



# 방구석 마켓

“

스마트한 쇼핑 with 방구석 마켓

”

# Index

01	02	03	04	05	06
●	●	●	●	●	●
팀원 소개 & 역할 분담	개발 배경 & Flow Chart	전체 페이지 구성	상세 기능 설명  1. Andoroid 2. Data Base	기대 효과	차후 개발 계획

# 01 Team 잘먹사

김정희



성신여대 정보시스템공학과 전공  
junghee0592@gmail.com  
Part : Android , API

정연희



성신여대 정보시스템공학과 전공  
dusgml5328@gmail.com  
Part : Android , API

임서연



성신여대 정보시스템공학과 전공  
limsy0310@naver.com  
Part : Android , API

김혜진



성신여대 정보시스템공학과 전공  
20181019@shungshin.ac.kr  
Part : Android , DB

김윤진



성신여대 정보시스템공학과 전공  
bestyjin0604@gmail.com  
Part : Android , DB

## 02 개발배경

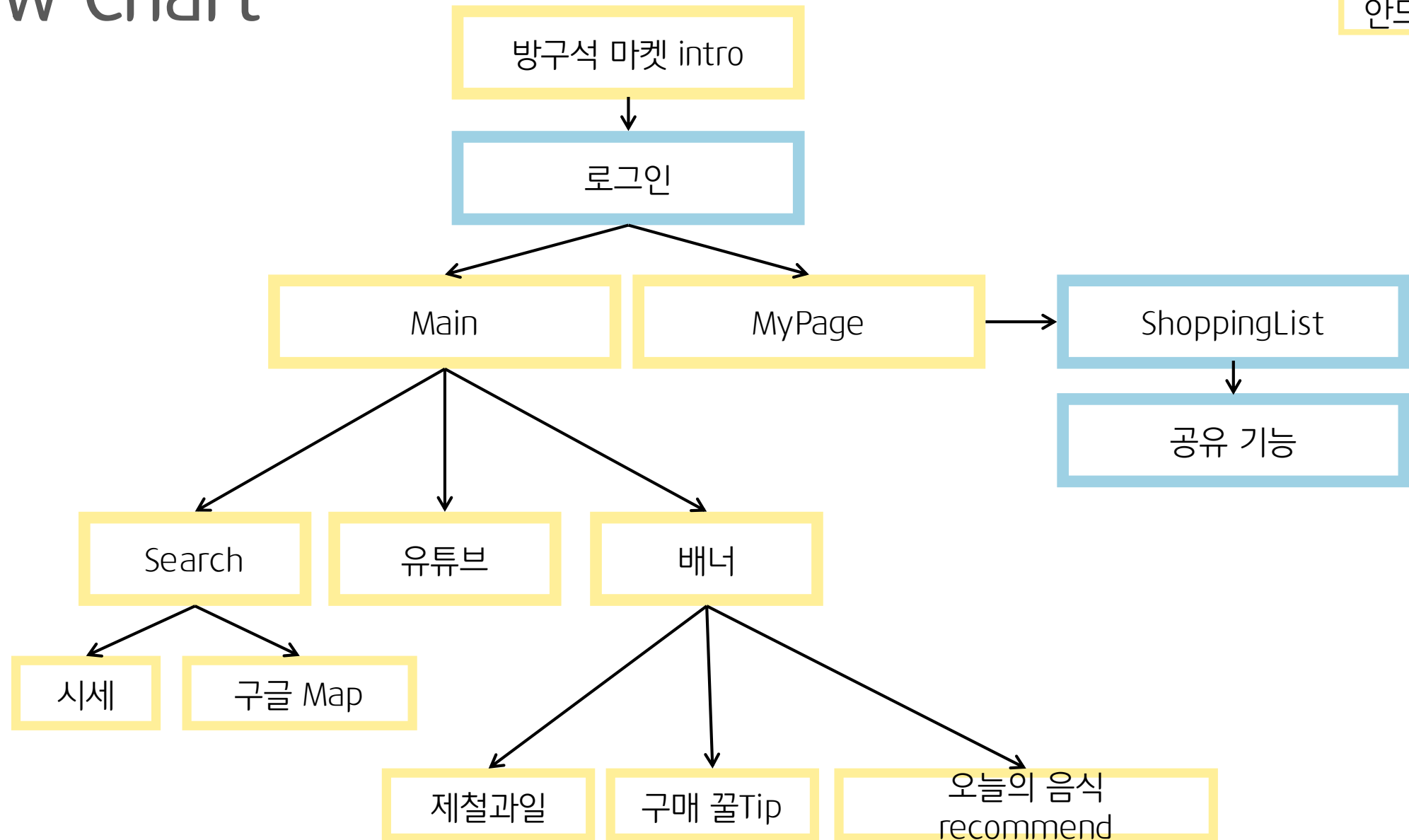
매일 달라지는 식재료의 시세를 쉽게 조회하고 싶은 바람

마트별로 다른 가격을 한눈에 확인할 수 없다는 불편함

전통시장의 식재료 가격도 쉽게 집에서 확인하고 싶다는 바람

주 타겟층 : 1인 가구의 자취생, 2인 가구 이상의 가정에서 식재료 구매를 담당하는 주부

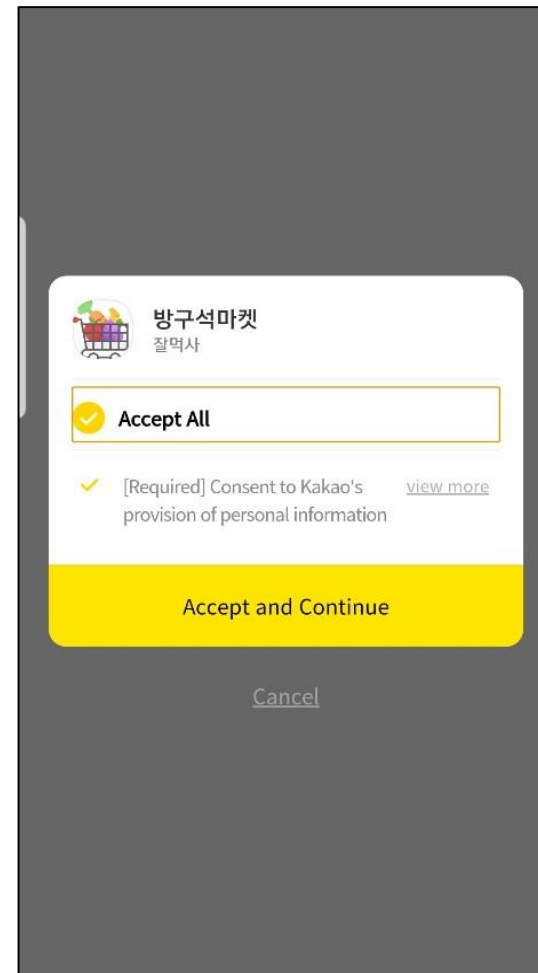
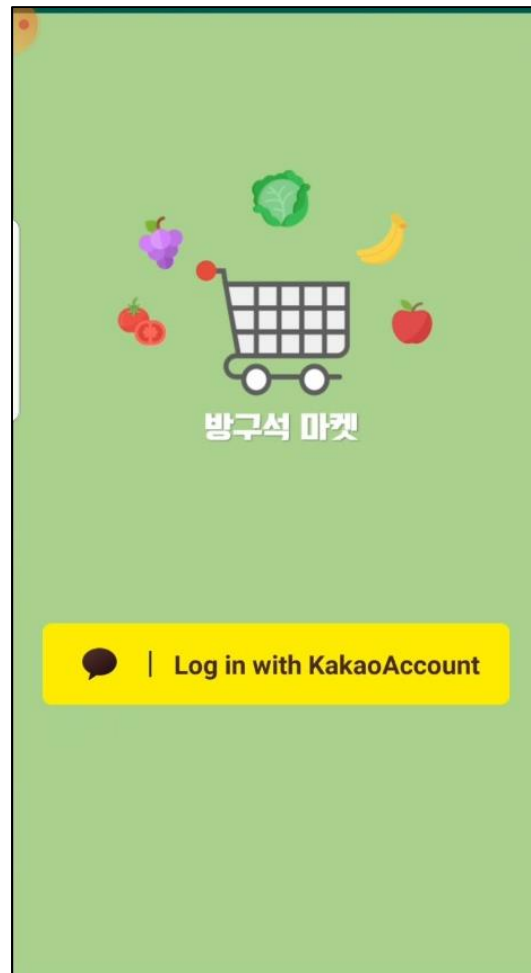
# Flow Chart



## 03 전체 구성 페이지



LoadingActivity.java



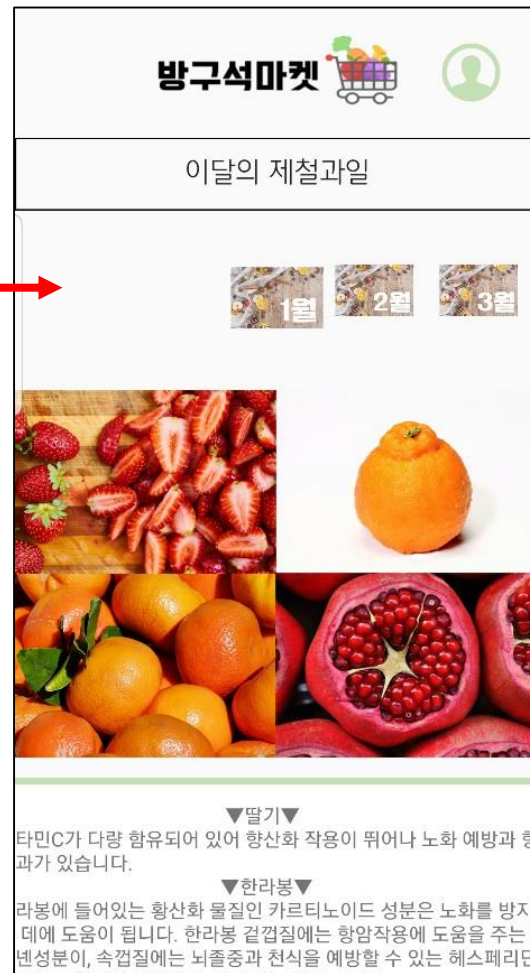
LoginPage.java



MainActivity.java



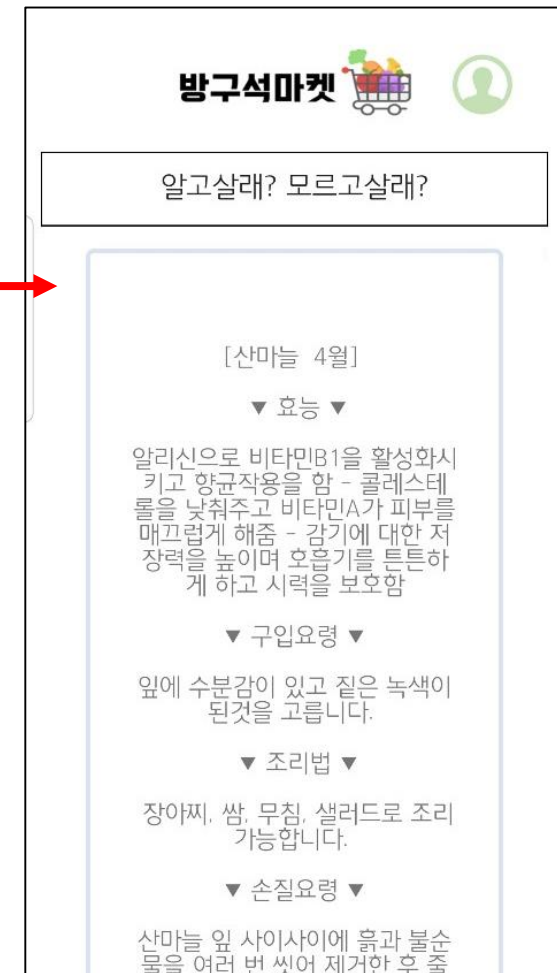
MainActivity.java



Jachal.java



MainActivity.java



Knowhow.java

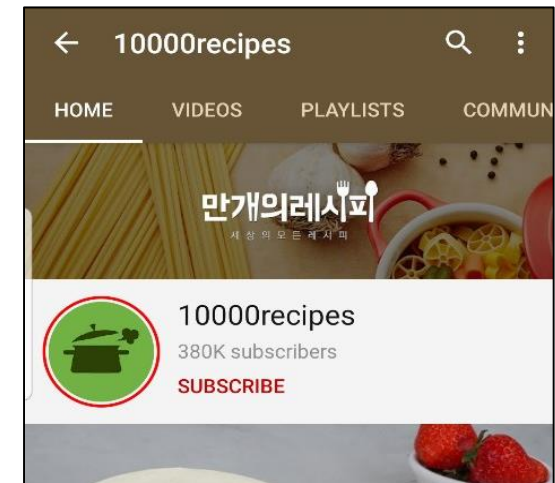
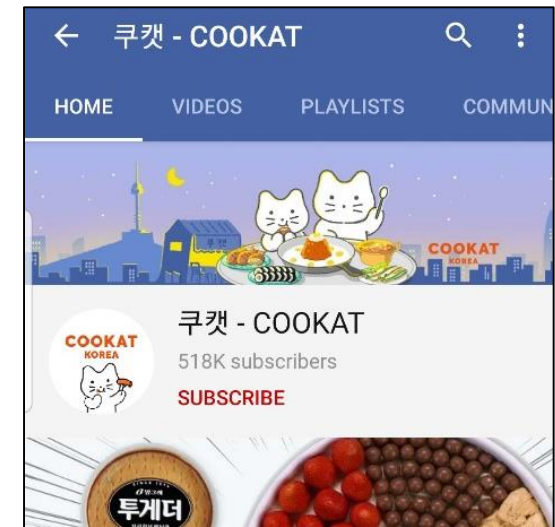




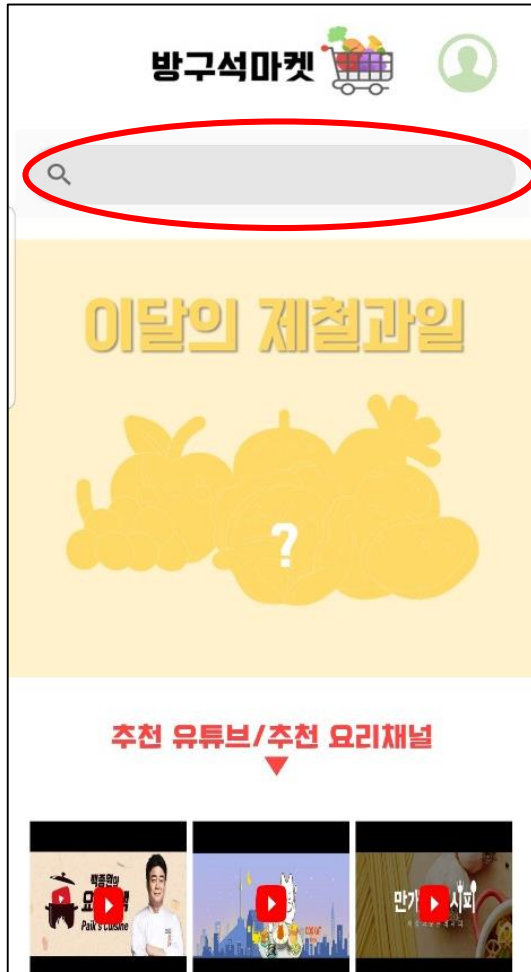
FoodRecomend.java



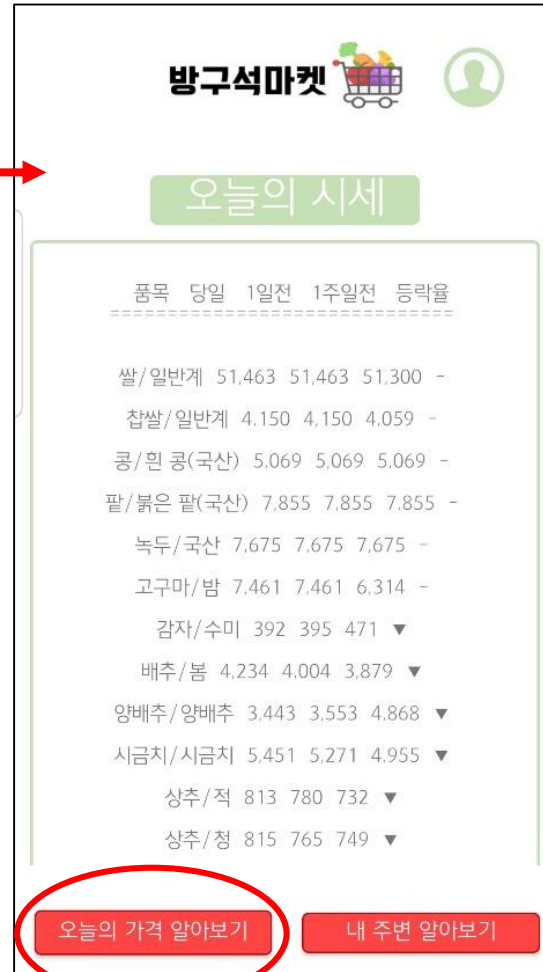
MainActivity.java







MainActivity.java

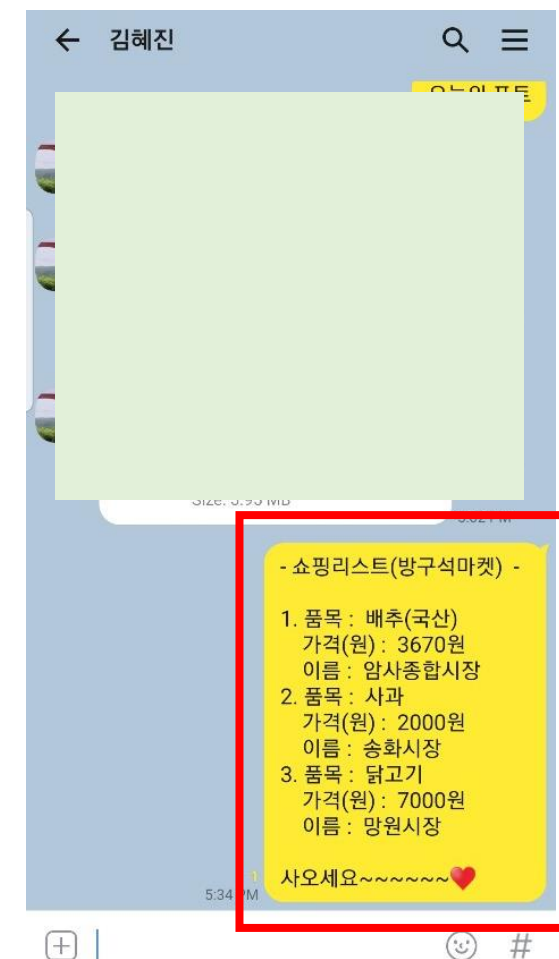
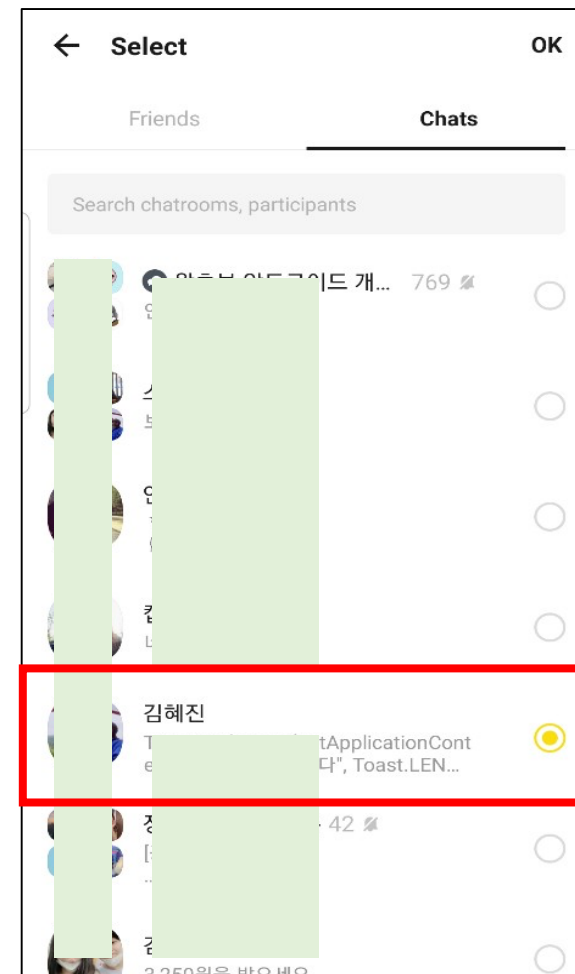
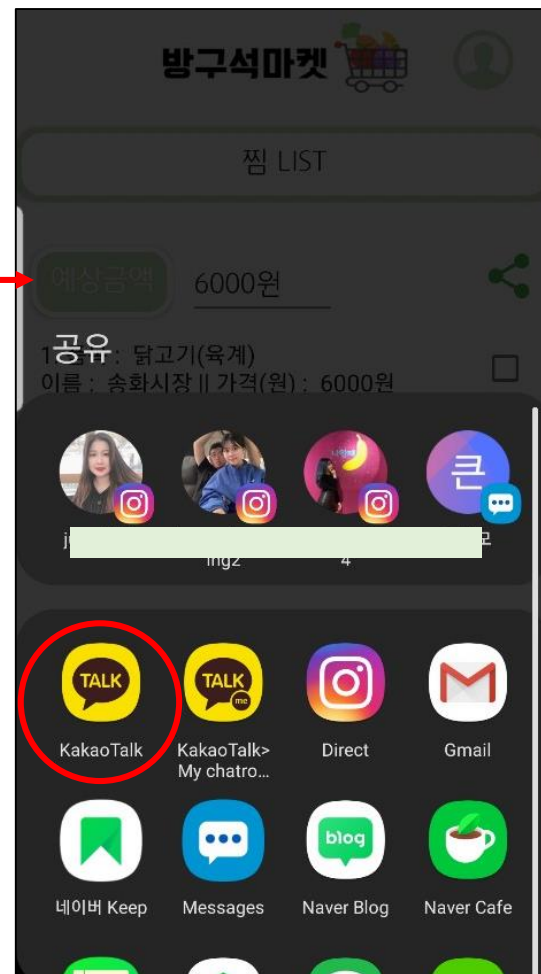


SearchAfterPage.java



UserChoiceSearch.java

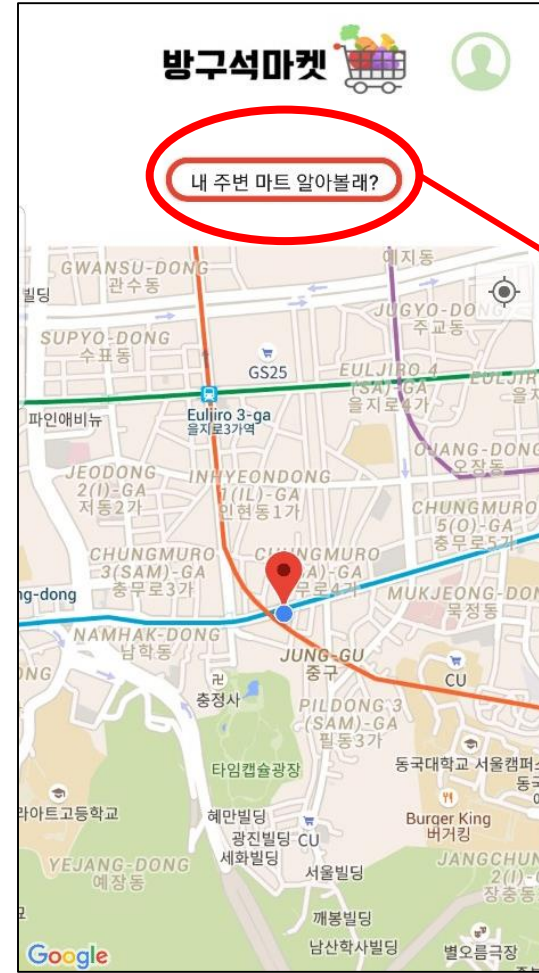
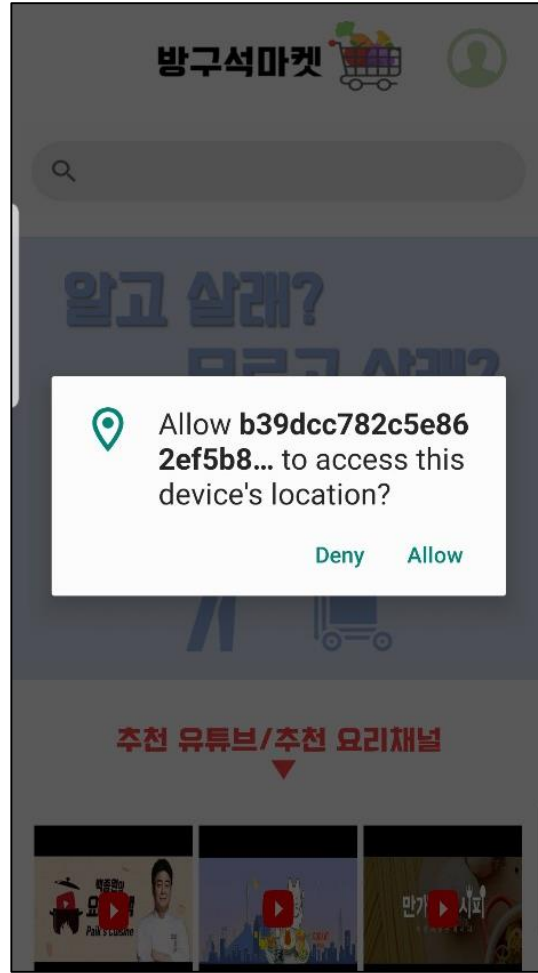




ShoppingList.java



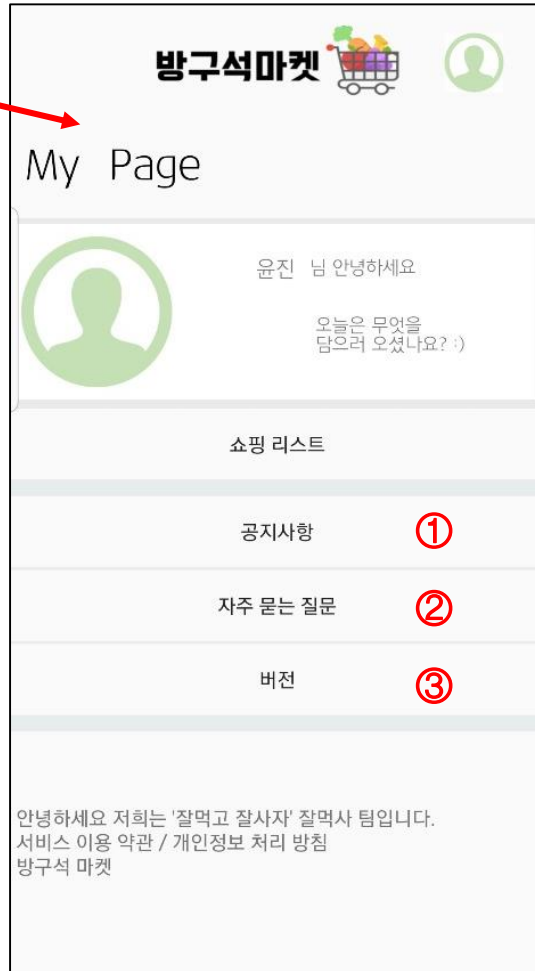
SearchAfterPage.java



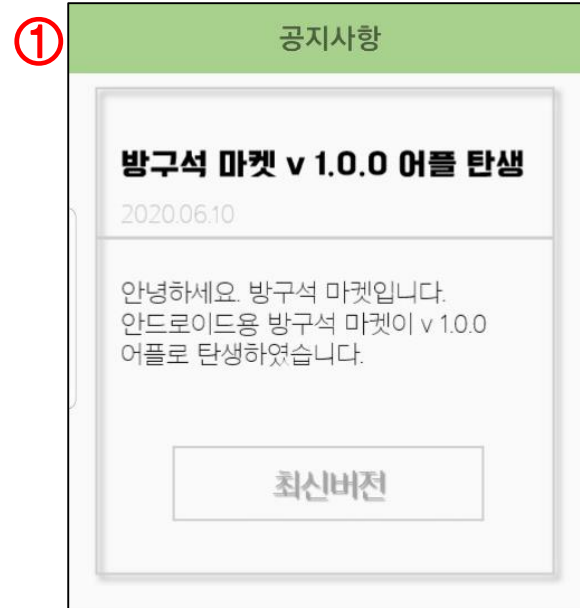
MapsActivity.java



MainActivity.java



MyPage.java



Notice.java



Question.java



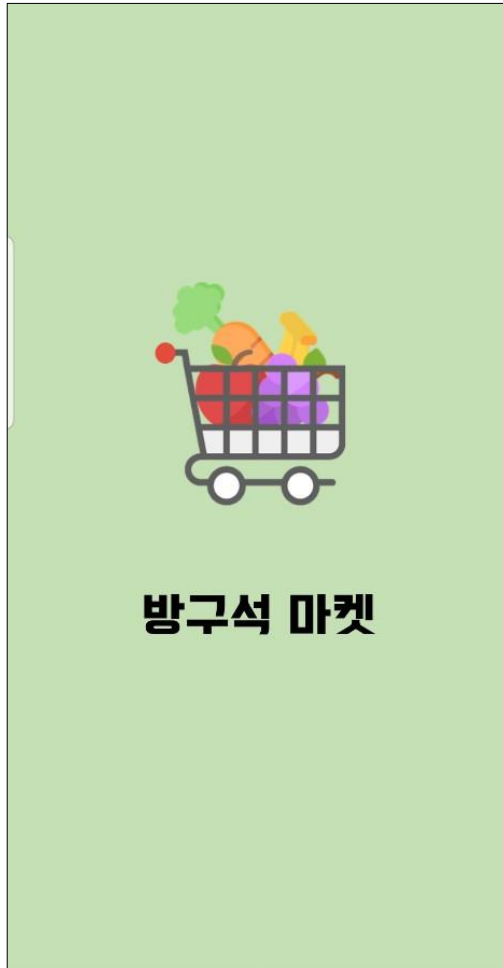
Version.java

## 04 상세 기능 설명



# Loading

- 로딩 후 자동으로 다음 페이지로 넘어감

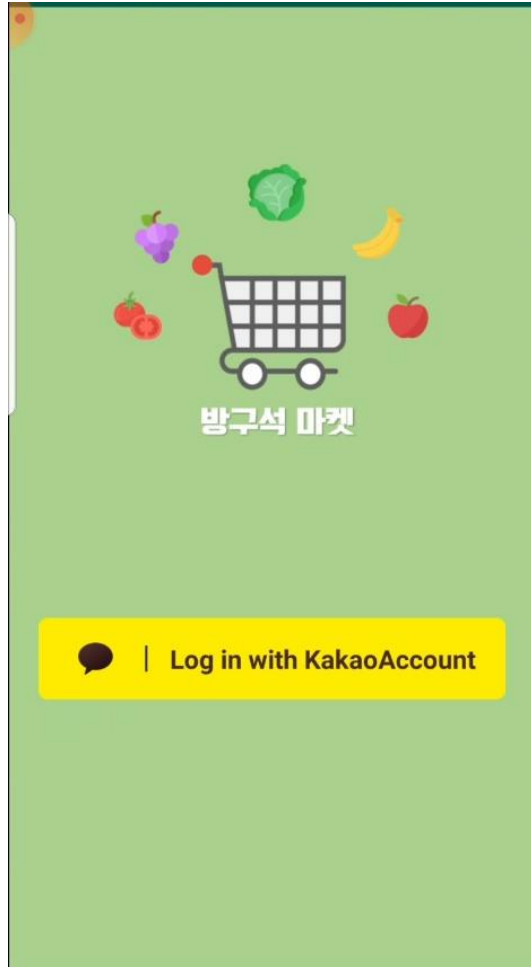


## 핵심 1. MANIFEST.XML을 다음과 같이 수정

```
<activity  
    android:name=".LoadingActivity"  
    android:label="@string/app_name"  
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
    </activity>
```



# LoginPage

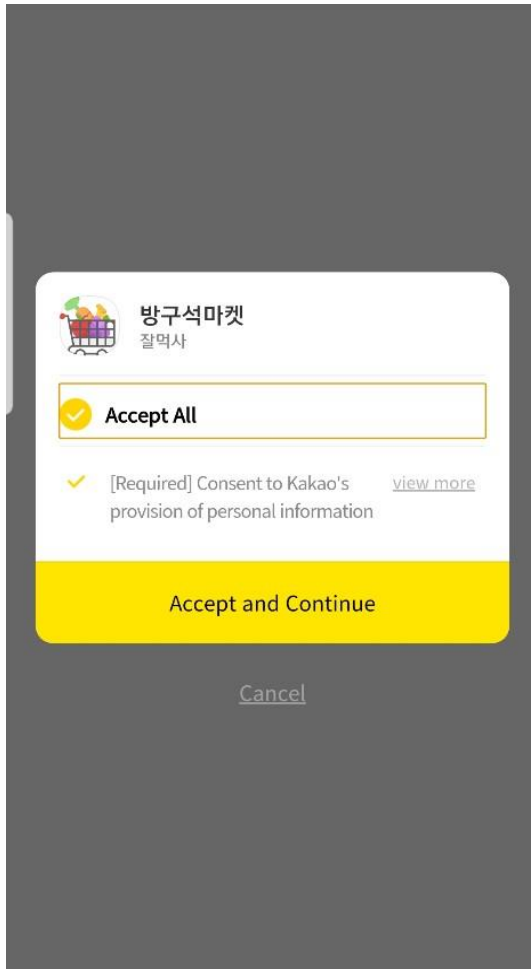


**핵심 1.** 로그인 버튼을 클릭 했을시 access token을 요청하도록 설정한다.@param savedInstanceState 기존 session 정보가 저장된 객체

```
KakaoSDK.init(new GlobalApplication.KakaoSDKAdapter());
callback = new SessionCallback();
com.kakao.auth.Session.getCurrentSession().addCallback(callback);
com.kakao.auth.Session.getCurrentSession().checkAndImplicitOpen();

com.kakao.auth.Session.getCurrentSession().open(AuthType.KAKAO_TALK_EXCLUDE_NATIVE
_LOGIN, LoginPage.this);
try {
    PackageInfo info =
        getPackageManager().getPackageInfo("your.package.name",
            PackageManager.GET_SIGNATURES);
    for (Signature signature : info.signatures) {
        MessageDigest md = MessageDigest.getInstance("SHA");
        md.update(signature.toByteArray());
        Log.e("KeyHash:", Base64.encodeToString(md.digest(), Base64.DEFAULT));
    }
} catch (PackageManager.NameNotFoundException e) {
    e.printStackTrace();
} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
}
}
```

# LoginPage



```
protected static class KakaoSDKAdapter extends KakaoAdapter
{
    @Override
    public ISessionConfig getSessionConfig() {
        return new ISessionConfig()
        {
            @Override
            public AuthType[] getAuthTypes()
            {
                return new AuthType[]{AuthType.KAKAO_LOGIN_ALL};
            }

            @Override
            public boolean isUsingWebviewTimer() {return false ;}

            @Override
            public boolean isSecureMode() {return false;}

            @Override
            public ApprovalType getApprovalType() {return ApprovalType.INDIVIDUAL; }

            @Override
            public boolean isSaveFormData() {return true;}
        };
    }

    @Override
    public IApplicationConfig getApplicationConfig() {return new IApplicationConfig()
    {
        @Override
        public Context getApplicationContext()
        {
            return GlobalApplication.getGlobalApplicationContext();
        }
    }
};
}
```

## 핵심 2.

Session Config에 대해서는 default값들이 존재한다.  
필요한 상황에서만 override해서 사용하면 됨  
@return Session의 설정값.

# Main - 스와이프시 다른 이미지로 넘어가는 배너 구현 및 배너 클릭시 해당 페이지 이동



## 핵심 1. PageViewAdapter&slider

```
public Object instantiateItem(ViewGroup container, final int position) {
    //포지션에 해당하는 페이지 생성
    inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View v = inflater.inflate(R.layout.slider, container, false);
    ImageView imageView = (ImageView) v.findViewById(R.id.imageview);
    imageView.setImageResource(images[position]);
```

```
    imageView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { //포지션에 해당하는 페이지 뷰 클릭시 페이지 이동
            if(position==0){
                Intent readMore = new Intent(v.getContext(), Jachal.class);
                v.getContext().startActivity(readMore);}
            if(position==1){
                Intent readMore = new Intent(v.getContext(), Knowhow.class);
                v.getContext().startActivity(readMore);}
            if(position==2){
                Intent readMore = new Intent(v.getContext(), FoodRecomend.class);
                v.getContext().startActivity(readMore);
            }
        }
    });
```

```
    container.addView(v); //컨테이너(뷰 페이지) 객체에 생성된 페이지 뷰 추가
    return v; //페이지를 식별할 수 있는 키 객체 리턴
}
```

# Main

- Android Youtube Intent로 실행하기(유튜브 영상 링크 열기)



## 핵심 2. 지정된 유튜브 링크를 앱으로 넘겨서 영상 확인

```
String urlyoutube = "https://www.youtube.com/channel/UCyn-K7rZLXjGI7VXGwellcA";  
String urlcookcat = "https://www.youtube.com/channel/UCjrwkxV0pN133r1Yp07583Q";  
String urlrecipe = "https://www.youtube.com/channel/UCKA_6r3CWC76x_EaF06jsPA";
```

```
youtubebtn = (ImageButton) findViewById(R.id.youtubebtn);  
youtubebtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(urlyoutube));  
        startActivity(intent.setPackage("com.google.android.youtube"));  
    }  
});
```

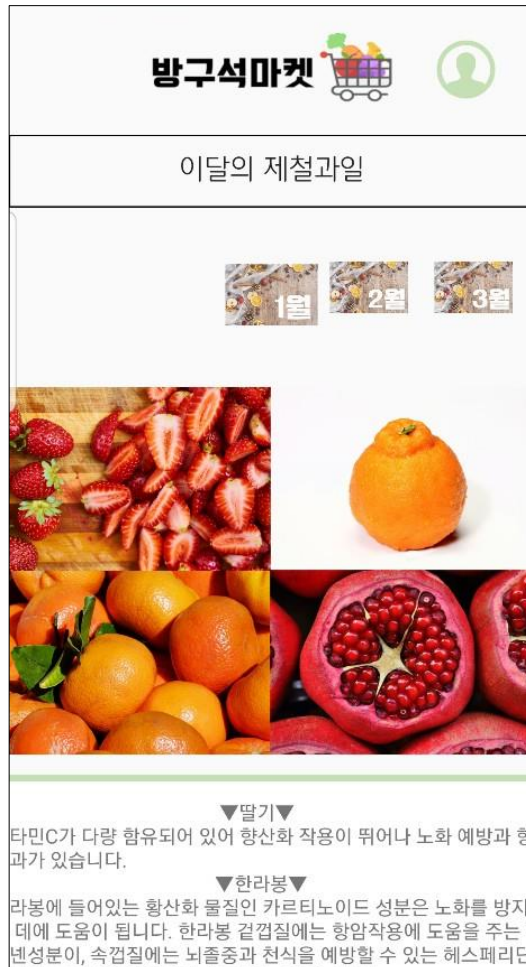
← 웹페이지 띄우기

```
cookcatbtn = (ImageButton) findViewById(R.id.cookcatbtn);  
cookcatbtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(urlcookcat));  
        startActivity(intent.setPackage("com.google.android.youtube"));  
    }  
});
```

// 어플리케이션 강제 설정  
← 앱이 설치 되어있으면 Intent로  
앱으로 자동 연결

```
recipebtn = (ImageButton) findViewById(R.id.recipebtn);  
recipebtn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(urlrecipe));  
        startActivity(intent.setPackage("com.google.android.youtube"));  
    }  
});
```

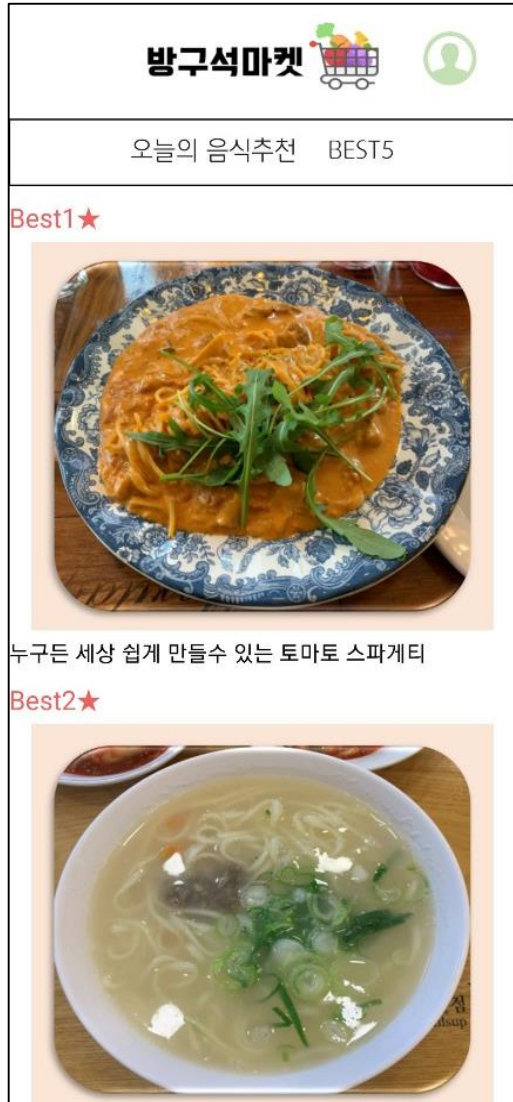
# Jachal - 각 월에 해당되는 제철과일 소개



## 핵심 1. Gallery사용을 위해 Adapter설정

```
public class MyGalleryAdapter extends BaseAdapter {
    Context context;
    Integer[] posterID={R.drawable.janu, R.drawable.febu, R.drawable.march, R.drawable.april, R.drawable.mayy, R.drawable.june, R.drawable.july,
    R.drawable.august, R.drawable.eqptember, R.drawable.october, R.drawable.november, R.drawable.december};
    Integer[] posterI = { R.drawable.jan, R.drawable.feb,
    R.drawable.mar, R.drawable.apr, R.drawable.may, R.drawable.jun, R.drawable.jul, R.drawable.aug, R.drawable.sep, R.drawable.oct, R.drawable.nov,
    R.drawable.dec};
    public MyGalleryAdapter(Context c) {
        context = c;}
    public int getCount() {
        return posterID.length;}
    public Object getItem(int arg0) {
        return null;}
    public long getItemId(int position) {
        return 0;}
    public View getView(int position, View convertView, ViewGroup parent) {
        ImageView imageview = new ImageView(context);
        imageview.setLayoutParams(new Gallery.LayoutParams(200, 300));
        imageview.setScaleType(ImageView.ScaleType.FIT_CENTER);
        imageview.setPadding(5, 5, 5, 5);
        imageview.setImageResource(posterID[position]);
        final int pos = position;
        imageview.setOnTouchListener(new View.OnTouchListener() {
            public boolean onTouch(View v, MotionEvent event) {
                ImageView ivPoster = (ImageView) findViewById(R.id.ivPoster);
                ivPoster.setScaleType(ImageView.ScaleType.FIT_CENTER);
                ivPoster.setImageResource(posterI[pos]);
                return false;
            }
        });
        return imageview;
    }
}
```

# Food Recomend- 앱 접속 시 매번 다른 음식추천으로 랜덤 발생 및 중복 방지 코드



## 핵심 1. 랜덤 발생 & while문과 중복처리하기 위한 for문 및 if문

```
Random ram = new Random(); //랜덤 발생
```

```
int num, count; int i=0;
```

```
while(i!=5){
```

```
    if(i==5)
```

```
        break;
```

```
    num = ram.nextInt(img.length); //랜덤 숫자 발생 0~10 5번
```

```
    count = 0;
```

```
    for(int j=0; j<=i; j++){
```

```
        //랜덤발생했던수와 현재 발생한 수 비교
```

```
        if(num!= numlist[j])
```

```
            count=count+1;
```

```
        else
```

```
            count = 0;
```

```
    }
```

```
    if(count-1 == i){
```

```
        numlist[i] = num; //이미지넘버
```

```
        numstr[i] = num; //멘트배열
```

```
        i++;
```

```
    }
```

```
}
```



```
best1.setImageResource(img[numlist[0]]);
```

```
text1.setText(str[numstr[0]]);
```

//Best 5위로 텍스트뷰, 이미지뷰 각각 5개씩  
setImageResource, setText로 구현했습니다.

```
// numlist[], numstr[]배열 선언 및 초기화
```

```
int numlist[] = new int[15]; //랜덤배열
```

```
int numstr[] = new int[15]; //랜덤배열
```

```
for(int i = 0 ; i<numlist.length;i++) {
```

```
    numlist[i]=9999; //초기화
```

```
    numstr[i] = 9999; //초기화 }
```



# Knowhow

- 농림축산식품 공공데이터 포털 API (제철 농산물 상세정보)

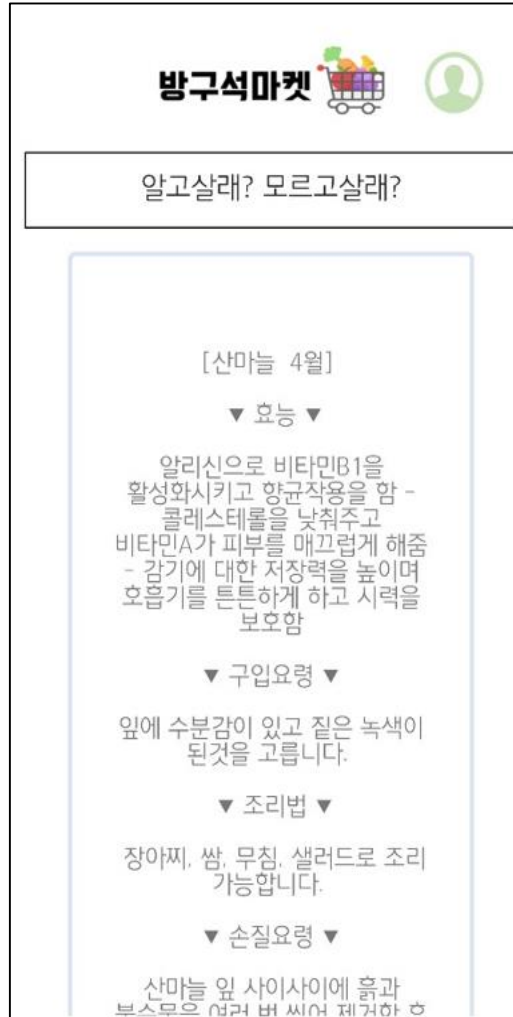
## 핵심 1. String Buffer에 XML 파싱 데이터 저장

**\*전반적인 방법은 Kamis Api 이용 방법과 동일\***

while (eventType != XmlPullParser.END\_DOCUMENT) {  
switch (eventType) {  
case XmlPullParser.START\_DOCUMENT:  
buffer.append("파싱 시작...WnWn");  
break;  
  
case XmlPullParser.START\_TAG:  
tag = xpp.getName();//태그 이름 얻어오기  
  
if (tag.equals("row")){// 첫번째 검색결과  
else if (tag.equals("PRDLST\_NM")) {  
//buffer.append("품목 : ");  
buffer.append("[");  
xpp.next();  
buffer.append(xpp.getText());//title 요소의 TEXT 읽어와서 문자열버퍼에 추가  
buffer.append(" "); //줄바꿈 문자 추가  
}.종략..  
break;  
  
case XmlPullParser.TEXT:  
break;  
  
case XmlPullParser.END\_TAG:  
tag = xpp.getName();//태그 이름 얻어오기  
  
if (tag.equals("row")) buffer.append("WnWn");// 첫번째 검색결과종료..줄바꿈  
break;  
}  
eventType = xpp.next();  
}  
}  
catch (Exception e) {  
e.printStackTrace();  
}  
}  
//buffer.append("파싱 끝Wn");  
return buffer.toString();//StringBuffer 문자열 객체 반환

//While문을 이용해 문서의 끝이 나타나기 전까지 태그의 이름과 같으면 파싱해서 String Buffer에 저장

```
<?xml version="1.0" encoding="UTF-8"?>
<Grid_Grid_20171128000000000572_1>
  <totalCnt>183</totalCnt>
  <startRow>1</startRow>
  <endRow>184</endRow>
  <result>
    <message>정상 처리되었습니다.</message>
    <code>INFO-000</code>
  </result>
  <row>
    <ROW_NUM>1</ROW_NUM>
    <IDNTFC_NO>227366</IDNTFC_NO>
    <PRDLST_NM>산마늘</PRDLST_NM>
    <M_DISTCTNS>4월</M_DISTCTNS>
    <M_DISTCTNS_ITM>01,02,03</M_DISTCTNS_ITM>
    <PRDLST_CL>
      <![CDATA[
        농산물 > 채소류
      ]]>
    </PRDLST_CL>
    <MTC_NM>광복 울릉, 영양 / 강원 평창, 인제, 홍천, 강릉 / 경남 하동</MTC_NM>
    <PRDCTN__ERA>3~5월</PRDCTN__ERA>
  </row>
</Grid_Grid_20171128000000000572_1>
```



//buffer.append("파싱 끝Wn");  
return buffer.toString();//StringBuffer 문자열 객체 반환

# SearchAfterPage - Kamis API (최근일자 지역별 도.소매가격정보(상품 기준))



품목	당일	1일전	1주일전	동락율
쌀/일반계	51,463	51,463	51,300	-
잡쌀/일반계	4,150	4,129	4,219	▼
콩/흰 콩(국산)	5,069	5,069	5,069	-
팥/붉은 팥(국산)	7,855	7,855	7,855	-
녹두/국산	7,675	7,675	7,675	-
고구마/밤	7,461	7,461	6,255	-
감자/수미	395	402	463	▼
배추/봄	4,004	4,044	null	▼
양배추/양배추	3,553	3,497	5,805	▼
시금치/시금치	5,271	5,430	4,997	▼
상추/적	780	810	724	▼
상추/청	765	805	719	▼

오늘의 가격 알아보기    내 주변 알아보기

**핵심 1.** Manifest 파일에 네트워크 사용 권한 허용  
+ http 프로토콜 접속 시 예외발생 조치

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<application android:usesCleartextTraffic="true">
```

**핵심 2.** String getXmlData()

- XmlPullParser를 이용해 Kamis 에서 제공하는 API안의 품목 당일 시세를 보여줌

```
URL url = new URL(queryUrl); // 문자열로 된 요청 url을 URL 객체로 생성.
InputStream is = url.openStream(); // url위치로 입력스트림 연결
```

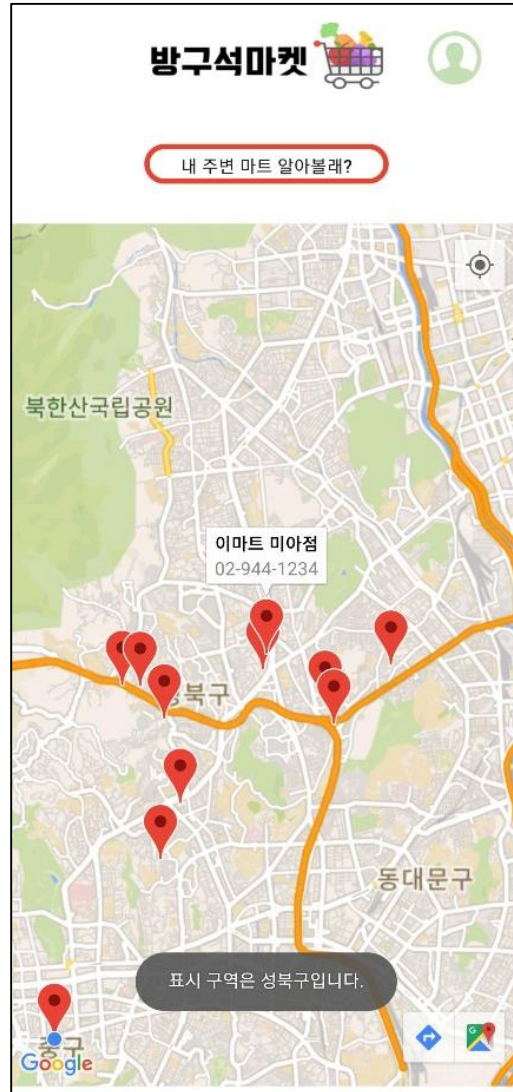
```
XmlPullParserFactory factory = XmlPullParserFactory.newInstance(); // xml파싱을 위한
XmlPullParser xpp = factory.newPullParser();
xpp.setInput(new InputStreamReader(is, "UTF-8")); // inputstream 으로부터 xml 입력받기
```

\* Android 4.0 이상 부터는 네트워크를 이용할 때 반드시 Thread 사용해야 함

```
new Thread(new Runnable() {
    @Override
    public void run() {
        data = getXmlData(); // 아래 메소드를 호출하여 XML data를 파싱해서 String 객체로 얻어오기

        runOnUiThread(new Runnable() { // UI Thread(Main Thread)를 제외한 어떤 Thread도 화면을 변경할 수 없기 때문에
            @Override
            public void run() { // runOnUiThread()를 이용하여 UI Thread가 TextView 글씨 변경하도록 함
                text.setText(data); // 받아온 xml 적기
            }
        });
    }
}).start();
```

# Google Map - 현재위치를 받아오고 버튼 클릭 시 성북구 근처 마트, 시장 표시



## 핵심 1. Manifest 파일에 권한 허용 + Key 추가

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

## 핵심 2. MapsActivity - setCurrentLocation()

```
public void setCurrentLocation(Location location, String markerTitle, String
markerSnippet) {
```

```
    if (currentMarker != null) currentMarker.remove();
```

```
    // 위도, 경도 얻어오기
```

```
    LatLng currentLatLng = new LatLng(location.getLatitude(),
```

```
    location.getLongitude());
```

```
    //마커찍기
```

```
    MarkerOptions markerOptions = new MarkerOptions();
```

```
    markerOptions.position(currentLatLng);
```

```
    markerOptions.title(markerTitle);
```

```
    markerOptions.snippet(markerSnippet);
```

```
    markerOptions.draggable(true);
```

```
    //마커 추가
```

```
    currentMarker = mMap.addMarker(markerOptions);
```

```
    //카메라 업데이트 및 이동
```

```
    CameraUpdate cameraUpdate =
```

```
    CameraUpdateFactory.newLatLng(currentLatLng);
```

```
    mMap.moveCamera(cameraUpdate); }
```

# UserChoiceSearch

농림축산식품 공공데이터 포털 API (제철 농산물 상세정보)

방구석마켓

▼ 알아보고 싶은 야채를 Click 해주세요 ▼

사과(부사)

배추(국산)

쇠고기(한우불고기)

닭고기

▼ 사는 동네를 Click 해주세요 ▼

강남구

강동구

강북구

강서구

▼ 마트 VS 시장 당신의 선택은 ▼

대형마트

전통시장

▼ 방문할 곳은 선택해주세요 ▼

이마트

홈플러스

Click →

검색하기

▼ 검색 결과를 확인해주세요 ▼

이름 : 홈플러스 강동점

품목 : 사과(부사),중급(대)

가격(원) : 2500원

방구석마켓

▼ 알아보고 싶은 야채를 Click 해주세요 ▼

사과(부사)

배추(국산)

쇠고기(한우불고기)

닭고기

▼ 사는 동네를 Click 해주세요 ▼

강남구

강동구

강북구

강서구

▼ 마트 VS 시장 당신의 선택은 ▼

대형마트

전통시장

▼ 방문할 곳은 선택해주세요 ▼

이마트

홈플러스

Click →

검색하기

품목 : 사과(부사),중급(대)

가격(원) : 2500원

찜 넣기 Click →

찜 페이지 GO

## 핵심 1. XML 파싱을 위한 기본 URL 생성 방법

샘플 URL [http://openAPI.seoul.go.kr:8088/\(인증키\)/xml/ListNecessariesPricesService/1/5/](http://openAPI.seoul.go.kr:8088/(인증키)/xml/ListNecessariesPricesService/1/5/)

### 요청인자

변수명	타입	변수설명	값설명
KEY	String(필수)	인증키	OpenAPI에서 발급된 인증키
TYPE	String(필수)	요청파일타입	xml:xml,xml파일:xml,엑셀파일:xls,json파일:json
SERVICE	String(필수)	서비스명	ListNecessariesPricesService
START_INDEX	INTEGER(필수)	요청시작위치	정수 입력 (페이징 시작번호입니다:데이터행 시작번호)
END_INDEX	INTEGER(필수)	요청종료위치	정수 입력 (페이징 끝번호입니다:데이터행 끝번호)
M_NAME_SCH	STRING(선택)	시장/마트 이름	시장/마트 이름 문자열
A_NAME	STRING(선택)	품목 이름	품목 이름
P_YEAR_MONTH	STRING(선택)	년도-월	YYYY-MM
M_TYPE_NAME	STRING(선택)	시장유형 구분 이름	시장유형
M_GU_NAME	STRING(선택)	자치구 이름	자치구 이름

요청 인자를 보고 KEY에는 미리 신청하고 발급받은 요청키를 넣고 이어 **순서에 따라 '/'**로 나누고, 나눈 곳에 원하는 변수 값을 넣어 URL을 완성.

## 핵심 2. 선택에 따른 정보를 보여주기 위한 URL 생성

```
buttonmart[0].setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(),
            buttonmart[0].getText(),
            Toast.LENGTH_SHORT).show();
        martNameApi = GN_M[0]; //URL에 들어가는 내용을 전역 변수에 저장
    }
});
```

String queryUrl="http://openapi.seoul.go.kr:8088/637076724e616e6439307949697164/xml/ListNecessariesPricesService/1/1000/" + martNameApi + "/" + imgApi + "/" + year + "-" + month + "/" + martTypeApi + "/" + guApi;

//미리 저장해놓은 변수를 이용해 URL 완성



# UserChoiceSearch - 앞서 설명한서울 공공데이터 API사용

## 핵심 1. INVISIBLE로 유저가 눌렀을 때 VISIBLE로 변환(모든버튼)



//강서구 < 예시 >

```
buttongu[3].setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), gu[3], Toast.LENGTH_SHORT).show();
        //클릭한거 Toast 띄우기
        guApi = gu[3];
        reset(); //이 함수내에는 다시 한번 필요버튼들은 INVISIBLE로 해주는 코드가 있다.
        choosemart.setVisibility(choosemart.VISIBLE);
        bigmart.setVisibility(bigmart.VISIBLE);
        smallmart.setVisibility(smallmart.VISIBLE);
```

```
bigmart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        martTypeApi = "대형마트"; //DB에게 넘겨줄 String 변수의 저장
        Toast.makeText(getApplicationContext(), bigmart.getText(), Toast.LENGTH_SHORT).show();
        buttonmart[0].setVisibility(buttonmart[0].VISIBLE);
        buttonmart[1].setVisibility(buttonmart[1].VISIBLE);
```

```
buttonmart[0].setText(GS_M[0]);
buttonmart[1].setText(GS_M[1]);
```

```
buttonmart[0].setOnClickListener(new View.OnClickListener() { //왼쪽 사진으로 예시를 들면 이마트 클릭시
    @Override
    public void onClick(View v) { //토스트 및 api와 DB를 위해 string에 저장한다.
        Toast.makeText(getApplicationContext(), buttonmart[0].getText(), Toast.LENGTH_SHORT).show();
        martNameApi = GS_M[0];
    }
});
```

```
buttonmart[1].setOnClickListener(new View.OnClickListener() { //buttonmart[0]과 동일
    @Override
    public void onClick(View v) {
        Toast.makeText(getApplicationContext(), buttonmart[1].getText(), Toast.LENGTH_SHORT).show();
```

```
martNameApi = GS_M[1]; } } } } };
```

모든 Button의 기본설정은  
INVISIBLE이기에  
클릭했을때 VISIBLE로  
설정하는것이 중요

# Shopping List – 원하는 품목, 마트, 가격 확인 및 쇼핑리스트에 담기



## 핵심 1. DBHelper.java 에서 DB 테이블 및 칼럼 작성

//찜리스트 DB

```
public static final String DATABASE_NAME = "Zimlist.db";
public static final String ZIMLIST_TABLE_NAME = "zimlist";
public static final String ZIMLIST_COLUMN_ID = "id";
public static final String ZIMLIST_COLUMN_DB_MARTNAME = "db_martname";
public static final String ZIMLIST_COLUMN_DB_ITEM = "db_item";
public static final String ZIMLIST_COLUMN_DB_PRICE = "db_price";
```

@Override

```
public void onCreate(SQLiteDatabase db) {
    db.execSQL(
        "create table zimlist " +
        "(id integer primary key,db_martname string, db_item string, db_price string)"
    );
}
```

```
public boolean insertZimlist(String db_martname, String db_item, String db_price)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();

    contentValues.put("db_martname", db_martname);
    contentValues.put("db_item", db_item);
    contentValues.put("db_price", db_price);

    db.insert("zimlist", null, contentValues);
    return true;
}
```



# Shopping List – 원하는 품목, 마트, 가격 확인 및 쇼핑리스트에 담기 (이어서)



## 핵심2. 찜 넣기 버튼 클릭 시 DB에 추가하기

```
//찜페이지 토스트
```

```
zimclickbtn = (ImageButton) findViewById(R.id.zimclickbtn);
zimclickbtn.setOnClickListener(new View.OnClickListener() {
```

```
@Override
```

```
public void onClick(View v) {
    Bundle extras = getIntent().getExtras();
```

```
    if (mydb.insertZimlist(DB_martname.getText().toString(),
        DB_item.getText().toString(), DB_price.getText().toString())) {
        Toast.makeText(getApplicationContext(), "찜에 담겼습니다",
            Toast.LENGTH_SHORT).show();
```

```
    }
    else {
        Toast.makeText(getApplicationContext(), "추가되지 않았음",
            Toast.LENGTH_SHORT).show();
```

```
    }
}
});
```

```
Bundle extras = getIntent().getExtras();
```

```
if (extras != null) {
```

```
    int Value = extras.getInt("id");
```

```
    if (Value > 0) {
```

```
        Cursor rs = mydb.getData(Value);
```

```
        id = Value;
```

```
        rs.moveToFirst();
```

```
        String n = rs.getString(rs.getColumnIndex(DBHelper.ZIMLIST_COLUMN_DB_MARTNAME));
```

```
        String i = rs.getString(rs.getColumnIndex(DBHelper.ZIMLIST_COLUMN_DB_ITEM));
```

```
        String p = rs.getString(rs.getColumnIndex(DBHelper.ZIMLIST_COLUMN_DB_PRICE));
```

```
        if (!rs.isClosed()) {
```

```
            rs.close();
```

```
    }
```

# Shopping List – 찜한 내역 확인



## 핵심1. DBHelper.java에 getZimlist() 함수 작성

```
public ArrayList getZimlist() {
    ArrayList array_list = new ArrayList();
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery("select * from zimlist", null);
    res.moveToFirst();

    int seq = 1;
    while (res.isAfterLast() == false) {
        array_list.add(seq + "." + // 목록 순서 번호 매기기
            res.getString(res.getColumnIndex(ZIMLIST_COLUMN_DB_ITEM)) + " ₩" +
            res.getString(res.getColumnIndex(ZIMLIST_COLUMN_DB_MARTNAME)) + " || " +
            res.getString(res.getColumnIndex(ZIMLIST_COLUMN_DB_PRICE)));

        array_seq[seq] = res.getString(res.getColumnIndex(ZIMLIST_COLUMN_ID)); // 배열에 PK값 넣기
        seq++;

        res.moveToNext();
    }
    return array_list;
}
```

## 핵심2. 찜 리스트로 이동시 찜한 목록 나타내는 onResume()함수 작성

//변화된 DB 내용 재 출력되도록 하는 함수

```
@Override
protected void onResume() {
    super.onResume();
    mAdapter.clear();
    mAdapter.addAll(mydb.getZimlist()); ← 리스트뷰에 DB 내용 나타내기
    mAdapter.notifyDataSetChanged();
}
```

# Shopping List – 전체 선택, 전체 해제

방구석마켓 



찜 LIST

예상금액



1. 품목 : 사과(부사) 이름 : 도곡시장    가격(원) : 2300원	<input checked="" type="checkbox"/>
2. 품목 : 달걀(10개) 이름 : 돈암제일시장    가격(원) : 3000원	<input checked="" type="checkbox"/>
3. 품목 : 돼지고기(생삼겹살) 이름 : 광장시장    가격(원) : 15000원	<input checked="" type="checkbox"/>
4. 품목 : 닭고기(육계) 이름 : 이마트 자양점    가격(원) : 6580원	<input checked="" type="checkbox"/>

전체 선택

전체 해제

선택삭제

방구석마켓 



찜 LIST

예상금액



1. 품목 : 사과(부사) 이름 : 도곡시장    가격(원) : 2300원	<input type="checkbox"/>
2. 품목 : 달걀(10개) 이름 : 돈암제일시장    가격(원) : 3000원	<input type="checkbox"/>
3. 품목 : 돼지고기(생삼겹살) 이름 : 광장시장    가격(원) : 15000원	<input type="checkbox"/>
4. 품목 : 닭고기(육계) 이름 : 이마트 자양점    가격(원) : 6580원	<input type="checkbox"/>

전체 선택

전체 해제

선택삭제

## 핵심1. For문을 이용하여 setItemChecked true/ false 만들기

// 전체 선택 버튼에 대한 이벤트 처리.

```
Button selectAllButton = (Button) findViewById(R.id.selectAll);
selectAllButton.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        int count = mAdapter.getCount();
        for (int i = 0; i < count; i++) {
            myListView.setItemChecked(i, true);
        }
    }
});
```

// 전체 해제 버튼에 대한 이벤트 처리.

```
Button selectNoneButton = (Button) findViewById(R.id.selectNone);
selectNoneButton.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        int count = mAdapter.getCount();
        for (int i = 0; i < count; i++) {
            myListView.setItemChecked(i, false);
        }
    }
});
```

# Shopping List – 다중 선택, 선택 삭제

방구석마켓 



찜 LIST

예상금액



1. 품목 : 사과(부사) 이름 : 도곡시장    가격(원) : 2300원	<input type="checkbox"/>
2. 품목 : 달걀(10개) 이름 : 돈암제일시장    가격(원) : 3000원	<input type="checkbox"/>
3. 품목 : 배(중품) 이름 : 남대문시장    가격(원) : 2700원	<input checked="" type="checkbox"/>
4. 품목 : 돼지고기(생삼겹살) 이름 : 광장시장    가격(원) : 15000원	<input type="checkbox"/>
5. 품목 : 상추(100g) 이름 : 세이브 마트    가격(원) : 700원	<input checked="" type="checkbox"/>
6. 품목 : 닭고기(육계) 이름 : 이마트 자양점    가격(원) : 6580원	<input type="checkbox"/>

전체 선택

전체 해제

선택삭제

방구석마켓 



찜 LIST

예상금액



1. 품목 : 사과(부사) 이름 : 도곡시장    가격(원) : 2300원	<input type="checkbox"/>
2. 품목 : 달걀(10개) 이름 : 돈암제일시장    가격(원) : 3000원	<input type="checkbox"/>
3. 품목 : 돼지고기(생삼겹살) 이름 : 광장시장    가격(원) : 15000원	<input type="checkbox"/>
4. 품목 : 닭고기(육계) 이름 : 이마트 자양점    가격(원) : 6580원	<input type="checkbox"/>

삭제되었습니다

전체 선택

전체 해제

선택삭제

## 핵심1. 리스트뷰에 어댑터 적용할 때 다중 선택 설정

```
mAdapter = new ArrayAdapter<this,
    android.R.layout.simple_list_item_multiple_choice, array_list>;
// 다중 선택가능
```

## 핵심2. DBHelper.java에서 deleteZimlist 함수 작성

```
public Integer deleteZimlist(Integer id) {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete("zimlist",
        "id = ? ",
        new String[]{Integer.toString(id)});
}
```

# Shopping List – 다중 선택, 선택 삭제 (이어서)

방구석마켓 



찜 LIST

예상금액



1. 품목 : 사과(부사) 이름 : 도곡시장    가격(원) : 2300원	<input type="checkbox"/>
2. 품목 : 달걀(10개) 이름 : 돈암제일시장    가격(원) : 3000원	<input type="checkbox"/>
3. 품목 : 배(중품) 이름 : 남대문시장    가격(원) : 2700원	<input checked="" type="checkbox"/>
4. 품목 : 돼지고기(생삼겹살) 이름 : 광장시장    가격(원) : 15000원	<input type="checkbox"/>
5. 품목 : 상추(100g) 이름 : 세이브 마트    가격(원) : 700원	<input checked="" type="checkbox"/>
6. 품목 : 닭고기(육계) 이름 : 이마트 자양점    가격(원) : 6580원	<input type="checkbox"/>

전체 선택

전체 해제

선택삭제

방구석마켓 



찜 LIST

예상금액



1. 품목 : 사과(부사) 이름 : 도곡시장    가격(원) : 2300원	<input type="checkbox"/>
2. 품목 : 달걀(10개) 이름 : 돈암제일시장    가격(원) : 3000원	<input type="checkbox"/>
3. 품목 : 돼지고기(생삼겹살) 이름 : 광장시장    가격(원) : 15000원	<input type="checkbox"/>
4. 품목 : 닭고기(육계) 이름 : 이마트 자양점    가격(원) : 6580원	<input type="checkbox"/>

삭제되었습니다

전체 선택

전체 해제

선택삭제

## 핵심3. 각 순서에 해당된 id(PK)값에 해당하는 리스트 삭제

```
// 삭제 버튼에 대한 이벤트 처리.
Button deleteButton = (Button) findViewById(R.id.delete);
deleteButton.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        String[] array = new String[50];
        SparseBooleanArray checkedItems = myListView.getCheckedItemPositions();
        int count = mAdapter.getCount();
        array = mydb.getArray_seq();

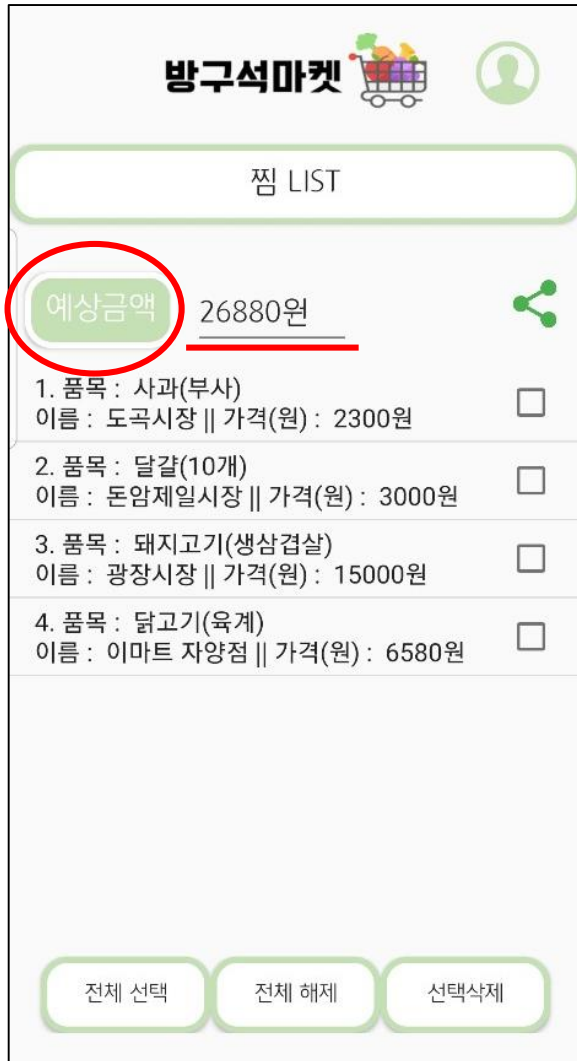
        for (int i = count - 1; i >= 0; i--) {
            if (checkedItems.get(i)) { // 체크된 목록이 있으면
                String tmp = array[i + 1]; // String으로 된 id값 String으로 받기
                int id = Integer.parseInt(tmp); // String id값 integer로 변환
                mydb.deleteZimlist(id); // id값 넣고 db 삭제

                Toast.makeText(getApplicationContext(), "삭제되었습니다", Toast.LENGTH_SHORT).show();
            }
        }
        myListView.clearChoices(); //선택된거 체크 풀기
        onResume(); // 자동 새로고침
    }
});
```

← onResume()함수를 이용하여  
자동 새로고침 되도록 하기



# Shopping List – 예상 금액 계산



**핵심1.** DBHelper.java에 가격 칼럼을 가져오는 함수 작성 →

```
// 총액 계산 버튼에 대한 이벤트 처리
total_p = (EditText) findViewById(R.id.total_p);
calButton = (Button) findViewById(R.id.cal);
calButton.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        result=0;

        int count = mAdapter.getCount();
        String[] array_p;
        int t;

        for(int i=0; i<count; i++) {
            array_p = mydb.getArray_price();
            price=array_p[i];
            t=Integer.parseInt(removeStringNumber(price));
            Log.i("price:",price);
            result=result+t;
        }
        total_p.setText(result+"원");
    }
});
```

```
public String[] getArray_price() {
    ArrayList array_list = new ArrayList();
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery("select * from zimlist", null);
    res.moveToFirst();
    int seq = 0;
    while (res.isAfterLast() == false) {
        array_price[seq] = res.getString(res.getColumnIndex
            (ZIMLIST_COLUMN_DB_PRICE));
        seq++;

        res.moveToNext();
    }
    return array_price;
}
```

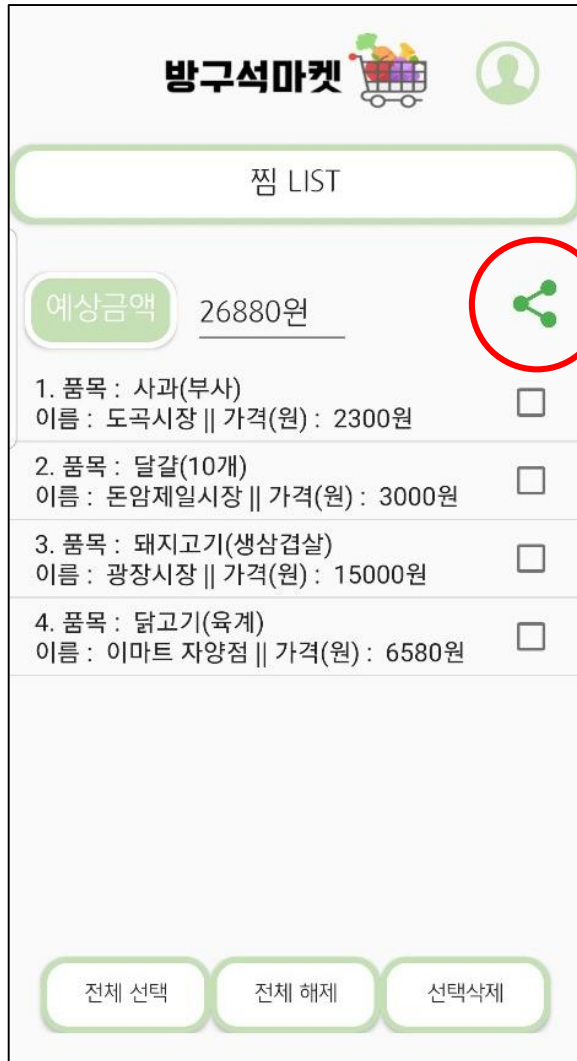
**핵심2.** for문을 이용하여 가격 칼럼만 더하기 ←

```
public static String removeStringNumber(String str) {
    return str.replaceAll("[^0-9]", "");
}
```

문자열에서 숫자만 추출하기



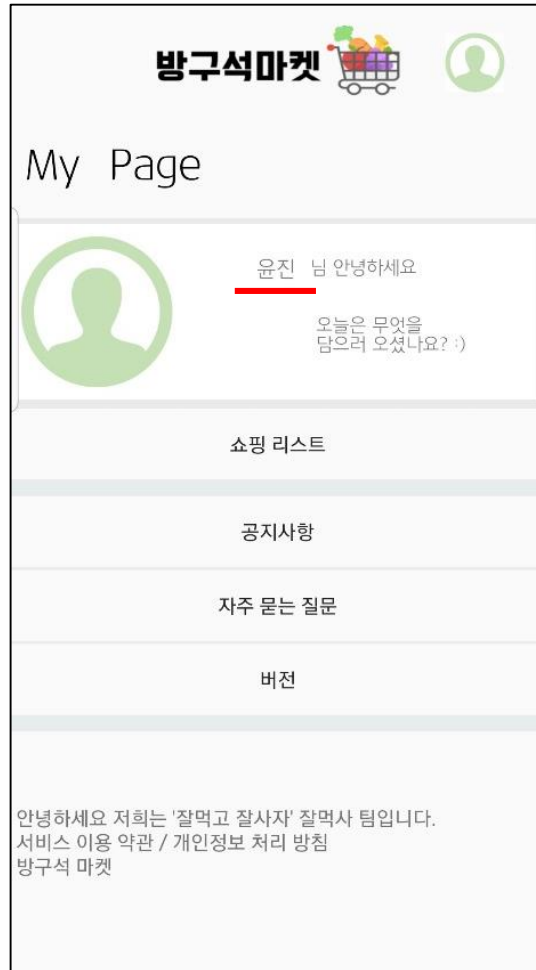
# Shopping List - 공유



## 핵심1. 다른 앱으로 간단한 데이터 보내기

```
share=(ImageButton)findViewById(R.id.share);
share.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        m="";
        int count = mAdapter.getCount();
        String[] array_p;
        String[] array_m;
        String[] array_i;
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_SEND);
        intent.setType("text/plain");
        for(int i=0; i<count; i++) {
            array_m = mydb.getArray_marname();
            content_marname=array_m[i];
            array_i = mydb.getArray_item();
            content_item=array_i[i];
            array_p = mydb.getArray_price();
            content_price=array_p[i];
            m=m+"\n"+(i+1)+" "+content_item+"\n" + " "+content_price+"\n" + " "+content_marname;
        }
        intent.putExtra(Intent.EXTRA_SUBJECT, "- 쇼핑리스트(방구석마켓) ");
        intent.putExtra(Intent.EXTRA_TEXT, "\n"+m+"\n사오세요~~~~~♥");
        Intent chooser = Intent.createChooser(intent, "공유");
        startActivity(chooser);
    }
});
```

# MyPage



## 핵심 1 DB에 회원 정보를 저장하고

```
public void onSuccess(MeV2Response result) {
    String nickName = result.getNickname();
    mydb.insertLogInfo(nickName);
    Log.d("myLog", "userProfile" + nickName);
    Toast.makeText(getApplicationContext(), "로그인 되었습니다",
    Toast.LENGTH_SHORT).show();
    //      Log.d("myLog", "userProfile" + result.getId());
    //      Log.d("myLog", "userProfile" + result.getNickname());
    //      Log.d("myLog",
    //      "userProfile" + result.getThumbnaillImagePath());
}
```

## 핵심 2 DB에 회원 정보를 저장하고 이 함수를 통해 textView에 보여줌

```
public String getNickname() {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery("select * from logInfo", null);
    res.moveToFirst();
    while (res.isAfterLast() == false) {
        nickname = res.getString(res.getColumnIndex(LOGIN_COLUMN_NICKNAME));
        res.moveToNext(); }
    return nickname;
}
```

## 05 기대효과

간편하게 집에서 가까운 위치의 마트 및 시장 가격 조회 가능

한눈에 식재료 시세 확인 가능

간편하게 장거리 목록 공유 가능

과일의 효능 및 야채의 구매 팁과 효능을 쉽게 알 수 있어 건강한 식습관에 도움이 됨

## 06 차후 개발 계획

구글맵

성북구에 한정되어 있는 마트 목록을 전국구로 확장

서울데이터

현재 16가지인 물품 종류를 더 다양하게 추가

공공데이터

서울에서만 한정되어 있는 데이터를 지역 확장하여 더 다양하게 구현

배너

배너에 다양한 마트 행사 추가

유튜브

유튜브에 머신 러닝 기능을 추가하여 개인에게 맞는 유튜브 추천

결제 기능

결제 기능 추가



감사합니다.