2015학년도 정보보호학과 졸업작품 보고서

범용 원격 관리 시스템

팀　　명 : C J S (Crazy Java Study)

지도 교수 :　　이병천　교수님

조　　장 :　　한　　솔

안　명　진

양　윤　태

안　병　훈

2015. 6

중부대학교 정보보호학과

목 록

1. 서 론

1.1 연구 배경

컴퓨터의 보급이 증가함에 따라 많은 기관이나 학교 등에서 PC를 이용한 교육이나 업무들이 증가하고 있다. 그러나 다수의 분산된 PC를 사용하여 운영하는 경우 개별 컴퓨터를 직접 가서 확인/관리가 불편하고 시간적인 소비가 많으므로 운영관리나 교육의 효율이 저하된다. 또한 관리자의 효율이 저하될 뿐만 아니라 PC를 활용한 교육이나 업무 시에 피교육자들이나 업무자들이 카톡이나 개인용 프로그램을 무분별하게 사용하는 행위를 제어하기가 힘들고, 그로 인해 발생할 수 있는 교육이나 업무 분위기 훼손, 교육 효과 감소 등의 문제점들이 발생하고 있다.

1.2 연구 필요성

연구에 앞서 몇 가지의 문제와 연구의 필요성을 제시한다. 첫 번째로 아이디/패스워드의 보안성 문제이다. 대부분 사람들에게 패스워드는 '기억하기 쉬운것' 이 첫 번째 원칙이다. 안전성보다는 편리하게 사용될 수 있는 패스워드, 최근에는 의무적으로 대소문자와 특수문자를 의무적으로 추가하는 곳이 증가하고 있지만, 아직은 부족할 따름이다.
온라인 상에서 특정 아이디와 그에 부합하는 패스워드를 입력할 경우, 본인임을 인정하는 아이디/패스워드 방식은 다른 본인 인증수단에 비해 시스템 구축이 비용이 낮고, 또 무엇보다 사용자가 편리하게 사용할 수 있다는 장점을 갖고 있기 때문이다. 그러나 이 아이디/패스워드 인증수단의 안전성 및 보안성ㅇ든 패스워드에 의해 좌우되기 때문에 패스워드만 알아낸다면, 주민등록번호, 주소 등 단순 개인정보뿐만 아니라, 이메일의 내용이나 금용정보까지도 해커들에게 노출될 수 있다. 특히, 적게는 1~2개의 아이디와 패스워드로 모든 웹 사이트에 적용하는 사용자들의 성향을 생각한다면, 더욱더 위험하다.

패스워드 크래킹이라고 들어봤을 것이다. 패스워드 크래킹으로 아이디와 패스워드의 일대일 매칭을 시도할 경우, 복잡하게 보이는 패스워드 일지라도 손쉽게 크래킹을 당할 수 있다. 한가지 예로, 제3자가 특정인의 아이디를 알고 있을 때 4자리의 패스워드를 알아내는데 필요한 경우의 수는 10,000개 (10^4)로 10,000의 시도만으로 패스워드를 알아낼 수 있으며, 영문자와 숫자를 조합한 4자리 패스워드 역시 1,678,616번일 대입하면 알 수 있다는 결과가 나온다. 아래의 < 표 1 >과 같다.

입력 문자	7자리	8자리
영문 소문자(26문자)	45분	20시간
영문소문자+숫자(36문자)	8시간	13일
영문 대·소문자+숫자(62문자)	25일	4년 6개월
영문 대·소문자+숫자+특수문자(94문자)	437일	114년

< 표 1 : 패스워드 크래킹 >

이에 따라 우리 팀(크자스)은 대체 인증방안으로 인증서를 선택하였고 아이디/패스워드 인증과 동시에 구현하였다.

두 번째로 PC를 사용하는 강의 시 피교육자들의 개인 프로그램이나 게임등을 무분별하게 사용하는 것이다. 우리 팀이 주제를 선정하기 위해 큰 영향을 준 계기이기도 하다. 컴퓨터를 사용하는 실습실에서 수업을 받는 학생이라면 카카오톡 등 개인 프로그램을 사용한 적이 한번쯤은 있을 것이라고 생각한다. 그러할 경우 수업분위기 훼손과 강의 효과가 감소 할 것이다. 만약 상황이 시험을 보고 있는 중이라면 그 피해는 배가 될 것이며, 상당수의 학생들의 불만을 초래할 가능성이 존재한다. 따라서 “범용 원격관리시스템”을 개발함으로 써 카카오톡 등 개인용 프로그램을 무분별한 활용을 억제하고 강의 분위기를 향상시킬 수 있다.

마지막으로 다수의 PC를 원격 관리하는 시스템을 사용하는 기업이나 단체에서 각각의 PC의 상태를 점검하기에는 어려움이 따른다. 여러 PC들이 존재하고 이를 관리해야 하는 입장이라면 어떨까? 그러면 우리는 각각의 PC가 이상이 있을 때 그 PC가 있는 곳으로 가서 이상 유무를 판단하여 처리해야 할 것이다. 이는 운영 관리의 효율성이 저하되고 금전적 손실도 따를 것이다. 만약 제택근무라면 더 큰 피해를 입을 것이다. 이러한 문제를 해결 하기위해 우리 프로그램 “범용 원격관리시스템”을 활용한다면 원격으로 개별 PC의 이상여부를 점검하므로 이점은 상당할 것이다.

1.3 연구 목적 및 주제선정

기업이나 학교 등 많은 인원들이 개별 PC를 사용하는 경우 개별 PC를 모니터링 하는 원격관리시스템을 개발, 업무나 교육, 운영관리의 효율성을 높이고 개인용 프로그램 등의 사용을 억제한다. 또한 프로그램을 사용할 때 보안성을 높이기 위해 인증서를 이용한 전자서명을 통해 보다 안전하게 로그인할 수 있는 프로그램을 구현한다. 이를 토대로 본 프로젝트의 주제를 “범용 원격 관리 시스템”으로 정하였다.

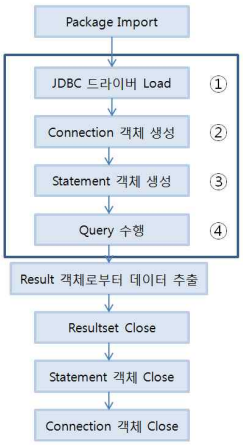
2. 관련 연구

2.1 JDBC

JDBC(Java Database Connectivity)는 자바로 작성된 프로그램을, 일반 데이터베이스에 연결하기 위한 응용프로그램 인터페이스 규격이다. 이 응용프로그램 인터페이스는 데이터베이스 관리시스템에 넘겨질 SQL 형태의 데이터베이스 접근요구 문장을, 각 시스템에 맞도록 바꾸어준다. 처리 결과도, 이와 비슷한 인터페이스를 통해 얻게 된다. JDBC는 ODBC(Open Database Connectivity)와 아주 유사해서, 조그만 연결 프로그램만 있으면, ODBC 인터페이스를 통해 데이터베이스에 연결하는 JDBC 인터페이스를 사용할 수 있다. 예를 들어, 다수의 운영체계 플랫폼 상에 있는 많은 데이터베이스 제품들을 연결하도록 설계된 프로그램을 작성할 수 있으며, JDBC 문장을 사용한 프로그램은 심지어, 윈도우95에서 운영되는 마이크로소프트 액세스 데이터베이스에 접근하는 것도 가능할 것이다.

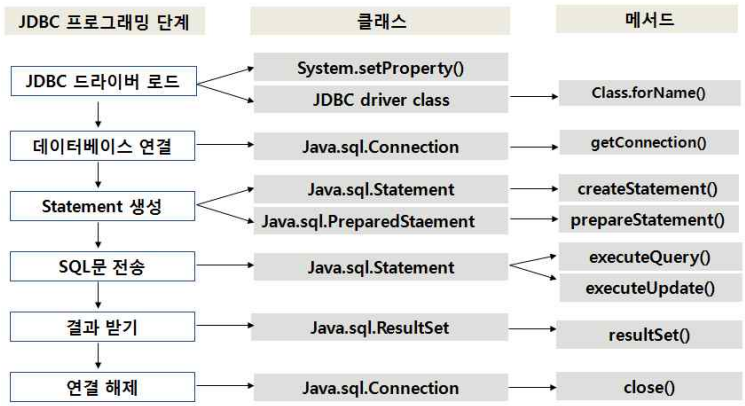
JDBC는 실제로는 두 계층의 인터페이스로 구성되어 있다. 주 인터페이스 외에도 JDBC "manager"에서 나온 API 가 있는데, 이것의 역할은 개별 데이터베이스 제품의 드라이버들과 차례대로 통신을 하는 것이다. 이때, 만약 필요하다면 JDBC-ODBC bridge와, 그리고 자바 프로그램이 원격 데이터베이스를 액세스하기 위해 네트워크 환경에서 실행되고 있다면 JDBC 네트워크 드라이버 등과의 통신도 수행한다.

JDBC가 원격 데이터베이스에 접근할 경우에는, 인터넷 파일 주소 구조의 강점을 이용하는데, 파일이름이 웹페이지 주소(URL) 체계와 아주 유사하게 보인다. JDBC는, 프로그래머가 SQL 요구를 만드는데 사용할, 일련의 객체지향 프로그램의 클래스들을 정의하고 있으며, 별도의 추가 클래스 모음집에 JDBC 드라이버 API가 기술되어 있다. 자바 데이터 형식에 대응된, 일반 SQL 데이터 형식들 대부분이 지원된다. JDBC는 특이한 실행을 위한 처리 요구와 함께, 트랜잭션을 성공적으로 마치는 commit 이나, 또는 현재의 트랜잭션을 취소하는 rollback 기능 등을 제공한다.



<그림 1 : JDBC 사용절차>

<그림 1>을 참고하면 JDBC를 사용하기 위해 Package를 import 시킨다. ① 다음으로 JDBC 드라이버를 Load한 후 ② Connection 객체를 생성하여 Ip 주소 및 port번호를 입력한다. 입력 방법은 <소스 1>을 참고하자. ③ JDBC를 이용하여 SQL문을 발행하기 위해서는 Statement 객체를 생성해야 한다. Statement 객체는 쿼리문을 수행하기 위한 객체이다. ④ 필요한 Query문을 입력함으로 서 필요한 데이터를 Result 객체로부터 추출한다. <그림 2> 는 클래스와 메서드를 자세히 소개하였다.



< 그림 2 : 프로그래밍 단계 >

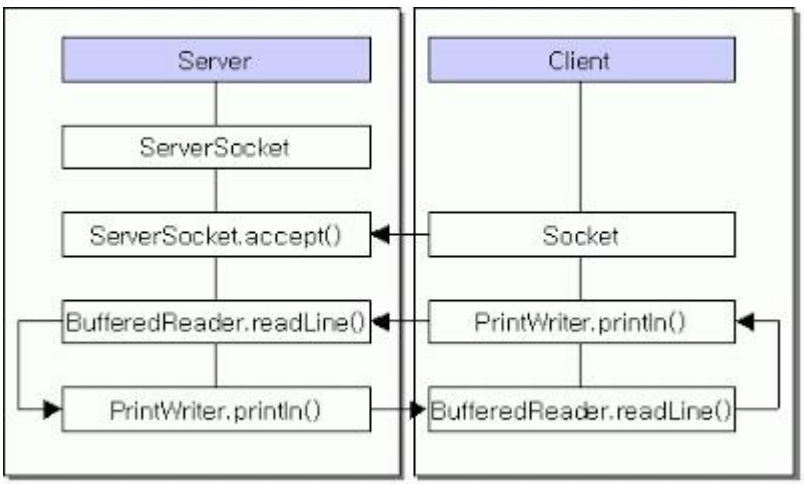
2.2 소켓 프로그래밍

자바 언어는 대부분의 네트워크 지원 기능을 소켓 프로그래밍의 방식으로 제공하고 있기 때문에 자바 네트워크 프로그래밍을 하기 위해서는 먼저 소켓 프로그래밍에 대해서 이해하고 있어야 한다. Socket은 원래 전구를 끼우는 구멍 또는 플러그를 꼽는 구멍을 말한다. 전구를 Socket에 끼우는 것에 대해 생각해보면 외부로부터 들어오는 전기선의 끝에 연결되어 있다. 그리고 전구를 그 Socket에 끼우면, 외부에서 들어온 전기의 힘으로 전구에 불이 켜지게 되는 것이다. 전구에 불을 켜기 위해서는 단지 Socket에 전구를 끼우기만 하면 된다. Socket 뒤의 문제들 즉, 전기선을 집까지 들여오는 일이나 실제 전기를 통제하는 문제들은 신경 쓰지 않아도 된다. Socket 프로그래밍은 실제 우리 생활 속의 Socket의 개념을 프로그래밍에 적용한 것이다. 즉, 프로그래머가 실제 네트워크 연결과 관련된 복잡한 것들에 대해서 신경 쓰지 않아도 마치 플러그를 꼽는 것처럼 Socket만 가지고 있다면 네트워크에 언제든지 쉽고 간편하게 연결할 수 있도록 해 놓은 것이다. 프로그래머는 단지 서버의 주소와 포트번호만 알고 있다면 아주 간단하게 서버에 접속할 수 있는 것이다.

인터넷상의 모든 컴퓨터는 고유의 IP주소를 가지고 있다. 따라서 인터넷에 연결되어 있는 어떤 컴퓨터에 접속하기 위해서는 그 컴퓨터의 IP주소를 알고 있어야 한다. 자바 네트워크 프로그래밍을 할 때에는 IP주소를 이용하여 서버에 접속할 수도 있지만 자바가 제공하는 InetAddress 클래스를 이용하여 서버에 접속하는 방법도 있다. InetAddress 클래스의 객체 역시 IP주소를 이용해서 만들지만 직접 IP주소를 이용하는

것보다 많은 장점을 가지고 있고 더욱 효율적이기 때문에 InetAddress 클래스를 이용하는 것이 바람직하다. 그 외에 domain 주소로도 가능하다. 이는 생성자 함수가 없기 때문에 getByName() 함수와 같은 정적 변수를 이용하여야 한다. 이 함수는 IP주소 문자열을 InetAddress 클래스의 객체로 만들어준다. 자바 언어 대부분의 네트워크 관련 클래스(java.net.InetAddress/ java.net.Socket)는 java.net 패키지에 포함되어 있으므로 import java.net.*;으로 import를 해주어야 한다.

Socket클래스는 클라이언트가 서버를 사용할 때 사용하는 클래스이다. 즉, 클라이언트 측 소켓이다. 클라이언트는 서버의 주소와 포트 번호를 이용하여 Socket 객체를 만들고 이 Socket 객체를 통하여 서버와 통신을 하게 된다. Socket은 서버의 주소와 서버의 포트 번호만으로 이루어지지 않는다. 그 두 가지 외에 클라이언트, 즉 자신의 컴퓨터 주소와 포트번호를 포함하고 있다. 클라이언트 주소와 포트번호를 지정하지 않은 경우 InetAddress.getLocalHost() 함수에 의해서 생성되는 주소를 사용하며, 클라이언트의 포트 번호는 컴퓨터가 임의로 할당한다.



< 그림 6 : Socket >

2.3 ROBOT 클래스

JDK1.3 부터 추가된 클래스 이다. java.awt.Robot에 위치하며, 자바로 작성된 애플리케이션 프로그램을 테스트할 수 있도록 마우스와 키보드의 입력을 프로그램 상으로 제어 할 수 있는 클래스이다. 다음과 같은 키보드와 마우스를 제어할 수 있는 메소드들이 존재한다.

클래스를 이용하면, 테스트의 자동화, 자동 실행의 데모 및 마우스나 키보드 제어가 필요한 어플리케이션을 위해서, 네이티브인 시스템 입력 이벤트를 생성할 수가 있다. Robot 의 주 된 목적은, Java 플랫폼 구현 테스트를 자동화하는 것이다.
클래스를 사용해, AWT 이벤트 큐에의 이벤트 전송 또는 플랫폼의 네이티브인 입력 큐

로 생성되는 AWT 컴퍼넌트와는 다른 입력 이벤트를 생성한다. 예를 들어 Robot.mouseMove 에서는, 마우스의 이동 이벤트를 생성하는 것만이 아니라 마우스의 커서를 실제로 움직인다.

일부의 플랫폼에서는, 저레벨 입력 제어에 액세스하기 위한, 특별한 특권 또는 확장 기능이 필요하다. 현재의 플랫폼 구성에서는 입력 제어를 할 수 없는 경우, Robot 오브젝트를 구축 하려고 하면 AWTException가 슬로우 된다. 예를 들어, X 서버로 XTEST 2.2 표준 확장 기능이 서포트되어 있지 않다. 또는 사용할 수 없는 경우, X Window 시스템은 예외를 슬로우 한다.

셀프테스트 이외의 목적으로 Robot를 사용하는 어플리케이션에서는, 이러한 예외 조건을 정상적으로 처리할 필요가 있다.

- public void mousePress(int buttons)
  마우스버튼이 눌려져 있는 상태로 만든다.
- public void mouseRelease(int buttons)
  마우스 버튼이 눌려져 있지 않은 상태로 만든다.
- public void keyPress(int keycode)
  해당 키(keycode)가 눌려져 있는 상태로 만든다.
- public void keyRelease(int keycode)
  해당키(keycode)가 눌려져 있지 않은 상태로 만든다.

```
public void updateData(Object object) {
    ArrayList Objects = (ArrayList) object;
    for (int i=0; i<Objects.size(); i++) {
        Object obj = Objects.get(i);

        if (obj instanceof MouseEvent)
            applyMouseEvent((MouseEvent)obj);
        else if (obj instanceof KeyEvent)
            applyKeyEvent((KeyEvent)obj);
    }
}

public void applyMouseEvent(MouseEvent evt) {
    rt.mouseMove(evt.getX(), evt.getY());
    int buttonMask = 0;
    int buttons = evt.getButton();
    if ((buttons == MouseEvent.BUTTON1)) buttonMask = InputEvent.BUTTON1_MASK;
    if ((buttons == MouseEvent.BUTTON2)) buttonMask |= InputEvent.BUTTON2_MASK;
    if ((buttons == MouseEvent.BUTTON3)) buttonMask |= InputEvent.BUTTON3_MASK;
    switch(evt.getID()) {
        case MouseEvent.MOUSE_PRESSED: rt.mousePress(buttonMask); break;
        case MouseEvent.MOUSE_RELEASED: rt.mouseRelease(buttonMask); break;
        case MouseEvent.MOUSE_WHEEL: rt.mouseWheel(
                ((MouseWheelEvent) evt).getUnitsToScroll()); break;
    }
}

public void applyKeyEvent(KeyEvent evt) {
    switch(evt.getID()) {
        case KeyEvent.KEY_PRESSED: rt.keyPress(evt.getKeyCode()); break;
        case KeyEvent.KEY_RELEASED: rt.keyRelease(evt.getKeyCode()); break;
    }
```

< 소스 1 : Robot.java >

2.4 인증

2.4.1 OTP발생기

OTP는 인터넷, 휴대폰, 전화 등을 다양한 매체를 이용하여 은행, 증권, 선물사의 전자금융 거래 시에 고정된 비밀번호 대신 사용되는 매번 새롭게 바뀌는 비번호이다.

보안카드의 경우 34개 이내의 비밀번호가 반복적으로 사용되므로 재사용 할 수 있지만, OTP는 일회용비밀번호로써 재사용이 불가능하고 생성된 OTP를 통해서 사용자의 비밀정보를 유추할 수 없다. 생성되는 숫자 패턴 또한 100만개 이상의 임의 숫자를 조합하여 생성하는 특성이 있어서 유추하는 것은 수학적으로 불가능하다.

< 그림 7 : OTP >

2.4.2 보안카드

보안카드는 35개 이내의 난수가 적혀진 카드로, 전자금융 거래 시 사용자가 카드에 인쇄된 번호를 직접 입력하고, 응답번호와 일치여부를 판단하여 전자금융거래를 수행한다.

현재 모든 은행과 대부분의 증권사가 전자금융거래를 위해 사용하고 있으며 인증을 위해 사용된다는 점은 같지만 안전카드, 시크릿카드 등 명칭이 각기 다르다.

< 그림 8 : 보안카드 >

2.4.3 HSM(Hardware Security Module)

MSM은 전자서명 생성키 등 비밀정보를 안전하게 저장·보관 및 키 생성, 전자서명 생성 등이 기기 내부에서 처리되도록 구현된 스마트 칩을 내장한 하드웨어 모듈로 휴대가 가능한 인증서 보안 기술이다.

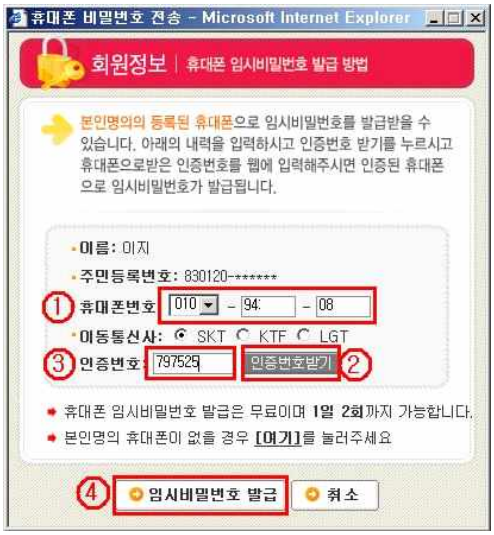
HSM은 <그림 9>과 같이 일반 USB 메모리스틱처럼 PC의 USB슬롯에 연결하여 사용한다. 연산장치와 메모리 등이 포함된 스마트카드 칩을 탑재해 전자서명과 암호화 등 모든 프로세스가 매체 내부에서 이루어지기 때문에 PC에 설치되 해킹 프로그램이나 악성코드를 통해서 HSM 내부에 저장된 비밀정보에 접근할 수 없다. HSM 내부에서 생성되어 안전한 메모리 공간에 저장된 전자서명키는 다른 매체로의 복사나 이동이 금지되어 있어, 전자서명키의 외부 유출을 원칙적으로 차단한다.

< 그림 9 : HSM 예 >

2.4.4 휴대폰 SMS(거래내역통보)

휴대폰 SMS는 인터넷뱅킹, 텔레뱅킹 등의 전자금융 서비스를 이용한 자금이체내역을 휴대폰으로 통지하는 서비스로 전자금융보안 2등급 거래 이용수단이다. 사용자의 주요 거래 또는 중요 통지사항을 사후에 실시간으로 알려주는 방식이다.

<그림 10>는 휴대폰 SMS인증 요청하는 화면이다. ① 인증 받을 휴대폰 번호와 이동통신사를 선택한다. ② [인증번호받기]를 클릭하면 하단의 화면과 같이 인증번호 전송완료에 대한 확인 메시지가 나오며 해당 휴대폰으로 인증번호 6자리를 문자 메시지로 전송된다. ③ 문자메시지로 받은 임시비밀번호 6자리를 공백란에 입력하고 버튼을 클릭하면 하단의 메시지 창이 나오면서 비밀번호 전송이 완료된다.



< 그림 10 : SMS 인증 >

2.4.5 바이오인증

바이오정보 인증과 관련해 가장 쉽게 접할 수 있는 것은 지문인식기술이다. 지문인식기술은 인터넷뱅킹 접속 시 또는 자금 이체 시 지문정보를 이용하여 인증을 수행하며, 목제 및 해킹 위험이 적다.

우리은행에서 도입한 지문인식기술은 지문정보 분석 알고리즘에 의해 사용자의 고유한 지문정보 템플릿을 생성한다. 지문정보는 복제가 불가능하기 때문에 타인이 지문을 복사해 사용할 수 없으며, 사용자 본인의 지문을 입력함으로서 안전한 사용자 인증을 수행한다. 특히 인터넷 뱅킹 접속시 ID와 비밀번호, 자금이체 시 필요한 보안카드나 인증서 비밀번호 대신 사용자의 지문정보를 입력하는 것으로 인터넷뱅킹을 이용할 수 있다. 바이오인증은 바이오 인식기기의 보급 및 사용자의 인식 등의 문제로 거의 사용되고 있지 않다.

국내 전자금융의 바이오인증 적용현황 조사결과, 우리은행이 바이오인증을 유일하게 적용하고 있다.

2.5 전자서명

전자서명이라 함은 서명자를 확인하고 서명자가 당해 전자문서에 서명을 하였음을 나타내는 데 이용하기 위하여 당해 전자문서에 첨부되거나 논리적으로 결합된 전자적 형태의 정보를 말한다. 공개키기반구조(PKI) 기술측면에서 전자서명이란 전자문서의 해시(HASH)값을 서명자의 개인키(전자서명생성정보)로 변환(암호화)한 것으로서 RSA사에서 만든 PKCS#7 표준이 널리 사용되고 있다.

즉, 원래의 문서 내용을 A라고 하면 A의 해쉬값을 잘 알려진 Hash 함수인 SHA1 같은 함수 하나를 정해 이런 Hash함수로 문서 A의 해쉬값을 구하고 이 해쉬값을, 보내는 사람(철수)의 개인키로 암호화 한다. 이런 다음 이렇게 암호화된 해쉬값을 원래 문서 A 끝에 첨부하여 이 문서 전체를 받는 사람(영희)에게 보낸다.

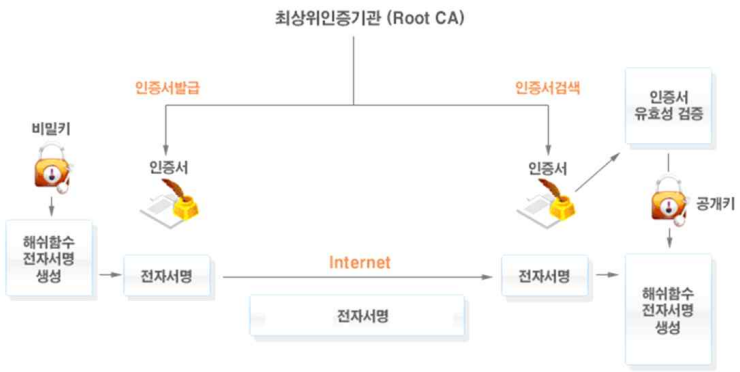
영희는 먼저 메시지가 오면 해쉬값을 뺀 앞부분의 문서에 대해 철수가 사용했던 Hash 함수를 이용해 받은 문서의 해쉬값을 구한다. 그 다음 문서 뒤에 달린 암호화된 해쉬값을 철수의 공개키로 복호화한 다음 이 복호화된 해쉬값을, 받은 문서의 해쉬값과 비교하게 된다. 이 두개의 해쉬값이 동일한 값이면 서명이 올바른 것이고 값이 서로 다르거나 변환에 오류가 있으면 서명이 틀린 것이다. 전자서명은 수기서명과 동일한 효력을 지닌다.

전자서명 생성 과정

- 원본문서를 작성한다.
- 원본문서를 해쉬함수를 이용하여 특정 크기의 해쉬값(메시지 요약)으로 변환한다.
- 이 값을 서명인의 개인키로 암호화한다.
- 암호화 한 데이타(전자서명)를 원본 문서에 붙여서 수신인에게 전송한다.
  (인증서를 포함하여 전송)

전자서명 검증 과정

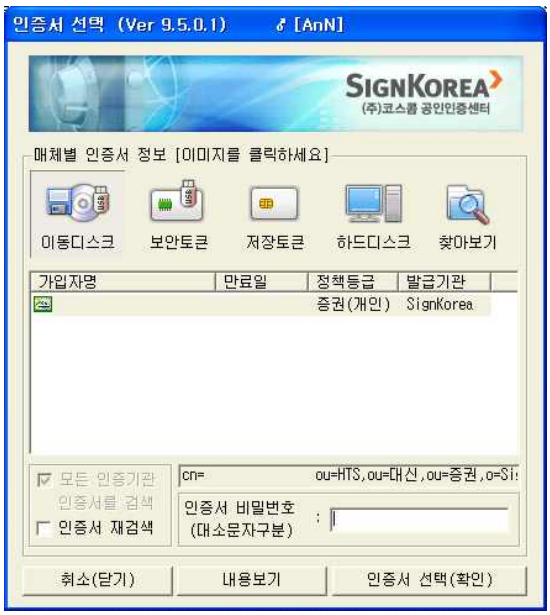
- 전송된 문서를 수신한다.
- 원본문서와 전자서명 데이터를 분리한다.
- 인증서에 담긴 공개키를 이용하여 전자서명을 복호화한다.
- 위에서 분리된 원문을 해쉬함수를 이용하여 메시지요약으로 변환한다.
- 이 메시지요약과 복호화 메시지요약을 비교한다.
  (같으면 무결성이 보장, 같지 않으면, 전송도중 변조되었음을 의미)



< 그림 5 : 전자서명 >

2.6 인증서

공인인증서는 전자 서명의 검증에 필요한 공개키(전자서명 법에는 전자서명검증정보로 표기)에 소유자 정보를 추가하여 인증기관이 서명한 일종의 전자 신분증(증명서)이다. 공개키 증명서, 디지털 증명서, 전자 증명서 등으로도 불린다. 공인인증서는 개인키(전자서명 법에는 전자서명생성정보로 표기)와 한 쌍으로 존재한다. 공인인증서는 OpenSSL의 ssl-ca나 수세 리눅스의 genssIcert와 같은 도구를 포함한 유닉스 기반 서버용으로 작성되었다. 비대면 온라인 방식의 전자상거래에서 상대방과의 계약서 작성, 신원확인 등에 전자서명이 필요하며 동시에 공인인증서로 해당 전자서명을 생성한 자의 신원을 확인하게 된다. 공개키 기반구조(PKI)는 전자서명을 생성하고 검증하는데 사용되는 개인키와 공개키를 안전하게 나누어주는 역할을 담당하는 신뢰된 제3자(인증기관)의 존재를 전제로 하고 있다. 한국의 공인인증서 제도 역시 공개키 기반구조에 입각한 제도이다. 공개키 기반구조에 입각한 인증서는 서버의 신원을 확인하는데 사용되는 서버인증서와 이용자의 신원을 확인하는데 사용되는 개인인증서로 나누어 볼 수 있다. 한국의 공인인증서도 이 두 가지 용도에 모두 사용될 수는 있지만, 한국의 공인인증서를 서버인증서로 사용할 경우, 파이어폭스 웹브라우저는 그러한 서버인증서를 신뢰하지 않으므로 현실적으로 서버 신원 확인 용도로 한국의 공인인증기관이 발급한 서버인증서를 사용하기는 무리가 따른다. 한국의 공인인증서는 따라서 개인인증서로 주로 사용되고 있다. 한국의 공인인증서 및 개인키 역시 파일 양식 자체는 국제표준을 따르고 있지만, 그 파일들이 보관, 저장되는 위치와 방법이 독특하여 웹브라우저로는 사용이 불가능하다. 그 결과, 한국의 공인인증서를 이용하려면 이용자가 추가프로그램을 반드시 설치해야만 한다.

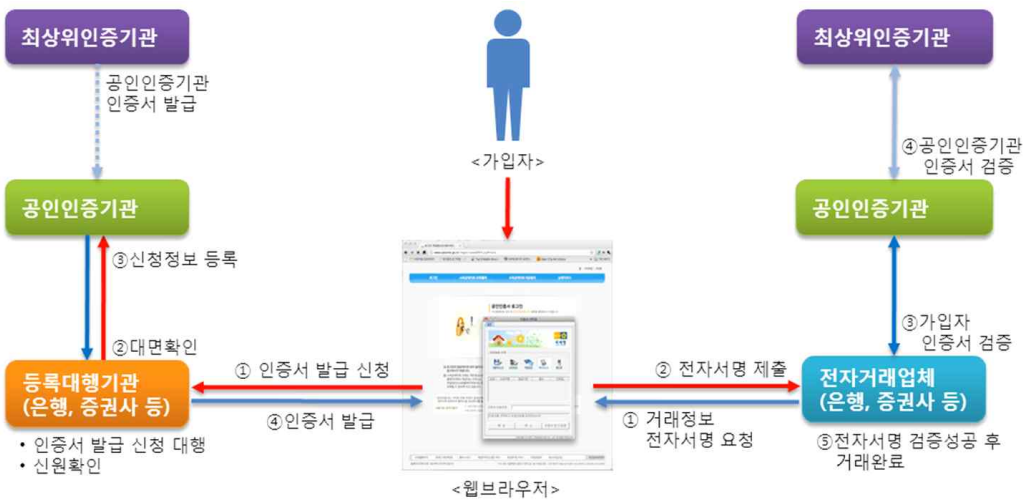


<그림 10 : 인증서 >

인증서는 원래 금융거래에만 사용되는 것이 아니라, 모든 전자적 거래(금전적이건 비금전적이건)에서 당사자의 신원을 확인하거나, 전자서명을 하는 용도로 사용될 수 있고, 한국의 공인인증서도 물론 그런 다양한 용도로 사용될 수도 있다. 그러나 현실적으로 공인인증서는 전자금융거래에서 주로 사용되고 있다. 금융위원회는 전자금융거래에 "공인인증서 등"을 사용하도록 강제하고 있다.

공개키 인증서에 들어갈 내용

- 일련 번호: 증명서를 개별 인증할 때 사용
- 주체: 사람의 이름이나 증명자
- 서명 알고리즘: 서명을 만드는 데 쓰이는 알고리즘
- 발행자: 정보를 검증하고 증명서를 발행하는 실체
- 유효 기간 (시작): 처음 효력을 발휘하는 기간
- 유효 기간 (끝): 효력 만기일
- 키 이용 목적: 공인키의 사용 목적. (이를테면 서명, 인증 서명 등)
- 공인 키: SSL 목적
- 지문 알고리즘: 인증서를 hash하는 데 쓰이는 알고리즘
- 지문: 증명서가 개봉되지 않았음을 증명하는 hash 자체

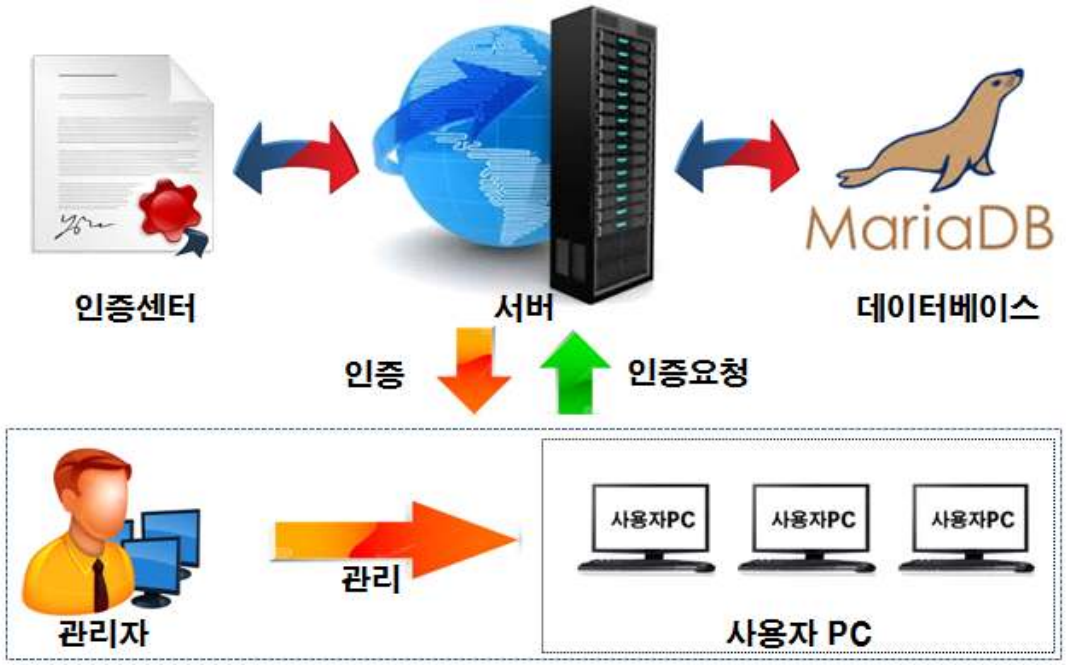


< 그림 4 : 인증서 발급>

3. 본 론

3.1 프로그램 구성

- 사용자는 서버를 통해 회원가입 후 로그인이 가능하다.
- 인증서로그인을 이용할 경우 사용자는 사전에 인증센터에 인증서 발급요청 후 발급받아야 한다.
- 인증서로그인시 서버는 사용자의 정보와 서명값을 인증서에 포함된 공개키로 서명값을 검증하여 접속자격 여부를 판단한다. 검증 값으로 접속여부 판단한다.
- 관리자는 서버에 접속된 사용자의 ip정보를 볼 수 있으며, 접속된 사용자의 PC 버튼은 활성화가 된다. 버튼 클릭 시 사용자의 컴퓨터를 모니터링 및 제어가 가능하다.

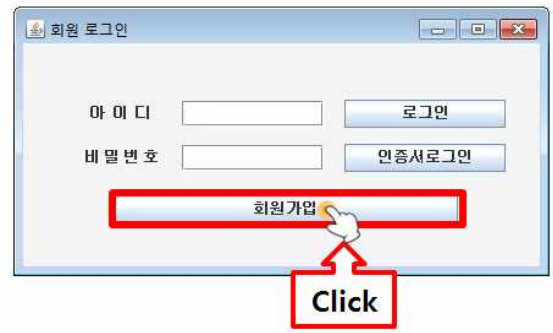


<그림 3 : 구상도>

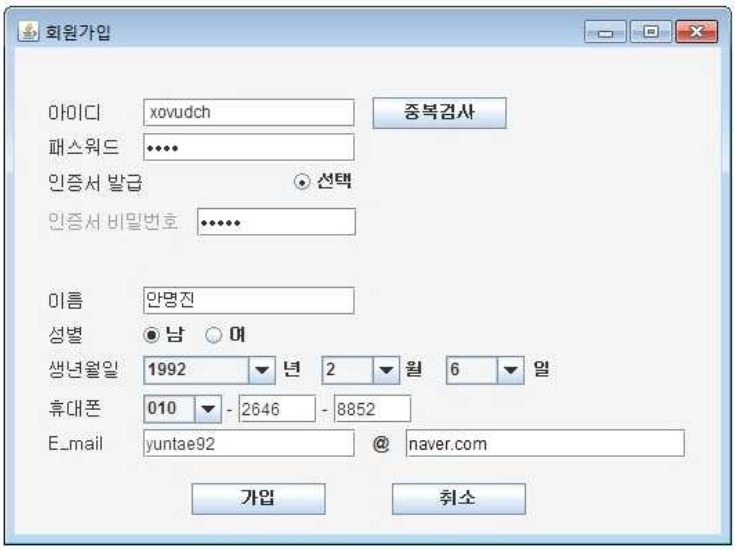
3.2 회원가입 및 로그인

기본적으로 회원가입 창에서 정보를 입력하게 되면 서버로 전송되고 서버에서는 DB에 저장하게 된다. 물론 중복된 아이디가 존재한다면 회원가입은 되지 않는다. 서버와 DB의 연동은 JDBC를 사용하여 연동하였다. 본 프로그램에서는 공인인증서 발급을 회원가입 창에서 선택 후 개인키 암호화 저장을 위한 패스워드를 입력하도록 하고 가입과 동시에 인증서를 발급 받도록 설계하였다. 이때 개인키의 암호화 저장을 위한 패스워드가 노출되면 피해가 크므로 안전하게 관리해야 한다. 따라서 패스워드는 사용자의 머릿속에만 있고 컴퓨터 내에 저장되는 것이 아니며 개인키는 패스워드로 암호화하여 KeyStore

객체로 만들어 개인키 파일에 저장하여 안전하게 보호된다. 패스워드 분실 시에는 인증서를 사용할 수 없으므로 재발급하여야 한다.



< 그림 4-1 : 회원가입 >



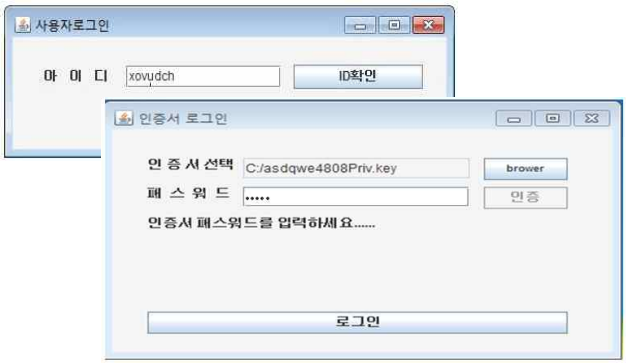
< 그림 4-2 : 회원가입 >

id와 password로 로그인 할 경우 사용자는 아래 “그림 5”번 과 같이 정보를 입력하고 로그인 버튼을 누른다. 이때 1차적으로 입력받은 정보를 서버에게 전송하게 되고 서버는 이정보를 DB에 확인 후 접근을 허용하게 된다. 마지막으로 이용자는 로그인되었다는 확인 메시지를 받게 된다. 이때 통신은 socket을 사용하였고 이대한 설명은 “2-1 관련연구” 에 있다.

< 그림 5 : ID/PW 이용한 로그인 >

다음으로 인증서를 통한 로그인 과정이다.

첫 번째로 <그림 5>에서 인증서 로그인 버튼을 누른다. 이때 아이디를 재입력하는 창 <그림 6>이 나온다. 아이디를 입력하고 인증서 로그인 버튼을 누르면 해당 아이디의 인증서를 불러오며 인증서 로그인 창이 새롭게 열린다. 전자서명은 개인키로 서명을 하여 서명값을 서버에게 전송해야 하기 때문에 이용자는 개인키를 암호화 저장할 때 입력한 패스워드를 입력해야 한다. 로그인 버튼 클릭 시 이용자의 pc에서 Server에 서명값을 전송하게 한다. 이때 개인키로 평문을 서명하게 되는데 평문은 편의상 ID로 지정하였다. 전송 받은 서버는 사용자의 정보와 서명값을 인증서에 포함된 공개키로 검증하여 사용자의 신원을 확인한다. ID와 PASSWORD를 사용한 로그인 방식은 공격자가 패스워드 훔쳐보기, 사전공격 등으로 공격 가능하지만 전자서명을 이용한 로그인 방식은 개인키를 가진 자용자만이 로그인할 수 있으므로 더 안전한 인증 방식이다.

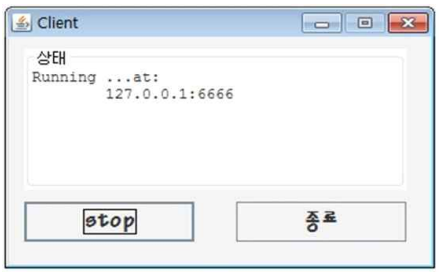


< 그림 6 : 인증서로그인 >

3.3 PC 원격제어

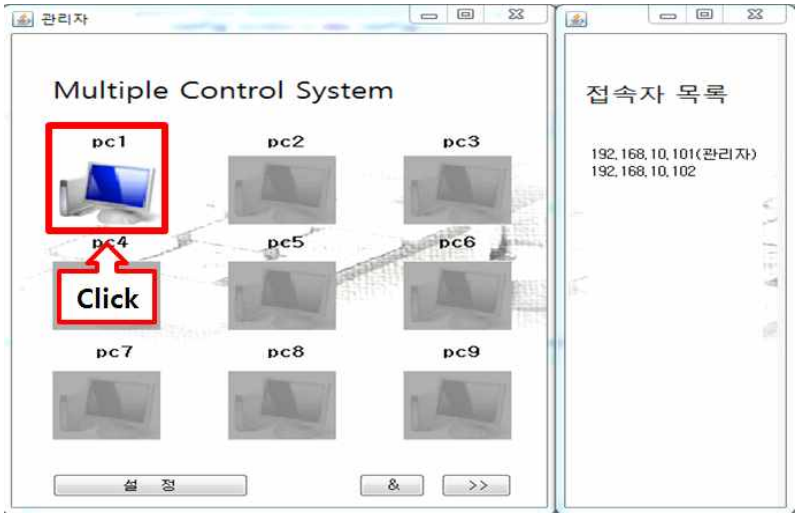
아이디와 패스워드를 통한 인증이나 인증서를 사용한 인증을 통하여 허가된 이용자는 프로그램을 사용하게 된다. 프로그램을 구동시키게 되면 서버로부터 원격제어가 허용된다. 서버는 간단한 버튼을 클릭하여 이용자를 원격제어 한다.

<그림 7>은 이용자가 로그인에 성공한 후 실행되는 프로그램이다. 상태 창에 간단하게 실행여부를 표시하게 하였고 본 팀(크자스)에서는 이용자가 프로그램을 실행하게 함으로 서버라도 아무 때나 무분별하게 제어할 수 없게 하였다. 따라서 start 버튼을 클릭 해야만 서버가 제어가능하게 구현하였다. 사용자는 종료 버튼을 눌러 서버의 모니터링 기능을 중지시킬 수 있다.



< 그림 7 : 사용자의 프로그램 >

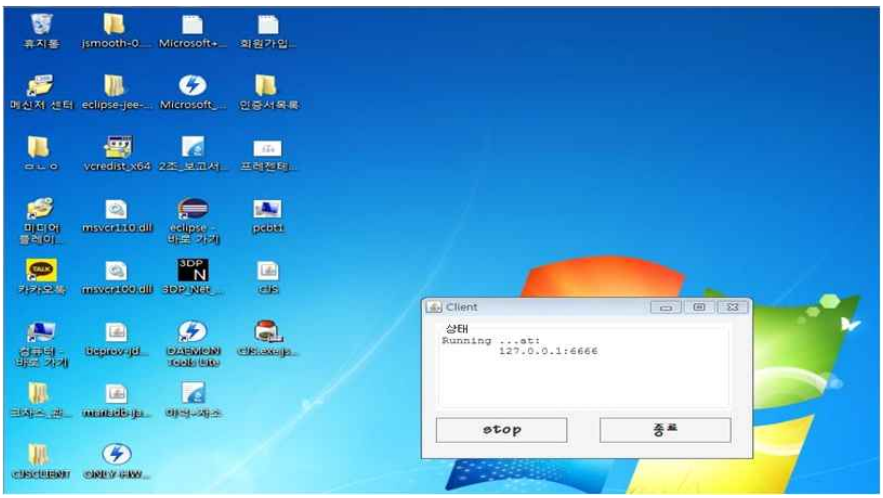
다음으로 <그림 8>은 관리자 화면이다.



< 그림 7: 관리자 프로그램 >

관리자 화면에는 크게 2가지 특징이 있다. 첫 번째는 서버에 접속되어 있는 컴퓨터들의 아이피를 접속자 목록에 추가 되게 하였고, 접속이 끊길 때도 사라지게 하였다. pc1, pc2, pc3…… 등은 실습실이나 작업장의 배치도로 활용 가능 하며 접속자 목록에 ip가 추가됨과 동시에 각 버튼색이 변하게 된다. 이는 사용자의 접속을 알리는 일이다.

밑에 “그림 8”은 제어된 이용자의 화면이다. 원격제어에 위에서 소개한 Robot class를 사용하고, Server, 인증센터, 이용자 간의 통신은 소켓을 사용했으며 “2.1 관련연구”에 소개되어있다.



< 그림 8 : 원격제어화면 >

4. 결론 및 향후과제

4.1 결론 및 기대효과

본 프로그램은 개별 pc를 원격 제어하고 모니터링 함으로 써 정상 작동 여부 및 이상 상황 점검 및 대응이 가능하며 분산된 pc 운영을 효율적이게 한다. 또한 실습실이나 작업공간에서의 사용자들의 비인가 프로그램 등록 및 사용을 점검하고 통제하여 업무나 실습의 효율성을 제공한다. 프로그램에 인증기법으로 사용된 인증서 기법을 사용하여 보안성을 높여 허가되지 않은 사용자 즉 인증되지 않은 사람의 접속으로 인한 재산피해를 줄일 수 있게 하였다.

앞으로 사용자 인증에 패스워드 및 인증서 로그인의 이중화 체제를 적용하고 비인기 접근통제 등 보안 기능을 대폭 보강하여 보다 안전한 프로그램사용을 기대할 수 있다.

마지막으로 우리 크자스는 본 프로그램을 개발하며 모든 조원들이 임무를 분담하여 인증서 로그인 등을 직접 구현, 보안 소프트웨어 개발 기량을 향상시키는 계기가 되었다.

4.2 향후계획

현재 인증서사용에 있어서 문제들이 몇가지 존재한다. 첫 번째로 ActiveX사용의 문제이다. (ActiveX ? 온라인에서 전자상거래, 음악·동영상 재생, 파일 다운로드 등에 필요한 프로그램을 설치하도록 지원하는 기술로써 인터넷 익스플로러와 결합해 사용) ActiveX의 사용은 국내 온라인 쇼핑몰을 방문하는 외국인들에게 장벽이 되었다.

외국인이 공인인증서를 사용하려면 외국인등록증을 가지고 가까운 대행기관을 직접 방문해야 한다. 언어가 불편한 외국인들에는 현실적으로 불가능에 가까운 일이다. 또한 공인인증서 프로그램의 대부분이 ActiveX 기술을 바탕으로 만들어졌기에 익스플로러가 아닌 다른 접속 프로그램에선 불가능하다.

하지만 전문가들 사이에서는 공인인증서를 대체하는 새로운 수단이 보안 상 완벽할 수 있을까라는 의문도 존재했다. 이것은 우리 팀도 같은 생각이다.

두 번째로 여러대의 컴퓨팅 기기들이 존재 할 때 기기인증을 외부의 인증기관에 의존해야 한다는 것이다. 사용자가 사용하고 있는 여러대의 컴퓨팅 기기들에 대한 기기인증은 외부의 인증기관에 의존하기보다 사용자가 직접 관리할 수 있도록 하는 것이 현재의 실제 사용환경에 부합하는 방법이다.

사용자가 이미 발급받아 사용하고 있는 인증서가 있다고 가정할 때 이것에 기반하여 자신이 소유한 기기들에게로 인증을 자체 확장하여 인증키를 만들고 사용할 수 있도록 하는 방식을 연구할 것이고 수행해 나갈 것이다.

※ 별 첨

1. 프로그램 소스

[Server]

```
package 졸작;

import java.awt.EventQueue;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.*;


public class login {
	String id,pw;
	private JFrame frame;
	private JTextField tx_id;
	private JPasswordField tx_pw;
	Connection conn = null;
	Statement stmt = null;
	PreparedStatement pstmt;
	ResultSet rs;
	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
```

```
                login window = new login();
                                        window.frame.setVisible(true);
                                        jrdesktop.main  m= new jrdesktop.main();
                                        m.main(args);
                                        new ListServer();



                                } catch (Exception e) {
                                        e.printStackTrace();
                                }

                        }
                });
        }

        /**
         * Create the application.
         */
        public login() {
                initialize();
        }

        /**
         * Initialize the contents of the frame.
         */
        private void initialize() {
                frame = new JFrame();
                frame.setTitle("WuAD00WuB9ACWuC790 WuB85CWuADF8WuC778");
                frame.setBounds(100, 100, 439, 209);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.getContentPane().setLayout(null);

                JLabel label = new JLabel("WuC544  WuC774  WuB514");
                label.setBounds(54, 50, 62, 15);
                frame.getContentPane().add(label);

                JLabel label_1 = new JLabel("WuBE44 WuBC00 WuBC88 WuD638");
                label_1.setBounds(50, 86, 62, 15);
                frame.getContentPane().add(label_1);
```

```
tx_id = new JTextField();
tx_id.setBounds(130, 47, 116, 21);
frame.getContentPane().add(tx_id);
tx_id.setColumns(10);

tx_pw = new JPasswordField();
tx_pw.setColumns(10);
tx_pw.setBounds(130, 83, 116, 21);
frame.getContentPane().add(tx_pw);

JButton btnNewButton = new JButton("회원가입");
btnNewButton.setBounds(79, 128, 258, 23);
frame.getContentPane().add(btnNewButton);
btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                        new join();
                        join window = new join();
                        window.frame.setVisible(true);
                }
});

JButton btnNewButton_1 = new JButton("로그인");
btnNewButton_1.setBounds(258, 45, 105, 25);
btnNewButton_1.setEnabled(true);
frame.getContentPane().add(btnNewButton_1);
btnNewButton_1.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                        id=tx_id.getText();
                        pw =tx_pw.getText();
                        me_login(id, pw);
                }
});

JButton btnNewButton_4 = new JButton("인증서로그인");
btnNewButton_4.setBounds(258, 80, 105, 25);
btnNewButton_4.setEnabled(true);

frame.getContentPane().add(btnNewButton_4);
```

```
	}
	public void conect() {
		//db 연결

		try {
			Class.forName("org.mariadb.jdbc.Driver");
			// jdbc 드라이버를 로드한다.
			conn = DriverManager.getConnection(
					"jdbc:mariadb://localhost:3306/cjs",
"root", "12345");
			// db에 연결.

			// conn.close();//db연결 종료
		} catch (ClassNotFoundException e) {
			// TODO Auto-generated catch block
			System.out.println("class를 찾을수 없다.");
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			System.out.println(e.getMessage());
			e.printStackTrace();
		}
	}

	public void me_login(String id, String pw) {
		conect();
		System.out.println(" 접속");
		try {

			if (id.length() == 0) {
				JOptionPane.showMessageDialog(null, "id를
입력해주세요.");
			}
			if (pw.length() == 0) {
				JOptionPane.showMessageDialog(null, "pw를
입력해주세요.");
			}
			// 아이디 패스워드 존재확인
			String sql = "select * from cjstb where id=? and pw =?";
```

```
                        pstmt = conn.prepareStatement(sql);
                        pstmt.setString(1, id);
                        pstmt.setString(2, pw);

                        rs = pstmt.executeQuery();

                        if (rs.next()) {
                                JOptionPane.showMessageDialog(null,
"로그인되었습니다.");
                                conn.close();
                                //sstmt.close();
                                pstmt.close();
                                frame.setVisible(false);
                                Server window = new Server();
                                window.frame.setVisible(true);
                                //new mainFrame();

                        }
                        else {
                                JOptionPane.showMessageDialog(null, "id/ pw가
옳바르지않습니다.");
                                conn.close();
                                stmt.close();
                                pstmt.close();
                        }

                } catch (SQLException e) {
                        // TODO Auto-generated catch block

                        System.out.println(e.getMessage());
                        e.printStackTrace();
                }
        }
}
package 졸작;

import java.awt.*;
import javax.swing.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
```

```
import 졸작.ListClient.Socket_thread;
import jrdesktop.main;
import jrdesktop.viewer.rmi.Viewer;
import java.io.*;
import java.net.*;
import java.util.*;
import java.awt.event.*;

public class Server {
	public JFrame frame;
	int port = 6666;
	public static String ip1 = "192.168.10.102", ip2 = "192.168.10.97", ip3,
			ip4, ip5, ip6, ip7, ip8, ip9;

	// Network Source
	private Socket socket = null;
	private String ip = "127.0.0.1";
	private int port1 = 7777;

	private InputStream inputStream;
	private OutputStream outputStream;
	private DataOutputStream dataOutputStream;
	private DataInputStream dataInputStream;

	// etc valuable
	private StringTokenizer stringTokenizer;
	/**
	 * Launch the application.
	 */
	static JButton btnNewButton, button;

	/**
	 * Create the application.
	 */
	public Server() {
		ListClient lc = new ListClient();
		frame = new JFrame();
		frame.getContentPane().setForeground(SystemColor.desktop);
		frame.setTitle("WuAD00WuB9ACWuC790");
```

```
frame.setBounds(265, 100, 412, 508);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.getContentPane().setLayout(null);

                JLabel lblPcnum = new JLabel("pc1");
                lblPcnum.setFont(new Font("굴림", Font.BOLD, 15));
                lblPcnum.setBounds(59, 100, 57, 15);
                frame.getContentPane().add(lblPcnum);

                JLabel lblPc_2 = new JLabel("pc4");
                lblPc_2.setFont(new Font("굴림", Font.BOLD, 15));
                lblPc_2.setBounds(59, 200, 57, 15);
                frame.getContentPane().add(lblPc_2);

                JLabel lblNewLabel_2 = new JLabel("pc7");
                lblNewLabel_2.setFont(new Font("굴림", Font.BOLD, 15));
                lblNewLabel_2.setBounds(62, 309, 57, 15);
                frame.getContentPane().add(lblNewLabel_2);

                JLabel lblPc_5 = new JLabel("pc8");
                lblPc_5.setFont(new Font("굴림", Font.BOLD, 15));
                lblPc_5.setBounds(189, 309, 57, 15);
                frame.getContentPane().add(lblPc_5);

                JLabel lblPc_3 = new JLabel("pc5");
                lblPc_3.setFont(new Font("굴림", Font.BOLD, 15));
                lblPc_3.setBounds(189, 200, 57, 15);
                frame.getContentPane().add(lblPc_3);

                JLabel lblPc = new JLabel("pc2");
                lblPc.setFont(new Font("굴림", Font.BOLD, 15));
                lblPc.setBounds(189, 100, 57, 15);
                frame.getContentPane().add(lblPc);

                JLabel lblPc_6 = new JLabel("pc9");
                lblPc_6.setFont(new Font("굴림", Font.BOLD, 15));
                lblPc_6.setBounds(319, 309, 57, 15);
                frame.getContentPane().add(lblPc_6);

                JLabel lblPc_4 = new JLabel("pc6");
```

```
lblPc_4.setFont(new Font("굴림", Font.BOLD, 15));
		lblPc_4.setBounds(316, 200, 57, 15);
		frame.getContentPane().add(lblPc_4);

		JLabel lblPc_1 = new JLabel("pc3");
		lblPc_1.setFont(new Font("굴림", Font.BOLD, 15));
		lblPc_1.setBounds(319, 100, 57, 15);
		frame.getContentPane().add(lblPc_1);
		btnNewButton = new JButton("");
		btnNewButton.setIcon(new ImageIcon(
				"C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
		btnNewButton.setForeground(Color.BLACK);
		btnNewButton.setBounds(30, 120, 80, 70);
		btnNewButton.setEnabled(false);
		frame.getContentPane().add(btnNewButton);
		btnNewButton.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
				jrdesktop.mainFrame1 m1 = new
jrdesktop.mainFrame1();
				m1.frame.setVisible(false);
				main.startViewer(ip1, port);
				System.out.println(ip1 + port);

			}
		});
		button = new JButton("");
		button.setEnabled(false);
		button.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
		button.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				jrdesktop.mainFrame1 m1 = new
jrdesktop.mainFrame1();
				m1.frame.setVisible(false);
				main.startViewer(ip2, port);
				System.out.println(ip2 + port);
			}
		});
```

```
                button.setBounds(160, 120, 80, 70);
                frame.getContentPane().add(button);

                JButton button_1 = new JButton("");
                button_1.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
                button_1.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                        }
                });
                button_1.setEnabled(false);
                button_1.setBounds(290, 120, 80, 70);
                frame.getContentPane().add(button_1);

                JButton button_2 = new JButton("");
                button_2.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
                button_2.setEnabled(false);
                button_2.setBounds(30, 225, 80, 70);
                frame.getContentPane().add(button_2);

                JButton button_3 = new JButton("");
                button_3.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
                button_3.setEnabled(false);
                button_3.setBounds(160, 225, 80, 70);
                frame.getContentPane().add(button_3);

                JButton button_4 = new JButton("");
                button_4.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
                button_4.setEnabled(false);
                button_4.setBounds(290, 225, 80, 70);
                frame.getContentPane().add(button_4);

                JButton button_5 = new JButton("");
                button_5.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
```

```
button_5.setEnabled(false);
                button_5.setBounds(30, 334, 80, 70);
                frame.getContentPane().add(button_5);

                JButton button_6 = new JButton("");
                button_6.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
                button_6.setEnabled(false);
                button_6.setBounds(160, 334, 80, 70);
                frame.getContentPane().add(button_6);

                JButton button_7 = new JButton("");
                button_7.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWcomputer2.jpg"));
                button_7.setEnabled(false);
                button_7.setBounds(290, 334, 80, 70);
                frame.getContentPane().add(button_7);

                JButton btnNewButton_1 = new JButton("설    정");
                btnNewButton_1.setForeground(SystemColor.desktop);
                btnNewButton_1.setBounds(30, 437, 145, 23);
                frame.getContentPane().add(btnNewButton_1);

                JLabel lblNewLabel = new JLabel("Multiple Control System");
                lblNewLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 23));
                lblNewLabel.setBounds(30, 33, 265, 43);
                frame.getContentPane().add(lblNewLabel);

                JButton button_8 = new JButton(">>");
                button_8.setForeground(SystemColor.desktop);
                button_8.setIcon(new
ImageIcon("C:WWUsersWWHansolWWDesktopWWlist.jpg"));
                button_8.setBounds(318, 437, 52, 23);
                frame.getContentPane().add(button_8);
                button_8.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent arg0) {
                                lc.setVisible(true);
                        }
                });
```

```
		JButton btnNewButton_2 = new JButton("WuB3C4WuC6COWuB9DOWrWn");
		btnNewButton_2.setBounds(209, 437, 97, 23);
		frame.getContentPane().add(btnNewButton_2);

		JLabel back = new JLabel("New label");
		back.setIcon(new
ImageIcon("C:WWUsersWWPCWWDesktopWWWuBC3OWuACBD3.jpg"));
		back.setBounds(0, 0, 396, 470);
		frame.getContentPane().add(back);
		btnNewButton_1.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
				config window = new config();
				window.frame.setVisible(true);
			}
		});
	}

	public static void reflash(Vector vector_user_list) {
		for (int i = 0; i < vector_user_list.size(); i++) {
			if (ip1.equals(vector_user_list.get(i)))
				btnNewButton.setEnabled(true);


			else if (ip2.equals(vector_user_list.get(i)))
				button.setEnabled(true);
			else  {
				button.setEnabled(false);
				btnNewButton.setEnabled(false);
			}
		}
	}

	public void ipset(String a1, String a2, String a3, String a4, String a5,
			String a6, String a7, String a8, String a9) {
		System.out.println(ip1 + port);
		ip1 = a1;
```

```
                ip2 = a2;
                ip3 = a3;
                ip4 = a4;
                ip5 = a5;
                ip6 = a6;
                ip7 = a7;
                ip8 = a8;
                ip9 = a9;
        }
}
package 졸작;

import java.awt.event.*;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.*;
import java.util.StringTokenizer;
import java.util.Vector;
import javax.swing.*;
import javax.swing.UIManager;


public class ListServer   {

        // Server Frame
        // Network Source
        private ServerSocket serverSocket;
        private Socket socket;
        private int port = 7777;
        private Vector vector_user = new Vector();


        private StringTokenizer stringTokenizer;
        private void server_start(){
                try {
```

```
                        UIManager

.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
                } catch (Exception e) {
                        // TODO: handle exception
                }
                try {
                        serverSocket = new ServerSocket(port);
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        JOptionPane.showMessageDialog(null, "이미 사용중인 포트",
"알림", JOptionPane.ERROR_MESSAGE);
                }
                if(serverSocket != null){ // socket ok!
                        connection();
                }

        }
        public class Socket_thread implements Runnable{
                public void run() {
                        // TODO Auto-generated method stub
                        while(true){
                                try { //wait client
                                        socket = serverSocket.accept();
                                        UserInfo userInfo = new UserInfo(socket);
                                        userInfo.start();
                                } catch (IOException e) {
                                        break;
                                }
                        }
                }
        }

        private void connection(){
                Thread thread = new Thread(new Socket_thread());
                thread.start();
        }
        public ListServer(){
```

```
        server_start();
//      start();//ACTION
}

public void serverstop() {
        // TODO Auto-generated method stub

                        try {
                                serverSocket.close();
                                vector_user.removeAllElements();
                        //      vector_room.removeAllElements();
                        } catch (IOException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
}
private class UserInfo extends Thread{
        private InputStream inputStream;
        private OutputStream outputStream;
        private DataOutputStream dataOutputStream;
        private DataInputStream dataInputStream;
        private Socket socket_user;
        private String Nickname = "";
        private String CurrentRoom = null;
        private boolean RoomCheck = true; // 기본적으로 만들 수 있는 상태
        public UserInfo(Socket socket){
                this.socket_user = socket;
                userNetwork();
        }
        public String getNicname(){
                return Nickname;
        }
        public void userNetwork(){


                try {
                        inputStream = socket_user.getInputStream();
                        dataInputStream = new
```

```
DataInputStream(inputStream);
                                outputStream = socket_user.getOutputStream();
                                dataOutputStream = new
DataOutputStream(outputStream);
                                Nickname = dataInputStream.readUTF();

                                //when a new user is connected

                                BroadCast("NewUser/" + Nickname);
                                for(int i = 0; i< vector_user.size(); i++){ //
server alert at existing user and send the new user's id
                                        UserInfo userInfo = (UserInfo)
vector_user.elementAt(i);
                                        this.send_Message("OldUser/" +
userInfo.getNicname());
                                }
                                vector_user.add(this);
                                BroadCast("user_list_update/ ");
                        } catch (IOException e) {
                                // TODO Auto-generated catch block
                                JOptionPane.showMessageDialog(null, "Stream설정
에러", "알림", JOptionPane.ERROR_MESSAGE);
                        }
                }
                private void BroadCast(String str){
                        for(int i = 0; i< vector_user.size(); i++){ // server
alert at existing user and send the new user's id
                                UserInfo userInfo = (UserInfo)
vector_user.elementAt(i);
                                userInfo.send_Message(str);
                        }
                }

                @Override
                public void run() {
                        // TODO Auto-generated method stub
                        super.run();
                        while(true){
```

```
                        try {
                                String msg = dataInputStream.readUTF();
                                InMessage(msg);
                        } catch (IOException e) {
                                // TODO Auto-generated catch block
                                try {
                                        dataInputStream.close();
                                        dataOutputStream.close();
                                        socket_user.close();
                                        vector_user.remove(this);
                                        BroadCast("UserOut/"+Nickname);
                                        BroadCast("user_list_update/ ");
                                } catch (IOException e1) {}

                                break;
                        }

                }
        }
        private void InMessage(String str){ //handle the message from
client
                stringTokenizer = new StringTokenizer(str, "/");
                String protocol = stringTokenizer.nextToken();
                String message = stringTokenizer.nextToken();
                System.out.println("protocol : " + protocol);

        }
        private void send_Message(String message){
                try {
                        dataOutputStream.writeUTF(message);
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                }
        }
    }
}
package 졸작;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.*;
import java.awt.image.ImagingOpException;
import java.io.*;

import java.net.*;
import java.util.*;
import javax.swing.*;
import java.awt.Font;
import javax.swing.ImageIcon;
public class ListClient extends JFrame {

	// Login Frame

	// Main Frame
	private JPanel contentPane;
	private JTextField textField_message;
	private JList list_user = new JList();
	private JTextArea textArea_chat = new JTextArea();

	// Network Source
	private Socket socket = null;
	private String ip = "127.0.0.1";
	private int port = 7777;
	private String id = "noname";
	private InputStream inputStream;
	private OutputStream outputStream;
	private DataOutputStream dataOutputStream;
	private DataInputStream dataInputStream;
	public String myip;

	// etc valuable
	static public Vector vector_user_list = new Vector();
	// private Vector vector_room_list = new Vector();
	private StringTokenizer stringTokenizer;

	public ListClient() {
		try {
```

```
                        InetAddress ip1 = InetAddress.getLocalHost();
                        System.out.println(ip1.getHostAddress());
                        myip = ip1.getHostAddress();
                        System.out.println(myip);

                } catch (UnknownHostException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }
        private void start() { // ACTION
                try {
                        UIManager

.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
                } catch (Exception e) {
                        // TODO: handle exception
                }
                network();
        }

        private void Main_init() { // Main GUI
                // this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                this.setBounds(675, 100, 179, 508);
                contentPane = new JPanel();
                contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
                this.setContentPane(contentPane);
                contentPane.setLayout(null);

                JLabel lblNewLabel = new JLabel("WuC811WuC18DWuC790
WuBAA9WuB85D");
                lblNewLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 20));
                lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
                lblNewLabel.setBounds(12, 45, 110, 25);
                contentPane.add(lblNewLabel);
                list_user.setBounds(17, 111, 129, 311);
                contentPane.add(list_user);
```

```
                list_user.setListData(vector_user_list);
                JLabel back = new JLabel("");
                back.setIcon(new
ImageIcon("C:₩₩Users₩₩PC₩₩Desktop₩₩₩uBC30₩uACBD2.jpg"));
                back.setBounds(0, 0, 157, 470);
                contentPane.add(back);
        }

        private void network() {
                try {
                        socket = new Socket(ip, port);
                        if (socket != null) { // socket ok!!
                                connection();
                        }
                } catch (UnknownHostException e) {
                        JOptionPane.showMessageDialog(null, "연결 실패", "알림",
                                        JOptionPane.ERROR_MESSAGE);
                } catch (IOException e) {
                        JOptionPane.showMessageDialog(null, "연결 실패", "알림",
                                        JOptionPane.ERROR_MESSAGE);
                }
        }

        private void connection() {
                try {
                        inputStream = socket.getInputStream();
                        dataInputStream = new DataInputStream(inputStream);
                        outputStream = socket.getOutputStream();
                        dataOutputStream = new DataOutputStream(outputStream);
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        JOptionPane.showMessageDialog(null, "연결 실패", "알림",
                                        JOptionPane.ERROR_MESSAGE);
                }
                this.setVisible(true);
                send_message(myip + "(관리자)"); // first connect -> send id
                vector_user_list.add(myip + "(관리자)"); // add my id in
user_list
```

```
            System.out.println(myip);

            Thread thread = new Thread(new Socket_thread());
            thread.start();

        }

        public class Socket_thread implements Runnable {
            public void run() {
                // TODO Auto-generated method stub
                while (true) {
                    try {

                        InMessage(dataInputStream.readUTF());
                    } catch (IOException e) {
                        // TODO Auto-generated catch block
                        try {
                            outputStream.close();
                            inputStream.close();
                            dataInputStream.close();
                            dataOutputStream.close();
                            socket.close();

JOptionPane.showMessageDialog(null, "서버와 접속 끊어짐", "알림",

JOptionPane.ERROR_MESSAGE);
                        } catch (IOException e1) {
                        }
                        break;
                    }
                }
            }
        }
        private void InMessage(String str) { // all message from server
            stringTokenizer = new StringTokenizer(str, "/");
            String protocol = stringTokenizer.nextToken();
```

```
                String message = stringTokenizer.nextToken();
            System.out.println("프로토콜 : " + protocol);
            System.out.println("내용 : " + message);
            if (protocol.equals("NewUser")) {

                vector_user_list.add(message);
            //      졸작.Server.reflash(vector_user_list);
            } else if (protocol.equals("OldUser")) {
                vector_user_list.add(message);
            //      졸작.Server.reflash(vector_user_list);
            }
            else if (protocol.equals("user_list_update")) {
                list_user.setListData(vector_user_list);
                졸작.Server.reflash(vector_user_list);

            } else if (protocol.equals("UserOut")) {
                vector_user_list.remove(message);
            //      졸작.Server.reflash(vector_user_list);

            }
        }

        private String getLocalServerIp() {
            try {
                for (Enumeration<NetworkInterface> en = NetworkInterface
                                .getNetworkInterfaces();
en.hasMoreElements();) {
                        NetworkInterface intf = en.nextElement();
                        for (Enumeration<InetAddress> enumIpAddr = intf
                                        .getInetAddresses();
enumIpAddr.hasMoreElements();) {
                                InetAddress inetAddress =
enumIpAddr.nextElement();
                                if (!inetAddress.isLoopbackAddress()
                                                &&
!inetAddress.isLinkLocalAddress()
                                                &&
inetAddress.isSiteLocalAddress()) {
```

```
                                                return
inetAddress.getHostAddress().toString();
                                        }
                                }
                        }
                }

                catch (SocketException ex) {
                }
                return null;
        }

        private void send_message(String message) { // button
                try {
                        dataOutputStream.writeUTF(message);
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                new ListClient();
        }

}
package 졸작;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.FlowLayout;
import javax.swing.ButtonGroup;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
```

```
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JTable;
import javax.swing.JRadioButton;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.swing.DefaultComboBoxModel;
import javax.swing.event.AncestorListener;


public class join  {
	Connection conn = null;
	Statement stmt = null;
	PreparedStatement pstmt;
	ResultSet rs;

	JFrame frame;
	private JTextField tx_id;
	private JPasswordField tx_pw;
	private JTextField tx_name;
	private JTextField tx_phon1;
	private JTextField tx_phon2;
	private JTextField tx_mail1;
	private JTextField tx_mail2;
	String id,pw,name,mail1,mail2,gender;
	int year,month,day,phon1;
	String phon2,phon3;
```

```
public join() {
	initialize();
}


/**
 * Initialize the contents of the frame.
 */
void initialize() {

	frame = new JFrame();
	frame.setBounds(100, 100, 527, 339);
	//frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	frame.getContentPane().setLayout(null);

	JPanel panel = new JPanel();
	panel.setBounds(12, 30, 358, 59);
	frame.getContentPane().add(panel);
	panel.setLayout(null);

	JLabel lblNewLabel = new JLabel("WuC544WuC774WuB514WrWn");
	lblNewLabel.setBounds(12, 10, 36, 15);
	panel.add(lblNewLabel);

	JLabel lblNewLabel_1 = newJLabel("WuD328WuC2A4WuC6CCWuB4DCWrWn");
	lblNewLabel_1.setBounds(12, 35, 57, 15);
	panel.add(lblNewLabel_1);

	tx_id = new JTextField();
	tx_id.setBounds(81, 7, 154, 21);
	panel.add(tx_id);
	tx_id.setColumns(10);


	tx_pw = new JPasswordField();
	tx_pw.setBounds(81, 32, 154, 21);
	panel.add(tx_pw);
	tx_pw.setColumns(10);
```

```
JButton btnNewButton = new JButton("중복검사");
btnNewButton.setBounds(247, 6, 97, 23);
panel.add(btnNewButton);
btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {
                id = tx_id.getText();
                idcheck(id);
                }
});

JPanel panel_1 = new JPanel();
panel_1.setBounds(12, 125, 484, 139);
frame.getContentPane().add(panel_1);
panel_1.setLayout(null);

JLabel lblNewLabel_5 = new JLabel("이름");
lblNewLabel_5.setBounds(12, 10, 57, 15);
panel_1.add(lblNewLabel_5);

JLabel lblNewLabel_3 = new JLabel("성별");
lblNewLabel_3.setBounds(12, 35, 57, 15);
panel_1.add(lblNewLabel_3);

JLabel lblNewLabel_6 = new JLabel("생년월일");
lblNewLabel_6.setBounds(12, 60, 57, 15);
panel_1.add(lblNewLabel_6);

JLabel lblNewLabel_7 = new JLabel("휴대폰");
lblNewLabel_7.setBounds(12, 88, 57, 15);
panel_1.add(lblNewLabel_7);

JLabel lblNewLabel_8 = new JLabel("E_mail");
lblNewLabel_8.setBounds(12, 113, 57, 15);
panel_1.add(lblNewLabel_8);

tx_name = new JTextField();
tx_name.setBounds(81, 7, 154, 21);
panel_1.add(tx_name);
```

```
		tx_name.setColumns(10);

		tx_mail1 = new JTextField();
		tx_mail1.setBounds(81, 110, 154, 21);
		panel_1.add(tx_mail1);
		tx_mail1.setColumns(10);

		JComboBox comboBox_1 = new JComboBox();
		comboBox_1.setModel(new DefaultComboBoxModel(new String[]
{"1996", "1995", "1994", "1993", "1992", "1991", "1990", "1989"}));

		comboBox_1.setBounds(81, 57, 96, 21);
		panel_1.add(comboBox_1);
		comboBox_1.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
					year=(int) comboBox_1.getSelectedItem();

				}
		});		JComboBox comboBox_2 = new JComboBox();
		comboBox_2.setModel(new DefaultComboBoxModel(new String[] {"1",
"2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));
		comboBox_2.setBounds(210, 57, 57, 21);
		panel_1.add(comboBox_2);
		comboBox_2.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
					month=(int) comboBox_2.getSelectedItem();
				}
		});

		ButtonGroup group = new ButtonGroup();

		JRadioButton male = new JRadioButton("WuB0A8WrWn");
		male.setBounds(77, 31, 41, 23);
		panel_1.add(male);
		male.addMouseListener(new MouseAdapter() {
```

```
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        gender="남";

                }
        });

        JRadioButton female = new JRadioButton("WuC5EC");
        female.setBounds(122, 31, 41, 23);
        panel_1.add(female);
        female.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        gender="여";

                }
        });

        group.add(male);group.add(female);


        JComboBox comboBox = new JComboBox();
        comboBox.setBounds(81, 85, 57, 21);
        comboBox.setModel(new DefaultComboBoxModel(new String[]
{"010","011","016","070","019"}));

        panel_1.add(comboBox);
        comboBox.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        phon1=(int) comboBox.getSelectedItem();

                }
        });

        JLabel lblNewLabel_4 = new JLabel("-WrWn");
        lblNewLabel_4.setBounds(141, 88, 13, 15);
        panel_1.add(lblNewLabel_4);
```

```
		tx_phon1 = new JTextField();
		tx_phon1.setBounds(150, 85, 57, 21);
		panel_1.add(tx_phon1);
		tx_phon1.setColumns(10);
		tx_phon2 = new JTextField();
		tx_phon2.setBounds(219, 85, 57, 21);
		panel_1.add(tx_phon2);
		tx_phon2.setColumns(10);

		JLabel label = new JLabel("-WrWn");
		label.setBounds(210, 88, 13, 15);
		panel_1.add(label);
		JLabel lblNewLabel_9 = new JLabel("WuB144");
		lblNewLabel_9.setBounds(183, 60, 18, 15);
		panel_1.add(lblNewLabel_9);

		JLabel label_1 = new JLabel("WuC6D4");
		label_1.setBounds(271, 60, 18, 15);
		panel_1.add(label_1);

		JLabel label_2 = new JLabel("WuC77CWrWn");
		label_2.setBounds(363, 60, 18, 15);
		panel_1.add(label_2);
		JLabel label_3 = new JLabel("@");
		label_3.setBounds(247, 113, 20, 15);
		panel_1.add(label_3);

		tx_mail2 = new JTextField();
		tx_mail2.setForeground(Color.BLACK);
		tx_mail2.setBounds(271, 110, 203, 21);
		panel_1.add(tx_mail2);
		tx_mail2.setColumns(10);

		JComboBox comboBox_3 = new JComboBox();
		comboBox_3.setModel(new DefaultComboBoxModel(new String[] {"1",
"2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16",
"17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
"30", "31"}));
```

```
		comboBox_3.setBounds(300, 57, 57, 21);
		panel_1.add(comboBox_3);
		comboBox_3.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
					day=(int) comboBox_3.getSelectedItem();

				}
		});

		JButton btnNewButton_1 = new JButton("저장");
		btnNewButton_1.setBounds(127, 271, 97, 23);
		frame.getContentPane().add(btnNewButton_1);
		btnNewButton_1.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
					id=tx_id.getText();
		pw=tx_pw.getText();name=tx_name.getText();

phon2=tx_phon1.getText();phon3=tx_phon2.getText();

mail1=tx_mail1.getText();mail2=tx_mail2.getText();

newJoin(id,pw,name,gender,year,month,day,phon1,phon2,phon3,mail1,mail2);

			}
		});
		JButton btnNewButton_2 = new JButton("취소");
		btnNewButton_2.setBounds(263, 272, 97, 23);
		frame.getContentPane().add(btnNewButton_2);
		btnNewButton_2.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
					new login();
				}
		});
	}
```

```
	public void conect() {
		//db 연결
		try {
			Class.forName("org.mariadb.jdbc.Driver");
			// jdbc 드라이버를 로드한다.
			conn = DriverManager.getConnection(
					"jdbc:mariadb://localhost:3306/cjs",
"root", "12345");
			// db에 연결.

			// conn.close();//db연결 종료
		} catch (ClassNotFoundException e) {
			// TODO Auto-generated catch block
			System.out.println("class를 찾을수 없다.");
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			System.out.println(e.getMessage());
			e.printStackTrace();
		}
	}
	public void idcheck(String id){
		conect();
		System.out.println("DDDDD");
		String sql = "select * from cjstb where id=?";
		try {
			pstmt = conn.prepareStatement(sql);
			pstmt.setString(1, id);


			rs = pstmt.executeQuery();
			if (id.length() == 0) {
				JOptionPane.showMessageDialog(null, "id를
입력해주세요.");
				System.out.println("DDDDD");
			}
			if (rs.next()) {
				JOptionPane.showMessageDialog(null,
"이미사용중입니다.");
```

```
        conn.close();
                                //sstmt.close();
                                pstmt.close();System.out.println("DDDDD");
                                //setVisible(false);
                        } else {
                                JOptionPane.showMessageDialog(null,
"사용가능한아이디입니다.");
                                conn.close();
                                stmt.close();
                                pstmt.close();System.out.println("DDDDD");
                        }
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        }
        public void newJoin(String id,String pw,String name,String gender,
                        int year,int month, int day, int phon1,String phon2,
                        String phon3, String mail1,String mail2) {
                conect();

                System.out.println("접속");

                try {
                         stmt = conn.createStatement();
                        String sql = "insert into cjstb
(id,pw,name,gender,birth,phon,mail) values "
                                        +
"('"+id+"','"+pw+"','"+name+"','"+gender+"','"+year+month+day+"','"+phon1+phon2+p
hon3+"','"+mail1+mail2+"')";
                        stmt.execute(sql);
                        JOptionPane.showMessageDialog(null,
"회원가입되었습니다.");
                        conn.close();
                        stmt.close();
                        pstmt.close();
                        frame.setVisible(false);
                } catch (SQLException e) {
```

```
                System.out.println("회원가입 에러 :  " + e.getMessage());
                        e.printStackTrace();
package 졸작;
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
//아이디 공개키 받는 서버 프로그램
public class IdCertServer {
        @SuppressWarnings("resource")
        public static void main(String[] ar) throws Exception {
                ServerSocket svsoc = null;
                String str, str2, filename;

                svsoc = new ServerSocket(8888);
                System.out.println(" 서버가 준비되었습니다.");

                while (true) {
                        try {
                                System.out.println("공개키 받기 전 연결요청을
기다립니다.");  Socket soc = svsoc.accept();

                                // InputStream is = soc.getInputStream();
                                // br = new BufferedReader(new
InputStreamReader(is));
                                DataInputStream is = new
DataInputStream(soc.getInputStream());

                                str = is.readUTF();// str = br.readLine();

                                System.out.println("클라이언트가 보낸값1 : " +
str);

                                str2 = is.readUTF();// str2 = br.readLine();
                                System.out.println("클라이언트가 보낸값2 : " +
str2);

                                filename = is.readUTF();// filename =
br.readLine();
```

```
                                System.out.println("클라이언트가 보낸값3 : " +
filename);

                                File f = new
File("C:WWUsersWWPCWWDesktopWW인증서목록WW", filename + ".pkey");

                                /* 기록할 파일 연결함 */
                                FileOutputStream outf = new FileOutputStream(f);
                                BufferedOutputStream bos = new
BufferedOutputStream(outf);
                                /* 보내온 파일의 끝까지 읽어서 파일로 씀 */

                                int i = 0;
                                while ((i = is.read()) != -1) {
                                        bos.write((char) i);
                                        System.out.println(i);
                                }

                                /* 자원정리 */bos.flush();
                                /*
                                 * is.close(); soc.close(); svsoc.close();
bos.close();
                                 * outf.close();// br.close();
                                 */
                                System.out.println("받은파일 C:/ 경로에
저장됨!");
                                Centersend cs = new Centersend();
                                cs.ipset(str);

                                File ss = new
File("C:WWUsersWWPCWWDesktopWW인증서목록WW", filename + ".der");

                                // if.. 문 필요.. !!!!
                                if (ss != null) {
                                        GiveCertServer gcs = new
GiveCertServer();
                                        gcs.conect();
                                }
                        } catch (IOException e) {
                                e.printStackTrace();
```

```
package 졸작;

import java.awt.*;
import java.sql.*;
import javax.swing.*;
public class id {
	Connection conn = null;
	Statement stmt = null;
	PreparedStatement pstmt;
	ResultSet rs;

	 JFrame frmPw;
	 private JTextField tx_name;
	 private JTextField tx_age;
	 private JTextField tx_mail;
	 String name,age,mail;

	/**
	 * Launch the application.
	 */


	/**
	 * Create the application.
	 */
	public id() {
		initialize();
	}
	/**
	 * Initialize the contents of the frame.
	 */
	private void initialize() {
		frmPw = new JFrame();
		frmPw.setTitle("ID WuCC3EWuAE30");
		frmPw.setBounds(100, 100, 389, 231);
	//	frmPw.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frmPw.getContentPane().setLayout(null);

		JLabel lblNewLabel = new JLabel("이름");
```

```
lblNewLabel.setBounds(50, 50, 57, 15);
frmPw.getContentPane().add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("생년월일");
lblNewLabel_1.setBounds(50, 80, 57, 15);
frmPw.getContentPane().add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("메일");
lblNewLabel_2.setBounds(50, 110, 57, 15);
frmPw.getContentPane().add(lblNewLabel_2);



tx_name = new JTextField();
tx_name.setBounds(119, 47, 192, 21);
frmPw.getContentPane().add(tx_name);
tx_name.setColumns(10);

tx_age = new JTextField();
tx_age.setBounds(119, 77, 192, 21);
frmPw.getContentPane().add(tx_age);
tx_age.setColumns(10);

tx_mail = new JTextField();
tx_mail.setBounds(119, 107, 192, 21);
frmPw.getContentPane().add(tx_mail);
tx_mail.setColumns(10);

JButton btnNewButton = new JButton("ID 찾기");
btnNewButton.setBounds(50, 143, 261, 23);
frmPw.getContentPane().add(btnNewButton);
btnNewButton.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent arg0) {

                name = tx_name.getText();
                age = tx_age.getText();
                mail = tx_mail.getText();
                if (name.length() == 0||age.length() ==
```

```
0||mail.length() == 0) {
                                        JOptionPane.showMessageDialog(null,
"정확히 입력해주세요.");
                                }
                                else
                                        idfind(name,age,mail);
                                }
                });
        }
        public void conect() {
                //db 연결

                try {
                        Class.forName("org.mariadb.jdbc.Driver");
                        // jdbc 드라이버를 로드한다.
                        conn = DriverManager.getConnection(
                                        "jdbc:mariadb://localhost:3306/cjs",
"root", "12345");
                        // db에 연결.

                        // conn.close();//db연결 종료
                } catch (ClassNotFoundException e) {
                        // TODO Auto-generated catch block
                        System.out.println("class를 찾을수 없다.");
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        System.out.println(e.getMessage());
                        e.printStackTrace();
                }
        }
        public void idfind(String name,String age,String mail){
                conect();
                System.out.println(name+age+mail);
                //String sql = "select
                try {
                        String sql = "select id from cjstb where name=? and
birth=? and mail=?";
                        pstmt = conn.prepareStatement(sql);
                        pstmt.setString(1, name);
                        pstmt.setString(2, age);
```

```
                        pstmt.setString(3, mail);

                        rs = pstmt.executeQuery();
                        System.out.println(rs.next());

                        boolean t_f = rs.next();

                        if (t_f) {
                                //System.out.println(rs.getString(sql));
                                String temp= rs.getString("temp");
                                System.out.println(temp);
                                JOptionPane.showMessageDialog(null,
"이미사용중입니다.");
                                conn.close();
                                //sstmt.close();
                                pstmt.close();
                                //setVisible(false);
                        } else {
                                JOptionPane.showMessageDialog(null, "입력정보가
존재하지않습니다.");
                                conn.close();
                        //      stmt.close();
                                pstmt.close();
                                t_f=(Boolean) null;
                        }
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }//catch (NullPointerException e){
                        //JOptionPane.showMessageDialog(null, "입력정보가
존재하지않습니다.");
                //}
        }
}
package 졸작;

import java.io.*;
import java.net.*;
import java.security.*;
import java.security.cert.*;
```

```
public class gumServer {
	static byte[] dd;
	static String id;

	public static void main(String args[]) throws Exception{
		gumServer con = new gumServer();
		con.con();
	}

	public void con() throws Exception {
		ServerSocket serverSocket = null;

		serverSocket = new ServerSocket(3000);
		System.out.println(" 서버가 준비되었습니다.");

		while (true) {
			try {
				System.out.println(" 검증 연결요청을
기다립니다.");
				Socket socket = serverSocket.accept();

				InputStream out = socket.getInputStream();
				DataInputStream dos = new DataInputStream(out);
// 기본형 단위로 처리하는

					// 보조스트림
				OutputStream in = socket.getOutputStream();
				DataOutputStream dis = new DataOutputStream(in);
// 기본형 단위로 처리하는

					// 보조스트림

				byte[] bb = new byte[4096];

				int cc = 0;
				id = dos.readUTF();
				System.out.println(id);
				cc = out.read(bb);
				if (cc != 0) {
```

```
                                dd = new byte[cc];
                                for (int i = 0; i < cc; i++) {
                                        dd[i] = bb[i];
                                        System.out.println(dd[i]);
                                }
                        }

                        dis.writeUTF(gumju());
                        out.close();
                        socket.close();
                        dis.close();
                        dos.close();

                } catch (IOException e) {
                        e.printStackTrace();
                } // try - catch
        } // while
}


/*public static void main(String args[]) throws Exception {
        gumServer con = new gumServer();
        con.con();
}*/


public String gumju() throws Exception {

        String sigID = id;
        Signature sig = Signature.getInstance("MD5WithRSA");
        byte[] data = sigID.getBytes("UTF8");

        System.out.println("Wn저장 값 : " + sigID);

        System.out.print("받은 바이트 : ");
        for (byte b : dd) {
                System.out.print(b + ",");
        }
```

```
        CertificateFactory cf1 = CertificateFactory.getInstance("X.509");
                FileInputStream fis1 = new FileInputStream(new
File("C:₩₩Users₩₩PC₩₩Desktop₩₩인증서목록₩₩" + sigID
                                + ".der"));
                X509Certificate cert1 = (X509Certificate)
cf1.generateCertificate(fis1);
                fis1.close();

                sig.initVerify(cert1.getPublicKey());
                sig.update(data);

                boolean t_f = sig.verify(dd);

                if (t_f) {
                        System.out.println("인증서 검증되었습니다");
                        String a = "true";
                        return a;
                        // conect(a);
                }

                else {
                        System.out.println("인증서 검증실패되었습니다");
                        String a = "false";
                        return a;
                        // conect(a);
                }
        }
}
package 졸작;

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;


import javax.crypto.BadPaddingException;

import javax.crypto.IllegalBlockSizeException;
```

```
import javax.crypto.NoSuchPaddingException;

//server program (인증서 발급 )

public class GiveCertServer {
/*      public  static void main(String[] args) throws NoSuchPaddingException,
IllegalBlockSizeException, BadPaddingException, IOException{

        }*/
        public void conect() throws IOException,
                        NoSuchPaddingException, IllegalBlockSizeException,
                        BadPaddingException {
                System.out.println("인증서 발급 준비중");
                ServerSocket svsoc = null;
                Socket soc = null;
                String str="한솔";

                svsoc = new ServerSocket(7778);
                soc = svsoc.accept();

                // InputStream is = soc.getInputStream();
                // br = new BufferedReader(new InputStreamReader(is));
                DataInputStream is = new DataInputStream(soc.getInputStream());

                str = is.readUTF();// str = br.readLine();

                System.out.println("클라이언트가 보낸값1 : " + str);

                DataInputStream dis = new DataInputStream(new FileInputStream(new
File("C:WWUsersWWPCWWDesktopWW인증서목록WW" + str + ".der")));
                DataOutputStream dos = new
DataOutputStream(soc.getOutputStream());


                // 바이트단위로 읽어서 스트림으로 쓰기
                int b = 0;
                while ((b = dis.read()) != -1) {
```

```
        dos.writeByte(b);
                        System.out.println(b);
                        dos.flush();
                }

                /* 자원정리 */
                is.close();
                dis.close();
                dos.close();
                soc.close();
                svsoc.close();
                System.out.println("전송완료!");
        }
}
package 졸작;

import java.sql.*;

import javax.swing.JOptionPane;
public class DBjoin extends join{
        Connection conn = null;
        Statement stmt = null;
        PreparedStatement pstmt;
        ResultSet rs;
        public void conect(){

                try {
                        Class.forName("org.mariadb.jdbc.Driver");
                        //jdbc 드라이버를 로드한다.
                        conn = DriverManager.getConnection
                        ("jdbc:mariadb://localhost:3306/cjs","root","12345");
                        //db에 연결.
                        System.out.println(" 접속");
                        //conn.close();//db연결 종료
                } catch (ClassNotFoundException e) {
                        // TODO Auto-generated catch block
```

```
                        System.out.println("class를 찾을수 없다.");
                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        System.out.println(e.getMessage());
                        e.printStackTrace();
                }
        }
        public void join(String id,String pw){
                conect();

                try {
                        String sql="select * from login where name=? and password
=?";
                        pstmt = conn.prepareStatement(sql);
                        pstmt.setString(1,id);
                        pstmt.setString(2,pw);

                        rs =pstmt.executeQuery();

                        if(rs.next()){

                        }

                } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        System.out.println(e.getMessage());
                        e.printStackTrace();
                }
        }
}
package 졸작;

import java.io.*;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.math.BigInteger;
import java.security.*;
```

```
import java.security.cert.*;
import java.security.spec.X509EncodedKeySpec;
import java.util.*;
import javax.crypto.*;
import javax.security.auth.x500.X500Principal;
import org.bouncycastle.asn1.x509.*;
import org.bouncycastle.jce.provider.BouncyCastleProvider;
import org.bouncycastle.x509.X509V3CertificateGenerator;
import org.bouncycastle.x509.extension.AuthorityKeyIdentifierStructure;
import org.bouncycastle.x509.extension.SubjectKeyIdentifierStructure;



                //server 프로그램 (인증서 발급)


@SuppressWarnings("deprecation")
public class Centersend {
        static String name;


        //인증서 생성 in server
        enum CertType {
                ROOT, ENDENTITY
        }; // 인증서의 종류: 루트인증기관 인증서, 중간 인증기관 인증서,
개인인증서

        // 함수 3개 선언
        // 1. RSA 키생성 함수 - KeyPair를 리턴
        public static KeyPair generateRSAKeyPair() throws
NoSuchAlgorithmException {
                KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
                kpg.initialize(1024);
                return kpg.genKeyPair();
        }
        // 2. 인증서 생성 함수
        public static X509Certificate generateCertificate(X500Principal
subjectDN, // 주체
                        PublicKey pubKey, // 공개키
                        PrivateKey signatureKey, // 발급키 (발급자의 서명키)
```

```
                        X509Certificate caCert, // 발급자 인증서
                        CertType type)
                        // 인증서 종류
                        throws CertificateEncodingException,
NoSuchProviderException,
                        NoSuchAlgorithmException, SignatureException,
InvalidKeyException,
                        CertificateParsingException {

                X509V3CertificateGenerator certGen = new
X509V3CertificateGenerator();

certGen.setSerialNumber(BigInteger.valueOf(System.currentTimeMillis()));
                // 인증서 일련번호를 현재시간정보로부터 설정

                if (type == CertType.ROOT) // 루트인증서인 경우 발급자와 사용자가
똑같이 루트임
                        certGen.setIssuerDN(subjectDN);
                else
                        // 일반 사용자 인증서인 경우 발급자는 함수에 입력된 값을
이용. caCert에 있는 subjectDN을 이용
                        certGen.setIssuerDN(caCert.getSubjectX500Principal());
                certGen.setSubjectDN(subjectDN); // 사용자(주체)의 DN을 설정
                GregorianCalendar currentDate = new GregorianCalendar(); //
발급시간은 현재

                                                        // 시간으로
                GregorianCalendar expiredDate = new GregorianCalendar(
                                currentDate.get(Calendar.YEAR) + 1,
                                currentDate.get(Calendar.MONTH),
                                currentDate.get(Calendar.DAY_OF_MONTH));
                // 만료시간, 인증서의 유효기간은 1년으로 설정했음

                certGen.setNotBefore(currentDate.getTime()); // 유효기간 시작
설정
                certGen.setNotAfter(expiredDate.getTime()); // 유효기간 만료 설정
                certGen.setPublicKey(pubKey); // 공개키 설정
                certGen.setSignatureAlgorithm("SHA1withRSAEncryption"); //
서명알고리즘 설정
```

```
            if (type != CertType.ROOT) { // 루트인증서인 경우의 확장영역

certGen.addExtension(X509Extensions.AuthorityKeyIdentifier, false,
                                    new
AuthorityKeyIdentifierStructure(caCert));
                    certGen.addExtension(X509Extensions.SubjectKeyIdentifier,
false,
                                    new
SubjectKeyIdentifierStructure(pubKey));
            } else
                    // 일반 사용자인 경우의 확장영역. 키의 사용용도는
전자서명용, 키암호화용으로 사용 가능
                    certGen.addExtension(X509Extensions.KeyUsage, true, new
KeyUsage(
                                    KeyUsage.digitalSignature |
KeyUsage.keyEncipherment));
            return certGen.generate(signatureKey, "BC"); // 인증서를 생성하여
결과로 리턴
        }

        public static void ipset(String str) throws IOException,
                    NoSuchPaddingException, IllegalBlockSizeException,
                    BadPaddingException {
            // BouncyCastle Provider 추가
            name = str;
            Security.addProvider(new BouncyCastleProvider()); // 프로바이더
추가
            // 1. 키생성 및 인증서 발급
            // 루트인증기관 인증서는 자신의 개인키를 이용하여 발행
            // 개인의 인증서는 루트인증기관의 개인키를 이용하여 발행
            System.out.println("Wn* 1. 인증서 생성 ");
            try {
                    FileInputStream fis;FileOutputStream fos;

                    //루트 인증서 읽어오기
                    CertificateFactory cf =
CertificateFactory.getInstance("X.509");
                    fis = new FileInputStream(new
File("C:WWUsersWWPCWWDesktopWW인증서목록WW"+"root.der"));
```

```
                        X509Certificate rootCert = (X509Certificate) cf
                                        .generateCertificate(fis); // 파일에서
읽어서 인증서 형식으로 할당
                        System.out.println("₩n* root인증서 불러오기함 ");

                        fis.close();
                        // root인증기관의 개인키 읽어오기
                        KeyStore ks;
                        char[] code = {'s','e','c','r','e','t','c','o','d','e'};
// 개인키 암호화를 위한 키 코드 설정. 편의상 고정하였음
                        fis = new FileInputStream(new
File("C:₩₩Users₩₩PC₩₩Desktop₩₩인증서목록₩₩"+"root.key"));
                        ks = KeyStore.getInstance(KeyStore.getDefaultType());
                        ks.load(fis,code);  // 파일에서 읽어와서 키스토어에 로드
                        fis.close();
                        System.out.println("₩n* root 개인키 불러옴 ");
                        PrivateKey rootPrivateKey =
(PrivateKey)ks.getKey("RootPrivateKeyAlias",code);
                                                System.out.println(name);
                        //사용자의 공개키 읽어오기
                        fis = new FileInputStream(new
File("C:₩₩Users₩₩PC₩₩Desktop₩₩인증서목록₩₩"+name+".pkey"));
                        int size = fis.available();
                        byte[] s = new byte[size];
                        fis.read(s);
                        fis.close();
                        X509EncodedKeySpec pubKeySpec = new
X509EncodedKeySpec(s);
                        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
                        PublicKey userPublicKey =
keyFactory.generatePublic(pubKeySpec);
                        System.out.println(userPublicKey);
                        System.out.println("₩n* 사용자 공개키 불러옴 ");
                        // 인증서 생성하기
                        X509Certificate userCert = generateCertificate(new
X500Principal(
                                        "C=KR,CN=" + name), userPublicKey,
rootPrivateKey,
                                        rootCert, CertType.ENDENTITY);
```

```
// 인증서를 파일로 저장하기
                        fos = new FileOutputStream(new
File("C:₩₩Users₩₩PC₩₩Desktop₩₩인증서목록₩₩"+name+".der"));
                        fos.write(userCert.getEncoded()); // 파일로 저장
                        fos.close();
                        System.out.println("₩n*이용자 인증서 저장완료 ");
                        System.out.println("인증서생성완료");


                } catch (Exception e) {
                        System.out.println("₩n* 실패 ");

                }
        }
}
package jrdesktop.server;
import java.awt.AWTException;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;
import java.awt.event.MouseEvent;
import java.awt.event.MouseWheelEvent;
import java.awt.image.BufferedImage;
import java.util.ArrayList;
import jrdesktop.utilities.ImageUtility;
/**
 * robot.java
 * @author benbac
 */
public class robot {
    private Robot rt;
    private Toolkit tk = null;
    private Rectangle screenRect;

    public robot() {
        tk = Toolkit.getDefaultToolkit();
        screenRect = new Rectangle(tk.getScreenSize());
```

```
try {
            rt = new Robot();
        }
        catch (AWTException awte) {
            awte.getStackTrace();
        }
    }
    public BufferedImage captureScreen() {
        screenRect = new Rectangle(tk.getScreenSize());
        return rt.createScreenCapture(screenRect);
    }
    public byte[] CaptureScreenByteArray() {
        return ImageUtility.toByteArray(captureScreen());
    }

    public Rectangle getScreenRect() {
        return screenRect;
    }

    public void updateData(Object object) {
        ArrayList Objects = (ArrayList) object;
        for (int i=0; i<Objects.size(); i++) {
            Object obj = Objects.get(i);
            if (obj instanceof MouseEvent)
                applyMouseEvent((MouseEvent)obj);
            else if (obj instanceof KeyEvent)
                applyKeyEvent((KeyEvent)obj);
        }
    }
    public void applyMouseEvent(MouseEvent evt) {
        rt.mouseMove(evt.getX(), evt.getY());
        int buttonMask = 0;
        int buttons = evt.getButton();
        if ((buttons == MouseEvent.BUTTON1)) buttonMask =
InputEvent.BUTTON1_MASK;
        if ((buttons == MouseEvent.BUTTON2)) buttonMask |=
InputEvent.BUTTON2_MASK;
        if ((buttons == MouseEvent.BUTTON3)) buttonMask |=
InputEvent.BUTTON3_MASK;
        switch(evt.getID()) {
```

```
   case MouseEvent.MOUSE_PRESSED: rt.mousePress(buttonMask); break;
            case MouseEvent.MOUSE_RELEASED: rt.mouseRelease(buttonMask); break;
            case MouseEvent.MOUSE_WHEEL: rt.mouseWheel(
                    ((MouseWheelEvent) evt).getUnitsToScroll()); break;
        }
    }

    public void applyKeyEvent(KeyEvent evt) {
        switch(evt.getID()) {
            case KeyEvent.KEY_PRESSED: rt.keyPress(evt.getKeyCode()); break;
            case KeyEvent.KEY_RELEASED: rt.keyRelease(evt.getKeyCode()); break;
        }
    }
}
```

[Client]

```
package 졸작;
import java.io.*;
//import java.net.ServerSocket; 서버소켓 필요없음
import java.net.Socket;
import java.net.UnknownHostException;
import java.security.KeyPair;


//클라이언트 프로그램
public class CertInfo   {
                String name;
                String pw;
                int n;
                KeyPair aliceKeyPair;
        CertInfo() throws UnknownHostException, IOException {
                Socket soc = null;
                InputStream in = null;
                OutputStream out = null;
}
        void set(String a, String b) throws UnknownHostException, IOException{
                name = a;
                pw = b;
                contect();
        }
```

```
void contect() throws UnknownHostException, IOException {

                Socket soc = null;

                  BufferedWriter bw = null;

                  soc = new Socket("192.168.0.23",8000);
                  DataInputStream dis=new DataInputStream(new FileInputStream(new
File("c:₩₩",name+".pkey")));
                    DataOutputStream dos=new
DataOutputStream(soc.getOutputStream());

                //  bw= new BufferedWriter(new
OutputStreamWriter(soc.getOutputStream()));
                  dos.writeUTF(name/*+"₩n"*/);//문자열 보낼때 개행문자 반드시
들어가야함
                //  bw.flush();//write에 문자열을 보낼때는 flush 해줘야 제대로
문자열이 전달됨
                  dos.writeUTF(pw/*+"₩n"*/);//문자열 보낼때 개행문자 반드시
들어가야함
                  //bw.flush();//write에 문자열을 보낼때는 flush 해줘야 제대로
문자열이 전달됨

                  dos.writeUTF(name/*+".pkey"+"₩n"*/);//bw.flush();
                  /* 선택한 파일로 부터 스트림을 읽어들여서 얻어놓은
OutputStream에 연결 */
                   // 바이트단위로 읽어서 스트림으로 쓰기
                    int b=0;
                    while( (b=dis.read()) != -1 ){
                     dos.writeByte(b);System.out.println(b);
                     dos.flush();
                    }
                    System.out.println("전송완료");
                   /* 자원정리 */
                    dis.close(); dos.close(); soc.close();

                    dis=null; dos=null; soc=null;
                        GetCertClient gcc =     new GetCertClient(name);
```

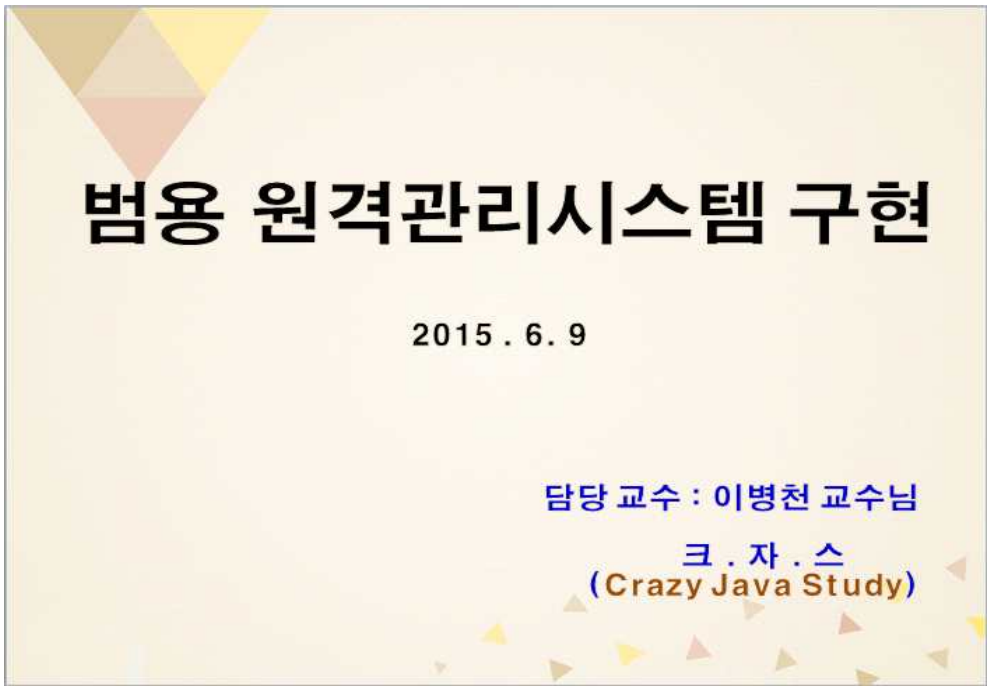
```
                gcc.conect();
                  }
}
package 졸작;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.*;
import java.security.*;
import java.text.*;
import java.util.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.event.*;
//검증 클라이언트
public class CertLogin {
        static String a;
        JFrame frame;
        private JTextField Choice;
        private JPasswordField PW;
        static String id;
        private PrivateKey alicePrivateKey;
        private KeyStore keyStore;
        private JFileChooser fileChooser = new JFileChooser();
        private File selectedFile;
        private String plainText;

        /*
         * public static void main(String[] args) { EventQueue.invokeLater(new
         * Runnable() { public void run() { try { CertLogin window = new
         * CertLogin(); window.frame.setVisible(true); } catch (Exception e) {
         * e.printStackTrace(); } } }); }
         */
        public CertLogin(String ID) throws Exception {

                initialize(ID);
        }

        private void initialize(String ID) throws Exception {
                frame = new JFrame();
```

2 발표 자료


범용 원격관리시스템 구현
2015 . 6. 9
담당 교수 : 이병천 교수님
크 . 자 . 스
(Crazy Java Study)


목 차
❍ 조원 역할
❍ 주제 선정 이유
❍ 구상도
❍ 추진경과
❍ 운영환경 및 개발프로그램
❍ 개발결과 및 운영절차
❍ 결론 및 기대효과

조원 역할

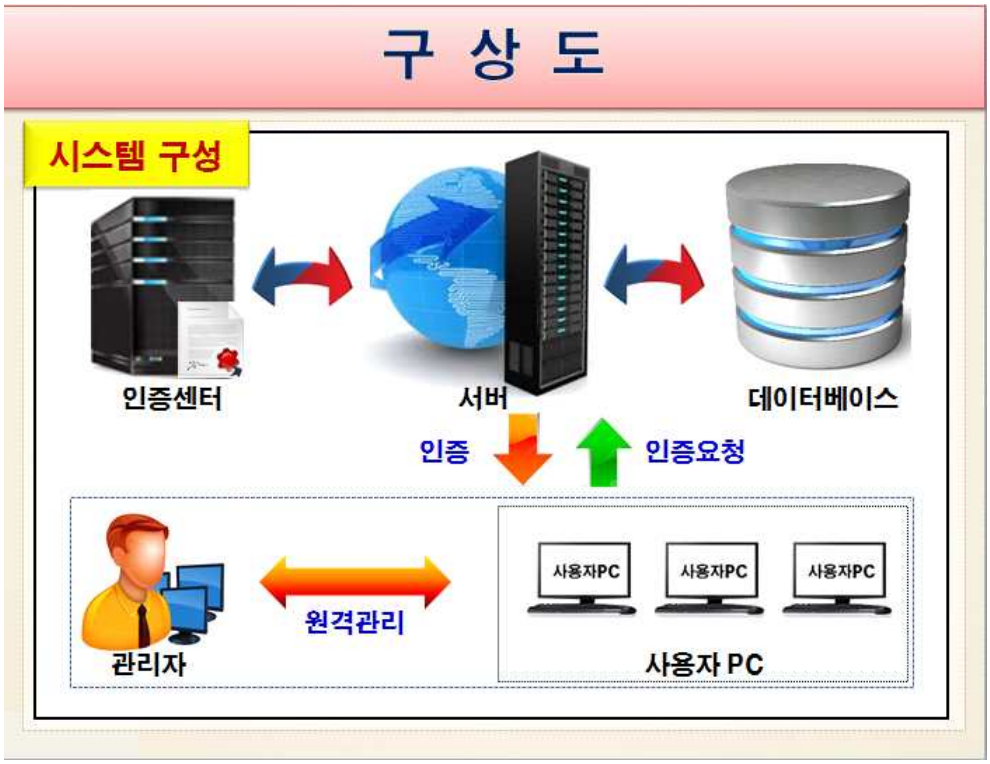
구 분	역 할
조장 한 솔	작품 총괄 및 프로그램 개발
조원 안명진	프로그램 개발 및 자료 조사
조원 양윤태	프로그램 개발 및 DB 구축
조원 안병훈	자료 조사 및 소스 분석

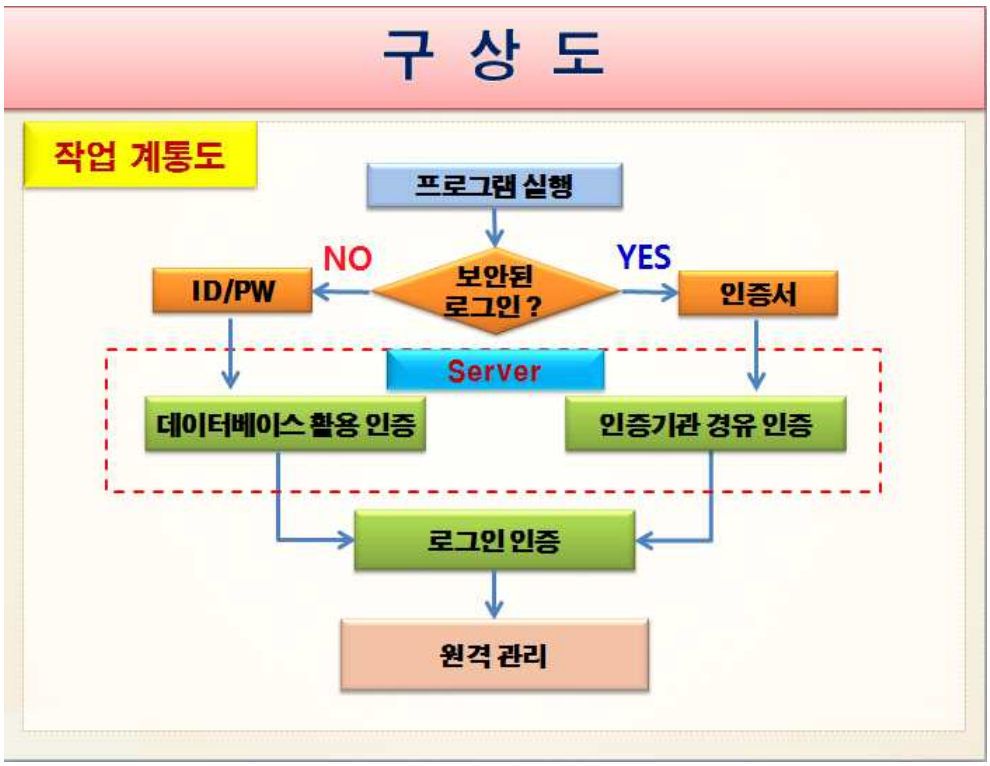
주제 선정 사유

❍ 다수의 PC를 원격관리하는 시스템에서 개별 PC의 이상 여부를 현장 방문 점검하는 경우 운영 관리의 효율성이 저하

❍ PC를 활용한 강의 시 피교육자들이 카톡 등 개인용 앱이나 게임 등을 무분별하게 사용, 강의 분위기 훼손 및 효과 감소

⇨ 개별 PC를 모니터링하는 범용 원격관리시스템을 개발, 운영관리 효율성을 높이고 개인용 프로그램 등의 무분별한 활용을 억제


구 상 도
시스템 구성
인증센터
서버
데이터베이스
인증
인증요청
관리자
원격관리
사용자PC
사용자PC
사용자PC
사용자 PC


구 상 도
작업 계통도
프로그램 실행
NO
보안된
로그인 ?
YES
ID/PW
인증서
Server
데이터베이스 활용 인증
인증기관 경유 인증
로그인 인증
원격 관리

추진 경과

구 분	기 간(2014.9 ~ 2015.6)									
	9	10	11	12	1	2	3	4	5	6
주제선정 및 자료조사	■									
DB구조 및 설계		■								
JDBC 연동			■	■						
GUI 프로그래밍				■	■	■	■			
Socket 프로그래밍					■	■	■			
Robot 클래스 구현						■	■	■		
인증서 구현							■	■	■	
PPT 및 보고서 완성										■

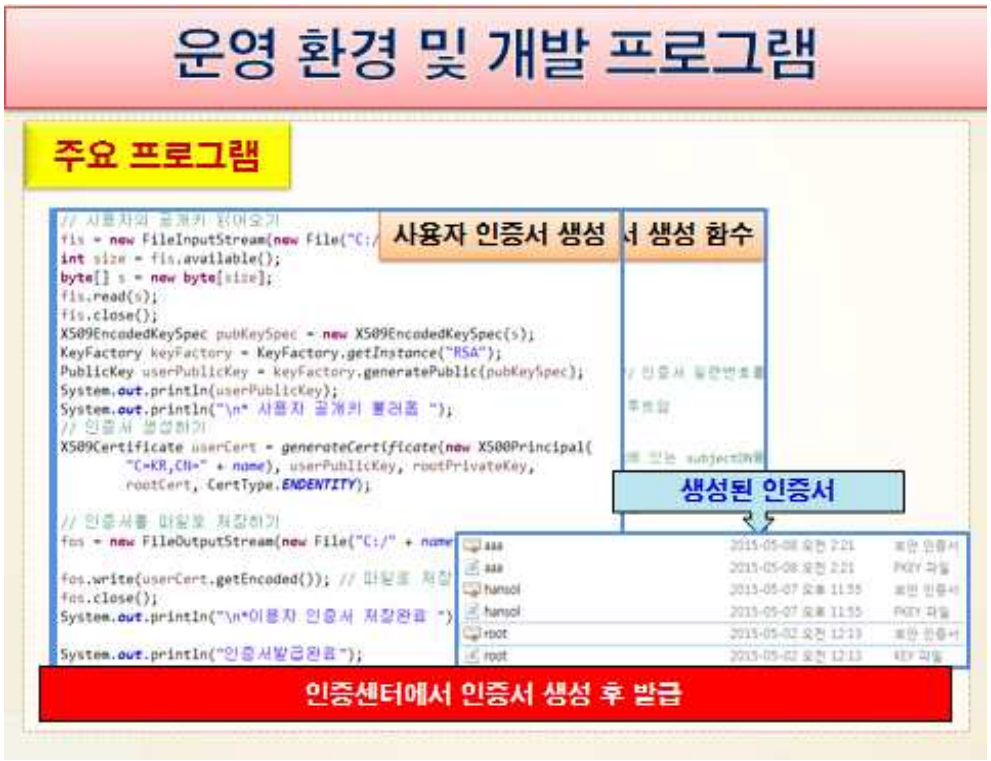
운영 환경 및 개발 프로그램

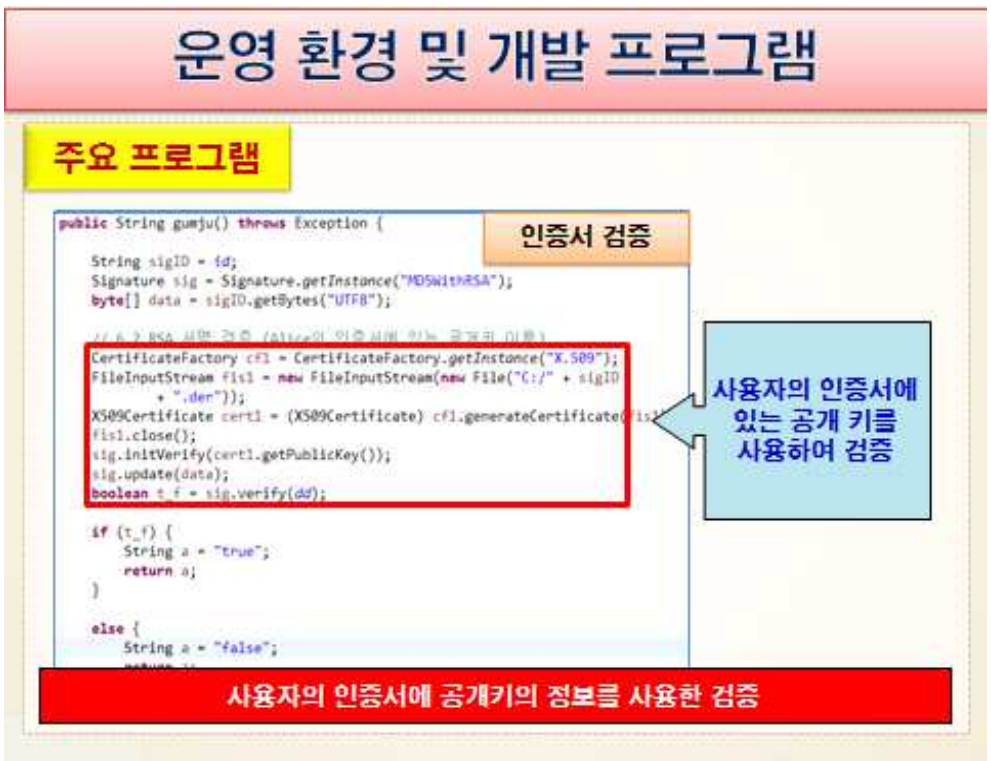
개발 환경

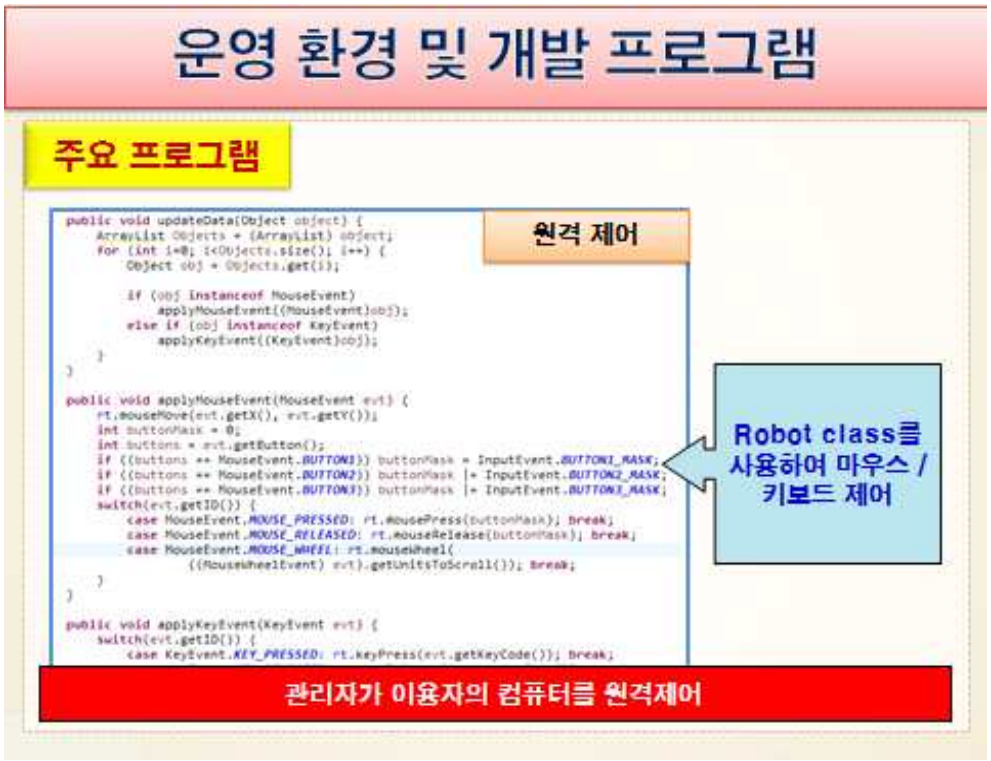
❍ 운영 환경 ⇨ Window 7

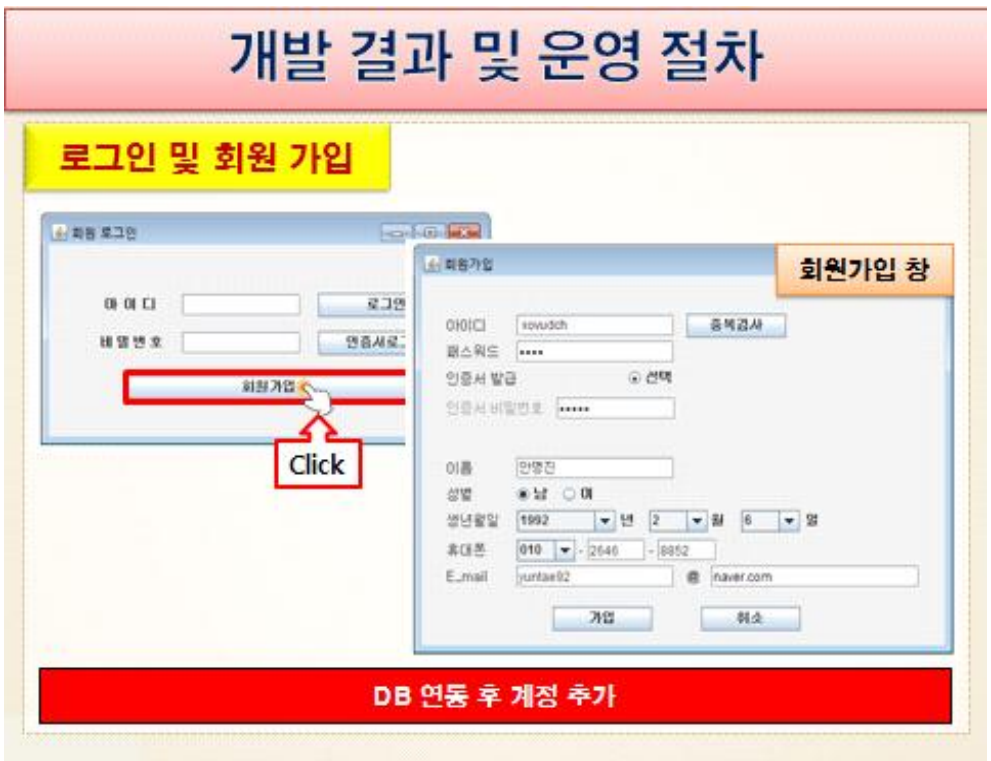
❍ 개발 체제 ⇨ 이클립스 EE
- Java

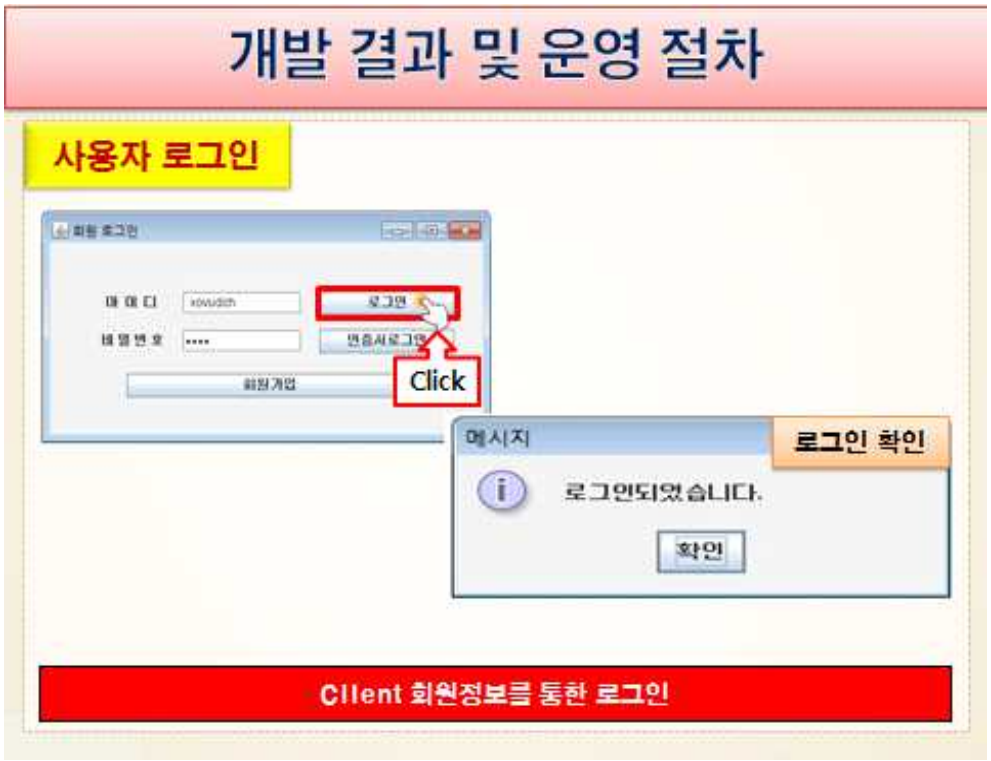
❍ 데이터베이스 ⇨ MariaDB

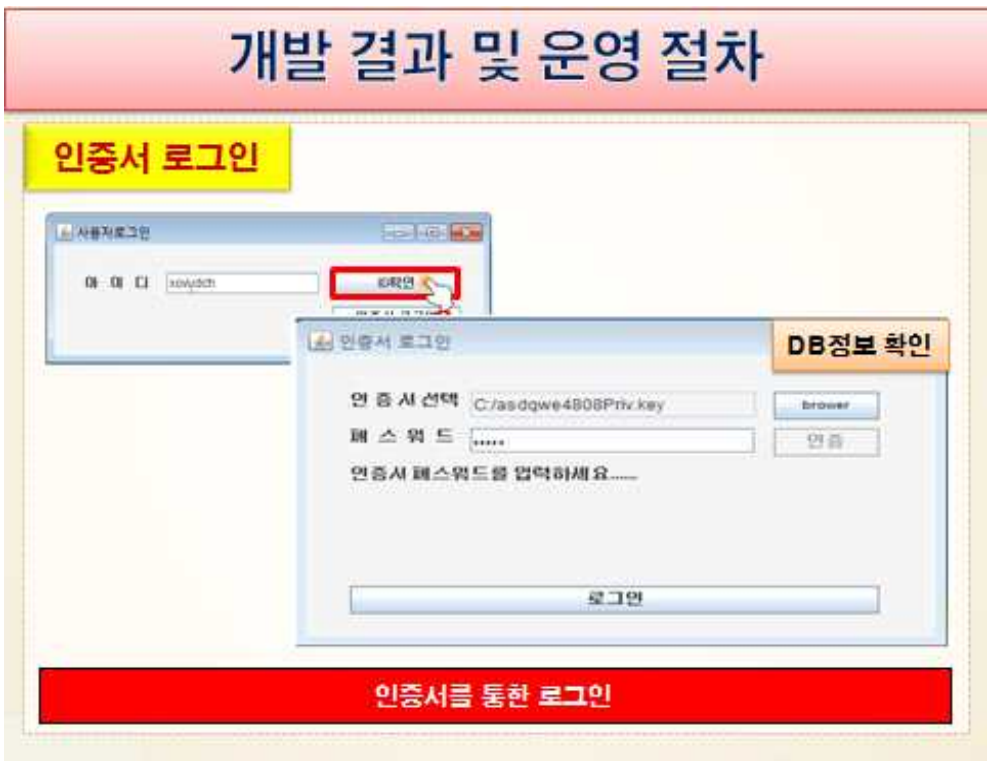

운영 환경 및 개발 프로그램
주요 프로그램
사용자 인증서 생성
생성된 인증서
인증센터에서 인증서 생성 후 발급


운영 환경 및 개발 프로그램
주요 프로그램
인증서 검증
사용자의 인증서에 있는 공개 키를 사용하여 검증
사용자의 인증서에 공개키의 정보를 사용한 검증


운영 환경 및 개발 프로그램
주요 프로그램
원격 제어
Robot class를 사용하여 마우스 / 키보드 제어
관리자가 이용자의 컴퓨터를 원격제어


개발 결과 및 운영 절차
로그인 및 회원 가입
회원가입 창
Click
DB 연동 후 계정 추가


개발 결과 및 운영 절차
사용자 로그인
Click
메시지
로그인되었습니다.
확인
로그인 확인
CIlent 회원정보를 통한 로그인


개발 결과 및 운영 절차
인증서 로그인
인증서 로그인
인 증 서 선택
C:/asdqwe4808Priv.key
brower
패 스 워 드
인증
인증서 패스워드를 입력하세요.....
로그인
DB정보 확인
인증서를 통한 로그인

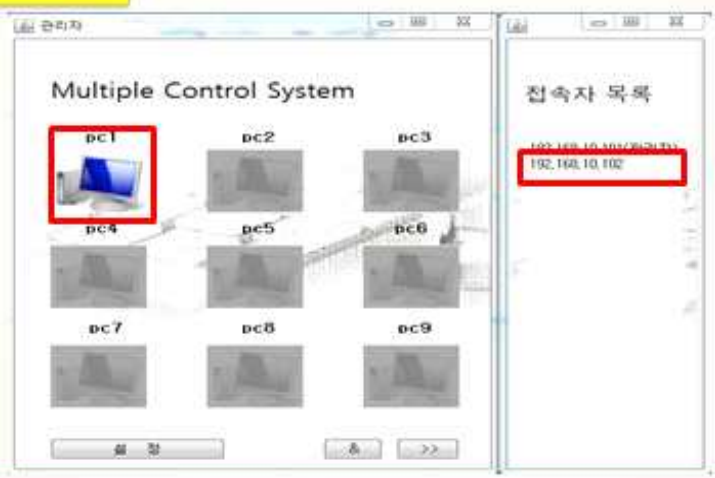
개발 결과 및 운영 절차

사용자 접속

Client 로그인 후 프로그램 실행(제어 허용)

개발 결과 및 운영 절차

관리자 화면



사용자 접속 시 버튼 활성화 및 접속자 목록 추가

개발 결과 및 운영 절차

결론 및 기대효과

○ 범용 원격관리 시스템 개발 성과

- 개별 PC를 원격 제어하고 모니터링함으로써 정상 작동 여부 및 이상 상황 점검·대응이 가능하여 분산된 PC 운영을 효율화

- 사용자들의 비인가 프로그램 등록 사용 등 점검 및 통제 가능

○ 기대 효과

- 사용자 인증에 패스워드 및 인증서 로그인 이중화 체제를 적용, 비인가자 접근통제 등 보안 기능을 대폭 보강

- 모든 조원들이 임무를 분담하여 인증서 로그인 등을 직접 구현, 보안 소프트웨어 개발 기량을 향상시키는 계기


Q&A
THANK YOU