mehmet Aykul, our general director and Tufan Şahinoğlu, software department manager, for their encouragements.

I would also like to thank my committee members: Dr. Haluk Bingöl and Assist. Prof. Tunga Güngör.

Finally, I would like to thank my parents and especially my fiancée, Selda, whom I am deeply grateful to, because of their support and good wishes.

# ABSTRACT

## A CONTROL FRAMEWORK FOR MOBILE PAYMENT SYSTEMS

Mobile payment on wireless devices will provide excellent business opportunities in the coming years and offers consumers convenience and flexibility of mobile services anytime and at any place, and is playing an increasingly important role in payments and banking. This prediction is supported when considering the high rate of penetration of mobile devices, especially mobile phones, PDA's and other. Also an additional aspect is that the mobile phone can be used as payment device for all types of payment situations. Creating secure and cost-effective wireless payment solutions to support mobile device users not only provides good business opportunities, but also brings new technical challenges and issues to engineers. So how to build secured, easily manageable**,** wireless payment systems to support mobile payment transactions becomes a hot research topic.

In this project, a stable, secure and reliable mobile payment system is designed and implemented. With rule-based, user-customizable fraud detection system and additionally implemented high level security precautions; user friendly and intuitive interface, this architecture offers a feasible mobile payment framework.

# ÖZET

## MOBİL ÖDEME SİSTEMLERİ İÇİN BİR KONTROL YAPISI

Yakın gelecekte, kablosuz ödeme sistemleri, üstün iş fırsatları oluşturmaya, tüketicilere mobil hizmetleri, elverişli ve esnek bir biçimde, istenilen yerde ve zamanda sağlamaya başlayacak, bu sistemlerin bankacılık sektöründe ve ödeme işlemlerinde önemi artacaktır. Cep telefonlarının, PDA'ların ve diğer mobil cihazların yayılımı düşünüldüğünde, mobil ödeme sistemleri hakkındaki bu öngörü desteklenmektedir. Ayrıca mobil ödeme sistemlerinin bir diğer yanı ise cep telefonlarının her tipte ödemeler için kullanılabilmesidir. Güvenli ve uygun maliyetli kablosuz ödeme çözümleri üretmenin, iş fırsatları yaratmasının yanında, mühendislere çözülmesi gereken yeni teknik konular getirmesi de bir diğer yönüdür. Bu yüzden, mobil ödemeleri destekleyen, güvenli ve kolay yönetilebilir kablosuz ödeme sistemleri üretmek popüler bir araştırma konusudur.

Bu projede, stabil, güvenli ve sağlam bir mobil ödeme sistemi tasarlanmış ve geliştirilmiştir. Kural tabanlı, kullanıcı tarafından özelleştirilebilen dolandırıcılık tespit etme modülü, ek olarak sisteme dahil edilen yüksek güvenlik önlemleri, kullanıcı dostu, anlaşılması kolay arayüzü ile bu mimari gerçeklenebilir bir mobil ödeme yapısı sunmaktadır.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| 1G | First Generation |
| 2.5G | Second And A Half Generation |
| 2G | Second Generation |
| 3G | Third Generation |
| AMPS | Advance Mobile Phone System |
| API | Application Programming Interface |
| B2B | Business To Business |
| B2C | Business To Consumer |
| CDMA | Code Division Multiple Access |
| CDMA2000 | Code Division Multiple Access 2000 |
| CDMAONE | Code Division Multiple Access One |
| CHTML | Compact HTML |
| E-COMMERCE | Electronic Commerce |
| EDGE | Enhanced Data Rate for GSM Evolution |
| ETSI | European Telecommunications Standards Institute |
| FDMA | Frequency Division Multiple Access |
| FTP | File Transfer Protocol |
| GPRS | General Packet Radio System |
| GSM | Global System for Mobile Communication |
| GUI | Graphical User Interface |
| GUID | Globally Unique Identifier |
| HSCSD | High Speed Circuit Switched Data |
| HTML | Hyper Text Markup Language |

| | |
|---|---|
| HTTP | Hypertext Transfer Protocol |
| IIS | Internet Information Server |
| I-MODE | Information Mode |
| IRDA | Infrared Data Association |
| LAN | Local Area Network |
| M-COMMERCE | Mobile Commerce |
| MEXE | The Mobile Station Application Execution Environment |
| M-PAYMENT | Mobile Payment |
| NTT | Nippon Telegraph & Telephone |
| P2P | Person to Person |
| PDA | Personal Digital Assistant |
| SDK | Software Development Kit |
| SHA-1 | Secure Hash Algorithm 1 |
| SIM | Subscriber Identity Module |
| SMS | Short Messaging Service |
| SMTP | Simple Mail Transfer Protocol |
| SOAP | Simple Object Access Protocol |
| SQL | Structured Query Language |
| SSL | Secure Socket Layer |
| TACS | Total Access Communication System |
| TCP/IP | Transmission control protocol / Internet protocol |
| TDMA | Time Division Multiple Access |
| UDDI | Universal Description Discovery and Integration |
| UDP | User Datagram Protocol |
| UMTS | Universal Mobile Telephone Standard |

| | |
|---|---|
| USSD | Unstructured Supplementary Services Data |
| VPOS | Virtual Point Of Sale |
| WAP | Wireless Application Protocol |
| WML | Wireless Markup Language |
| WSDL | Web Service Description Language |
| XML | Extensible Markup Language |
| XMPP | Extensible Messaging and Presence Protocol |

# 1. INTRODUCTION

The demand for next generation mobile services is increasing as more mobile services are becoming available to the mainstream services market. Additional growth in this area is dependent on the technological and infrastructural support available. 2.5G and 3G mobile technologies presently are beginning to be adopted as a platform for the deployment of communication, business and leisure mobile services. With the technology gradually becoming available, the development and deployment of mobile services is increasingly an attractive market for Internet service providers, content providers and Mobile Commerce (M-Commerce) solution providers. M-Commerce can be defined as any electronic transaction or information interaction conducted using a mobile device and mobile networks, which leads to transfer of real or perceived value in exchange for information, services, or goods.

The growth of M-Commerce relies vitally on effective payment solutions, provided by mobile payment services. Mobile payment services are currently provided by mobile network operators, financial institutions and independent vendors. Many differences exist between these enclosed proprietary payment solutions. Although, there are a few organizations which were setup to develop a common mechanism for deploying mobile payment services, but as yet no common standard has been adopted for mobile payment services.

M-Commerce like E-Commerce can be B2B (business to business), P2P (person to person) or B2C (business to customer) oriented and offers consumers convenience and flexibility of mobile services anytime and at any place, and is playing an increasingly important role in payments and banking.

The future of mobile payments is promising, considering the high rate of penetration of mobile devices, especially mobile phones, PDA’s and other. Also an interesting aspect about M-Payments is that the mobile phone can be used as payment device for all types of payment situations.

The two basic forces listed below, grant the glorious future of mobile payments. These forces are:

- Mobile payments can be used for all types of payments (E-Commerce and standard commerce) any where and any time and can theoretically be carried out via any wireless device. The most obvious devices are mobile phones, PocketPCs, Laptops and PDAs.
- The increasing spread of mobile phones and technology.

Considering these aspects, mobile payment on wireless devices will provide excellent business opportunities in the coming years. Therefore, creating secure and cost-effective wireless payment solutions to support mobile device users not only provides good business opportunities, but also brings new technical challenges and issues to engineers, so how to build secured, easily manageable**,** wireless payment systems to support mobile payment transactions becomes a hot research topic.

## 1.1. Motivation

The aim of the thesis is to design a payment system architecture and a mobile client application which provide to make different types of payments by using especially mobile devices and a user interface to manage and monitor such payments. In order to achieve these three major tasks, a payment service will be developed to realize payment functionality, a mobile client application and a web application will be designed and implemented to provide user interaction.

### 1.1.1. Payment Service

Payment Service will be developed upon web services technologies and be able to receive payment request from mobile device users, communicate with banks through virtual point of sale (VPOS) APIs and inform the payer about the response.

As payment service is a web service, it will have the interoperability, platform independency, flexibility, ease of use functionalities that make it easier to integrate the service with third party systems such as end user applications, B2B, B2C, P2P services and other payment services.

Payment Service will also work as a security gateway for credit card payments with built-in security functionalities, and will provide protection against fraud. Security mechanism will have a rule-based, user customizable, flexible architecture and also supported with a Secure Socket Layer (SSL) certificate to sustain network transmission security.

By having the functions explained above and the user informative features like E-Mail and SMS notifications, payment service will offer an advanced and secure framework for most types of payments.

### 1.1.2. Management Application

To control and manage payment service, security gateway and system modules like notification module, authentication module; a web application will be developed. Using this application, money transfers and payment requests will be observed, security filters will be managed, various notifications will be set and all other user interaction related tasks will be achieved.

Both payment service and management application will be build upon Microsoft.NET Framework 2.0 and Sql Server 2005 will be used for database services.

### 1.1.3. Mobile Client Application

To receive payment requests from mobile devices, mobile client application will be developed. This application will be able to work on mobile phones, responsible for getting the payment information, sending to the payment service and displaying request results to client. For compatibility and consistency issues, a web based application will be designed, which can be executed by all kinds of mobile browsers and independent of the mobile operating system or device model.

## 1.2. Outline

In Chapter 2, related technologies and required background of this thesis will be presented. In Chapter 3, system’s core parts, such as payment service, security concepts, management and mobile client application design will be examined. The last chapter will summarize the work and discuss the future work which will improve and add value to the system.

# 2. BACKGROUND

This chapter describes various technologies related to this thesis, such as web services, mobile network technologies and mobile communication services. Some of these mobile technologies are currently in existence on global mobile networks, while the other technologies are gradually becoming adopted by mobile operators.

## 2.1. Web Services

The term “Web Services” describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an internet protocol backbone [3]. It is also defined by the W3C [4] as "a software system designed to support interoperable Machine to Machine interaction over a network". Web services are frequently just Web APIs that can be accessed over a network, such as the internet, and executed on a remote system hosting the requested services. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients. Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

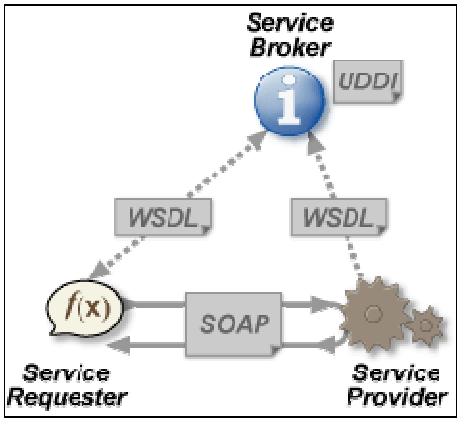


Figure 2.1. Web Services Architecture [5]

Unlike traditional client/server models, such as a web server / web page system, web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. The applications interface, not the users. Developers can then add the web service to a GUI (such as a web page or an executable program) to offer specific functionality to users.

Web services allow different applications from different sources to communicate with each other without time-consuming custom coding, and because all communication is in XML. Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl, Windows applications can talk with UNIX applications etc.

Web services do not require the use of browsers or HTML, and are sometimes called application services.

The specifications that define web services are intentionally modular, and as a result there is no one document that contains them all. Additionally, there is neither a single, nor a stable set of specifications. There are a few "core" specifications that are supplemented by others as the circumstances and choice of technology dictates, including:

### 2.1.1. SOAP

An XML-based, extensible message envelope format with “bindings” to underlying protocols. The primary protocols are HTTP and HTTPS, although bindings for others, including SMTP and XMPP, have been written [5].

### 2.1.2. Web Services Description Language (WSDL)

An XML format that allows service interfaces to be described along with the details of their bindings to specific protocols. Typically used to generate server and client code, and for configuration [5].

### 2.1.3. Universal Description Discovery and Integration (UDDI)

A protocol for publishing and discovering metadata about Web services that enables applications to find them, either at design time or runtime [5].

Most of these core specifications have come from W3C, including XML, SOAP, and WSDL; UDDI comes from OASIS.

## 2.2. Mobile Technologies

### 2.2.1. Network Technologies

Mobile network [1] technologies have evolved from analog based systems to digital based systems and from circuit switching to packet switching technologies. This evolution can be described by different generations of mobile technologies, i.e. first generation (1G), second-generation (2G), 2.5G and third-generation (3G) technologies. Only 1G is based on analog technology. Some of the main standards for each generation technology are:

- 1G: Advance Mobile Phone System (AMPS) in North America, Total Access Communication System (TACS) in UK, Nippon Telegraph & Telephone (NTT) in Japan, Code Division Multiple Access One (CDMAONE).
- 2G: Global System for Mobile Communication (GSM), Code Division Multiple Access 2000 (CDMA2000), High Speed Circuit Switched Data Technology (HSCSD).
- 2.5G: General Packet Radio System (GPRS), Enhanced Data Rate for GSM Evolution (EDGE).
- 3G: Universal Mobile Telephone Standard (UMTS)

### 2.2.2. GSM

Global System for Mobile Communication [1] is a second generation standard for mobile communication, developed by the European Telecommunications Standards Institute (ETSI) and now currently owned by the Third Generation Partnership Project (3GPP). Operating in the 900 MHz and the 1800 MHz frequency band [2], GSM is the most widespread mobile standard currently in use across Europe and the Asia-Pacific region. GSM was designed using digital techniques, unlike with previous analog cellular systems like AMPS in the US and TACS in the United Kingdom. The techniques used are a combination of Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA), which are primarily for voice transmission and control. Since all users must share a limited radio spectrum, these techniques are used to divide the bandwidth among as many users as possible. Also, Space Division Multiple Access is used to provide a system based on a series of base stations each covering a limited area.

FDMA divides the radio frequency into several frequency carriers of 200 Hz, while TDMA enables 8 voice channels in each 200 Hz carrier by dividing each one in time.

<u>2.2.2.1. GSM Services</u> : Telecommunication services can be divided into bearer services, teleservices, and supplementary services. The most basic teleservice supported by GSM is telephony.

Data Services are as follows :

- Internet Services: GSM users can send and receive data, at rates up to 9.6K bps, to users on POTS (Plain Old Telephone Service), ISDN, Packet Switched Public Data Networks, and Circuit Switched Public Data Networks.
- SMS (Short Messaging Service): SMS is a bidirectional service for short alphanumeric (up to 160 bytes) messages
- Facsimile: Sending and receiving of fax messages, using a GSM phone and a laptop computer.

- Secure Corporate LAN Access: secure access to e-mails, faxes, and file transfer via an encrypted link to a corporate LAN.
- Supplementary Services: [1] Such services include call forwarding, call barring, caller identification, call waiting and multiparty conversation. These services can be controlled via service applications using a GSM network API, allowing application developers to access GSM network capabilities.

GSM technologies are limited due to its low data transmission speed, therefore with the growth in data services the long term future of GSM is uncertain, unless it is changed to offer high bandwidth data services. Also, internet browsing using GSM phones is subject to charging of on-line duration and reconnection is necessary for each browsing session, as opposed to with GPRS (General Packet Radio Service), in which charging is based on the data received or viewed and all time connectivity is available.

2.2.2.2. HSCSD : High Speed Circuit Switched Data [1, 2] is a circuit switched protocol based on GSM, providing an enhancement of data services. HSCSD enables higher rates by using multiple channels as apposed to single voice channel with GSM. Transmissions rates can be up 57.6 Kbps by using 4 radio channels simultaneously. Typically, HSCSD [1] was directed at mobile PCs rather than smart phones, where a PCMCIA card is used with transmission speeds of 42.3 Kbps downstream and 28.8 Kbps upstream. HSCSD was intended as a temporary substitute for GPRS, to improve the transmission rates of existing mobile data applications.

2.2.2.3. GPRS : General Packet Radio Service [1] is packet switched wireless protocol providing non-voice value added services that allows information to be sent and received across a mobile telephone network. It is described as a 2.5G technology which supplements Circuit Switched technology such as GSM. Data transmissions speeds of 9.6 kbps to a theoretical maximum speed of up to 171.2 kbps are achievable with GPRS using all eight timeslots at the same time. In addition to higher data rates, GPRS provides users with all time connectivity while only charged for the data viewed or received with a minimal online charge.

GPRS is an evolutionary step towards 3G technologies, such as EDGE (Enhanced Data GSM Environment) and UMTS (Universal Mobile Telephone Service). GPRS may be considered as an overlay network on the GSM networks, using the GSM resources to the fullest potential. To enable this, extra network elements are required for this packet based mobile network. Certain hardware elements are added to provide the IP infrastructure needed for packet based services. The SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support node) are the mobile network equivalents of routers and gateways. Other main additions are the upgrading with new software to existing cellular infrastructure.

GPRS only uses its radio resources when users are actually sending or receiving data, therefore the available radio resource can be concurrently shared between several mobile data users, rather than dedicating a radio channel to a single user for a fixed period of time. This efficient use of scarce radio resources means that large numbers of GPRS users can potentially share the same bandwidth and be served from a single cell. GPRS uses the same radio channel as voice calls, a channel that is 200 kHz wide and which carries a raw digital radio stream of 271 kbps. For voice calls this channel is divided into 8 separate data streams, each carrying about 34 kbps. After protocol and error correction overhead, 13 kbps is left for each voice connection or about 14 kbps for data. Packet-switched data can use several channels where as circuit-switched data uses one voice channel. GPRS can combine up to 8 of these channels, and with 14 kbps of data throughput each, the delivered bandwidth can be up to 100 Kbps. Most economical phones will be ones that are limited to 56 kbps, as not all eight voice channels have to be used. A mobile station can request the amount of bandwidth it desires at the time it establishes a data session. GPRS applications includes Intranet and Internet access, E-Mail, Fax, and Unified messaging, using a single mailbox for all messages, including voice mail, faxes, e-mail, short message service (SMS), and pager messages.

Limitations of GPRS [1, 2]:

- The limited cell capacity during voice and GPRS transmission calls. The use of a bearer for a different type of radio resource, such as SMS, would better utilize the cell capacity.

- Achieving the theoretical maximum GPRS data transmission speed of 172.2 kbps would require a single user taking over all eight timeslots which is unlikely that a network operator will allow all timeslots to be used by a single GPRS user. The bandwidth available to a GPRS user will therefore be severely limited.
- Suboptimal Modulation - GPRS employs a modulation technique called Gaussian minimum-shift keying (GMSK) while the EGDE uses a new modulation technique to allow a much higher bit rate across an air interface, called eight phase-shift keying (8PSK) modulation. This type of modulation is used for 3G systems, so upgrading to 3G technology seems inevitable for a network operator.
- Transit Delays - GPRS sends data packets through different channels to reach a destination, therefore data corruption or data loss may occur. Data integrity and retransmission capabilities are used to avoid this, but the result is that potential delays can occur.
- No Store and Forward - Unlike SMS technology, GPRS doesn't provide a store and forward mechanism for data transmission, therefore SMS may be need to enable sending and receiving of short messages.

2.2.2.4. EDGE : Enhanced Data for Global Evolution [1] is a higher bandwidth version of GPRS permitting transmission speeds of up to 384 Kbps. It is compatible with the GSM protocol, but it requires higher quality radio signals to reach the increased speed. Deploying EDGE will allow mobile network operators to offer high-speed, mobile multimedia applications. It allows a migration path from GPRS to UMTS, because the modulation changes that will be necessary for UMTS at a later stage will already be implemented. The opportunity window for EDGE may be very short, unless major delays occur during UMTS deployment.

2.2.2.5. 3G : 3rd Generation [1] is the generic term for the next big step in mobile technology development. The formal standard for 3G is the IMT-2000 (International Mobile Telecommunications 2000). There are three optional modes as part of the 3G standard. W-CDMA (Wireless Code Division Multiple Access) is for Europe and for the Asian GSM countries, CDMA (Code Division Multiple Access) is for North America, and then TDD/CDMA (Time Division Duplex/CDMA) for China.

2.2.2.6. UMTS : Universal Mobile Telephone System [1] is designed to provide for 3G mobile data services. Realistic expectations suggest a maximum capacity in metropolitan areas of 384 Kbps, at least in the early years of its deployment. The same transmission rate can be achieved much earlier with EDGE. This third generation mobile phone system is already available in Japan. The system enables the transmission of video, data and voice communication at a high speed and low cost.

2.2.2.7. CDMA : Code Division Multiple Access [1] is a proprietary standard for mobile communication, where GSM is an open standard. CDMA was pioneered by Qualcomm and enhanced by Ericsson. Both standards are in competition for dominance in the cellular world. CDMA is adopted mostly in US where it has a large subscription base. CDMA is a spread spectrum technology, which means that it spreads the information contained in a particular signal of interest over a much greater bandwidth than the original signal. A CDMA call starts with a standard rate of 9.6 kbps, which is then spread to a transmitted rate of about 1.23 Mbps.

### 2.2.3. Mobile Communication Services

2.2.3.1. SMS : Short Messaging Service [1] was created as a part of the GSM Phase 1 standard to send and receive short text messages, of 70-160 alphanumeric characters in length, to and from mobile phones. The number of characters which can be sent is dependent on the language in use, with language support limited to the European Languages, Chinese and Arabic. This service is widely popular in Europe and Asia while in the US it is practically non-existent. SMS requires digital wireless interface standard (GSM) which is slowly being adopted in the US. In the US the ‘mobile-party-pays’ pricing model is commonly used, so mobile users pay for incoming as well as outgoing calls. Similarly this is the case with text messaging, so paying for messages received is slowing down the adoption of SMS in the US.

SMS is a smart service, as it can store messages when to the target mobile device is switched off and forwards the messages when the unit is again in use. SMS applications are voicemail/fax notifications, delivery of replacement ring-tones, operator logos and group graphics, unified messaging, personal communication (text messaging), and information services. Basically, any information that fits into a short text message can be delivered by SMS.

The majority of SMS messages are peer-to-peer (mobile-to-mobile) text messages at around 90% of SMS traffic, and the remaining 10%, are mobile transaction services such as news, stock prices, weather, horoscope, etc. SMS continues to grow more as a payment medium, e.g. reverse SMS billing, premium SMS numbering, and as a combination with advanced messaging solutions built around instant messaging via GPRS or e-mail.

2.2.3.2. WAP and WML : Wireless Application Protocol [1] is a technology which provides a mechanism for displaying internet information on a mobile phone or any wireless device. This is done by translating internet information in to a format which can be displayed within the constraints of a mobile device. WAP is an open standard, developed by the WAP Forum, which has over 500 members. Its founder members include the major wireless vendors of Nokia, Ericsson and Motorola, plus the US software company, Phone.com (formerly Unwired Planet).

To obtain Internet access on a mobile device, the device should be WAP-enabled and the web site information should be described in WML (Wireless Markup Language) format. WML [6], based on XML, is a content format for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones, and preceded the use of other markup languages now used with WAP, such as XHTML and even standard HTML.

WML [6] documents are XML documents that validate against the WML DTD (Document Type Definition). The W3C Markup Validation service (http://validator.w3.org/) can be used to validate WML documents.

Wireless Markup Language [6] is a lot like HTML (Hyper Text Markup Language) in that it provides navigational support, data input, hyperlinks, text and image presentation, and forms. A WML document is known as a “deck”. Data in the deck is structured into one or more “cards” (pages) – each of which represents a single interaction with the user. The introduction of the terms "deck" and "card" into the internet and mobile phone communities was a result of the user interface software and its interaction with wireless communications services having to comply with the requirements of the laws of two or more nations.

WML decks are stored on an ordinary web server trivially configured to serve the text/vnd.wap.wml MIME type in addition to plain HTML and variants. The WML cards when requested by a device are accessed by a bridge WAP gateway, which sits between mobile devices and the World Wide Web, passing pages from one to the other much like a proxy. The gateways radio the WML pages in a form suitable for mobile device reception. This process is hidden from the phone, so it may access the page in the same way as a browser accesses HTML, using a URL provided the mobile phone operator has not specifically locked the phone to prevent access of user-specified URLs.

WML has a scaled down set of procedural elements which can be used by the author to control navigation to other cards.

It is an error and misconception to think of WML as a pinhole view of the Internet. The real magic and value of WML is that it provides an interface with the phone hardware to initiate a call based on web content requested by user query.

The acceptance of WML has been limited by the fact that cell phone providers require separate activation and additional fees for data support, and also because telecommunications companies have sought to limit data access to only "approved" data providers operating under "license" of the signal carrier.

A WAP gateway is also necessary between the client mobile device and the WML host server, to translate the WAP request. The response from the host server is translated into a WAP response by the WAP gateway, which can be displayed on the mobile device. An application environment, called WAE (WAP Application Environment), is defined by the WAP standard to enabling the development of advanced services and applications. These include micro-browsers, scripting facilities, e-mail, www-to-mobile messaging, and mobile to telefax access.

There has being difficulties with the launch of WAP, especially in Europe, due to the slow speed and high charges when using WAP on GSM technology. The increase use of GPRS sees an increase popularity of WAP usage. WAP has been very popular in Asia, except in Japan where I-mode is dominate in this market. WAP is an open standard in contrast to I-mode, which is a proprietary standard.

2.2.3.3. I-Mode : I-mode [1] (I standing for information) is a wireless technology developed by a Japanese company called NTT DoCoMo, which enables users to access Internet services via their cellular phones. I-Mode can be used to exchange e-mail with computers, personal digital assistants (PDAs) and other I-Mode cellular phones. I-Mode has already dominated the Japanese market and is being considered a success story in the world of M-Commerce.

I-Mode's underlying technology is uncomplicated, which makes it easy for content providers to create new I-Mode services and easy for customers to find and use them. The service is based on the Asian cellular standard PDC and uses Compact HTML (cHTML) markup language. cHTML is basically a scaled down version of HTML. It is relatively easy and it takes little time to rewrite HTML into cHTML. I-Mode’s transmission speed is just 9.6kbps, but fast enough for its services. DoCoMo operates a packet-switched network, which means that customers pay not for time elapsed but for the packets of data they download. Packet switching also means that I-Mode is always on, so customers don't have to log into the service or wait for a connection, but have immediate access to services, similarly with GPRS.

2.2.3.4. USSD : Unstructured Supplementary Services Data [1] is a mechanism of transmitting information via a GSM network. Similar to SMS, but it is only basically a store and forward service. USSD offers a real-time connection during a session. It is said that USSD will grow with the further market penetration of WAP. Its main uses will be for mobile financial services, shopping and payment.

2.2.3.5. Cell Broadcast : Cell broadcast is a technology [1] that is designed for simultaneous delivery of short messages to multiple mobile users within a specified region or nation-wide. Cell broadcast is similar to SMS, but it is a one-to-many service rather than a one-to-one or one-to-few. It is a mass distribution media mainly for news and generic information.

Usually, cell broadcast services are distributed to the consumer on at no cost basis. The network operator charges the content provider for sending the messages and the content provider will try to make money on follow-up services, such as advertising.

2.2.3.6. SIM Toolkit : SIM (Subscriber Identity Module) Toolkit [1] is an ETSI/SMG standard for value added services and e-commerce using GSM phones to perform the transactions. SIM Toolkit programmed into the special GSM SIM card enables the SIM card, using the GSM handset, to build up an interactive exchange between a network application and the end user and access or control access to the network. Therefore, it provides the SIM card with a proactive role in the handset. This means that the SIM initiates commands independently of the handset and the network. SIM Toolkit is targeted at phones that do not yet fall into the smart phone category. Although SIM Toolkit was being heavily pushed by the smartcard industry, it will be an interim technology and will not be able to survive once GPRS terminals take over the market, since WAP is be the GPRS-supported protocol.

2.2.3.7. Web Clipping : The Web Clipping [1] service for 3Com's Palm handheld device has been very successful, utilizing Palm's 75% market share of PDA market in the US. Web clipping is a Palm proprietary format for delivery of web-based information to Palm devices via synchronization or wireless communication to the Palm platform. Web clipping may exist with WAP in the fragmented US market. However, in Europe it is likely to be superseded, even on the Palm platform, by WAP based services.

2.2.3.8. MexE : The Mobile Station Application Execution Environment [1] is the incorporation of a Java virtual machine into the mobile phone, allowing full application programming. The protocol is integrating location services, sophisticated intelligent customer menus and a variety of interfaces, such as voice recognition. MExE will incorporate WAP, but also provides additional services exceeding the WAP functionality.

2.2.3.9. Base Network Protocols : Infrared data association (IrDA) [1] is a protocol stack which represents the physical characteristics of infrared communication. This wireless communication mechanism enables establishment of connections between devices, which must be in line of sight of each other.

Bluetooth [1] has become the predominant standard for lower power and short-range radio link to exchange information, enabling wireless connectivity between devices and peripherals. It had been adopted by many mobile phone manufacturers, and introduced as an addition communication feature on most new phones.

Hypertext transfer protocol (HTTP) [1] is a text-based protocol for content transfer over the internet. This protocol is used to access web content via a web browser on a mobile device.

Transmission control protocol / Internet protocol (TCP/IP) [1] is a protocol suite consisting of several protocols at the transport and network layers. At the transport layer, there is the TCP and UDP (User Datagram Protocol) protocols, which are considered reliable and unreliable protocols, respectively. TCP is a stream-oriented and UDP is a packet based protocol. Both can be used to establish socket connections between networked devices.

# 3. SYSTEM CORE

## 3.1. Overview

Designed System consist of three main sections: a payment service, which handles the payment operations, management application, provides user interaction and a mobile client application for payment interface. Also whole system is named as "EverPayNET" and domain name "www.everpay.net " has been registered for demonstration purposes.

Main elements of the system and relations between them are visualized in the figure below.

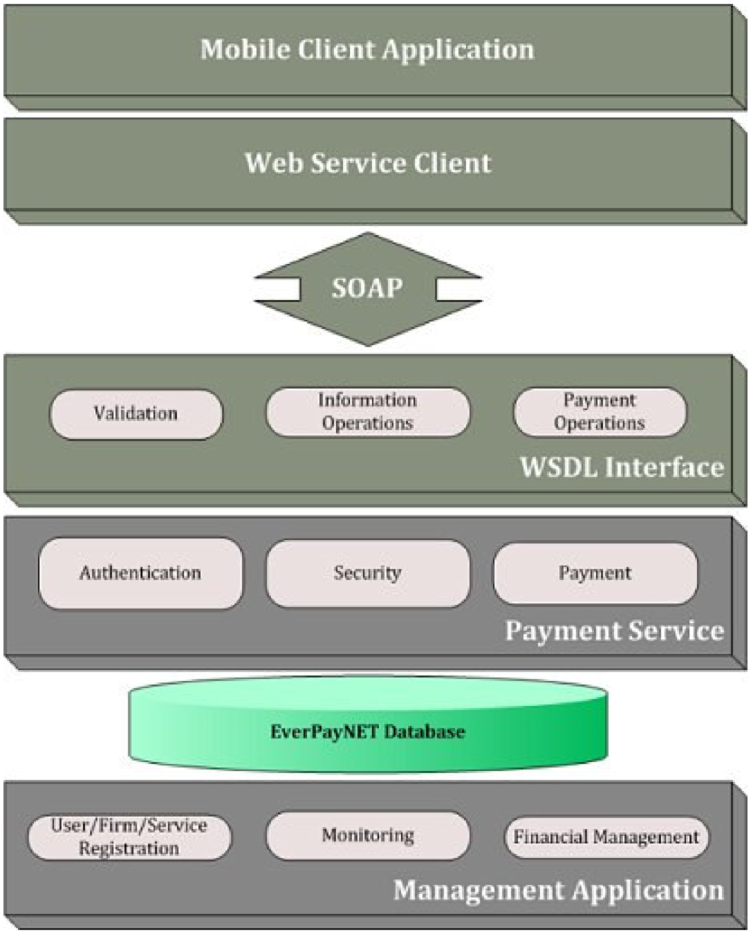


Figure 3.1. System Architecture

EverPayNET system is built upon Microsoft.NET Framework 2.0 Web Service technology and coded using C#. NET on Visual Studio 2005 with ServicePack 1. Having over 150 classes and approximately 10.000 lines of code, system tries to use the power of .NET technologies at maximum levels.

As the system is highly database dependant; a consistent, reliable and secure database is needed. Microsoft SQL 2005 is selected for database services due to these. Having more than 50 tables, 20 views and 60 stored procedures can be considered as a proof of high database dependency. Instead of using hard coded SQL queries, stored procedures are created on the database for decreasing the query execution latency.

### 3.2. Payment Service

EverPayNET Payment Service is responsible for getting payment requests from clients, processing the requests within the security concepts, communicating with the bank's virtual points of sale (VPOS), getting the response and informing the client about these results. As payment service is a web service, both mobile and non-mobile clients, web and operating system applications are able to communicate with the system. This thesis is focused on only mobile client communication. Able to work with other types of clients can be considered as a future promising feature of the system.

In the next page, system application and main payment operation sequence diagrams can be observed.

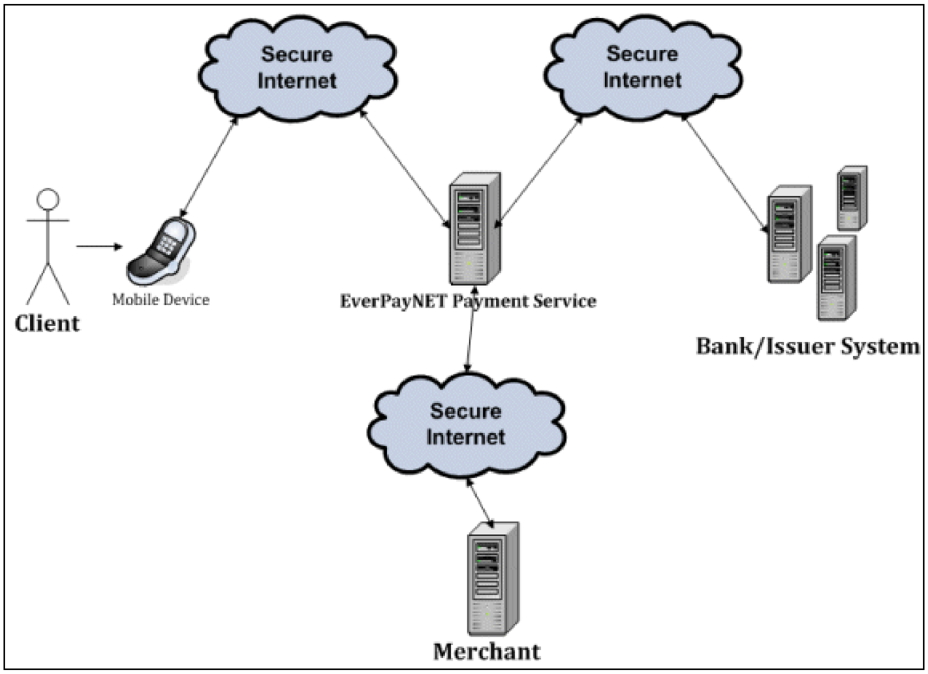


Figure 3.2. System Application Diagram

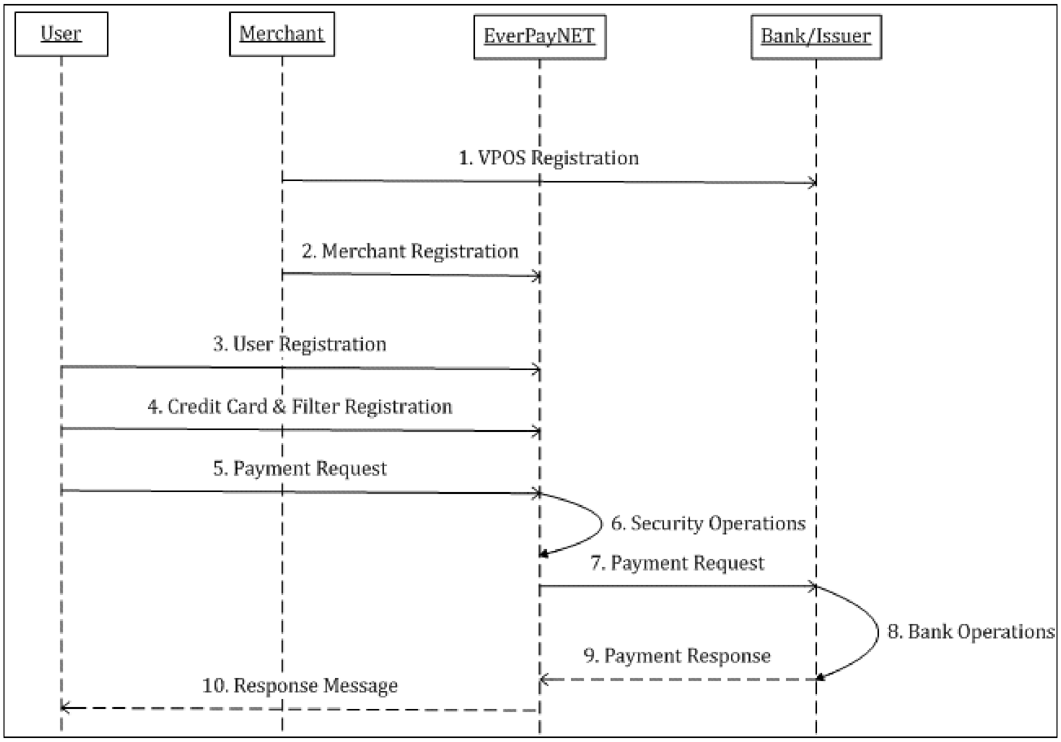
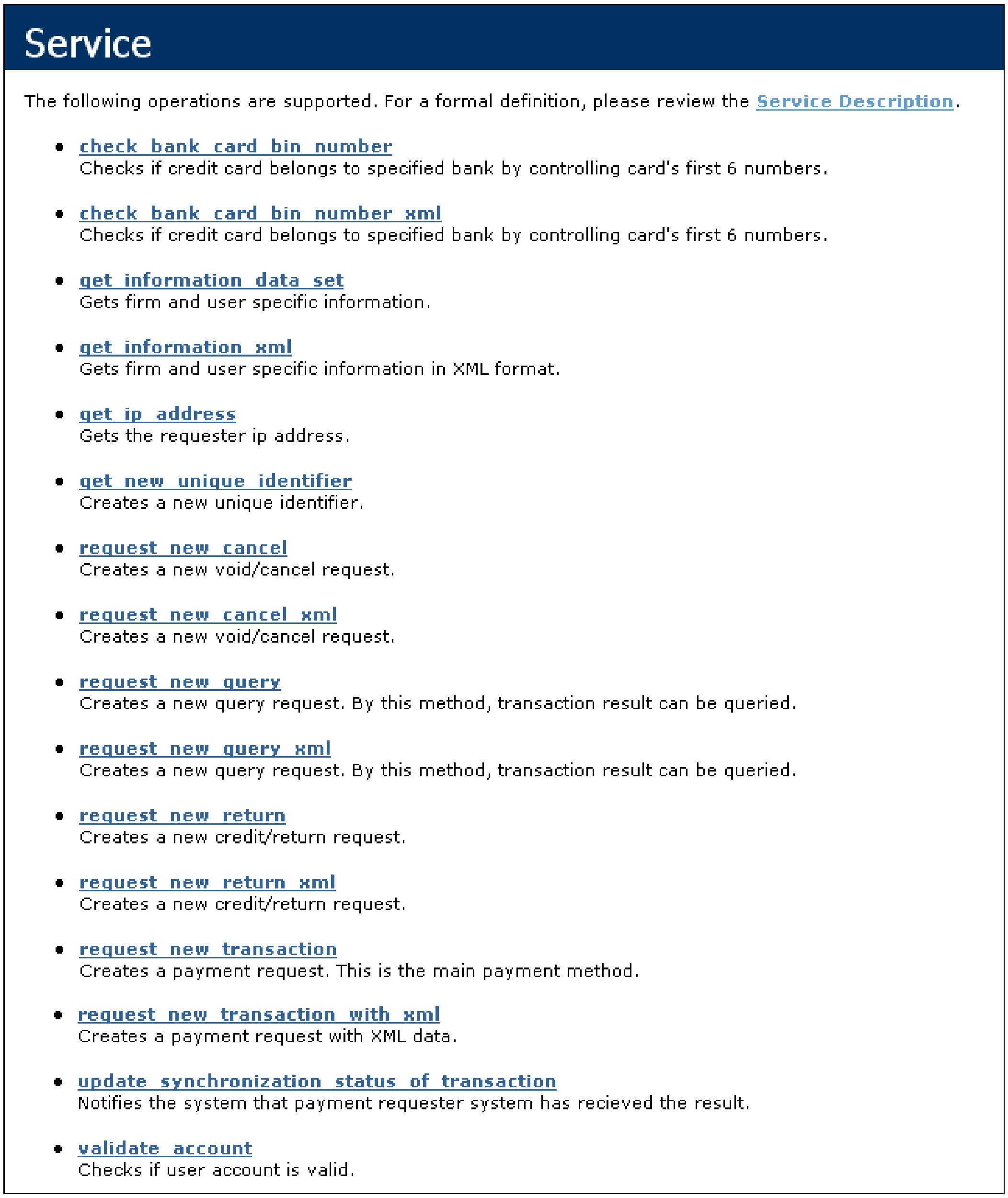


Figure 3.3. Main Payment Process Sequence

### 3.2.1. Web Service Interface

EverPayNET Payment Service's web service description language (WSDL) file, in a human readable form can be seen below.



## Service

The following operations are supported. For a formal definition, please review the Service Description.

- check bank card bin number
  Checks if credit card belongs to specified bank by controlling card's first 6 numbers.
- check bank card bin number xml
  Checks if credit card belongs to specified bank by controlling card's first 6 numbers.
- get information data set
  Gets firm and user specific information.
- get information xml
  Gets firm and user specific information in XML format.
- get ip address
  Gets the requester ip address.
- get new unique identifier
  Creates a new unique identifier.
- request new cancel
  Creates a new void/cancel request.
- request new cancel xml
  Creates a new void/cancel request.
- request new query
  Creates a new query request. By this method, transaction result can be queried.
- request new query xml
  Creates a new query request. By this method, transaction result can be queried.
- request new return
  Creates a new credit/return request.
- request new return xml
  Creates a new credit/return request.
- request new transaction
  Creates a payment request. This is the main payment method.
- request new transaction with xml
  Creates a payment request with XML data.
- update synchronization status of transaction
  Notifies the system that payment requester system has recieved the result.
- validate account
  Checks if user account is valid.

Figure 3.4. Web Service WSDL Interface

Payment Service consumer applications can do required payment related tasks by using listed web methods like "*request_new_transaction* ", "*request_new_cancel"* etc. Some methods return or receive data *in Microsoft Framework 2.0 System.Data.DataSet* format. This format is very handy if consumer application is a .NET application. However non - .NET applications may have difficulties in adopting .NET dataset into their systems. To overcome this issue, every DataSet returning or receiving web method has an XML clone. Although DataSet is an XML driven type, it may require more effort than simple XML string.

Short descriptions of methods are listed below in an alphabetical order.

3.2.1.1. check_bank_card_bin_number : First 6 digit of 16 digit credit cards, called "bin number", indicates the card's issuer bank. Payment Service has almost every credit card-bank relation information in its database. By using this method, these relations can be checked if they are valid or not.

Table 3.1. Input parameters of check_bank_card_bin_number

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |
| bank_id | *String* | Bank ID |
| bin_number | *String* | Bin Number |

Table 3.2. Return value of check_bank_card_bin_number

| Type | Description |
|---|---|
| *System.Data.DataSet* | Bank-Bin relation |

3.2.1.2. check_bank_card_bin_number_xml : This method is an XML returning clone of "*check_bank_card_bin_number* " method.

Table 3.3. Input parameters of check_bank_card_bin_number_xml

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |
| bank_id | *String* | Bank ID |
| bin_number | *String* | Bin Number |

Table 3.4. Return value of check_bank_card_bin_number_xml

| Type | Description |
|---|---|
| *String* | Bank-Bin relation in XML format |

3.2.1.3. get_information_data_set : Payment Service holds the firm-user relations, firm-vpos relations, vpos-instalment relations and more. By using this method firm and user specific payment related information can be received.

Table 3.5. Input parameters of get_information_data_set

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |

Table 3.6. Return value of get_information_data_set

| Type | Description |
|---|---|
| *System.Data.DataSet* | Information dataset |

3.2.1.4. get_information_xml : This method is an XML returning clone of "*get_information_data_set*" method.

Table 3.7. Input parameters of get_information_xml

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |

Table 3.8. Return value of get_information_xml

| Type | Description |
|---|---|
| *String* | Information data in XML format |

3.2.1.5. get_ip_address : Using this method, client application get its current ip address.

Table 3.9. Input parameters of get_ip_address

| Param Name | Type | Description |
|---|---|---|
| No parameters. | | |

Table 3.10. Return value of get_ip_address

| Type | Description |
|---|---|
| *String* | IpAddress |

3.2.1.6. get_new_unique_identifier : Global unique identifier (GUID) is a term used by Microsoft for a number that its programming generates to create a unique identity for an entity. GUIDs can be created in a number of ways, but usually they are a combination of a few unique settings based on specific point in time (e.g., an IP address, network MAC address, clock date/time, etc.) [14].

Using this method, client applications can create GUIDs, if they are unable to create one.

Table 3.11. Input parameters of get_new_unique_identifier

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User’s Username |
| user_password | *String* | Firm User’s Password |

Table 3.12. Return value of get_new_unique_identifier

| Type | Description |
|---|---|
| *String* | An new GUID |

3.2.1.7. request_new_cancel : This method is used for voiding/canceling a successful transaction. Void/cancel operations can be done for transactions, requested in the same day before the “Day End” operation. “Day End” operations are automatically occurs everyday on the issuer’s payment systems.

Table 3.13. Input parameters of request_new_cancel

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User’s Username |
| user_password | *String* | Firm User’s Password |
| return_number | *Double* | Human Readable Transaction Number |

Table 3.14. Return value of request_new_cancel

| Type | Description |
|---|---|
| *System.Data.DataSet* | DataSet Containing Result Info. |

3.2.1.8. request_new_cancel_xml : This mehtod is an XML returning clone of "*request_new_cancel* " method.

Table 3.15. Input parameters of request_new_cancel_xml

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |
| return_number | *Double* | Human Readable Transaction Number |

Table 3.16. Return value of request_new_cancel_xml

| Type | Description |
|---|---|
| *String* | XML Data Containing Result Info. |

3.2.1.9. request_new_query : This is a method for querying the transaction result in case of connection interruption or other problems.

Table 3.17. Input parameters of request_new_query

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |
| return_number | *Double* | Human Readable Transaction Number |

Table 3.18. Return value of request_new_query

| Type | Description |
|---|---|
| *System.Data.DataSet* | DataSet Conatining Result Info. |

3.2.1.10. request_new_query_xml : This method is an XML returning clone of *"request_new_query"* method.

Table 3.19. Input parameters of request_new_query_xml

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User’s Username |
| user_password | *String* | Firm User’s Password |
| return_number | *Double* | Human Readable Transaction Number |

Table 3.20. Return value of request_new_query_xml

| Type | Description |
|---|---|
| *String* | XML Data Containing Result Info. |

3.2.1.11. request_new_return : This method is used for crediting/refunding a successful transaction. Credit/Refund operations can be done for transactions which have had “Day End” operation.

Table 3.21. Input parameters of request_new_return

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User’s Username |
| user_password | *String* | Firm User’s Password |
| return_number | *Double* | Human Readable Transaction Number |
| return_amont | *Float* | Amount to credit |

Table 3.22. Return value of request_new_return

| Type | Description |
|---|---|
| *System.Data.DataSet* | DataSet Containing Result Info. |

3.2.1.12. request_new_return_xml : This method is an XML returning clone of “ *request_new_return* ” method.

Table 3.23.  Input parameters of request_new_return_xml

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User's Username |
| user_password | *String* | Firm User's Password |
| return_number | *Double* | Human Readable Transaction Number |
| return_amont | *Float* | Amount to credit |

Table 3.24.  Return value of request_new_return_xml

| Type | Description |
|---|---|
| *String* | XML Data Containing Result Info. |

3.2.1.13. request_new_transaction：This is the main transaction request operation.

Table 3.25.  Input parameters of request_new_transaction

| Param Name | Type | Description |
|---|---|---|
| request_data_set | *System.Data.DataSet* | DataSet containing credential, payment, product info |

Table 3.26.  Return value of request_new_transaction

| Type | Description |
|---|---|
| *System.Data.DataSet* | DataSet Containing Result Info. |

3.2.1.14. request_new_transaction_with_xml：This method is an XML returning clone of *" request_new_transaction"* method.

Table 3.27.  Input parameters of request_new_transaction_with_xml

| Param Name | Type | Description |
|---|---|---|
| request_xml | *String* | XML string, containing credential, payment, product info |

Table 3.28. Return value of request_new_transaction_with_xml

| Type | Description |
|---|---|
| *String* | XML Data Containing Result Info. |

3.2.1.15. update_synchronization_status_of_transaction : Using this method, client application notifies Payment Service that the client has seen the result successfully.

Table 3.29. Input parameters of update_synchronization_status_of_transaction

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User’s Username |
| user_password | *String* | Firm User’s Password |
| response_id | *String* | Unique identifier Transaction Response ID |

Table 3.30. Return value of update_synchronization_status_of_transaction

| Type | Description |
|---|---|
| No return value. | |

3.2.1.16. validate_account： Account validation/credential operations can be achieved by using this method.

Table 3.31. Input parameters of validate_account

| Param Name | Type | Description |
|---|---|---|
| firm_id | *String* | Firm ID |
| user_name | *String* | Firm User’s Username |
| user_password | *String* | Firm User’s Password |

Table 3.32. Return value of validate_account

| Type | Description |
|---|---|
| *String* | UserID if account is valid, else *Null* value. |

### 3.3. Security Concepts

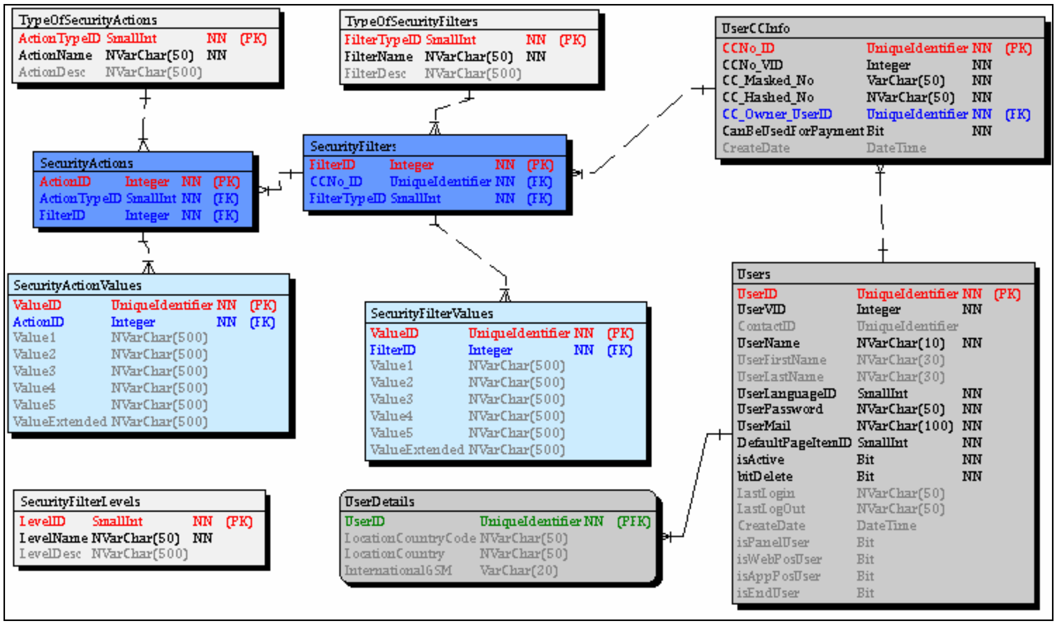
Designing a payment system needs more effort than a regular database oriented web service because of the security issues. Client's credit card information must be safely received, processed; user confidentiality must be preserved and a strong user authentication system should be implemented [13]. This payment system provides all of these and more.

First of all, to sustain the Internet transmission security and eliminate the network sniffing threat, 128 Bit Secure Sockets Layer (SSL) certificate is implemented on Internet Information Server (IIS) which is the web server application on Microsoft operating systems. SSL [8] is a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data - a public key known to everyone and a private or secret key known only to the recipient of the message. It is the most common way to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with "https" instead of "http".

Clients' credit card details are very sensitive, high security requiring and critical type of information [12, 13]. Online payment services should follow some certain policies and are responsible for providing such security. Considering these responsibilities, proposed payment service does not hold any abusable credit card data on its database. In a transaction process, sensitive data is stored in the web server's "Session" object which is unique and specific for browser request. For security filters, which will be detailed later, credit card number is hashed by SHA1 algorithm before storing on the database. SHA-1 (Secure Hash Algorithm) [9] is a most commonly used from SHA series of cryptographic hash functions, designed by the National Security Agency of USA and published as their government standard. It takes a message of less than 264 bits in length and produces a 160-bit message digest designed so that it is computationally very expensive to find a text string that matches a given hash.

EverPayNET Payment Service also works as a security gateway for credit card payments. It runs a rule based, user customizable security mechanism. These rules are defined as “Security Filters” in the system. Each filter belongs to a credit card and has various values that determine the filter behavior when being checked against transactions. Also each filter can have one or more security actions like email or sms notification, transaction denial etc. Actions are programmed to be done if their belonging filters’ conditions become true. All filter and action types will be detailed in the next chapter.

The database entities of security system can be observed in the figure below. “SecurityFilters” and “SecurityActions” are the main tables to hold filter and action records. Value tables hold filter values, “TypeOf” prefixed tables hold filter and action type information and “UserCCInfo” table holds user credit card information.



| TypeOfSecurityActions | | | |
|---|---|---|---|
| ActionTypeID | SmallInt | NN | (PK) |
| ActionName | NVarChar(50) | NN | |
| ActionDesc | NVarChar(500) | | |

| TypeOfSecurityFilters | | | |
|---|---|---|---|
| FilterTypeID | SmallInt | NN | (PK) |
| FilterName | NVarChar(50) | NN | |
| FilterDesc | NVarChar(500) | | |

| UserCCInfo | | | |
|---|---|---|---|
| CCNo_ID | UniqueIdentifier | NN | (PK) |
| CCNo_VID | Integer | NN | |
| CC_Masked_No | VarChar(50) | NN | |
| CC_Hashed_No | NVarChar(50) | NN | |
| CC_Owner_UserID | UniqueIdentifier | NN | (FK) |
| CanBeUsedForPayment | Bit | NN | |
| CreateDate | DateTime | | |

| SecurityActions | | | |
|---|---|---|---|
| ActionID | Integer | NN | (PK) |
| ActionTypeID | SmallInt | NN | (FK) |
| FilterID | Integer | NN | (FK) |

| SecurityFilters | | | |
|---|---|---|---|
| FilterID | Integer | NN | (PK) |
| CCNo_ID | UniqueIdentifier | NN | (FK) |
| FilterTypeID | SmallInt | NN | (FK) |

| SecurityActionValues | | | |
|---|---|---|---|
| ValueID | UniqueIdentifier | NN | (PK) |
| ActionID | Integer | NN | (FK) |
| Value1 | NVarChar(500) | | |
| Value2 | NVarChar(500) | | |
| Value3 | NVarChar(500) | | |
| Value4 | NVarChar(500) | | |
| Value5 | NVarChar(500) | | |
| ValueExtended | NVarChar(500) | | |

| SecurityFilterValues | | | |
|---|---|---|---|
| ValueID | UniqueIdentifier | NN | (PK) |
| FilterID | Integer | NN | (FK) |
| Value1 | NVarChar(500) | | |
| Value2 | NVarChar(500) | | |
| Value3 | NVarChar(500) | | |
| Value4 | NVarChar(500) | | |
| Value5 | NVarChar(500) | | |
| ValueExtended | NVarChar(500) | | |

| Users | | | |
|---|---|---|---|
| UserID | UniqueIdentifier | NN | (PK) |
| UserVID | Integer | NN | |
| ContactID | UniqueIdentifier | | |
| UserName | NVarChar(10) | NN | |
| UserFirstName | NVarChar(30) | | |
| UserLastName | NVarChar(30) | | |
| UserLanguageID | SmallInt | NN | |
| UserPassword | NVarChar(50) | NN | |
| UserMail | NVarChar(100) | NN | |
| DefaultPageItemID | SmallInt | NN | |
| isActive | Bit | NN | |
| bitDelete | Bit | NN | |
| LastLogin | NVarChar(50) | | |
| LastLogOut | NVarChar(50) | | |
| CreateDate | DateTime | | |
| isPanelUser | Bit | | |
| isWebPosUser | Bit | | |
| isAppPosUser | Bit | | |
| isEndUser | Bit | | |

| SecurityFilterLevels | | | |
|---|---|---|---|
| LevelID | SmallInt | NN | (PK) |
| LevelName | NVarChar(50) | NN | |
| LevelDesc | NVarChar(500) | | |

| UserDetails | | | |
|---|---|---|---|
| UserID | UniqueIdentifier | NN | (PFK) |
| LocationCountryCode | NVarChar(50) | | |
| LocationCountry | NVarChar(50) | | |
| InternationalGSM | VarChar(20) | | |

Figure 3.5. Security Entities

Configuring the security filters is an easy task, by the help of user-friendly, fast and stable management interface. End Customers must register their credit cards to the system first before defining custom filters. Credit card registration is more complex than it sounds. It is needed to have valid card information and system must be sure that the current user is the real owner of the credit card. To satisfy these needs, system gets the full credit card information including credit card number, expiration date and the CVC/CV2 code from the card owner, tries to charge a small amount of money by sending the charge request to the bank. If charge request is successful, system asks the user to enter the random charged amount to prove being the card owner. At this stage, users must check their bank accounts to find out the charged amount. After the validation process, the card is added to the system and charged amount is voided. Whole process sequence can be seen in the diagram below.

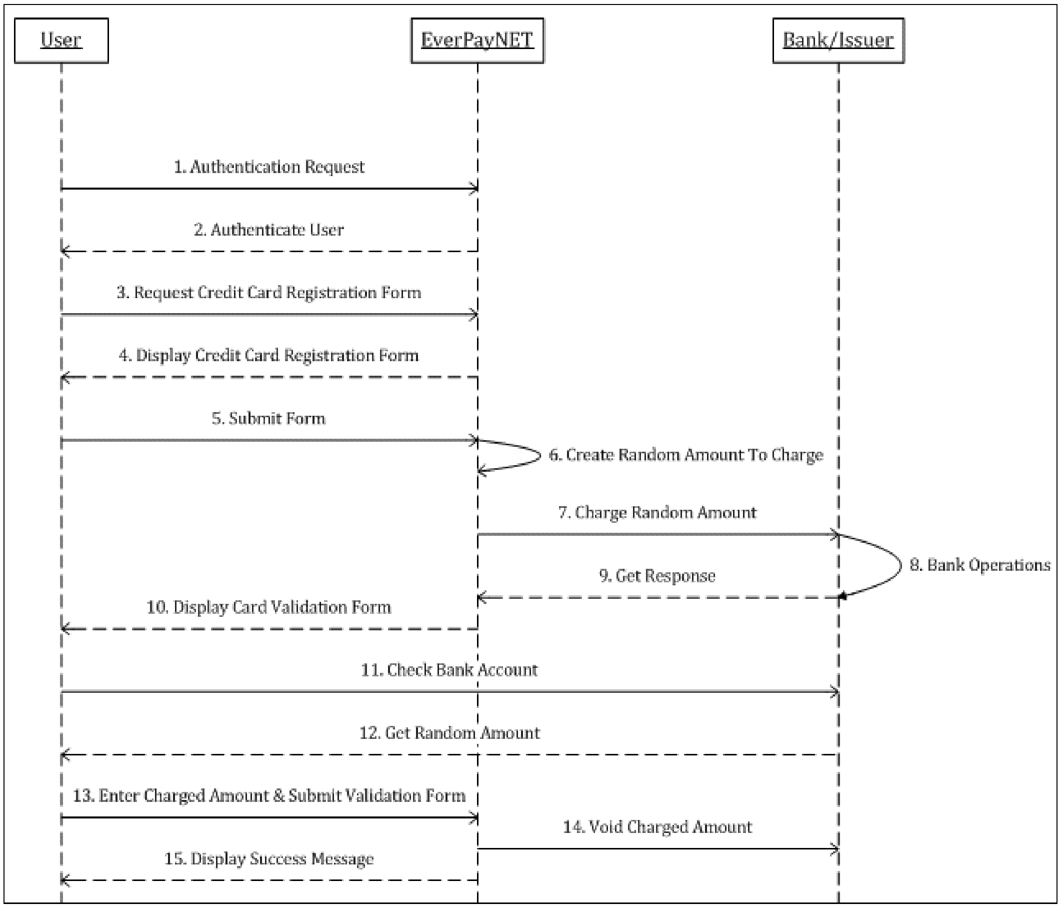

User
EverPayNET
Bank/Issuer
1. Authentication Request
2. Authenticate User
3. Request Credit Card Registration Form
4. Display Credit Card Registration Form
5. Submit Form
6. Create Random Amount To Charge
7. Charge Random Amount
8. Bank Operations
9. Get Response
10. Display Card Validation Form
11. Check Bank Account
12. Get Random Amount
13. Enter Charged Amount & Submit Validation Form
14. Void Charged Amount
15. Display Success Message

Figure 3.6. Credit Card Registration Sequence

Also EverPayNET keeps history of logging attempts and knows actively logged on users. For security reasons, system does not let two active logons with same account at the same time.

### 3.4. Management Application

EverPayNET Management Application is designed to control, manage and monitor transactions by providing user interactivity. It is a web based application, and highly dependant on database services as much as the payment service. With its intuitive and stylish design, this application is easy-to-use and handy tool for managing the whole system. It also supports multiple languages. There exist xml files for each language containing language info. System displays the preferred language for user after logon.

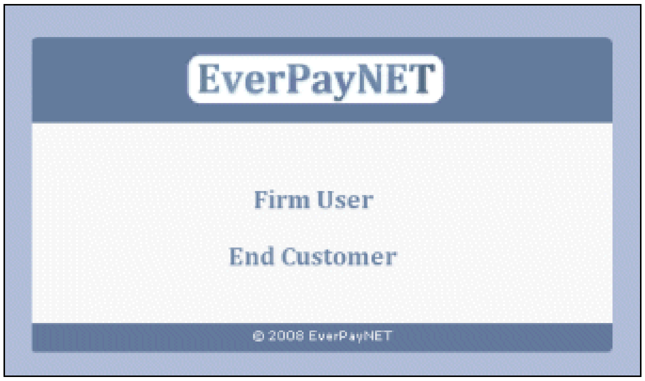


Figure 3.7. Management Application User Type Selection Screen

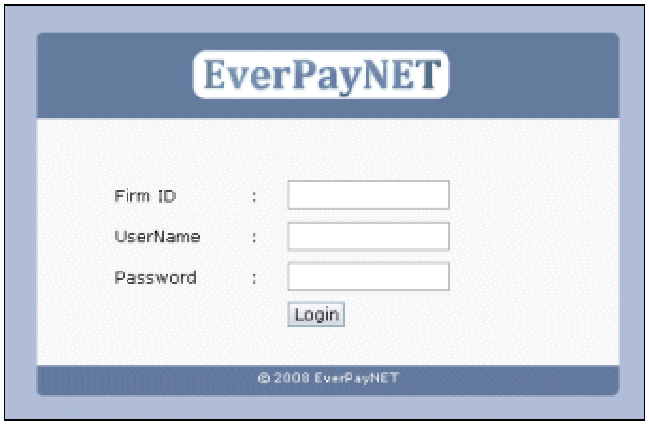


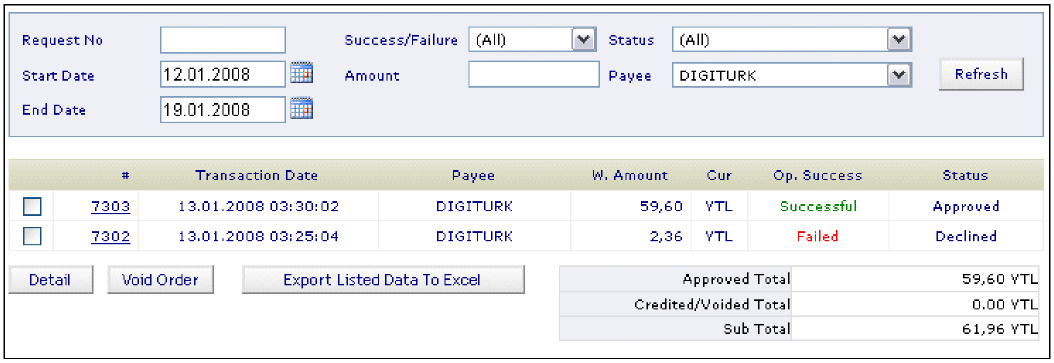
Figure 3.8. Management Application Login Screen

Three main user levels are defined in the system and Management Application can provide different content to these user levels. These user levels are; system administrator, firm user and end customer. System administrator has all access rights, can view and use every module for every firm and user. Firm users are able to manage only their firm related actions and view only their firm related information. End customers are able to monitor their payments and manage security filters for their credit card payments.

Management Application has two main modules containing sub modules with different functionality and design. These modules are detailed below.

### 3.4.1. Monitoring Tools

This module contains transaction lists that let end customers track their payments, firm users list received payments and provide void/refund functionality. This list also has a strong filtering capability with the help of implemented search tool, and able to export listed data to popular “Excel” format.



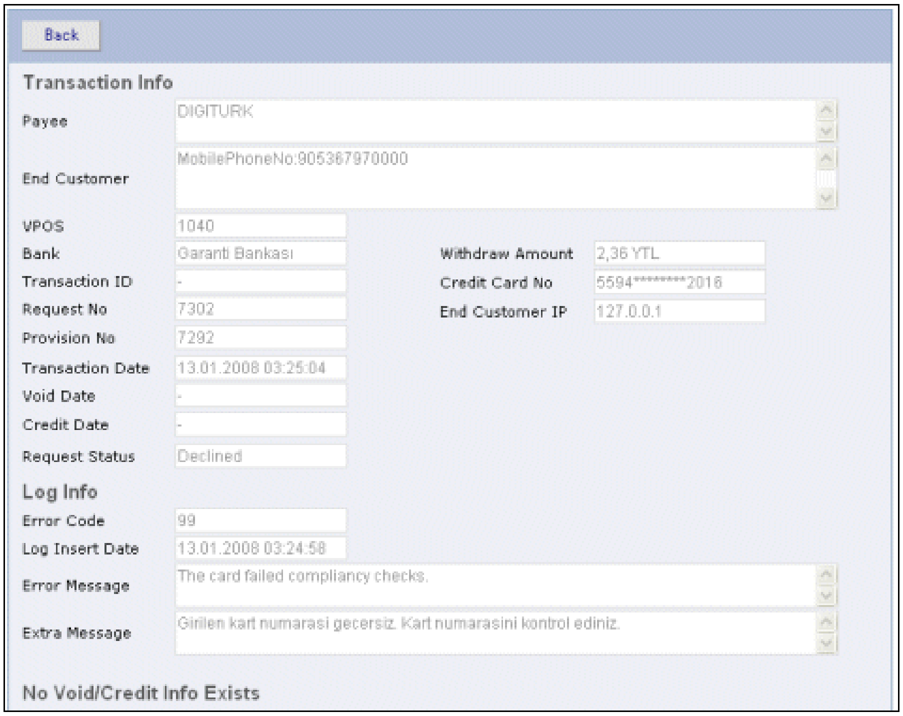
| | | | | | |
|---|---|---|---|---|---|
| Request No | | Success/Failure | (All) | Status | (All) |
| Start Date | 12.01.2008 | Amount | | Payee | DIGITURK |
| End Date | 19.01.2008 | | | | |

| | # | Transaction Date | Payee | W. Amount | Cur | Op. Success | Status |
|---|---|---|---|---|---|---|---|
| □ | 7303 | 13.01.2008 03:30:02 | DIGITURK | 59,60 | YTL | Successful | Approved |
| □ | 7302 | 13.01.2008 03:25:04 | DIGITURK | 2,36 | YTL | Failed | Declined |

| | |
|---|---|
| Approved Total | 59,60 YTL |
| Credited/Voided Total | 0.00 YTL |
| Sub Total | 61,96 YTL |

Figure 3.9. Main Monitoring Screen

Further details of transaction info can be seen in the “Transaction Detail” page including client ip, masked credit card no, error messages if transaction is declined and void/credit history if exists.



Transaction Info

| | | | |
|---|---|---|---|
| Payee | DIGITURK | | |
| End Customer | MobilePhoneNo:905367970000 | | |
| VPOS | 1040 | | |
| Bank | Garanti Bankası | Withdraw Amount | 2,36 YTL |
| Transaction ID | - | Credit Card No | 5594********2016 |
| Request No | 7302 | End Customer IP | 127.0.0.1 |
| Provision No | 7292 | | |
| Transaction Date | 13.01.2008 03:25:04 | | |
| Void Date | - | | |
| Credit Date | - | | |
| Request Status | Declined | | |

Log Info

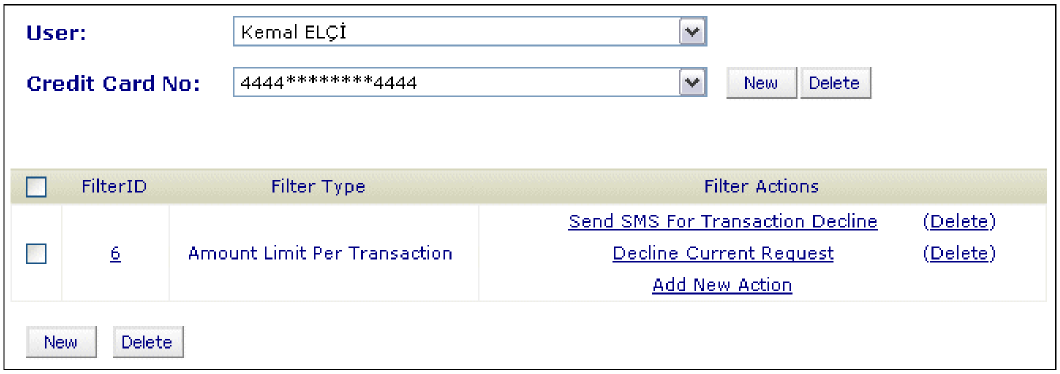
| | |
|---|---|
| Error Code | 99 |
| Log Insert Date | 13.01.2008 03:24:58 |
| Error Message | The card failed compliancy checks. |
| Extra Message | Girilen kart numarasi gecersiz. Kart numarasini kontrol ediniz. |

No Void/Credit Info Exists

Figure 3.10. Transaction Detail Screen

### 3.4.2. Security Tools

This module forms the most crucial component of the EverPayNET system. Fraud detection systems can be considered as the heart of payment systems for the sake of payers. This module contains filtering tools for defining custom rules to detect and prevent fraud. Credit card registration, custom filter and custom action management tools belong to this module.



User: Kemal ELÇİ

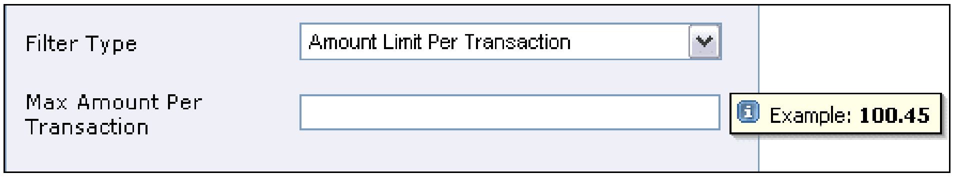
Credit Card No: 4444********4444

| □ | FilterID | Filter Type | Filter Actions | |
|---|---|---|---|---|
| □ | 6 | Amount Limit Per Transaction | Send SMS For Transaction Decline<br>Decline Current Request<br>Add New Action | (Delete)<br>(Delete) |

Figure 3.11. Security Management Screen

There are 10 types of security filters predefined in EverPayNET system. These filters are:

3.4.2.1. Amount Limit Per Transaction : This filter defines a maximum limit for a transaction.



Filter Type: Amount Limit Per Transaction

Max Amount Per Transaction:

Example: 100.45

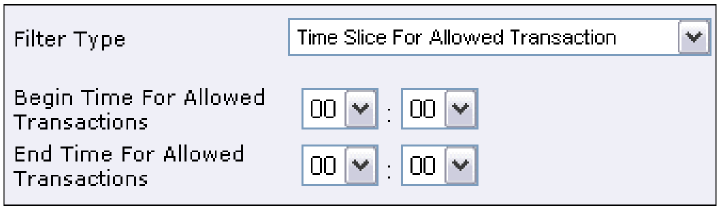
Figure 3.12. Amount Limit per Transaction Filter Screen

3.4.2.2. Amount Limit Per Day : This filter defines a maximum limit of total transaction amounts per day.

3.4.2.3. Amount Limit Per Week : This filter defines a maximum limit of total transaction amounts per week.

3.4.2.4. Amount Limit Per Month : This filter defines a maximum limit of total transaction amounts per month.

3.4.2.5. Time Slice For Allowed Transactions : This filter defines a time slice for valid transactions.



Filter Type: Time Slice For Allowed Transaction

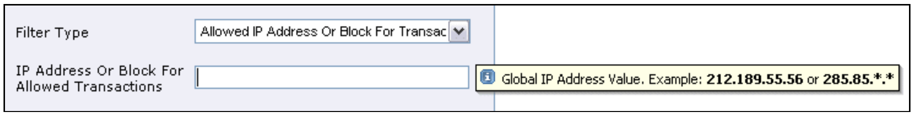
Begin Time For Allowed Transactions: 00 : 00

End Time For Allowed Transactions: 00 : 00

Figure 3.13. Time Slice For Allowed Transactions Filter Screen

3.4.2.6. Time Slice For Forbidden Transactions **:** This filter defines a time slice for invalid transactions. Requesting a transaction inside this time slice fires the filter.

3.4.2.7. Allowed IP Address Or IP Block For Transaction : This filter defines a IP address or IP address block for a valid transaction. Transaction requests from machines having ip address other than this ip address are blocked.



Filter Type: Allowed IP Address Or Block For Transac

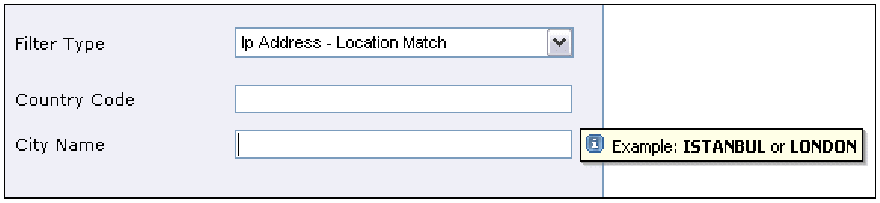
IP Address Or Block For Allowed Transactions

Global IP Address Value. Example: **212.189.55.56** or **285.85.*.***

Figure 3.14. Allowed IP Address Or IP Block For Transaction Filter Screen

3.4.2.8. Prohibited IP Address Or IP Block For Transaction : This filter defines a IP address or IP address block for an invalid transaction. Transaction requests from machines having this ip address are blocked.

3.4.2.9. Ip Address – Location Match : Some online public web services provide information about ip address-location match. Using these services, client location can be determined from ip address. EverPayNET uses *FraudLabs™ IP2Location Web Service* [10] to retrieve location info. Defining this filter with a location data, users can distinguish transactions from locations other than user's own.



Filter Type: Ip Address - Location Match

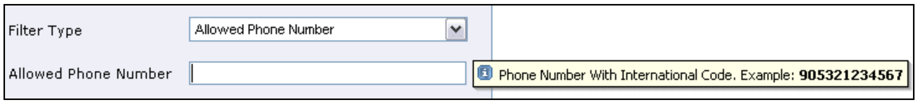
Country Code

City Name

Example: **ISTANBUL** or **LONDON**

Figure 3.15. Ip Address – Location Match Filter Screen

3.4.2.10. Allowed Phone Number : In mobile payment systems, mobile phone numbers can be obtained from request variables created by mobile web applications. The catch is phone number variable is not public, an official agreement needed to make with Mobile Phone Operators. In this project, phone number variable is just simulated.



Filter Type: Allowed Phone Number

Allowed Phone Number

Phone Number With International Code. Example: **905321234567**

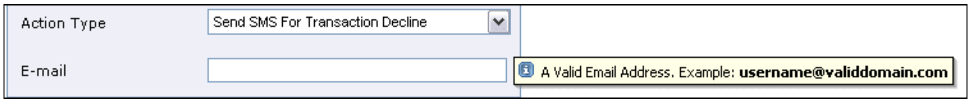
Figure 3.16. Allowed Phone Number Filter Screen

Also EverPayNET system has 4 types of customizable actions that each one can be attached to security filters. Each filter can have one or more actions which can be done if filter conditions become true. These actions are:

3.4.2.11. Decline Current Request : By defining this action, current request can be blocked.

3.4.2.12. Decline All Requests : This action not only blocks the current request but also block all future transaction request from that account.

3.4.2.13. Send Email For Transaction Decline : If users need to be notified by email, this action can be used. EverPayNET system uses .NET Framework 2.0 System.Net.Mail namespace components to send emails.



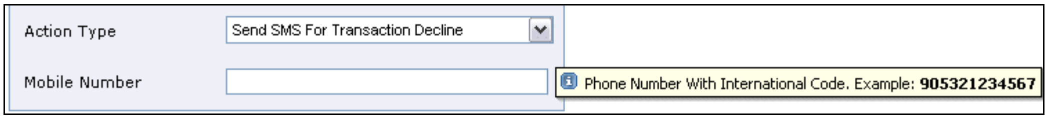
Action Type: Send SMS For Transaction Decline

E-mail:

A Valid Email Address. Example: **username@validdomain.com**

Figure 3.17. Send Email for Transaction Decline Filter Screen

3.4.2.14. Send SMS For Transaction Decline : SMS notification are handled by this action. EverPayNET uses *Clickatell Bulk SMS Gateway Service* [11] to send text messages.



Action Type: Send SMS For Transaction Decline

Mobile Number:

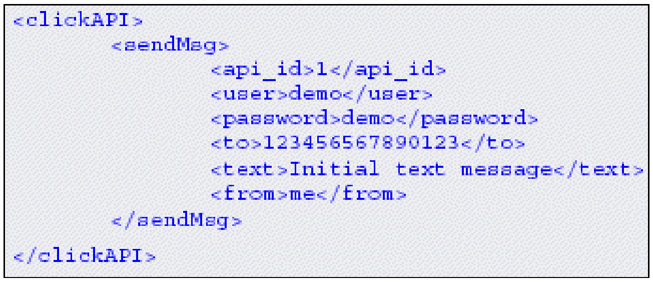
Phone Number With International Code. Example: **905321234567**

Figure 3.18. Send SMS for Transaction Decline Filter Screen

*Clickatell* is the world's leading provider of bulk SMS messaging services and SMS gateway connectivity. This service offers access to a secure, dependable, high capacity SMS messaging platform also provides easy-to-use APIs to help developers build SMS services for their systems.

*Clickatell* provides various types of APIs to developers such as HTTP/S API, FTP API, SMTP API, COM Object API and XML API. EverPayNET uses XML API for extensibility and simplicity. XML API interface has its own set of DTDs and supports XML over HTTP.

An XML sample can be seen below.



```
<clickAPI>
        <sendMsg>
                <api_id>1</api_id>
                <user>demo</user>
                <password>demo</password>
                <to>123456567890123</to>
                <text>Initial text message</text>
                <from>me</from>
        </sendMsg>
</clickAPI>
```

Figure 3.19. SMS XML API Sample

### 3.5. Mobile Client Application

To complete the payment framework, a mobile client application is designed and implemented. This is a web based application, coded using WML and .NET Mobile SDK. Its main task is getting payment information from client on a mobile device, convert this information to a form that payment service can process, and send. Designing this application as a web application eliminates mobile operating system compatibility issues. Any mobile device with any operating system which is able to connect and surf the internet can display and execute this application without crucial problems.

Figure below is the enter screen of the application:

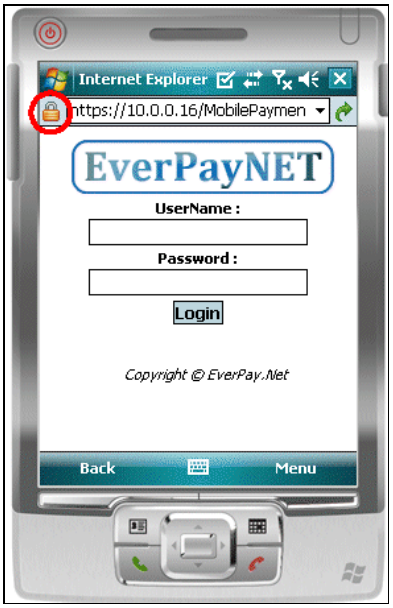
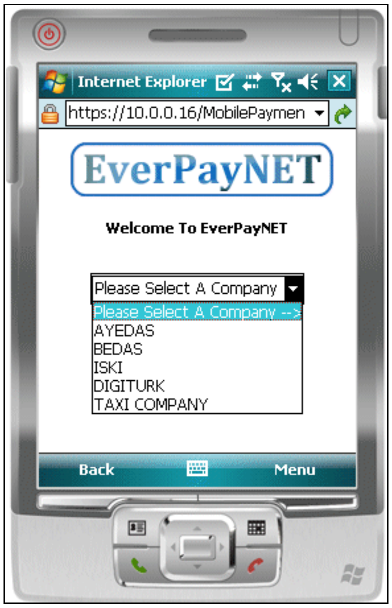


Figure 3.20. Mobile Client Application Login Screen

Users enter their EverPayNET username and password to login to the system. Also the lock sign can be seen on the top left of the screen. This indicates that application is running on a SSL certificate installed server and the current communication is secure.

On the second screen, users select a payee, to make payment to.



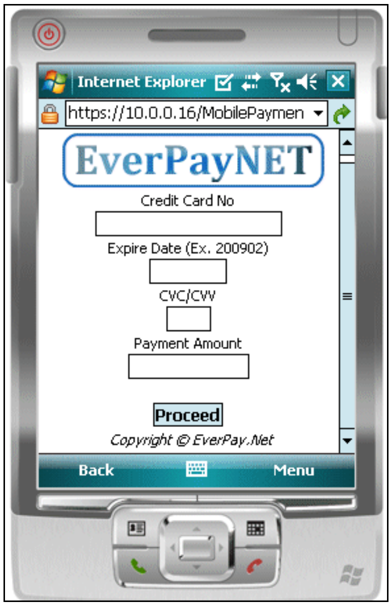
Welcome To EverPayNET

Please Select A Company

Please Select A Company -->
AYEDAS
BEDAS
ISKI
DIGITURK
TAXI COMPANY

Figure 3.21. Mobile Client Application Payee Selection Screen

On the next screen users enter their credit card info and payment amount then click to proceed. After process complete, request result will be shown to user.



Credit Card No

Expire Date (Ex. 200902)

CVC/CVV

Payment Amount

Proceed

Copyright © EverPay.Net

Figure 3.22. Mobile Client Application Payment Screen

# 4. CONCLUSION

In this research a mobile payment system is developed and implemented including a payment service, a mobile and a management application. Each part of the system has been designed considering security, flexibility and interoperability issues. Also user-friendly, intuitive and smart front end designs of the applications have added value to the system. Evaluating the whole architecture, it can be said that this design can form a framework for mobile payment systems.

Also system is open for future developments with the help of well designed, well documented and object oriented architecture. As future works, there will be some additional tasks to improve the system. Next paragraphs are describing these improvements.

For improving the fraud detection module, an artificial intelligent system may be implemented. This type of systems usually determine user habits by processing the past payment data. After this process, every user has a payment habit chart, which will be used when a new transaction request comes. New transaction data is compared to the past data and any possible deviation will be considered as fraud attempt. This detection system relies on the training data and without having a consistent one, system fails. This type of a fraud detection system is considered as future work because of the absence of this training data.

Being a web service, payment service can serve not only mobile clients but also pc clients and accept transaction request from all kinds of applications running of different platforms. Using this power, this framework can be expanded from mobile to an online payment framework.

# REFERENCES

1. Amit Vyas and Peter O’Grady, “A Review of Mobile Commerce Technologies”, Department of Industrial Engineering, University of Iowa, 2001

2. John Scourias, “Overview of the Global System for Mobile Communication”, University of Waterloo, 2001

3. Web Services, “http://www.webopedia.com/TERM/W/Web_services.html”, August 2007

4. Web Services Architecture (W3C Working Group Note), “http://www.w3.org/TR/ws-arch/” , August 2007

5. Web Services, “http://en.wikipedia.org/wiki/Webservices”, August 2007

6. WML, “http://en.wikipedia.org/wiki/Wireless_Markup_Language”, August 2007

7. WAP / WML Tutorial, “http://www.w3schools.com/wap/default.asp”, August 2007

8. Secure Socket Layer, “http://www.webopedia.com/TERM/S/SSL.html”, August 2007

9. What is SHA-1, “http://www.accuhash.com/what-is-sha1.html”, August 2007

10. FraudLabs IP2Location Geolocation Web Service, “http://www.fraudlabs.com/ip2location.aspx”, November 2007

11. Clickatell SMS Gateway, “https://www.clickatell.com”, November 2007

12. Natali Deli and Ana Vukašinovi, "Mobile Payment Solution – Symbiosis between banks, application service providers and mobile network operators", IEEE, 2006

13. Seema Nambiar, Chang-Tien Lu and Lily R. Liang, "Analysis of Payment Transaction Security in Mobile Commerce", IEEE, 2004

14. What is GUID,
"http://searchcio-midmarket.techtarget.com/sDefinition/
0,,sid183_gci213990,00.html", August 2007

# REFERENCES NOT CITED

Antoinette Leung, Zhuang Yan and Simon Fong, “On Designing a Flexible E-Payment System with Fraud Detection Capability”, IEEE, 2004.

Chuang-Cheng Chiu and Chieh-Yuan Tsai, “A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection”, IEEE, 2004

Osama Dandash, Phu Dung Le and Bala Srinivasan, “Security Analysis for Internet Banking Models”, IEEE, 2007

Jun Liu, Jianxin Liao and Xiaomin Zhu, “A System Model and Protocol for Mobile Payment”, IEEE, 2005

T.N.T. Nguyen, P. Shum and E. H. Chua, “Secure End-To-End Mobile Payment System”, IEEE, 2005