一、开发过程

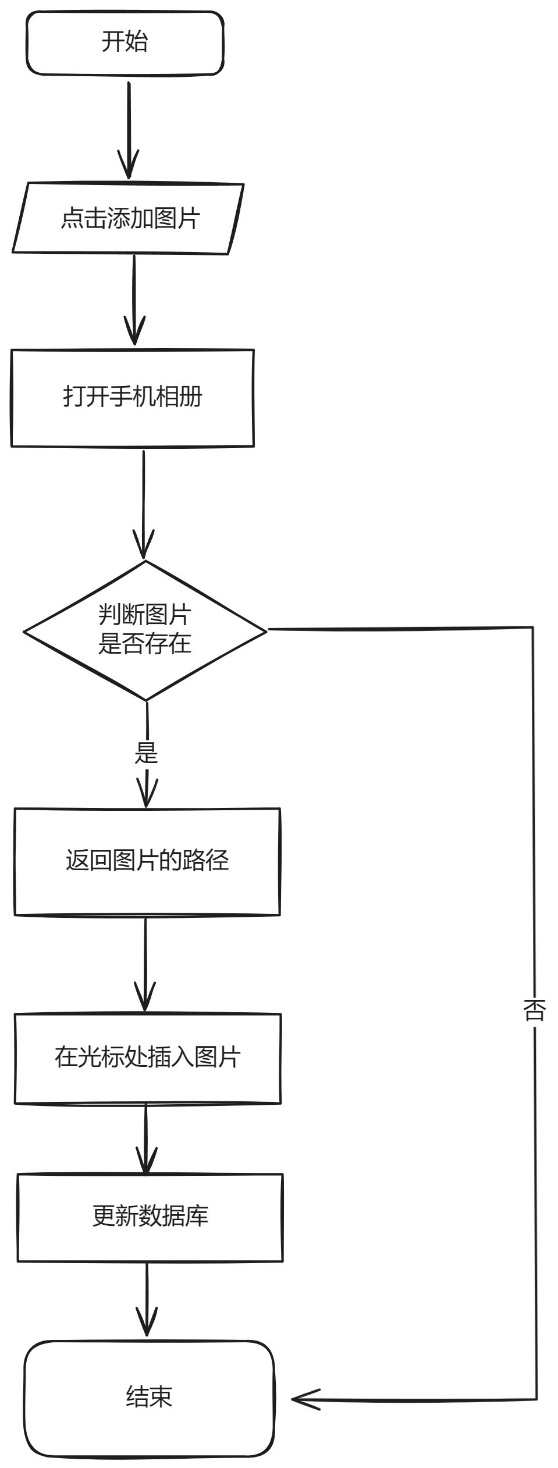
首先在note\_edit.xml文件中添加add\_img\_btn按钮；

在NoteEditActivity.java文件的onCreate()方法中，为这个“添加图片”按钮设置监听器，点击添加图片按钮时，会触发点击事件；

重写onActivityResult()来处理返回的数据，并将图片的路径也写入到数据库；

点击一个note后，会初始化note的内容，并通过convertToImage()将路径转化为图片；

在退出清单模式之后，仍应该将图片路径的位置替换为图片。



### 三、源代码

* 添加图片按钮的xml代码(FilePath: ***MiNotes\app\src\main\res\layout\note\_edit.xml***)

<ImageButton

android:id="@+id/add\_img\_btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="7dp"

android:layout\_marginTop="600dp"

android:layout\_marginBottom="7dp"

android:src="@android:drawable/ic\_menu\_gallery" />

* onCreate()方法(FilePath: ***MiNotes\app\src\main\java\net\micode\notes\ui\NoteEditActivity.java***)

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

this.setContentView(R.layout.note\_edit);

if (savedInstanceState == null && !initActivityState(getIntent())) {

finish();

return;

}

initResources();

//根据id获取添加图片按钮

final ImageButton add\_img\_btn = (ImageButton) findViewById(R.id.add\_img\_btn);

//为点击图片按钮设置监听器

add\_img\_btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Log.d(TAG, "onClick: click add image button");

//ACTION\_GET\_CONTENT: 允许用户选择特殊种类的数据，并返回（特殊种类的数据：照一张相片或录一段音）

Intent loadImage = new Intent(Intent.ACTION\_GET\_CONTENT);

//Category属性用于指定当前动作（Action）被执行的环境.

//CATEGORY\_OPENABLE; 用来指示一个ACTION\_GET\_CONTENT的intent

loadImage.addCategory(Intent.CATEGORY\_OPENABLE);

loadImage.setType("image/\*");

startActivityForResult(loadImage, PHOTO\_REQUEST);

}

});

}

* convertToImage()方法(FilePath: ***MiNotes\app\src\main\java\net\micode\notes\ui\NoteEditActivity.java***)

//路径字符串格式 转换为 图片image格式

private void convertToImage() {

NoteEditText noteEditText = (NoteEditText) findViewById(R.id.note\_edit\_view); //获取当前的edit

Editable editable = noteEditText.getText();//1.获取text

String noteText = editable.toString(); //2.将note内容转换为字符串

int length = editable.length(); //内容的长度

//3.截取img片段 [local]+uri+[local]，提取uri

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

for(int j = i; j < length; j++) {

String img\_fragment = noteText.substring(i, j+1); //img\_fragment：关于图片路径的片段

if(img\_fragment.length() > 15 && img\_fragment.endsWith("[/local]") && img\_fragment.startsWith("[local]")){

int limit = 7; //[local]为7个字符

//[local][/local]共15个字符，剩下的为真正的path长度

int len = img\_fragment.length()-15;

//从[local]之后的len个字符就是path

String path = img\_fragment.substring(limit,limit+len);//获取到了图片路径

Bitmap bitmap = null;

Log.d(TAG, "图片的路径是："+path);

try {

bitmap = BitmapFactory.decodeFile(path);//将图片路径解码为图片格式

} catch (Exception e) {

e.printStackTrace();

}

if(bitmap!=null){ //若图片存在

Log.d(TAG, "图片不为null");

ImageSpan imageSpan = new ImageSpan(NoteEditActivity.this, bitmap);

//4.创建一个SpannableString对象，以便插入用ImageSpan对象封装的图像

String ss = "[local]" + path + "[/local]";

SpannableString spannableString = new SpannableString(ss);

//5.将指定的标记对象附加到文本的开始...结束范围

spannableString.setSpan(imageSpan, 0, ss.length(), Spannable.SPAN\_EXCLUSIVE\_EXCLUSIVE);

Log.d(TAG, "Create spannable string success!");

Editable edit\_text = noteEditText.getEditableText();

edit\_text.delete(i,i+len+15); //6.删掉图片路径的文字

edit\_text.insert(i, spannableString); //7.在路径的起始位置插入图片

}

}

}

}

}

* onActivityResult()方法(FilePath: ***MiNotes\app\src\main\java\net\micode\notes\ui\NoteEditActivity.java***)

//重写onActivityResult()来处理返回的数据

protected void onActivityResult(int requestCode, int resultCode, Intent intent) {

super.onActivityResult(requestCode, resultCode, intent);

ContentResolver resolver = getContentResolver();

switch (requestCode) {

case PHOTO\_REQUEST:

Uri originalUri = intent.getData(); //1.获得图片的真实路径

Bitmap bitmap = null;

try {

bitmap = BitmapFactory.decodeStream(resolver.openInputStream(originalUri));//2.解码图片

} catch (FileNotFoundException e) {

Log.d(TAG, "onActivityResult: get file\_exception");

e.printStackTrace();

}

if(bitmap != null){

//3.根据Bitmap对象创建ImageSpan对象

Log.d(TAG, "onActivityResult: bitmap is not null");

ImageSpan imageSpan = new ImageSpan(NoteEditActivity.this, bitmap);

String path = getPath(this,originalUri);

//4.使用[local][/local]将path括起来，用于之后方便识别图片路径在note中的位置

String img\_fragment= "[local]" + path + "[/local]";

//创建一个SpannableString对象，以便插入用ImageSpan对象封装的图像

SpannableString spannableString = new SpannableString(img\_fragment);

spannableString.setSpan(imageSpan, 0, img\_fragment.length(), Spannable.SPAN\_EXCLUSIVE\_EXCLUSIVE);

//5.将选择的图片追加到EditText中光标所在位置

NoteEditText e = (NoteEditText) findViewById(R.id.note\_edit\_view);

int index = e.getSelectionStart(); //获取光标所在位置

Log.d(TAG, "Index是: " + index);

Editable edit\_text = e.getEditableText();

edit\_text.insert(index, spannableString); //将图片插入到光标所在位置

mWorkingNote.mContent = e.getText().toString();

//6.把改动提交到数据库中,两个数据库表都要改的

ContentResolver contentResolver = getContentResolver();

ContentValues contentValues = new ContentValues();

final long id = mWorkingNote.getNoteId();

contentValues.put("snippet",mWorkingNote.mContent);

contentResolver.update(Uri.parse("content://micode\_notes/note"), contentValues,"\_id=?",new String[]{""+id});

ContentValues contentValues1 = new ContentValues();

contentValues1.put("content",mWorkingNote.mContent);

contentResolver.update(Uri.parse("content://micode\_notes/data"), contentValues1,"mime\_type=? and note\_id=?", new String[]{"vnd.android.cursor.item/text\_note",""+id});

}else{

Toast.makeText(NoteEditActivity.this, "获取图片失败", Toast.LENGTH\_SHORT).show();

}

break;

default:

break;

}

}

* getPath()方法(FilePath: ***MiNotes\app\src\main\java\net\micode\notes\ui\NoteEditActivity.java***)

//获取文件的real path

public String getPath(final Context context, final Uri uri) {

final boolean isKitKat = Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.KITKAT;

// DocumentProvider

if (isKitKat && DocumentsContract.isDocumentUri(context, uri)) {

// ExternalStorageProvider

// if (isExternalStorageDocument(uri)) {

// final String docId = DocumentsContract.getDocumentId(uri);

// final String[] split = docId.split(":");

// final String type = split[0];

//

// if ("primary".equalsIgnoreCase(type)) {

// return Environment.getExternalStorageDirectory() + "/" + split[1];

// }

// }

// // DownloadsProvider

// else if (isDownloadsDocument(uri)) {

// final String id = DocumentsContract.getDocumentId(uri);

// final Uri contentUri = ContentUris.withAppendedId(Uri.parse("content://downloads/public\_downloads"), Long.valueOf(id));

// return getDataColumn(context, contentUri, null, null);

// }

// MediaProvider

// else

if (isMediaDocument(uri)) {

final String docId = DocumentsContract.getDocumentId(uri);

final String[] split = docId.split(":");

final String type = split[0];

Uri contentUri = null;

if ("image".equals(type)) {

contentUri = MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI;

}

final String selection = "\_id=?";

final String[] selectionArgs = new String[]{split[1]};

return getDataColumn(context, contentUri, selection, selectionArgs);

}

}

// Media

else if ("content".equalsIgnoreCase(uri.getScheme())) {

return getDataColumn(context, uri, null, null);

}

// File

else if ("file".equalsIgnoreCase(uri.getScheme())) {

return uri.getPath();

}

return null;

}