

# 파싱 알고리즘을 이용한 편의점 정보 검색 시스템 구현

김승욱, 유희경, 정인철, 김태우\*  
성공회대학교 정보통신공학과

## Implementation of The Shopping Information Retrieval System using Parsing Algorithm

Seung-Uk Kim, Hee-Gyeong You, In-Cheol Jeong, Tai-Woo Kim\*  
Department of Computer and Communication, SungKongHoe University

**요약** 소비자가 각 편의점에서 제공하는 제품과 행사 상품 등을 값싸게 구매하기 위해서는 해당 편의점의 홈페이지를 들어가거나 직접 방문하여 찾아야 한다. 본 연구에서는 각 편의점에서 어떤 제품이 행사 상품이며, 어떤 편의점에서 어떤 서비스를 제공하는지 여부를 검색하는 앱 프로그램을 개발한다. 본 앱을 사용하여 사용자는 여러 편의점에서 제공하는 행사 정보를 비롯한 각종 서비스를 실시간을 검색할 수 있다.

**주제어** : 파싱, 알고리즘, 안드로이드, 편의점, 정보 검색

**Abstract** In order for consumers to purchase products and event products offered at each convenience store at a low price, visitors should visit the convenience store's homepage or visit the store directly. In this study, we developed an app program for each convenience store to find out which products are event products and which services are provided at certain convenience stores. Using this app, users can search real-time on various services including event information provided by various convenience stores.

**Key Words** : Android App, Convenience Store, Parsing Algorithm, Information Retrieval

### 1. 서론

현재 초등학생부터 노인까지 남녀노소 상관없이 편하게 사용하는 편의점의 개수가 급속도로 늘어나고 있다. 심지어 1인 가족 형태의 증가와 함께 편의점을 활용해 식생활을 해결하려는 경향도 늘고 있다.

하지만 소비자가 여러 가지 이유로 편의점을 방문하여 상품을 구매하는 경우, 각 편의점에서 제공하는 제품과 행사 상품 등을 값싸게 구매하기 위해서는 해당 편의점의 홈페이지를 들어가거나 직접 방문하여 찾아보지 않으면 어떤 제품이 행사 상품이며, 어떤 편의점에서 어떤

서비스를 제공하는지 여부를 알 수 없다.

본 연구에서는 여러 편의점에서 제공하는 행사 정보를 비롯한 각종 서비스를 실시간 검색하여 소비자에게 제공하는 시스템을 개발한다. 본 연구를 통하여 소비자 들은 각 편의점의 행사 정보와 함께 제공 서비스를 실시간으로 검색해서 값싸게 제품을 구매하거나 최적의 서비스를 이용할 수 있다. 본 연구에서 제공되는 지도 검색을 통하여 가장 가까운 편의점을 방문할 수 있도록 도와주거나, 댓글을 통해 각 제품의 평가를 할 수 있도록 하여 실생활에서 활용도가 높으며, 간편하게 사용할 수 있는 편의점 정보 검색 시스템(일명 편돌이)을 개발하였다.

본 논문은 2016년도 성공회대학교 정보통신공학과 통신개발 프로젝트 결과물입니다.

\*교신저자 : 김태우(ktw@skhu.ac.kr)

접수일 2016년 11월 20일 심사완료일 2016년 12월 12일

본 연구의 의의는 CU, GS25, 세븐 일레븐, 미니스톱 등과 같은 각기 다른 회사 편의점에서 각각 제공하는 각종 상품 정보를 통합하여 일관적으로 사용자에게 제공하여 실생활에 활용될 수 있는 시스템을 개발하였다는 점이다. 또한, 소비자가 단지 휴대폰만 사용하여 각 상품의 평가를 공유하거나 회사별 행사 상품 특징을 비교하여 소비자가 경제적이고 효율적인 구매할 수 있다는 점에서 그 연구의 의의가 있다.

## 2. 관련 연구

본 연구를 위해 개발된 편의점의 개발에 필요한 기법 들로는 각 편의점 홈페이지에서 필요한 상품 항목을 갖고 오기 위한 파싱(Parsing) 기법과 갖고 온 항목을 정규 화하기 위한 문자열 치환 기법, 그리고 여러 필드의 정보 로 구성된 상품의 각 항목을 구분해주는 토큰화 (Tokenizer) 기법 등이 있다. 다음은 본 연구에서 사용된 기법 또는 알고리즘들이다.

### 2.1 JSOUP 파싱 기법

JAVA에서의 파싱에는 HTML 파싱, XML 파싱, JSON 파싱으로 나누어지며, 본 연구에서 사용하는 HTML 파싱은 예전에 HttpURLConnection() 함수를 사용하였던 방법을 사용하였다. 하지만 이를 좀 더 쉽게 만들어 파이선의 BeautifulSoup와 유사한 JSOUP 라이브 러리를 사용한 파싱 기법이다[1].

HTML 파싱 기법에서 파싱 데이터를 갖고 오는 방법 에 따라 파싱 결과물을 순서대로 한 줄씩 이벤트 핸들러 를 이용해 갖고 오는 SAX 방식과 파싱 결과물은 한 번에 갖고 와 메모리에 올린 후 트리 형태로 탐색하여 이용하는 DOM 방식이 있다.

본 연구에서 사용한 파싱 방법은 HTML 방식을 이용 한 DOM 방식의 JSOUP 파싱 기법이며, 이는 실제 사용 하는 HTML을 처리할 수 있는 자바 라이브러리이다.

JSOUP 라이브러리는 HTML 문서를 URL이나 파일, 문자열 형태로 전달해서 파싱할 수 있으며, DOM, CSS, JQuery와 유사한 메소드들을 사용할 수 있다. [Fig. 2.1] 은 JSOUP를 이용한 파싱 알고리즘의 예이다.

```
import java.io.IOException;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class Main{
    public static void main(String[] args) throws IOException {
        Document doc = Jsoup.connect("http://blog.acronym.co.kr").get();
        Elements titles = doc.select(".title");

        //print all titles in main page
        for(Element e: titles){
            System.out.println("text: " + e.text());
            System.out.println("html: " + e.html());
        }

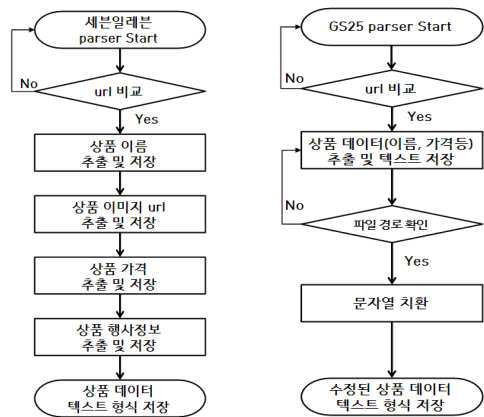
        //print all available links on page
        Elements links = doc.select("a[href]");
        for(Element l: links){
            System.out.println("link: " + l.attr("abs:href"));
        }
    }
}
```

[Fig. 2.1] JSOUP를 이용한 파싱 코드의 예[2]

### 2.2 문자열 치환 기법

CU, GS25, 세븐 일레븐, 미니스톱 등과 같은 각기 다 른 회사 편의점에서는 상품 정보를 각기 다른 형태로 갖고 있다. 본 연구에서는 각기 다른 형태의 정보를 통일된 형태로 DB에 저장하기 위해 문자열을 치환하는 기법을 사용하였다.[3]

본 연구에서 사용된 자바의 문자열 치환 메소드는 텍 스트 에디터의 문자열 바꾸기처럼 지정한 문자열을 기존 의 텍스트에서 찾아 자동으로 바꾸어주는 replace 클래스 를 사용한다. [Fig. 2.2]는 본 연구에서 세븐 일레븐과 GS25 편의점에 대해 파싱된 문자열을 통일된 형태로 DB 에 저장하기 위해 “(쌍 따옴표)를 삭제한 한 줄의 상품 정보로 치환하는 알고리즘이다. [Fig. 2.2]의 세븐 일레븐 에서는 데이터 구조가 DB구조와 일치하기 때문에 문자 열 치환을 하지 않는 반면 GS25에서는 문자열 치환을 한 다는 것이다.



[Fig. 2.2] 문자열 치환 알고리즘

### 2.3 토큰화 기법

자바에서 스트링 형태의 데이터를 토큰 단위로 끊어 주는 StringTokenizer 클래스를 제공한다. 문자열 치환으로 얻어진 상품 정보를 각 필드별로 분류하여 DB에 저장하기 위해서는 토큰화 기법이 필요하며, [Fig. 2.3]은 “성 공 회 대 학 교”를 입력받아 이를 처리하여 토큰으로 출력한 예이다.

```
import java.util.StringTokenizer;
public class h3bak_StringTokenizer {
    public static void main(String[] args) {
        StringTokenizer st =
            new StringTokenizer("성 공 회 대 학 교");
        // 토큰이 있는 동안 while문이 실행됨
        while(st.hasMoreTokens()){
            String temp = st.nextToken(); // 토큰을 temp 변수에 저장
            System.out.println("temp="+temp);
        }
        System.out.println("=====");
        // StringTokenizer(str, delim) str= 추출할문자, delim =구획문자
        StringTokenizer st1 =
            new StringTokenizer("성공회대학교", "");
        while(st1.hasMoreTokens()){
            String temp = st1.nextToken();
            if(st1.equals(""))(System.out.println("\n"));
            else (System.out.println(temp));
        }
    }
}
```

(a) 토큰화 알고리즘 예

```
temp=성
temp=공
temp=회
temp=대
temp=학
temp=교
=====
성
공
회
대
학
교
```

(b) 실행 결과

[Fig. 2.3] 토큰화 기법 예

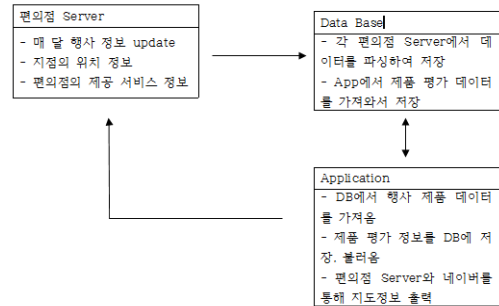
## 3. 편의점 정보 검색 시스템 설계 및 구현

### 3.1 편의점 정보 검색 시스템 구조

본 연구에서 제안하는 편의점 정보 검색 시스템(일명 편돌이)는 [Fig. 3.1]과 같이 구성되어 있다. 본 시스템은 크게 나누어 각 편의점에서 진행되는 행사 및 상품 정보를 갖고 와서 분석하는 파싱하는 부분, 갖고 온 상품 정보를 가공하여 DB에 저장하거나 검색하는 부분, 사용자와의 인터페이스 부분 등으로 구성되어 있다.

[Fig. 3.1]에서 편의점 서버 부분은 편의점 홈페이지에서 실시간으로 제공되는 각 상품 정보, 각 편의점의 위치

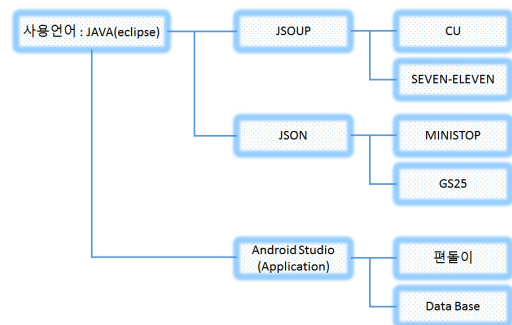
및 제공 서비스를 검색하여 본 시스템에 맞게 가공한다. DB 부분에서는 가공된 데이터를 저장하거나 안드로이드 앱에서 입력하거나 참조된 정보를 가공하여 DB에 저장한다.



[Fig. 3.1] 편돌이 시스템 구조

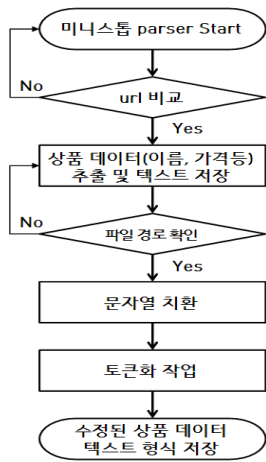
### 3.2 편의점 상품 정보 추출을 위한 파싱 알고리즘

편의점 홈페이지의 상품 정보를 갖고 오기 위해 본 연구에서 개발된 파싱 알고리즘은 각 편의점에 따라 그 형태와 구조가 다르다. 본 연구에서는 대표적인 편의점인 CU, GS25, 세븐 일레븐, 미니스톱 등의 홈페이지를 파싱하여 상품 정보를 갖고 오는 파싱 알고리즘을 각각 별도로 개발하였다. [Fig. 3.2]는 각 편의점에 따라 각각 다른 파싱 알고리즘을 사용하며, CU와 세븐 일레븐은 JSOUP 파싱 방식을 사용하는 반면, 미니스톱과 GS25는 JSON 파싱 기법을 사용한다는 것을 나타내고 있다.



[Fig. 3.2] 각 편의점에 따른 파싱 알고리즘

[Fig. 3.3]은 미니 스톱 홈페이지로부터 데이터를 추출하기 위한 파싱 알고리즘이다. [Fig. 3.3]에서 url 변수에 상품 페이지의 주소를 넣어주고, postUrl에 저장한다. 한글 인코딩을 위해서 속성 값에 “UTF-8”을 설정해준다.[4]



[Fig. 3.3] 미니 스톱 홈페이지 파싱 알고리즘

미니스톱의 경우 JSON구문으로 제작된 페이지이기 때문에, object형식으로 작성되어 있는 recordList의 값을 가져온다. 그 후, recordList 내의 배열 형식으로 작성되어 있는 항목의 값을 가져오며, 가져온 데이터는 텍스트 형식 파일로 저장된다.

편의점 홈페이지를 파싱하여 상품 정보를 갖고 올 경우 안드로이드 OS가 마시멜로우로 업그레이드되면서, 앱 실행에 필요한 Permission의 값을 직접 구동할 때 체크해 줘야 하도록 변경되었다. 따라서, Permission을 체크하는 코드를 MainActivity에 적어주어 처음 실행 시 해당 Permission 값을 체크하도록 하였고, Permission 값을 받아오지 못할 경우 RuntimePermission 클래스에 별도의 함수를 작성하여 다시 Permission을 받을 수 있도록 [Fig. 3.4]와 같이 체크하도록 설계하였다.[5]

```

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults){
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    int permissionCheck = PackageManager.PERMISSION_GRANTED;
    for(int permission : grantResults){
        permissionCheck = permissionCheck + permission;
    }
    if ((grantResults.length > 0) && PackageManager.PERMISSION_GRANTED == permissionCheck){
        onRequestGranted(requestCode);
    } else{
        Snackbar.make(findViewById(android.R.id.content), mErrorMessage, get(requestCode),
            Snackbar.LENGTH_INDEFINITE).setAction("ENABLE", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent();
                i.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
                i.setData(Uri.parse("package:" + getPackageName()));
                i.addCategory(Intent.CATEGORY_DEFAULT);
                i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                i.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
                i.addFlags(Intent.FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS);
                startActivity(i);
            }
        }).show();
    }
}
  
```

[Fig. 3.4] 안드로이드 Permission 체크 코드

```

product - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
["431","미장센미니확스M7","5000","3500","","[BH2SUZX0_]sale_04.jpg"]
["430","미장센미니확스M9","5000","3500","","[BH2SUZX0_]sale_03.jpg"]
["429","말레프렌기세정제(원벤더)","4900","3450","","[BH2SUZX0_]sale_02.jpg"]
["428","청성사이다500팩","2000","1600","","[BH2SUZX0_]sale_01.jpg"]
["431","미장센미니확스M7","5000","3500","","[BH2SUZX0_]sale_04.jpg"]
["430","미장센미니확스M9","5000","3500","","[BH2SUZX0_]sale_03.jpg"]
["429","말레프렌기세정제(원벤더)","4900","3450","","[BH2SUZX0_]sale_02.jpg"]
["428","말레프렌기세정제(원벤더)","4900","3450","","[BH2SUZX0_]sale_02.jpg"]
["428","청성사이다500팩","2000","1600","","[BH2SUZX0_]sale_01.jpg"]
  
```

[Fig. 3.5] 미니 스톱 파싱 결과 값

미니스톱 홈페이지를 파싱하여 추출된 상품 정보는 [Fig. 3.5]와 같으며, 여기서 각 줄은 각각 하나의 상품을 나타내고, 각 상품 정보는 "(쌍 따옴표)로 표시되고, 각 필드는 ,(콤마)로 구분된다.

### 3.3 문자열 치환 및 토큰화 알고리즘

편의점 홈페이지를 파싱하여 갖고 온 상품 정보는 DB에 저장하기 위해 문자열 치환과 함께 토큰화를 한다. 미니스톱과 GS25에는 각각의 문자열 치환 및 토큰화 알고리즘이 필요한 반면, CU와 세븐 일레븐에는 문자열 치환이 필요 없다. 본 연구에서는 모든 편의점에 각각에 대해 문자열 치환과 토큰화 알고리즘을 개발하였지만 본 논문에서는 미니스톱 편의점에 대해서만 기술한다. [Fig. 3.5]에서 얻어진 파싱 결과 값을 DB에 저장하기 위해서 데이터 정규화를 해주어야 한다.

[Fig.3.6]은 미니스톱 문자열 치환을 위한 코드이다. [Fig. 3.6]에서 bufferedReader를 통해서, 해당 파일을 불러오고, replaceAll을 이용하여 "(바꿀 문자열)", "(바뀐 문자열)"형식으로 입력해준다. 그리고, write를 통해 수정된 텍스트 파일을 경로에 저장한다.

```

package Ministop;
import java.io.*;
public class replace
{
    public static void main(String args[])
    {
        try
        {
            File file = new File("D:\convenience\Ministop\sale\product.txt");
            BufferedReader reader = new BufferedReader(new FileReader(file));
            String line = "", oldtext = "";
            while(line = reader.readLine() != null)
            {
                oldtext += line + "\r\n";
            }
            reader.close();
            String newtext = oldtext.replaceAll("(","");
            FileWriter writer = new FileWriter("D:\convenience\Ministop\sale\product.txt");
            writer.write(newtext);writer.close();
        }
        catch (IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
}
  
```

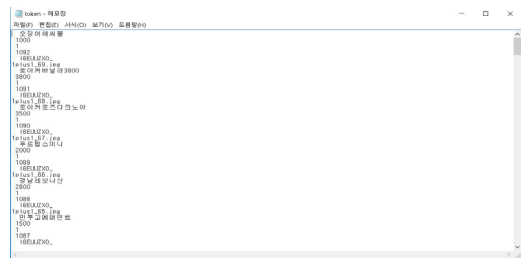
[Fig. 3.6] 미니스톱 문자열 치환 코드

문자열 치환을 통해 얻어진 "(쌍 따옴표)가 삭제된 파일을 DB에 저장하기 위해서는 토큰화 해야 하는데 이를 위한 코드는 [Fig. 3.7]과 같다.

```
public class tokenizer {
    public static void main(String[] args) throws IOException{
        FileReader reader = new FileReader("D:\convenience\Ministop\sale\product.txt");
        // 주수가 저장된 파일을 읽는다
        BufferedReader rd = new BufferedReader(reader);
        ArrayList<String> list = new ArrayList<String>();
        String line = null; // 파일로 부터 읽어들이는 조각들을 저장하는것
        String getLine = ""; // 조각들을 연결해서 하나의 String 으로 만드는것
        while ((line = rd.readLine()) != null) {
            System.out.println(line);
            getLine = getLine + line; }
        rd.close();
        // 파일로 부터 읽어 들인 문자열은 " 공백으로 구분해서 쉼라준다
        StringTokenizer st = new StringTokenizer(getLine, ",");
        while (st.hasMoreTokens()) {
            list.add(String.valueOf(st.nextToken())); }
        String st_list = "";
        FileWriter writer = new FileWriter("D:\convenience\Ministop\sale\token.txt");
        // 정렬한 값을 저장할 파일 생성
        for (int i = 0; i < list.size(); i++) {
            st_list = st_list + "\r\n" + list.get(i); }
        writer.write(st_list);
        writer.close();
        System.out.println("정렬종료"); } }
```

[Fig. 3.7] 문자열 토큰화 코드

[Fig. 3.7]에서 bufferedReader를 통하여, 치환된 텍스트 파일을 읽어오고, readLine으로 한 줄 씩 읽어서 저장하고 파일을 닫는다. StringTokenizer를 이용하여, 기준 문자를 선정하여 (“,”) 기준 문자를 만날 때 마다 줄 바꿈이 일어나도록 실행한 후, 정렬한 값을 파일에 저장시킨다.



[Fig. 3.8] 미니ストップ 편의점의 상품 정보 토큰

그 결과 [Fig. 3.8]과 같이 치환과 코드화가 완료된 텍스트 파일이 얻어진다. 최종 얻어진 상품 정보의 각 필드는 각각 한 줄로 표현된다.

### 3.4 데이터베이스 액세스를 위한 알고리즘

각 편의점 홈페이지를 파싱하여 얻어진 상품정보의 치환과 토큰화가 완성된 다음 DB에 저장해야 한다. 본 연구에서 사용한 DB는 최근에 많이 사용되는 Firebase를 사용하였으며, Firebase는 구글에서 제공하는 다목적 클라우드 서비스를 제공하는 데이터베이스이다. 파싱과 치환 및 토큰화가 종료된 데이터를 SQL을 복잡하게 사용하지 않고, 안드로이드 앱에서 사용하도록 설계되었다.

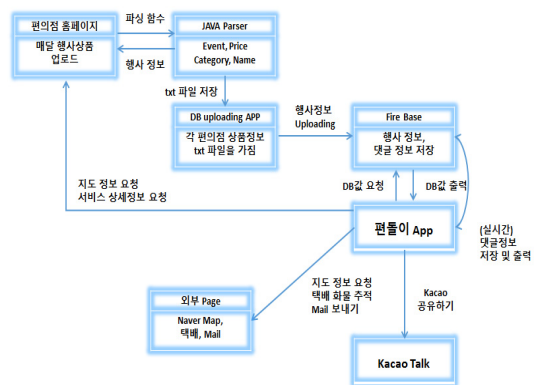
```
databaseReference.child("product").child("cu").addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        Product product = dataSnapshot.getValue(Product.class); // product를 가져오고
        if (product != null) {
            product_name[DBindex] = product.getProductLine();
            product_price[DBindex] = product.getProductPrice();
            event_value[DBindex] = product.getProductEvent();
            base_url[DBindex] = product.getProductURL();
            product_category[DBindex] = product.getProductCategory();
            DBindex++;
        }
    }
    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {
    }
    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {
    }
    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {
    }
    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});
```

[Fig. 3.9] Firebase 액세스 코드

[Fig. 3.9]는 Firebase 데이터베이스 액세스를 구현한 코드이며, 본 코드는 해당 DB의 Reference 값을 가져와서 사용한다. Reference 값을 가져온 Firebase에 대하여 EventListener을 구현하여, 데이터베이스에 값이 추가, 변경 될 때 마다 자동으로 값을 수정할 수 있도록 구현하였다. 하지만, 본 어플리케이션에서는 한 달에 한 번씩 행사 정보가 바뀌기만 할 뿐 다른 경우에 행사 상품에 대한 DB 값이 바뀌는 경우가 없으므로, 추가 되었을 때, 해당 값을 받아오는 부분의 메소드만 Override 하여서 사용하였다.

## 4. 편의점 정보 검색 시스템 구현 결과

본 연구에서 구현한 부분은 [Fig. 4.1]과 같이 구성되어 있다.



[Fig. 4.1] 편들이 구성도

편의점 홈페이지를 파싱하여 새로운 행사 정보를 데이터베이스에 저장하는 부분, 편돌이 앱에서 DB를 액세스하여 실시간으로 행사 정보와 댓글 정보를 검색하는 사용자 인터페이스 부분, 네이버나 구글 지도와 택배 회사와 같은 외부의 다른 사이트 정보를 액세스하기 위한 부분, 그리고 카카오 톡과 같은 외부 앱과 연동하기 위한 부분 등으로 구성되어 있다.

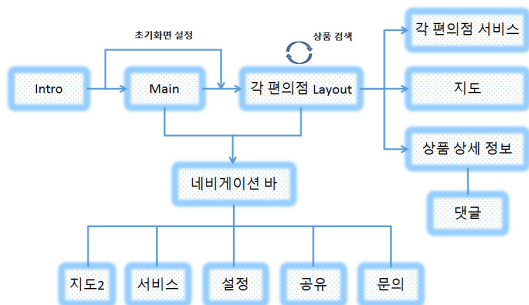
### 4.1 편돌이 제공 기능

본 연구에서 제안된 시스템의 주요 기능들은 다음과 같다.

- ① 각 편의점 마다 행사 중인 제품을 실시간으로 나열하여 출력하여 사용자가 관심 제품 정보를 볼 수 있다.
- ② 검색창, 검색 옵션을 통하여 원하는 제품을 빠르게 검색하는 기능을 제공한다.
- ③ 자주 이용하는 편의점을 초기 화면으로 설정하거나 기타 여러 가지 설정을 하는 기능을 제공한다.
- ④ 각 편의점 마다 각각 다르게 제공되는 서비스를 분류하거나 분석하여 일관적으로 표시하여 주는 기능을 제공한다.
- ⑤ 각 편의점의 제공 서비스에 따라 지도 검색을 통하여 편의점을 찾는 서비스와 GPS를 기반으로 자신의 위치에 따른 편의점 검색 서비스를 제공한다.
- ⑥ 사용자가 해당 제품의 평가, 별점을 댓글로 기록하여 다른 소비자가 해당 제품 구매에 도움을 주는 기능을 제공한다.

### 4.2 모바일 앱

편돌이 앱의 전반적인 구성도는 [Fig. 4.2]와 같다.



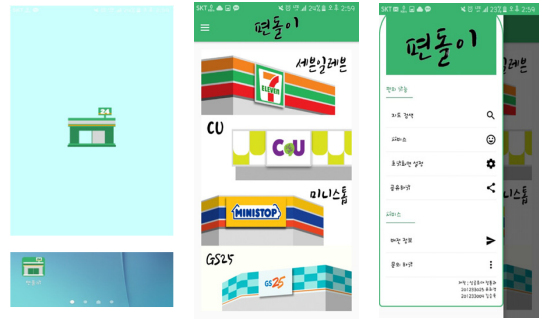
[Fig. 4.2] 편돌이 앱 블록다이어그램

[Fig. 4.2]에서 편돌이 앱은 초기 화면 및 설정을 거쳐 구동되며, 각 편의점에 따라 제공되는 행사 상품과 서비스, 지도 등을 검색하는 기능을 갖는다.

#### 4.2.1 초기 화면

[Fig. 4.3]은 편돌이의 초기 화면이며, [Fig. 4.3](a)는 앱을 클릭했을 때의 스플래시 화면이며, (b)는 CU, GS25, 세븐 일레븐, 미니스톱 등과 같이 편의점을 선택할 수 있으며, (c)는 앱에서의 다양한 편의 기능을 제공하는 화면이다.

[Fig. 4.3]에서 메인 화면과, 좌측 상단의 버튼을 누르면 나오는 네비게이션 바이다. 메인 화면에서 보이는 4개의 편의점 이미지는 각 편의점 사진을 보고 포토샷을 사용하여 그렸으며, 모두 버튼 형식으로 사용되는 부분이다. 네비게이션 바는, 안드로이드 스튜디오에서 제공하는 아이콘을 사용하였으며, XML파일을 사용하여 직접 디자인하였다.

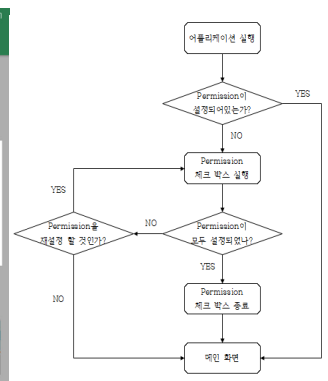


(a) 시작화면 (b) 편의점 선택화면 (c) 검색 화면

[Fig. 4.3] 편돌이 초기 화면



(a) 앱 설정 화면



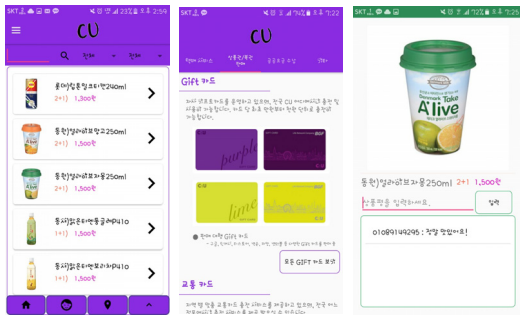
(b) 앱 알고리즘

[Fig. 4.4] Permission 설정을 위한 화면과 알고리즘

[Fig. 4.4]는 각 편의점 홈페이지로부터 파싱 정보를 갖고 오고 앱을 정상적으로 구동하기 위하여 필요한 permission을 설정하기 위한 화면과 안드로이드 앱 코드 알고리즘이다.

### 4.2.2 화면 레이아웃

[Fig. 4.5]는 편의점 레이아웃과, 서비스 레이아웃, 상품 정보 레이아웃이다.



(a) 편의점레이아웃 (b) 서비스레이아웃 (c) 상품정보 레이아웃

[Fig. 4.5] 화면 레이아웃

편의점 레이아웃의 경우 각 편의점마다 색을 다르게 구현하였으며, RecyclerView를 커스터마이징 하여서 보기 편하게 각 상품 리스트를 구분하게 만들었다. 서비스 레이아웃의 경우 각 편의점 레이아웃과 같은 색으로 구현하였으며 사용되는 각 서비스에 사용되는 이미지는 편의점 홈페이지에서 제공되는 이미지로 구현하였다. 상품 정보 레이아웃의 경우, 이미지는 홈페이지에서 파싱한 이미지의 크기를 다시 조정하여 보기 좋고, 적당한 용량으로 수정하여 지연시간 없이 보여주도록 개발하였다. 댓글이 입력되는 부분, 버튼의 모양 등은 직접 xml으로 코딩하여 제작하였다.

### 4.2.3 상품 검색 기능

본 연구에서 개발된 편월이는 행사 상품의 이름과 행사 종류별로 검색 가능하며, 음료, 과자, 생필품, 아이스크림, 즉석식품, 간편식품 등으로 구분하여 검색할 수 있는 기능을 제공한다. [Fig. 4.6]은 이러한 상품 검색 화면이다.



[Fig. 4.6] 상품 검색 화면

### 4.2.4 댓글 기능

사용자가 해당 제품의 평가를 통하여 다른 소비자가 해당 제품 구매에 도움을 주는 기능을 제공하기 위해 댓글 기능을 제공하며, [Fig. 4.7]과 같이 각 상품에 대한 댓글을 직접 적을 수 있도록 코드를 개발하였다. [Fig. 4.7]에서 사용자가 댓글을 입력하였을 때 즉시 DB에 그 내용이 저장되며, 다른 사용자들은 DB를 검색하여 입력된 댓글을 볼 수 있도록 구현하였다.

```

final FirebaseListAdapter<String> adapter =
    new FirebaseListAdapter<String>(this, String.class,
        android.R.layout.simple_list_item_1, messagesRef.child(name)) {
        @Override
        protected void populateView(View view, String s, int i) {
            TextView textView = (TextView) view.findViewById(android.R.id.text1);
            textView.setText(s);
        }
    };
mListView.setAdapter(adapter);
    
```

(a) 댓글 기능 코드



(b) 댓글 기능 화면

[Fig. 4.7] 댓글 기능 코드와 화면

## 5. 결론 및 향후과제

본 연구에서는 각 편의점 홈페이지에서 제공되는 각종 행사 정보와 다양한 서비스를 값싸고 효율적으로 이용하기 위한 편의점 정보 검색 시스템을 구현하였다.

본 앱을 이용하여 서로 다른 편의점을 비교하여 알뜰한 상품 구매가 가능하며, 편의점에서 제공하는 다양한 서비스를 편하게 이용할 수 있다. 또한, 소비자의 의견을 반영하여 상품 구매 여부를 결정하거나 평가할 수 있는 스마트 편의점 정보 앱을 개발하였다는 점에서 그 의미가 있다.

향후 연구 과제로는 지도 정보 검색을 좀 더 세밀하게 할 수 있는 기능과 상품 평가 항목을 구체화하는 부분을 개발할 예정이다. 또한 DB에 정규화 되어 있는 값을 사용하여 각 편의점 별 검색 뿐 아니라, 해당 상품에 대한 가격을 비교하여 더 저렴한 편의점을 보여주는 기능을 개발할 예정이다.

## REFERENCES

- [1] Jsoup을 이용한 HTML 파싱 개요, <http://blog.naver.com/PostView.nhn?blogId=occidere&logNo=220833730985>
- [2] jsoup을 활용해서 Java에서 HTML 파싱하는 방법 정리, <http://blog.acronym.co.kr/337>
- [3] java 문자열 치환, 바꾸기(replace), replaceLast() 사용법, <http://pandorica.tistory.com/5>
- [4] JSP Programming with a workbook, 이지선, 장병옥 공저, 이한 출판사, 2007.
- [5] 그림으로 쉽게 설명하는 안드로이드 프로그래밍, 천인국저, 생능출판사, 2012.

김 승 욱(Seung-Uk Kim)



- 2012년 2월 : 용문고등학교 졸업
- 2012년 3월 : 성공회대학교 정보통신공학과 입학
- 현재 : 성공회대학교 정보통신공학과 재학(학사)

<관심분야>

안드로이드 앱, 파싱 알고리즘, 프로그래밍

유 희 경(Hee-Gyeong You)



- 2012년 2월 : 세현고등학교 졸업
- 2012년 3월 : 성공회대학교 정보통신공학과 입학
- 현재 : 성공회대학교 정보통신공학과 재학(학사)

<관심분야>

안드로이드 프로그래밍, 데이터베이스, 정보검색

정 인 철(In-Cheol Jeong)



- 2000년 3월 : Keio Univ. 전기공학 박사
- 2002년 2월 : 한국전자통신연구원 선임연구원
- 2002년 3월 ~ 현재 : 성공회대학교 정보통신공학과 교수

<관심분야>

근거리통신, 무선통신, 사물인터넷

김 태 우(Tai-Woo Kim)

[중신회원]



- 1996년 2월 : 고려대학교 컴퓨터학과(박사)
- 1984년 1월 ~ 1986년 10월 : (주) LG전자 컴퓨터사업부 시스템 엔지니어
- 1986년 10월 ~ 1997년 2월 : 한국전자통신연구소(ETRI) 슈퍼컴퓨터센터 선임연구원

- 1997년 3월 ~ 현재 : 성공회대학교 정보통신공학과 교수

<관심분야>

사물인터넷, 정보보호, 컴퓨터 네트워크, 분산처리