

RFID 기술을 이용한 무인 자전거 스테이션 원격 유지 관리*

정성훈** · 김상철***

The remote maintenance system using RFID technology for the unmanned bicycle station

Jung, Sung Hoon · Kim, Sang Chul

〈Abstract〉

In this paper, the management system for a bicycle station using 900 MHz RFID technology has been developed. Based on the several reasons such as environmental pollution, high oil prices, and the government's eco-friendly policies, a bicycle usage is increasing nowadays. Accordingly, a need for bicycle parking spaces has already been emerging and increasing around a bicycle station. But most of the bicycle parking system are operated by manually, and it causes somewhat inefficient.

Therefore, this paper suggests an unmanned bicycle station using RFID technology. The proposed system is supported by the mobile applications that are operated in the smart phones, and which gives the real-time access to the information of bicycle station. The proposed system yields owners of bicycle owners the convenience and efficiency of the station management in order to maximize the function of the bicycle stations.

Key Words : UHF RFID, Unmanned Bicycle Station, USN, Mobile Applications, Green IT

I. 서론

환경오염, 고유가, 정부의 친환경 정책등과 같은 여러 가지 이유로 자전거 사용량이 늘고 있다. 이에 따라 자전거 전용 주차공간에 대한 필요도 대두되어 이미 자전거 스테이션은 주위에서 어렵지 않게 찾아볼 수 있다. 하지만 사람이 상주하거나 수동으로 자전거를 주차해야 하는

비효율적인 면이 있는 것은 사실이다.

따라서 RFID 태그를 사용한 무인자전거 스테이션과 이를 시간과 장소에 관계없이 실시간 관리가 가능한 모바일 어플리케이션을 하나로 묶어서 자전거 사용자의 스테이션의 이용의 편리함과 더불어 스테이션 소유자의 관리의 효율을 극대화 하는 시스템을 기술한다.

본 논문의 구성은 2장에서 관련기술 및 사용 플랫폼, 3장에서 시스템 구조를 설명하고 4장에서 실행 결과를 보인다. 마지막으로 5장에서 결론과 향후 계획에 대해 기술한다.

* 본 논문은 2011년도 산학연공동기술개발사업(No. 0045590), 2010년도 국민대학교 교내연구지원금과 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2011-0027426, No. 2011-0002519)

** 국민대학교 컴퓨터공학과 학사 재학

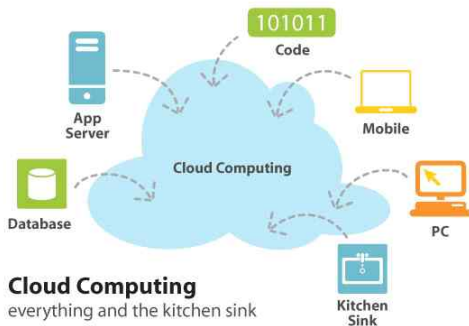
*** 국민대학교 컴퓨터공학과 부교수(교신저자)

II. 관련연구

2.1 RFID

RFID란 라디오 주파수(radio frequency)를 이용하여 물건, 사람 등과 같이 대상을 식별하는 기술로서, 안테나와 칩으로 구성된 RF 태그(tag)에 사용 목적에 알맞은 정보를 저장하고 적용 대상에 부착한 후 판독기에 해당하는 RFID 리더(reader)을 통하여 정보를 인식하는 방법으로 활용되는 기술이다. 기존 바코드 시스템에 비해 RFID 기술은 라디오 주파수의 특성으로 인해 인식 거리가 길고 동시에 다수의 태그 인식이 가능하다는 장점을 가지고 있다[1]. 이 RFID태그를 자전거에 부착하여 각각의 자전거 및 소유주를 식별하고 주차를 진행하며 해당 태그정보를 서버로 전송한다.

2.2 중개 서버(Platform : Google App Engine)

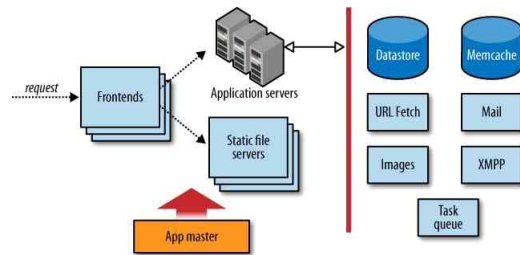


<그림 1> 클라우드 컴퓨팅

안드로이드 애플리케이션은 외부의 DB에 직접 접속을 공식적으로 지원하지 않는다. 따라서 중간에 웹 서버를 두고 데이터의 저장과 호출을 담당하도록 설계하는 것이 추세이다. 그런데 데이터베이스와 서버를 물리적으로 따로 구축한다면, 구축과 관리의 어려움이 있다. 그래서 <그림1>에서 보이듯이 서버와 데이터베이스를 하나

의 묶는 구글(Google)에서 제공하는 Google App Engine(이하 GAE)[2]이라는 클라우드 컴퓨팅 기반의 PaaS*를 사용하여 전체적인 구조를 간략하게 구성하고 개발과 관리를 쉽게 구성했다.

(*Platform as a Service : 미리 정의된 다양한 인터페이스를 이용하여 플랫폼을 서비스로 제공)



<그림 2> GAE의 아키텍처 구성

<그림2>에서 보듯이 개발자는 따로 물리적인 서버 및 데이터베이스 구축이 필요 없고 오직 앞단의 애플리케이션만 업로드하면 서비스를 구축할 수 있다. 또한 초기에 제공되는 무료 트래픽 용량으로 비용절감의 효과가 있다. 또한 국내뿐 아니라 해외에 확장하려는 서비스에도 적합하다. GAE가 제공하는 웹페이지로 어디서나 간편하게 서버관리가 가능하다.

단점으로는 관계형 데이터베이스를 지원하지 않기 때문에 자체적으로 지원하는 데이터베이스를 사용해야한다. 따라서 새로운 기술 습득에 대한 부담이 있다.

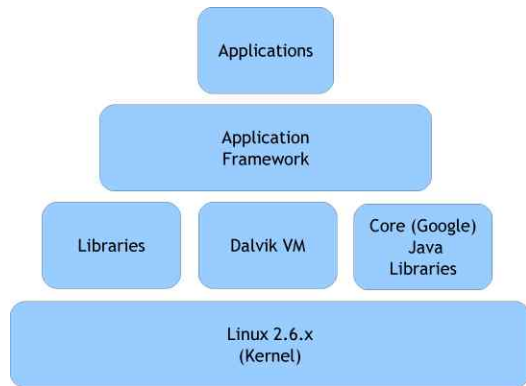
2.3 안드로이드 애플리케이션

안드로이드(Android)는 휴대전화를 비롯, 모바일 디바이스를 위한 운영체제와 미들웨어 그리고 핵심 어플리케이션을 포함하고 있는 소프트웨어 스택이다. 안드로이드는 개발자들이 자바(Java) 언어로 어플리케이션을 작성 할 수 있게 하였으며, 컴파일된 바이트코드를 구동할 수 있는 런타임 라이브러리를 제공한다. 또한 안드로이드 SDK를 통해 어플리케이션을 개발하기 위해 필요한

각종 도구들과 API를 제공한다.

안드로이드는 리눅스 커널 위에서 동작하며, 다양한 안드로이드 시스템 컴포넌트에서 사용되는 C/C++ 라이브러리들을 포함하고 있다. 안드로이드는 기존의 자바 가상 머신과는 다른 가상 머신인 달빅 가상 머신을 통해 자바로 작성된 어플리케이션을 별도의 프로세스에서 실행하는 구조로 되어있다.

2005년 안드로이드사를 구글에서 인수한 후 2007년 11월, 안드로이드 플랫폼을 휴대전화용 OS로서 무료 공개한다고 발표한 후 48개의 하드웨어, 소프트웨어, 통신 회사가 모여 만든 OHA(Open Handset Alliance)에서 공개 표준을 위해 개발하고 있다. 구글은 안드로이드의 모든 소스 코드를 오픈 소스 라이선스인 아파치 라이선스로 배포하고 있다. 아래 <그림 3>은 안드로이드 프레임워크의 계층을 보여준다[3].



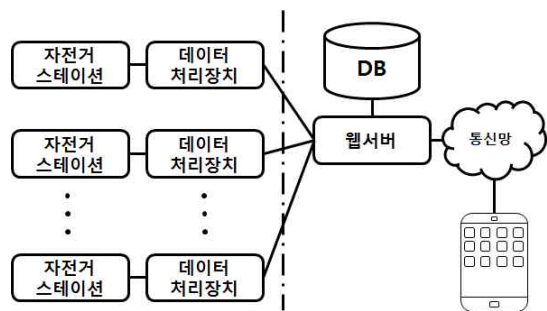
<그림 3> Android Framwork

III. 본론

3.1 시스템 설계

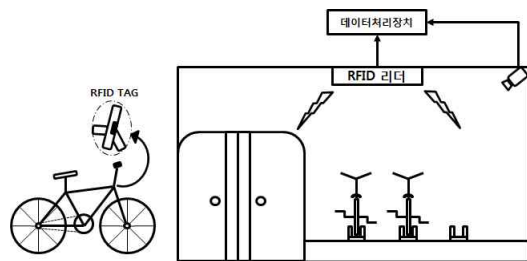
<그림 4>의 시스템은 전체적으로 3개의 부분으로 나뉘 볼 수 있다. 자전거 스테이션에서 주차 및 출차시 발

생하는 자전거 RFID Tag 정보를 전송하고 수신 받는 데이터처리 장치, 스테이션에서 넘어온 자전거정보를 데이터베이스에 저장 및 스테이션에 결과를 리턴하고, 모바일 어플리케이션의 요청에 따라 해당정보를 제공하는 일을 하는 웹 서버 및 DB, 마지막으로 스테이션의 정보를 웹 서버로부터 받아서 관리자에게 보여주는 모바일 어플리케이션이다.



<그림 4> 전체 시스템 구조

3.1.1 자전거 스테이션



<그림 5> 자전거 스테이션 구조

<그림 5>의 자전거 스테이션의 구조를 보게되면 자전거가 들어오고 나갈 때 자전거의 Tag가 RFID 리더기에 읽히고 해당 Tag의 정보가 웹 서버로 전송된다[4]. 특징으로는 낮은 주파수의 리더기는 한 번에 하나의 태그밖에 못 읽는 단점이 있다. 그래서 900Mhz 주파수 대역 RFID 리더기의 사용으로 다수의 자전거 태그를 동시에

읽어 Thread로 처리한다. 이로서 사용자가 몰리더라도 기다리는 수고를 덜게 할 수 있다[5].

3.1.2 중개 서버



<그림 6> 중개 서버 구조

<그림6>은 중개 서버의 구조를 보여준다. 중개 서버는 크게 2가지의 일을 수행한다.

첫 번째로 다수의 자전거 스테이션에서 주차되고 출차 되는 자전거의 RFID 태그번호가 서버로 전송되면 해당 태그번호를 기반으로 소유주를 검색하여 주차 시에는 주차정보를 출차 시에는 요금정산 및 결과를 데이터베이스에 저장하는 일을 담당한다.

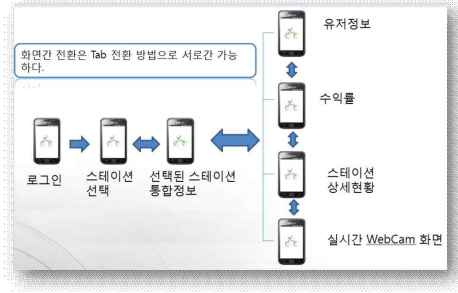
두 번째로 관리자의 애플리케이션에서 요청하는 특정 스테이션의 정보를 검색, 가공하여 전송하는 일이다. 전송 데이터 포맷은 XML보다 JSON*을 사용한다. 이유는 해당 프로젝트가 주고받는 데이터는 구조가 매우 단순하고 양이 적기 때문에 XML보다는 가벼워 모바일에 더욱 유리하기 때문이다. JSON은 단순한 텍스트 타입이고 다양한 데이터타입을 표현 할 수 있다. 또한 안드로이드에서도 JSON파서를 기본으로 제공한다.

(JSON : JavaScript Object Notation)[6]

3.1.3 안드로이드 애플리케이션

<그림7>은 애플리케이션 화면구성을 보여준다. 관리자가 로그인후 스테이션 선택화면에서 원하는 스테이션

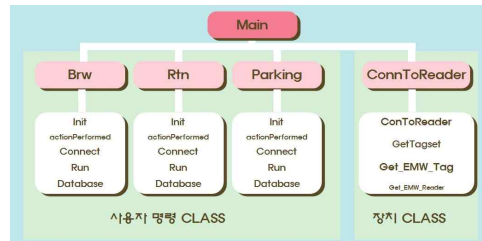
을 선택하면 해당 스테이션의 정보를 서버로부터 전송받아 보여준다. 관리자는 주제별로 구분된 화면을 이동하면서 스테이션의 정보를 확인할 수 있다.



<그림 7> 애플리케이션 화면 구성

3.2 시스템 구현

3.2.1 자전거 스테이션



<그림 8> 클래스 구조

<그림 8>과 같은 4가지 클래스로 이루어져 있다. Brw 와 Rtn은 자전거 대여 관련, Parking은 입출고를 관리, ConnToReader는 PC와 RFID 안테나간의 연결을 관리한다. 데이터베이스 구조는 <그림 11>과 같다. UserA와 Card의 경우는 자전거를 대여 하는 사용자의 이름, TagNumber, 정보 등을 다루는 테이블이며, Bicycle은 대여하는 자전거의 테이블, RenReturn은 자전거의 대여와 반납을 이용하는데 사용되는 테이블이다.

```

public void Database (String str, int TotalCount, int b)
{
    try
    {
        String Stn = null;
        String Tag = str;

        HttpClient httpClient = new DefaultHttpClient();

        List<NameValuePair> params = new ArrayList<NameValuePair>();

        params.add(new BasicNameValuePair("tag", Tag));
        params.add(new BasicNameValuePair("stn", Stn));

        UriEncodedFormEntity entity = new UriEncodedFormEntity(params, "EUC-KR");

        HttpPost httpPost = new HttpPost("http://localhost:8080/");

        httpPost.setEntity(entity);

        HttpResponse response = httpClient.execute(httpPost);

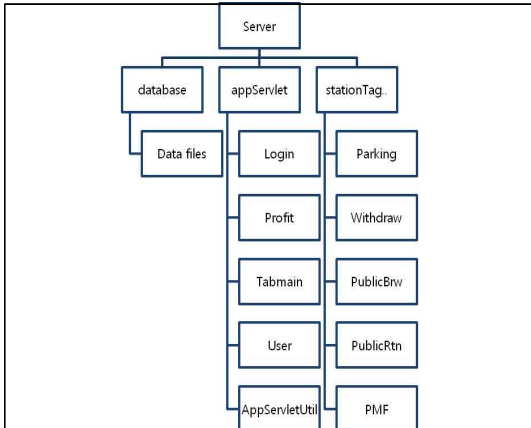
        HttpEntity RespEntity = response.getEntity();

        String parsingResult = parsingData(RespEntity.getContent(), b);
        System.out.println(parsingResult);
    } catch (Exception e)
    {
    }
}
    
```

<그림 9> 서버 접속 처리

<그림 9>는 자전거 스테이션에서 발생된 주차정보를 서버로 전송하는 부분을 보여주는 소스 코드이다.

3.2.2 중개 서버



<그림 10> 서버 Servlet 구조

<그림 10>은 웹 서버에서 구동되는 서블릿들의 모음이다. 그림에서 보듯이 서버에 올라가는 서블릿의 구조는 크게 3부분으로 나뉜다. 먼저 database package는 각종 데이터베이스 자바 빈 파일을 저장한다. GAE의 데이터 스토어의 테이블 구조는 자바 빈 클래스 형식으로 정의한다. 다음으로 appServlet package는 어플리케이션에서 요청하는 기능들을 제공하는 서블릿들이 저장된다.

어플리케이션에서 스테이션의 정보를 요청할 때 app 패키지의 서블릿들이 각각 맡은 영역의 정보를 DB에서 검색해서 JSON타입으로 전송한다.

	PRIMARY KEY	FOREIGN KEY			
UserA	Usenum	Bycenum	NameA	Phonenum	Price
Bycle	Bycenum	Locationnum	Stationnum	Type	
Station	Locationnum	Stationnum	Isable		
Card	Cardnum	NameA	Phonenum	Price	
RenReturn	Num	Cardnum	Usenum	Rentaltime	Returntime

<그림 11> 데이터베이스 테이블

마지막으로 stationTagReader package는 스테이션에서 넘어온 정보들을 저장하고 관리하는 서블릿들이 저장된다. 개인자전거의 주차 혹은 대여되는 공용 자전거의 태그정보를 가지고 자전거의 소유주, 주차시간, 주차위치, 대여시간, 대여위치 등을 DB에 저장한다. 이후에 대여자전거의 반납이나 개인자전거의 인출 시 기존의 정보를 검색하여 요금산출과 같은 작업을 수행한다. 아래 <그림12>은 데이터베이스에 접근하는 객체로서 비용이 크기 때문에 싱글톤 패턴으로 구현한다.

```

Class Name : PMF
Desc : Google App Engine DataStore Persistence Manager Factory
메모리에 활성화하는데 비용도 크고 해플리케이션 주기 동안 약 한 번 만들어낼 수 있어서,
static final로 넣어두고 실제 일을 도와줄 PersistenceManager를 불러 준다.

package stationTagReader;

import javax.jdo.JDOHelper;

public final class PMF {
    private static final PersistenceManagerFactory pmfInstance =
        JDOHelper.getPersistenceManagerFactory("transactions-optional");

    private PMF() {}

    public static PersistenceManagerFactory get() {
        return pmfInstance;
    }
}
    
```

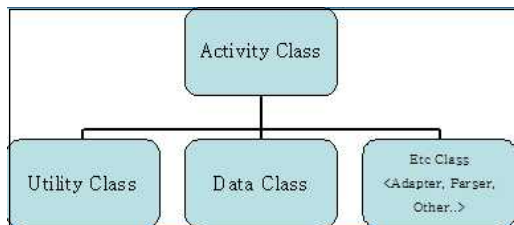
<그림 12> 데이터베이스 처리

3.2.3 안드로이드 애플리케이션

애플리케이션의 화면구성을 살펴보면 먼저 Activity Class는 애플리케이션의 각 화면을 담당하는 메인 클래스다. <그림13>에서 보듯이 각 화면을 구성하기 위해 다른 클래스의 객체를 이용한다. 다음으로 Utility Class는 Activity 클래스에서 사용되는 연산이나 기타 일을 모아 놓은 클래스이다.

Data Class는 웹 서버와 통신하면서 데이터를 가져오는 부분은 거의 모든 화면에서 중복이 되므로 하나의 클래스로 정의하여 공유한다. 또한 순차적으로 진행되면 해당 기능이 진행되는 동안 애플리케이션은 입력을 받거나 (ANR : Android Not Responding, 5초 이상 입력에 반응하지 않을 때 발생하는 오류) 다른 일을 수행 할 수 없기 때문에 스레드와 핸들러를 사용하여 구현한다.

마지막으로 Etc Class는 각 화면에 필요한 어댑터나 파서, 객체 클래스를 말한다.



<그림 13> 애플리케이션 Activity 구성

```

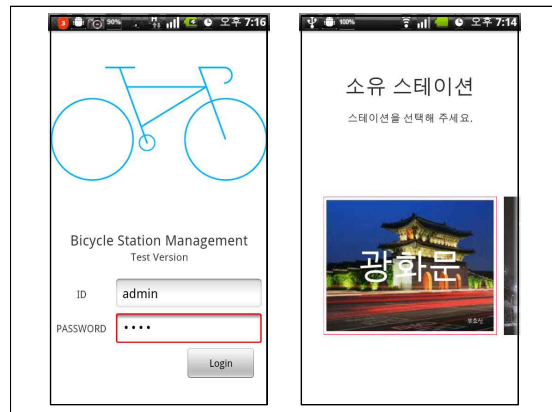
url.append("https://chart.googleapis.com/chart?cht=lx");
url.append("&choo=3399CC,FF9900,FF0000");
url.append("&chs=400x250");
url.append("&chdl=4E288C80&EC978AC142A8D09C&EC9D48B8&EC%A285&ED4958A9");
url.append("&chls=9|3|3");
url.append("&chm=N,3399CC,0,-1,12,-1|N,FF9900,1,-1,12,-1|N,FF0000,2,-1,15,1");
url.append("&schd=" + chartData);
url.append("&schxt=x,y");
url.append("&schma=|75");
url.append("&schds=0,0,0,+ max + ",0,0,0,+ max + ",0,0,0,+ max");
url.append("&schtr=4EC88898&EC9D48B8&ED4958A9");
url.append("&chxl=" + garoLabel + seroLabel);
url.append("&chts=000000,20");//*1|OR|50R|100R|"

return url.toString();
    
```

<그림 14> 수익률 그래프 구현

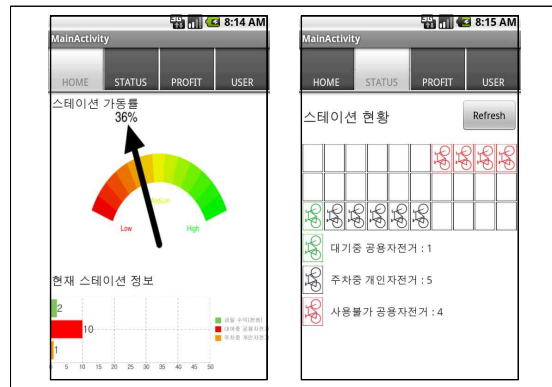
3.2.3.1 애플리케이션 화면구성

아래는 애플리케이션의 주요 화면별 스냅샷 및 설명이다. <그림 15>의 경우 로그인과 소유스테이션 선택화면을 나타낸다.



<그림 15> 로그인 & 소유 스테이션 선택 화면

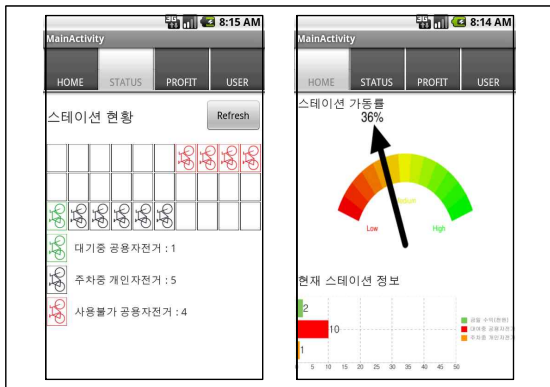
<그림 16> 홈 화면은 현재 스테이션의 종합적인 상황을 요약하고, 주차 현황은 스테이션 내부의 현황을 층별로 구분하여 그래픽화 하여 보여준다.



<그림 16> 홈 & 주차 현황 화면

다음 <그림 17>에서는 수익률과 유저에 관련된 화면이다. 수익률에서는 선택한 스테이션에서 발생하는 수익

을 차트화하여 제공한다. 유저 화면의 경우 무인 자전거 스테이션에 개인자전거를 등록하여 사용하는 유저의 회원정보를 검색할 수 있다.



<그림 17> 수익률 & 유저 화면



<그림 18> 현재 국민대 스테이션 모습



<그림 19> 3대의 자전거 주차

IV. 실험 결과

가상의 국민대 스테이션에 3대의 개인 자전거가 주차되는 상황을 설정하고 주차가 완료되었을 때 관리자가 안드로이드 스마트폰을 통해 해당 정보를 확인하는 실험을 실행 한다.

기존시스템과 비교하여 실험을 진행하고 결과를 내는 것이 타당하나, 해당 주제와 같은 시스템은 아직 찾아볼 수 없어 비교할 수 없고 비교가 가능한 시스템은 인력이 관제센터에 상주하는 시스템만 존재한다. 상주인력 시스템에 비교 가능한 항목은 경제성과 장소에 관계없는 실시간 관리 이점이다. 따라서 이번 실험에서는 실시간 스테이션 주차 현황 관찰에 관하여 진행한다.

<그림 18>을 보면 해당 국민대 스테이션에는 주차된 개인자전거가 없다는 것을 애플리케이션으로 확인한다.

다음 <그림 19>은 국민대 스테이션에 3대의 개인 자전거가 동시에 주차된 하면을 보여준다.

<그림 20>에서 보듯이 관리자가 스마트폰 애플리케이션을 통해서 주차가 된 상황을 볼 수 있다. 또한 <그림



<그림 20> 3대의 개인자전거 주차

21>처럼 해당 자전거를 터치하면 자전거의 상세 정보를 알 수 있다. 이와 같이 관리자는 스테이션내부의 자전거

주차 정보를 알아 볼 수 있다. 뿐만 아니라 수익률 및 가입 회원정보 검색 등 여러 가지 정보를 종합적으로 확인할 수 있다.



<그림 21> 자전거 상세 정보

V. 결론



<그림 22> 기존 무인 자전거 대여 시스템[7]

<그림 22>는 기존 무인 자전거 대여 시스템을 보여준다. 이렇듯 기존 자전거 관리 시스템들은 지자체 중심으로 확산되어 가는 추세이며 별도의 관제실에서 상주인력을 통해 시스템을 관리한다. 또한 자전거 스테이션을 시찰 하기위하여 직접 인력이 투입되어 여러 지역에 있는 자전거 스테이션에 직접 찾아가야 된다.

하지만 자전거 스테이션 소유자가 본 논문에서 개발된 스마트폰 어플리케이션을 사용한다면, 여러 지역에

있는 자신 소유의 자전거 스테이션을 직접 찾아가지 않고 자전거 스테이션의 상황을 파악할 수 있으며, 이는 상주인력을 두지 않음으로 인한 경제성을 도모할 수 있고, 원격으로 자신 소유의 자전거 스테이션의 상황을 파악할 수 있는 편리한 이점이 생긴다.

본 논문에서는 자전거 스테이션이라는 하나의 제한된 주제로서 설명 하였지만 이런 스마트폰 어플리케이션을 이용한 원격 관리는 얼마든지 다른 사업이나 산업에 응용되어 적용될 수 있겠다.

참고문헌

- [1] 김선진, 김내수, RFID/USN 산업동향 및 발전전망, [ETRI]전자통신동향분석. 제 20권, 제3호, 2005년.
- [2] Google App Engine, Alexander Zahariev Helsinki University of Technology, a. zahariev@abv.bg
- [3] 모바일 개방형 플랫폼 - 안드로이드(Android) 정보통신산업진흥원, [IITA] 정보통신연구진흥원 학술정보 인터넷이슈리포트 2007권 12호.
- [4] 정병호, 김상철, UHF RFID 기술을 이용한 자전거 주차 관리 시스템 개발, 디지털산업정보학회, 디지털산업정보학회 논문지, 제7권, 제2호, 2011년 6월.
- [5] 고석용, 김휴찬, 김형수, UHF RFID를 활용한 후대 지 짐품인증시스템 설계 및 구현, 디지털산업정보학회, 디지털산업정보학회 논문지, 제4권, 제2호, 2008년 6월.
- [6] D Crockford, The application/json media type for javascript object notation (json), Internet Engineering Task Force IETF Request for Comments (2006).
- [7] 황정훈, 대구시 공공자전거 시스템 도입 방안, 2009.

■ 저자소개 ■



정 성 훈
Jung, Sung Hoon

2004년 3월~현재
국민대학교 컴퓨터공학부 학사 과정

관심분야 : 모바일, 웹
E-mail : necropolise@naver.com



김 상 철
Kim, Sang Chul

2006년 3월~현재
국민대학교 컴퓨터공학부 부교수
2005년 미국 오클라호마 주립대학교
Electrical&Computer Eng.
(공학박사)
1994년~1999년
삼성 SDS, 삼성 테크윈 시스템
엔지니어

관심분야 : 정보통신 및 이동무선통신망
E-mail : sckim7@kookmin.ac.kr

논문접수일 : 2011년 8월 10일
수정일 : 2011년 11월 03일
게재확정일 : 2011년 11월 23일