

임베디드 시스템 실습 lab14

2016310936 우승민

이번 exercise 는 content provider 을 사용하여 간단한 앱을 구현하는 것입니다.

먼저 layout 코드입니다.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Name"
            android:layout_marginLeft="100dp"
            android:layout_marginTop="100dp"/>

        <EditText
            android:id="@+id/txtName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="100dp"
            android:ems="10"/>

        <Button
            android:id="@+id/btnAdd"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="onClickAddDetails"
            android:layout_marginLeft="100dp"
            android:text="Add User" />

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

```

<Button
    android:id="@+id/btnDel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onClickDelDetails"
    android:layout_marginLeft="100dp"
    android:text="Delete User" />

<Button
    android:id="@+id/btnRetrieve"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onClickShowDetails"
    android:layout_marginLeft="100dp"
    android:text="Show Users" />

<TextView
    android:id="@+id/res"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:clickable="false"
    android:ems="10" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

이름을 입력하는 EditText 와 3 개의 Button, 2 개의 TextView 로 이루어져 있습니다.

Manifest 코드입니다.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.vogella.android.service.week14">
    <uses-permission android:name="com.example.week14.StudentsProvider.Permission.READ" />
    <uses-permission android:name="com.example.week14.StudentsProvider.Permission.WRITE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Week14">
        <provider
            android:authorities="com.example.week14.StudentsProvider"
            android:name=".StudentsProvider"
            android:readPermission="com.example.week14.StudentProvider.Permission.READ"
            android:writePermission="com.example.week14.StudentsProvider.Permission.WRITE"
            android:grantUriPermissions="true"
            android:exported="false"></provider>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

permission 을 통해 read 와 write 권한을 주고 중간에 provider 을 넣었습니다.

Main Activity 코드입니다.

```
package com.vogella.android.service.week14;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.view.MotionEvent;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event){
        InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), flags: 0);
        return true;
    }

    public void onClickAddDetails(View view){
        ContentValues values = new ContentValues();
        values.put(StudentsProvider.name, ((EditText) findViewById(R.id.txtName)).getText().toString());
        getContentResolver().insert(StudentsProvider.CONTENT_URI, values);
        Toast.makeText(getBaseContext(), text: "New Record Inserted", Toast.LENGTH_LONG).show();
    }

    public void onClickDelDetails(View view){
        String[] values = {((EditText) findViewById(R.id.txtName)).getText().toString()};
        getContentResolver().delete(StudentsProvider.CONTENT_URI, where: StudentsProvider.name + " LIKE ?", values);
        Toast.makeText(getBaseContext(), text: "Record Deleted", Toast.LENGTH_LONG).show();
    }

    public void onClickShowDetails(View view){
        TextView resultView = (TextView) findViewById(R.id.res);
        Cursor cursor = getContentResolver().query(Uri.parse("content://com.example.week14.StudentsProvider/users"));
        if(cursor.moveToFirst()){
            StringBuilder strBuild = new StringBuilder();
            while(!cursor.isAfterLast()){
                strBuild.append("\n"+cursor.getString(cursor.getColumnIndex(columnName: "id"))+"-"+cursor.getString(cursor.moveToNext());
            }
            resultView.setText(strBuild);
        }
        else{
            resultView.setText("No Records Found");
        }
    }
}
```

add, del, show 버튼 마다 각각 수행할 동작을 구분해주었습니다.

Students Provider 코드입니다.

```
public class StudentsProvider extends ContentProvider {
    static final String PROVIDER_NAME = "com.example.week14.StudentsProvider";
    static final String URL = "content://" + PROVIDER_NAME + "/users";
    static final Uri CONTENT_URI = Uri.parse(URL);
    static final String id = "id";
    static final String name = "name";
    static final int uriCode = 1;
    static final UriMatcher uriMatcher;
    private static HashMap<String, String> values;
    static {
        uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
        uriMatcher.addURI(PROVIDER_NAME, path: "users", uriCode);
        uriMatcher.addURI(PROVIDER_NAME, path: "users/*", uriCode);
    }
    @Override
    public boolean onCreate(){
        Context context = getContext();
        DatabaseHelper dbHelper = new DatabaseHelper(context);
        db = dbHelper.getWritableDatabase();

        return db != null;
    }
    @Override
    public String getType(Uri uri){
        switch (uriMatcher.match(uri)){
            case uriCode:
                return "vnd.android.cursor.dir/users";
            default:
                throw new IllegalArgumentException("UNSUPPORTED uri: " + uri);
        }
    }
    :

    public Cursor query(Uri uri, String[] projection, String selection,
                        String[] selectionArgs, String sortOrder) {
        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
        qb.setTables(TABLE_NAME);
        switch (uriMatcher.match(uri)) {
            case uriCode:
                qb.setProjectionMap(values);
                break;
            default:
                throw new IllegalArgumentException("Unknown URI " + uri);
        }
        if (sortOrder == null || sortOrder == "") {
            sortOrder = id;
        }
        Cursor c = qb.query(db, projection, selection, selectionArgs,
            groupBy: null, having: null, sortOrder);
        c.setNotificationUri(getContext().getContentResolver(), uri);
        return c;
    }
    @Override
    public int delete(Uri uri, String selection, String[] selectionArgs) {
        int count = 0;
        switch (uriMatcher.match(uri)) {
            case uriCode:
                count = db.delete(TABLE_NAME, selection, selectionArgs);
                break;
            default:
                throw new IllegalArgumentException("Unknown URI " + uri);
        }
        getContext().getContentResolver().notifyChange(uri, observer: null);
        return count;
    }
}
```

```

@Override
public Uri insert(Uri uri, ContentValues values) {
    long rowID = db.insert(TABLE_NAME, nullColumnHack: "", values);
    if (rowID > 0) {
        Uri _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
        getContext().getContentResolver().notifyChange(_uri, observer: null);
        return _uri;
    }
    throw new SQLiteException("Failed to add a record into " + uri);
}

@Override
public int update(Uri uri, ContentValues values, String selection,
    String[] selectionArgs) {
    int count = 0;
    switch (uriMatcher.match(uri)) {
        case uriCode:
            count = db.update(TABLE_NAME, values, selection, selectionArgs);
            break;
        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }
    getContext().getContentResolver().notifyChange(uri, observer: null);
    return count;
}
}

```

```

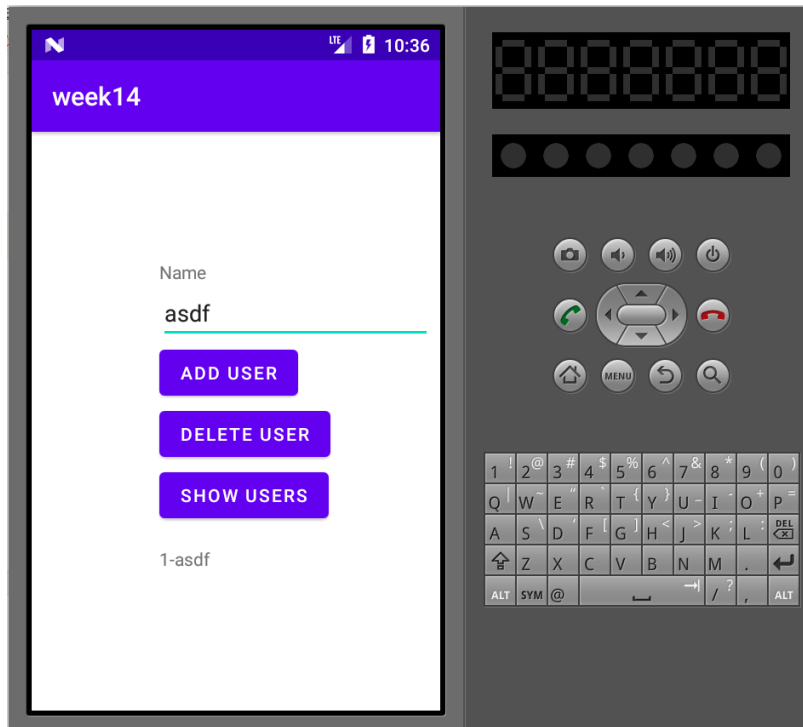
private SQLiteDatabase db;
static final String DATABASE_NAME = "EmpDB";
static final String TABLE_NAME = "Employees";
static final int DATABASE_VERSION = 1;
static final String CREATE_DB_TABLE = " CREATE TABLE " + TABLE_NAME
    + " (id INTEGER PRIMARY KEY AUTOINCREMENT, "
    + " name TEXT NOT NULL);";

private static class DatabaseHelper extends SQLiteOpenHelper{
    DatabaseHelper(Context context){
        super(context,DATABASE_NAME, factory: null,DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db){
        db.execSQL(CREATE_DB_TABLE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
        db.execSQL("DROP TABLE IF EXISTS" + TABLE_NAME);
        onCreate(db);
    }
}
}

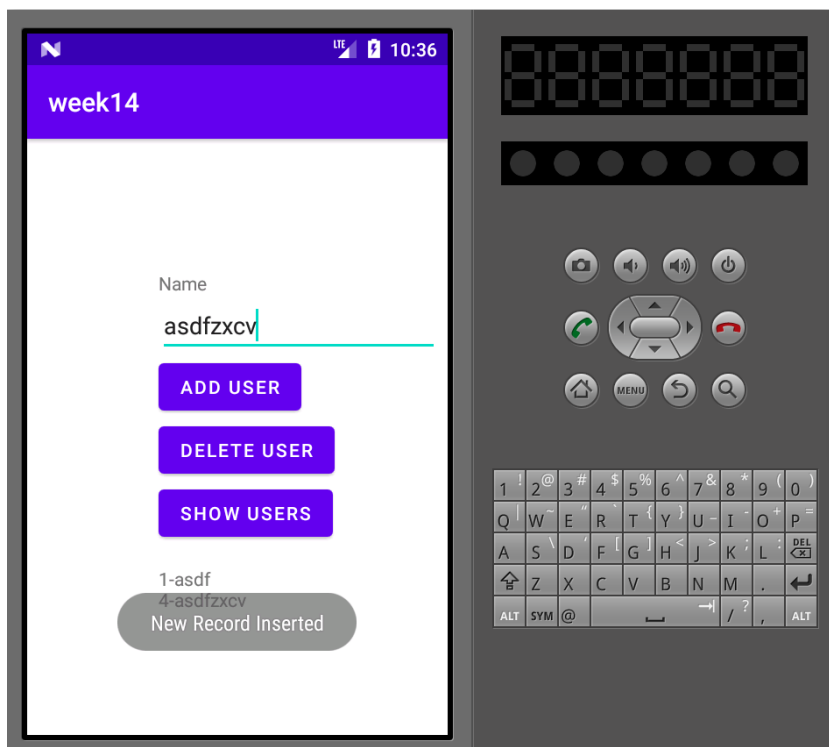
```

시연 화면입니다.

asdf 을 ADD 한 후 Show 를 한 화면입니다.



asdfzxcv 을 추가하는 화면입니다.



asdf 을 지우는 화면입니다.

