院 系 数据科学与计算机学院 学 号 　16337327　 姓 名 郑映雪

【实验题目】**Android数据库实验**

【实验目的】学习Android的ContentProvider、ActionBar、Json转换、SQlite操作、Fragment、PreferenceActivity、CustomListView、SimpleListView、AlertDialog、ProgressDialog、menu、Handler等。

【功能说明】

本实验需要编制两个项目程序，第一个是词典程序(词典app)，词典程序除了本身作为词典使用外，还可以把单词提供给其它需要使用的单词和例句的程序使用。第二个是背单词程序(被单词app)，利用词典程序获得单词，然后通过测验的方法背单词。

【实验说明】

（a）下面的实验内容把本次实验分成了很多步骤来完成，如果只完成其中一部分功能，在最后指明完成了哪些步骤。

（b）除了单步测试，调试可以采用Log.d("测试", ""+i); 通过Android Monitor，还可以使用单元检测。

（c）做每个步骤前，最好先把参考源码运行一遍。

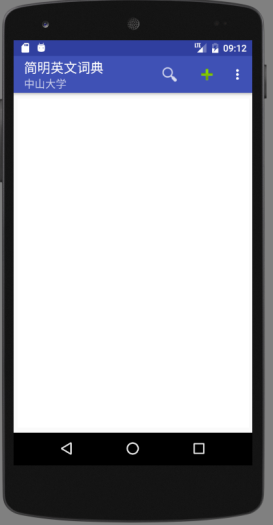
（d）可以先看一下老师的类设计（在本报告末尾）

【实验内容】

（一）词典核心功能设计

项目：GrandDictionary

1、做一个如下ActionBar界面。

**编程说明**： **android:icon="@drawable/ic\_action\_search"**

**android:icon="@android:drawable/ic\_input\_add"**

**android:checkable="true"**

**android:checked="false"**

**参考源码**：ActionBar

2、点击下载菜单完成单词下载(Json格式)，转换为Object，并保存到SQlite数据库中。如果单词已存在，则覆盖它。



**数据表：**db.execSQL(**"CREATE TABLE dict(\_id integer primary key autoincrement, word varchar(64) unique, explanation text, level int default 0, modified\_time timestamp)"**);

其中，level为单词难度，值越大越难

查看Json数据：**http://172.18.187.9:8080/dict/**

**注意事项**：

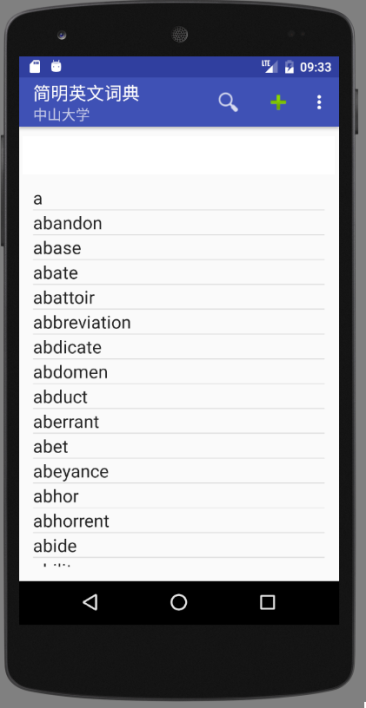
(1)AndroidManifest.xml中加入允许互联网访问：  
<**uses-permission android:name="android.permission.INTERNET"**/>

(2) 最好采用子线程进入下载：

**new** Thread() {  
 @Override  
 **public void** run() {  
 **try** {  
 DownloadDict downloadDict = **new** DownloadDict(**mContext**,**dictDb**);  
 downloadDict.start(**""**);  
 } **catch** (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
}.start();

**参考源码或数据**：WebServiceDemo（getJsonWeather方法）、Json1（推荐）、Json2、SqliteCv

3、用SimpleListView把数据库中的数据显示出来。



**参考源码**： SqliteCv、ListViewSimple (也可使用Cursor，参见ListViewCursor，不推荐)

4、增加一个ContentProvider。（可以结合背单词功能ContentResolver一起做）

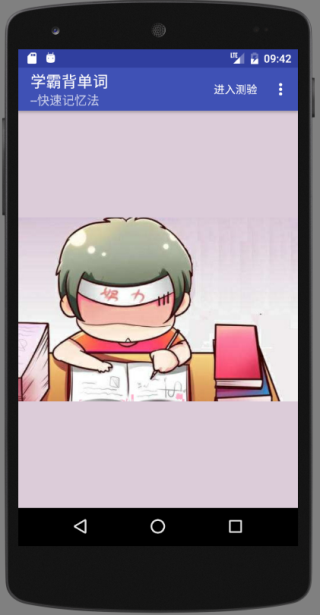
**参考源码**： SqliteCv、FirstProvider

**提示**：修改AndroidManifest.xml(用到的四大组件都要在其中配置)

（二）词典核心功能设计

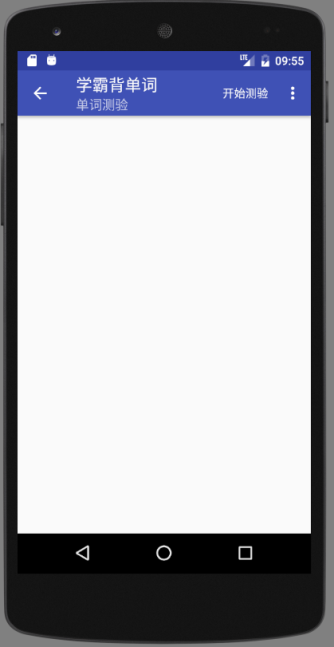
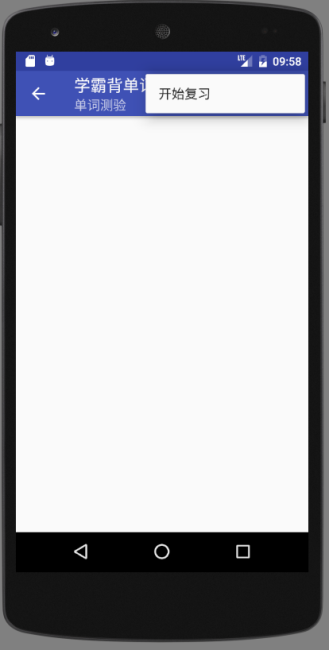
项目：GrandWordRemember

5、做一个主界面(MainActivity)

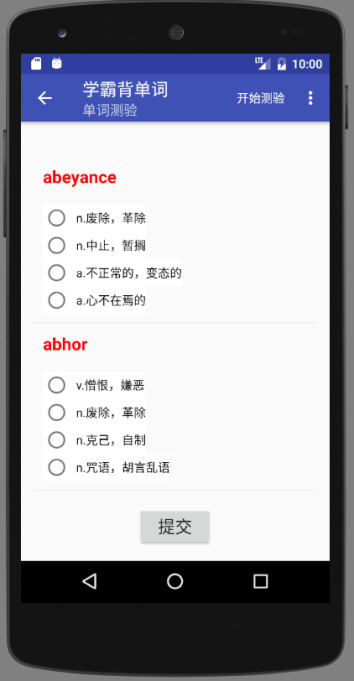
**参考资源**： rem.jpg

6、点击“进入测验”，进入测验界面（TestActivity）

**参考源码**：NewStartActivity

7、点击上面界面的开始测验，编写通过ContentResolver从词典程序中取前10个单词的程序并显示出来。选项从这10个单词的解释explanantion中随机选取，并包含要测验的单词，顺序也要打乱。



**参考源码**：FirstResolver、SpinnerCustom（参考做成自定义ListViewCustom）

**编程说明：**取10条记录的方法，OrderBy参数加上limit 10，即 "word limit 10"

**参考地址**：https://blog.csdn.net/Sunjianhua360/article/details/51394488

8、点击菜单“新增单词”，通过对话框为词典程序增加单词。



**操作说明**：要查询输入的单词是否存在，不存在则增加，存在则修改（允许覆盖）。

**编程提示：**可以单独定义一个类放对话框的功能。把Activity的上下文Context带入就可以了。上下文Context表示在什么地方显示这些对话框。

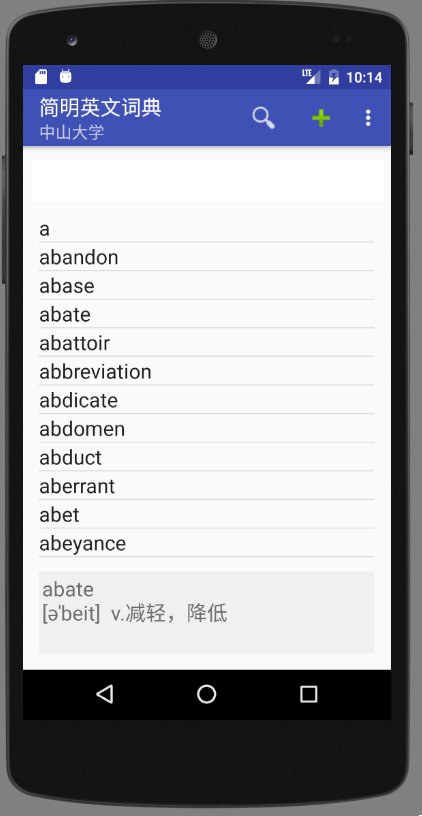
\* Context在Activity中Context可以用this或ActivityName.this取得，在事件处理程序中可以用ActivityName.this取得。

**参考源码**：FirstProvider 、FirstResolver、SqliteCv、AlertDialog

（三）词典扩展功能设计

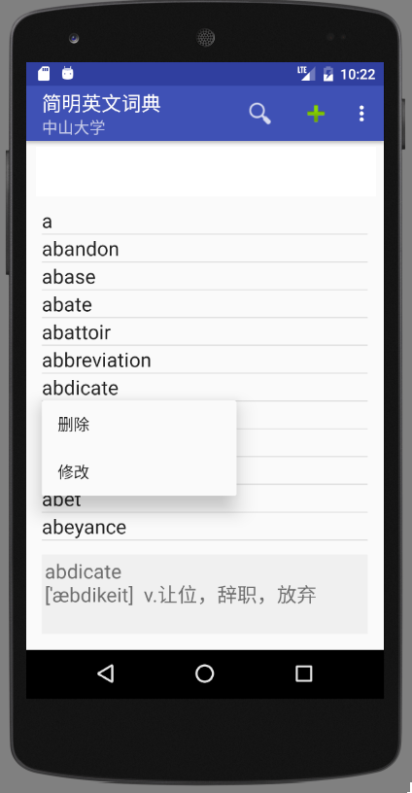
项目：GrandDictionary

9、增加信息显示栏，点击单词(setOnItemClickListener)将显示含义。



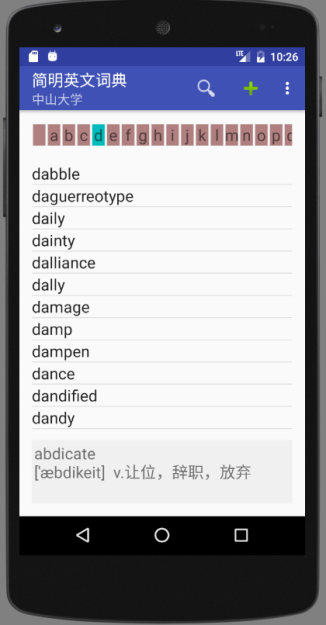
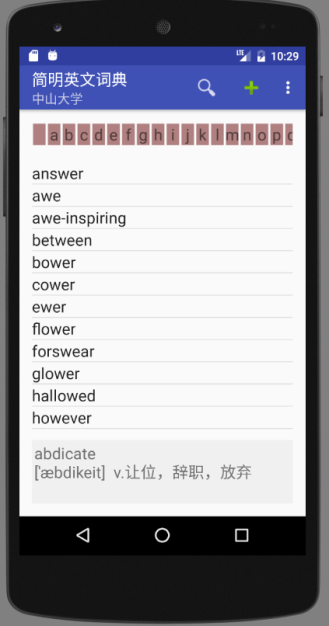
**参考源码**：ListViewSimple，SqliteCv

10、长点击单词(setOnItemLongClickListener)将显示弹出式菜单。



**参考源码**：Menu （PopupMenu）

11、增加单词查询功能

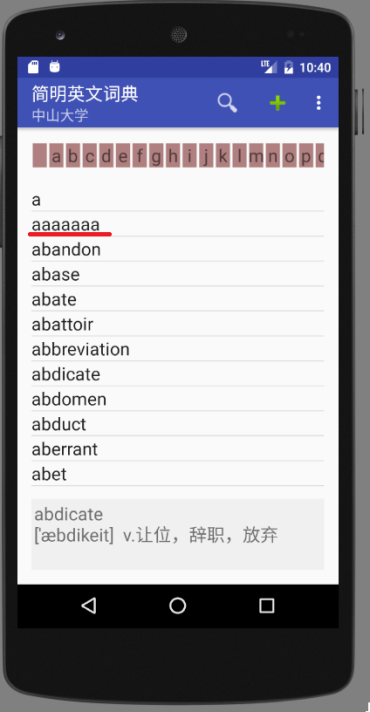
**功能说明**：在列表上方增加一个字母表，点击时只显示以这些字母开头的单词；点击查询图标可以查询包含输入子串的单词。

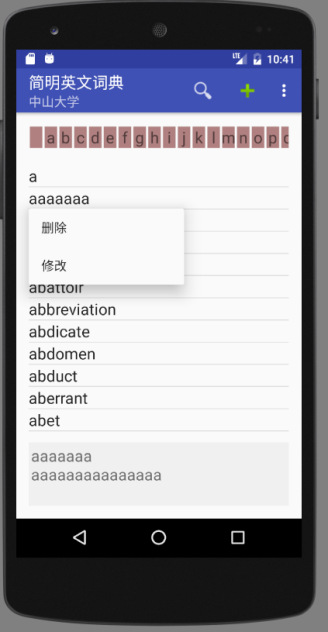
**参考源码**：DynamicView,HorizontalScrollView（自己上网查找使用方法）、SqliteCv

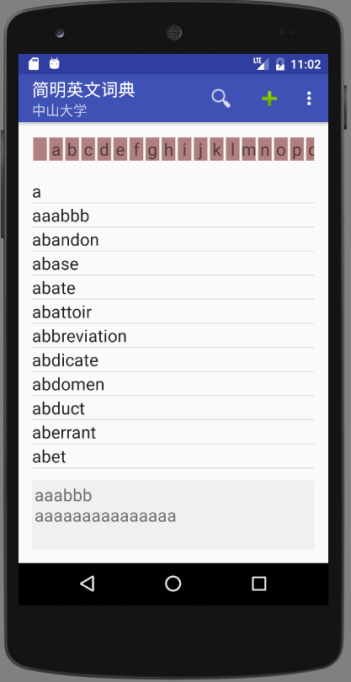
**编程参考**：每个单词可以从字母表” a…z ”取出一个字符(""+charAt()或substring()),然后创建TextView。空格查出所有单词

查询采用like的方法

12、通过图标和弹出式菜单用对话框方式实现增加、删除和修改单词。



**参考源码**：AlertDialog、SqliteCv

13、通过菜单可以让词典直接显示语义。



**参考源码**：ListViewSimple、SqliteCv，getPxFromDpi.txt

**功能说明**：显示语义时不显示信息栏。

14、下载和保存词典时增加一个进度显示。



**编程说明：**进度对话框放在UI主线程，下载和保存到数据库放到子线程，通过消息把进度传递给UI主线程。

**参考源码**：ProgressDialog（里面有一个Handler的例子），NewHandlerRunnable

（四）背单词扩展功能设计

项目：GrandWordRemember

15、建立统计数据库

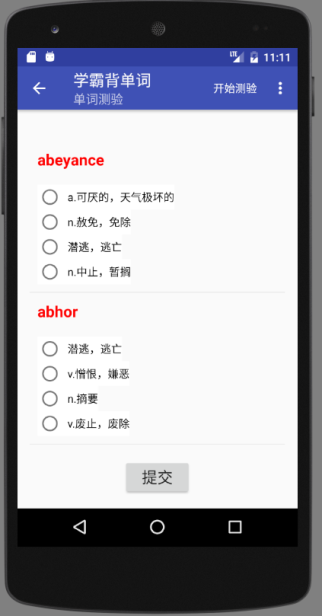
db.execSQL(**"CREATE TABLE words(\_id integer primary key autoincrement,word varchar(64) unique,level int default 0, test\_count int default 0, correct\_count int default 0,last\_test\_time timestamp)"**);

level 为单词单独级别（来自词典），test\_count为测试总次数，correct\_count为答对的次数。

16、每次取出的单词都是上述words表中没有的，实际做法可以先找出words最大的单词，然后从词典程序中按词典序取回接着的10个单词。

**参考源码**：FirstProvider，FirstResolver

17、提交后进行批改，看哪些题答对了，用背景色表示正确答案，并保存新单词到本地words表，修改已存在单词的统计数据。未选的作为错误。提交后隐藏提交按钮。

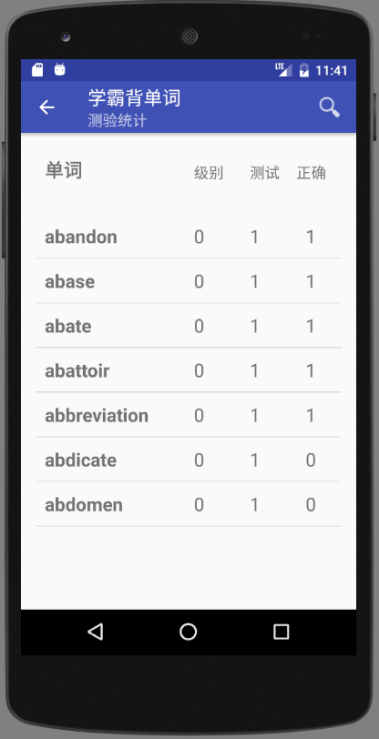
  

**参考源码**：SpinnerCustom（改为ListView），TestAdapter.java

**参考地址**：*https://blog.csdn.net/Sunjianhua360/article/details/51394488*

*https://blog.csdn.net/h623691640/article/details/52893166(*滚动混乱问题)

18、显示学习统计数据(StatActivity)



**参考源码**：ListViewSimple

19、增加“系统设置”功能，可以选择单词颜色，可以设置每次测试的单词数，以及保存时是否进行统计。要修改程序符合已设置的参数。

**参考源码**：NewPreferenceActivity

（五）自己设计功能（选做）

例如：

（a) 背单词的开始复习是从已学过的单词中选取要记忆的单词

（b）背单词的查找单词，从词典中找出单词。

（c）背单词的统计加入首字母筛选查询。

（d）把背单词的ActionBar上开始测试用带背景图的圆形图标代替

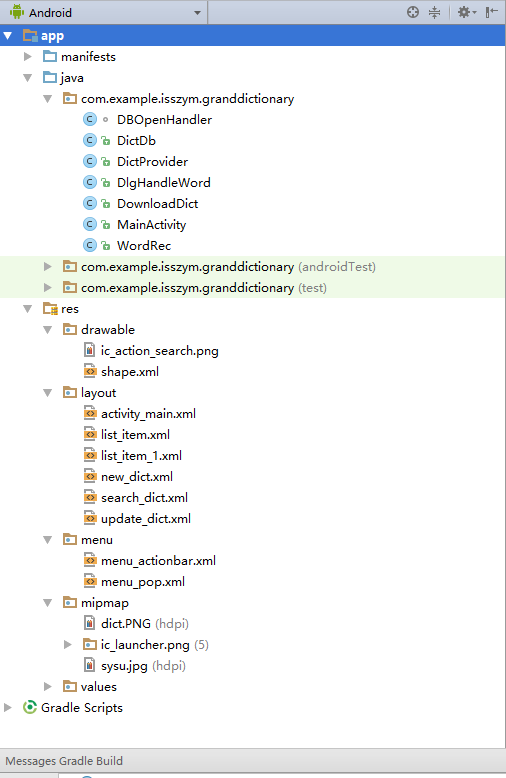
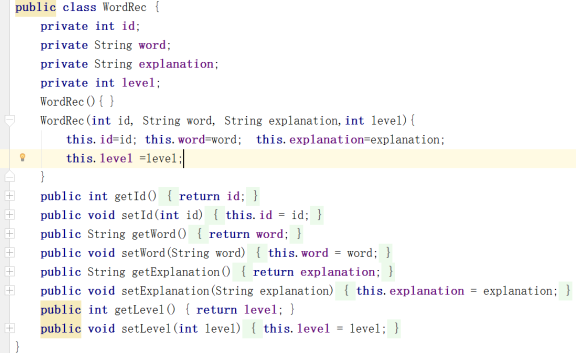
（e）利用中文解释，选择英文单词

（f）给出中文解释，填写英文单词

*……*

【参考Android Studio工程截屏】

词典程序(GrandDictionary)

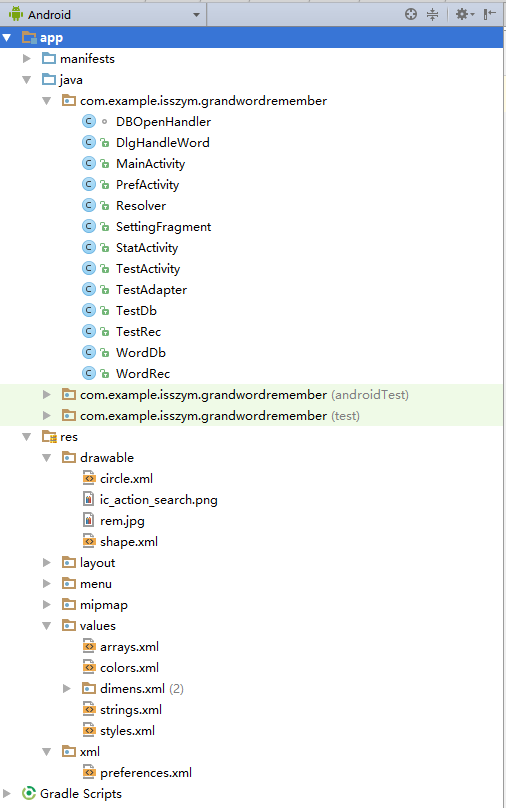
 

DbOpenHandler：打开数据库类 DictDb:所有数据库操作类

DictProvider：词典ContentProvider类 DlgHandleWord：所有对话框类

DownloadDict：下载词典类 MainActivity：主界面类 WordRec：保存单词记录类（见上图）

词典程序(GrandWordRemember)



DbOpenHandler：打开本地数据库 DlgHandleWord：包含所有对话框操作的类

MainActivity：主界面类 PrefActivity：配置界面类

Resolver：用于访问ContentProvider的ContentResolver类

SettingFragment：配置用的Fragment的类 StatActivity：统计界面类

TestActivity：测验界面类 TestAdapter：用作测验界面ListView的Adapter类（继承SimpleAdapter）

TestDb：本地数据库操作类 TestRec：本地统计记录类 WordDb：（远程）词典操作类

WordRec：单词记录类

*完成后程序运行截屏及简要说明：*

【词典程序】

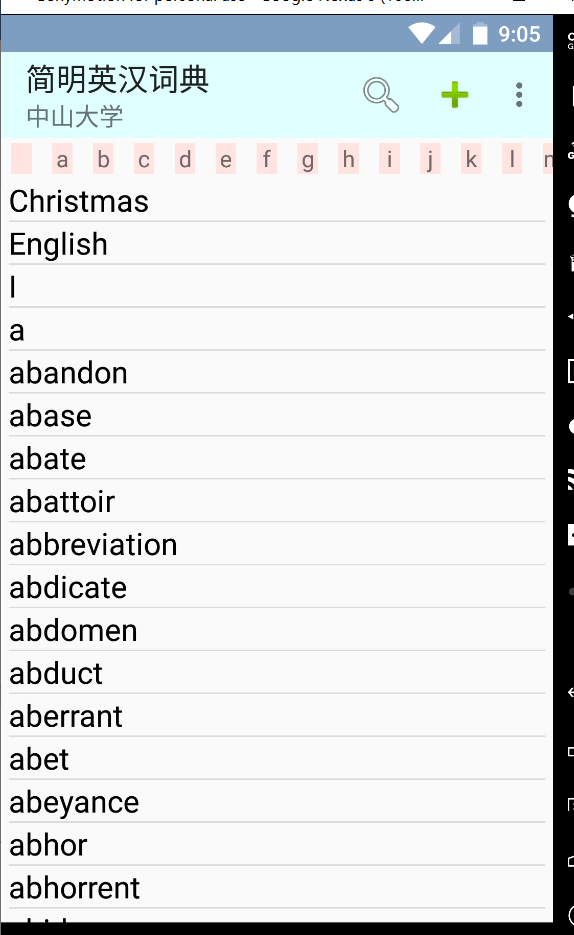
1. 初始界面



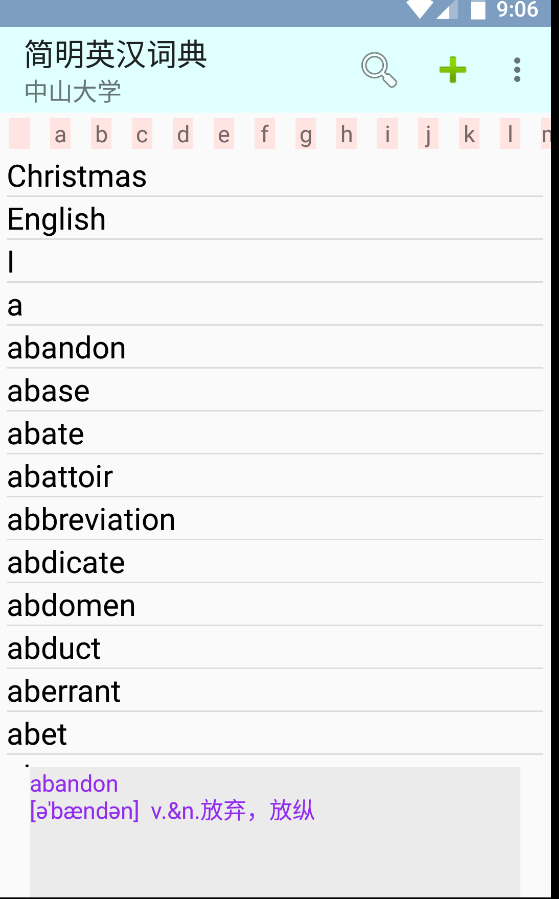
1. 下载单词



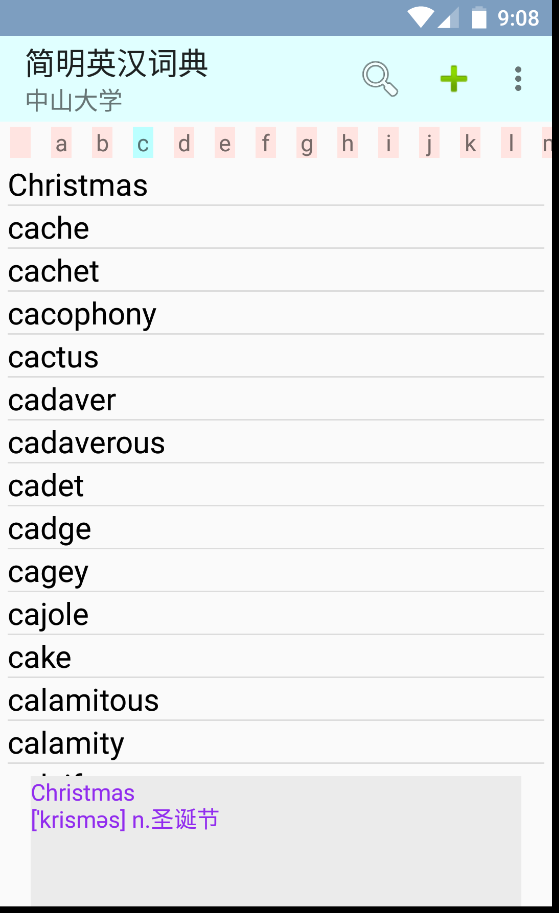
1. 显示界面



1. 点击显示词义



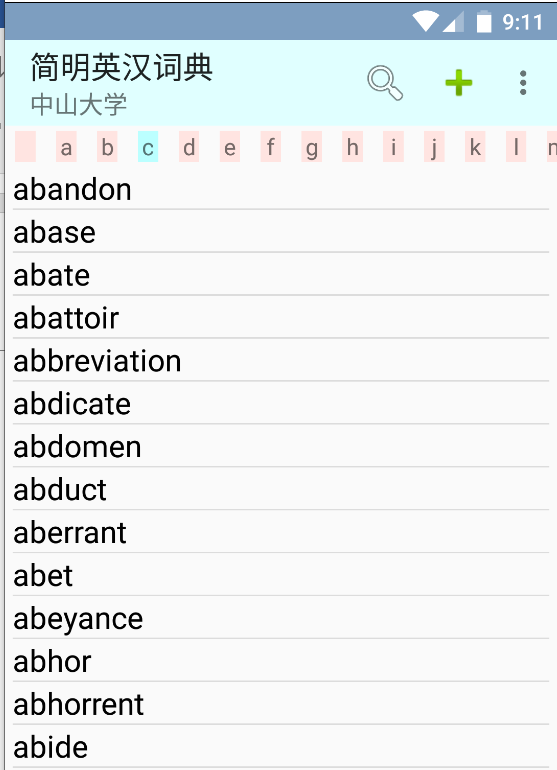
1. 点击索引栏的索引（以C为例）



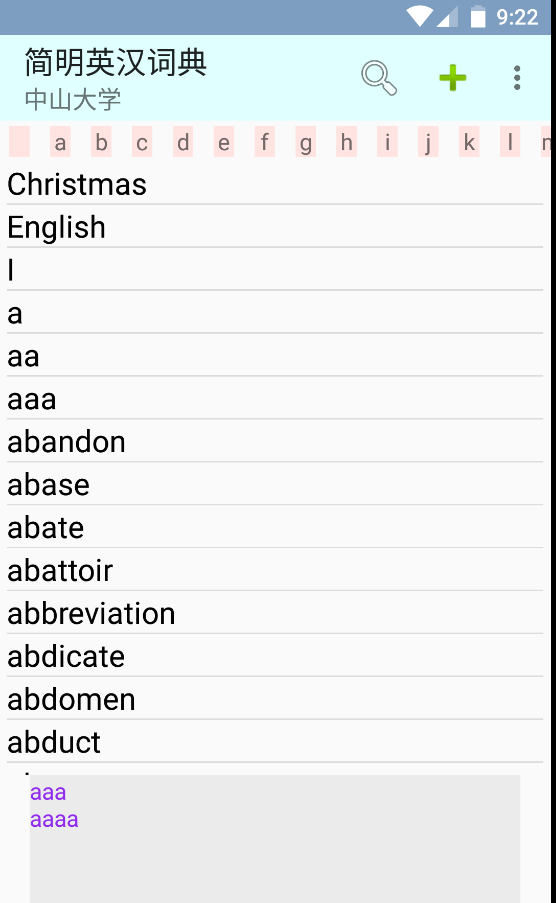
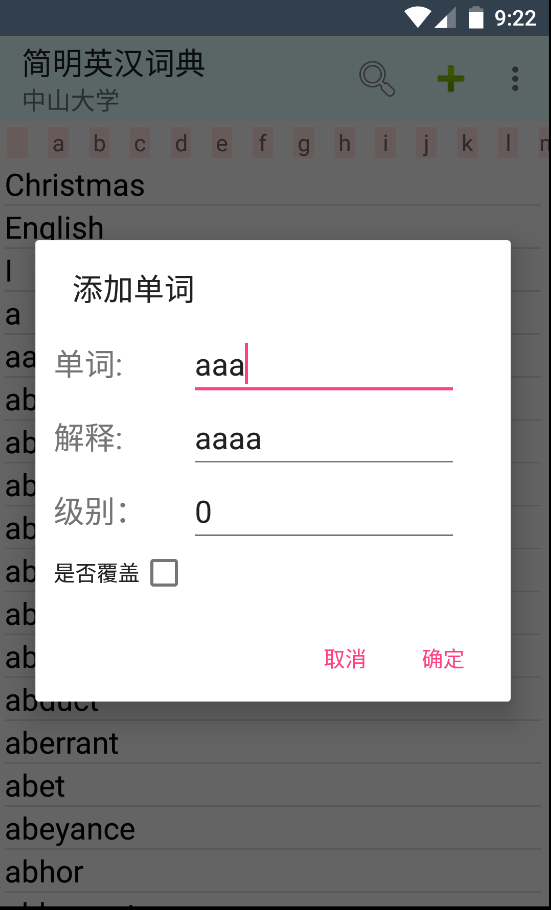
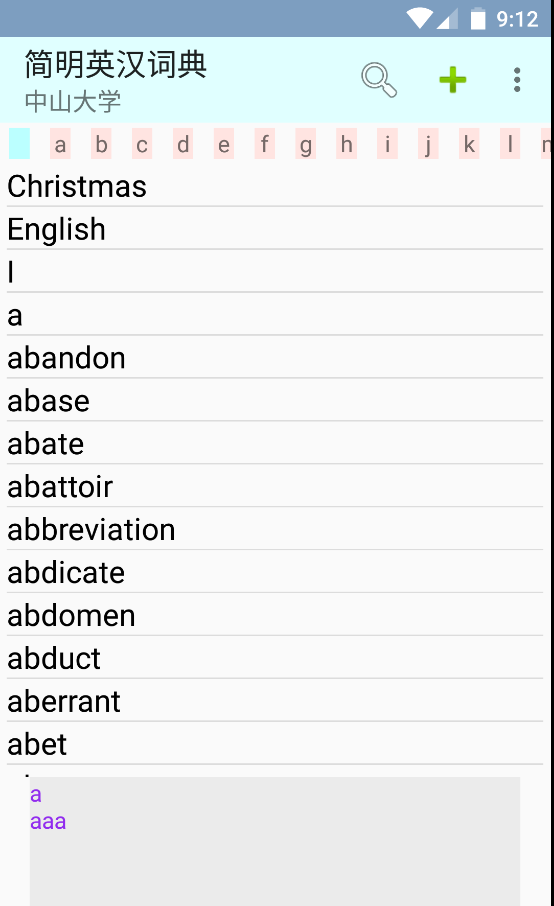
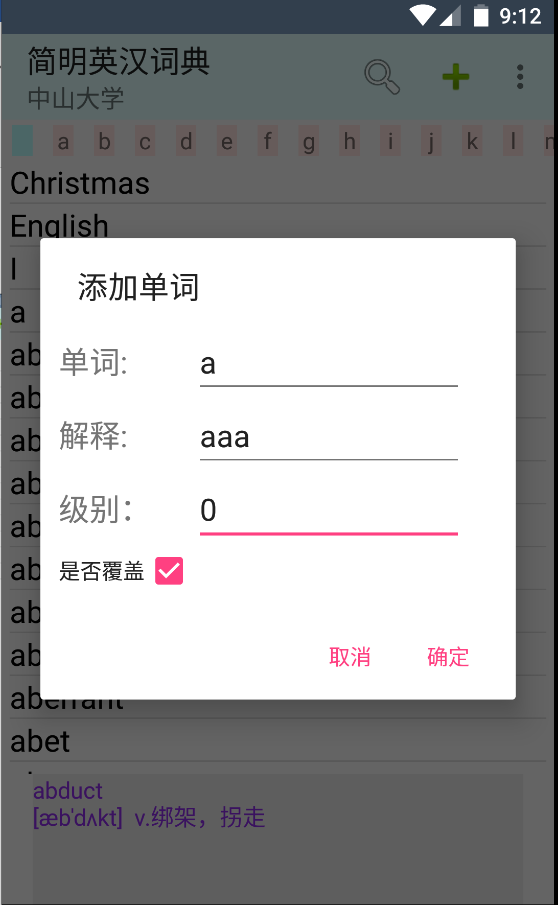
1. 直接显示词义



1. 查找单词

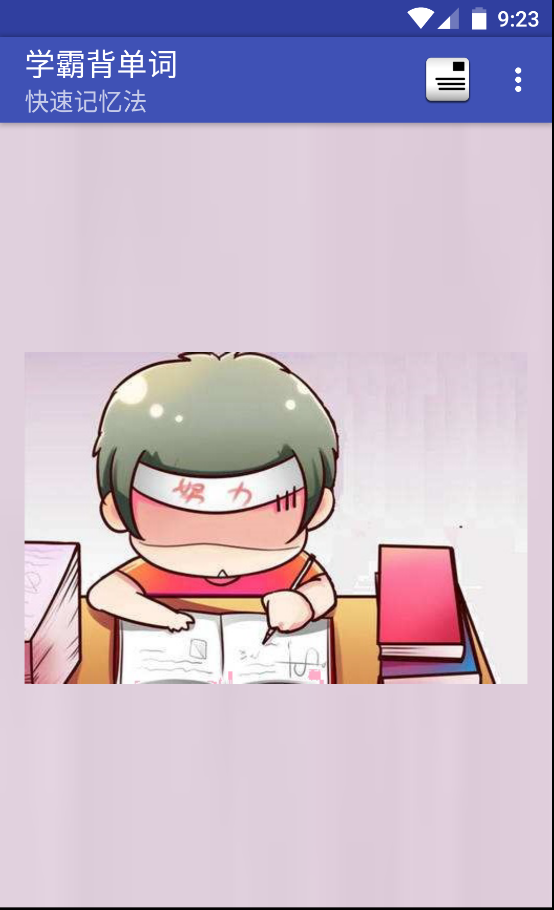


1. 添加单词

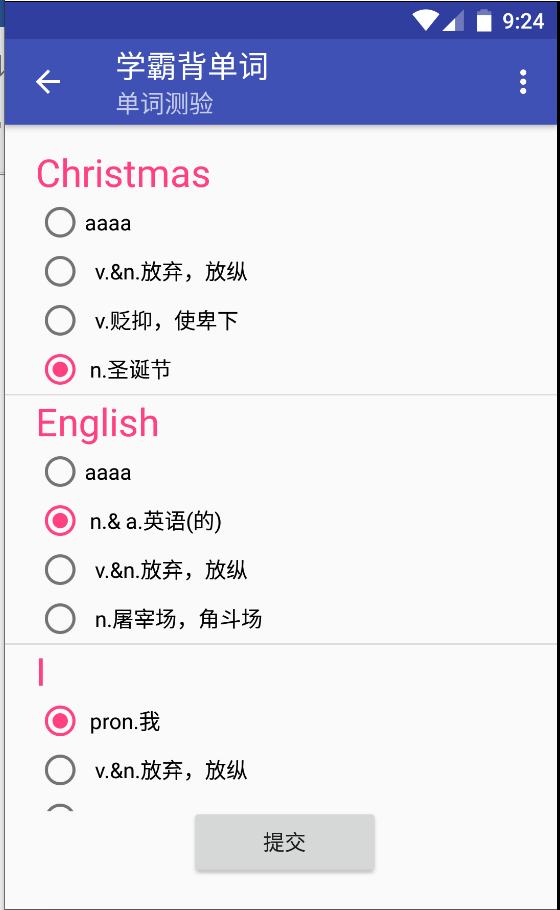


【背单词程序】

1. 主页面



1. 开始测试



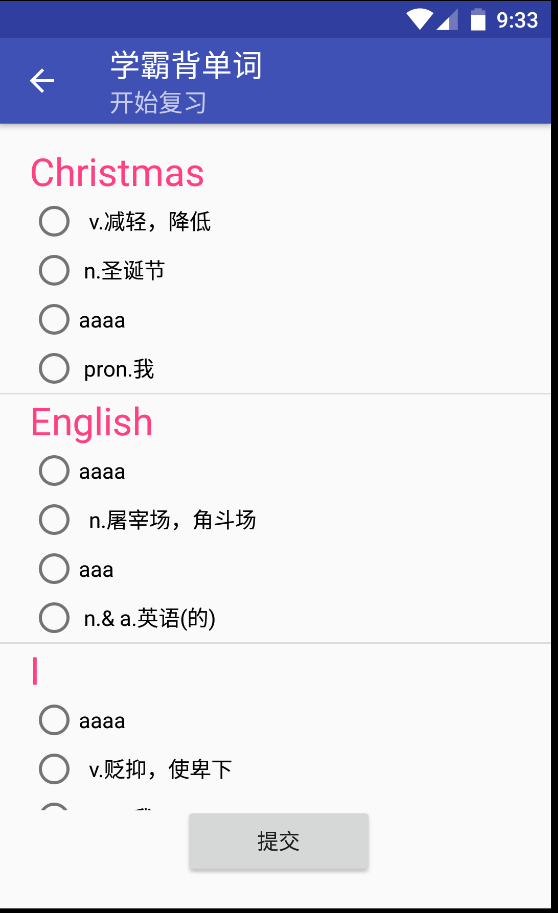
1. 测试完毕，提交



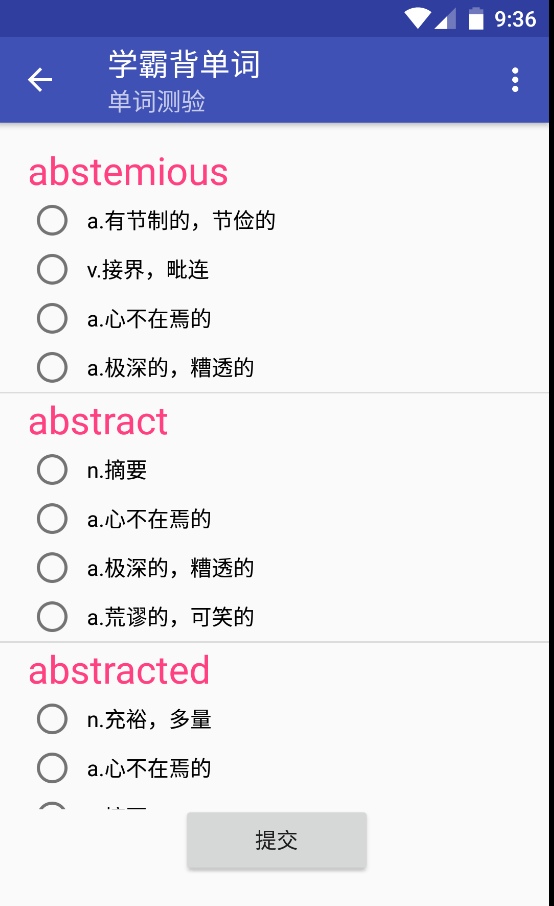
1. 测验统计



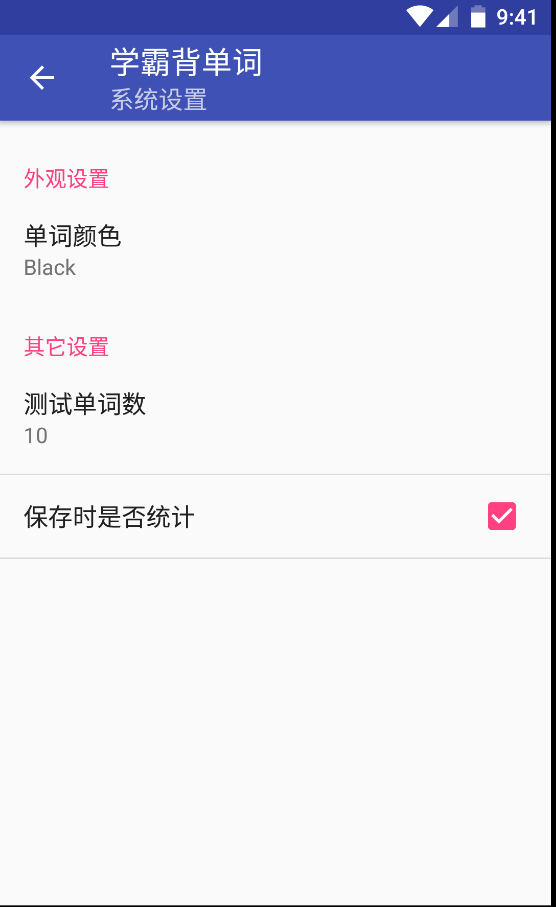
1. 新增单词界面与词典程序一样，故不赘述
2. 复习旧单词



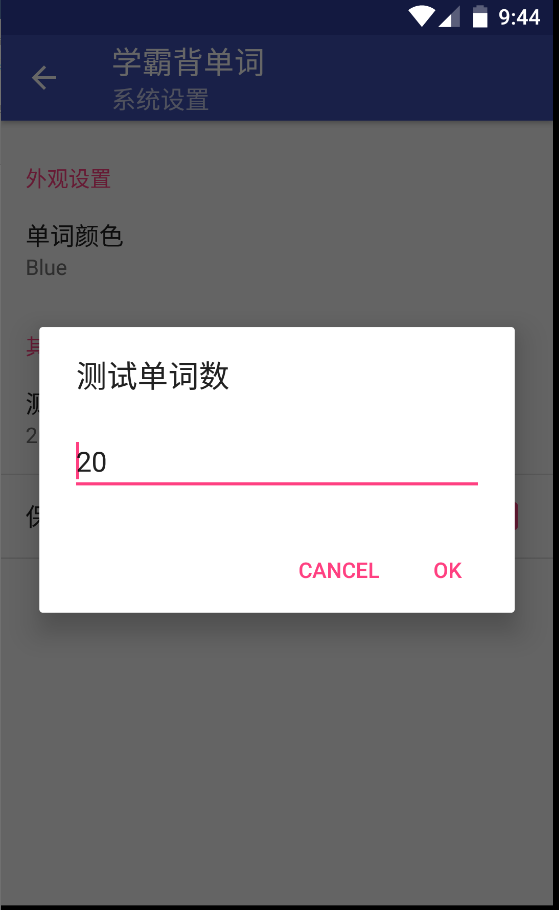
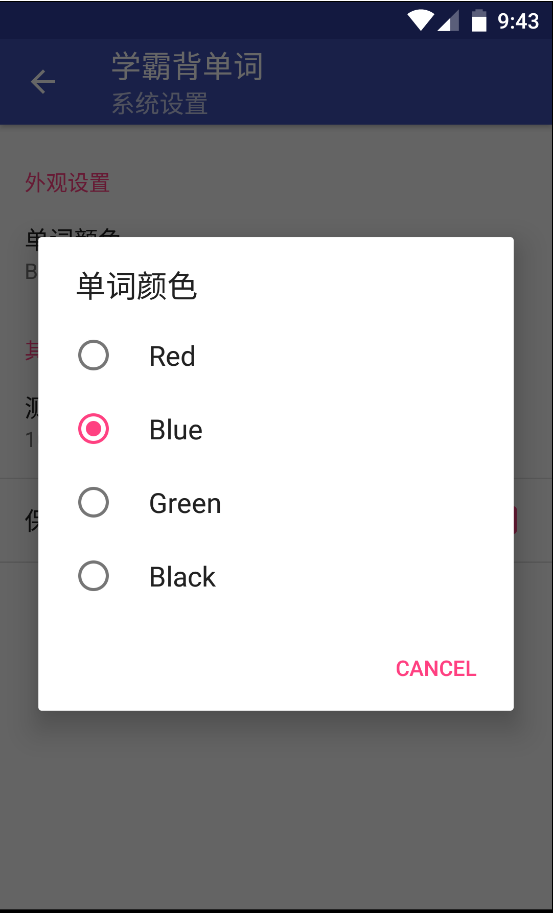
1. 之后的测试都是新的单词

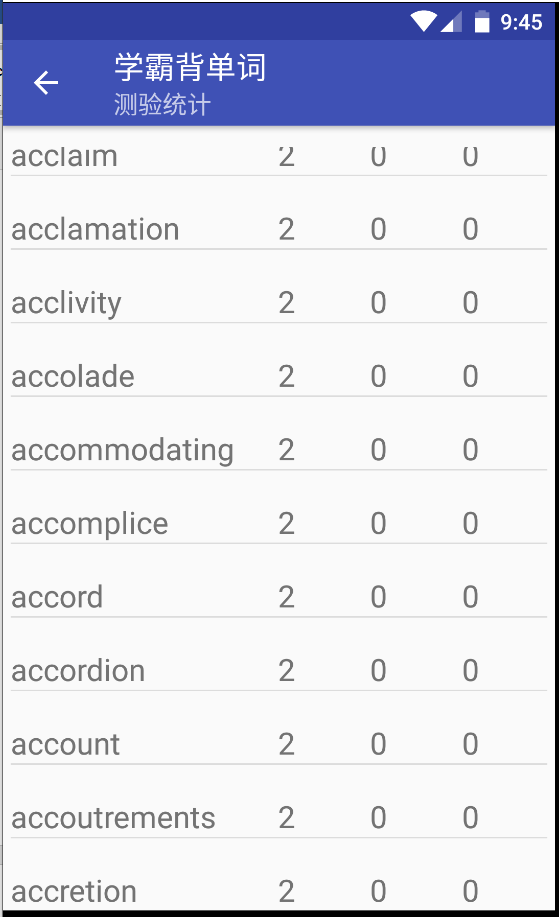


1. 系统设置



1. 参数设置（蓝色字体，20个单词，不统计）





*完成后的主要源码(.java和.xml)：*

【词典程序】

（1）Database.java

**package** com.example.isszym.actionbar;  
  
**import** android.content.Context;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.database.sqlite.SQLiteOpenHelper;  
**import** android.widget.Toast;  
  
*/\*\*  
 \* Created by 64635 on 2019/5/14.  
 \*/***public class** Database **extends** SQLiteOpenHelper {  
 **public static final** String ***CREATE\_DICTIONARY***=**"create table dictionary(word text primary key,explanation text,level integer)"**;  
 **private** Context **mContext**;  
 **public** Database(Context context, String name, SQLiteDatabase.CursorFactory factory,**int** version){  
 **super**(context,name,factory,version);  
 **mContext**=context;  
 }  
 @Override  
 **public void** onCreate(SQLiteDatabase db){  
 db.execSQL(***CREATE\_DICTIONARY***);  
 Toast.*makeText*(**mContext**,**"Create succeed"**,Toast.***LENGTH\_SHORT***).show();  
 }  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db,**int** oldversion,**int** newversion){}  
}

（2）DictProvider.java

**package** com.example.isszym.actionbar;  
  
**import** android.content.ContentProvider;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.net.Uri;  
  
*/\*\*  
 \* Created by 64635 on 2019/5/15.  
 \*/***public class** DictProvider **extends** ContentProvider {  
 **private** Context **mContext**;  
 Database mDbHelper;  
 SQLiteDatabase **db** = **null**;  
 @Override  
 **public boolean** onCreate() {  
 mDbHelper=**new** Database(getContext(),**"dictionary"**,**null**,1);  
 **return true**;  
 }  
  
 @Override  
 **public** String getType(Uri uri){ *// 返回本ContentProvider所提供数据的MIME类型* **return null**;  
 }  
  
 @Override  
 **public** Cursor query(Uri uri, String[] projection, String selection,  
 String[] selectionArgs, String sortOrder) {  
 **db** = mDbHelper.getWritableDatabase();  
 **return db**.query(**"dictionary"**,projection,selection,selectionArgs,**null**,**null**,sortOrder,**null**);  
 }  
  
 @Override  
 **public** Uri insert(Uri uri, ContentValues cv){  
 **db** = mDbHelper.getWritableDatabase();  
 **db**.insert(**"dictionary"**, **null**, cv);  
 **db**.close();  
 **return** uri;  
 }  
  
 **public int** delete(Uri uri, String where, String[] whereArgs) {  
 **db** = mDbHelper.getWritableDatabase();  
 **int** ret = **db**.delete(**"dictionary"**, where, whereArgs);  
 **db**.close();  
 **return** ret;*//影响的记录数* }  
  
 **public int** update(Uri uri, ContentValues cv, String where, String[] whereArgs) {  
 **db** =mDbHelper.getWritableDatabase();  
 **int** ret = **db**.update(**"dictionary"**, cv, where, whereArgs);  
 **db**.close();  
 **return** ret; *//影响的记录数* }  
  
 **private** String sh(Uri uri){  
 **return "\r\n-----"**+uri+**"-----\r\n"**;  
 }  
}

（3）SupportToolbarActivity.java

**package** com.example.isszym.actionbar;  
  
**import** android.app.AlertDialog;  
**import** android.app.ProgressDialog;  
**import** android.content.ContentResolver;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.res.Resources;  
**import** android.database.Cursor;  
**import** android.graphics.Color;  
**import** android.net.Uri;  
**import** android.os.AsyncTask;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Message;  
**import** android.support.v7.app.ActionBar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.DisplayMetrics;  
**import** android.util.Log;  
**import** android.view.Gravity;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.view.WindowManager;  
**import** android.widget.AdapterView;  
**import** android.widget.CheckedTextView;  
**import** android.widget.CursorAdapter;  
**import** android.widget.EditText;  
**import** android.widget.HorizontalScrollView;  
**import** android.widget.LinearLayout;  
**import** android.widget.ListView;  
**import** android.widget.PopupMenu;  
**import** android.widget.SimpleCursorAdapter;  
**import** android.widget.TableLayout;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.google.android.gms.appindexing.Action;  
**import** com.google.android.gms.appindexing.AppIndex;  
**import** com.google.android.gms.appindexing.Thing;  
**import** com.google.android.gms.common.api.GoogleApiClient;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONObject;  
  
**import** java.io.BufferedReader;  
**import** java.io.InputStream;  
**import** java.io.InputStreamReader;  
**import** java.net.URL;  
**import** java.net.URLConnection;  
**import** java.util.ArrayList;  
**import** java.util.Collections;  
**import** java.util.List;  
  
**import static** com.example.isszym.actionbar.R.id.***textView***;  
  
**public class** SupportToolbarActivity **extends** AppCompatActivity {  
 ListView **listview**;  
 **private** Toolbar **mToolbar**;  
 ContentResolver **resolver**;  
 SimpleCursorAdapter **adapter**;  
 SimpleCursorAdapter **adapter2**,**adapter3**;  
 Cursor **cursor**;  
 Cursor **cursor2**,**cursor3**;  
 Uri **uri**;  
 TextView **textViewbottom**;  
 **private** Context **mContext**;  
 String **tmpword**;  
 **private boolean isopen** = **true**;  
 **private** AlertDialog **alertDialog** = **null**;  
 **private** AlertDialog.Builder **dialogBuilder** = **null**;  
 **int flag** = 0, **flag2** = 0;  
 **int nowheight**;  
 CheckedTextView **textView2**;  
 List<TextView> **listguard**;  
 **private** HorizontalScrollView **horizontalScrollView**;  
 **private** LinearLayout **container**;  
 **private** String **alphabet**[] = **new** String[]{**" "**, **"a"**, **"b"**, **"c"**, **"d"**, **"e"**, **"f"**, **"g"**, **"h"**, **"i"**, **"j"**, **"k"**, **"l"**, **"m"**, **"n"**, **"o"**, **"p"**, **"q"**, **"r"**, **"s"**, **"t"**, **"u"**, **"v"**, **"w"**, **"x"**, **"y"**, **"z"**};  
 **private** ArrayList<String> **data** = **new** ArrayList<>();  
 **private** TextView **testTextViewguard**;  
 String **guardword**;  
 ProgressDialog **pd**;  
 **int ifover**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_toolbar***);  
 **mContext** = **this**;  
 *//We have to tell the activity where the toolbar is* **mToolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**mToolbar**);  
 ActionBar actionBar = getSupportActionBar();  
 *//Display home with the "up" arrow indicator* actionBar.setTitle(**"简明英汉词典"**);  
 actionBar.setSubtitle(**"中山大学"**);  
 **uri** = Uri.*parse*(**"content://com.example.providers.firstprovider1/"**);  
  
 setguard();  
  
 **resolver** = getContentResolver();  
 **listview** = (ListView) findViewById(R.id.***listview***);  
 **cursor** = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation"**}, **null**, **null**, **"\_id"**);  
 **cursor2** = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation"**}, **null**, **null**, **"\_id"**);  
 **adapter** = **new** SimpleCursorAdapter(**this**, R.layout.***item***, **cursor**, **new** String[]{**"\_id"**}, **new int**[]{R.id.***word***}, CursorAdapter.***FLAG\_REGISTER\_CONTENT\_OBSERVER***);  
 **adapter2** = **new** SimpleCursorAdapter(**this**, R.layout.***item2***, **cursor2**, **new** String[]{**"\_id"**, **"explanation"**}, **new int**[]{R.id.***word2***, R.id.***subex***}, CursorAdapter.***FLAG\_REGISTER\_CONTENT\_OBSERVER***);  
 **listview**.setAdapter(**adapter**);  
 **textViewbottom** = (TextView) findViewById(***textView***);  
 ViewGroup.LayoutParams params2 = **listview**.getLayoutParams();  
 **final int** heighttmp=params2.**height**;  
 Resources resources = **this**.getResources();  
 DisplayMetrics dm = resources.getDisplayMetrics();  
 **int** height2 = dm.**heightPixels**;  
 params2.**height** = height2 - 20;  
 **listview**.setLayoutParams(params2);  
 **listview**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
  
 @Override  
 **public void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  
 **if** (**flag** == 0) {  
 ViewGroup.LayoutParams params3 = **listview**.getLayoutParams();  
 params3.**height**=heighttmp;  
 **listview**.setLayoutParams(params3);  
 TextView nowclick = (TextView) view.findViewById(R.id.***word***);  
 String tmpword = nowclick.getText().toString();  
 Cursor cursor2 = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **" \_id=?"**, **new** String[]{tmpword}, **null**);  
 cursor2.moveToFirst();  
 **textViewbottom**.setText(tmpword + **"\n"** + cursor2.getString(cursor2.getColumnIndex(**"explanation"**)));  
 }  
 ;  
 }  
 });  
 **listview**.setOnItemLongClickListener(**new** AdapterView.OnItemLongClickListener() {  
 @Override  
 **public boolean** onItemLongClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  
 PopupMenu popup = **new** PopupMenu(SupportToolbarActivity.**this**, view);  
 popup.getMenuInflater().inflate(R.menu.***menu\_pop***, popup.getMenu());  
 TextView nowclick;  
 **if** (**flag** == 0) {  
 nowclick = (TextView) view.findViewById(R.id.***word***);  
 } **else** {  
 nowclick = (TextView) view.findViewById(R.id.***word2***);  
 }  
 **tmpword** = nowclick.getText().toString();  
 popup.setOnMenuItemClickListener(**new** PopupMenu.OnMenuItemClickListener() {  
 @Override  
 **public boolean** onMenuItemClick(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** R.id.***delete***:  
 simpledelete();  
 **break**;  
 **case** R.id.***correct***:  
 simplecorrect();  
 **break**;  
 }  
 **return true**;  
 }  
 });  
 popup.show();  
 **return true**;  
 }  
 });  
 }  
  
 Handler **handler** = **new** Handler() {  
 @Override  
 **public void** handleMessage(Message message) {  
 **if** (message.**what** == 0)  
 {  
 Log.*v*(**"lll"**,String.*valueOf*(**ifover**));  
 **pd**.setProgress(**ifover**);}  
 **else if** (message.**what** == 1) {  
 **new** RefreshList().execute();  
 }  
 }  
 };  
 **public void** show(**int** max\_length) {  
 **pd** = **new** ProgressDialog(SupportToolbarActivity.**this**);  
 **pd**.setMax(max\_length);  
 **pd**.setTitle(**"下载单词"**);  
 **pd**.setCancelable(**false**);  
 **pd**.setProgressStyle(**pd**.***STYLE\_HORIZONTAL***);  
 **pd**.setIndeterminate(**false**);  
 **pd**.show();  
  
 }  
 **public void** setguard() {  
 Collections.*addAll*(**data**, **alphabet**);  
 **horizontalScrollView** = (HorizontalScrollView) findViewById(R.id.***horizontalScrollView***);  
 **container** = (LinearLayout) findViewById(R.id.***horizontalScrollViewItemContainer***);  
 LinearLayout.LayoutParams layoutParams = **new** LinearLayout.LayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***,  
 ViewGroup.LayoutParams.***WRAP\_CONTENT***);  
 layoutParams.**gravity** = Gravity.***CENTER***;  
 layoutParams.setMargins(20, 10, 20, 10);  
 **for** (**int** i = 0; i < **data**.size(); i++) {  
 TextView textView = **new** TextView(**this**);  
 textView.setText(**data**.get(i));  
 textView.setTextSize(15);  
 textView.setWidth(40);  
 textView.setGravity(Gravity.***CENTER***);  
 textView.setBackgroundColor(Color.*rgb*(255, 228, 225));  
 textView.setLayoutParams(layoutParams);  
  
 **final int** tmp = i;  
 textView.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 view.setBackgroundColor(Color.*rgb*(187, 255, 255));  
 **for** (**int** j = 0; j < **container**.getChildCount(); j++) {  
 **if** (tmp == j) **continue**;  
 **container**.getChildAt(j).setBackgroundColor(Color.*rgb*(255, 228, 225));  
 }  
 **guardword** = **data**.get(tmp);  
 **if** (**guardword**.equals(**" "**)) {  
 **if** (**flag** == 0)  
 **new** RefreshList().execute();  
 **else  
 new** RefreshListexp().execute();  
 }  
 **if** (!**guardword**.equals(**" "**)) {  
 **if** (**flag** == 0)  
 **new** RefreshListguard().execute();  
 **else  
 new** RefreshListguardexp().execute();  
 }  
 }  
 });  
 **container**.addView(textView);  
 **container**.invalidate();  
 }  
 }  
  
 **private class** RefreshList **extends** AsyncTask<Void, Void, Cursor> {  
 **protected** Cursor doInBackground(Void... params) {  
 Cursor newCursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **null**, **null**, **" \_id"**);  
 **return** newCursor;  
 }  
  
 **protected void** onPostExecute(Cursor newCursor) {  
 **adapter**.changeCursor(newCursor);  
 **cursor**.close();  
 **cursor** = newCursor;  
 }  
 }  
  
 **private class** RefreshListguard **extends** AsyncTask<Void, Void, Cursor> {  
 **protected** Cursor doInBackground(Void... params) {  
 Cursor newCursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **"word like ?"**, **new** String[]{**guardword** + **"%"**}, **" \_id"**);  
 **return** newCursor;  
 }  
  
 **protected void** onPostExecute(Cursor newCursor) {  
 **adapter**.changeCursor(newCursor);  
 **cursor**.close();  
 **cursor** = newCursor;  
 }  
 }  
  
 **private class** RefreshListexp **extends** AsyncTask<Void, Void, Cursor> {  
 **protected** Cursor doInBackground(Void... params) {  
 Cursor newCursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **null**, **null**, **" \_id"**);  
 **return** newCursor;  
 }  
  
 **protected void** onPostExecute(Cursor newCursor) {  
 **adapter2**.changeCursor(newCursor);  
 **cursor2**.close();  
 **cursor2** = newCursor;  
 }  
 }  
  
 **private class** RefreshListguardexp **extends** AsyncTask<Void, Void, Cursor> {  
 **protected** Cursor doInBackground(Void... params) {  
 Cursor newCursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **"word like ?"**, **new** String[]{**guardword** + **"%"**}, **" \_id"**);  
 **return** newCursor;  
 }  
 **protected void** onPostExecute(Cursor newCursor) {  
 **adapter2**.changeCursor(newCursor);  
 **cursor2**.close();  
 **cursor2** = newCursor;  
 }  
 }  
  
 **private class** RefreshListlookfor **extends** AsyncTask<Void, Void, Cursor> {  
 **protected** Cursor doInBackground(Void... params) {  
 Cursor newCursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **"word like ?"**, **new** String[]{**"%"**+**guardword** + **"%"**}, **" \_id"**);  
 **return** newCursor;  
 }  
 **protected void** onPostExecute(Cursor newCursor) {  
 **adapter**.changeCursor(newCursor);  
 **cursor**.close();  
 **cursor** = newCursor;  
 }  
 }  
 **private class** RefreshListexplookfor **extends** AsyncTask<Void, Void, Cursor> {  
 **protected** Cursor doInBackground(Void... params) {  
 Cursor newCursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id,explanation,level"**}, **"word like ?"**, **new** String[]{**"%"**+**guardword** + **"%"**}, **" \_id"**);  
  
 **return** newCursor;  
 }  
 **protected void** onPostExecute(Cursor newCursor) {  
 **adapter2**.changeCursor(newCursor);  
 **cursor2**.close();  
 **cursor2** = newCursor;  
 }  
 }  
  
 **public void** simpledelete() {  
 **dialogBuilder** = **new** AlertDialog.Builder(**mContext**);  
 **alertDialog** = **dialogBuilder** .setTitle(**"删除单词"**)  
 .setMessage(**"是否确定要删除该单词？"**)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(**mContext**, **"取消删除"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **resolver**.delete(**uri**, **"word=?"**, **new** String[]{**tmpword**});  
 **new** RefreshList().execute();  
 **new** RefreshListexp().execute();  
 Toast.*makeText*(**mContext**, **"删除成功"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .create(); *// 创建AlertDialog对象* **alertDialog**.show(); *// 显示对话框* }  
  
 **public void** simplecorrect() {  
 TableLayout wordcorrect = (TableLayout) getLayoutInflater()  
 .inflate(R.layout.***correctword***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(**mContext**);  
 **alertDialog** = **dialogBuilder** *// 设置对话框标题* .setTitle(**"修改单词"**)  
 *// 设置对话框显示的View对象* .setView(wordcorrect)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(**mContext**, **"取消修改"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 EditText wordex = (EditText) **alertDialog**.findViewById(R.id.***word\_exp***);  
 EditText level = (EditText) **alertDialog**.findViewById(R.id.***level***);  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(**"word"**, **tmpword**);  
 contentValues.put(**"explanation"**, String.*valueOf*(wordex.getText()));  
 contentValues.put(**"level"**, Integer.*parseInt*(String.*valueOf*(level.getText())));  
 **resolver**.update(**uri**, contentValues, **"word=?"**, **new** String[]{**tmpword**});  
 **new** RefreshList().execute();  
 **new** RefreshListexp().execute();  
 Toast.*makeText*(**mContext**, **"修改成功"**, Toast.***LENGTH\_SHORT***).show();  
 **textViewbottom**.setText(**tmpword** + **"\n"** + String.*valueOf*(wordex.getText()));  
 }  
 })  
 .create(); *// 创建AlertDialog对象* **alertDialog**.show();  
 }  
  
 **public void** lookforword(){  
 TableLayout wordfind = (TableLayout) getLayoutInflater()  
 .inflate(R.layout.***lookfor***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(**mContext**);  
 **alertDialog** = **dialogBuilder** *// 设置对话框标题* .setTitle(**"查询单词"**)  
 *// 设置对话框显示的View对象* .setView(wordfind)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(**mContext**, **"取消查询"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 EditText wordex = (EditText) **alertDialog**.findViewById(R.id.***lookforword***);  
 **guardword**=String.*valueOf*(wordex.getText());  
 Log.*v*(**"lll"**,**guardword**);  
 **new** RefreshListlookfor().execute();  
 **new** RefreshListexplookfor().execute();  
 Toast.*makeText*(**mContext**, **"查询成功"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .create(); *// 创建AlertDialog对象* **alertDialog**.show();  
 }  
  
 **public void** addword() {  
 TableLayout wordadd = (TableLayout) getLayoutInflater()  
 .inflate(R.layout.***addword***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(**mContext**);  
 **alertDialog** = **dialogBuilder** *// 设置对话框标题* .setTitle(**"添加单词"**)  
 *// 设置对话框显示的View对象* .setView(wordadd)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(**mContext**, **"取消添加"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 EditText newword = (EditText) **alertDialog**.findViewById(R.id.***new\_word***);  
 EditText wordex = (EditText) **alertDialog**.findViewById(R.id.***word\_exp***);  
 EditText level = (EditText) **alertDialog**.findViewById(R.id.***level***);  
 Cursor cursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id"**}, **" \_id=?"**, **new** String[]{String.*valueOf*(newword.getText())}, **null**);  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(**"word"**, String.*valueOf*(newword.getText()));  
 contentValues.put(**"explanation"**, String.*valueOf*(wordex.getText()));  
 contentValues.put(**"level"**, Integer.*parseInt*(String.*valueOf*(level.getText())));  
 **if** (cursor.getCount() == 0) {  
 **resolver**.insert(**uri**, contentValues);  
 **new** RefreshList().execute();  
 **new** RefreshListexp().execute();  
 Toast.*makeText*(**mContext**, **"添加成功"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 **textView2** = (CheckedTextView) **alertDialog**.findViewById(R.id.***cb1***);  
 **if** (**textView2**.isChecked()) {  
 **resolver**.update(**uri**, contentValues, **"word=?"**, **new** String[]{String.*valueOf*(newword.getText())});  
 Toast.*makeText*(**mContext**, **"覆盖成功"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 Toast.*makeText*(**mContext**, **"单词已存在"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }  
 })  
 .create(); *// 创建AlertDialog对象* **if** (!isFinishing()) {  
 **alertDialog**.show();  
 }  
 }  
  
 @Override  
 **protected void** onPostCreate(Bundle savedInstanceState) {  
 **super**.onPostCreate(savedInstanceState);  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***support***, menu);  
 **return true**;  
 }  
  
 **public static** WordRec parserJSON(String str) {  
 JSONObject jsonObject = **null**;  
 **try** {  
 jsonObject = **new** JSONObject(str);  
  
 } **catch** (Exception ex) {  
 ex.printStackTrace();  
 }  
 WordRec wordRec = **new** WordRec();  
 wordRec.setWord(jsonObject.optString(**"word"**));  
 wordRec.setExplanation(jsonObject.optString(**"explanation"**));  
 wordRec.setLevel(Integer.*parseInt*(jsonObject.optString(**"level"**)));  
 **return** wordRec;  
 }  
 **public void** clicker(View v) {  
 CheckedTextView checkedTextView = (CheckedTextView) v;  
 checkedTextView.toggle();  
 }  
 **public void** getdata() {  
 show(100);  
 **new** Thread() {  
 @Override  
 **public void** run() {  
 **try** {  
 Log.*d*(**"out:"**, **"downloading..."**);  
 String tmp = **"http://172.18.187.9:8080/dict/"**;  
 URL url = **new** URL(tmp);  
 URLConnection connection = url.openConnection();  
 InputStream inputStream = connection.getInputStream();  
 InputStreamReader inputStreamReader = **new** InputStreamReader(inputStream);  
 BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);  
 StringBuilder stringBuilder = **new** StringBuilder();  
 String line;  
 **while** ((line = bufferedReader.readLine()) != **null**) {  
 stringBuilder.append(line);  
 }  
 String str = stringBuilder.toString();  
 JSONArray jsonArray = **new** JSONArray(str);  
 **for** (**int** i = 0; i < jsonArray.length(); ++i) {  
 WordRec wordRec = *parserJSON*(jsonArray.get(i).toString());  
 Log.*v*(**"out:"**, wordRec.toString());  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(**"word"**, wordRec.getWord());  
 contentValues.put(**"explanation"**, wordRec.getExplanation());  
 contentValues.put(**"level"**, wordRec.getLevel());  
 **resolver**.insert(**uri**, contentValues);  
 **ifover** = 100 \* (i + 1) / jsonArray.length();  
 **handler**.sendEmptyMessage(0);  
 }  
 **pd**.dismiss();  
 **handler**.sendEmptyMessage(1);  
 } **catch** (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
 }.start();  
  
  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 ViewGroup.LayoutParams params = **listview**.getLayoutParams();  
  
 **switch** (item.getItemId()) {  
 **case** R.id.***addword***:  
 addword();  
 **break**;  
 **case** R.id.***lookfor***:  
 lookforword();  
 **break**;  
 **case** R.id.***menu1***:  
 getdata();  
 **break**;  
 **case** R.id.***menu2***:  
 **if** (item.isChecked()) {  
 **flag** = 0;  
 **listview**.setAdapter(**adapter**);  
 params.**height** = **nowheight**;  
 **listview**.setLayoutParams(params);  
 item.setChecked(**false**);  
 } **else** {  
 **flag** = 1;  
 **listview**.setAdapter(**adapter2**);  
 **nowheight** = params.**height**;  
 Resources resources = **this**.getResources();  
 DisplayMetrics dm = resources.getDisplayMetrics();  
 **int** height2 = dm.**heightPixels**;  
 params.**height** = height2 - 20;  
 **listview**.setLayoutParams(params);  
 **textViewbottom**.setEnabled(**false**);  
 item.setChecked(**true**);  
 **new** RefreshList().execute();  
 }  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
 **return true**;  
 }  
}

(4) WordRec.java(来自老师的类设计)

**package** com.example.isszym.actionbar;  
  
*/\*\*  
 \* Created by 64635 on 2019/5/15.  
 \*/***public class** WordRec {  
 **private** String **word**;  
 **private** String **explanation**;  
 **private int level**;  
 WordRec(){}  
 WordRec(String word,String explanation,**int** level){  
 **this**.**word**=word;  
 **this**.**explanation**=explanation;  
 **this**.**level**=level;  
 }  
 **public** String getWord(){**return word**;}  
 **public void** setWord(String word){**this**.**word**=word;}  
 **public** String getExplanation(){**return explanation**;}  
 **public void** setExplanation(String explanation){**this**.**explanation**=explanation;}  
 **public int** getLevel(){**return level**;}  
 **public void** setLevel(**int** level){**this**.**level**=level;}  
  
}

（5）activity\_toolbar.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:minHeight="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:theme="@style/ThemeOverlay.AppCompat.Light"**/>  
 <**HorizontalScrollView  
 android:id="@+id/horizontalScrollView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:scrollbars="none"** >  
  
 <**LinearLayout  
 android:id="@+id/horizontalScrollViewItemContainer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"** />  
 </**HorizontalScrollView**>  
  
  
  
  
 <**ListView  
 android:layout\_width="350dp"  
 android:layout\_gravity="center"  
 android:layout\_height="400dp"  
 android:id="@+id/listview"** />  
  
 <**TextView  
 android:layout\_width="320dp"  
 android:layout\_height="100dp"  
 android:textSize="15dp"  
 android:layout\_gravity="center"  
 android:textColor="@color/colorexp"  
 android:background="@color/colorbg"  
 android:id="@+id/textView"** />  
  
</**LinearLayout**>

(6)addword.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/studentForm"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="12dp"**>  
 <**TableRow**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="单词:"  
 android:textSize="20dp"**/>  
 <**EditText  
 android:id="@+id/new\_word"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="请填写单词"  
 android:textSize="20dp"  
 android:selectAllOnFocus="true"**/>  
 </**TableRow**>  
 <**TableRow**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="解释:"  
 android:textSize="20dp"**/>  
 <**EditText  
 android:id="@+id/word\_exp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:hint="请填写解释"**/>  
 </**TableRow**>  
 <**TableRow**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="级别："  
 android:textSize="20dp"**/>  
 <**EditText  
 android:id="@+id/level"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="请填写级别（0-8）"  
 android:textSize="20dp"  
 android:inputType="phone"  
 android:selectAllOnFocus="true"** />  
 </**TableRow**>  
 <**TableRow**>  
 <**CheckedTextView  
 android:id="@+id/cb1"  
 android:text="是否覆盖"  
 android:checkMark="?android:attr/listChoiceIndicatorMultiple"  
 android:gravity="center"  
 android:checked="true"  
 android:onClick="clicker"** />  
 </**TableRow**>  
</**TableLayout**>

(7)correctword.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/studentForm"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="12dp"**>  
  
 <**TableRow**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="解释:"  
 android:textSize="20dp"**/>  
 <**EditText  
 android:id="@+id/word\_exp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:hint="请填写解释"**/>  
 </**TableRow**>  
 <**TableRow**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="级别："  
 android:textSize="20dp"**/>  
 <**EditText  
 android:id="@+id/level"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="请填写级别（0-8）"  
 android:textSize="20dp"  
 android:inputType="phone"  
 android:selectAllOnFocus="true"** />  
 </**TableRow**>  
  
</**TableLayout**>

(8)item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:textSize="20dp"  
 android:id="@+id/word"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorblack"** />  
</**LinearLayout**>

(9)item2.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:textSize="20dp"  
 android:id="@+id/word2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorgraydark"** />  
 <**TextView  
 android:textSize="15dp"  
 android:id="@+id/subex"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorgray"** />  
</**LinearLayout**>

(10)item3.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:textSize="20dp"  
 android:id="@+id/word3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/colorblack"** />  
</**LinearLayout**>

(11)lookfor.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/studentForm"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="12dp"**>  
  
 <**TableRow**>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="单词:"  
 android:textSize="20dp"**/>  
 <**EditText  
 android:id="@+id/lookforword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="20dp"  
 android:hint="请填写单词"**/>  
 </**TableRow**>  
</**TableLayout**>

【背单词程序】

（1）AnalysisActivity.java

**package** com.example.a64635.grandwordremember;  
  
**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.design.widget.Snackbar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
  
  
**import** android.os.Bundle;  
**import** android.support.design.widget.FloatingActionButton;  
**import** android.support.design.widget.Snackbar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.View;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.ListView;  
**import** android.widget.SimpleAdapter;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.Map;  
  
**import static** android.R.id.***list***;  
**import static** com.example.a64635.grandwordremember.R.id.***listview***;  
  
**public class** AnalysisActivity **extends** AppCompatActivity {  
 **private** ListView **listView**;  
 ArrayList<Map<String, String>> **list**;  
 SimpleAdapter **adapter**;  
 SQLiteDatabase **db**;  
 Database **dbHelper**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_analysis***);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 toolbar.setTitle(**"学霸背单词"**);  
 toolbar.setSubtitle(**"测验统计"**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);*//左侧添加一个默认的返回图标* getSupportActionBar().setHomeButtonEnabled(**true**); *//设置返回键可用* **listView**=(ListView) findViewById(R.id.***listviewanalysis***);  
 **dbHelper** = **new** Database(**this**, **"rememberword.db"**, **null**, 1);  
 **dbHelper**.getWritableDatabase();  
 **db** = **dbHelper**.getWritableDatabase();  
 Cursor cursor = **db**.query(**"rememberwords"**, **null**, **null**, **null**, **null**, **null**, **null**);  
 **list**= **new** ArrayList<Map<String, String>>();  
 Map<String, String> map1 = **new** HashMap<String,String>();  
 map1.put(**"word"**,**"单词"**);  
 map1.put(**"level"**,**"级别"**);  
 map1.put(**"text"**,**"测试"**);  
 map1.put(**"right"**,**"正确"**);  
 **list**.add(map1);  
 **if**(cursor!=**null**&&cursor.moveToFirst()){  
 **do**{  
 Map<String, String> map = **new** HashMap<String,String>();  
 map.put(**"word"**,cursor.getString(cursor.getColumnIndex(**"word"**)));  
 map.put(**"level"**,cursor.getString(cursor.getColumnIndex(**"level"**)));  
 map.put(**"text"**,cursor.getString(cursor.getColumnIndex(**"test\_count"**)));  
 map.put(**"right"**,cursor.getString(cursor.getColumnIndex(**"correct\_count"**)));  
 **list**.add(map);  
 }**while**(cursor.moveToNext());  
 }  
 **adapter** = **new** SimpleAdapter(**this**, **list**,  
 R.layout.***ana\_item***, **new** String[] { **"word"**, **"level"**, **"text"**,**"right"** },  
 **new int**[] { R.id.***anaword***, R.id.***analevel***, R.id.***anatext***,R.id.***anaright*** });  
 **listView**.setAdapter(**adapter**);  
  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***menu\_analysis***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* **int** id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement* **if** (id == android.R.id.***home***) {  
 **this**.finish();  
 }  
 **else super**.onOptionsItemSelected(item);  
 **return true**;  
 }  
}

（2）Database.java 与词典的数据库类同，故不赘述。

（3）MainActivity.java

**package** com.example.a64635.grandwordremember;  
  
**import** android.app.AlertDialog;  
**import** android.content.ContentResolver;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.database.Cursor;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.View;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.CheckedTextView;  
**import** android.widget.EditText;  
**import** android.widget.TableLayout;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 Toolbar **toolbar**;  
 **private** Context **mContext**;  
 **private boolean**[] **checkItems**;  
 Uri **uri**,**uri2**;  
 ContentResolver **resolver**;  
 **private** AlertDialog **alertDialog** = **null**;  
 **private** AlertDialog.Builder **dialogBuilder** = **null**;  
 CheckedTextView **textView**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**toolbar**);  
 **toolbar**.setTitle(**"学霸背单词"**);  
 **toolbar**.setSubtitle(**"快速记忆法"**);  
 **resolver** = getContentResolver();  
 **uri** = Uri.*parse*(**"content://com.example.providers.firstprovider1/"**);  
 **mContext** = **this**;  
  
  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***menu\_main***, menu);  
 **return true**;  
 }  
  
 **public void** ifupdate(ContentResolver resolver, Context mContext, ContentValues contentValues, **int** i, EditText newword) {  
 **if** (i == 0) {  
 Toast.*makeText*(mContext, **"单词已存在"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 resolver.update(**uri**, contentValues, **"word=?"**, **new** String[]{String.*valueOf*(newword.getText())});  
 Toast.*makeText*(mContext, **"覆盖成功"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 **public void** customView() {  
 TableLayout wordadd = (TableLayout) getLayoutInflater()  
 .inflate(R.layout.***addword***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(**mContext**);  
 **alertDialog** = **dialogBuilder** *// 设置对话框标题* .setTitle(**"添加单词"**)  
 *// 设置对话框显示的View对象* .setView(wordadd)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(**mContext**, **"取消添加"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 EditText newword = (EditText) **alertDialog**.findViewById(R.id.***new\_word***);  
 EditText wordex = (EditText) **alertDialog**.findViewById(R.id.***word\_exp***);  
 EditText level = (EditText) **alertDialog**.findViewById(R.id.***level***);  
 Cursor cursor = **resolver**.query(**uri**, **new** String[]{**"word as \_id"**}, **" \_id=?"**, **new** String[]{String.*valueOf*(newword.getText())}, **null**);  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(**"word"**, String.*valueOf*(newword.getText()));  
 contentValues.put(**"explanation"**, String.*valueOf*(wordex.getText()));  
 contentValues.put(**"level"**, Integer.*parseInt*(String.*valueOf*(level.getText())));  
 **if** (cursor.getCount() == 0) {  
 **resolver**.insert(**uri**, contentValues);  
 Toast.*makeText*(**mContext**, **"添加成功"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 **textView** = (CheckedTextView) **alertDialog**.findViewById(R.id.***cb***);  
 **if** (**textView**.isChecked()) {  
 **resolver**.update(**uri**, contentValues, **"word=?"**, **new** String[]{String.*valueOf*(newword.getText())});  
 Toast.*makeText*(**mContext**, **"覆盖成功"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
 Toast.*makeText*(**mContext**, **"单词已存在"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }  
 })  
 .create(); *// 创建AlertDialog对象* **if** (!isFinishing()) {  
 **alertDialog**.show();  
 }  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* **int** id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement* **if** (id == R.id.***entertest***) {  
 Intent intent = **new** Intent(**this**, SecondActivity.**class**);  
 startActivity(intent);  
 **return true**;  
 } **else if** (id == R.id.***add\_word***) {  
 customView();  
 }  
 **else if** (id==R.id.***stat***){  
 Intent intent=**new** Intent(**this**,AnalysisActivity.**class**);  
 startActivity(intent);  
 }  
 **else if** (id==R.id.***set***){  
 Intent intent=**new** Intent(**this**,PreferenceActivity.**class**);  
 startActivity(intent);  
 }  
 **else if** (id==R.id.***find\_word***){  
  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **public void** clicker(View v) {  
 CheckedTextView checkedTextView = (CheckedTextView) v;  
 checkedTextView.toggle();  
 }  
}

(4)SecondActivity.java

**package** com.example.a64635.grandwordremember;  
  
**import** android.content.ContentResolver;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.preference.PreferenceManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.CursorAdapter;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
**import** android.widget.SimpleCursorAdapter;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.util.ArrayList;  
**import** java.util.Map;  
  
**import static** com.example.a64635.grandwordremember.R.id.***listview***;  
  
  
*/\*\*  
 \* Created by 64635 on 2019/5/15.  
 \*/***public class** SecondActivity **extends** AppCompatActivity {  
 Toolbar **toolbar**;  
 **public** Database **dbHelper**;  
 ContentResolver **resolver**;  
 Myadapter **adapter**;  
 ListView **listview**;  
 Uri **uri\_user**;  
 **private** Context **mContext**;  
 **int flag**=0;  
 **public int color**=0;  
 SQLiteDatabase **db**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_second***);  
 **toolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**toolbar**);  
 **toolbar**.setTitle(**"学霸背单词"**);  
 **toolbar**.setSubtitle(**"单词测验"**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);*//左侧添加一个默认的返回图标* getSupportActionBar().setHomeButtonEnabled(**true**); *//设置返回键可用* **final** SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(**this**);  
 String urii = **"content://com.example.providers.firstprovider1/"**;  
 **mContext**=**this**;  
 **dbHelper** = **new** Database(**this**, **"rememberword.db"**, **null**, 1);  
 **dbHelper**.getWritableDatabase();  
 **uri\_user** = Uri.*parse*(urii);  
 **resolver** = getContentResolver();  
 **listview** = (ListView) findViewById(R.id.***listview***);  
 Cursor cursor = **resolver**.query(**uri\_user**, **new** String[]{**"word as \_id,explanation,level"**}, **null**, **null**, **"\_id"**);  
 **final** ArrayList mylist = **new** ArrayList<WordRec>();  
 **db** = **dbHelper**.getWritableDatabase();  
 Cursor cursor2 = **db**.query(**"rememberwords"**, **null**, **null**, **null**, **null**, **null**, **null**);  
 **if** (cursor2.getCount() != 0) {  
 cursor2.moveToLast();  
 String nowword = cursor2.getString(cursor2.getColumnIndex(**"word"**));  
 Log.*v*(**"lll"**, nowword);  
 **int** i = 0;  
 ContentValues values = **new** ContentValues();  
 **for** (cursor.moveToFirst(); !cursor.isAfterLast(); cursor.moveToNext()) {  
 **if** (cursor.getString(cursor.getColumnIndex(**"\_id"**)).compareTo(nowword) <= 0)  
 **continue**;  
 WordRec tmp = **new** WordRec();  
 tmp.setWord(cursor.getString(cursor.getColumnIndex(**"\_id"**)));  
 tmp.setExplanation(cursor.getString(cursor.getColumnIndex(**"explanation"**)));  
 tmp.setLevel(cursor.getColumnIndex(**"level"**));  
 mylist.add(tmp);  
 String num2=**""**;  
 **int** num =Integer.*parseInt*(settings.getString(**"edittext\_key"**, num2)) ;  
 Cursor cursor3 = **db**.query(**"rememberwords"**, **new** String[]{**"word as \_id"**}, **"\_id = ?"**, **new** String[]{tmp.getWord()}, **null**, **null**, **null**);  
 **if** (cursor3.getCount() == 0) {  
 values.put(**"word"**, tmp.getWord());  
 values.put(**"explanation"**, tmp.getExplanation());  
 values.put(**"level"**, tmp.getLevel());  
 **db**.insert(**"rememberwords"**, **null**, values);  
 i+=1;  
 values.clear();  
 }  
 **if**(i==num) **break**;  
  
 }  
 }  
 **else**{  
 **int** i=0;  
 ContentValues values = **new** ContentValues();  
 **for** (cursor.moveToFirst(); !cursor.isAfterLast(); cursor.moveToNext()) {  
 WordRec tmp = **new** WordRec();  
 tmp.setWord(cursor.getString(cursor.getColumnIndex(**"\_id"**)));  
 tmp.setExplanation(cursor.getString(cursor.getColumnIndex(**"explanation"**)));  
 tmp.setLevel(cursor.getColumnIndex(**"level"**));  
 mylist.add(tmp);  
 values.put(**"word"**, tmp.getWord());  
 values.put(**"explanation"**, tmp.getExplanation());  
 values.put(**"level"**, tmp.getLevel());  
 **db**.insert(**"rememberwords"**, **null**, values);  
 i+=1;  
 **if**(i==10) **break**;  
  
 }  
 }  
 cursor.moveToFirst();  
 **adapter** = **new** Myadapter(mylist, **this**);  
 **adapter**.**color**=settings.getString(**"list\_key"**, **""**);  
 **listview**.setAdapter(**adapter**);  
  
 **final** Button submit=(Button)findViewById(R.id.***submit***);  
 submit.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **adapter**.**judge**=1;  
 **adapter**.notifyDataSetChanged();  
 Map<String, String> nowmap=**adapter**.**user\_choose**;  
 **if** (settings.getBoolean(**"checkbox\_key"**,**true**)){  
 **for** (**int** j=0;j<mylist.size();j++){  
 WordRec tmp=(WordRec) mylist.get(j);  
 ContentValues values2=**new** ContentValues();  
 values2.put(**"test\_count"**,1);  
 **db**.update(**"rememberwords"**,values2,**"word=?"**,**new** String[]{tmp.getWord()});  
 values2.clear();  
 Log.*v*(**";;;"**,String.*valueOf*(settings.getBoolean(**"checkbox\_key"**,**true**)));  
 **if** (nowmap.get(tmp.getWord())!=**null**){  
 String tmpexp=nowmap.get(tmp.getWord());  
 String nowexp=tmp.getExplanation().substring(tmp.getExplanation().length()-tmpexp.length());  
 **if** (tmpexp.equals(nowexp)) {  
 values2.put(**"correct\_count"**,1);  
 **db**.update(**"rememberwords"**,values2,**"word=?"**,**new** String[]{tmp.getWord()});  
 values2.clear();  
 }  
 }  
 }  
 }  
  
 submit.setVisibility(View.***INVISIBLE***);  
 }  
  
 });  
  
  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***menu\_second***, menu);  
 **return true**;  
 }  
  
 **public void** clickHandler(View source) {  
 finish();  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 **this**.finish();  
 **break**;  
 **case** R.id.***start\_review***:  
 Intent intent = **new** Intent(**this**, ReviewActivity.**class**);  
 startActivity(intent);  
 **break**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
 **return true**;  
 }  
}

(5)MyAdapter.java

**package** com.example.a64635.grandwordremember;  
  
  
**import** java.util.ArrayList;  
**import** java.util.Collections;  
**import** java.util.HashMap;  
**import** java.util.List;  
**import** java.util.Map;  
**import** android.content.Context;  
**import** android.content.SharedPreferences;  
**import** android.graphics.Color;  
**import** android.preference.PreferenceManager;  
**import** android.util.Log;  
**import** android.util.SparseIntArray;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.RadioButton;  
**import** android.widget.RadioGroup;  
**import** android.widget.RadioGroup.OnCheckedChangeListener;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import static** android.R.attr.***checked***;  
**import static** android.graphics.Color.*rgb*;  
  
*/\*\*  
 \* Created by 64635 on 2019/5/20.  
 \*/***public class** Myadapter **extends** BaseAdapter {  
 **public** Map<String, String> **user\_choose** = **new** HashMap<>();  
 **public int judge**=0;  
 **private** ArrayList<WordRec> **list**;  
 **private** Context **mContext**;  
 **private** SparseIntArray **ifcheck**=**new** SparseIntArray();  
 LayoutInflater **inflater**;  
 Map<Object, ArrayList<Object>> **map** = **new** HashMap<>();  
 Map<String, String> **sel** = **new** HashMap<>();  
 **public** String **color**=**"Black"**;  
  
  
 **public** Myadapter(ArrayList<WordRec> list, Context mContext) {  
 **super**();  
 **this**.**list** = list;  
 **this**.**mContext** = mContext;  
 **inflater** = LayoutInflater.*from*(mContext);  
 }  
  
 @Override  
 **public int** getCount() {  
 **return list**.size();  
 }  
  
 @Override  
 **public** Object getItem(**int** position) {  
 **return list**.get(position);  
 }  
  
  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 **public** String trans(String tmp2){  
 String tmp=**""**;  
 **int** flag=0;  
 **for** (**int** i=0;i<tmp2.length();++i){  
 **if** (tmp2.substring(i,i+1).equals(**"["**)) flag=1;  
 **if** (tmp2.substring(i,i+1).equals(**"]"**)) {  
 flag=0;  
 **continue**;  
 }  
 **if** (flag==0){  
 tmp+=tmp2.substring(i,i+1);  
 }  
 }  
 **return** tmp;  
 }  
  
 @Override  
 **public** View getView(**final int** position, View convertView, ViewGroup parent) {  
 ViewHolder holder = **null**;  
 **if**(convertView == **null**){  
 convertView = **inflater**.inflate(R.layout.***item***, **null**);  
 holder = **new** ViewHolder();  
 holder.**rg** = (RadioGroup) convertView.findViewById(R.id.***ex***);  
 holder.**rb1** = (RadioButton) convertView.findViewById(R.id.***ex1***);  
 holder.**rb2** = (RadioButton) convertView.findViewById(R.id.***ex2***);  
 holder.**rb3** = (RadioButton) convertView.findViewById(R.id.***ex3***);  
 holder.**rb4** = (RadioButton) convertView.findViewById(R.id.***ex4***);  
 holder.**tv** = (TextView) convertView.findViewById(R.id.***word***);  
 convertView.setTag(holder);  
  
 }**else** holder = (ViewHolder) convertView.getTag();  
 ArrayList<Object> list2 = **new** ArrayList<>();  
 **if** (!**map**.containsKey(position)) {  
 list2.add(position);  
 **for** (**int** i = 0; i < 3; ++i) {  
 **int** num = position;  
 **while** (list2.contains(num)) num = (**int**) (Math.*random*() \* 10);  
 list2.add(num);  
 }  
 Collections.*shuffle*(list2);  
 **map**.put(position, list2);  
 }  
 **else** list2 = **map**.get(position);  
 holder.**tv**.setText(**list**.get(position).getWord());  
 holder.**rb1**.setText(trans(**list**.get((**int**)list2.get(0)).getExplanation()));  
 holder.**rb2**.setText(trans(**list**.get((**int**)list2.get(1)).getExplanation()));  
 holder.**rb3**.setText(trans(**list**.get((**int**)list2.get(2)).getExplanation()));  
 holder.**rb4**.setText(trans(**list**.get((**int**)list2.get(3)).getExplanation()));  
 holder.**rg**.setOnCheckedChangeListener(**null**);  
 holder.**rg**.clearCheck();  
 **switch** (**color**){  
 **case "Black"**:  
 holder.**rb1**.setTextColor(Color.***BLACK***);  
 holder.**rb2**.setTextColor(Color.***BLACK***);  
 holder.**rb3**.setTextColor(Color.***BLACK***);  
 holder.**rb4**.setTextColor(Color.***BLACK***);  
 **break**;  
 **case "Red"**:  
 holder.**rb1**.setTextColor(Color.***RED***);  
 holder.**rb2**.setTextColor(Color.***RED***);  
 holder.**rb3**.setTextColor(Color.***RED***);  
 holder.**rb4**.setTextColor(Color.***RED***);  
 **break**;  
 **case "Green"**:  
 holder.**rb1**.setTextColor(Color.***GREEN***);  
 holder.**rb2**.setTextColor(Color.***GREEN***);  
 holder.**rb3**.setTextColor(Color.***GREEN***);  
 holder.**rb4**.setTextColor(Color.***GREEN***);  
 **break**;  
 **case "Blue"**:  
 holder.**rb1**.setTextColor(Color.***BLUE***);  
 holder.**rb2**.setTextColor(Color.***BLUE***);  
 holder.**rb3**.setTextColor(Color.***BLUE***);  
 holder.**rb4**.setTextColor(Color.***BLUE***);  
 **break**;  
  
 }  
  
 **if**(**ifcheck**.indexOfKey(position)!=-1) holder.**rg**.check(**ifcheck**.get(position));  
 **else** holder.**rg**.clearCheck();  
 holder.**rg**.setOnCheckedChangeListener(**new** RadioGroup.OnCheckedChangeListener(){  
 @Override  
 **public void** onCheckedChanged(RadioGroup rg,**int** index){  
 **if** (index>-1) {  
 **ifcheck**.put(position,index);  
 **user\_choose**.put(**list**.get(position).getWord(), ((RadioButton)rg.findViewById(index)).getText().toString());  
 }  
 **else if**(**ifcheck**.indexOfKey(position) > -1) **ifcheck**.removeAt(**ifcheck**.indexOfKey(position));  
 }  
 });  
 holder.**rb1**.setBackgroundColor(*rgb*(250,250,250));  
 holder.**rb2**.setBackgroundColor(*rgb*(250,250,250));  
 holder.**rb3**.setBackgroundColor(*rgb*(250,250,250));  
 holder.**rb4**.setBackgroundColor(*rgb*(250,250,250));  
 **int** position2=list2.indexOf(position);  
 **if** (**judge**==1){  
 **if** (position2==0)  
 holder.**rb1**.setBackgroundColor(*rgb*(193,255,193));  
 **else if** (position2==1)  
 holder.**rb2**.setBackgroundColor(*rgb*(193,255,193));  
 **else if** (position2==2)  
 holder.**rb3**.setBackgroundColor(*rgb*(193,255,193));  
 **else** holder.**rb4**.setBackgroundColor(*rgb*(193,255,193));  
 }  
 **return** convertView;  
 }  
  
 **public class** ViewHolder{  
 RadioGroup **rg**;  
 RadioButton **rb1**;  
 RadioButton **rb2**;  
 RadioButton **rb3**;  
 RadioButton **rb4**;  
 TextView **tv**;  
 }  
}

(6)PreferenceActivity.java

**package** com.example.a64635.grandwordremember;  
  
**import** android.app.FragmentManager;  
**import** android.app.FragmentTransaction;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.Window;  
  
**import static** com.example.a64635.grandwordremember.R.id.***toolbar***;  
  
**public class** PreferenceActivity **extends** AppCompatActivity {  
  
 *// private SettingFragment fg; https://blog.csdn.net/lincyang/article/details/20609673  
 // private FragmentManager fManager;* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_set***);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
  
 Bundle bundle;  
 setSupportActionBar(toolbar);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);*//左侧添加一个默认的返回图标* getSupportActionBar().setHomeButtonEnabled(**true**); *//设置返回键可用* toolbar.setTitle(**"学霸背单词"**);  
 toolbar.setSubtitle(**"系统设置"**);  
 FragmentManager fManager = getFragmentManager();  
  
 FragmentTransaction fTransaction = fManager.beginTransaction();  
 *//if(fg == null){* SettingFragment fg = **new** SettingFragment();  
 bundle=**new** Bundle();  
 bundle.putString(**"data"**, **"第一个Fragment"**);  
 fg.setArguments(bundle);  
 fTransaction.add(R.id.***ly\_content***,fg);  
 *// }else{  
 // fTransaction.show(fg);  
 //}* fTransaction.commit();  
 }  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***menu\_analysis***, menu);  
 **return true**;  
 }  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* **int** id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement* **if** (id == android.R.id.***home***) {  
 **this**.finish();  
 }  
 **else super**.onOptionsItemSelected(item);  
 **return true**;  
 }  
}

(7)ReviewActivity.java

**package** com.example.a64635.grandwordremember;  
  
**import** android.content.ContentResolver;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.preference.PreferenceManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.CursorAdapter;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
**import** android.widget.SimpleCursorAdapter;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.util.ArrayList;  
**import** java.util.Map;  
  
**import static** com.example.a64635.grandwordremember.R.id.***listview***;  
  
  
*/\*\*  
 \* Created by 64635 on 2019/5/15.  
 \*/***public class** ReviewActivity **extends** AppCompatActivity {  
 Toolbar toolbar;  
 **public** Database **dbHelper**;  
 ContentResolver **resolver**;  
 Myadapter **adapter**;  
 ListView **listview**;  
 Uri **uri\_user**;  
 **private** Context **mContext**;  
 **int flag** = 0;  
 **public int color** = 0;  
 SQLiteDatabase **db**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_second***);  
 toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
 toolbar.setTitle(**"学霸背单词"**);  
 toolbar.setSubtitle(**"开始复习"**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);*//左侧添加一个默认的返回图标* getSupportActionBar().setHomeButtonEnabled(**true**); *//设置返回键可用* **final** SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(**this**);  
 **mContext** = **this**;  
 **dbHelper** = **new** Database(**this**, **"rememberword.db"**, **null**, 1);  
 **dbHelper**.getWritableDatabase();  
 **listview** = (ListView) findViewById(R.id.***listview***);  
 **final** ArrayList mylist = **new** ArrayList<WordRec>();  
 **db** = **dbHelper**.getWritableDatabase();  
 Cursor cursor = **db**.query(**"rememberwords"**, **null**, **null**, **null**, **null**, **null**, **null**);  
 **for** (cursor.moveToFirst(); !cursor.isAfterLast(); cursor.moveToNext()) {  
 WordRec tmp = **new** WordRec();  
 tmp.setWord(cursor.getString(cursor.getColumnIndex(**"word"**)));  
 tmp.setExplanation(cursor.getString(cursor.getColumnIndex(**"explanation"**)));  
 tmp.setLevel(cursor.getColumnIndex(**"level"**));  
 mylist.add(tmp);  
  
 }  
  
  
 adapter = **new** Myadapter(mylist, **this**);  
 adapter.color = settings.getString(**"list\_key"**, **""**);  
 listview.setAdapter(adapter);  
  
 **final** Button submit = (Button) findViewById(R.id.submit);  
 submit.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 adapter.judge = 1;  
 adapter.notifyDataSetChanged();  
 submit.setVisibility(View.INVISIBLE);  
 }  
  
 });  
  
  
 }  
  
  
 **public void** clickHandler(View source) {  
 finish();  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.home:  
 **this**.finish();  
 **break**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
 **return true**;  
 }  
}

(8) SettingFragment.java

**package** com.example.a64635.grandwordremember;  
  
**import** android.content.SharedPreferences;  
**import** android.content.SharedPreferences.OnSharedPreferenceChangeListener;  
**import** android.os.Bundle;  
**import** android.preference.CheckBoxPreference;  
**import** android.preference.EditTextPreference;  
**import** android.preference.ListPreference;  
**import** android.preference.PreferenceFragment;  
**import** android.util.Log;  
  
**public class** SettingFragment **extends** PreferenceFragment **implements** OnSharedPreferenceChangeListener {  
   
 **private** EditTextPreference **mEtPreference**;  
 **private** ListPreference **mListPreference**;  
 **private** CheckBoxPreference **mCheckPreference**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 addPreferencesFromResource(R.xml.***preferences***);  
 initPreferences();  
 Bundle bundle=**this**.getArguments();  
 String content = bundle.getString(**"data"**);  
 Log.*e*(**"测试setting"**,content);  
 *// txt\_content.setText(content);* }  
   
 **private void** initPreferences() {  
 **mEtPreference** = (EditTextPreference)findPreference(**"edittext\_key"**);  
 **mListPreference** = (ListPreference)findPreference(**"list\_key"**);  
 **mCheckPreference** = (CheckBoxPreference)findPreference(**"checkbox\_key"**);  
 }  
   
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
  
 *// Setup the initial values* SharedPreferences sharedPreferences = getPreferenceScreen().getSharedPreferences();  
 **mListPreference**.setSummary(sharedPreferences.getString(**"list\_key"**, **""**));  
 **mEtPreference**.setSummary(sharedPreferences.getString(**"edittext\_key"**, **"linc"**));  
   
 *// Set up a listener whenever a key changes* sharedPreferences.registerOnSharedPreferenceChangeListener(**this**);  
 }  
  
 @Override  
 **public void** onPause() {  
 **super**.onPause();  
 *// Unregister the listener whenever a key changes* getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(**this**);  
 }   
   
 @Override  
 **public void** onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {  
 **if** (key.equals(**"edittext\_key"**)) {  
 **mEtPreference**.setSummary(  
 sharedPreferences.getString(key, **"20"**));  
 } **else if**(key.equals(**"list\_key"**)) {  
 **mListPreference**.setSummary(sharedPreferences.getString(key, **""**));  
 }  
 }  
}

(9)WordRec与词典相同，故不赘述。

(10)activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay"** />  
  
 </**android.support.design.widget.AppBarLayout**>  
  
 <**include layout="@layout/content\_main"** />  
  
</**android.support.design.widget.CoordinatorLayout**>

(11)activity\_second.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay"** />  
  
 </**android.support.design.widget.AppBarLayout**>  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="504dp"  
 android:orientation="horizontal"**>  
 <**ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="450dp"  
 android:layout\_marginTop="70dp"  
 android:id="@+id/listview"**/>  
 </**RelativeLayout**>  
  
 <**Button  
 android:id="@+id/submit"  
 android:layout\_width="125dp"  
 android:layout\_height="wrap\_content"  
 android:text="提交"  
 android:layout\_marginLeft="120dp"  
 android:layout\_marginTop="500dp"** />  
</**android.support.design.widget.CoordinatorLayout**>

(12)activity\_set.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay"** />  
  
 </**android.support.design.widget.AppBarLayout**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="501dp"  
 android:orientation="horizontal"**>  
  
 <**FrameLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/ly\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="68dp"**>  
 </**FrameLayout**>  
 </**RelativeLayout**>  
  
  
</**android.support.design.widget.CoordinatorLayout**>

（13）activity\_analysis.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay"** />  
  
 </**android.support.design.widget.AppBarLayout**>  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"**>  
 <**ListView  
 android:layout\_width="350dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="70dp"  
 android:id="@+id/listviewanalysis"**/>  
  
 </**RelativeLayout**>  
</**android.support.design.widget.CoordinatorLayout**>

(14)addword.xml与词典相同故不赘述。

（15）ana\_item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 <**TextView  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:text="单词"  
 android:textSize="20dp"  
 android:layout\_marginTop="20dp"  
 android:id="@+id/anaword"** />  
 <**TextView  
 android:layout\_width="60dp"  
 android:layout\_height="wrap\_content"  
 android:text="级别"  
 android:id="@+id/analevel"  
 android:textSize="20dp"  
 android:layout\_marginTop="20dp"  
 android:textAlignment="center"** />  
 <**TextView  
 android:layout\_width="60dp"  
 android:layout\_height="wrap\_content"  
 android:text="测试"  
 android:id="@+id/anatext"  
 android:layout\_marginTop="20dp"  
 android:textSize="20dp"  
 android:textAlignment="center"** />  
 <**TextView  
 android:layout\_width="60dp"  
 android:layout\_height="wrap\_content"  
 android:text="正确"  
 android:id="@+id/anaright"  
 android:layout\_marginTop="20dp"  
 android:textSize="20dp"  
 android:textAlignment="center"** />  
  
</**LinearLayout**>

（16）content\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/content\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/pink"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"  
 tools:showIn="@layout/activity\_main"**>  
  
 <**ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/rem"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginBottom="80dp"  
 android:id="@+id/imageView"** />  
</**RelativeLayout**>

(17)item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/content\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/pink"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"  
 tools:showIn="@layout/activity\_main"**>  
  
 <**ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/rem"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginBottom="80dp"  
 android:id="@+id/imageView"** />  
</**RelativeLayout**>

(18)menu\_main.xml

<**menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"**>  
 <**item android:id="@+id/entertest"  
 android:title="进入测验"  
 android:icon="@mipmap/text"  
 app:showAsAction="ifRoom"** />  
 <**item  
 android:id="@+id/add\_word"  
 android:orderInCategory="100"  
 android:title="新增单词"  
 app:showAsAction="never"** />  
 <**item  
 android:id="@+id/stat"  
 android:orderInCategory="100"  
 android:title="学习统计"  
 app:showAsAction="never"** />  
 <**item  
 android:id="@+id/find\_word"  
 android:orderInCategory="100"  
 android:title="查找单词"  
 app:showAsAction="never"** />  
 <**item  
 android:id="@+id/set"  
 android:orderInCategory="100"  
 android:title="系统设置"  
 app:showAsAction="never"** />  
  
  
</**menu**>

(19)menu\_second.xml

<**menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 tools:context="com.example.a64635.grandwordremember.MainActivity"**>  
  
 <**item  
 android:id="@+id/start\_review"  
 android:orderInCategory="100"  
 android:title="开始复习"  
 app:showAsAction="never"** />  
</**menu**>

(20)preferences.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"** >  
 <**PreferenceCategory android:title="外观设置"**>  
 <**ListPreference  
 android:key="list\_key"  
 android:defaultValue="Black"  
 android:title="单词颜色"  
 android:summary="Black"  
 android:entries="@array/list\_preference"  
 android:entryValues="@array/list\_preference"  
 android:dialogTitle="单词颜色"** />  
 </**PreferenceCategory**>  
 <**PreferenceCategory android:title="其它设置"**>  
 <**EditTextPreference  
 android:key="edittext\_key"  
 android:defaultValue="10"  
 android:summary="10"  
 android:title="测试单词数"** />  
 <**CheckBoxPreference  
 android:key="checkbox\_key"  
 android:defaultValue="true"  
 android:summary=""  
 android:title="保存时是否统计"** />   
 </**PreferenceCategory**>  
</**PreferenceScreen**>

【完成情况】

完成了哪些内容？(√完成 ×未做或未完成)

1 [√ ] 2 [√ ] 3[√ ] 4[√ ] 5 [√ ] 6 [√ ] 7[ √] 8[√ ] 9[√ ] 10[ √]

11 [√ ] 12 [ √] 13[√ ] 14[√ ] 15 [√ ] 16 [ √] 17[√ ] 18[√ ] 19[ √]

自己增加的功能：  
（1）背单词的开始复习是从已学过的单词中选取要记忆的单词。

（2）把背单词的ActionBar上开始测试用图标代替

【实验体会】

首先说一个不算制作里的问题：打不开老师的文件，没有办法build……最后通过修改app文件夹里build.gradle文件里的sdk版本成功解决。下面是制作时遇到的问题或体会：

第一个项目：

首先在做界面的时候，我发现老师给的demo主题是绿色的，好奇如何可以改变主题的颜色。后来新建了一个项目与demo做对比，发现只要在values文件夹里，先修改colors.xml里的各个颜色，再在styles.xml里设置各个部分的颜色即可。在创建数据库时，发现app出现崩溃现象，原因在于不可以把autoincrement加在text类型的属性上，会出现：AUTOINCREMENT is only allowed on an INTEGER PRIMARY KEY。

在数据库导入的操作中，遇到的主要的内容总结如下：

一、把Json转换为object的步骤：

1、读取输入流为JsonArray的格式；

2、遍历JsonArray,创建WordRec类将其转化为object，提取键值对插入到数据库中。

二、Cursor对象绑定数据时，Cursor必须包含“\_id”字段，否则将无法完成数据绑定。所以在写sql语句时，要使用select xxx as \_id……不然会造成程序崩溃。

在写contentprovider时，出现报错：

Installation failed with message INSTALL\_FAILED\_CONFLICTING\_PROVIDER。原因是这个provider已经有应用使用（可能我原本运行了老师的示例代码），只要在注册时改一个名字即可。（比如我就随性地在后面加了个数字）

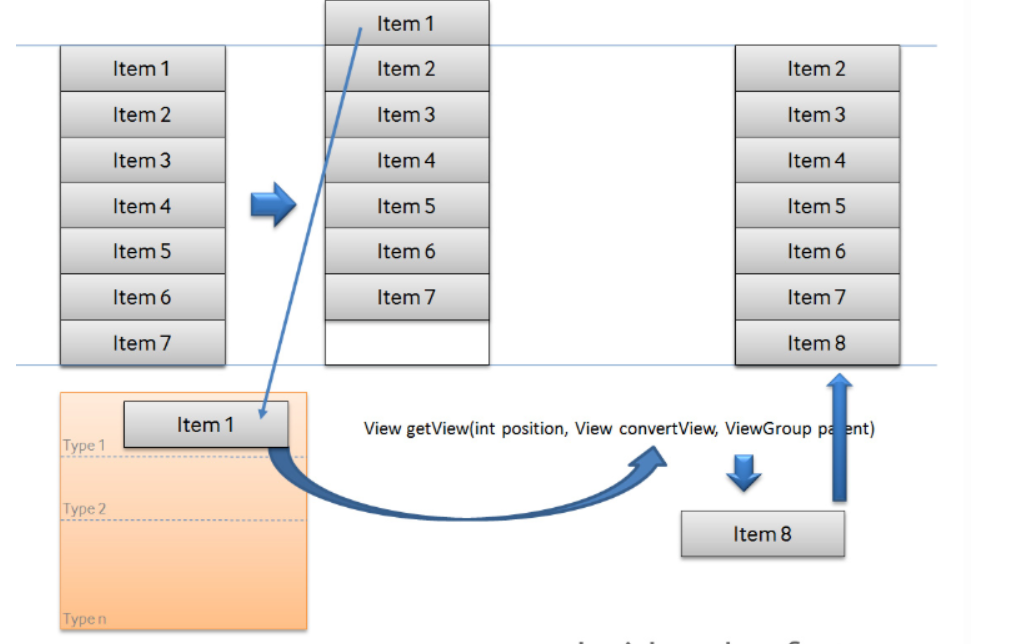
第二个项目：

在做界面的时候，首先设置背景颜色要为粉红色，但是要修改style文件很不方便。这个时候我想了一个投机取巧的办法——截图rem的背景，保存到drawable中，此时就可以直接使用background啦。

此时连接项目一的provider发现返回的是一个空指针，说明这个时候的数据库是一个空对象。经已经踩过雷的小伙伴帮助，我发现原来项目一创建了一个新的数据库，此时那些下载部分的insert需要调用content的insert，修改之后，项目二可以连接上数据库下载的单词了。

在做listview嵌套radiogroup时，自带的适配器已经无法起作用，所以需要进行重构。在进行音标删除的时候，我也特别注意到要使用equals而不是==判断相等才能成功。

在测试单词功能制作时出现了ListView嵌套RadioGroup时选中混乱、显示现象，原因是listview的复用机制导致的监听器混乱，查阅了一下可以如下图所示：



博客作者给的解释是：屏幕滑动第一条数据消失，复用机制就会发生了，先把item1放到recycler中，等待屏幕底部数据出现，当item8出现时，就会复用item1的布局和数据。解决方法是每次取消掉监听器，每次选时进行先行条件判断，如果已经选中则设置为未选中，每次进行恢复就可以了。

在做“新增单词”对话框的时候，进入自带的模拟器，对话框显示了一两秒，模拟器就会崩溃退出，遍寻不到原因。最终我下载了Genymotion（反正自带的模拟器太……慢……），在Genymotion运行就不会闪退了……看来是模拟器的问题，这也是我无法解决的问题。

在做长按事件时，需要注意的细节是把onItemLongClick返回设置为true，否则长按是会执行setOnItemClickListener。

在做增删改功能时，由于我使用适配器时SimpleCursorAdapter，所以实时更新非常麻烦。上网查询之后发现可以直接从最基本的cursor入手，在每进行一步相关操作，即设置一个新的cursor并赋给当前的cursor，即可实现实时的修改了显示了。在做查找条时，我如法炮制，点击一个按钮后即赋予一个新的cursor，这个cursor来源于以该按钮字母为开头的query集合。在变背景颜色的时候出了一些小插曲，即不点击了也不会变成原来的粉色背景，这时候我遍历了layout元素里面的所有其他元素，全部设置为原背景即可。

在进行“显示语义”实现时，出现了只能根据当前页面显示，导航都无法显示的问题，解决办法是另用一个adapter和cursor刷新。在写查找的时候也犯了一个低级错误——取出edittext的时候没有gettext导致查找的是空白（没有想到低级错误，导致查别的错查了很久）。

其余的功能实现是较为简单的，很多都是以前的照葫芦画瓢，或是可以直接参考老师代码，所以不赘述了。

小插曲：用AVD自带的模拟器在弹出对话框的时候有可能会有opengl的问题，我以为是bug修了一晚上毫无头绪，结果换了genymotion模拟器，啥事儿没有……

两个app从0开始做完，筋疲力尽，但获益良多。

【交实验报告】

(a) 每位同学独立完成本实验内容并填写实验报告。

(b) 截止时间：2019年6月2日23:00

上交作业网站：<http://172.18.187.9/netdisk/default.aspx?vm=16and>

文件夹: /实验上交/02、安卓数据库实验

上传文件：学号\_姓名\_数据库实验.doc (实验报告)   
学号\_姓名\_数据库实验.rar (包含所有项目，注意删除build和app/build目录)

学号\_姓名\_数据库实验.mp4 (动态运行)