# CS355

## Mobile Application Development
## การพัฒนาโปรแกรมประยุกต์สำหรับอุปกรณ์พกพา


Pakorn Leesutthipornchai, Ph.D.
Assistant Professor
ผศ.ดร.ปกรณ์ ลี้สุทธิพรชัย
pakornl@cs.tu.ac.th

COMPUTER SCIENCE
THAMMASAT UNIVERSITY
www.cs.tu.ac.th


MA11: SQLite Database, Calling Other Activities and Getting Results from Called Activity
ฐานข้อมูล SQLite การเรียกกิจกรรมอื่น และ การรับค่าจากกิจกรรมที่ถูกเรียก

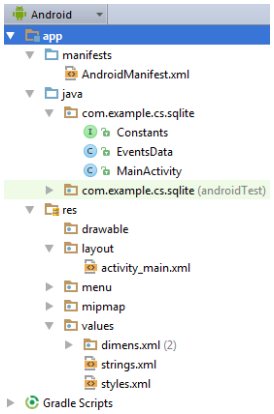


## SQLite : Layout


```
<LinearLayout          android:layout_width="match_parent"
                       android:layout_height="match_parent"
                       android:orientation="vertical"
                       android:gravity="center" >
  <EditText            android:id="@+id/editText"
                       android:text="@string/hello_world" />
  <LinearLayout>
    <ImageButton       android:id="@+id/button1"
          android:src="@android:drawable/ic_menu_add" />
    <ImageButton       android:id="@+id/button2"
          android:src="@android:drawable/ic_menu_edit" />
    <ImageButton       android:id="@+id/button3"
          android:src="@android:drawable/ic_menu_delete" />
    <ImageButton       android:id="@+id/button4"
          android:src="@android:drawable/ic_menu_revert" />
  </LinearLayout>

  <ScrollView          android:layout_width="match_parent"
                       android:layout_height="wrap_content">
    <TextView          android:id="@+id/text"
                       android:layout_width="wrap_content"
                       android:layout_height="wrap_content"
                       android:text="@string/hello_world" />
  </ScrollView>
</LinearLayout>
```



## SQLite Project




Constants.java

```
package com.example.cs.sqlite;
import android.provider.BaseColumns;

public interface Constants extends BaseColumns {
    public static final String TABLE_NAME = "events";
    public static final String TIME = "time";
    public static final String TITLE = "title";
}
```

EventsData.java

```
package com.example.cs.sqlite;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import static android.provider.BaseColumns._ID;
import static com.example.cs.sqlite.Constants.TABLE_NAME;
import static com.example.cs.sqlite.Constants.TIME;
import static com.example.cs.sqlite.Constants.TITLE;

public class EventsData extends SQLiteOpenHelper {
    public EventsData(Context ctx) { super(ctx, "events.db", null, 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME + " ("
                + _ID +" INTEGER PRIMARY KEY AUTOINCREMENT, "
                + TIME + " INTEGER, "
                + TITLE + " TEXT NOT NULL);" );
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS events");
        onCreate(db);
    }
}
```



## SQLite : Show Records

```
private Cursor getEvents() {
    String[] FROM = {_ID, TIME, TITLE};
    String ORDER_BY = TIME + " DESC";
    SQLiteDatabase db = events.getReadableDatabase();
    Cursor cursor = db.query(TABLE_NAME, FROM, null, null, null, null, ORDER_BY);
    return cursor;
}

private void showEvents(Cursor cursor) {
    StringBuilder builder = new StringBuilder("Saved events:\n");
    while(cursor.moveToNext()) {
        long id = cursor.getLong(0);
        long time = cursor.getLong(1);
        String title = cursor.getString(2);
        builder.append(id).append("  :  ");
        builder.append(time).append("  :  ");
        builder.append(title).append("\n ");
    }
    TextView text1 = (TextView) findViewById(R.id.text);
    text1.setText(builder);
}

private long getLastId(){
    long id = 0;
    SQLiteDatabase db = events.getWritableDatabase();
    String[] FROM = {_ID};
    String ORDER_BY = TIME + " DESC";
    Cursor cursor = db.query(TABLE_NAME, FROM, null, null, null, null, ORDER_BY, "1");
    while(cursor.moveToNext()){
        id = cursor.getLong(0);
    }
    return id;
}
```

SELECT ....
FROM ....
WHERE ....
GROUP BY ....
HAVING ....
ORDER BY ....
LIMIT ....

query (table, columns[], selection, selectionArgs[], groupBy, having, orderBy, limit)

```
String[] args = { "first string", "second string@string.com" };
Cursor cursor = db.query(TABLE_NAME, null, "name=? AND email=?", args, null, null, null);
```

# SQLite : Add Record

```
final ImageButton button1 = (ImageButton) findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        events = new EventsData(MainActivity.this);
        try{
            addEvent();
            Cursor cursor = getEvents();
            showEvents(cursor);
        }finally{
            events.close();
        }
    }
});

private void addEvent() {
    EditText et1 = (EditText) findViewById(R.id.editText);
    String string = String.format("%1$s", et1.getText());
    SQLiteDatabase db = events.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(TIME, System.currentTimeMillis());
    values.put(TITLE, string);
    db.insert(TABLE_NAME, null, values);
}
```



# SQLite : Edit Record

```
final ImageButton button2 = (ImageButton) findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        events = new EventsData(MainActivity.this);
        try{
            editEvent();
            Cursor cursor = getEvents();
            showEvents(cursor);
        }finally{
            events.close();
        }
    }
});
private void editEvent() {
    EditText et1 = (EditText) findViewById(R.id.editText);
    String string = String.format("%1$s", et1.getText());
    SQLiteDatabase db = events.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(TIME, System.currentTimeMillis());
    values.put(TITLE, string);
    db.update(TABLE_NAME, values, "ROWID="+getLastId(), null);
}
```

update(String table, ContentValues values, String whereClause, String[] whereArgs)



# SQLite : Delete Record

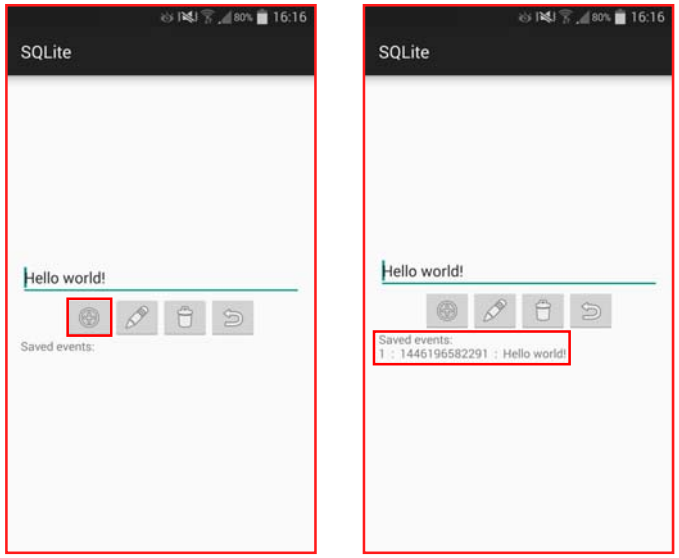
```
final ImageButton button3 = (ImageButton) findViewById(R.id.button3);
button3.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        events = new EventsData(MainActivity.this);
        try{
            deleteEvent();
            Cursor cursor = getEvents();
            showEvents(cursor);
        }finally{
            events.close();
        }
    }
});

private void deleteEvent() {
    SQLiteDatabase db = events.getWritableDatabase();
    db.delete(TABLE_NAME, "ROWID="+getLastId(), null);
}
```



# SQLite : Reset Auto Increment

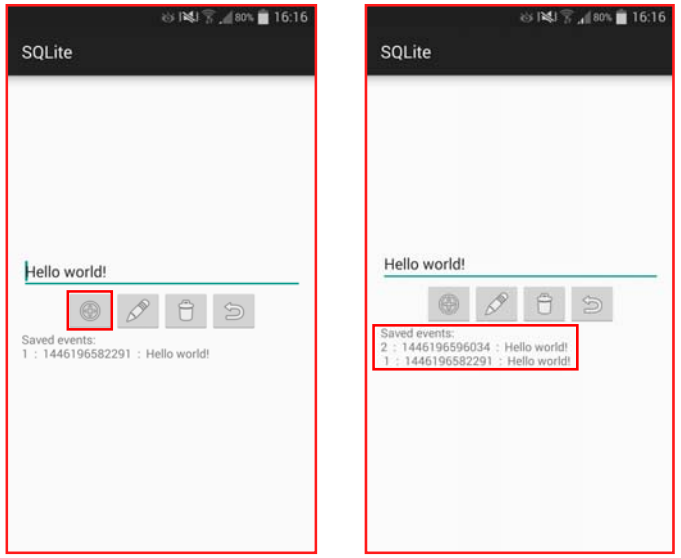
```
final ImageButton button4 = (ImageButton) findViewById(R.id.button4);
button4.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        events = new EventsData(MainActivity.this);
        try{
            resetAutoInc();
            Cursor cursor = getEvents();
            showEvents(cursor);
        }finally{
            events.close();
        }
    }
});

private void resetAutoInc() {
    SQLiteDatabase db = events.getWritableDatabase();
    db.delete(TABLE_NAME, null, null);
    db.execSQL("DELETE FROM SQLITE_SEQUENCE WHERE NAME = '" +
               TABLE_NAME + "'");
}
```

## SQLite : Before and After Click 'Add' (1)



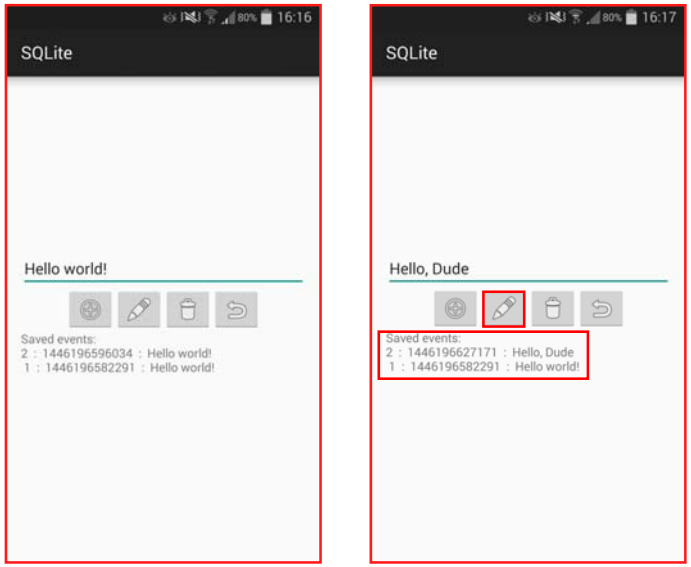



## SQLite : Before and After Click 'Add' (2)



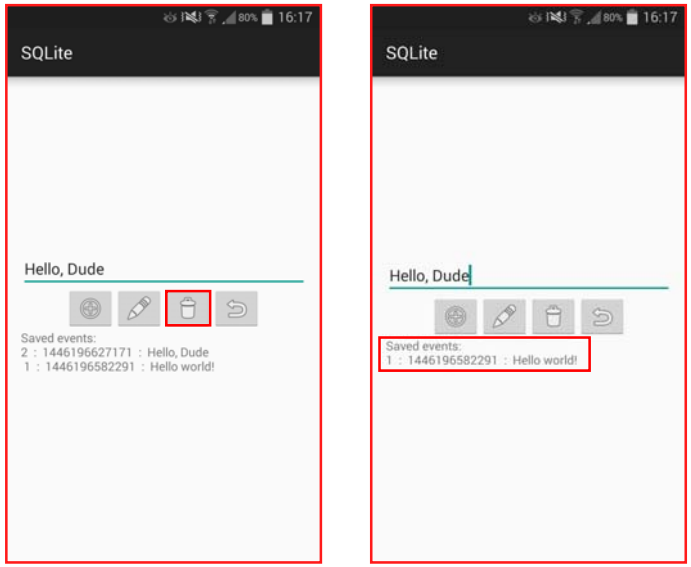



## SQLite : Before and After Click 'Edit'





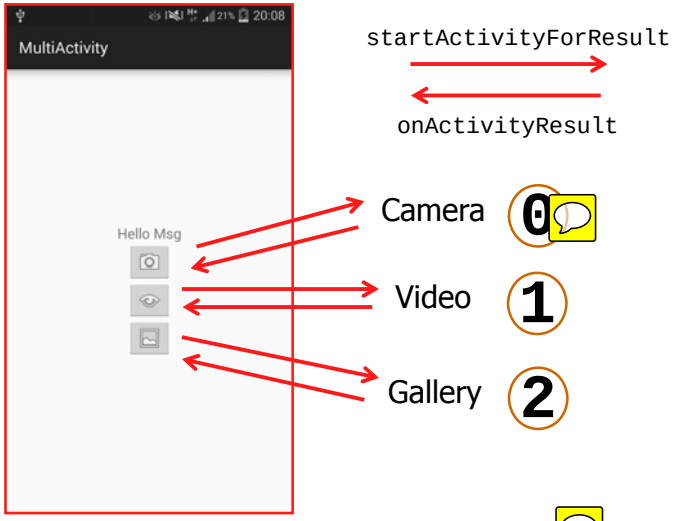

## SQLite : Before and After Click 'Delete'



## SQLite : Before and After Click 'Reset AutoIncrement'




## MultiActivity Project




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



## MultiActivity : MainActivity.java : 'Camera Image Button' Calls 'Camera Activity'

```
final ImageButton btn1 = (ImageButton) findViewById(R.id.camera_btn);
btn1.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss")
                                                    .format(new Date());
        String imageFileName = timeStamp + ".jpg";
        File f = new File(Environment.getExternalStoragePublicDirectory(
                    Environment.DIRECTORY_PICTURES),imageFileName);
        Uri fileUri = Uri.fromFile(f);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
        filePath = fileUri.toString();
        startActivityForResult(intent, 0);
    }
});
```



## MultiActivity : MainActivity.java : 'Video Image Button' Calls 'Video Activity'

```
final ImageButton btn2 = (ImageButton) findViewById(R.id.video_btn);
btn2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
        intent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY, 0);
        intent.putExtra(MediaStore.EXTRA_DURATION_LIMIT, 5);
        startActivityForResult(intent, 1);
    }
});
```

EXTRA_VIDEO_QUALITY: Added in API level 3
Currently value 0 means low quality, suitable for MMS messages, and value 1 means high quality.
In the future other quality levels may be added.

EXTRA_DURATION_LIMIT: Added in API level 8
Specify the maximum allowed recording duration in seconds.

# MultiActivity : MainActivity.java : 'Gallery Image Button' Calls 'Pick Image Activity'

```
final ImageButton btn3 = (ImageButton)
findViewById(R.id.gallery_btn);
btn3.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent(Intent.ACTION_PICK,
                    MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(intent, 2);
    }
});
```



# Return to Caller (Camera)

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode== 0 && resultCode == Activity.RESULT_OK) {
        try {
            Bitmap bmpPic = BitmapFactory.decodeFile(
                                                filePath.replace("file://",""));
            FileOutputStream bmpFile = new FileOutputStream(
                                                filePath.replace("file://",""));
            bmpPic = Bitmap.createScaledBitmap(bmpPic, 600, 400, true);
            Matrix mat = new Matrix();
            mat.postRotate(90);
            bmpPic = Bitmap.createBitmap(bmpPic, 0, 0,
                                bmpPic.getWidth(), bmpPic.getHeight(), mat, true);
            bmpPic.compress(Bitmap.CompressFormat.JPEG, 50, bmpFile);
            bmpFile.flush();
            bmpFile.close();
            Toast.makeText(MainActivity.this, "Image file is saved "+
                        filePath.replace("file://",""), Toast.LENGTH_LONG).show();
            ImageButton img = (ImageButton) findViewById(R.id.camera_btn);
            img.setImageBitmap(bmpPic);
            TextView txt = (TextView) findViewById(R.id.textView);
            txt.setText("Camera");
        } catch (Exception e) {
            Log.e("Log", "Error from Camera Activity");
        }
    }
```



# Return to Caller (Video)

```
if (requestCode== 1 && resultCode == Activity.RESULT_OK && data!=null) {
    try {
        Uri selectedImage = data.getData();
        String[] filePathColumn = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(selectedImage,
                                        filePathColumn, null, null, null);
        cursor.moveToFirst();
        int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
        String fileString = cursor.getString(columnIndex);
        cursor.close();

        Toast.makeText(MainActivity.this, "Video file is saved "+fileString,
                                                Toast.LENGTH_LONG).show();
        ImageButton img = (ImageButton) findViewById(R.id.video_btn);
        img.setImageResource(android.R.drawable.ic_media_play);
        TextView txt = (TextView) findViewById(R.id.textView);
        txt.setText("Video");
    } catch (Exception e) {
        Log.e("Log", "Error from Video Activity");
    }
}
```



# Return to Caller (Gallery)

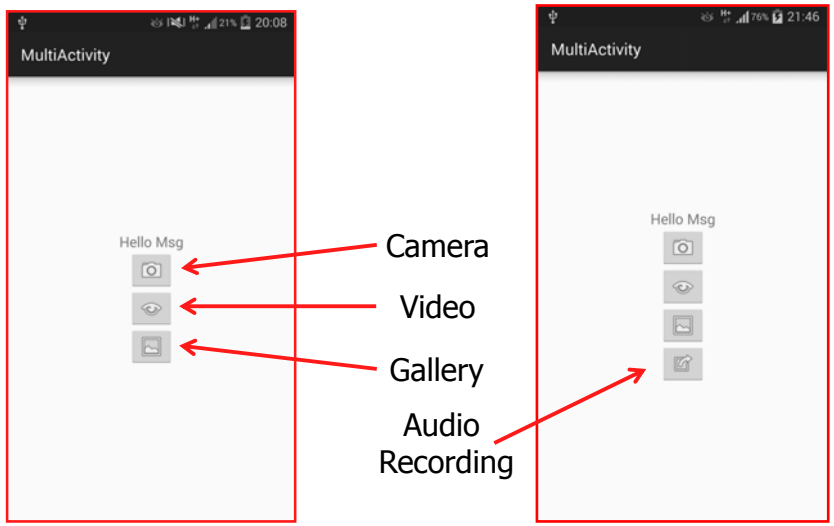
```
    if (requestCode== 2 && resultCode == Activity.RESULT_OK && data!=null) {
        try {
            Uri selectedImage = data.getData();
            String[] filePathColumn = {MediaStore.Images.Media.DATA};
            Cursor cursor = getContentResolver().query(selectedImage,
                                                filePathColumn, null, null, null);
            cursor.moveToFirst();
            int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
            String imgDecodableString = cursor.getString(columnIndex);
            cursor.close();

            Bitmap bmpPic = BitmapFactory.decodeFile(imgDecodableString);
            ImageButton img = (ImageButton) findViewById(R.id.gallery_btn);
            img.setImageBitmap(bmpPic);
            TextView txt = (TextView) findViewById(R.id.textView);
            txt.setText("Gallery");
        } catch (Exception e) {
            Log.e("Log", "Error from Gallery Activity");
        }
    }
}//end onActivityResult
```

# References

- http://developer.android.com
- http://developer.android.com/guide/topics/data/data-storage.html
- https://developer.android.com/guide/topics/providers/content-provider-basics.html



# MultiActivity Project : In Class Assignment




เพิ่ม Activity "Audio Recording" ดึงค่าไฟล์เสียงที่อัดจากทรัพยากรเครื่องมาใช้