**基于Android的学生书籍管理软件**

班级:信1405-1班

姓名:杨程鑫

学号:20142863

[一、 系统的功能分析 3](#_Toc481944644)

[系统目标 3](#_Toc481944645)

[功能及技术需求 3](#_Toc481944646)

[运行环境 3](#_Toc481944647)

[二、 数据库表设计 3](#_Toc481944648)

[三、 源代码(除主配置文件，其它布局文件略) 4](#_Toc481944649)

[AndroidManifest.xml 4](#_Toc481944650)

[DataBase.java 5](#_Toc481944651)

[BookAddActivity.java 7](#_Toc481944652)

[BookFragment.java 9](#_Toc481944653)

[BookReviseActivity.java 10](#_Toc481944654)

[ClassifyFragment.java 13](#_Toc481944655)

[CountFragment.java 15](#_Toc481944656)

[Login.java 17](#_Toc481944657)

[MainActivity.java 18](#_Toc481944658)

[NotesAddActivity.java 33](#_Toc481944659)

[NotesFragment.java 34](#_Toc481944660)

[NotesReviseActivity.java 36](#_Toc481944661)

[UserAddActivity.java 37](#_Toc481944662)

[UserFragment.java 38](#_Toc481944663)

[四、 运行截图 40](#_Toc481944664)

[五、 心得体会 44](#_Toc481944665)

1. 系统的功能分析

## 系统目标

为了学习和巩固Android的UI设计，Activity以及SQLiteDatabase等知识点而设计的系统。

## 功能及技术需求

**功能要求：**

登陆，书籍分类管理，书籍管理，我的读书笔记，汇总每个用户的笔记数量，统计某一个用户的每一本书的笔记数量。

**技术要求：**

菜单，关于：软件版本的说明，软件中包含多个活动，数据库：SQLite，语言：Java，操作系统：Android，

## 运行环境

在Android Studio中安装的android SDK，测试环境API版本为21及以上，内存为512MB及以上，系统版本为Android6.0及以上。

1. 数据库表设计

**用户表：User**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文含义 | 类型 | 长度 | 是否为空 | 是否主键 | 是否外键 |
| Student\_ID | 学生编号(账号) | Text | - | N | Y | N |
| Password | 密码 | Text | - | N | N | N |

**图书分类表：BooksClassify**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文含义 | 类型 | 长度 | 是否为空 | 是否主键 | 是否外键 |
| Classify\_ID | 分类编号 | Text | - | N | Y | N |
| Name | 分类名称 | Text | - | N | N | N |

**图书表：Books**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文含义 | 类型 | 长度 | 是否为空 | 是否主键 | 是否外键 |
| Books\_Id | 图书编号 | Text | - | N | Y | N |
| Name | 图书名称 | Text | - | N | N | N |
| Author | 作者 | Text | - | N | N | N |
| Publisher | 出版社 | Text | - | N | N | N |
| ISBN | ISBN | Text | - | N | N | N |
| Classify\_ID | 分类编号 | Text | - | N | N | Y |
|  |  |  |  |  |  |  |

**图书笔记表：Notes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 中文含义 | 类型 | 长度 | 是否为空 | 是否主键 | 是否外键 |
| Notes\_Id | 笔记编号 | Text | - | N | Y | N |
| Books\_Id | 图书编号 | Text | - | N | N | Y |
| Student\_ID | 学生编号 | Text | - | N | N | Y |
| Content | 内容 | Text | - | N | N | N |

1. 源代码(除主配置文件，其它布局文件略)

## AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="ycx.ebook">

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name=".Login"

android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".BookReviseActivity" />

<activity

android:name=".MainActivity"

android:theme="@style/AppTheme.NoActionBar" />

<activity android:name=".BookAddActivity" />

<activity android:name=".UserAddActivity" />

<activity android:name=".NotesAddActivity"/>

<activity android:name=".NotesReviseActivity"/>

</application>

</manifest>

## DataBase.java

package ycx.ebook;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.util.Log;

public class DataBase extends SQLiteOpenHelper

{

public DataBase(Context context, String name, SQLiteDatabase.CursorFactory factory, int version)

{

super(context,name,factory,version);

}

public void onCreate(SQLiteDatabase db)

{

String sql = "Create Table User" +

"(" +

"Student\_ID text not null," +

"Password text not null," +

"constraint PK\_Studnet primary key(Student\_ID)" +

")";

db.execSQL(sql);

sql = "Insert into User Values('123','123')";

db.execSQL(sql);

sql = "Create Table BooksClassify" +

"(" +

"Classify\_ID text not null," +

"Name text not null," +

"constraint PK\_Studnet primary key(Classify\_ID)" +

")";

db.execSQL(sql);

sql = "Insert into BooksClassify Values('0','玄幻');";

db.execSQL(sql);

sql = "Insert into BooksClassify Values('1','奇幻');";

db.execSQL(sql);

sql = "Insert into BooksClassify Values('2','言情');";

db.execSQL(sql);

sql = "Insert into BooksClassify Values('3','科幻');";

db.execSQL(sql);

sql = "Insert into BooksClassify Values('4','异能');";

db.execSQL(sql);

sql = "Insert into BooksClassify Values('5','网游');";

db.execSQL(sql);

sql = "Create Table Books" +

"(" +

"Books\_ID text not null," +

"Name text not null," +

"Author text not null," +

"Publisher text not null," +

"ISBN text not null," +

"Classify\_ID text not null," +

"constraint PK\_Studnet primary key(Books\_ID)," +

"constraint FK\_Classify foreign key(Classify\_ID) references BooksClassify(Classify\_ID)" +

")";

db.execSQL(sql);

sql = "Insert into Books Values('0','机破星河','骷髅精灵','不知道','ISBN0000','3');";

db.execSQL(sql);

sql = "Insert into Books Values('1','斗破苍穹','天蚕土豆','不知道','ISBN0001','0');";

db.execSQL(sql);

sql = "Insert into Books Values('2','盘龙','我吃西红柿','不知道','ISBN0002','1');";

db.execSQL(sql);

sql = "Create Table Notes" +

"(" +

"Notes\_ID text not null," +

"Books\_ID text not null," +

"Student\_ID text not null," +

"Content text not null," +

"constraint PK\_Studnet primary key(Notes\_ID)," +

"constraint FK\_Classify foreign key(Books\_ID) references Books(Books\_ID)" +

"constraint FK\_Classify foreign key(Student\_ID) references User(Student\_ID)" +

")";

db.execSQL(sql);

sql = "Insert into Notes Values('0','0','123','不知道');";

db.execSQL(sql);

Log.d("Ebooks","调用DB.onCreate()");

}

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

{

Log.d("Ebooks","调用DB.onUpgrade()");

}

}

## BookAddActivity.java

package ycx.ebook;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import android.widget.TextView;

import java.util.ArrayList;

import java.util.List;

public class BookAddActivity extends AppCompatActivity

{

private List<String> classifyID = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_book\_add);

Cursor cursor = getClassify();

Spinner spinner = (Spinner) findViewById(R.id.classifyAddSpinner);

List<String> classifyName = new ArrayList<>();

while(cursor.moveToNext())

{

classifyName.add(cursor.getString(cursor.getColumnIndex("Name")));

classifyID.add(cursor.getString(cursor.getColumnIndex("Classify\_ID")));

}

ArrayAdapter<String> adapter = new ArrayAdapter<>(this, R.layout.spinner,classifyName);

spinner.setAdapter(adapter);

}

private Cursor getClassify()

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from BooksClassify";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

dataBase.close();

return cursor;

}

public void addButton\_Click(View view)

{

TextView nameTextView = (TextView)findViewById(R.id.bookNameAddText);

String name = nameTextView.getText().toString();

TextView authorTextView = (TextView)findViewById(R.id.authorAddText);

String author = authorTextView.getText().toString();

TextView publisherTextView = (TextView)findViewById(R.id.publisherAddText);

String publisher = publisherTextView.getText().toString();

TextView ISBNTextView = (TextView)findViewById(R.id.ISBNAddText);

String ISBN = ISBNTextView.getText().toString();

Spinner spinner = (Spinner) findViewById(R.id.classifyAddSpinner);

int index = spinner.getSelectedItemPosition();

String classify = classifyID.get(index);

addBook(name,author,publisher,ISBN,classify);

}

private void addBook(String name,String author,String publisher,String ISBN,String classify)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select Books\_ID from Books Order By Books\_ID DESC limit 1";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

int book\_ID = cursor.getInt(0);

sql = "Insert Into Books Values('" + String.valueOf(book\_ID + 1) +

"','" + name + "','" + author + "','" + publisher + "','" + ISBN + "','" + classify + "')";

dbo.execSQL(sql);

cursor.close();

dataBase.close();

finish();

}

}

## BookFragment.java

package ycx.ebook;

import android.database.Cursor;

import android.os.Bundle;

import java.util.ArrayList;

import java.util.HashMap;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ListView;

import android.widget.SimpleAdapter;

import android.database.sqlite.SQLiteDatabase;

public class BookFragment extends Fragment

{

private ArrayList<HashMap<String, Object>> listItems = new ArrayList<>(); //存放文字、图片信息

private ListView listView;

@Override

public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstanceState)

{

listView = (ListView)inflater.inflate(R.layout.fragment,this.listView);

return listView;

}

private void initListView()

{

listItems.clear();

getBooksArrayList();

//生成适配器的Item和动态数组对应的元素

SimpleAdapter listItemAdapter = new SimpleAdapter(getActivity(),

listItems, // listItems数据源

R.layout.list\_book, //ListItem的XML布局实现

new String[] {"ItemImage","ItemID","ItemTitle","ItemButton",}, //动态数组与ImageItem对应的子项

new int[ ] {R.id.BookImage,R.id.BookID,R.id.BookTitle,R.id.bookReviseButton} //list\_book.xml布局文件里面的一个ImageView的ID,一个TextView 的ID

);

listView.setAdapter(listItemAdapter);

}

private void getBooksArrayList()

{

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Books";

Cursor cursor = dbo.rawQuery(sql,null);

while (cursor.moveToNext())

{

HashMap<String, Object> map = new HashMap<>();

String id = cursor.getString(cursor.getColumnIndex("Books\_ID"));

String name = cursor.getString(cursor.getColumnIndex("Name"));

String author = cursor.getString(cursor.getColumnIndex("Author"));

map.put("ItemID",id); //存储id

map.put("ItemImage", R.drawable.camera); //图片

map.put("ItemTitle", "\t" + name + "\t" + author);

map.put("ItemButton","编辑"); //按钮

listItems.add(map);

}

cursor.close();

dataBase.close();

}

@Override

public void onResume()

{

super.onResume();

initListView();

}

}

## BookReviseActivity.java

package ycx.ebook;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import android.widget.TextView;

import java.util.ArrayList;

import java.util.List;

public class BookReviseActivity extends AppCompatActivity

{

private List<String> classifyID = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_book\_revise);

Cursor cursor = getClassify();

Spinner spinner = (Spinner) findViewById(R.id.classifyReviseSpinner);

List<String> classifyName = new ArrayList<>();

while(cursor.moveToNext())

{

classifyName.add(cursor.getString(cursor.getColumnIndex("Name")));

classifyID.add(cursor.getString(cursor.getColumnIndex("Classify\_ID")));

}

ArrayAdapter<String> adapter = new ArrayAdapter<>(this, R.layout.spinner,classifyName);

spinner.setAdapter(adapter);

Bundle bundle = this.getIntent().getExtras();

String book\_ID = bundle.get("book\_ID").toString();

cursor = getBook(book\_ID);

TextView nameTextView = (TextView)findViewById(R.id.bookNameEditText);

nameTextView.setText(cursor.getString(cursor.getColumnIndex("Name")));

TextView authorTextView = (TextView)findViewById(R.id.authorEditText);

authorTextView.setText(cursor.getString(cursor.getColumnIndex("Author")));

TextView publisherTextView = (TextView)findViewById(R.id.publisherEditText);

publisherTextView.setText(cursor.getString(cursor.getColumnIndex("Publisher")));

TextView ISBNTextView = (TextView)findViewById(R.id.ISBNEditText);

ISBNTextView.setText(cursor.getString(cursor.getColumnIndex("ISBN")));

String classify = cursor.getString(cursor.getColumnIndex("Classify\_ID"));

for(int i = 0;i < classifyID.size();i++)

{

if(classifyID.get(i).equals(classify))

{

spinner.setSelection(i,true);

}

}

}

public void reviseButton\_Click(View view)

{

Bundle bundle = this.getIntent().getExtras();

String book\_ID = bundle.get("Book\_ID").toString();

TextView nameTextView = (TextView)findViewById(R.id.bookNameEditText);

String newName = nameTextView.getText().toString();

TextView authorTextView = (TextView)findViewById(R.id.authorEditText);

String newAuthor = authorTextView.getText().toString();

TextView publisherTextView = (TextView)findViewById(R.id.publisherEditText);

String newPublisher = publisherTextView.getText().toString();

TextView ISBNTextView = (TextView)findViewById(R.id.ISBNEditText);

String newISBN = ISBNTextView.getText().toString();

Spinner spinner = (Spinner) findViewById(R.id.classifyReviseSpinner);

int index = spinner.getSelectedItemPosition();

String classify = classifyID.get(index);

updateBook(book\_ID,newName,newAuthor,newPublisher,newISBN,classify);

}

private Cursor getBook(String book\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Books Where Books\_ID = '" + book\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

dataBase.close();

return cursor;

}

private Cursor getClassify()

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from BooksClassify";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

dataBase.close();

return cursor;

}

private void updateBook(String book\_ID,String name,String author,String publisher,String ISBN,String classifyID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "update Books set Name = '" + name + "',Author = '" + author +

"',Publisher = '" + publisher + "',ISBN = '" + ISBN + "',Classify\_ID = '" +

classifyID + "' Where Books\_ID = '" + book\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

finish();

}

}

## ClassifyFragment.java

package ycx.ebook;

import android.database.Cursor;

import android.os.Bundle;

import java.util.ArrayList;

import java.util.HashMap;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ListView;

import android.widget.SimpleAdapter;

import android.database.sqlite.SQLiteDatabase;

public class ClassifyFragment extends Fragment

{

private ArrayList<HashMap<String, Object>> listItems = new ArrayList<HashMap<String, Object>>();; //存放文字、图片信息

private SimpleAdapter listItemAdapter; //适配器

private ListView listView;

@Override

public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstanceState)

{

listView = (ListView)inflater.inflate(R.layout.fragment,null);

return listView;

}

private void initListView()

{

listItems.clear();

getUserArrayList();

//生成适配器的Item和动态数组对应的元素

listItemAdapter = new SimpleAdapter(getActivity(),

listItems, // listItems数据源

R.layout.list\_classify, //ListItem的XML布局实现

new String[] {"ItemID","ItemButton",}, //动态数组与ImageItem对应的子项

new int[ ] {R.id.UserID,R.id.userEditButton} //list\_book.xml布局文件里面的一个ImageView的ID,一个TextView 的ID

);

listView.setAdapter(listItemAdapter);

}

private void getUserArrayList()

{

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from BooksClassify";

Cursor cursor = dbo.rawQuery(sql,null);

while (cursor.moveToNext())

{

HashMap<String, Object> map = new HashMap<String, Object>();

String id = cursor.getString(cursor.getColumnIndex("Classify\_ID"));

String name = cursor.getString(cursor.getColumnIndex("Name"));

map.put("ItemID","编号：" + id + "\t类别：" + name); //存储id

map.put("ItemButton","编辑"); //按钮

listItems.add(map);

}

dataBase.close();

}

@Override

public void onResume()

{

super.onResume();

initListView();

}

}

## CountFragment.java

package ycx.ebook;

import android.content.Context;

import android.content.SharedPreferences;

import android.database.Cursor;

import android.os.Bundle;

import java.util.ArrayList;

import java.util.HashMap;

import android.support.v4.app.Fragment;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ListView;

import android.widget.SimpleAdapter;

import android.database.sqlite.SQLiteDatabase;

public class CountFragment extends Fragment

{

private ArrayList<HashMap<String, Object>> listItems = new ArrayList<HashMap<String, Object>>();; //存放文字、图片信息

private SimpleAdapter listItemAdapter; //适配器

private ListView listView;

@Override

public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstanceState)

{

listView = (ListView)inflater.inflate(R.layout.fragment,null);

return listView;

}

private void initListView()

{

listItems.clear();

getUserArrayList();

//生成适配器的Item和动态数组对应的元素

listItemAdapter = new SimpleAdapter(getActivity(),

listItems, // listItems数据源

R.layout.list\_count, //ListItem的XML布局实现

new String[] {"Content"}, //动态数组与ImageItem对应的子项

new int[ ] {R.id.countContent} //list\_book.xml布局文件里面的一个ImageView的ID,一个TextView 的ID

);

listView.setAdapter(listItemAdapter);

}

private void getUserArrayList()

{

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select Count(\*),Student\_ID from Notes Group By Student\_ID";

Cursor cursor = dbo.rawQuery(sql,null);

while (cursor.moveToNext())

{

HashMap<String, Object> map = new HashMap<String, Object>();

String content = "";

content += "用户：" + cursor.getString(1) + "\t\t";

content += "笔记数量：" + cursor.getString(0);

map.put("Content",content); //存储id

listItems.add(map);

}

dataBase.close();

}

@Override

public void onResume()

{

super.onResume();

initListView();

}

private String getBookName(String book\_ID)

{

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Books Where Books\_ID = '" + book\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

String bookName = cursor.getString(cursor.getColumnIndex("Name"));

dataBase.close();

return bookName;

}

}

## Login.java

package ycx.ebook;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.database.sqlite.SQLiteDatabase;

import android.util.Log;

import android.view.View;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.TextView;

import android.database.Cursor;

public class Login extends AppCompatActivity

{

private DataBase dataBase = new DataBase(this,"Ebooks",null,1);

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

dataBase.getReadableDatabase();

setContentView(R.layout.activity\_login);

}

public void loginButton\_click(View view)

{

SQLiteDatabase dbo = dataBase.getWritableDatabase();

TextView userText = (TextView)findViewById(R.id.loginUserTextView);

String student\_ID = userText.getText().toString();

TextView passwordText = (TextView)findViewById(R.id.loginPasswordTextView);

String password = passwordText.getText().toString();

String sql = "Select Password from User Where Student\_ID = '" + student\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

while (cursor.moveToNext())

{

String realPassword = cursor.getString(cursor.getColumnIndex("Password"));

if(password.equals(realPassword))

{

SharedPreferences sharedPreferences = this.getSharedPreferences("User",Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("student\_ID",student\_ID);

editor.commit();

Bundle bundle = new Bundle();

bundle.putString("student\_ID",student\_ID);

Intent intent = new Intent(this,MainActivity.class);

intent.putExtras(bundle);

startActivity(intent);

finish();

}

}

cursor.close();

dataBase.close();

}

}

## MainActivity.java

package ycx.ebook;

import android.app.AlertDialog;

import android.app.Dialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.support.design.widget.FloatingActionButton;

import android.support.v4.app.Fragment;

import android.support.v4.app.FragmentManager;

import android.support.v4.app.FragmentTransaction;

import android.support.design.widget.NavigationView;

import android.support.v4.view.GravityCompat;

import android.support.v4.widget.DrawerLayout;

import android.support.v7.app.ActionBarDrawerToggle;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.PopupMenu;

import android.support.v7.widget.Toolbar;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.EditText;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener

{

private FragmentManager fragmentManager = getSupportFragmentManager();

private Fragment fragment;

private int fragmentFlag = 1;

private String student\_ID;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

Bundle bundle = this.getIntent().getExtras();

student\_ID = bundle.getString("student\_ID");

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.book\_add);

fab.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View view)

{

if(fragmentFlag == 1)

{

Intent intent = new Intent(MainActivity.this,BookAddActivity.class);

startActivity(intent);

}

else if(fragmentFlag == 2)

{

Intent intent = new Intent(MainActivity.this,UserAddActivity.class);

startActivity(intent);

}

else if(fragmentFlag == 3)

{

LayoutInflater layoutInflater = LayoutInflater.from(MainActivity.this);

final View classifyReviseView = layoutInflater.inflate(R.layout.classify\_add, null);

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle("新增类别").

setView(classifyReviseView).

setNegativeButton("确认", new DialogInterface.OnClickListener()

{

@Override

public void onClick(DialogInterface dialog, int which)

{

EditText classifyReviseEditView = (EditText)classifyReviseView.findViewById(R.id.classifyAdd);

String classify = classifyReviseEditView.getText().toString();

addClassify(classify);

initClassifyFragment();

}

}).

setPositiveButton("取消", new DialogInterface.OnClickListener()

{

@Override

public void onClick(DialogInterface dialog, int which)

{

}

}).

create();

alertDialog.show();

}

else if(fragmentFlag == 4)

{

String message = notesCount();

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle("数量汇总").

setMessage(message).

create();

alertDialog.show();

}

else if(fragmentFlag == 5)

{

Toast.makeText(MainActivity.this,"未定义操作",Toast.LENGTH\_SHORT);

}

}

});

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);

ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer, toolbar, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close);

drawer.setDrawerListener(toggle);

toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav\_view);

navigationView.setNavigationItemSelectedListener(this);

initBookFragment();

}

private void initBookFragment()

{

FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

fragment = new BookFragment();

fragmentTransaction.replace(R.id.content\_main,fragment);

fragmentTransaction.commit();

}

private void initUserFragment()

{

FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

fragment = new UserFragment();

fragmentTransaction.replace(R.id.content\_main,fragment);

fragmentTransaction.commit();

}

private void initClassifyFragment()

{

FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

fragment = new ClassifyFragment();

fragmentTransaction.replace(R.id.content\_main,fragment);

fragmentTransaction.commit();

}

private void initNotesFragment()

{

FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

fragment = new NotesFragment();

fragmentTransaction.replace(R.id.content\_main,fragment);

fragmentTransaction.commit();

}

private void initCountFragment()

{

FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

fragment = new CountFragment();

fragmentTransaction.replace(R.id.content\_main,fragment);

fragmentTransaction.commit();

}

@Override

public void onBackPressed()

{

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);

if (drawer.isDrawerOpen(GravityCompat.START))

{

drawer.closeDrawer(GravityCompat.START);

}

else

{

super.onBackPressed();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu)

{

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item)

{

int id = item.getItemId();

if (id == R.id.writeOff)

{

finish();

Intent intent = new Intent(this,Login.class);

startActivity(intent);

}

else if(id == R.id.about)

{

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle("关于").

setMessage("版本：V1.0\n开发人员：杨程鑫\n开发时间：2017/4/7-2017/4/13").

create();

alertDialog.show();

}

return super.onOptionsItemSelected(item);

}

@SuppressWarnings("StatementWithEmptyBody")

@Override

public boolean onNavigationItemSelected(MenuItem item)

{

int id = item.getItemId();

if (id == R.id.book)

{

FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

fragmentTransaction.replace(R.id.content\_main,new BookFragment());

fragmentTransaction.commit();

fragmentFlag = 1;

}

else if (id == R.id.user)

{

initUserFragment();

fragmentFlag = 2;

}

else if (id == R.id.classify)

{

initClassifyFragment();

fragmentFlag = 3;

}

else if (id == R.id.notes)

{

initNotesFragment();

fragmentFlag = 4;

}

else if(id == R.id.notesCount)

{

initCountFragment();

fragmentFlag = 5;

}

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);

drawer.closeDrawer(GravityCompat.START);

return true;

}

public void bookReviseButton\_Click(View view)

{

showBookPopupMenu(view);

}

private void showBookPopupMenu(final View view)

{

PopupMenu popupMenu = new PopupMenu(this, view);

popupMenu.getMenuInflater().inflate(R.menu.book\_menu, popupMenu.getMenu());

popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()

{

@Override

public boolean onMenuItemClick(MenuItem item)

{

LinearLayout linearLayout = (LinearLayout) view.getParent();

TextView textView = (TextView) linearLayout.getChildAt(0);

String book\_ID = textView.getText().toString();

if(item.getItemId() == R.id.notes\_add)

{

Bundle bundle = new Bundle();

bundle.putString("student\_ID",student\_ID);

bundle.putString("book\_ID",book\_ID);

Intent intent = new Intent(MainActivity.this,NotesAddActivity.class);

intent.putExtras(bundle);

startActivity(intent);

}

else if(item.getItemId() == R.id.book\_revise)

{

Bundle bundle = new Bundle();

bundle.putString("book\_ID",book\_ID);

Intent intent = new Intent(MainActivity.this,BookReviseActivity.class);

intent.putExtras(bundle);

startActivity(intent);

}

else if(item.getItemId() == R.id.info\_show)

{

Cursor cursor = getBook(book\_ID);

String name = cursor.getString(cursor.getColumnIndex("Name"));

String author = cursor.getString(cursor.getColumnIndex("Author"));

String publisher = cursor.getString(cursor.getColumnIndex("Publisher"));

String ISBN = cursor.getString(cursor.getColumnIndex("ISBN"));

String classify\_id = cursor.getString(cursor.getColumnIndex("Classify\_ID"));

String classify = getClassifyName(classify\_id);

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle(name).

setMessage("作者：" + author + "\n" + "分类：" + classify + "\n"

+ "出版社：" + publisher + "\n" + "ISBN：" + ISBN).

create();

alertDialog.show();

}

else if(item.getItemId() == R.id.book\_del)

{

bookDelete(book\_ID);

initBookFragment();

}

return false;

}

});

popupMenu.setOnDismissListener(new PopupMenu.OnDismissListener()

{

@Override

public void onDismiss(PopupMenu menu)

{

return;

}

});

popupMenu.show();

}

public void notesReviseButton\_Click(View view)

{

showNotesPopupMenu(view);

}

private void showNotesPopupMenu(final View view)

{

PopupMenu popupMenu = new PopupMenu(this, view);

popupMenu.getMenuInflater().inflate(R.menu.notes\_menu, popupMenu.getMenu());

popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()

{

@Override

public boolean onMenuItemClick(MenuItem item)

{

LinearLayout linearLayout = (LinearLayout) view.getParent();

TextView notesTextView = (TextView) linearLayout.getChildAt(0);

String notes\_ID = notesTextView.getText().toString();

TextView contentTextView = (TextView) linearLayout.getChildAt(2);

String content = contentTextView.getText().toString();

if(item.getItemId() == R.id.notes\_revise)

{

Bundle bundle = new Bundle();

bundle.putString("notes\_ID",notes\_ID);

bundle.putString("content",content);

Intent intent = new Intent(MainActivity.this,NotesReviseActivity.class);

intent.putExtras(bundle);

startActivity(intent);

}

else if(item.getItemId() == R.id.notes\_del)

{

notesDelete(notes\_ID);

initNotesFragment();

}

return false;

}

});

popupMenu.setOnDismissListener(new PopupMenu.OnDismissListener()

{

@Override

public void onDismiss(PopupMenu menu)

{

return;

}

});

popupMenu.show();

}

public void userEditButton\_Click(View view)

{

showUserPopupMenu(view);

}

private void showUserPopupMenu(final View view)

{

PopupMenu popupMenu = new PopupMenu(this, view);

popupMenu.getMenuInflater().inflate(R.menu.user\_menu, popupMenu.getMenu());

popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()

{

@Override

public boolean onMenuItemClick(MenuItem item)

{

LinearLayout linearLayout = (LinearLayout) view.getParent();

TextView textView = (TextView) linearLayout.getChildAt(0);

String[] textViewString = textView.getText().toString().split("：");

final String student\_ID = textViewString[1];

if(item.getItemId() == R.id.password\_revise)

{

LayoutInflater layoutInflater = LayoutInflater.from(MainActivity.this);

final View passwordReviseView = layoutInflater.inflate(R.layout.password\_revise, null);

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle("修改密码").

setView(passwordReviseView).

setNegativeButton("确认", new DialogInterface.OnClickListener()

{

@Override

public void onClick(DialogInterface dialog, int which)

{

EditText passwordReviseEditView = (EditText)passwordReviseView.findViewById(R.id.passwordRevise);

String newPassword = passwordReviseEditView.getText().toString();

EditText passwordReviseConfirmEditView = (EditText)passwordReviseView.findViewById(R.id.passwordReviseConfirm);

String newPasswordConfirm = passwordReviseConfirmEditView.getText().toString();

if(newPassword.equals(newPasswordConfirm))

{

updatePassword(student\_ID,newPassword);

}

}

}).

setPositiveButton("取消", new DialogInterface.OnClickListener()

{

@Override

public void onClick(DialogInterface dialog, int which)

{

}

}).

create();

alertDialog.show();

}

else if(item.getItemId() == R.id.password\_show)

{

Cursor cursor = getUser(student\_ID);

String name = cursor.getString(cursor.getColumnIndex("Password"));

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle("账号：" + student\_ID).

setMessage("密码：" + name).

create();

alertDialog.show();

}

else if(item.getItemId() == R.id.user\_del)

{

userDelete(student\_ID);

initUserFragment();

}

return false;

}

});

popupMenu.setOnDismissListener(new PopupMenu.OnDismissListener()

{

@Override

public void onDismiss(PopupMenu menu)

{

return;

}

});

popupMenu.show();

}

public void classifyEditButton\_Click(View view)

{

showClassifyPopupMenu(view);

}

private void showClassifyPopupMenu(final View view)

{

PopupMenu popupMenu = new PopupMenu(this, view);

popupMenu.getMenuInflater().inflate(R.menu.classify\_menu, popupMenu.getMenu());

popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener()

{

@Override

public boolean onMenuItemClick(MenuItem item)

{

LinearLayout linearLayout = (LinearLayout) view.getParent();

TextView textView = (TextView) linearLayout.getChildAt(0);

String[] textViewString = textView.getText().toString().split("\\t");

String[] textViewPrefixString = textViewString[0].split("：");

final String classify\_ID = textViewPrefixString[1];

if(item.getItemId() == R.id.classify\_revise)

{

LayoutInflater layoutInflater = LayoutInflater.from(MainActivity.this);

final View classifyReviseView = layoutInflater.inflate(R.layout.classify\_revise, null);

Dialog alertDialog = new AlertDialog.Builder(MainActivity.this).

setTitle("修改类别").

setView(classifyReviseView).

setNegativeButton("确认", new DialogInterface.OnClickListener()

{

@Override

public void onClick(DialogInterface dialog, int which)

{

EditText classifyReviseEditView = (EditText)classifyReviseView.findViewById(R.id.classifyRevise);

String newClassify = classifyReviseEditView.getText().toString();

updateClassify(classify\_ID,newClassify);

initClassifyFragment();

}

}).

setPositiveButton("取消", new DialogInterface.OnClickListener()

{

@Override

public void onClick(DialogInterface dialog, int which)

{

}

}).

create();

alertDialog.show();

}

else if(item.getItemId() == R.id.classify\_del)

{

classifyDelete(classify\_ID);

initClassifyFragment();

}

return false;

}

});

popupMenu.setOnDismissListener(new PopupMenu.OnDismissListener()

{

@Override

public void onDismiss(PopupMenu menu)

{

return;

}

});

popupMenu.show();

}

private String getClassifyName(String classify)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from BooksClassify Where Classify\_ID = '" + classify + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

dataBase.close();

return cursor.getString(cursor.getColumnIndex("Name"));

}

private Cursor getBook(String book\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Books Where Books\_ID = '" + book\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

dataBase.close();

return cursor;

}

private Cursor getUser(String student\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from User Where Student\_ID = '" + student\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

dataBase.close();

return cursor;

}

private void bookDelete(String book\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "delete from Books Where Books\_ID = '" + book\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

private void userDelete(String student\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "delete from User Where Student\_ID = '" + student\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

private void classifyDelete(String classify\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "delete from BooksClassify Where Classify\_ID = '" + classify\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

private void notesDelete(String notes\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "delete from Notes Where Notes\_ID = '" + notes\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

private void updatePassword(String student\_ID,String newPassword)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "update User Set Password = '" + newPassword + "' Where Student\_ID = '" + student\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

private void updateClassify(String classify\_ID,String newClassify)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "update BooksClassify Set Name = '" + newClassify + "' Where Classify\_ID = '" + classify\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

private void addClassify(String classify)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select Classify\_ID from BooksClassify Order By Classify\_ID DESC limit 1";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

int classify\_ID = cursor.getInt(0);

sql = "Insert Into BooksClassify Values('" + String.valueOf(classify\_ID + 1) + "','" + classify + "')";

dbo.execSQL(sql);

dataBase.close();

}

private String notesCount()

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select Count(\*),Books\_ID from Notes Where Student\_ID = '" + student\_ID + "' Group By Books\_ID";

Cursor cursor = dbo.rawQuery(sql,null);

String message = "";

while(cursor.moveToNext())

{

message += getBookName(cursor.getString(1)) + "：";

message += cursor.getString(0) + "\n";

}

dataBase.close();

return message;

}

private String getBookName(String book\_ID)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Books Where Books\_ID = '" + book\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

String bookName = cursor.getString(cursor.getColumnIndex("Name"));

dataBase.close();

return bookName;

}

}

## NotesAddActivity.java

package ycx.ebook;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

public class NotesAddActivity extends AppCompatActivity

{

private String student\_ID;

private String book\_ID;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_notes\_add);

Bundle bundle = this.getIntent().getExtras();

student\_ID = bundle.getString("student\_ID");

book\_ID = bundle.getString("book\_ID");

}

public void notesAddButton\_Click(View view)

{

EditText notesEditText = (EditText)findViewById(R.id.NotesText);

addNotes(notesEditText.getText().toString());

finish();

}

private void addNotes(String content)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select Notes\_ID from Notes Order By Notes\_ID DESC limit 1";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

int Notes\_ID = cursor.getInt(0);

sql = "Insert Into Notes Values('" + String.valueOf(Notes\_ID + 1) +

"','" + book\_ID + "','" + student\_ID + "','" + content + "')";

dbo.execSQL(sql);

dataBase.close();

}

}

## NotesFragment.java

package ycx.ebook;

import android.content.Context;

import android.content.SharedPreferences;

import android.database.Cursor;

import android.os.Bundle;

import java.util.ArrayList;

import java.util.HashMap;

import android.support.v4.app.Fragment;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ListView;

import android.widget.SimpleAdapter;

import android.database.sqlite.SQLiteDatabase;

public class NotesFragment extends Fragment

{

private ArrayList<HashMap<String, Object>> listItems = new ArrayList<HashMap<String, Object>>();; //存放文字、图片信息

private SimpleAdapter listItemAdapter; //适配器

private ListView listView;

@Override

public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstanceState)

{

listView = (ListView)inflater.inflate(R.layout.fragment,null);

return listView;

}

private void initListView()

{

listItems.clear();

getUserArrayList();

//生成适配器的Item和动态数组对应的元素

listItemAdapter = new SimpleAdapter(getActivity(),

listItems, // listItems数据源

R.layout.list\_notes, //ListItem的XML布局实现

new String[] {"NotesID","BooksID","Content","ItemButton",}, //动态数组与ImageItem对应的子项

new int[ ] {R.id.notes\_ID,R.id.book\_ID,R.id.content,R.id.notesEditButton} //list\_book.xml布局文件里面的一个ImageView的ID,一个TextView 的ID

);

listView.setAdapter(listItemAdapter);

}

private void getUserArrayList()

{

SharedPreferences sharedPreferences = getActivity().getSharedPreferences("User",Context.MODE\_PRIVATE);

String student\_ID = sharedPreferences.getString("student\_ID","");

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Notes Where Student\_ID = '" + student\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

while (cursor.moveToNext())

{

HashMap<String, Object> map = new HashMap<String, Object>();

String notes\_ID = cursor.getString(cursor.getColumnIndex("Notes\_ID"));

String books\_ID = cursor.getString(cursor.getColumnIndex("Books\_ID"));

String content = cursor.getString(cursor.getColumnIndex("Content"));

map.put("NotesID",notes\_ID); //存储id

map.put("BooksID",getBookName(books\_ID)); //存储id

map.put("Content",content); //存储id

map.put("ItemButton","编辑"); //按钮

listItems.add(map);

}

dataBase.close();

}

@Override

public void onResume()

{

super.onResume();

initListView();

}

private String getBookName(String book\_ID)

{

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from Books Where Books\_ID = '" + book\_ID + "'";

Cursor cursor = dbo.rawQuery(sql,null);

cursor.moveToNext();

String bookName = cursor.getString(cursor.getColumnIndex("Name"));

dataBase.close();

return bookName;

}

}

## NotesReviseActivity.java

package ycx.ebook;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

public class NotesReviseActivity extends AppCompatActivity

{

private String notes\_ID;

private String content;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_notes\_revise);

Bundle bundle = this.getIntent().getExtras();

notes\_ID = bundle.getString("notes\_ID");

content = bundle.getString("content");

final EditText notesEditText = (EditText)findViewById(R.id.NotesReviseText);

notesEditText.setText(content);

notesEditText.addTextChangedListener(new TextWatcher()

{

@Override

public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2)

{

}

@Override

public void onTextChanged(CharSequence charSequence, int i, int i1, int i2)

{

}

@Override

public void afterTextChanged(Editable editable)

{

updateNotes(notesEditText.getText().toString());

}

});

}

private void updateNotes(String content)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "Update Notes Set Content = '" + content + "' Where Notes\_ID = '" + notes\_ID + "'";

dbo.execSQL(sql);

dataBase.close();

}

}

## UserAddActivity.java

package ycx.ebook;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

public class UserAddActivity extends AppCompatActivity

{

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_user\_add);

}

public void userAddButton\_Click(View view)

{

TextView student\_IDViewText = (TextView)findViewById(R.id.userAddText);

String student\_ID = student\_IDViewText.getText().toString();

TextView passwordViewText = (TextView)findViewById(R.id.passwordAddText);

String password = passwordViewText.getText().toString();

TextView passwordConfirmViewText = (TextView)findViewById(R.id.passwordConfirmText);

String passwordConfirm = passwordConfirmViewText.getText().toString();

if(passwordConfirm.equals(password))

{

userAdd(student\_ID,password);

finish();

}

else

{

Toast.makeText(this,"密码不一致",Toast.LENGTH\_LONG);

}

}

private void userAdd(String student\_ID,String password)

{

DataBase dataBase = new DataBase(this,"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "Insert Into User Values('" + student\_ID + "','" + password + "')";

dbo.execSQL(sql);

dataBase.close();

}

}

## UserFragment.java

package ycx.ebook;

import android.database.Cursor;

import android.os.Bundle;

import java.util.ArrayList;

import java.util.HashMap;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ListView;

import android.widget.SimpleAdapter;

import android.database.sqlite.SQLiteDatabase;

public class UserFragment extends Fragment

{

private ArrayList<HashMap<String, Object>> listItems = new ArrayList<HashMap<String, Object>>();; //存放文字、图片信息

private SimpleAdapter listItemAdapter; //适配器

private ListView listView;

@Override

public View onCreateView(LayoutInflater inflater,ViewGroup container,Bundle savedInstanceState)

{

listView = (ListView)inflater.inflate(R.layout.fragment,null);

return listView;

}

private void initListView()

{

listItems.clear();

getUserArrayList();

//生成适配器的Item和动态数组对应的元素

listItemAdapter = new SimpleAdapter(getActivity(),

listItems, // listItems数据源

R.layout.list\_user, //ListItem的XML布局实现

new String[] {"ItemID","ItemButton",}, //动态数组与ImageItem对应的子项

new int[ ] {R.id.UserID,R.id.userEditButton} //list\_book.xml布局文件里面的一个ImageView的ID,一个TextView 的ID

);

listView.setAdapter(listItemAdapter);

}

private void getUserArrayList()

{

DataBase dataBase = new DataBase(getActivity(),"Ebooks",null,1);

SQLiteDatabase dbo = dataBase.getWritableDatabase();

String sql = "select \* from User";

Cursor cursor = dbo.rawQuery(sql,null);

while (cursor.moveToNext())

{

HashMap<String, Object> map = new HashMap<String, Object>();

String id = cursor.getString(cursor.getColumnIndex("Student\_ID"));

map.put("ItemID","账户：" + id); //存储id

map.put("ItemButton","编辑"); //按钮

listItems.add(map);

}

dataBase.close();

}

@Override

public void onResume()

{

super.onResume();

initListView();

}

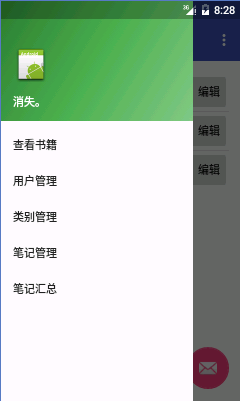
}

1. 运行截图

登陆界面： 图书管理(主界面)：

点击关于： 侧边栏(消失是我的QQ昵称。)：

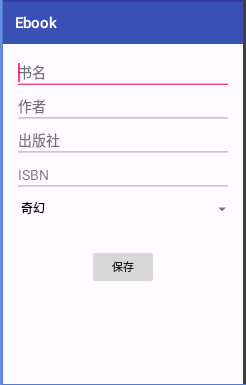
主界面菜单： 编辑菜单：

登陆后主界面默认显示所有图书的信息。

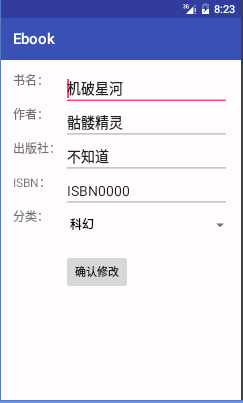
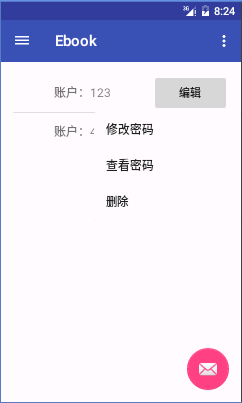
 

添加图书，此时提供的图书类别下拉列表时动态的， 查看图书详细信息：

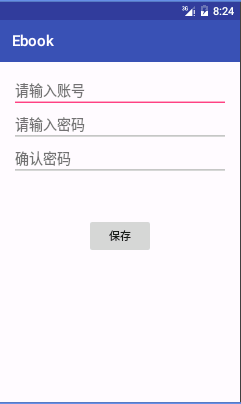
即增加图书列别后，下拉列表也会新增。

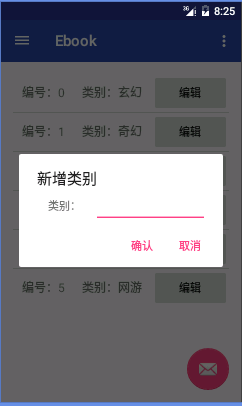
修改图书信息： 用户管理界面：

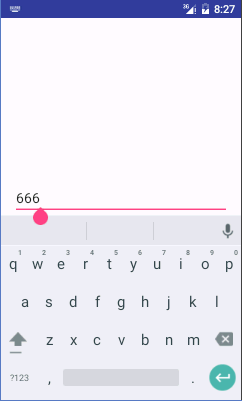
新增用户： 类别管理主界面：

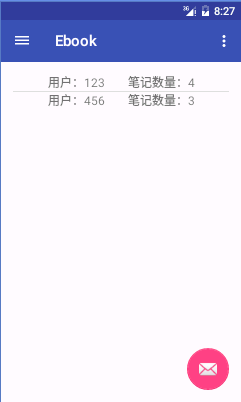
新增类别界面： 笔记管理主界面：

新增/编辑界面： 笔记汇总界面：

所有用户笔记数量汇总：



1. 心得体会

这是我第一次接触Android程序，从一开始什么也不会到现在可以说是入门，付出了也不少。从安装Android Studio，配置SDK和AVD都是一片空白。我对Android开发可以总结为以下几大方面：

1. 对Android组件的理解，比如说一个按钮的位置，程序大多推荐匹配内容和匹配父容器，应该是手机的像素有大有小吧。与java swing的设置理念不同，也和java web标签的位置声明也不同。然后，拖拽也不是很方便，所以困扰了我很多。最终还是设置成了横向匹配屏幕，纵向匹配内容了，也发现，好多软件的输入框也是这么设计的。
2. 对xml配置文件和适配器的理解。先实现线性布局的时候，就用到了大量的适配器，意思大致就是适配器读取xml，通过适配器操作xml中动态的部分，甚至可以批量生成。
3. 对点击事件的理解，现在只知道配置文件中声明的事件要写到activity中，不然的话，函数是找不到的。一开始，发现事件中总会有一个形参是View，编程的过程中，渐渐发现，View是大多数组件的抽象类，这个View指向用户点击的组件。
4. 对SQLite数据库的理解，在开发的过程中，因为设计不当和功能拓展造成的数据库表结构，数量的改变。就会导致修改datebase文件。按理说，应该是在onUpdate中写新的sql语句吧，然后提高数据库的版本。但是由于当时设计不够好，没有吧数据库名字和版本写入静态字符串，导致的多处修改。于是想，修改n个数据库版本和onUpdate和修改n个数据库名称，所以选择了后者。  
   还有就是发现SQLite语句与mysql更为相似，而且insert语句只支持执行一次，有些麻烦。
5. 对左侧滑动菜单和浮动菜单的理解，实现这个是因为兴趣。这两个方面实现的难点，大多在xml，只要有xml文件，只需要简单的调用就可以运行了。左侧滑动菜单配合fragment可以减少冗余的activity，实现多内容的显示。
6. Android较Java有更为细致的警告，敲代码的过程中，我没有理会警告。但是我完成了作业回头去看时，发现警告确实有道理，也修改了不少，下次就会注意了。