안드로이드 기반 푸드코트 종합 관리시스템

이용덕°*, 이지영*, 최영미*, 공기석*, 이상우**
*한국산업기술대학교 컴퓨터공학과
**유니챌 주식회사
e-mail:{footballboy°, l9j13y, cymi0118, kskong}@kpu.ac.kr
swlee@unichal.com

Android based food court management system

Yong-Duk Lee°*, Ji-Young Lee* , Young-Mi Choi*, Ki-Sok Kong*,
Sang-Woo Lee**
*Department of Computer Engineering, Korea Polytechnic University
**Unichal, Inc.

요 약

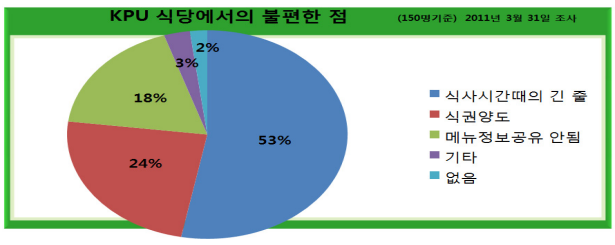
식당 이용에 있어서 평소 특정 시간대에 사람이 많이 몰려 혼잡한 상황을 완화시켜보고자 이 시스템을 창안하였다. 사용자가 음식예약, 음식나눔(SNS), 공지사항을 선택하면 각 선택메뉴에 맞는 서비스를 제공해 준다. 음식예약을 선택하면 원하는 만큼의 음식을 예약 할 수 있고, 사용자는 자신의 주문 내역을 확인할 수 있다. 이 어플리케이션을 사용하기 위해서는 회원으로 가입이 되어 있어야 하고, SNS의 경우 트위터API를 사용하여 연동 구현을 하였다.

1. 서론

최근 스마트폰 사용자 통계에 따르면, 스마트폰을 사용하는 인원이 09년 12월에 80만명, 11년 6월은 1500만명으로 집계된 바 있다. 요즈음은 TV에서도 일반 휴대폰 광고 보다 스마트폰 광고가 그 자리를 대신하고 있으며, 이에서 스마트폰 사용자의 수가 기하급수적으로 증가하고 있는 추세다[1][2].

스마트폰의 편리함은 굉장하다. 걸어 다니는 컴퓨터라고 말할 수 있을 정도로, 정보검색, 길찾기, SNS, 예약 등 다양한 서비스를 제공해주는 만능 폰이다.

이 만큼 스마트폰은 현대사회에 있어서 없어서는 안 될 필수 불가결한 존재가 되어가고 있다. 우리는 스마트폰의 장점 중, 예약 부분을 이용하여 학교식당관련 앱을 개발하였다. 음식예약시스템을 만들게 된 계기는 식당 이용에서부터였다. 개강과 동시에, 대학구내식당은 늘 분주하다. 지난 3월, 식당을 이용하는 학생들을 대상으로 설문조사를 한 결과 (그림 1)과 같이 식당 이용 시 가장 불편한 점으로 식사시간대의 혼잡을 1위로 뽑았다.



(그림 1) 한국산업기술대학교 식당에서의 불편한 점

우리 시스템은 이러한 불편함을 해소하여 쾌적한 푸드코트의 이용을 위하여 고안되었다. 예약을 통하여 손쉽게 주문 및 결제와 함께 예약내역을 살펴 볼 수도 있다. 또한 SNS 기능을 통하여 메뉴에 대해서 이용자 간의 다양한 정보를 공유할 수 있다. 관리자 입장에서는 공지사항이 있을 시, 프로그램을 통해 손쉽게 고객들에게 공지사항을 알릴 수 있다.

2. 관련 연구

2.1 기존 유사 앱과의 비교

아래의 <표1>은 본 논문의 앱과 기존의 유사 앱들 간의 기능에 대해서 비교하고 있다.

<표1> 기존 관련사례 종합

	롯데시네마	SNS서비스	SK M&C	3C
인증기능	로그인 인증	로그인 인증	-로그인인증 OR -인증번호를통한인증	**로그인 or 핸드폰번호 인증**
커뮤니티기능	–	SNS를 통한 커뮤니티	–	**SNS를 통한 푸드메뉴정보 공유**
예약 기능	원하는 날짜의 예약 가능 (4매까지 가능)	-	선물기능밖에 없음	**원하는 날짜의 예약 및 개수 선택 가능 (개수 제한 없음)**

어플리케이션 시장에서는 본 어플리케이션과 같이 학교식당에서 제공해주는 앱이 개발되어있지 않다. 그래서 우리는 예약, SNS처럼 각각 기능을 주로 가지고 있는 앱들을 비교분석한 뒤, 각각의 장점들을 통합하여 사용하기

편리한 음식종합예약시스템을 개발하였다.

기존에 있는 롯데시네마 예약시스템과 다른 점은 학생 식당의 특성상 많은 사용자들이 있으며 시간이 정해져 있는 것이 아니기 때문에 좌석 예약은 비효율적이라 생각하여 좌석예약제는 반영하지 않았다.

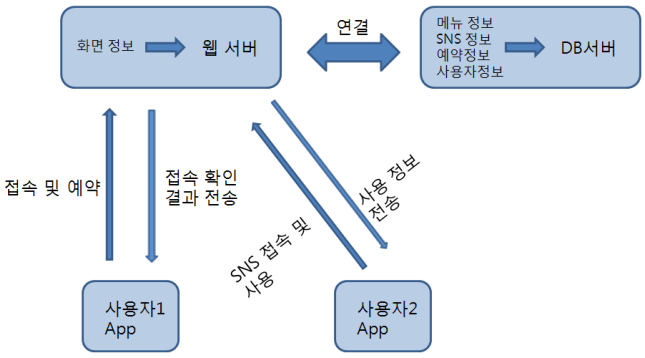
2.2 기존 임베디드리눅스와 안드로이드 차이점

기존 임베디드리눅스와 안드로이드의 차이점 중 한 가지는 장치 드라이버들(그래픽, 오디오, 카메라 등)이 커널 영역이 아닌 유저모드에서 동작한다는 점이다. 여기서 이야기하는 유저 모드란 커널영역을 제외한 하드웨어 추상 레이어를 의미한다. 유저 모드로 동작시키면 하드웨어 제어 시 MMU등을 참조하는 등의 추가적인 작업이 필요하기 때문에 조금 불편할 수 있지만 자신에게 할당되어 있는 메모리 번지에만 접근할 수 있기 때문에 조금 더 안전하다[3].

3. 설계 및 구현

3.1 시스템 구성

아래 그림은 음식종합관리시스템의 구조를 보여준다.

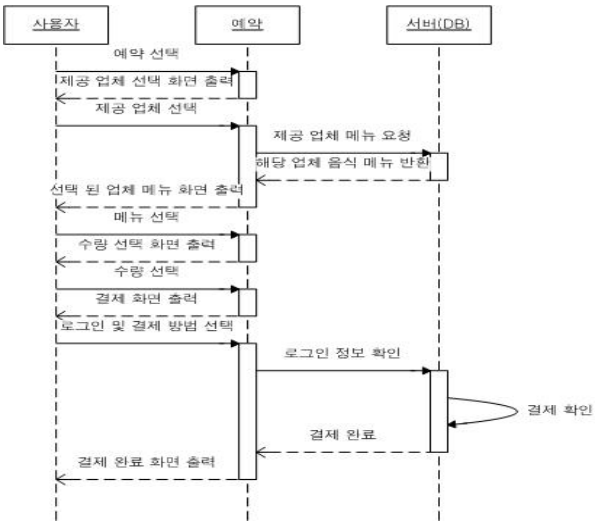


(그림 2) 음식 예약시스템 구조

이 어플리케이션의 기본적인 동작방식은 클라이언트와 서버간의 통신 방법이다. 서버에는 DB가 있고 그 DB에는 음식메뉴에 대한 정보뿐만 아니라 예약 정보라든지 공지사항 이라든지 하는 정보들이 담겨 있다. 이런 정보들은 사용자단의 클라이언트가 동작을 시작할 때 서버로부터 정보를 받아와 사용자의 화면에 출력해주게 된다[5].

3.2 설계

이 어플리케이션은 클라이언트/서버 방식으로 운영이 된다. SNS의 운영은 트위터의 API를 사용하기 때문에 트위터에서 데이터를 받는다. 안드로이드 플랫폼에서 운영되는 것이 클라이언트이고 메뉴나 공지사항 등을 담고 있는 것이 서버가 된다. 클라이언트가 실행이 되고 사용자가 예약, 공지사항 등을 선택하면 서버에서 해당되는 정보들을 받아와 화면에 출력하게 된다. 아래의 그림 3은 사용자가 예약을 선택하는 절차를 시퀀스 다이어그램으로 작성한 것이다.



(그림 3) 예약 시퀀스 다이어그램

3.3 개발환경

본 논문의 어플리케이션은 윈도우 XP 환경에서 개발되었다. Java SE Development Kit (JDK6) 설치, Eclipse IDE 설치, Eclipse IDE 에 Android Plug-in 설치, Android SDK 2.2 설치 후 실제 단말기와 에뮬레이터에서 개발되었다. 개발에 사용된 단말기는 삼성에서 개발한 '갤럭시탭'이다. Wi-Fi, 멀티터치 등 본 어플리케이션이 작동하기에 충분한 기능을 갖추고 있기에 선택하였다[4].

3.4 핵심 알고리즘

3.4.1 main 구성

(그림 4)를 보면 "메인화면(main.xml)"에 "상단 메뉴바(top.xml)"를 "include"하여 레이아웃을 구성하였음을 확인할 수 있다.

```
<include layout="@layout/top" android:id="@+id/top" android:layout_width="fill_parent"
 android:layout_height="30px" />
```

(그림 4) 상단 메뉴바를 메인화면에 include

"상단 메뉴바"는 다른 화면에서도 공통으로 사용할 레이아웃이다. 이러한 경우, 독립적인 레이아웃 파일을 만들고, 이 파일을 "include"하면, 각 화면마다 디자인을 해야 하는 번거로움을 없앨 수 있고, 유지 및 관리 차원에서 수정해야 할 필요가 발생될 때에 매우 편하다.

메인화면에 "공지사항", "음식나눔터", "음식예약" 3가지 내용을 메뉴 아래영역에 출력한다. 이는 3개의 화면을 하나의 화면에 출력하는 것을 의미한다.

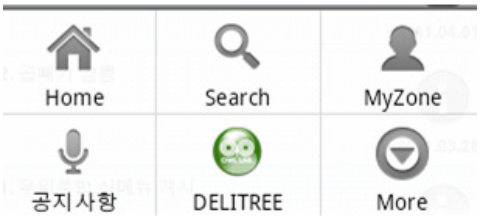
이 경우 역시 "상단 메뉴바"와 같은 방식으로 각 메뉴마다 독립적인 "레이아웃 XML"파일을 만들고 이 파일을 "main.xml"에서 "include"하는 형식으로 구현할 수 있다.

3개의 레이아웃에 대한 액티비티는 (그림 5)에 정의했다.

```
public void mainContentToggle(View v) {
        findViewById(R.id.main_tab1).setVisibility(View.INVISIBLE);
        findViewById(R.id.main_tab2).setVisibility(View.INVISIBLE);
        findViewById(R.id.main_tab3).setVisibility(View.INVISIBLE);

        if (v.getId() == R.id.mainMenu2) {
                findViewById(R.id.main_tab2).setVisibility(View.VISIBLE);
                loadInfoHTML();
        } else if (v.getId() == R.id.mainMenu3) {
                findViewById(R.id.main_tab3).setVisibility(View.VISIBLE);
                category categoryTmp = new category(this);
                categoryTmp.getXMLDataList();
        } else {
                findViewById(R.id.main_tab1).setVisibility(View.VISIBLE);
                notice noticeTmp = new notice(this);
                noticeTmp.getXMLDataList();
        }
```

(그림 5) mainSite.java

(그림 6) 탭으로 구별된 메인화면



(그림 7) 옵션메뉴

3.4.2 SNS 연동 (HTML뷰어)

개발한 안드로이드 앱에서는 매뉴얼에서 보는 바와 같이 HTML기반으로 구성된 웹페이지를 출력할 수 있는 “android.webkit”을 제공한다. 이에 속하는 클래스 중에 가장 중심이 되는 클래스가 “WebView”클래스다.

“메인화면”의 “음식나눔터”탭에서 “WebView”를 사용하여 앱 정보를 동적으로 출력하는 형식을 구현하는 과정을 통해 ”앱(App)”과 “웹(Web)”을 연동한다.

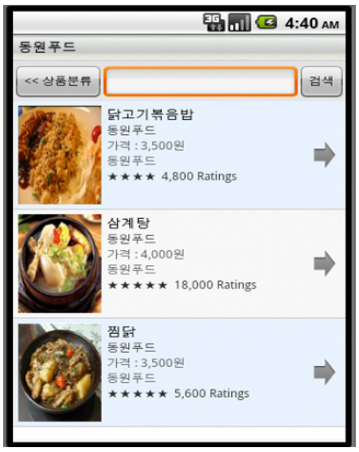
```
public void loadInfoHTML() {
        WebView webView = (WebView) findViewById(R.id.mainWebView);
    webView.getSettings().setJavaScriptEnabled(true);
    webView.getSettings().setDefaultZoom(ZoomDensity.FAR);
    webView.getSettings().setSupportZoom(false);
    webView.loadUrl("http://twitter.com/#!/kpu3c");

}
```

(그림 8) mainSite.java

(그림 9) 앱과 웹 연동을 통한 SNS연결

3.4.3 음식 예약 기능

목록에 출력할 이미지 객체를 담아 둘 HashMap 객체를 선언한다. 목록을 스크롤할 때 화면에 행이 나타나면서 getView()메소드를 실행하는데 웹서버에서 한번 가져왔던 이미지를 메모리에 기록해두지 않으면, 매 번 웹서버에서 이미지를 반복적으로 가져오기 때문에 불필요한 트래픽을 유발한다. 이러한 소모성 트래픽을 방지하기 위해 (그림 11)와 같이 웹서버에서 한번 가져온 이미지들을 HashMap을 통하여 메모리에 임시로 기록해 둘 필요가 있다.



(그림 10) 음식예약 이미지 출력

```
Uri uri = Uri.parse(hm.get("img[" + position + "]"));
Log.v(TAG, uri.getPath());

try {
        if (hmImg.get("img[" + position + "]") == null) {
                String urlstr = hm.get("img[" + position + "]");
                URL url = new URL(urlstr);
                URLConnection conn = url.openConnection();
                conn.connect();
                BufferedInputStream bis = new BufferedInputStream(conn
                                .getInputStream(), 512 * 1024);
                Bitmap bm = BitmapFactory.decodeStream(bis);
                bis.close();
                hmImg.put("img[" + position + "]", bm);
        }
        if (hmImg.get("img[" + position + "]") != null) {
                ((ImageView) convertView.findViewById(R.id.prodImg))
                                .setImageBitmap(hmImg.get("img[" + position + "]"));
        }

} catch (IOException e) {
        Log.e(TAG, e.toString());
        ((ImageView) convertView.findViewById(R.id.prodImg))
                        .setImageDrawable(convertView.getResources().getDrawable(
                                        R.drawable.blank));
}
```

(그림 11) 웹 서버에서 음식예약 이미지 출력하는 소스

3.4.4 공지사항 기능

Structure구조를 이용하여 클래스 간에 데이터를 공유하는 방법은 일일이 데이터를 담는 객체를 선언해서 ArrayList에 담아야 하고, 데이터 항목에 변동이 있거나 추가될 때 수정해야 할 소스가 많아 개발하는데 매우 번거로웠다.

그래서 ListAdapter와 Activity가 데이터를 공유하는데 Key:Value형식으로 데이터를 저장하고 호출하는 HashMap방식을 사용했다. ArrayList를 이용한 noticeListAdapter생성을 그대로 두고, 공지 데이터인 hm을 noticeListAdapter객체에 전달하는 방식을 이용하였다.

```
ArrayList<noticeMainItem> noticeMI = new ArrayList<noticeMainItem>();
hm = xml2HashMap(tmpData);
for (int i = 0; i < Integer.parseInt(hm.get("count")); i++) {
        noticeMI.add(new noticeMainItem(hm.get("subject[" + i + "]"), hm
                        .get("reg_date[" + i + "]")));
}

noticeListAdapter noticeListAdapter = new noticeListAdapter(act,
                R.layout.notice_row_main, noticeMI);
noticeListAdapter.hm = hm;

ListView noticeListMain = (ListView) act.findViewById(R.id.noticeListMain);
noticeListMain.setAdapter(noticeListAdapter);
}
```

(그림 12) 공지사항 소스

(그림 13) 공지사항

4. 결론 및 향후 연구과제

현재 안드로이드 관련 앱사이트[7]에서 제공하는 학교식당 관련 앱은 없다. 본 어플리케이션을 사용하면 원하는 날짜에 학교식당을 보다 편리하게 사용할 수 있다는 장점이 있다. 부가적으로 자신이 먹은 음식에 대한 정보공유가 가능하고, 공지사항 등을 확인해봄으로써 학교식당을 편리하게 이용할 수 있다.

우리 어플리케이션에는 아쉬운 점이 보이는데 세부적인 예약과 관리자 기능의 아쉬움이다. 초기에 이러한 점들을 깊게 생각하지 못하여서 구현이 완료된 후에 조언을 듣고서 작업을 해 보았지만 생각보다 기능을 붙이는 작업이 순조롭지 못했다. 향후에 이러한 기능들을 기존 프로그램에 통합하는 부분을 생각해 볼 필요가 있다고 생각한다.

이번 과제를 수행하면서 안드로이드 수행에 필요한 Java 문법과 다른 구현 방법들을 배우므로 해서 많은 경험과 지식을 쌓을 수 있었다. 다음에 또 다른 안드로이드 개발 기회가 있게 된다면 개발이 지금보다는 조금 더 쉬울 것이라 생각한다.

참고문헌

[1] 스마트폰 사용자 1500만 시대, 그 실태는? "http://www.newsway.kr/news/articleView.html?idxno=125103".

[2] 강현주, 조경국, 이윤환 지음, 갤럭시탭 USING BIBLE, 황금부엉이, 2011년 7월.

[3] 박헌재 지음, 전문가로 가는 안드로이드 마스터 프로젝트, 교학사, 2010년 8월.

[4] 로런다시, 셰인콘더 지음(이대엽 역), 하루 한 시간 자율학습 안드로이드 프로그래밍, 위키북스, 2010년 9월.

[5] 로런다시, 셰인콘더 지음(류광 역), 시작하세요! 안드로이드프로그래밍: 모바일 소프트웨어 개발, 위키북스, 2009년 11월.

[6] 구글 안드로이드 개발자 홈페이지 "http://developer.android.com".

[7] 안드로이드 앱스 "http://www.androidapps.kr/".