

# NFC를 활용한 출입통제 시스템

팀 명 : Summit  
팀 장 : 김덕경(09)  
팀 원 : 김정수(09)  
신민기(09)  
양지현(11)  
담당교수 : 김대호 교수님

2014. 5

중부대학교 정보보호학과

# 목 차

1. 서론 .....	2
2. 관련 연구	
2.1 NFC .....	2
2.2 안드로이드 .....	3
2.3 자바 .....	4
2.4 PHP .....	5
2.5 MySQL .....	6
3. 연구 내용 .....	7
3.1 프로그램 구현 .....	7
3.2 프로그램 소스 .....	10
4. 결론 .....	31
4.1 향후 과제 .....	31
5. 참고문헌 .....	31
6. 발표 PPT 자료 .....	32

## 요약

최근 스마트폰의 NFC(Near Field Communication)기능이 광고, 티켓팅, 의료서비스, 안내/제품정보, 결제 및 계좌이체, 개인인증 등 많은 분야에서 활용되고 있다. 그 중 개인 인증분야의 출입문 개폐(도어락)를 보면 현재 대부분의 회사에서의 출입통제에 RFID(Radio Frequency Identification) 카드(사원증)와 리더기를 이용하고 있다. 그러나 이러한 RFID 카드를 사용하는 출입통제는 분실의 위험이 다소 높으며 리더기에 사용자 정보를 저장함으로서 악의적인 해킹에 노출되어 있고, 또 리더기 설치에 많은 비용이 발생하기 때문에 NFC를 활용한 출입통제 시스템에서는 기존 출입 단말기 대신 NFC 태그를, 기존 사원증 형태의 RFID 카드 대신 스마트폰의 NFC 기능을 이용하여 출입통제를 할 수 있도록 설계 및 구현하였다.

NFC를 활용한 출입통제 시스템을 구현하기 위해서 관리자, 사용자 스마트폰 어플리케이션을 구현하고 Apache, PHP, MySQL을 이용해 서버와 데이터베이스를 구축하였으며 관리자 프로그램은 자바를 이용하여 구현하였다.

## 1. 서론

정보통신 기술의 발전으로 여러 기술이 융합된 다양한 서비스가 출현하고, 언제 어디서나 서비스를 상용할 수 있는 응용 서비스들이 활발하게 정의 되고 있으며, 무선네트워크 기술에도 도약적인 발전을 가져왔다. 이러한 발전은 금융, 통신, 유통 등 다양한 응용 서비스들이 활발하게 정의 되고 있다. 스마트폰, 태블릿 PC 등 2010년부터 폭발적으로 보급 확산된 스마트 디바이스의 영향으로 모바일 결제 분야에 가장 핫 이슈로 떠오른 기술이 바로 NFC 기술이다. 사실, NFC기술은 2004년부터 표준이 개발되었지만 이를 적용할 디바이스 및 서비스의 부재로 활용이 되고 있지 않다가 2011년 구글이 구글 웰렛 서비스를 시작하면서 본격적으로 NFC 기반 서비스가 적용되기 시작하였다. 이러한 스마트 디바이스의 확산과 무선 네트워크의 발전은 우리의 일상생활을 윤택하고 편리하게 변화시켜 나가고 있으며 앞으로도 다양한 분야로 지속적으로 확산되어 갈 것으로 예상된다. 또한, NFC기반 통신 기능을 채용한 스마트기기 및 OS 등이 확산됨에 따라 기존 RFID를 이용하던 다양한 응용 분야에 대하여 NFC로의 대체가 진행되고 있다. 개인 인증 분야에서도 기존 RFID 카드 대신 NFC로 대체되고 있는데 NFC를 활용한 출입통제 시스템은 스마트폰의 기능 중 하나인 NFC를 이용하여 사원증 소지의 불편함과 리더기 설치 시의 비용, 기존 리더기에 사용자 정보를 저장함으로서 악의적인 해킹에 노출되어 있는 문제를 해소하고자 출입통제 시스템을 설계 및 구현하였다.

## 2. 관련 연구

### 2.1 NFC

NFC는 기존 RFID의 확장된 개념으로 Sony와 NXP가 공동 개발한 개방형 RFID 기술의 일종으로 이동통신 단말기에 탑재하여 다양한 부가서비스가 가능하고 13.56MHz 주파수 대역을 사용하여 비접촉식으로 10cm 이내의 단거리에서 기기간 데이터를 전송하는 근거리 무선 통신기술이다. 기존의 비접촉식 스마트카드 기술(ISO/IEC 14443

Proximity-card 표준)을 기반으로 개발되었다.

신용카드, 신분증 등을 대체할 수도 있으며, 노트북의 사용자 인증, 모바일 티켓, 쿠폰 등 다양한 분야에서 활용될 수 있는 성장 잠재력이 큰 기술이다. 정보의 읽기 뿐 아니라 쓰기도 가능한 양방향 통신을 지원 한다. 또한, 정보의 Encryption으로 타 RFID 보다 안정성이 뛰어나다. 최근, RFID 기술의 발달로 PassiveType과 ActiveType을 포함해서 최대 900m 까지 사용 거리가 늘어났으며 리더와 태그 개념을 적용해 비즈니스 모델에 따라 활용 범위가 크게 확대 될 것이라고 전망된다. 실생활과 연계된 복잡한 정보활동에 대한 해결책으로, 모든 타입의 사용자 장치에 대한 'Touch-and-Start' 조작으로 접촉과 실행에 대한 직관적 접촉 편의성을 제공하여 보안 처리가 내장된(Built-Insecurity) 지불/재정 등의 응용을 사용자들이 쉽게 사용할 수 있게 하였으며 스마트폰, AV(Audio Visual)장비, 디지털 카메라, PDA, 세트 탑 박스 등과 컴퓨터를 직관적으로 접속하게 하여, 전화번호, 그림, 티켓, MP3파일, 북마크 등의 모든 콘텐츠를 용이하게 동작 시킬 수 있다.

능동모드 혹은 수동모드에서 운영될 수 있어, 비 접촉 스마트카드나 RF 태그와 같은 다양한 수동형 장치와 폭넓게 통신할 수 있다. 또한, 휴대폰, 블루투스 그리고 무선 802.11 장치에 대한 무선 네트워크의 자동 설정과 무선 통신 서비스를 제공하는 가상 NFC 연결 시스템으로 연결함으로써 개방형 무선 인터페이스 플랫폼으로 자리매김 하고 있다.

NFC기술은 다양한 근거리 무선 통신들의 다양한 성질들을 결합하여 단말기의 ON/OFF와 관계없이 항상 리더기를 통해 인식이 가능한 카드 에뮬레이션 모드, NFC 활성화 상태에서 RFID Tag 정보를 인식하여 휴대폰이 카드 리더기로써 작동하는 리더/라이터 모드, 두 대의 NFC 휴대폰이 카드 리더기로써 작동하여 데이터를 상호간에 전송할 수 있는 P2P 모드로 작동 가능하다. 이 모드들은 RFID와 같이 데이터를 읽고 수정할 수 있는 모드, 블루투스 등과 연결하여 데이터 통신 서비스 가능 모드, 카드에 탑재되어 비접촉식 카드의 성질을 지원하면서 스마트 카드의 안정성을 기반으로 하는 안전한 서비스들이 지원 가능하다.

#### (1)Peer-to-Peer(P2P)모드

P2P 운영모드(ISO 18092)는 두 개의 NFC 디바이스간의 링크 수준의 통신을 지원한다.. 블루투스 페어링 절차를 NFC기술로 대체 하여 연결 초기 절차를 단순화 한다. 연결 확립을 위해 클라이언트(NFC Peer-to-Peer Initiator)는 호스트(NFC Peer-to-PeerTarget)를 검색하고 NDEF(NFC DataExchangeFormat)메시지 형식을 통해 데이터를 전송한다. 두 대의 NFC 휴대폰이 카드리더기로써 작동하여 데이터를 상호간에 전송 할 수 있는 모드이며, 또한 능동 모드로써 데이터 전송을 위해 독자적인 RF필드를 생성해야 하므로 전력 소모가 크다.

#### (2)Reader/Writer모드

NFC 디바이스는 NFC 트랜스폰더에 저장된 데이터를 읽고 수정할 수 있다. 사용자는 SmartPoster와 같이 NFC 디바이스가 스마트포스터의 태그를 읽어 추가정보를 조회할 수 있는 기술이다. URL 주소가 저장되어 있는 태그에서 NFC 디바이스를 터치하면 URL 주소를 읽고 그 주소의 웹사이트에 접근을 지원한다.

#### (3)Card Emulation모드

NFC 디바이스가 스마트 카드(ISO 14443)처럼 동작하는 모드이기 때문에 외부NFC 리더기는 스마트 카드와 NFC 디바이스를 구분할 수 없다. 이 모드에서는 비 접촉식 지불, 티켓팅 서비스가 가능하다.

## 2.2 안드로이드

안드로이드는 모바일 디바이스들을 위한 소프트웨어 스택(Stack)으로써, 운영체제(OS) 및 미들웨어(Middleware), 그리고 핵심 어플리케이션(Key Application)을 포함하고 있다. 또한 안드로이드는 특정 Vender에 종속되지 않고 개방되어 있어 무료라는 원칙을 가지고 있는 모바일 플랫폼이다. 이러한 오픈 플랫폼인 안드로이드의 대표적인 특징 5가지를 적어보았다.

### ① 리눅스 커널 2.6을 사용한다.

리눅스(Linux) 커널을 이용하여 운영체제가 구성되어 있으므로 안드로이드 플랫폼 또한 오픈 소스인 것이다. 대신에, 기존의 리눅스 커널은 GPL 라이선스를 사용하기 때문에 실제 Vender들이 사용하기에는 많은 제약 사항이 있었다. 하지만 안드로이드에서는 커널 등 일부를 제외한 나머지는 LGPL 라이선스보다도 낮은 BSD나 Apache 라이선스를 사용하기 때문에, Vender들이 상용화하기 쉽게 하였다는 장점이 있다.

### ② Java 언어의 사용

내부 커널은 리눅스이므로 역시 C언어를 사용하고 있고, 라이브러리의 경우는 C/C++을 사용하고 있으나 어플리케이션 프레임워크 부분부터는 Java를 사용하고 있기 때문에 안드로이드만의 전용 JVM인 Dalvik가 상머신을 통해서 어플리케이션을 구동하고 있다. Java언어와 C언어 사이는 JNI(Java Native Interface)를 이용해 연결하여 사용하고 있는데 안드로이드에서는 이를 NDK라는 형태로 자동화하여, 개발자가 사용할 수 있도록 지원하고 있다.

### ③ 다양한 라이브러리의 제공

안드로이드에서는 다양한 라이브러리를 포함하고 있으며, 이를 API로 제공하고 있다. OpenGL ES를 통해서 3D를 지원하고 있고 Webkit을 통해 웹 브라우저를 지원하며, FreeType을 통해서 폰트를 지원하고 있다. 이 외에도 다양한 센서 및 통신에 관한 라이브러리와 디바이스 드라이버들을 지원하고 있다. 여기서 안드로이드의 큰 특징 중 하나는 어플리케이션이 직접 하드웨어와 통신하는 것이 아니라 Binder를 통해 하드웨어와 통신함으로써 응용프로그램이 직접 Root권한을 얻지 않아도 되고, blocking 방식에서 문제가 발생해도 복구하기가 용이하다.

### ④ Eclipse에 기반한 개발환경 지원

Eclipse에 기반한 통합 개발환경을 지원한다. 특히 x86에서도 QEMU기반의 ARM 에뮬레이터를 통하여, 실제 폰과 같이 환경을 제공해주고 있다. 또한, 디버깅 및 메모리 및 성능 프로파일링을 지원하여, 어플리케이션 개발자가 손쉽게 개발할 수 있는 환경을 제공하고 있다.

### ⑤ Bionic C 라이브러리 사용

기존 리눅스와 같이 glibc를 사용하는 것이 아닌 Bionic C 라이브러리를 사용하고 있다. 따라서 기존 리눅스의 glibc를 이용한 소스를 그대로 컴파일하기에는 무리가 있어 호환성을 유지하고는 있기는 하나, 약간의 수정을 거쳐야 하는 경우가 있다.

## 2.3 자바

자바는 썬 마이크로시스템즈의 제임스 고슬링(James Gosling)과 다른 연구원들이 개발한 객체 지향적 프로그래밍 언어이며, 썬 마이크로시스템즈에서 무료로 제공하고 있다.

1991년 그린 프로젝트(Green Project)라는 이름으로 시작해 1995년에 발표했다. 처음에는 가전제품 내에 탑재해 동작하는 프로그램을 위해 개발했지만 현재 웹 어플리케이션 개발에 가장 많이 사용하는 언어 가운데 하나이고, 모바일 기기용 소프트웨어 개발에도 널리 사용하고 있다. 현재 버전 7까지 출시했다.

자바의 개발자들은 유닉스 기반의 배경을 가지고 있었기 때문에 문법적인 특성은 파스칼이 아닌 C++의 조상인 C 언어와 비슷하다. 자바를 다른 컴파일언어와 구분 짓는 가장 큰 특징은 컴파일된 코드가 플랫폼 독립적이라는 점이다. 자바 컴파일러는 자바 언어로 작성된 프로그램을 바이트코드라는 특수한 바이너리 형태로 변환한다. 바이트코드를 실행하기 위해서는 JVM(자바 가상 머신, Java Virtual Machine)이라는 특수한 가상 머신이 필요한데, 이 가상 머신은 자바 바이트코드를 어느 플랫폼에서나 동일한 형태로 실행시킨다. 때문에 자바로 개발된 프로그램은 CPU나 운영 체제의 종류에 관계없이 JVM을 설치할 수 있는 시스템에서는 어디서나 실행할 수 있으며, 이 점이 웹 어플리케이션의 특성과 맞아떨어져 폭발적인 인기를 끌게 되었다.

고슬링의 목표는 C/C++ 스타일의 언어와 가상 머신을 구현하는 것이었다. 첫 공개 자바 버전은 1995년의 자바 1.0 이었다. 한 번 쓰고 어느 곳에도 실행 "Write Once, Run Anywhere"하는 것을 약속하였고 인기 플랫폼에 무료 런타임을 제공하였다. 이 플랫폼은 꽤 안정성을 지녔고 보안 시스템은 여러 설정을 통해 네트워크 및 파일 접근을 통제할 수 있었다. 대부분의 브라우저들은 곧 자바 애플릿(Java Applet)을 웹 페이지 안에서 실행할 수 있었고 자바의 인기는 급상승했다. 자바2(JDK 1.2 ~ 1.4)의 출현으로 여러 다양한 플랫폼에서 사용할 수 있는 설정(Configuration)들을 만들었다. 예를 들어 J2EE는 엔터프라이즈 어플리케이션을 실행할 수 있고, J2ME는 모바일 어플리케이션을 실행할 수 있다. J2SE는 스탠다드 에디션으로 지정되었다. 2006년에 마케팅용으로 Java EE, Java ME, Java SE 순서대로 명칭을 바꾸었다.

썬 마이크로시스템즈는 1997년 ISO/IEC JTC1 표준화 그룹, 그리고 나중에는 ECMA International 그룹과 접촉하여 정식 승인을 받으려 했으나 곧 이 프로세스에서 철수했다. 자바는 사실상 (de facto)의 표준이며 자바 커뮤니티 프로세스(Java Community Process)를 통해 관리되고 있다. 썬 마이크로시스템즈는 자바의 대부분을 무료로 배포하였으나 오픈이 아닌 사유 소프트웨어였다. 썬 마이크로시스템즈는 자바 엔터프라이즈 시스템 같은 특정 라이선스를 팔아서 수입을 올렸다. 썬(SUN) 사는 자사의 개발용(소프트웨어 개발 키트 (SDK))과 개발용의 일부인 실행용(자바 런타임 환경, JRE)을 구분한다. 실행용에는 컴파일러, 유틸리티 및 여러 개발용 헤더 파일이 포함되어 있지 않다.

2006년 11월 13일 썬 마이크로시스템즈는 대부분은 자바를 GPL 라이선스로 소스를 오픈하였으며 2007년 5월 8일 이 과정을 마쳤다. 썬 마이크로시스템즈가 권한을 가지고 있지 않은 대부분의 코어 부분을 오픈하였다. 2009년 4월 20일 제작사인 썬 마이크로시스템즈가 오라클과 인수 합병됨에 따라 자바에 대한 권리 및 유지보수 또한 오라클로 넘어가게 되었다.

## 2.4 PHP

PHP는 웹 사이트 제작을 비롯한 웹 응용 프로그램을 개발하는 웹 프로그래밍 언어로, 일반적으로 HTML/CSS 및 자바 스크립트와 함께 사용된다. HTML/CSS는 포토샵을 비

롯한 다양한 응용 프로그램으로 작성된 이미지, 동영상, 사운드, 텍스트 등을 웹 페이지에 표현하는 스크립트 언어이다. HTML/CSS 코드로만 작성된 웹 페이지는 사용자 게시판, 쇼핑몰 장바구니, 배송 조회처럼 사용자와 상호작용하는 페이지(동적 웹 페이지)도 있기 때문에 HTML/CSS 외에 PHP와 같은 웹 프로그래밍 언어가 필요하다.

PHP는 1994년 라스머스 레돌프가 온라인 이력서를 작성하기 위해 만든 C 언어로 된 CGI(Common Gate Interface) 이진 코드에서 시작되었다. 라스머스는 작성한 코드에 Personal Home Page Tools라는 이름을 붙였고, 이를 줄여서 PHP Tools로 부르기 시작했다. 시간이 지나 사용자의 요구를 증가하면서 라스머스는 방명록과 같은 동적인 웹 페이지 제작을 위해 데이터베이스 인터페이스 같은 기능을 추가했다.

1995년 라스머스는 개발자들이 자유롭게 코드를 수정할 수 있도록 대중들에게 PHP Tools의 코드를 공개했다. PHP Tools는 Perl의 변수 방식을 차용하고, C 언어의 문법과 작동원리를 모방했다. 그리고 HTML 코드를 삽입할 수 있는 형태로 발전해 현재 사용하고 있는 PHP의 틀을 마련하였다. 실제로 PHP는 C 언어와 유사한 점이 많기 때문에 C 언어에 익숙한 사람이라면 누구나 쉽게 익힐 수 있다.

PHP는 윈도우에서만 동작하는 ASP.NET과 달리 윈도우는 물론이고 리눅스 같은 유닉스 계열의 운영체제에서도 사용할 수 있다. 그렇기 때문에 리눅스에서 구축 구한 웹 사이트의 서버를 윈도우 서버로 변경하더라도 코드 수정이 적다. 데이터베이스도 ASP가 Microsoft의 MSSQL만 사용할 수 있는 반면, PHP는 공개용 데이터베이스인 MySQL뿐만 아니라 Oracle, PostgreSQL, Sybase 등 다양한 데이터베이스를 지원한다.

PHP는 간편하게 설치할 수 있고 배우기도 쉬워 JSP보다 개발 기간이 짧다. 또한 내부 엔진이 가벼워 웹 사이트 운영시 처리 속도가 빠르며, 리눅스와 MySQL 데이터베이스가 무료라 비용 측면에서도 유리하기 때문에 중소 규모의 웹 사이트 개발에 많이 이용된다.

## 2.5 MySQL

MySQL은 세계적으로 가장 널리 사용되고 있는 오픈소스 데이터베이스로 C/C++, java 등과 연결 가능한 API도 제공되며, 다양한 웹서버와도 연결이 용이하다. 이러한 오픈소스 데이터베이스인 MySQL의 특징 몇 가지를 소개하겠다.

① MySQL은 관계형 데이터베이스 관리 시스템이다.

관계형 데이터베이스는 데이터를 하나의 커다란 저장 공간에 저장하지 않고 서로 별개의 테이블에 나누어서 저장을 하는 시스템으로 이를 통해 처리 속도와 유연성이 확보된다. SQL(Structured Query Language)은 ANSI/ISO표준에서 정의한 데이터베이스 접속을 위한 가장 일반적인 표준 언어이다.

② MySQL 소프트웨어는 오픈소스 이다.

MySQL 데이터베이스는 GPL(GNU Public License)를 준수하는 오픈소스 데이터베이스이며, GPL을 준수해서 사용하는 모든 사용자에게 무료로 배포되고 있다.

③ MySQL 서버는 클라이언트/서버 또는 임베디드 시스템에서 사용할 수 있다.

MySQL 데이터베이스 소프트웨어는 다중-쓰레드 SQL 서버로 구성된 클라이언트/서버 시스템이다. 또한, MySQL 서버를 임베디드 형태로도 다양한 용도로 사용할 수가 있다.

### 3. 연구내용

NFC를 활용한 출입통제 시스템은 NFC 기능을 장착한 스마트폰의 R/W 기능과 NFC Tag를 이용하였고 구현하기 위하여 안드로이드, Apache, PHP, MySQL, 자바를 이용했다.

스마트폰 어플리케이션은 관리자용과 사용자용 두가지를 구현하였다. 관리자 어플리케이션은 NFC Tag에 사무실 정보를 입력하는데 사용되고 사용자 어플리케이션은 NFC Tag에서 받은 출입문 정보와 함께 사용자 정보, 현재 시간을 출입통제 서버로 전송한다.

출입통제 서버는 사용자 어플리케이션으로부터 받은 정보를 MySQL과의 질의를 통해 인증 여부를 확인하여 출입승인여부를 결정한다. NFC를 활용한 출입통제 시스템은 출입문 개폐장치와 직접 연동하지는 않았고 출입승인정보를 서버에 XML 파일로 저장하고 사용자 스마트폰에선 XML파일을 읽어와 출입승인여부를 표시하는 형태로 구현했다.

자바 프로그램은 관리자용 프로그램으로써 관리자가 사용자 정보관리 및 출입조회를 쉽게 할 수 있도록 구현하였다.

그림1은 출입통제 시스템의 전체 구성도를 나타낸 것이다.



그림1. 출입통제 시스템의 구성도

#### 3.1 프로그램 구현

그림 2는 관리자 어플리케이션으로써 사무실 정보를 NFC Tag에 저장하는 용도이다. makeReadOnly() 함수를 이용하여 한번 NFC Tag에 저장된 정보는 수정 및 삭제가 불가능하다.



관리자 어플리케이션의 동작은 다음과 같은 형태로 수행된다.

- ① 스마트폰의 NFC 기능을 활성화 한 후 '사무실 번호 입력'란에 알맞은 정보를 입력한다.
- ② save 버튼을 누르면 "Tag에 가까이 하세요" 라는 창이 뜬다.
- ③ 스마트폰을 NFC Tag에 태깅한다.

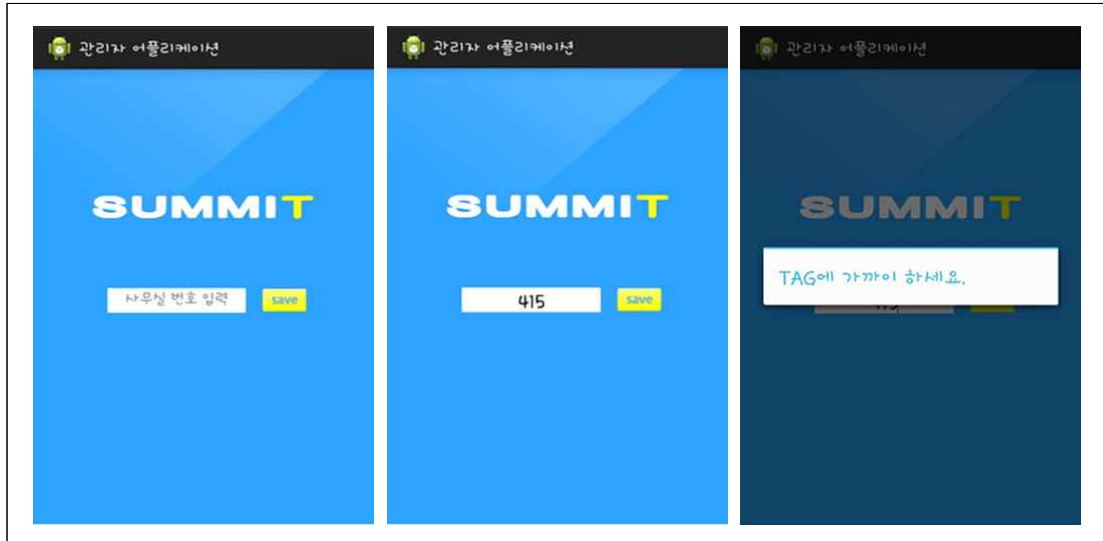


그림2. 관리자 어플리케이션

그림 3은 사용자 어플리케이션으로써 스마트폰을 NFC Tag에 태깅 시 출입승인여부 정보를 확인 할 수 있다. 사용자 인증결과에 따라 출입승인여부 정보가 표시된다.

사용자 어플리케이션의 동작은 다음과 같은 형태로 수행된다.

- ① NFC Tag에 스마트폰을 태깅 시 NFC Tag에 있는 출입문 정보를 스마트폰이 읽어온다.
- ② 읽어온 출입문 정보와 사용자 정보(스마트폰 번호, 현재 시간)를 출입통제 서버에 url.openstream을 이용하여 전송한다.
- ③ 출입통제 서버에서는 DB와의 질의 후 출입승인여부를 XML파일 형태로 저장하는데 사용자 어플리케이션은 이것을 읽어와 승인여부를 화면에 표시한다.

현재까지는 개폐기능을 연동할 수 있는 장치가 없어서 개폐기능을 구현하지 않았으며 대신 사용자 스마트폰 화면에 승인 여부를 표시하도록 했다. 출입문의 개폐장치와 연동하면 실제 출입관리 시스템으로 이용 가능하다.

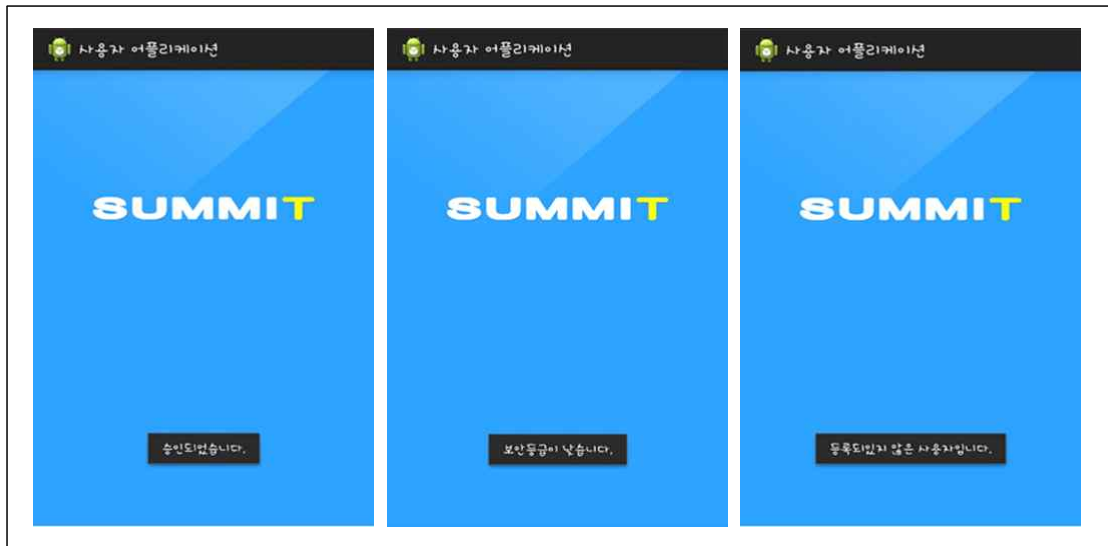


그림3. 사용자 어플리케이션

관리자 프로그램은 관리자는 새로운 사용자를 등록할 수 있고, 사용자 정보를 조회, 수정, 삭제할 수 있다. 출입조회 버튼을 누르면 사용자가 태깅 시 남겨지는 출입정보를 확인할 수 있으며 검색기능을 이용하여 편리한 관리가 가능하다.

그림4는 관리자 프로그램의 로그인 창으로써 서버 IP, DB ID, PW를 이용하여 로그인한다.

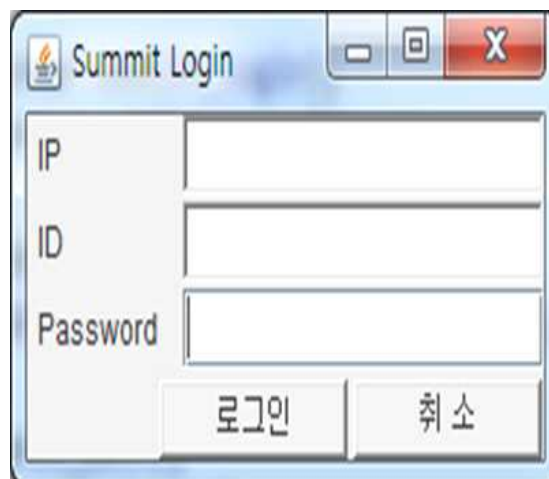


그림4. 관리자 프로그램 로그인

그림5는 사용자 정보관리 창으로써 사용자를 등록할 수 있고, 수정, 삭제, 조회할 수 있다.

사원번호	이름	번호	보안등급
1111111	김덕경	01012345678	1
2222222	김정수	01023456789	2
3333333	신민기	01034567890	3

그림 5. 관리자 프로그램 사용자 정보관리

그림6은 사용자 출입조회 창으로써 사원번호, 이름, 사무실, 출입일시를 이용하여 조회할 수 있다.

사원번호	이름	사무실	출입일시
3333333	신민기	415	2014/05/16/02:33
2222222	김정수	415	2014/05/16/02:32
1111111	김덕경	415	2014/05/16/02:31

그림 6. 관리자 프로그램 사용자 출입조회

## 3.2 안드로이드 소스

### 관리자 어플리케이션

```
import android.app.Activity;
import android.app.AlertDialog;
import android.app.PendingIntent;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.IntentFilter.MalformedMimeTypeException;
import android.nfc.NdefMessage;
```

```

import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.nfc.Tag;
import android.nfc.tech.Ndef;
import android.nfc.tech.NdefFormatable;
import android.os.Bundle;
import android.os.Parcelable;
import android.text.Editable;
import android.text.Html;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class SummitWriterActivity extends Activity {

    private static final String TAG = "summit";
    private boolean mResumed = false;
    private boolean mWriteMode = false;
    NfcAdapter mNfcAdapter;
    EditText mNote;
    Intent mIntent;
    PendingIntent mNfcPendingIntent;
    IntentFilter[] mWriteTagFilters;
    IntentFilter[] mNdefExchangeFilters;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        mNfcAdapter = NfcAdapter.getDefaultAdapter(getApplicationContext());
        findViewById(R.id.write_tag).setOnClickListener(mTagWriter);
        mNote = (EditText) findViewById(R.id.note);
        mNote.setHint(Html.fromHtml("<small>" + "사무실 번호 입력" +
        "</small>"));
        mNote.addTextChangedListener(mTextWatcher);
        mNfcPendingIntent = PendingIntent.getActivity(this, 0, new Intent(this,
        getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0);
        IntentFilter ndefDetected = new
        IntentFilter(NfcAdapter.ACTION_NDEF_DISCOVERED);

        try {
            ndefDetected.addDataType("text/summit");

```

```

    } catch (MalformedMimeTypeException e) {
        mNdefExchangeFilters = new IntentFilter[] { ndefDetected };
    }

    IntentFilter tagDetected = new IntentFilter(
        NfcAdapter.ACTION_TAG_DISCOVERED);
    mWriteTagFilters = new IntentFilter[] { tagDetected };
}

@Override
protected void onResume() {
    super.onResume();
    mResumed = true;
}

private NdefMessage[] getNdefMessages(Intent intent) {
    NdefMessage[] msgs = null;
    String action = intent.getAction();

    if (NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)
        ||
        NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
        Parcelable[] rawMsgs =
            intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);

        if (rawMsgs != null) {
            msgs = new NdefMessage[rawMsgs.length];
            for (int i = 0; i < rawMsgs.length; i++) {
                msgs[i] = (NdefMessage) rawMsgs[i];
            }
        } else {
            byte[] empty = new byte[] {};
            NdefRecord record = new
                NdefRecord(NdefRecord.TNF_UNCHANGED,
                    empty, empty, empty);
            NdefMessage msg = new NdefMessage(new NdefRecord[] { record
            });

            msgs = new NdefMessage[] { msg };
        }
    } else {
        Log.d(TAG, "알려지지 않은 인텐트.");
        finish();
    }
    return msgs;
}

```

```

@Override
protected void onPause() {
    super.onPause();
    mResumed = false;
    mNfcAdapter.disableForegroundNdefPush(this);
}

@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);

    if (mWriteMode == NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent.getAction())) {
        Tag detectedTag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        writeTag(getNoteAsNdef(), detectedTag);
    }
}

boolean writeTag(NdefMessage message, Tag detectedTag) {
    int size = message.toByteArray().length;
    try {
        Ndef ndef = Ndef.get(detectedTag);
        if (ndef != null) {
            ndef.connect();
            if (!ndef.isWritable()) {
                toast("읽기 전용 태그.");
                return false;
            }
            if (ndef.getMaxSize() < size) {
                toast("태그 용량은 " + ndef.getMaxSize() + " 바이트이고, 메시지 크기는 " + size + " 바이트이다.");
                return false;
            }
            ndef.writeNdefMessage(message);
            ndef.makeReadOnly();
            toast("태그에 기록했음.");
            return true;
        } else {
            NdefFormatable format = NdefFormatable.get(detectedTag);
            if (format != null) {
                try {
                    format.connect();

```

```

        format.format(message);
        toast("태그를 포맷을 했고 메시지를 썼음.");
        return true;
    } catch (IOException e) {
        toast("태그 포맷을 실패했음.");
        return false;
    }
} else {
    toast("태그가 NDEF를 지원하지 않음.");
    return false;
}
} catch (Exception e) {
    toast("태그에 쓰기 실패.");
}
return false;
}

private void toast(String text) {
    Toast.makeText(this, text, Toast.LENGTH_LONG).show();
}

private View.OnClickListener mTagWriter = new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        disableNdefExchangeMode();
        enableTagWriteMode();
        new AlertDialog.Builder(SummitWriterActivity.this).setTitle("TAG에
가까이 하세요.") .setOnCancelListener(
            new DialogInterface.OnCancelListener() {
                @Override
                public void onCancel(DialogInterface dialog) {
                    disableTagWriteMode();
                    enableNdefExchangeMode();
                }
            })
        .create().show();
    }
};

private TextWatcher mTextWatcher = new TextWatcher() {
    @Override
    public void afterTextChanged(Editable arg0) {
        if (mResumed) {

```

```

mNfcAdapter.enableForegroundNdefPush(SummitWriterActivity.this,
    getNoteAsNdef());
    }
}

.
.
.

```

## 사용자 어플리케이션

```

import java.io.InputStream;
import java.net.URL;
import java.net.URLEncoder;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserFactory;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.Parcelable;
import android.telephony.TelephonyManager;
import android.util.Log;
import android.widget.Toast;

public class SummitActivity extends Activity {

    private static final String TAG = "Summit";
    private static final String SERVER_ADDRESS = "서버 IP 입력";
    ArrayList<String> array_id;
    String tagname, content, office, phonenum, login, empty;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        array_id=new ArrayList<String>();
    }
    @Override

```



```

        protected void onResume() {
            super.onResume();

            i
(NfcAdapter.ACTION_NDEF_DISCOVERED.contentEquals(getIntent().getAction())) { f
            NdefMessage[] messages = getNdefMessages(getIntent());
            byte[] payload = messages[0].getRecords()[0].getPayload();
            setNoteBody(new String(payload));
            setIntent(new Intent());
        }
    }

    protected void onNewIntent(Intent intent)
    {
        i
(NfcAdapter.ACTION_NDEF_DISCOVERED.equals(intent.getAction())) f
        {
            NdefMessage[] msgs = getNdefMessages(intent);
            setNoteBody(new String(msgs[0].getRecords()[0].getPayload()));
        }
    }

    private NdefMessage[] getNdefMessages(Intent intent) {
        NdefMessage[] msgs = null;
        String action = intent.getAction();

        if (NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)
            || NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action))
        {
            Parcelable[] rawMsgs =
intent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);

            if (rawMsgs != null) {
                msgs = new NdefMessage[rawMsgs.length];
                for (int i = 0; i < rawMsgs.length; i++) {
                    msgs[i] = (NdefMessage) rawMsgs[i];
                }
            } else {
                byte[] empty = new byte[] {};
                NdefRecord record = new NdefRecord(NdefRecord.TNF_UNKNOWN,
                    empty, empty, empty);
                NdefMessage msg = new NdefMessage(new NdefRecord[] { record });
                msgs = new NdefMessage[] { msg };
            }
        } else {
            Log.d(TAG, "알려지지 않은 인텐트.");
            finish();
        }
    }
}

```

```

        return msgs;
    }

    private void setNoteBody(String string) {
        office = string;
        TelephonyManager mTelephonyMgr = (TelephonyManager)
        getSystemService(Context.TELEPHONY_SERVICE);
        phonenum = mTelephonyMgr.getLine1Number();
        new Thread(new Runnable(){
            @Override
            public void run() {
                try{
                    String year = new SimpleDateFormat("yyyy").format(new Date());
                    String month = new
SimpleDateFormat("MM").format(new Date());
                    String day = new
SimpleDateFormat("dd").format(new Date());
                    String hour = new
SimpleDateFormat("HH").format(new Date());
                    String min = new
SimpleDateFormat("mm").format(new Date());
                    String result = "";
                    result = result + year;
                    result = result + "/" + month;
                    result = result + "/" + day;
                    result = result + "/" + hour;
                    result = result + ":" + min;
                    String ret="";

                    String rss = SERVER_ADDRESS + "/xmlTest/";
                    String tagname="";

                    XmlPullParserFactory factory =
XmlPullParserFactory.newInstance();
                    factory.setNamespaceAware(true);
                    XmlPullParser xpp = factory.newPullParser();

                    URL url = new
URL(SERVER_ADDRESS+"/xmlTest/insert2.php?"+"office="+URLEncoder.encode(off
ice,"euc-kr")+"&phonenum="+URLEncoder.encode(phonenum,"euc-kr")+"&nowtime=
"+URLEncoder.encode(result,"euc-kr"));
                    url.openStream();
                    URL server = new URL(rss + "insert.xml");
                    InputStream is = server.openStream();
                    xpp.setInput(is, "euc-kr");

                    int eventType = xpp.getEventType();

```

```

        while(eventType != XmlPullParser.END_DOCUMENT) {
            if(eventType == XmlPullParser.START_TAG) {

                tagname=xpp.getName();

            }
            else if(eventType == XmlPullParser.TEXT){

                if(tagname.equals("phonenum")){
                    ret=xpp.getText();
                }
                if(tagname.equals("office")){
                    ret=xpp.getText();
                }
                if(tagname.equals("empty")){
                    ret=xpp.getText();
                }

                .
                .
                .
            }
        }
    }
}

```

### 3.2 PHP 소스

```

<head>
<meta http-equiv = "content-Type" content = "text/html" charset = "euc-kr">
</head>
<?
include"dbconn.php";

$phonenum=$_REQUEST['phonenum'];
$office=$_REQUEST['office'];
$now=$_REQUEST['nowtime'];

$dir = "파일이 있을 디렉토리";
$sql="select * from member where phone=('$phonenum');";
$result=mysql_query($sql);
$sql1="select * from officeinfo where office=('$office');";
$result1=mysql_query($sql1);

$xmlcode = "<?xml version = \"1.0\" encoding = \"euc-kr\"?>\n";
$xmlcode .="<user>\n";
while($obj = mysql_fetch_object($result))
{
    $check_num=$obj->phone:
}
}

```

```

$check_level=$obj->level;
$check_name=$obj->name;
$check_idx=$obj->idx;
}

if($check_num != NULL) {

    while($obj = mysql_fetch_object($result1))
    {
        $office_level=$obj->level;
        .
        .
        .
    }
}

```

### 3.3 자바 소스

#### 관리자프로그램 로그인

```

import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.CardLayout;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Panel;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.HashMap;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;

public class Login extends JFrame implements ActionListener {
    Panel button_panel;
    private Label l_ip = new Label(" IP");
    private Label l_id = new Label(" ID");
    private Label l_passwd = new Label(" Password");
}

```

```

private Label lblBlank = new Label("
");
private TextField txt_ip = new TextField(15);
private TextField txt_id = new TextField(15);
private JPasswordField txt_passwd = new JPasswordField(15);
private Button bt_login = new Button("로그인");
private Button bt_cancel = new Button("취 소");

String driver, url, id, passwd;
Connection con;
public Login(String title){
    super(title);

    BorderLayout BorderLayout = new BorderLayout(0, 3);
    setLayout(BorderLayout);

    Panel panel_West = new Panel(new GridLayout(3, 1, 0, 2));
    Panel panel_Center = new Panel(new GridLayout(3, 1, 0, 2));
    Panel panel_South = new Panel(new BorderLayout());

    panel_West.add(l_ip);
    panel_West.add(l_id);
    panel_West.add(l_passwd);

    panel_Center.add(txt_ip);
    panel_Center.add(txt_id);
    panel_Center.add(txt_passwd);
    JPanel panel_South_West = new JPanel(new CardLayout());
    JPanel panel_South_Center = new JPanel(new GridLayout(1, 2, 3, 0));

    panel_South_West.add("west", lblBlank);

    panel_South_Center.add(bt_login);
    panel_South_Center.add(bt_cancel);

    panel_South.add("West", panel_South_West);
    panel_South.add("Center", panel_South_Center);

    add("West", panel_West);
    add("Center", panel_Center);
    add("South", panel_South);

    bt_login.addActionListener(this);
    bt_cancel.addActionListener(this);

```

```

        this.setDefaultCloseOperation(EXIT_ON_CLOSE);

        this.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        }); // end windowListner

        start();

        pack();
        this.setLocation( ((this.getToolkit().getScreenSize()).width -
this.getWidth())/2, ((this.getToolkit().getScreenSize()).height -
this.getHeight())/2);
        setVisible(true);
    }

    public HashMap<String,String> getDBMStype(String ip, String port, String
dbname,String id,String pass){
        HashMap<String,String> map=new HashMap<String,String>();

        map.put("driver", "com.mysql.jdbc.Driver");
        map.put("ip", ip=(ip==null)? "192.168.0.99":ip);
        map.put("port", port=(port==null)? "3306":port);
        map.put("dbname", dbname=(dbname==null)?
"user":dbname);

        map.put("id", id=(id==null)? "root":id);
        map.put("pass", pass=(pass==null)? "summit":pass);

url="jdbc:mysql:"+"//"+ip+": "+port+"/"+dbname+"?useUnicode=true&characterEnc
oding=euc_kr";

        driver=map.get("driver");
        txt_ip.setText(ip);
        txt_id.setText(id);
        txt_passwd.setText(pass);
        return map;
    }

    public void start() {

        getDBMStype(null, null, null,null,null);
    }

    public void check(){

        if(txt_ip.getText()==null

```

```

txt_ip.getText().length() == 0 || txt_ip.getText().length() > 40) {
    JOptionPane.showMessageDialog(getParent(), "아이피를 정확
히 입력해주세요(ex)192.168.0.1 또는 localhost");
    txt_ip.setText("");
    txt_ip.requestFocus();
    return;
}
.
.
.

```

### 관리자프로그램 사용자 관리

```

import java.awt.BorderLayout;
import java.awt.Choice;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.UnsupportedEncodingException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.table.DefaultTableCellRenderer;

```

```

import javax.swing.table.TableColumnModel;

public class Member extends JFrame implements ActionListener {
    JLabel m_label, m_label2, l_idx, l_name, l_phone, l_level;
    JTextField txt_idx, txt_name, txt_phone, txt_level, txt_keyword;

    Choice ch_category;
    JTable table;
    JScrollPane scroll;
    JButton bt_regist, bt_edit, bt_delete, bt_select, bt_getListAll, bt_exit;
    JPanel south, north, center, p1, p2, p3, p4, p5, p6, p7;

    Login log;
    String driver, url, userid, passwd;

    Connection con;
    PreparedStatement pstmt;
    ResultSet rs;

    String sql;
    String category;
    MemberModel model = new MemberModel();
    int selRow;
    int idx;

    public Member(Login log) {
        super("사용자 관리");
        this.log=log;
        this.con=log.con;

        l_idx = new JLabel("사 원 번 호");
        l_name = new JLabel("이 름");
        l_phone = new JLabel("연 락 처");
        l_level = new JLabel("보 안 등 급");

        txt_idx = new JTextField(12);
        txt_name = new JTextField(12);
        txt_phone = new JTextField(12);
        txt_level = new JTextField(12);
        txt_keyword = new JTextField(22);

        ch_category=new Choice();
        ch_category.add("조 회 선 택");
        ch_category.add("사 원 번 호");
    }
}

```



```

ch_category.add("이름");
ch_category.add("번호");
ch_category.add("보안등급");
bt_regist = new JButton("등록");
bt_edit = new JButton("수정");
bt_delete = new JButton("삭제");
bt_select = new JButton("조회");
bt_getListAll = new JButton("전체조회");
bt_exit = new JButton("출입 조회");
table = new JTable(model);
scroll = new JScrollPane(table);
south = new JPanel();
north = new JPanel();
center = new JPanel();
p2 = new JPanel();
p3 = new JPanel();
p4 = new JPanel();
p5 = new JPanel();
p6 = new JPanel();
north.setPreferredSize(new Dimension(500, 200));
center.setPreferredSize(new Dimension(500, 400));
south.setPreferredSize(new Dimension(500, 50));
add(north, BorderLayout.NORTH);
add(center, BorderLayout.CENTER);
add(south, BorderLayout.SOUTH);
north.setLayout(new GridLayout(5, 1));
north.add(p2);
north.add(p3);
north.add(p4);
north.add(p5);
north.add(p6);
p2.add(l_idx);
p2.add(txt_idx);
p3.add(l_name);
p3.add(txt_name);
p4.add(l_phone);
p4.add(txt_phone);
p5.add(l_level);
p5.add(txt_level);
p6.setLayout(new FlowLayout(FlowLayout.LEFT));
p6.setBackground(Color.lightGray);
p6.add(ch_category);
p6.add(txt_keyword);
p6.add(bt_select);
p6.add(bt_getListAll);

```

```

ch_category.addItemListener(new ItemListener() {
public void itemStateChanged(ItemEvent ie) {
    Object obj = ie.getSource();
    category=(String)ie.getItem();
    if(category=="사 원 번 호"){
        category="idx";
    }else if(category=="이 름"){
        category="name";
    }else if(category=="번 호"){
        category="phone";
    }else if(category=="보 안 등 급"){
        category="level";
    }
}

});

center.setLayout(new GridLayout(1, 1));
center.add(scroll);
table.setAutoCreateRowSorter(true);
DefaultTableCellRenderer tScheduleCellRenderer = new
DefaultTableCellRenderer();

tScheduleCellRenderer.setHorizontalAlignment(SwingConstants.CENTER);
TableModel tcmSchedule = table.getColumnModel();
for (int i = 0; i < tcmSchedule.getColumnCount(); i++) {
    tcmSchedule.getColumnModel(i).setCellRenderer(tScheduleCellRenderer);
}

south.setLayout(new GridLayout(1, 4));
south.add(bt_regist);
south.add(bt_edit);
south.add(bt_delete);
south.add(bt_exit);

bt_regist.addActionListener(this);
bt_edit.addActionListener(this);
bt_delete.addActionListener(this);
bt_select.addActionListener(this);
bt_getListAll.addActionListener(this);
bt_exit.addActionListener(this);
table.addMouseListener(new MouseAdapter() {
    public void mouseClicked(MouseEvent e) {
        selRow = table.getSelectedRow();
        System.out.println(selRow);
        String idx = table.getValueAt(selRow, 0).toString();
        String name = (String) table.getValueAt(selRow, 1);
    }
});

```

```

        String phone = table.getValueAt(selRow, 2).toString();
        String level = table.getValueAt(selRow, 3).toString();

        txt_idx.setText(idx);
        txt_name.setText(name);
        txt_phone.setText(phone);
        txt_level.setText(level);

    }

});

this.setDefaultCloseOperation(EXIT_ON_CLOSE);
pack();

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

int xPos = screenSize.width/2 - getSize().width/2 ;
int yPos = screenSize.height/2 - getSize().height/2 ;
setLocation(xPos,yPos);

setVisible(true);
getListAll();
}

public boolean getListAll() {
    boolean result = false;
    String sql = "select * from member order by idx desc";
    try {
        .
        .
        .
    }
}

```

### 관리자프로그램 사용자 출입조회

```

import java.awt.BorderLayout;
import java.awt.Choice;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;

```

```

import java.awt.event.ItemListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.TableColumnModel;

public class Ask extends JFrame implements ActionListener {
    JLabel l_idx, l_name, l_phone, l_level;
    JTextField txt_keyword;
    Choice ch_category;
    JTable table;
    JScrollPane scroll;
    JButton bt_regist, bt_select, bt_getListAll, bt_exit;
    JPanel south, north, center, p1, p6, p7;
    Login log;
    String driver, url, userid, passwd;
    Connection con;
    PreparedStatement pstmt;
    ResultSet rs;
    String sql;
    String category;
    AskModel model = new AskModel();
    int selRow;
    int idx;
    public Ask(Login log) {
        super("사용자 출입조회");
        this.log=log;
        this.con=log.con;

        txt_keyword = new JTextField(22);

```

```

ch_category=new Choice();
ch_category.add("조 회 선 택");
ch_category.add("사 원 번 호");
ch_category.add("이      름");
ch_category.add("사  무  실");
ch_category.add("출 입 일 시");
bt_regist = new JButton("이전");
bt_select = new JButton("조회");
bt_getListAll = new JButton("전체조회");
bt_exit = new JButton("종료");
table = new JTable(model);
scroll = new JScrollPane(table);
south = new JPanel();
north = new JPanel();
center = new JPanel();
p1 = new JPanel();
p6= new JPanel();
north.setPreferredSize(new Dimension(500, 40));
center.setPreferredSize(new Dimension(500, 500));
south.setPreferredSize(new Dimension(500, 50));
add(north, BorderLayout.NORTH);
add(center, BorderLayout.CENTER);
add(south, BorderLayout.SOUTH);
north.setLayout(new GridLayout(1, 1));
north.add(p6);
p6.setLayout(new FlowLayout(FlowLayout.LEFT));
p6.setBackground(Color.lightGray);
p6.add(ch_category);
p6.add(txt_keyword);
p6.add(bt_select);
p6.add(bt_getListAll);
ch_category.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ie) {
        Object obj = ie.getSource();
        category=(String)ie.getItem();
        txt_keyword.setText("");
        if(category=="사 원 번 호"){
            category="idx";
        }else if(category=="이      름"){
            category="name";
        }else if(category=="사  무  실"){
            category="room";
        }else if(category=="출 입 일 시"){
            category="time";
        }
        txt_keyword.setText("2014/04/17-2014/04/20");
    }
});

```

```

    }

    }

    });

    center.setLayout(new GridLayout(1, 1));
    center.add(scroll);
    table.setAutoCreateRowSorter(true);

    DefaultTableCellRenderer tScheduleCellRenderer = new
DefaultTableCellRenderer();

tScheduleCellRenderer.setHorizontalAlignment(SwingConstants.CENTER);
    TableColumnModel tcmSchedule = table.getColumnModel();
    for (int i = 0; i < tcmSchedule.getColumnCount(); i++) {
        tcmSchedule.getColumn(i).setCellRenderer(tScheduleCellRenderer);
    }
    south.setLayout(new GridLayout(1, 2));
    south.add(bt_regist);
    south.add(bt_exit);
    bt_regist.addActionListener(this);
    bt_select.addActionListener(this);
    bt_getListAll.addActionListener(this);
    bt_exit.addActionListener(this);
    this.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            closeDB();
            System.exit(0);
        }
    });

    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    pack();

    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

    int xPos = screenSize.width/2 - getSize().width/2 ;
    int yPos = screenSize.height/2 - getSize().height/2 ;
    setLocation(xPos,yPos);

    setVisible(true);
    getListAll();
}

public boolean getListAll() {
    boolean result = false;

```

```

String sql = "select * from search order by idx desc";
try {
    pstmt = con.prepareStatement(sql);
    rs = pstmt.executeQuery();
    Vector list = new Vector();
    while (rs.next()) {
        Vector record = new Vector();
        record.add(rs.getInt("idx"));
        record.add(rs.getString("name"));
        record.add(rs.getInt("room"));
        record.add(rs.getString("time"));
        list.add(record);
    }
    model.setList(list);
    this.repaint();
    result = true;
} catch (SQLException e) {
    e.printStackTrace();
}
return result;
}

public boolean select(String category, String keyword){
    boolean result = false;
    if (category=="time"){
        String[] timesearch=keyword.split("-");
        sql ="select  *   from  search   where  "+category+"   between
""+timesearch[0]+" and ""+timesearch[1]+" order by time desc";
    }
    else{
        sql ="select  *   from  search   where  "+category+"   like
'%" +keyword+"%' order by time desc";
    }
    try {
        pstmt=con.prepareStatement(sql);
        rs = pstmt.executeQuery();
        Vector list = new Vector();
        while (rs.next()) {
            Vector record = new Vector();
            record.add(rs.getInt("idx"));
            record.add(rs.getString("name"));
            record.add(rs.getString("room"));
            .
            .
            .

```

## 4. 결론

NFC의 활용 분야가 광고, 티켓팅, 의료서비스, 안내/제품정보, 결제 및 계좌이체, 개인 인증 등의 보다 넓게 확장되고 있어 많은 스마트 기기들이 NFC를 채택하고 있으며 지속 확장되면서 NFC 스마트기기 하나를 소유하면 인증, 차량 이용 등 많은 부분의 서비스를 이용할 수 있을 것으로 기대되고 있다. 또한, 기존 RFID를 이용하던 다양한 응용분야에 대하여 NFC로 대체가 진행되고 있다. 기존 RFID 출입통제 시스템의 사용 경우에는 리더기에 사용자 정보를 저장함으로써 악의적인 해킹에 노출되어 있으며, 사원증 소지의 불편함과 분실위험성, 리더기 설치의 비용의 문제가 있다. NFC를 활용한 출입통제 시스템을 이용하면 이러한 문제점을 보완해줄 수 있다.

NFC를 활용한 출입통제 시스템은 Apache, PHP, MySQL을 이용해 서버와 데이터베이스를 구축하였으며 관리자가 관리자 어플리케이션을 통해 쉽게 NFC Tag를 관리하고 자바로 만든 관리자 프로그램을 이용하여 사용자의 정보관리, 출입조회를 손쉽게 할 수 있고 사용자는 사용자 어플리케이션을 이용하여 간편한 인증을 할 수 있다. NFC를 활용한 출입통제 시스템을 구현하기 위해 학교내 와이파이를 이용하여 서버와 안드로이드 간의 통신을 테스트해 정상적으로 구현하였다.

### 4.1 향후 과제

NFC를 활용한 출입관리 시스템은 스마트 폰의 NFC의 기능과 NFC Tag를 활용하여 설계하고 구현하였다. 스마트폰은 사용자가 항상 휴대하고 다니는 물품이므로 분실에 대한 우려가 상대적으로 낮아서 인증용 도구로 유용하다. 현재까지는 스마트폰에 내장된 사용자의 휴대폰번호만으로 인증이 가능하도록 구현하였지만 향후 스마트폰에 저장된 공인인증서와 이를 이용한 전자서명을 이용하는 인증 방식으로 구현할 계획이다.

## 5. 참고문헌

- [1] 한민기, “NFC환경에서 개인정보 보호를 위한 안전한 모바일 결제 시스템 설계” 숭실대학교 일반대학원: 학위논문(석사) 2012.2
- [2] 백종현, 염홍열, “NFC 기반 모바일 서비스 보안 위협 및 대책” 정보보호학회지 제 23권 제2호, 2013.4, 55-65
- [3] 임선희, 전재우, 정임진, 이옥연 “NFC보안 기술 분석 및 UICC적용 효과 연구”, 한국통신학회논문지, 2011.1
- [4] ISO/IEC 14443, Identification cards-Contactless Integrated Circuit cards-Proximitycards, ISO, Geneva, Switzerland, 2008
- [5] NFC Forum, “WhatisNFC?”, <http://www.nfc-forum.org/aboutnfc>
- [6] [http://ko.wikipedia.org/wiki/%EC%9E%90%EB%B0%94\\_\(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\\_%EC%96%B8%EC%96%B4\)](http://ko.wikipedia.org/wiki/%EC%9E%90%EB%B0%94_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D_%EC%96%B8%EC%96%B4))
- [7] NFC 세미나 발표자료 - NFC 기술개요 및 동향
- [8] 손성용, “NFC 기반 능동형 인증 기법을 활용한 출입통제 시스템의 설계 및 구현” 가천대학교 일반대학원: 모바일소프트웨어학과 모바일소프트웨어 학위논문(석사) 2012.8

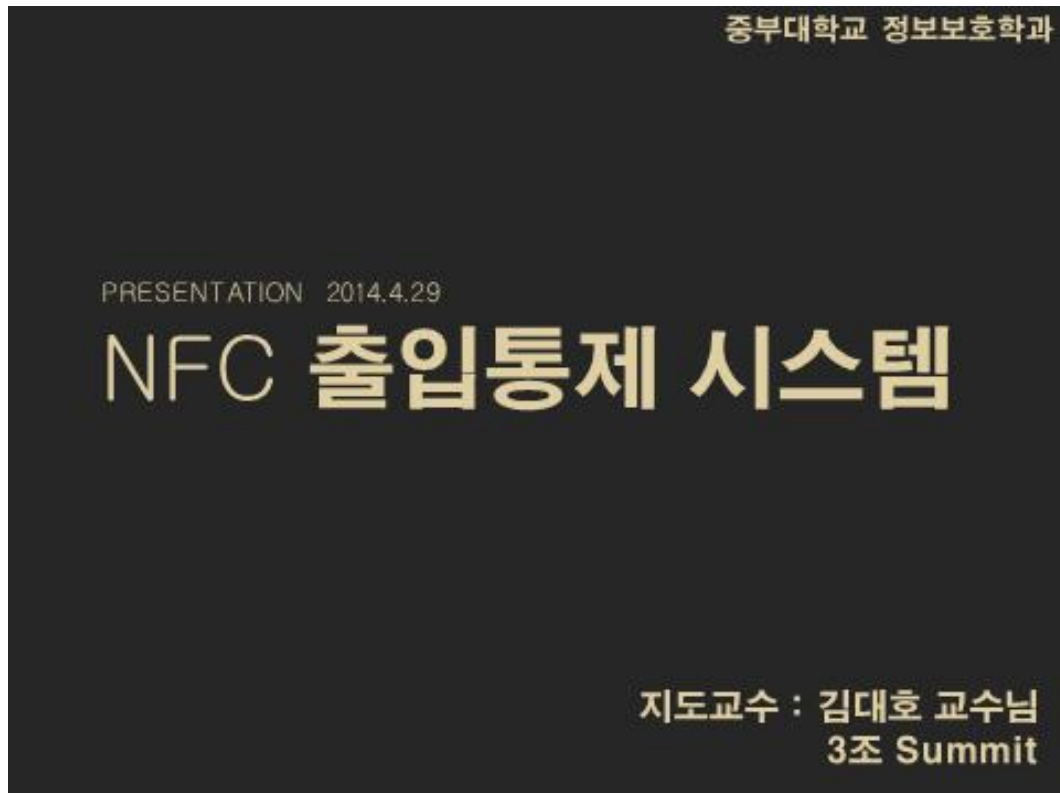


[9] 황재호, “PHP 프로그래밍 입문 기초부터 다지는 웹 사이트 제작 실습” - 한빛미디어 2005.12

[10] 삼성소프트웨어멤버십 - <http://blog.secmem.org/60>

[11] 상상e비즈 - <http://www.mysqlkorea.co.kr/>

## 6. 발표 PPT 자료



# Index

- 1 조 원 소 개
- 2 NFC출입 통제 시스템
- 3 시스템 설계
- 4 시스템 개발 및 구현
- 5 결론

## 1 PART 01

### 조 원 소 개

## 조원 소개

- 김덕경 : 개발 총괄
- 김정수 : 자료 조사
- 신민기 : 기술 연구

## PART 02

### NFC 출입 통제 시스템

## NFC 출입 통제 시스템

- **NFC 출입 통제 시스템**은 스마트폰의 NFC 기능을 이용하여  
인가된 사람만 출입할 수 있고 출입자, 출입 시간 등을 서버  
에 저장하여 관리자가 출입 관리를 할 수 있는 시스템
- 스마트폰을 이용한 손쉬운 출입 인증
- 관리자 프로그램을 이용하여 **사용자 정보** 및 **출입 조회**를  
효율적으로 관리

## NFC 출입 통제 시스템



- **10cm 이내의 거리에서 무선기기 간의 통신을 가능케 해주는 기술**

(13.56MHz 주파수 대역을 이용한 비접촉식 근거리 무선 통신의 한 종류)

- 안드로이드는 2.3부터 NFC 지원

### [ NFC 동작 방식 ]

구분	카드 에뮬레이션	리드/라이터	P2P
상세 내용	단말기의 ON/OFF와 관계 없이 항상 결제기를 통해 인식이 가능한 모드	NFC 활성화 상태에서 RFID Tag 정보를 인식, 휴대폰이 카드 리더기로 작동하는 모드	두 대의 NFC 휴대폰이 카드 리더기로서 작동하여 데이터를 상호간에 전송할 수 있는 모드

## NFC 출입 통제 시스템

### ■ NFC 활용 분야



## NFC 출입 통제 시스템

### • 개발 도구

- Java JDK 1.7
- Eclipse 및 Eclipse - **A**ndroid **D**evelopment **T**oolkit

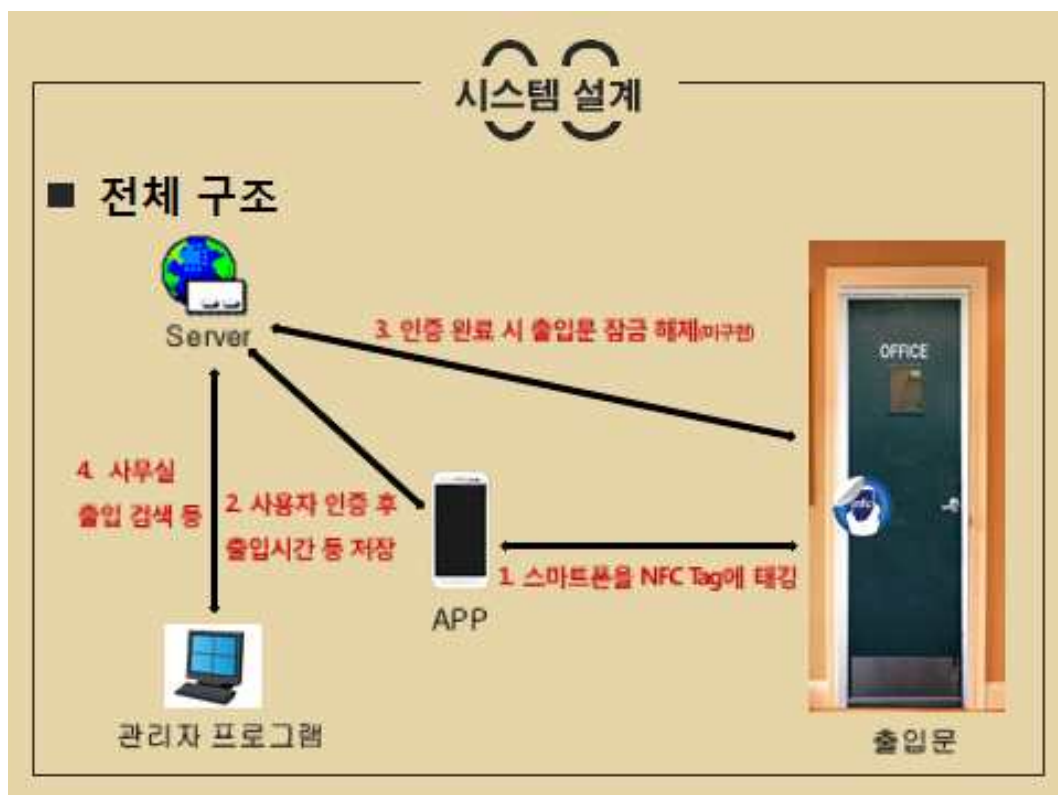
### • 개발 환경

- Windows 7
- Android 4.4.2(Kitkat)
- **A**pache + **P**HP + **M**ysql(APM setup)

# 3

PART 03

## 시스템 설계



## 시스템 설계

### ■ 시스템 구조 (사용자 어플리케이션 및 서버)



## 시스템 설계

### ■ 시스템 구조 (관리자 자바 프로그램)



# 4

PART 04

## 시스템개발 및 구현

### 시스템 개발 및 구현

#### ■ 관리자 어플리케이션



사무실 정보를 **NFC Tag**에 입력

`makeReadOnly()` 함수를 이용하여  
**NFC Tag**에 저장된 데이터는 **수정 불가**



## 시스템 개발 및 구현

### ■ 사용자 어플리케이션



NFC 기능을 활성화 한 스마트폰을  
NFC Tag에 태깅

사용자 등급에 따른 인증 결과 표시

## 시스템 개발 및 구현

### ■ 관리자 자바 프로그램



IP, ID, Password를 이용해 로그인

사용자 정보 관리

사용자 출입 검색

## 시스템 개발 및 구현

### ■ 데이터 베이스 (MySQL)

구분	컬럼명	컬럼 ID	DATA TYPE	길이	비고
1	사원번호	Idk	Varchar	20	1111111
2	이름	Name	Varchar	15	김덕경
3	번호	Phone	Varchar	20	0102345678
4	보안등급	level	Varchar	15	1

사용자 정보 관리용 DB

구분	컬럼명	컬럼 ID	DATA TYPE	길이	비고
1	사원번호	Idk	Varchar	20	1111111
2	이름	Name	Varchar	15	김덕경
3	사무실	Office	Varchar	20	415
4	출입시간	Time	Varchar	15	2014/04/17 /17:20

사용자 출입 관리용 DB

## 시스템 개발 및 구현

### ■ 데이터 베이스 (MySQL)

구분	컬럼명	컬럼 ID	DATA TYPE	길이	비고
1	사무실	Office	Varchar	20	415
2	보안등급	Level	Varchar	15	1

사무실 정보 관리용 DB

# 5

PART 05

## 결론 및 의견

### 35 결론

#### ◆ 개발 성과

- 스마트폰의 NFC 기능을 활용한 출입통제 시스템을 개발
- 사용자 데이터베이스 관리, 출입 인증 및 조회 등을 통한 안정적인 출입통제 및 보안관리체제 구축

#### ◆ 의견

- 개발과정에서 안드로이드 어플리케이션, 자바 GUI 프로그램, PHP 등을 직접 프로그램으로 구현
- 시스템 개발에 필수적인 프로그래밍 개발 역량을 배양하고, 보안관리의 중요성을 체험

Thank You  
Q&A

Summit