

인터넷뱅킹을 위한 어플리케이션의 설계 및 구현

남철기*, 장길상*

Design and Implementation of an Application for Internet Banking

Chulki Nam, Gilsang Jang

Abstract

Recently, Internet services are rapidly spreading all over the world. Internet will have a great influence on financial market and banking industry. Customers will obtain full banking services over Internet. Thus, Internet banking is an unavoidable choice to strengthen the competitive edge of banks.

This paper designs and develops an application for Internet banking services in the existing banks. The implementation of Internet banking requires the construction of website, web pages and links to other related information. Using Oracle DBMS(Database Management System), Oracle Application Server, PL/SQL(Procedural Language/SQL), Java Applet, and Java Script, a prototype for Internet banking is implemented.

Key Word : 인터넷뱅킹(Internet banking), 금융서비스, 은행산업(Banking Industry), 어플리케이션

* 한국오라클(주)

1. 서 론

인터넷은 전세계적으로 걸쳐있는 수많은 네트워크의 모임으로서, 거대한 컴퓨터들의 네트워크에 힘입어 엄청난 컴퓨팅의 파워를 보여주고 있다. 시간이 흐르면서 사용자의 수와 규모, 정보 제공자들이 계속적으로 빠른 속도로 증가하고 있다. 따라서, 정보, 거래 및 여가를 위하여 인터넷에 접속하는 회수는 엄청나게 증가하여 왔다. 인터넷상의 방대한 양의 서비스에 대한 요구는 현재 사용자들의 요구를 만족시키기 위해 다양한 어플리케이션이 개발되게 되었다. 오늘날의 정보 제공자들은 반드시 인터넷과 온라인 서비스를 사용해서 고객의 요구에 부응하여야 한다. 그렇지 않으면 주요 고객들을 새로운 정보 제공자들에게 빼앗길 수 있다. 인터넷 및 WWW (World-Wide Web)의 폭발적인 증가와 수많은 홈 소비자의 직접적인 디지털 쌍방향 통신(Interaction)의 가능성은 은행, 신용카드회사, 보험회사 등을 포함한 금융서비스 업계에게 엄청난 도전과 기회를 제시하고 있다[Leong, Srikanthan, Hura, 1999].

최근 인터넷을 기반으로 한 금융서비스의 요구가 증대되면서 이 분야에 많은 연구가 진행되어 왔다. 김창수 등(1999)은 정보기술(Information Technology)에 대한 지출이 은행산업(Banking Industry)의 경영성과에 영향을 미치는 지에 관하여 분석하였다. 분석결과는 은행의 규모나 전략(Wholesale vs Retail)에 따라 다소 다르겠지만, 지역적 은행보다 많은 고객과 전자적 banking시스템들을 가지고 있는 전국 규모의 은행에 있어

서 정보기술에 대한 지출이 경영성과를 증대시키는 것으로 나타났다[김창수, 허영빈, 1999]. 유극렬 등(1998)은 인터넷뱅킹의 동향을 분석하여 발전방향을 제시하고, 그것이 지급결제시스템(Transfer System) 및 금융시장 전반에 미치는 영향을 상세하게 분석하였다[유극렬, 조권익, 1998]. Dratva(1995)는 홈뱅킹(Home Banking) 서비스의 요구사항을 정의하고, 이것을 충족시키는 기술로써 인터넷과 WWW의 장점과 제한점을 논하고, WWW 기반의 홈뱅킹 서비스 구현전략을 제시하였다[Richard Dratva, 1995]. Kolletzki(1996)는 안전한 인터넷뱅킹을 위하여, 주문양식을 안전하게 교환하는 프라이버시 향상 메일(Privacy Enhanced Mail)이라는 프로토콜을 설계하고 구현하였다[Stephan Kolletzki, 1996]. Leong 등(1998)은 은행의 온라인 뱅킹(On-line Banking)을 위한 인터넷 어플리케이션의 설계 및 개발에 관하여 기술하였다. 이 논문에서는 온라인 뱅킹의 구현을 위하여 인터넷 기술에서 매우 잘 알려져 있는 2 가지 언어, Java 와 Perl 을 사용하였다[Leong, Srikanthan, Hura, 1998].

지금까지 인터넷뱅킹에 관한 연구들은 주로 인터넷뱅킹의 현황분석과 전망, 금융산업 및 은행산업에 미치는 영향, 보안기술 등에 관하여 주로 수행되어졌다. Leong 등(1998)은 온라인 뱅킹을 위한 인터넷 어플리케이션을 구축하였으나, Perl 언어를 트랜잭션 처리엔진에 사용함으로써 텍스트 처리능력이나 리포팅 기능은 좋지만 인터넷뱅킹에서 가장 중요한 요소중의 하나인 트랜잭션 처리에는 미흡했다.

본 논문에서는 기존 은행에서 인터넷뱅

킹 서비스를 제공하기 위한 어플리케이션의 설계 및 구축에 관하여 기술하고자 한다. 인터넷뱅킹을 위한 프로토타입의 구현을 위하여, 본 논문에서 사용자들의 트랜잭션 처리 및 보안을 위해서는 Oracle Application Server, 대부분의 사용자 인터페이스는 오라클의 PL/SQL(Procedural Language/SQL)과 자바 스크립트, 그리고 차트는 자바 애플릿을 이용했으며 은행 계정계 데이터베이스를 구축하기 위해 Oracle DBMS(Dataabase Management System)를 사용하였다.

2. 인터넷뱅킹

● 인터넷뱅킹의 배경

인터넷뱅킹이란 인터넷을 이용하여 잔액조회, 거래명세조회, 계좌이체, 예금 입·출금, 송금, 전자상거래 대금지급 등과 같은 결제업무를 자동화하고 금융서비스의 네트워크화를 구현하는 것을 의미한다.

금융업무 전산화의 발전방향을 살펴보면 다음과 같다. 1960년대부터 금융업무가 전산화 되기 시작하였고, 1980년대에는 PC 및 전화를 이용한 홈뱅킹(Home Banking)과 펌뱅킹(Firm Banking)이 등장하면서 네트워크를 이용한 금융서비스가 발전하였다. 1990년대는 사이버 은행이 등장하면서 인터넷을 이용한 금융업무가 시작되었다. 이러한 인터넷은 현재까지는 금융업무에서 고객의 점점중 거의 이용되지 않고 있으나, 10년 후에는 가장 중요한 비중을 차지할 것으로 예상된다[유극렬, 조권익, 1998].

<표 1> 10년후 고객접점의 중요도 예상순위

순위	항목
1	인터넷
2	PC
3	전화
4	Smart Card
5	비전통적 지점
6	ATM
7	쌍방향 TV
8	전통적 지점
9	화상전화

이러한 인터넷뱅킹이 발전하게 된 요인을 은행이 직면한 환경변화를 중심으로 살펴보자. 먼저, 정보통신기술의 발전이다. 현재 및 미래의 사회를 정보화시대라고 하고, 정보화를 주도하는 분야가 컴퓨터와 통신이다. 오늘날 컴퓨터 통신은 인터넷이 주도하고 있고, 이러한 정보통신기술의 발달은 은행 입장에서 볼 때 온라인 시스템이나 업무 전산화 등의 단순히 은행 내부적인 커뮤니케이션에만 영향을 미치는 것이 아니라 홈뱅킹/펌뱅킹 등의 고객과의 커뮤니케이션 채널을 변화시킴으로서 전통적인 은행의 물리적 장소 개념을 통째로 바꾸어가고 있다. 다음으로, 전자상거래의 발전이다. 인터넷이 크게 각광을 받으면서 단순한 부가가치 통신망으로서의 이용가치를 벗어나 기존의 물리적 상거래 장소 및 행위를 대체할 수 있는 비즈니스적 사업가치를 인정 받게 되었다. 전자상거래가 활성화될 경우 결제가 인터넷 상에서 이루어져야 하는데, 이를 전자결제(Electronic

Payment)라고 하며 은행계좌나 신용카드를 통한 결제 지시가 이루어지게 된다. 또한 고객들은 거래명세조회나 잔고조회 및 계좌이체 등의 서비스가 인터넷 상에서 동시에 이루어지기를 바랄 것이다. 따라서 전자상거래의 확산은 인터넷상에서 전자결제의 필요성과 함께 인터넷뱅킹 서비스의 필요성을 더욱 요구할 것이다. 마지막으로, 금융시장의 개혁이다. 최근 국제통화기금(IMF) 체제하에서 금융기관의 통폐합과 같은 구조조정과 금융시장의 개방화 등으로 은행의 경쟁력 제고를 위한 필사의 노력을 요구하고 있다[남상조, 1997].

요약하면, 인터넷뱅킹은 인터넷 서비스의 급속한 확산과 인터넷 사용자 수의 급증, 인터넷 상에서의 모든 은행 서비스의 제공이 가능하게 됨으로써 최근 주목을 받고 있다. 그리고, 새로운 형태의 홈뱅킹, 웹뱅킹 서비스의 요구가 증가함으로써, 은행들은 수동적인 뱅킹서비스로부터 능동적인 뱅킹서비스로 전환할 필요가 있고, 또한 서비스 환경 개선과 금융시장의 개방화에 따른 글로벌 비즈니스를 준비하기 위하여, 인터넷뱅킹은 피할 수 없는 선택이 되었다.

이러한 인터넷뱅킹의 장점은 은행의 입장에서 지점을 설치, 운영하는데 소요되는 인적, 물적 비용이 절감되며, 고객의 입장에서 은행에 오가는 시간과 노력을 없앨 수 있고 언제 어디서나 인터넷 접속이 가능한 곳에서 은행업무를 볼 수 있다는 것이다.

● 인터넷뱅킹의 현황

현재 인터넷에는 많은 은행들이 웹페이지를 구축하여 금융상품 소개 등의 서비스를 제공하고 있는데, 미국 은행들 중, 세계 최초의 사이버뱅크인 Security First Network Bank, 기존 은행들 중 Well Fargo, Bank of America, First Union, Atlanta Internet Bank, Mark Twain Bank 등 6개 은행은 잔액조회, 계좌이체, 대금지급, 전자화폐 사용 등의 인터넷뱅킹 서비스를 제공하고 있다. 또한, 일본 은행들도 미국 은행들 보다 일반 대중들을 위한 인터넷 뱅킹 서비스의 도입이 느리지만, 현재 3개의 은행, Sumitomo Bank, Sanwa Bank, 그리고 Asahi Bank 가 인터넷뱅킹 서비스를 제공하고 있다. Daiwa Bank 와 Daiichi Kangyo Bank 에서는 곧 인터넷뱅킹 서비스를 제공할 계획이다. 우리나라의 경우는 현재 국민은행이 한국통신과 함께 CO-LAN 을 이용해 전용 에플래이터나 인터넷을 통한 가상은행 서비스를 할 수 있도록 추진중이다. 대부분의 은행이 독자적인 망이나 PC 통신망을 통하여 홈뱅킹이나 폰뱅킹을 실시하고 있는 정도이다. 현재 부산은행과 조흥은행이 인터넷뱅킹을 위한 사이버은행을 마련해 놓고 있어 법적인 절차와 보안시스템이 확정된다면 언제든지 개시할 수 있는 체제를 갖추고 있는 등 국내 금융기관들의 인터넷 뱅킹 서비스도 곧 시작될 수 있을 것으로 판단된다[남상조, 1997; 유극렬, 조권익, 1998]

3. 기술적 고려사항

3.1 개발환경과 개발도구 선정

이 장에서는 인터넷뱅킹의 구현에 필요한 요소 기술에 초점을 맞춘다. 대부분의 시스템에서도 마찬가지로겠지만, 인터넷 상에서의 시스템 구현 시에는 규모의 확장성, 신뢰성, 트랜잭션 지원, SSL(Secure Socket Layer) 기반의 암호화 지원이 가능해야 한다. 시스템 구축시 실패하는 가장 흔한 예는 응용 프로그램의 이용자 수, 프로그램의 복잡도를 고려하지 않고 전개한 경우이다. CGI(Common Gateway Interface)를 이용하여 어플리케이션을 구현한 경우, 사용자 요구 1 회당 서버측에 1 개의 프로세스가 생성되게 된다. 이 경우 이용자 폭주에 따른 시스템 자원 부족으로 서버의 정상적인 운용을 기대하기 힘들게 된다. 또 은행의 계정계 시스템 거의 대부분은 데이터베이스 관리시스템을 이용하게 되는데, Web 과 DB 와의 연동이 원활하지 못한 경우 트랜잭션 처리, DB 보안, 시스템 자원 문제를 피할 수 없게 된다. 이러한 문제들은 이미 여러 차례 보고되고 있고 HTTP Listener 위에 벤더 고유의 인터페이스 계층을 없거나 자체적으로 TP(Transaction Processing) 모니터를 개발하는 등 유지 보수가 거의 불가능한 형태로 접근되기도 한다.

다음 장에서 설명되는 시스템 아키텍처에서 어플리케이션 계층에서는 견고한 시스템 구성을 위해 분산시스템 기반으로 요청(Request)의 균동 처리, 자동 복구 기능(Failure Recovery), Open API를 제공하고

또한 분산객체 모델에서 현재 산업표준으로 자리잡고 있는 CORBA(Common Object Request Broker Architecture), DCOM(Distributed Component Object Model) 등의 모델을 지원하며, IIOP(Internet Inter ORB Protocol) 등 ORB(Object Request Broker)간 표준 프로토콜의 지원, OTS(Object Transaction Service), Java, EJB(Enterprise Java Beans), SSL3, X.509 등 표준 보안의 규격지원이 가능한 Oracle Application Server 4.0.7(이하 OAS로 칭한다.)를 사용한다. 인터넷과 분산 객체 기술을 바탕으로 새로운 흐름에 이미 접어들고 있으며, 현재 여러 종류의 시스템에서 적용되고 있다. 이 흐름에서 가장 핵심적인 역할을 하는 것은 분산 객체 미들웨어이며 선택에 따라 개발 직후의 성능뿐 아니라 추후의 확장성, 이식성, 호환성을 보장 받게 될 것이다. 또, 다양한 클라이언트(FatPC, thinPC, NC ...)들을 별도의 부하 없이 일관된 형태로 지원할 수 있는지 여부도 중요한 기준이 된다.

3.2 인터넷 뱅킹에서의 보안

인터넷뱅킹의 특성상, HTTP(HyperText Transfer Protocol)을 사용하는데 이 프로토콜의 장점이자 단점은 접속상태가 계속유지되지 않는다는 것이다(Connectionless). 이로 인해 은행시스템에서 가장 중요시 되는 보안문제가 우선 해결되어야 한다. 현재 국내 대부분의 은행에서는 경쟁력 향상과 능동적인 banking서비스를 위해 대부분 인터넷뱅킹을 준비는 하고 있지만, 보안상의 염려로 인해 활성화가 늦어지고 있는 실정이다.

본 시스템에서는 뱅킹서비스의 구현에 중점을 두었기 때문에 보안문제는 다음과 같이 OAS에서 제공해 주는 기능을 사용한다.

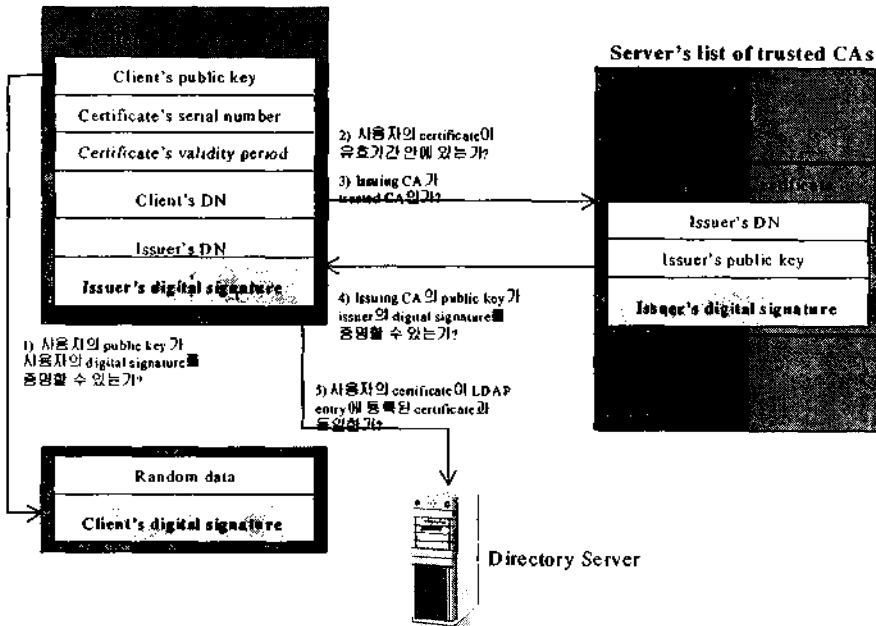
● Authentication & Restriction : 클라이언트가 정확한 Username 과 Password 를 입력해야 요청된 서비스가 이루어진다. Username 이 파일에 저장되는 방식이 Basic Scheme 과 Digest Scheme 이고, Oracle Database 에 저장되는 방식이 Basic Oracle(Database) Scheme 이다. 클라이언트가 입력한 Username 과 Password 가 Encryption 없이 서버로 전송되고 클라이언트와 내용을 상호 통신하는 것은 완전한 보안을 보장하지 못한다. SSL V3 상의 X.509 Digital Certificate, LDAP (Lightweight Directory Access Protocol) V3 를 지원하는 Directory Server, 그리고 OAS 를 연동하면 위와 같은 상황을 피하면

서 클라이언트의 Authentication 을 이룰 수 있고, Single Sign-on 을 구현할 수 있다. 한편, 클라이언트의 IP 주소나 도메인 이름으로 접근을 제한 할 수 있다.

● Encryption : CA(Certifying Authority)나 인증서로부터 인증을 받아서 Oracle Web Listener 가 SSL V3 를 사용할 수 있도록 지정한다. 데이터는 Public Key 에 의해 Encryption 되고 Private Key 에 의해서만 Decryption 된다. (Public-Key Encryption)

● Integrity : SSL V3 는 MD5 message-digest 알고리즘을 사용함으로써 전송된 데이터가 변경되었는지를 판단한다.

<그림 1>은 Client Authentication 과 LDAP Server 를 보여주고 있으며 그 처리과정은 다음과 같다.



<그림 1> Client Authentication 과 LDAP Server

① 사용자의 Public Key 를 가지고 사용자의 Digital Signature 가 유효한지를 판단한다. (사용자가 데이터를 전송하기 전, 그 데이터에 대해 해쉬 알고리즘을 사용한다. 그 결과인 단방향(One-Way) 해쉬를 사용자의 Private Key 로 Encryption 한다. 이 Encryption 된 단방향(One-Way) 해쉬와 해쉬 알고리즘을 Digital Signature 라고 한다. 데이터뿐 아니라 이 Digital Signature 도 전송된다. 서버는 사용자의 Public Key 를 가지고 사용자의 Digital Signature 를 Decryption 하여 단방향(One-Way) 함수를 생성하고, 또 전송된 데이터에 대해 같은 해쉬 알고리즘을 사용하여 다른 단방향(One-Way) 함수를 생성한다. 이 두 단방향(One-Way) 해쉬가 일치하는지를 체크해서, 데이터가 전송도중 변경되었는지를 판단하게 된다. ② 사용자의 Certificate 이 유효기간 안에 있는지를 검사한다.

③ 클라이언트에게 Certificate 을 부여한 CA 가 서버에 등록된 CA 인지 검사한다.

④ 서버에 등록된 CA 의 Public Key 를 사용해서 전송된 Certificate 안의 Issuer(CA)'s Digital Signature 를 검사한다. 여기까지 정상적으로 이루어져야 사용자의 Certificate 을 유효하다고 판단한다.

⑤ 전송된 사용자의 Certificate 과 Directory Server 에 등록된 그 사용자의 Certificate 가 동일한지를 검사한다.

⑥ 서버는 카트리지 서비스에 대해 접근 가능한 사용자/그룹 정보 등을 접근제한목록(Access Control List: ACL)에 정의해 놓는다. 이 ACL 과 사용자의 DN(Distinguished Name)을 이용해, 카트리지 서비스를 요청한 사용자가 실제로 그 서비스를 받을 수 있는지를 검사한다.

4. 인터넷 뱅킹시스템의 구현

4.1 시스템 구현 범위

우리나라 현 상황에서는 실명확인 요구와 같은 여러가지 법적제약 때문에 실제 사이버 은행의 계좌를 따로 개설하여 은행 계좌간의 이체를 인터넷을 통해 직접한다든가 아니면 미국의 SFNB(Secure First Network Bank)처럼 물리적 영업점이 존재하지 않는 완전한 인터넷 은행은 조금은 시기상조인 것 같고, 본 시스템에서는 기존 은행이 인터넷 상에 뱅킹 서비스 업무를 부가하는 개념에서 구현하였다. 또한 시스템 구현범위도 기존의 PC 뱅킹의 기능을 통합하면서 인터넷의 장점을 살릴 수 있도록 설계하였다. 세부 구현 내용은 아래의 <표 2>를 참고 하기 바란다.

<표 2> 시스템 세부 구현내용

회원관리	회원등록, 회원정보 변경
계좌정보	계좌별 내역, 최근 거래내역, 업무처리별 조회, 평균잔액, 그래프
이체	일반이체, 예약이체, 예약서비스(Pending Service)
대출	대출상품정보, 대출신청, 대출상담, 대출상환 내역
서비스	환율조회, 승차권예매(URL Link), 신상품 메일 전송
상담투자	예금상담을 채팅으로 구현
게시판	Q & A
내역신청	개인별 계좌내역, 통장거래내역 신청
상담신청	해당담당자에게 상담신청

4.2 업무분석

<표 2>의 시스템세부 구현내용 중에서 회원관리, 계좌정보, 이체, 대출, 서비스에 대한 상세한 업무분석은 다음과 같다.

● 회원관리

기본적으로 로그인 화면에서 Guest로 접속한다. 접속한 후 해당 은행에 계좌가 있

는 경우에는 회원가입을 할 수 있다. 비회원인 경우는 신상품 소개, 경매, 기념 주화 판매, 상품 상담 코너만 이용 가능하다.

● 계좌 정보

회원가입을 한 후 회원 ID로 접속하면 최초의 화면에서는 회원이 해당 은행에 가

지고 있는 모든 계좌가 출력되며 계좌별 잔액조회, 사용내역 조회, 평균잔액, 거래별 내역을 그래프로 출력한다.

● 이체

일반이체(즉시이체)를 통해서 일반적인 이체를 할 수가 있고, 일종의 예약이체인 예약서비스를 통해서 예약이체 항목을 수정/삭제/등록하고 항목을 스케줄링하며, 이체보류 항목에 대한 내용을 메일로 전송해서 즉시 확인 가능하게 한다.

● 대출

대출 이자를 조회하고 대출신청서를 작성하며, 개인에 대한 대출내역을 조회해 준다. 또한 대출에 대한 상담도 하게 된다.

● 서비스

환율정보 조회, 기념주화 판매, 승차권 예매(철도청), 복권판매, 신상품 정보를 등

특회원에 한해서 HTML로 가입 폼을 작성하여 메일로 전송한다. (고객은

자신의 메일 시스템에서 바로 확인 후 가입도 할수 있다). 예금액/예치기간/이자구분/가입대상 등을 입력 받아 예금 이율을 계산한다. 고객의 카드결제금액/내역, 거래내역을 월별로 메일로 전송한다. 고객은 기간별/카드별/계좌별 거래내역에 대한 메일 전송 서비스를 신청할 수 있다.

4.3 사용 제품

본 논문에서 제시한 인터넷뱅킹 서비스의 어플리케이션을 구축하기 위하여, 서버로는 SUN Ultra 1을, 데이터베이스로는 Oracle7 RDBMS를, Web 서버로는 Oracle

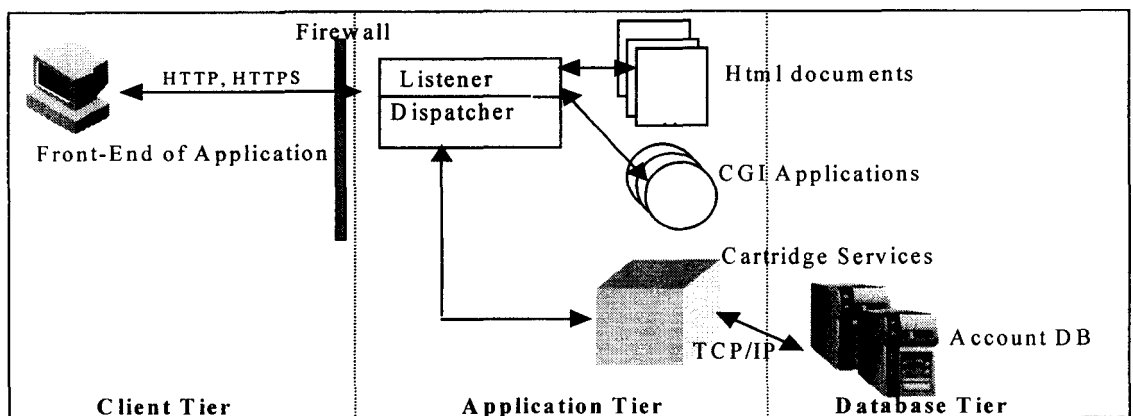
Application Server 4.0.7 , 개발언어로는 Java와 Java Script 그리고 PL/SQL을 사용하였다. 세부사항은 아래 표와 같다.

<표 3> 사용 제품

SUN Ultra 1	Oracle7 Enterprise Server 7.3.4
- Disk: 4G	Oracle Application Server 4.0.7
- Memory: 128M	- PL/SQL cartridge
- Cpu: 1개	- PERL cartridge
	- Live!ltd cartridge
	Oracle Context option
	Oracle Internet Commerce Server 1.1
	Oracle Jdeveloper 1.0
	Netscape Communicator 4.05
	Java script

4.4 시스템 아키텍처

<그림 2>는 본 시스템에서 구현하고자 하는 시스템 아키텍처를 보이고 있다.



<그림 2> 시스템 아키텍처

크게 3 개의 계층으로 구성되어 있다. 첫 번째 계층인 Client Tier 는 Web Browser 를 탑재한 PC 를 의미한다. Application Tier 는 Web Server 의 기능을 포함하는 OAS 를 의미한다. 각 구성요소에 대해서는 개발환경 부분에서 논한다. Database Tier 는 은행의 계정계 시스템을 의미한다. 계정계 시스템이 방대하기 때문에 본 시스템 구축에서는 범위를 고객센터 서비스 부분에 중점을 둔다

4.5 개발 도구

OAS 는 사용자의 요구나 개발 수준에 따라 다양한 개발 방법을 제공한다. 기존의 HTTP/CGI 방법을 지원함은 물론 Java, CORBA, Scripting 에 의한 개발 방법을 지원한다. 각 방법에서 이용되는 언어 등은 차이가 있지만 대부분의 관리기능은 일반적으로 지원되어 용이한 운용 환경을 제공한다.

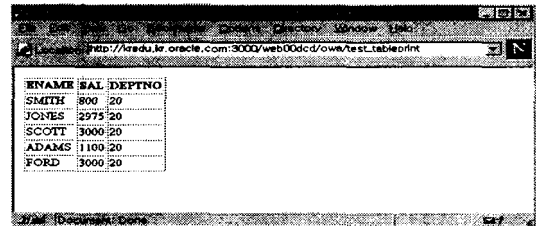
- PL/SQL 카트리지

PL/SQL 카트리지는 사용자의 브라우저에서 Oracle 데이터베이스에 저장된 PL/SQL 프로시저를 호출할 수 있는 환경을 제공한다. 저장프로시저(Stored Procedure)는 데이터베이스에서 필요한 데이터를 추출하고, 그 데이터를 포함하는 동적인 HTML 페이지를 생성한다. Common Schema 에 인스톨된 PL/SQL Web Toolkit 을 이용해서 사

용자의 요청으로부터 정보를 추출하고 Content Type 이나 쿠키(Cookie)와 같은 HTTP 헤더(Header)를 정의하고 HTML 태그를 생성한다. 즉, HTML 태그와 PL/SQL Web Toolkit 에 의해 생성된 프로시저는 1 대 1 로 대응한다. 다음의 예는 기본적으로 제공되는 여러 패키지중 tableprint 라는 프로시저를 사용해서 해당되는 부서의 팀원들을 출력하는 예이다.

```
CREATE PROCEDURE test_tableprint
IS
    dummy boolean;
    v_deptno emp.deptno%TYPE = 20;
BEGIN
    dummy := owa_util.tableprint('emp','border=1',
        OWA_UTIL.HTML_TABLE,'ename, sal, deptno',
        'where deptno = '||v_deptno);
END;
```

위와 같이 작성한후 컴파일하게 되면 데이터베이스 내부에 Test_Tableprint 라는 프로시저로 저장이 된다. 웹브라우저 상에서 해당 프로시저를 호출하면 아래와 같은 결과를 얻게 된다.



<그림 3> 실행 결과

위의 예에서와 같이 PL/SQL 카트리지를

이용하는 것이 구현상의 용이함, 보안 등의 부분에서 다른 개발도구보다는 훌륭하다는 것을 알수 있다. 인터넷 뱅킹 시스템 구현 시 대부분의 기능들이 이 카트리지를 이용해서 개발했다. 계좌 이체의 경우는 반드시 트랜잭션이 지원되어야 한다. 현재 CGI를 이용해서는 트랜잭션 처리가 불가능하며 이를 위해 대부분 자바언어를 사용하게 된다. 하지만 본 시스템에서는 PL/SQL 카트리지를 이용해 가장 간단하게 트랜잭션기능을 구현할 수 있다. Transaction Begin, Commit, Rollback, Boundary를 URL로 지정함으로써 X/Open의 XA model에 기초한 트랜잭션 서비스를 사용할 수 있다. Transaction Begin에 해당하는 URL을 요청함으로써 트랜잭션은 시작되고 Transaction Commit이나 Rollback에 해당하는 URL을 요청하면 트랜잭션이 종료되는 것이다.

● 자바 애플릿(Java Applet)

본 시스템에서는 구현 범위 중에 계좌별 평균잔액, 거래현황, 통화별 환율 변동량, 상담 등을 자바 애플릿을 사용해서 개발하였다. 자바 애플릿을 만들기 위해 JDK(Java Development Toolkit) 1.1.2를 사용하였기 때문에 Java 1.1.2가 지원 가능한 웹 브라우저를 사용해야 한다. 참고로 여기서는 Netscape Communicator 4.05를 사용하였다.

OAS는 다수의 HTTP 서버와 함께 동작하며 HTTP 서버를 OAS에 요청을 던지는

Listener로 배치하고 있다. 요청의 내용이 웹서버상의 정적인 HTML 파일이면 해당되는 파일을 돌려주고 CGI Application이라면 프로세스를 하나 생성해서 요청을 처리하게 된다. 하지만 이 논문에 구현된 시스템은 CGI를 사용하지 않고 OAS를 구성하는 여러가지의 카트리지에서 PL/SQL 카트리지와 PERL(Practical Extraction and Report Language) 카트리지를 사용한다. 따라서 요청이 HTML이나 CGI Application을 요구하지 않으면 적절한 카트리지를 통해 해당 내용을 처리해서 클라이언트로 넘겨주게 된다.

4.6 데이터베이스 설계

실제 은행의 계정계 시스템의 데이터베이스는 방대한 규모인 관계로 본 시스템에서는 고객 서비스 구현 부분에 중심을 두어서 테이블의 규모를 최소화 하였다.

<그림 4>는 인터넷 뱅킹을 위한 어플리케이션 구현에 필요한 ERD(Entity Relationship Diagram)을 보이고 있다. <그림 4>에서 통장거래내역, 통장, 고객, 예약이체 항목의 4개 개체(Entity)를 관계형 데이터베이스의 테이블로 맵핑하면 아래와 같다.

- 테이블명: 통장(통장에 관한정보)

<표 6> 통장 테이블

Column Name	Type	Meaning
ACCID	Varchar2(20)	계좌번호
START_DATE	Date	개설일자
DESTROY_DATE	Date	폐기일자
CUST_ID	Varchar2(08)	고객번호
PASSWD	Varchar2(20)	비밀번호
SAVE_GUBUN	Varchar2(01)	통장구분
BALANCE	Number(14)	잔액
CARD_ISSUE	Varchar2(01)	카드발급여부
CARD_NO	Varchar2(20)	카드번호
LONE_GUBUN	Varchar2(01)	대출가능여부
CONF_EMPNO	Varchar2(08)	발급자

- 테이블명: 통장거래내역(고객의 통장별 거래내역 정보관리)

<표 7> 통장거래내역 테이블

Column Name	Type	Meaning
ACC_TRADE_ID	Number	거래내역 Seq
CUST_ID	Varchar2(08)	고객 ID
ACCID	Varchar2(20)	거래계좌번호
TRADE_GUBUN	Varchar2(01)	거래구분
TRADE_ACCID	Varchar2(20)	거래계좌번호
RECEIPTS_ACCID	Varchar2(20)	입금계좌
DEPLOYMENT_ACCID	Varchar2(20)	출금계좌
TRADE_DATE	Date	거래일시
TRADE_AMT	Number(14)	거래금액
BALANCE	Number(14)	잔액
BANK_ID1	Varchar2(05)	이체되는 은행
BANK_ID2	Varchar2(05)	이체받는 은행

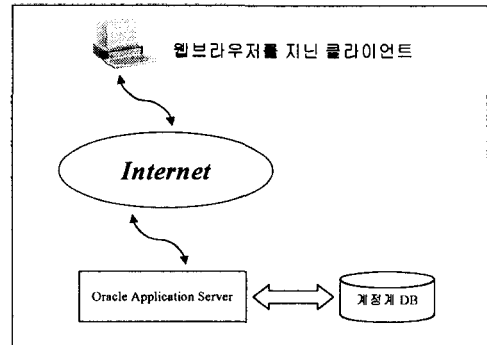
- 테이블명: 예약이체항목(예약이체항목 정보)

<표 8> 예약이체항목 테이블

Column Name	Type	Meaning
ITEM_ID	NUMBER	이체항목 ID
NAME	VARCHAR2(20)	이체항목명
CUST_ID	VARCHAR2(08)	고객 ID
ACCID	VARCHAR2(20)	이체계좌번호
UNIT	Varchar2(10)	이체주기

5. 인터넷뱅킹을 위한 웹사이트 설계

웹사이트는 <그림 5>에서 처럼 UNIX 서버에서 구현되었다. 대부분의 사용자인터페이스는 OAS의 PL/SQL 카트리지를 사용해서 구현했고 사용자들의 효율적인 분석을 위해서 차트가 필요한 부분은 자바 애플릿을 사용했다. 또한, 계정개 시스템을 구축하기 위해 오라클 DBMS가 UNIX 서버에 탑재되어 있다.

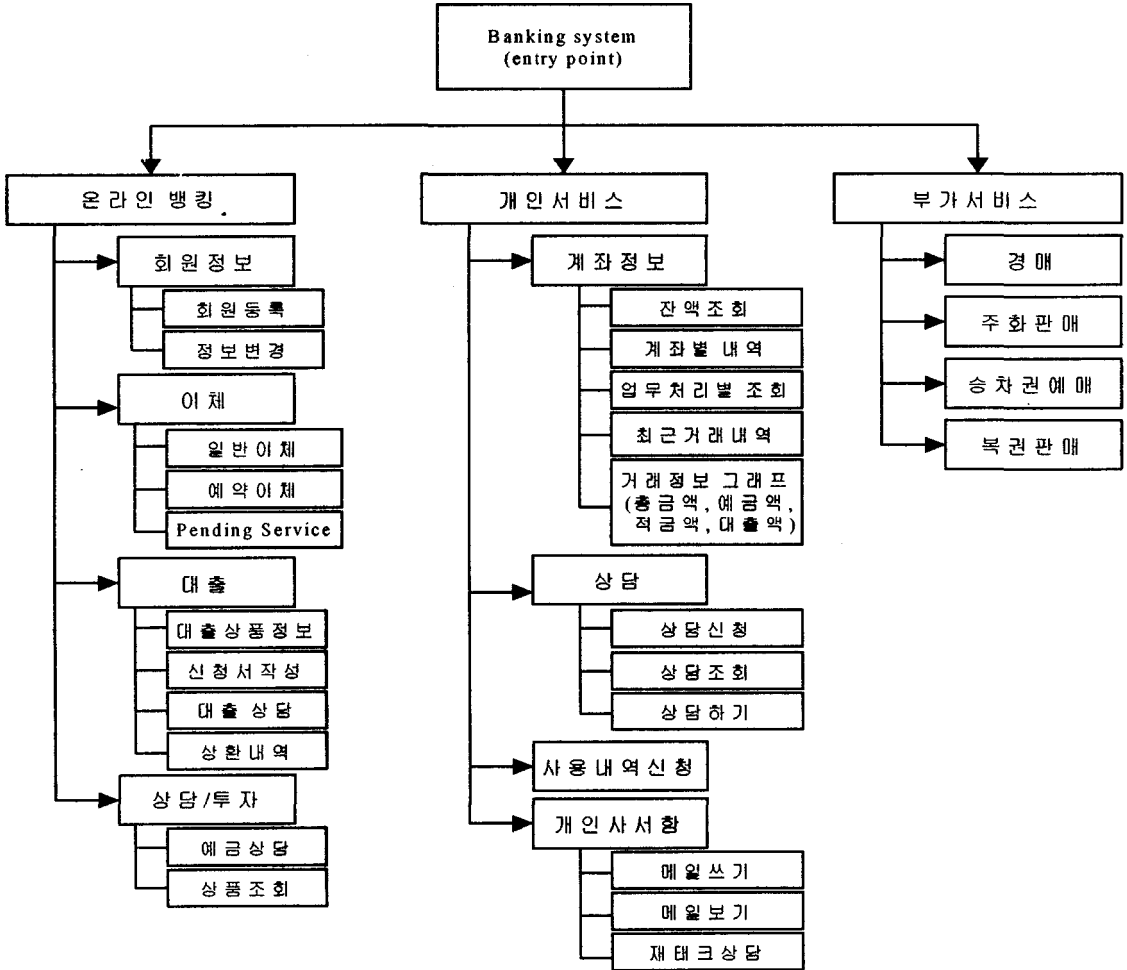


<그림 5> 웹사이트의 시스템 구성도

사용자가 입력한 데이터의 검증(Verification)은 대부분 네트워크를 통해 서버에 보내져 서버에서 처리해서 오류가 발생하면 다시 네트워크를 통해서 클라이언트의 웹브라우저에 보여지게 되지만 이 경우 네트워크 부하를 줄 뿐만 아니라 응답속도도 감소되고 서버에 부하를 주게 되기 때문에 데이터의 검증은 자바스크립트를 이용해서 처리를 했다. 사용자들의 효율적인 탐색을

위해서 기본적으로 3 개의 프레임으로 나눈 후 메뉴부분은 자바스크립트를 사용해서 사용자 인터페이스를 강화 시켰다. 웹사이트는 크게 <그림 6>과 같이 온라인

뱅킹, 개인서비스, 부가서비스로 나누어지며 탐색에 들어가면 다시 세부적인 메뉴로 나누어지게 된다. 각각의 메뉴는 PL/SQL 을 이용한 프로시저로 작성되어 진다.



<그림 6> 웹사이트의 메뉴 구성도

아래의 <표 9>는 <그림 6>의 메뉴를 구성하는 오라클 데이터베이스의 오브젝트인 프로시저와의 관계를 나타내고 있다.

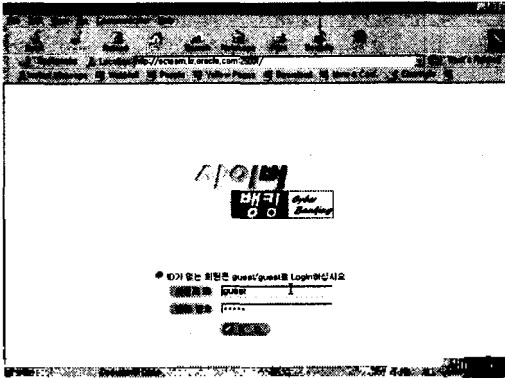
<표 9> 메뉴구성 및 해당 프로시저

주메뉴(Level1)	부메뉴(Level2)	하위메뉴(Level3)	프로시저
온라인뱅킹	회원정보	회원 등록	Member reg
		회원 정보 변경	Member update
	이체	일반 이체	Normal trans
		예약 이체	Pre trans
		예약 서비스	Pend ser
	대출	대출 상품 정보	Loan product info
		신청서 작성	Loan request
		대출 상담	Loan consul
		상환 내역	Loan repayment
	상담/투자	예금 상담	Deposit consul
상품 조회		Prod search	
개인서비스	계좌정보	잔액 조회	Balance search
		계좌별 내역	List per account
		업무 처리별 조회	List per kinds
		최근 거래내역	Exchange latest
		거래정보 그래프	Exchange graph
	상담	상담 신청	Consult request
		상담 조회	Consult search
		상담 하기	Consult
	사용내역신청		Usage list request
	개인사서함	메일쓰기	Mail write
메일보기		Mail read	
재태크상담결과		Consult result	
부가서비스	경매		Action
	주화판매		Coin sell
	복권판매		Tiket sell
	승차권예매		Tiket reserve

5.1 주요 화면 설계

● 로그인 화면: 기본적으로 회원 가입을 위해서 Guest 로 접속한 후 온-라인서비스

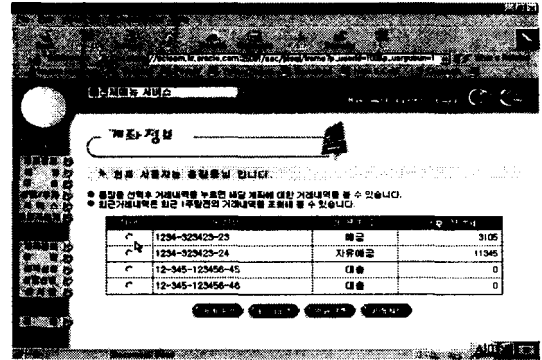
를 사용하기 위해서 회원가입을 한다. 회원 가입의 원칙은 해당은행에 계좌가 있을 경우로 제한 한다.



<그림 7> 로그인 화면

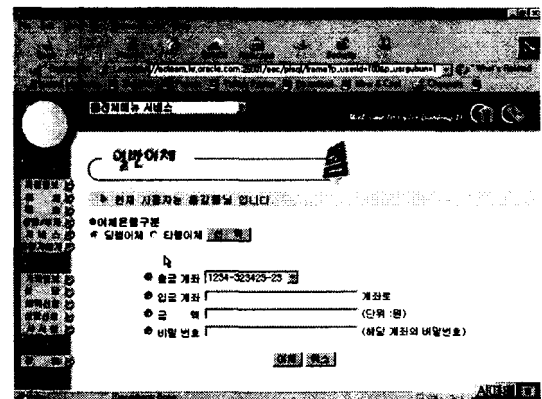
● 초기 화면: 회원 가입 후 다시 로그인을 하면 사용자 인증을 거친 후 사용자가 해당 은행에 있는 모든 계좌가 나타나게 되며 자주 사용하는 서비스가 나타나게 된다. 또한 클라이언트 정보를 저장하기 위해 쿠키를 사용한다. 이 화면에서 계좌번호를 선택 후 거래내역을 클릭하게 되면 날짜를 지정해서 해당 계좌의 거래내역을 조회할 수가 있고 최근내역은 일주일 간의 거래내역을 그리고 평균잔액, 거래차트는 자바 애플릿을 사용해서 효율적인 인터페이스를 제공한다. 현재 이런 서비스는 기존의 PC뱅킹에서는 불편한 사용자 인터페이스로 인해 구현하기가 힘든 부분이다.

● 이체 화면: 이체는 일반이체, 예약이체, 예약서비스로 나누어지며 이체는 특히 트랜잭션 처리가 중요하다. 예를 들면 갑의 계좌에서 을의 계좌로 1000 원을 이체할 경우 갑의 계좌를 변경(Update) 하고 을의 계좌를 변경하는 일련의 절차가 하나의



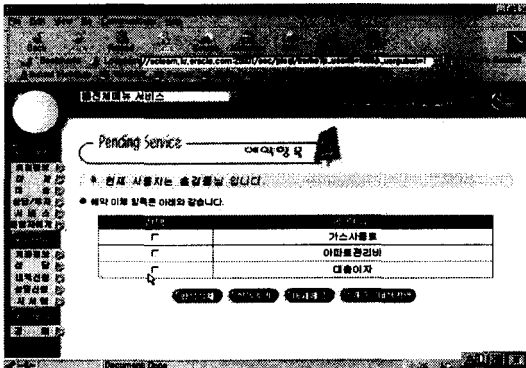
<그림 8> 초기화면

트랜잭션으로 처리가 되어야 한다. 만약 갑의 계좌를 변경하고 어떤 장애로 인해 을의 계좌를 변경할 수 없다면 갑의 계좌를 변경한 부분은 취소(Rollback)되어야 한다. 만약 취소가 되지 않은 다면 치명적인 문제를 야기할 것이다. 전통적인 CGI로는 해결하기 어렵고 본 논문에서는 PL/SQL을 이용해서 처리를 했다.



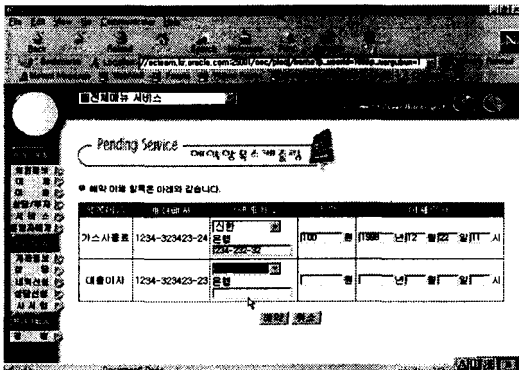
<그림 9> 이체 화면

● 예약서비스(Pending Service) 화면: 일종의 예약이체와 비슷한 성격의 서비스로 웹의 장점을 이용하면서 또한 효율적인 banking 서비스를 제공해 줄 수가 있다. <그림 10>에서 처럼 정기적으로 납부를 해야 하는 항목을 추가해서 이체단위를 정해서 정해진 날에 반복적인 이체를 할 수가 있다. 공과금 납부 등은 이런 서비스를 이용하게 되면 쉽게 자동이체를 신청할 수가 있다.



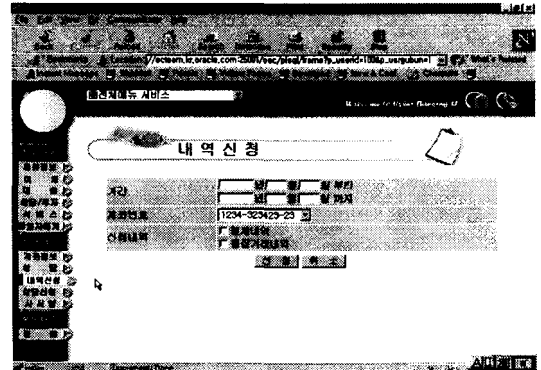
<그림 10> 예약서비스 화면

● 예약서비스 예약 화면: 항목에 대해 스케줄링을 하는 화면이다.



<그림 11>예약서비스 예약 화면

● 사용내역 메일전송 신청 화면: 특정 계좌에 대해 기간을 정해서 거래내역등과 같은 서비스를 전자메일을 통해서 받아 볼 수 있다.



<그림 12> 사용내역 메일전송 화면

6. 결론

인터넷 및 전자상거래의 급속한 확산과 전자금융 서비스에 대한 고객의 요구에 부응하기 위하여, 은행들에게 인터넷뱅킹은 피할 수 없는 선택이 되었다. 본 논문은 기존 은행들이 이러한 인터넷뱅킹 서비스를 제공하기 위한 어플리케이션을 설계하고 개발한 사례를 제시한 것이다. 인터넷뱅킹을 위한 웹사이트와 웹페이지의 구축에는 확장성, 신뢰성, 트랜잭션 처리, 보안을 위하여 Oracle DBMS, Oracle Application Server(OAS), PL/SQL, 그리고 Java Applet을 사용하였다.

본 논문에서 제시된 인터넷뱅킹을 위한

어플리케이션은 기존의 PC뱅킹이나 폰뱅킹에서 서비스하기 어려웠던 부분, 예를들면 사용상의 어려움, One-Stop 서비스 등을 웹의 장점을 최대한 이용해서 사용자들에게

능동적이고 안전한 뱅킹서비스를 제공할 수가 있을 것으로 판단된다.

참고문헌

- [김창수, 허영빈, 1999] 김창수, 허영빈, “소매금융 및 대형화 전략 도구로서의 정보기술과 은행의 경영성과”, *경영정보학연구*, 제 9 권 제 1 호, 1999.
- [남상조, 1997] 남상조, “사이버 은행”, *Computer World*, 1997.
- [유극렬, 조권익, 1998] 유극렬, 조권익, “인터넷뱅킹의 현황과 금융환경 변화”, *The Journal of Korea Institute of CALS/EC*, Vol. 3, No. 1, June, 1998.
- [Leong, Srikanthan, Hura, 1999] Leong S.K., Srikanthan T., Hura G.S., “An Internet application for on-line banking”, *Computer Communications*, Vol 20, 1998, pp.1534-1540.
- [Richard Dratva, 1995] Dratva R., “WWW-based home banking services in Switzerland: a case study”, *Computer Networks and ISDN Systems*, Vol. 28, 1995, pp 199-208.
- [Stephan Kolletzki, 1996] Kolletzki S., “Secure Internet banking with Privacy Enhanced Mail – A protocol for reliable exchange of secured order forms”, *Computer Networks and ISDN Systems*, Vol. 28, 1996, pp 1891-1899.

저자 소개

남철기 (cknam@kr.oracle.com)

울산대학교 전자계산학과 학사

울산대학교 정보통신대학원 정보통신공학 석사

현재 한국오라클(주) 근무

관심분야: 데이터베이스, 정보검색, 인터넷/인트라넷, EC, KMS

장길상 (gsjang@kr.oracle.com)

울산대학교 산업공학과 학사

한국과학기술원(KAIST) 산업공학 석사

한국과학기술원(KAIST) 경영정보공학 박사

현재 한국오라클(주) 근무

관심분야: 데이터베이스, 정보공학, 생산정보시스템, ERP, EC, KMS