院 系 数据科学与计算机学院 学 号 16337233 姓 名 王凯祺

【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，可以只完成其中一部分功能，其中1~8是需要做的步骤，9~19是选做步骤，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。见 “Android程序设计(六).ppt” p19 p41。

（c）做每个步骤前，最好先把参考源码运行一遍。

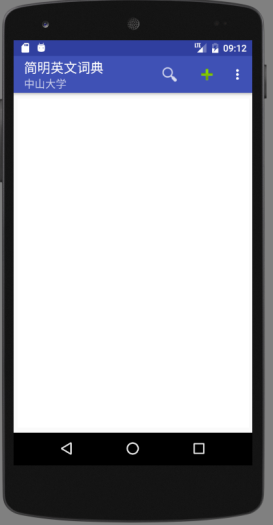
（d）可以先看一下老师的类设计（在本报告末尾）

【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**：

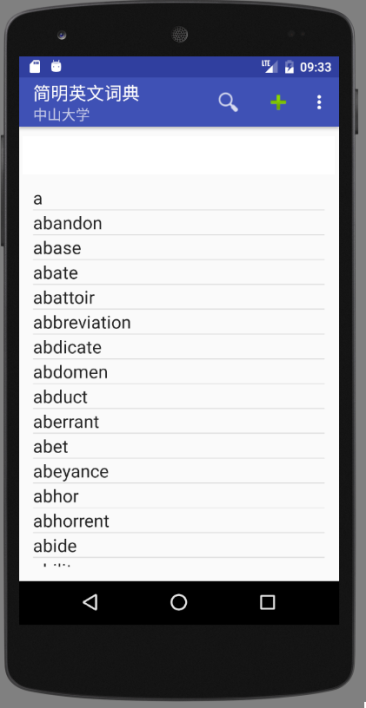
(1)AndroidManifest.xml中加入允许互联网访问：  
<**uses-permission android:name="android.permission.INTERNET"**/>

(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）

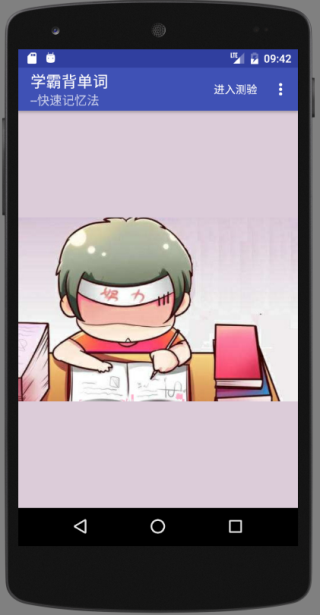
**参考源码**： SqliteCv、FirstProvider

**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

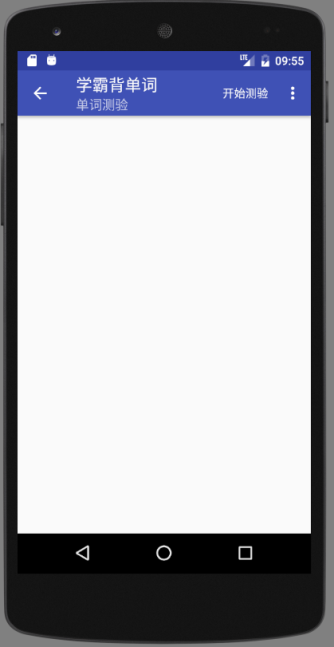
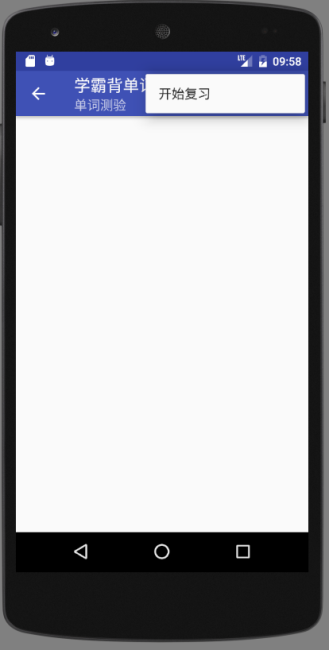
项目：GrandWordRemember

5、做一个主界面(MainActivity)

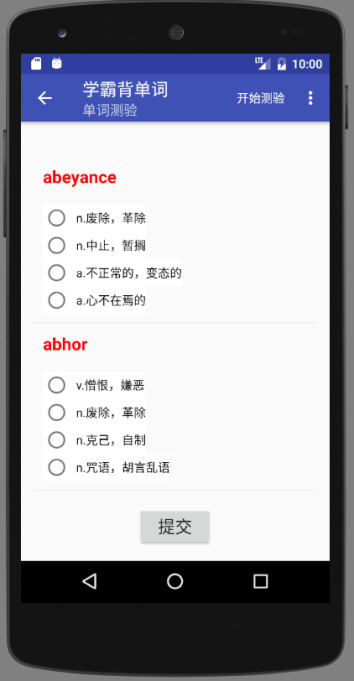
**参考资源**： rem.jpg

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。

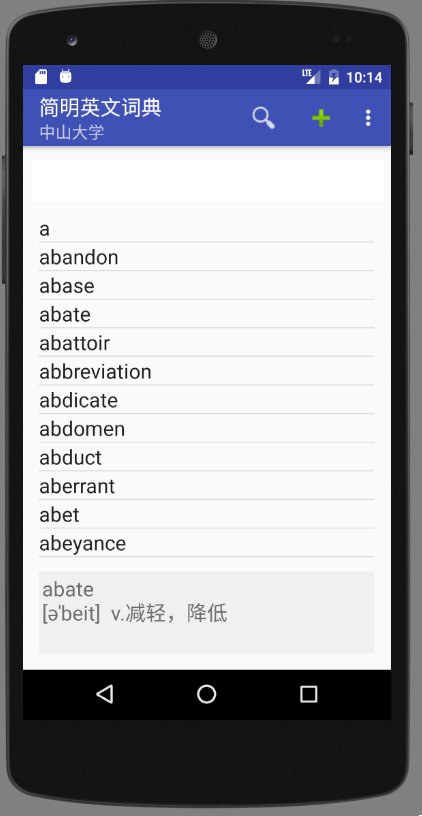
\* Context在Activity中Context可以用this或ActivityName.this取得，在事件处理程序中可以用ActivityName.this取得。

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

（三）词典扩展功能设计

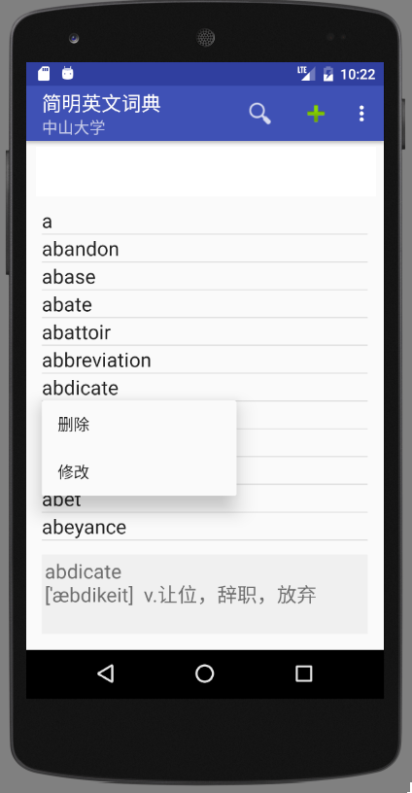
项目：GrandDictionary

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



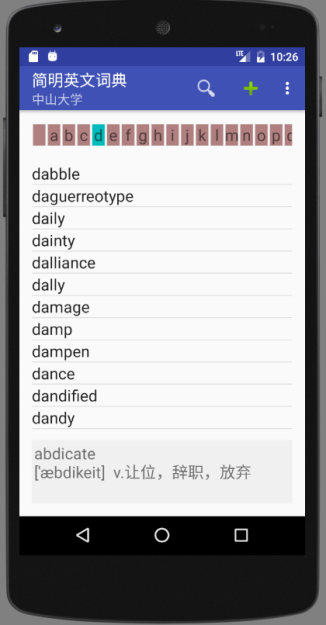
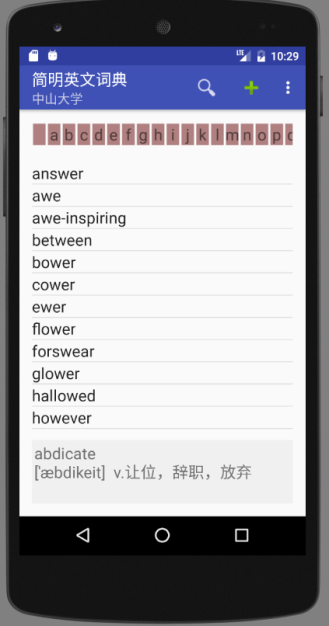
**参考源码**：ListViewSimple，SqliteCv

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

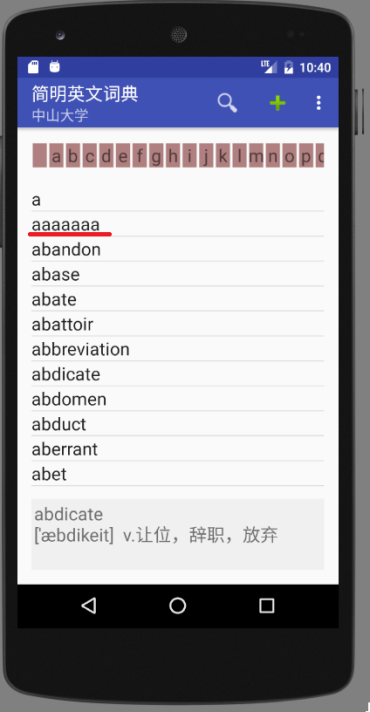
**功能说明**：在列表上方增加一个字母表，点击时只显示以这些字母开头的单词；点击查询图标可以查询包含输入子串的单词。

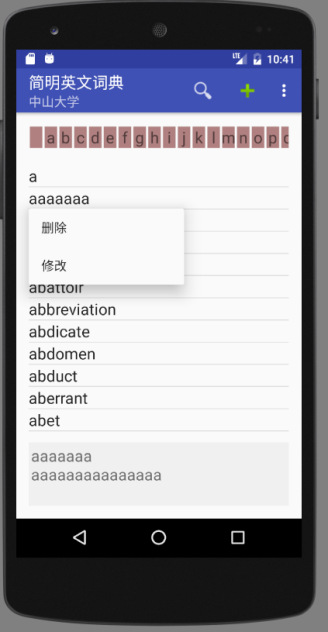
**参考源码**：DynamicView,HorizontalScrollView（自己上网查找使用方法）、SqliteCv

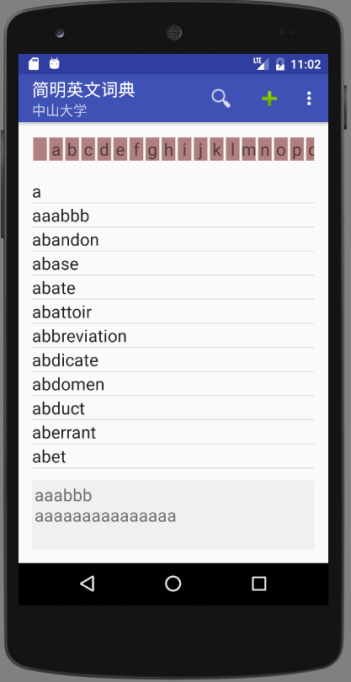
**编程参考**：每个单词可以从字母表” a…z ”取出一个字符(""+charAt()或substring()),然后创建TextView。空格查出所有单词

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

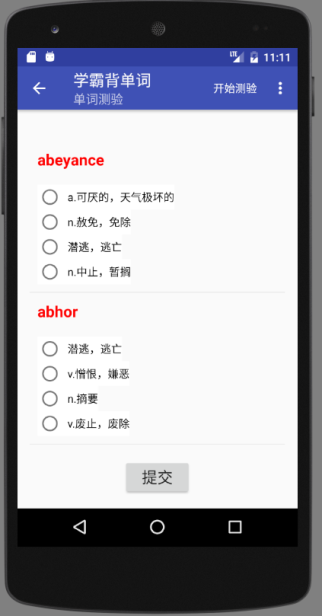
db.execSQL(**"CREATE TABLE words(\_id integer primary key autoincrement,word varchar(64) unique,level int default 0, test\_count int default 0, correct\_count int default 0,last\_test\_time timestamp)"**);

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。

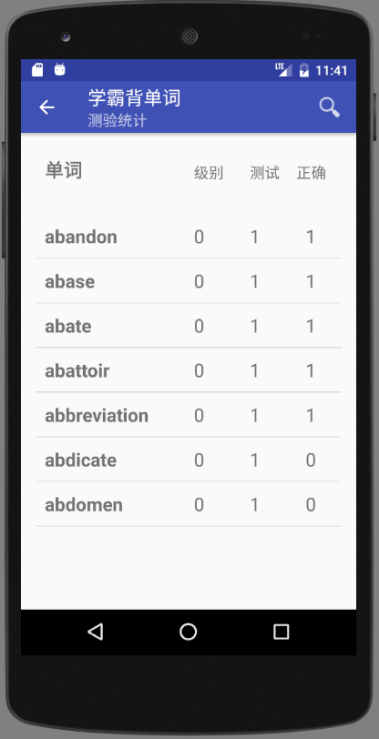
  

**参考源码**：SpinnerCustom（改为ListView），TestAdapter.java

**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

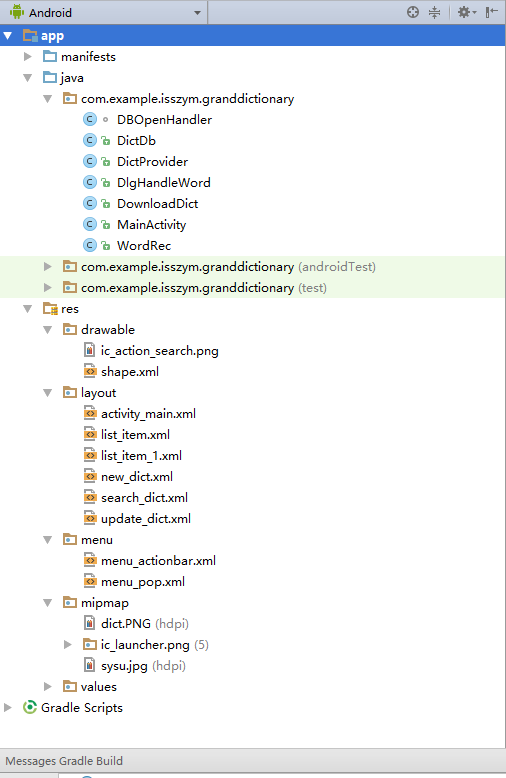
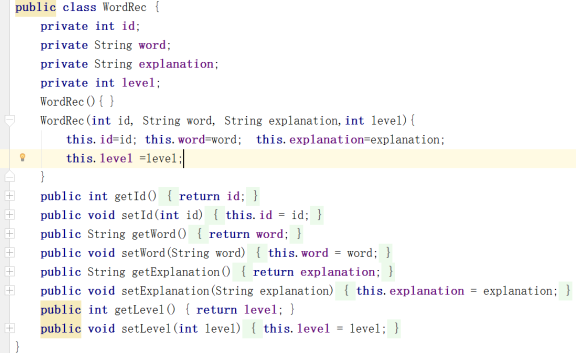
（e）利用中文解释，选择英文单词

（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)

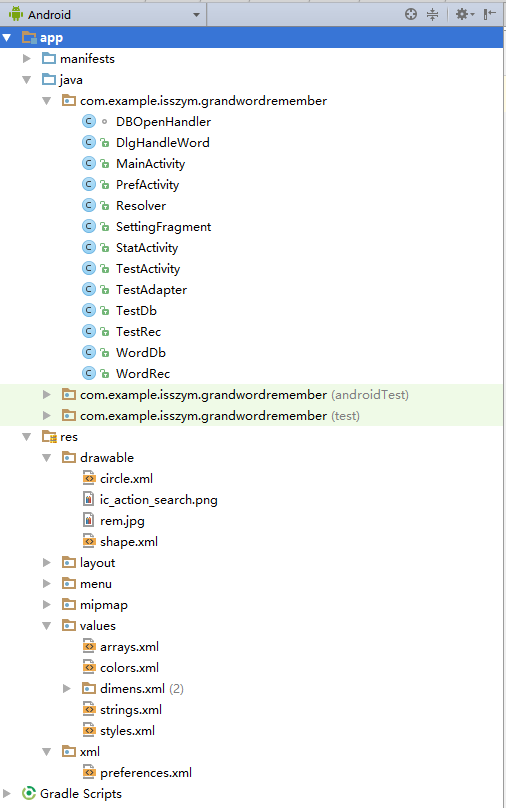
 

DbOpenHandler：打开数据库类 DictDb:所有数据库操作类

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）

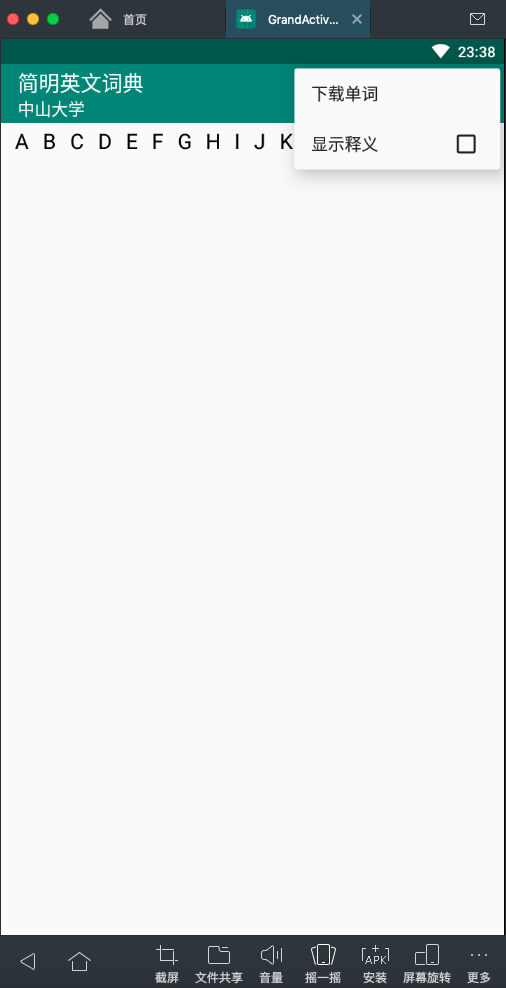
TestDb：本地数据库操作类 TestRec：本地统计记录类 WordDb：（远程）词典操作类

WordRec：单词记录类

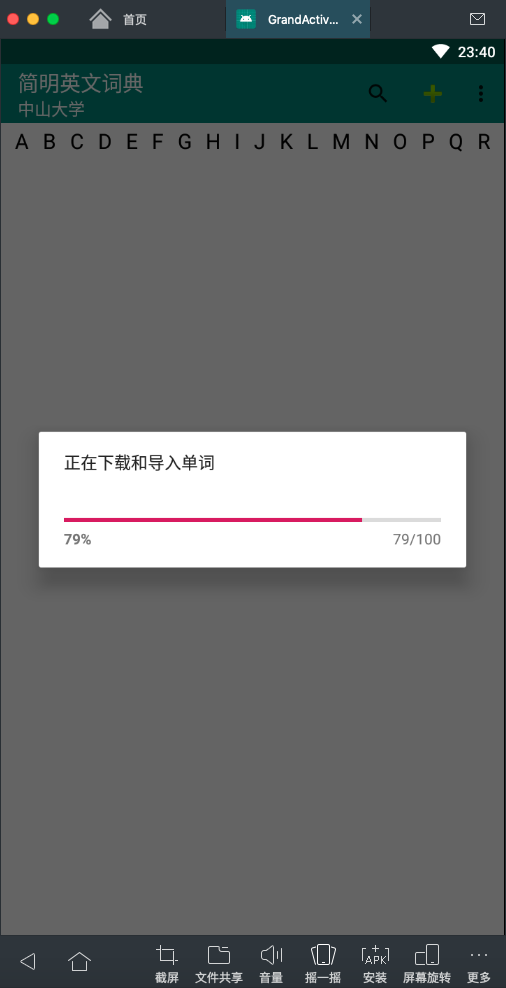
*完成后程序运行截屏及简要说明：*

【词典程序】

1. Action Bar界面



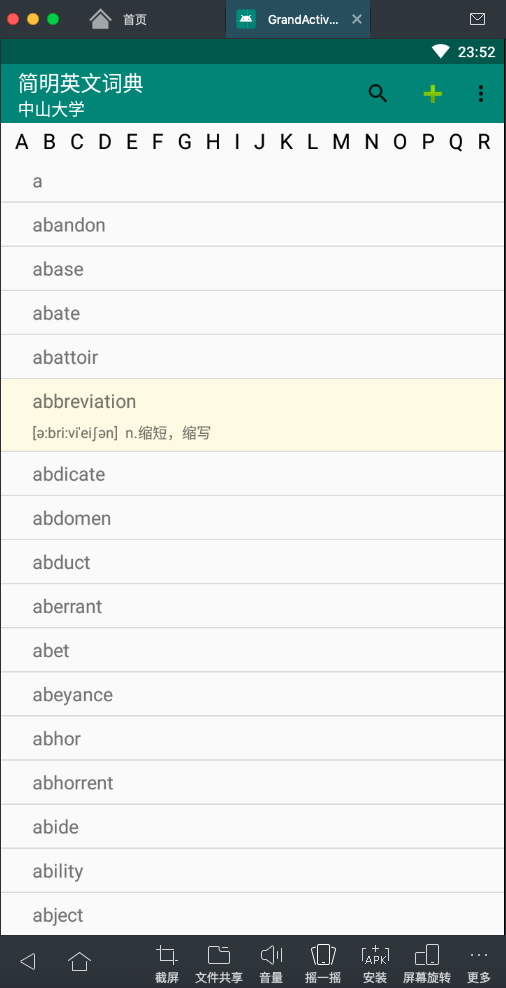
1. 下载单词，显示进度条

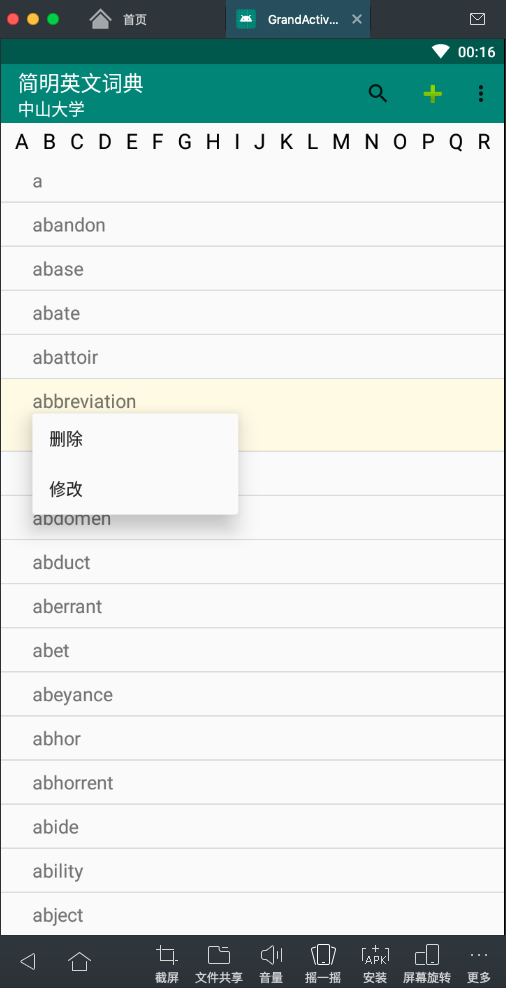
1. 用ListView将单词显示出来



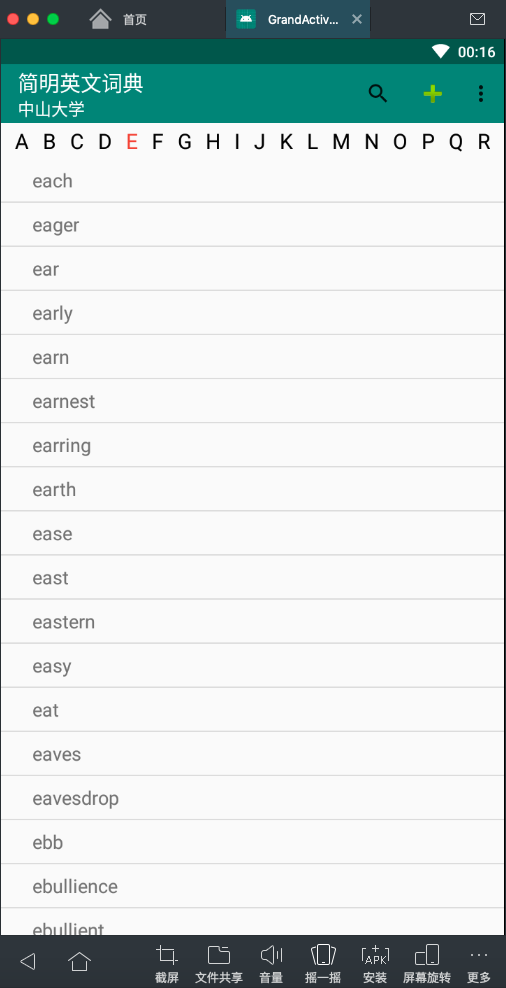
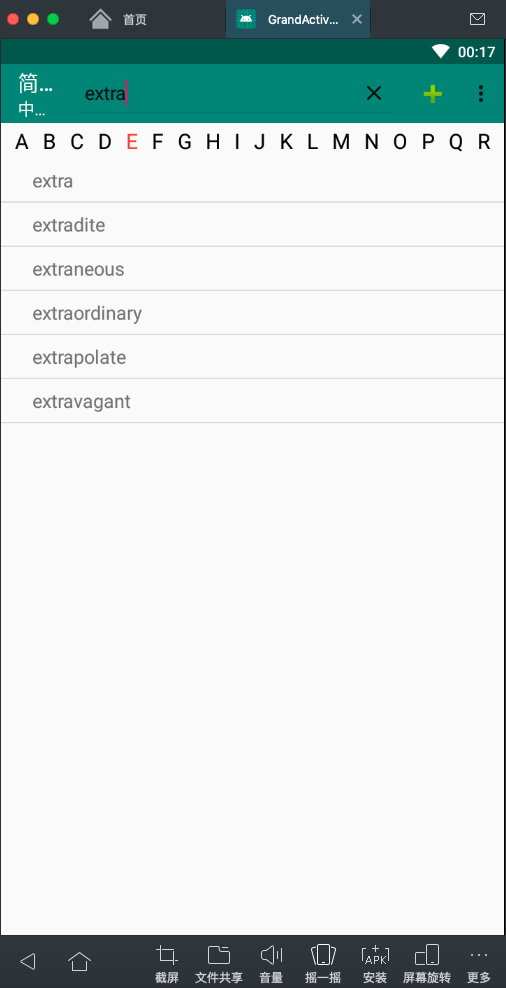
1. 点击单词显示释义

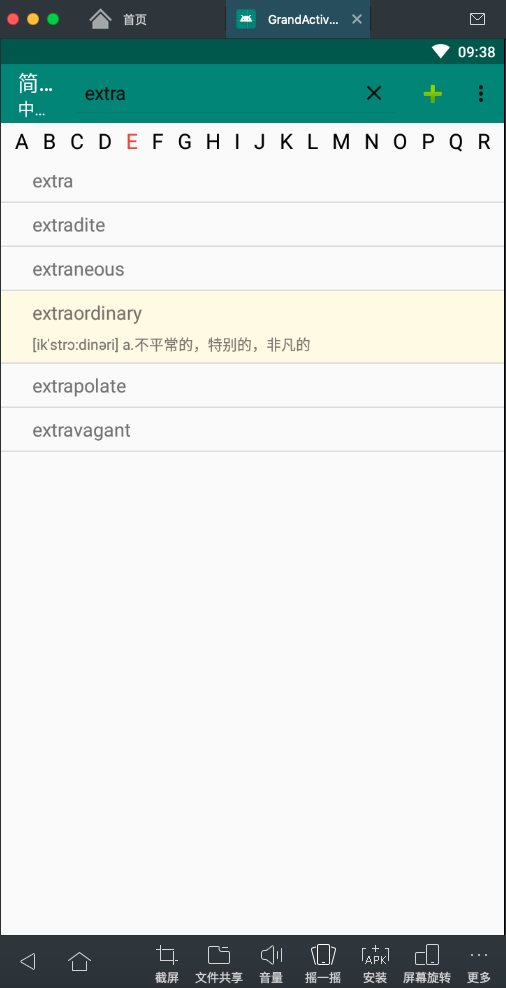


1. 长按单词显示弹出式菜单

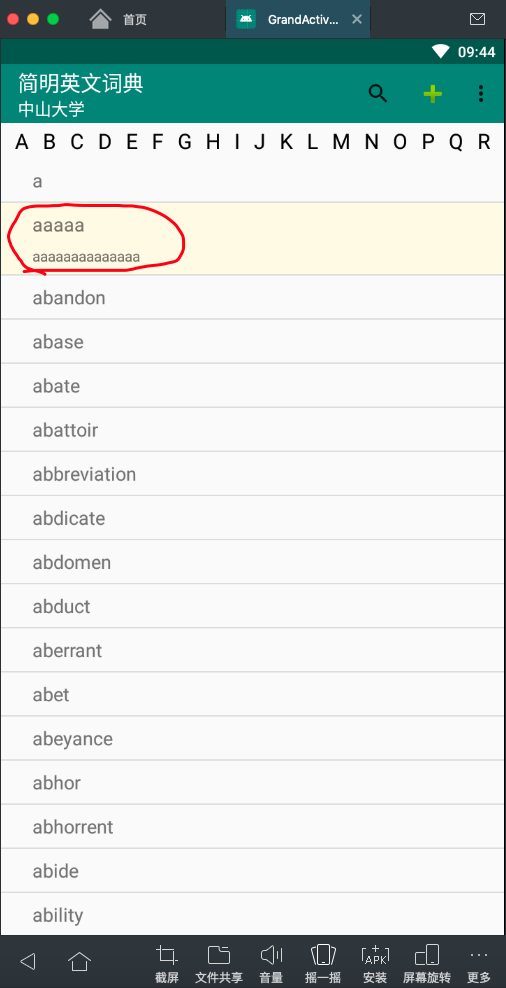


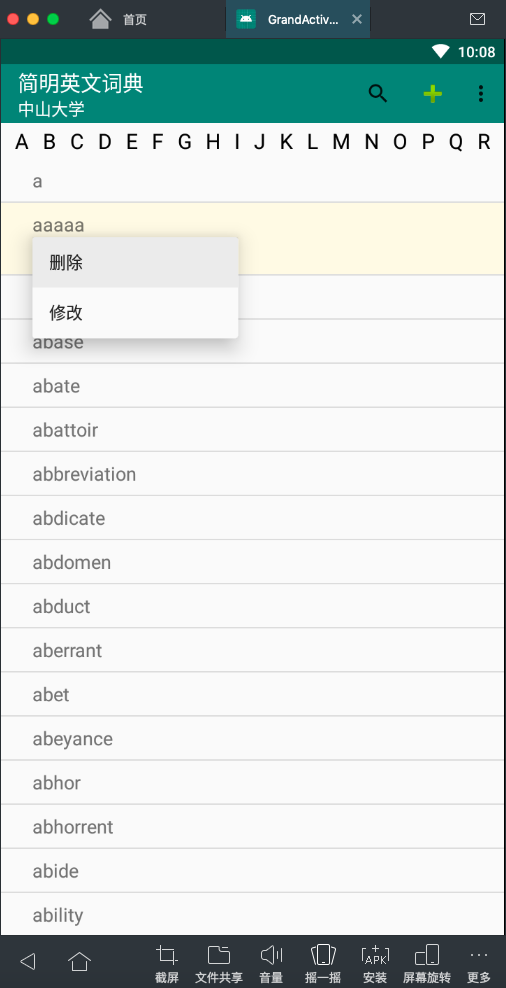
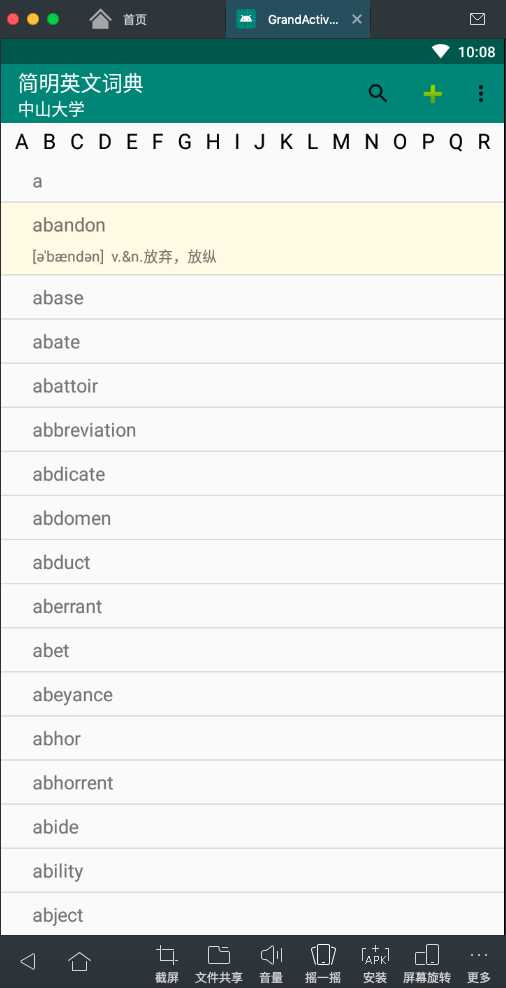
1. 单词查询功能

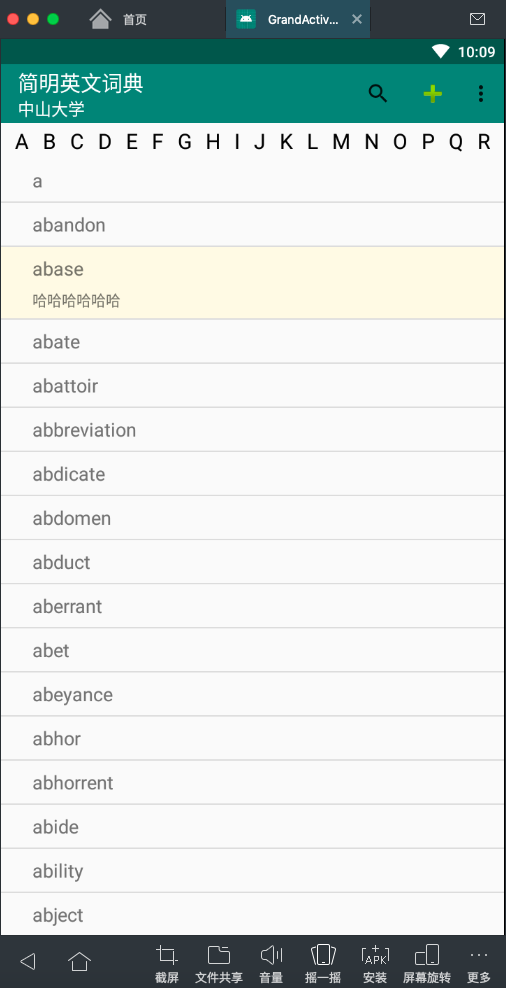
 



1. 增加、删除、修改单词

1. 显示释义



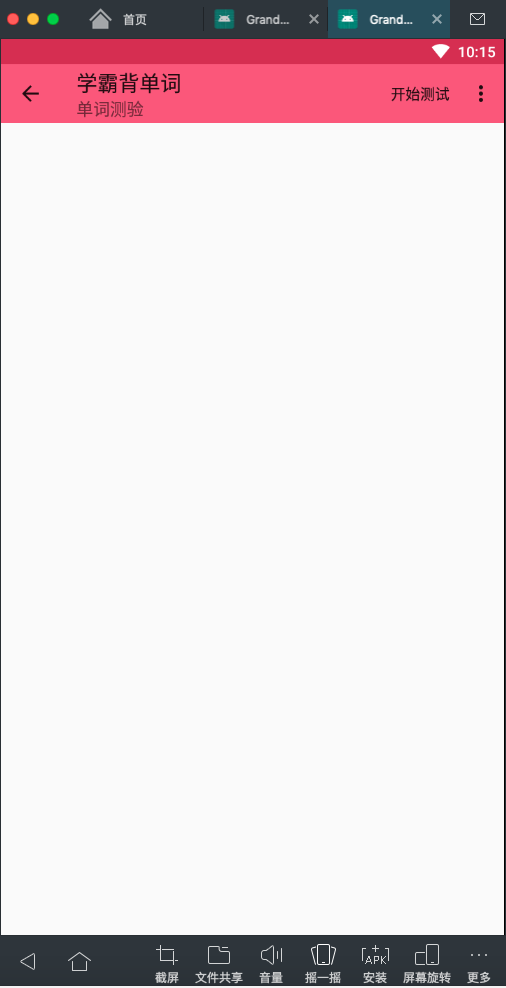
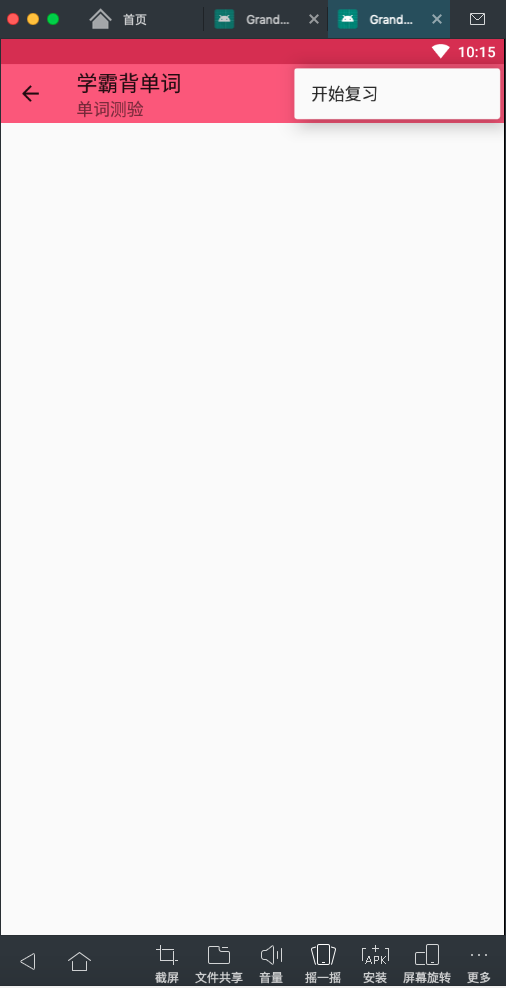
…

【背单词程序】

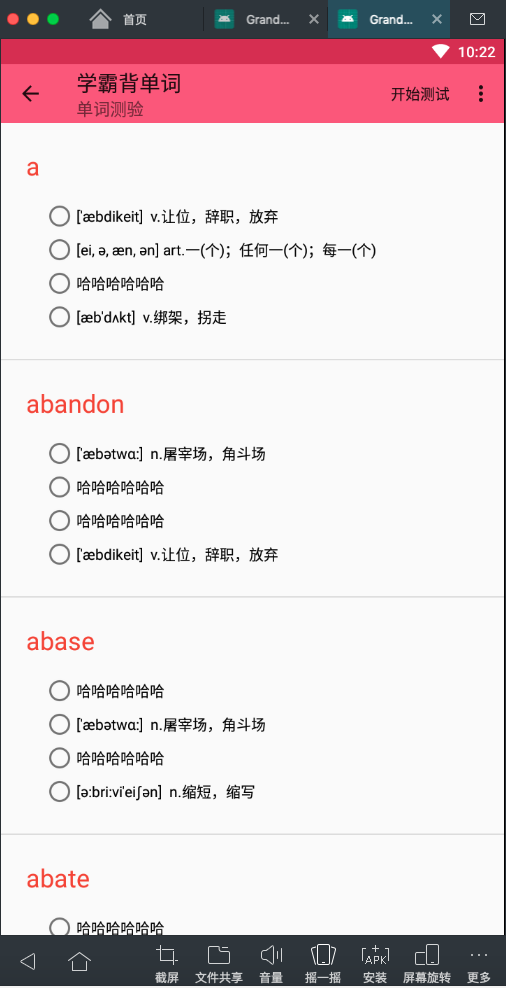
1. 做一个主界面

1. 点击进入测验，进入测验界面

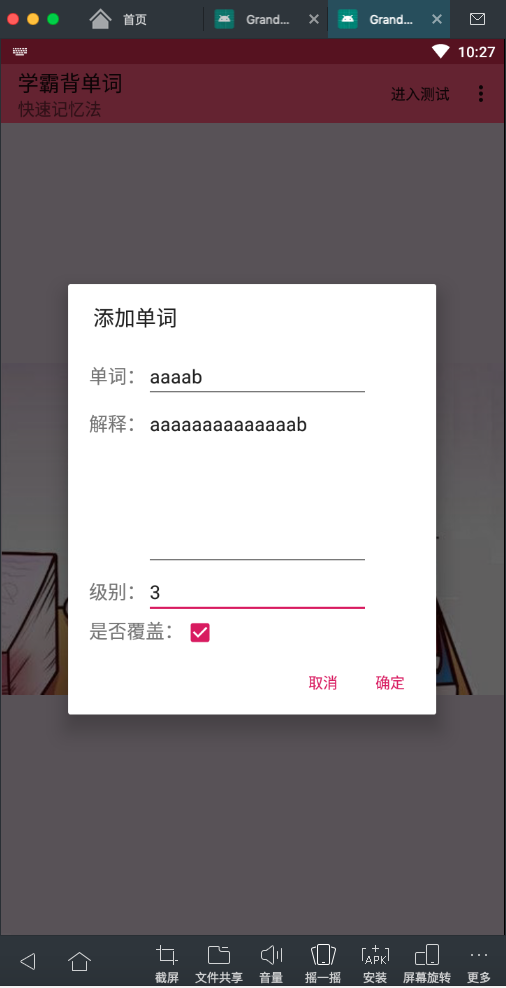
1. 开始测验



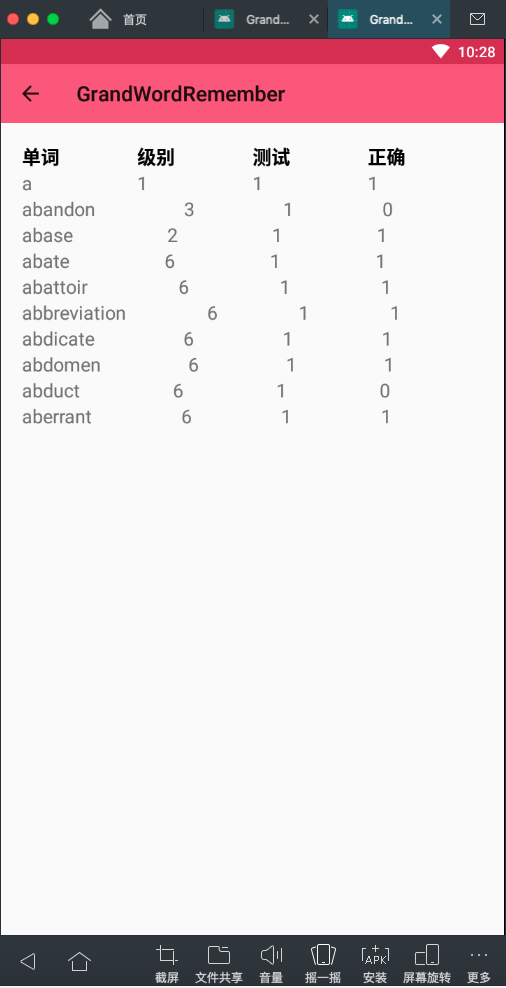
1. 测验完提交评分

1. 增加单词

1. 统计数据



*完成后的主要源码(.java和.xml)：*

【词典程序】

1. MainActivity.java

**package** com.example.grandactivity;  
  
**import** android.app.Activity;  
**import** android.app.AlertDialog;  
**import** android.app.ProgressDialog;  
**import** android.content.ContentValues;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.graphics.Color;  
**import** android.support.constraint.ConstraintLayout;  
**import** android.support.v7.app.ActionBar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.Log;  
**import** android.util.Pair;  
**import** android.util.TypedValue;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.LinearLayout;  
**import** android.widget.ListView;  
**import** android.widget.ProgressBar;  
**import** android.widget.SearchView;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** org.w3c.dom.Text;  
  
**import** java.io.File;  
**import** java.util.List;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** ProgressDialog **mProgressDialog**;  
 **private** Toolbar **mToolbar**;  
 **private** ActionBar **actionBar**;  
 **private** DBOpenHandler **dbOpenHandler**;  
 **private** SQLiteDatabase **db**;  
 **private final** String **dictURL** = **"http://172.18.187.9:8080/dict/"**;  
 **private final** String **dictURL\_bak** = **"http://test.wronganswer.cn:8090/dict.json"**;  
 **private** OnDownloadListener **downListener**;  
 String **localdict**;  
 **private** AlertDialog **alertDialog** = **null**;  
 **private** AlertDialog.Builder **dialogBuilder** = **null**;  
 **private** ContentValues **cv**;  
 **private boolean alwaysShowExplanation**;  
 **private** LinearLayout **linearLayout**;  
 **private int nowABCDEFG**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// top ABCDEFG* **nowABCDEFG** = -1;  
 **linearLayout** = (LinearLayout)findViewById(R.id.***linearLayout***);  
 **for** (**int** i = 0; i < 26; ++i) {  
 TextView textView = **new** TextView(**this**);  
 textView.setText(String.*valueOf*((**char**)(**'A'** + i)));  
 textView.setTextSize(TypedValue.***COMPLEX\_UNIT\_SP***, 20);  
 textView.setPadding(i == 0 ? 20 : 10, 5, i == 25 ? 20 : 10, 5);  
 textView.setTextColor(Color.***BLACK***);  
 textView.setId(100000 + i);  
 textView.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 TextView textView = (TextView)v.findViewById(v.getId());  
 **for** (**int** j = 0; j < 26; ++j) {  
 TextView t = (TextView)findViewById(100000 + j);  
 t.setTextColor(Color.***BLACK***);  
 }  
 **if** (**nowABCDEFG** == textView.getId()) {  
 **new** ImportTask(MainActivity.**this**, **db**, **alwaysShowExplanation**).execute();  
 } **else** {  
 **nowABCDEFG** = textView.getId();  
 textView.setTextColor(Color.*parseColor*(**"#F44336"**));  
 **new** SearchABCDEFGTask(MainActivity.**this**, **db**, **alwaysShowExplanation**, String.*valueOf*((**char**)(**'a'** + (**nowABCDEFG** - 100000)))).execute();  
 }  
 }  
 });  
 **linearLayout**.addView(textView);  
 }  
  
 *//We have to tell the activity where the toolbar is* **mToolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**mToolbar**);  
 **actionBar** = getSupportActionBar();  
 *//Display home with the "up" arrow indicator* **actionBar**.setHomeButtonEnabled(**false**);  
 *//actionBar.setDisplayHomeAsUpEnabled(true);* File file = **new** File(**this**.getFilesDir(), **"dict.db3"**);  
 Log.*d*(**"DATABASE"**, **"sql file = "** + file.getAbsolutePath());  
 **dbOpenHandler** = **new** DBOpenHandler(**this**.getApplicationContext(), file.getAbsolutePath(), **null**, 1);  
 **db** = **dbOpenHandler**.getReadableDatabase();  
 **final** String create\_table\_sql = **"CREATE TABLE IF NOT EXISTS dict(\_id integer primary key autoincrement, word varchar(64) unique COLLATE NOCASE, explanation text, level int default 0, modified\_time timestamp)"**;  
 **db**.execSQL(create\_table\_sql);  
 **cv** = **new** ContentValues();  
 **alwaysShowExplanation** = **false**;  
 **new** ImportTask(MainActivity.**this**, **db**, **alwaysShowExplanation**).execute();  
 }  
  
 @Override  
 **public boolean** onPrepareOptionsMenu(Menu menu) {  
 menu.findItem(R.id.***app\_bar\_switch***).setChecked(**alwaysShowExplanation**);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 Intent intent = **new** Intent(**this**,MainActivity.**class**);  
 startActivity(intent);  
 **break**;  
 **case** R.id.***app\_bar\_add***:  
 ConstraintLayout addWordForm = (ConstraintLayout) getLayoutInflater()  
 .inflate(R.layout.***add\_word\_dialog***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(MainActivity.**this**);  
 **alertDialog** = **dialogBuilder** .setTitle(**"添加单词"**)  
 .setView(addWordForm)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 String word = ((EditText)**alertDialog**.findViewById(R.id.***editWord***)).getText().toString();  
 String explanation = ((EditText)**alertDialog**.findViewById(R.id.***editExplanation***)).getText().toString();  
 String level = ((EditText)**alertDialog**.findViewById(R.id.***editLevel***)).getText().toString();  
 **boolean** overwrite = ((CheckBox)**alertDialog**.findViewById(R.id.***checkOverwrite***)).isChecked();  
 **cv**.clear();  
 **cv**.put(**"word"**, word);  
 **cv**.put(**"explanation"**, explanation);  
 **cv**.put(**"level"**, level);  
 **db**.insert(**"dict"**, **null**, **cv**);  
 **if** (overwrite) {  
 **db**.update(**"dict"**, **cv**, **"word=?"**, **new** String[]{word});  
 }  
 **new** ImportTask(MainActivity.**this**, **db**, **alwaysShowExplanation**).execute();  
 }  
 })  
 .create();  
 **alertDialog**.show();  
 **break**;  
 **case** R.id.***app\_bar\_search***:  
 **return super**.onOptionsItemSelected(item);  
 **case** R.id.***app\_bar\_download***:  
 File file = **new** File(**this**.getCacheDir(), **"dict.json"**);  
 **localdict** = file.getAbsolutePath();  
 *// instantiate it within the onCreate method* **mProgressDialog** = **new** ProgressDialog(MainActivity.**this**);  
 **mProgressDialog**.setMessage(**"正在下载和导入单词"**);  
 **mProgressDialog**.setIndeterminate(**true**);  
 **mProgressDialog**.setProgressStyle(ProgressDialog.***STYLE\_HORIZONTAL***);  
 **mProgressDialog**.setCancelable(**true**);  
 *// execute this when the downloader must be fired* **downListener** = **new** OnDownloadListener() {  
 @Override  
 **public void** onSuccess() {  
 **new** ImportTask(MainActivity.**this**, **db**, **alwaysShowExplanation**).execute();  
 }  
  
 @Override  
 **public void** onFailed() {  
 }  
 };  
 **final** DownloadTask downloadTask = **new** DownloadTask(MainActivity.**this**, **db**, **mProgressDialog**, **downListener**);  
 **mProgressDialog**.setOnCancelListener(**new** DialogInterface.OnCancelListener() {  
 @Override  
 **public void** onCancel(DialogInterface dialog) {  
 downloadTask.cancel(**true**);  
 }  
 });  
 downloadTask.execute(**dictURL\_bak**, **localdict**);  
 **break**;  
 **case** R.id.***app\_bar\_switch***:  
 **alwaysShowExplanation** = !**alwaysShowExplanation**;  
 **new** ImportTask(MainActivity.**this**, **db**, **alwaysShowExplanation**).execute();  
 **break**;  
 **default**:  
 **return super**.onOptionsItemSelected(item);  
 }  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_main***, menu);  
 MenuItem searchItem = menu.findItem(R.id.***app\_bar\_search***);  
 SearchView searchView = **null**;  
 **if** (searchItem != **null**) {  
 searchView = (SearchView)searchItem.getActionView();  
 }  
 searchView.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
 **new** SearchTask(MainActivity.**this**, **db**, **alwaysShowExplanation**, query).execute();  
 **return false**;  
 }  
  
 @Override  
 **public boolean** onQueryTextChange(String newText) {  
 **return false**;  
 }  
 });  
 searchView.setOnCloseListener(**new** SearchView.OnCloseListener() {  
 @Override  
 **public boolean** onClose() {  
 **new** ImportTask(MainActivity.**this**, **db**, **alwaysShowExplanation**).execute();  
 **return false**;  
 }  
 });  
 **return true**;  
 }  
}

1. DownloadTask.java

**package** com.example.grandactivity;  
  
**import** android.app.Activity;  
**import** android.app.ProgressDialog;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.os.AsyncTask;  
**import** android.os.PowerManager;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.ListView;  
**import** android.widget.ProgressBar;  
**import** android.widget.Toast;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONObject;  
  
**import** java.io.FileInputStream;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.OutputStream;  
**import** java.net.HttpURLConnection;  
**import** java.net.URL;  
**import** java.sql.Timestamp;  
  
**public class** DownloadTask **extends** AsyncTask<String, Integer, String> {  
 *// Usage: DownloadTask.execute(String URL, String downloadLocation);* **private** Context **context**;  
 **private** PowerManager.WakeLock **mWakeLock**;  
 **private** SQLiteDatabase **db**;  
 **private** ProgressDialog **mProgressDialog**;  
 **private** ProgressBar **progressBar**;  
 **private** WordListViewAdapter **adapter**;  
 **private** OnDownloadListener **mListener**;  
  
 **public** DownloadTask(Context context, SQLiteDatabase db, ProgressDialog mProgressDialog, OnDownloadListener mListener) {  
 **this**.**context** = context;  
 **this**.**db** = db;  
 **this**.**mProgressDialog** = mProgressDialog;  
 **this**.**mListener** = mListener;  
 }  
  
 @Override  
 **protected** String doInBackground(String... sUrl) {  
 InputStream input = **null**;  
 OutputStream output = **null**;  
 HttpURLConnection connection = **null**;  
 **try** {  
 URL url = **new** URL(sUrl[0]);  
 connection = (HttpURLConnection) url.openConnection();  
 connection.connect();  
 *// expect HTTP 200 OK, so we don't mistakenly save error report  
 // instead of the file* **if** (connection.getResponseCode() != HttpURLConnection.***HTTP\_OK***) {  
 **return "Server returned HTTP "** + connection.getResponseCode()  
 + **" "** + connection.getResponseMessage();  
 }  
 *// this will be useful to display download percentage  
 // might be -1: server did not report the length* **int** fileLength = connection.getContentLength();  
 *// download the file* input = connection.getInputStream();  
 output = **new** FileOutputStream(sUrl[1]);  
 **byte** data[] = **new byte**[4096];  
 **long** total = 0;  
 **int** count;  
 **while** ((count = input.read(data)) != -1) {  
 *// allow canceling with back button* **if** (isCancelled()) {  
 input.close();  
 **return null**;  
 }  
 total += count;  
 *// publishing the progress....* **if** (fileLength > 0) *// only if total length is known* publishProgress((**int**) (total \* 50 / fileLength));  
 output.write(data, 0, count);  
 }  
 } **catch** (Exception e) {  
 **return** e.toString();  
 } **finally** {  
 **try** {  
 **if** (output != **null**)  
 output.close();  
 **if** (input != **null**)  
 input.close();  
 } **catch** (IOException ignored) {  
 }  
 **if** (connection != **null**)  
 connection.disconnect();  
 }  
  
 **try** {  
 input = **new** FileInputStream(sUrl[1]);  
 **byte** data[] = **new byte**[4096];  
 **int** len = 0;  
 StringBuilder s = **new** StringBuilder();  
 **while** ((len = input.read(data)) != -1)  
 s.append(**new** String(data, 0, len));  
 JSONArray jsonArray = **new** JSONArray(s.toString());  
 Log.*d*(**"JSON"**, **"Length "** + String.*valueOf*(jsonArray.length()));  
 **for** (**int** i = 0; i < jsonArray.length(); ++i) {  
 JSONObject word\_row = (JSONObject)jsonArray.get(i);  
 String word = word\_row.getString(**"word"**);  
 String explanation = word\_row.getString(**"explanation"**);  
 **int** level = word\_row.optInt(**"level"**);  
 ContentValues cv = **new** ContentValues();  
 cv.put(**"word"**, word);  
 cv.put(**"explanation"**, explanation);  
 cv.put(**"level"**, level);  
 cv.put(**"modified\_time"**, **"2019-01-01 00:00:00"**);  
 **db**.insert(**"dict"**, **null**, cv);  
 publishProgress((**int**) (i \* 50 / jsonArray.length()) + 50);  
 }  
 } **catch** (Exception e) {  
 **return** e.toString();  
 } **finally** {  
 **try** {  
 input.close();  
 } **catch** (IOException ignored) {  
 }  
 }  
  
 **return null**;  
 }  
  
 @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 *// take CPU lock to prevent CPU from going off if the user  
 // presses the power button during download* PowerManager pm = (PowerManager) **context**.getSystemService(Context.***POWER\_SERVICE***);  
 **mWakeLock** = pm.newWakeLock(PowerManager.***PARTIAL\_WAKE\_LOCK***,  
 getClass().getName());  
 **mWakeLock**.acquire();  
 **mProgressDialog**.show();  
 }  
  
 @Override  
 **protected void** onProgressUpdate(Integer... progress) {  
 **super**.onProgressUpdate(progress);  
 *// if we get here, length is known, now set indeterminate to false* **mProgressDialog**.setIndeterminate(**false**);  
 **mProgressDialog**.setMax(100);  
 **mProgressDialog**.setProgress(progress[0]);  
 }  
  
 @Override  
 **protected void** onPostExecute(String result) {  
 **mWakeLock**.release();  
 **mProgressDialog**.dismiss();  
 **if** (result != **null**) {  
 Toast.*makeText*(**context**, **"Download error: "** + result, Toast.***LENGTH\_LONG***).show();  
 **mListener**.onFailed();  
 } **else** {  
 Toast.*makeText*(**context**, **"File downloaded"**, Toast.***LENGTH\_SHORT***).show();  
 **mListener**.onSuccess();  
 }  
 }  
  
 @Override  
 **protected void** finalize() **throws** Throwable {  
 **context** = **null**;  
 **mProgressDialog** = **null**;  
 **super**.finalize();  
 }  
}

1. WordListViewAdapter.java

**package** com.example.grandactivity;  
  
  
**import** android.app.AlertDialog;  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.support.constraint.ConstraintLayout;  
**import** android.view.LayoutInflater;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.PopupMenu;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.io.File;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** WordListViewAdapter **extends** BaseAdapter {  
 **private** Context **context**;  
 **private** List<String> **datas** = **new** ArrayList<String>();  
 **private** List<String> **expls** = **new** ArrayList<String>();  
 **private int show\_explanation**;  
 **private** DBOpenHandler **dbOpenHandler**;  
 **private** SQLiteDatabase **db**;  
 **private** ContentValues **cv**;  
 **private** AlertDialog **alertDialog** = **null**;  
 **private** AlertDialog.Builder **dialogBuilder** = **null**;  
 **private boolean alwaysShowExplanation**;  
  
 **public** WordListViewAdapter(Context context, **boolean** alwaysShowExplanation) {  
 **super**();  
 **this**.**context** = context;  
 **show\_explanation** = -1;  
 File file = **new** File(context.getFilesDir(), **"dict.db3"**);  
 **dbOpenHandler** = **new** DBOpenHandler(context, file.getAbsolutePath(), **null**, 1);  
 **db** = **dbOpenHandler**.getReadableDatabase();  
 **final** String create\_table\_sql = **"CREATE TABLE IF NOT EXISTS dict(\_id integer primary key autoincrement, word varchar(64) unique COLLATE NOCASE, explanation text, level int default 0, modified\_time timestamp)"**;  
 **db**.execSQL(create\_table\_sql);  
 **cv** = **new** ContentValues();  
 **this**.**alwaysShowExplanation** = alwaysShowExplanation;  
 }  
  
 */\*\* 添加item数据 \*/* **public void** addData(String text, String expl) {  
 **if** (**datas** != **null**) {  
 **datas**.add(text);  
 **expls**.add(expl);  
 }  
 }  
  
  
 **public void** clear() {  
 **datas**.clear();  
 **expls**.clear();  
 }  
  
 @Override  
 **public int** getCount() {  
 **if** (**datas** == **null**)  
 **return** 0;  
 **return datas**.size();  
 }  
  
 @Override  
 **public** Object getItem(**int** position) {  
 **return datas**.get(position);  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 @Override  
 **public** View getView(**final int** position, View convertView, ViewGroup parent) {  
 **if** (position != **show\_explanation** && **alwaysShowExplanation** == **false**) {  
 convertView = View.*inflate*(**context**, R.layout.***list\_item***, **null**);  
 String text = **datas**.get(position);  
 ((TextView) convertView.findViewById(R.id.***title***)).setText(text);  
 } **else** {  
 convertView = View.*inflate*(**context**, R.layout.***list\_item\_with\_explanation***, **null**);  
 String word = **datas**.get(position);  
 String expl = **expls**.get(position);  
 ((TextView) convertView.findViewById(R.id.***title***)).setText(word);  
 ((TextView) convertView.findViewById(R.id.***explanation***)).setText(expl);  
 }  
 convertView.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (**show\_explanation** == position) {  
 **show\_explanation** = -1;  
 } **else** {  
 **show\_explanation** = position;  
 }  
 notifyDataSetChanged();  
 }  
 });  
 convertView.setOnLongClickListener(**new** View.OnLongClickListener() {  
 @Override  
 **public boolean** onLongClick(**final** View v) {  
 PopupMenu popup = **new** PopupMenu(**context**, v.findViewById(R.id.***title***));  
 popup.getMenuInflater().inflate(R.menu.***menu\_pop***, popup.getMenu());  
 popup.setOnMenuItemClickListener(**new** PopupMenu.OnMenuItemClickListener() {  
 @Override  
 **public boolean** onMenuItemClick(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** R.id.***delete***:  
 **db**.delete(**"dict"**, **"word=?"**, **new** String[] {**datas**.get(position)});  
 **datas**.remove(position);  
 **expls**.remove(position);  
 notifyDataSetChanged();  
 **break**;  
 **case** R.id.***modify***:  
 ConstraintLayout modifyWordForm = (ConstraintLayout) LayoutInflater.*from*(**context**)  
 .inflate(R.layout.***modify\_word\_dialog***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(**context**);  
 **alertDialog** = **dialogBuilder** .setTitle(**"添加单词"**)  
 .setView(modifyWordForm)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 String explanation = ((EditText)**alertDialog**.findViewById(R.id.***editExplanation***)).getText().toString();  
 String level = ((EditText)**alertDialog**.findViewById(R.id.***editLevel***)).getText().toString();  
 **cv**.clear();  
 **cv**.put(**"explanation"**, explanation);  
 **cv**.put(**"level"**, level);  
 **db**.update(**"dict"**, **cv**, **"word=?"**, **new** String[] {**datas**.get(position)});  
 **expls**.set(position, explanation);  
 notifyDataSetChanged();  
 }  
 })  
 .create();  
 **alertDialog**.show();  
 **break**;  
 }  
 **return true**;  
 }  
 });  
 popup.show();  
 **return true**;  
 }  
 });  
 **return** convertView;  
 }  
  
}

1. DictContentProvider.java

**package** com.example.grandactivity;  
  
**import** android.content.ContentProvider;  
**import** android.content.ContentValues;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.net.Uri;  
**import** android.util.Log;  
  
**import** java.io.File;  
  
**public class** DictContentProvider **extends** ContentProvider {  
  
 **private** DBOpenHandler **dbOpenHandler**;  
 **private** SQLiteDatabase **db**;  
  
 **public** DictContentProvider() {  
 }  
  
 @Override  
 **public int** delete(Uri uri, String selection, String[] selectionArgs) {  
 *// Implement this to handle requests to delete one or more rows.* **throw new** UnsupportedOperationException(**"Not yet implemented"**);  
 }  
  
 @Override  
 **public** String getType(Uri uri) {  
 *//* ***TODO: Implement this to handle requests for the MIME type of the data*** *// at the given URI.* **throw new** UnsupportedOperationException(**"Not yet implemented"**);  
 }  
  
 @Override  
 **public** Uri insert(Uri uri, ContentValues values) {  
 *//* ***TODO: Implement this to handle requests to insert a new row.*** String word = (String)values.get(**"word"**);  
 String explanation = (String)values.get(**"explanation"**);  
 String level = (String)values.get(**"level"**);  
 **boolean** overwrite = (**boolean**)values.get(**"overwrite"**);  
 ContentValues cv = **new** ContentValues();  
 cv.put(**"word"**, word);  
 cv.put(**"explanation"**, explanation);  
 cv.put(**"level"**, level);  
 cv.put(**"modified\_time"**, **"2019-01-01 00:00:00"**);  
 **db**.insert(**"dict"**, **null**, cv);  
 **if** (overwrite) {  
 **db**.update(**"dict"**, cv, **"word=?"**, **new** String[]{word});  
 }  
 **return** uri;  
 }  
  
 @Override  
 **public boolean** onCreate() {  
 File file = **new** File(getContext().getFilesDir(), **"dict.db3"**);  
 **dbOpenHandler** = **new** DBOpenHandler(getContext(), file.getAbsolutePath(), **null**, 1);  
 **db** = **dbOpenHandler**.getReadableDatabase();  
 **final** String create\_table\_sql = **"CREATE TABLE IF NOT EXISTS dict(\_id integer primary key autoincrement, word varchar(64) unique COLLATE NOCASE, explanation text, level int default 0, modified\_time timestamp)"**;  
 **db**.execSQL(create\_table\_sql);  
 **return true**;  
 }  
  
 @Override  
 **public** Cursor query(Uri uri, String[] projection, String selection,  
 String[] selectionArgs, String sortOrder) {  
 String table = **"dict"**;  
 String groupBy = **""**;  
 String having = **""**;  
 Cursor cursor = **db**.query(table, projection, selection, selectionArgs, groupBy, having, sortOrder);  
 **return** cursor;  
 }  
  
 @Override  
 **public int** update(Uri uri, ContentValues values, String selection,  
 String[] selectionArgs) {  
 *//* ***TODO: Implement this to handle requests to update one or more rows.* throw new** UnsupportedOperationException(**"Not yet implemented"**);  
 }  
}

1. activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout 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"  
 tools:context=".MainActivity"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/colorPrimary"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="?attr/actionBarTheme"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:subtitle="@string/subtitle"  
 app:title="@string/title"  
 app:subtitleTextAppearance="@style/Toolbar.SubtitleText"  
 app:titleTextAppearance="@style/Toolbar.TitleText"** />  
  
 <**HorizontalScrollView  
 android:id="@+id/scrollView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@+id/toolbar"**>  
  
 <**LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toTopOf="@+id/scrollView"  
 android:orientation="horizontal"**/>  
 </**HorizontalScrollView**>  
  
 <**ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="8dp"  
 android:indeterminate="true"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/scrollView"** />  
  
 <**ListView  
 android:id="@+id/listWords"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/scrollView"** />  
  
</**android.support.constraint.ConstraintLayout**>

1. list\_item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 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"**>  
  
 <**TextView  
 android:id="@+id/title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="30dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="30dp"  
 android:layout\_marginBottom="8dp"  
 android:text="TextView"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
</**android.support.constraint.ConstraintLayout**>

1. add\_word\_dialog.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout  
 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"**>  
  
 <**TableRow  
 android:id="@+id/t1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**>  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="单词："  
 android:textSize="18sp"** />  
  
 <**EditText  
 android:id="@+id/editWord"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:hint="请填写单词"** />  
  
 </**TableRow**>  
  
 <**TableRow  
 android:id="@+id/t2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/t1"**>  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="解释："  
 android:textSize="18sp"** />  
  
 <**EditText  
 android:id="@+id/editExplanation"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="160dp"  
 android:ems="10"  
 android:gravity="start|top"  
 android:inputType="textMultiLine"  
 android:hint="请填写解释"** />  
 </**TableRow**>  
  
 <**TableRow  
 android:id="@+id/t3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/t2"**>  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="级别："  
 android:textSize="18sp"** />  
  
 <**EditText  
 android:id="@+id/editLevel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:gravity="start|top"  
 android:inputType="number"  
 android:hint="请填写难度（0-8）"** />  
  
 </**TableRow**>  
  
 <**TableRow  
 android:id="@+id/t4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/t3"**>  
  
 <**TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="是否覆盖："  
 android:textSize="18sp"** />  
  
 <**CheckBox  
 android:id="@+id/checkOverwrite"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:checked="true"**/>  
  
 </**TableRow**>  
</**android.support.constraint.ConstraintLayout**>

…

【背单词程序】

1. MainActivity.java

**package** com.example.grandwordremember;  
  
**import** android.content.ContentResolver;  
**import** android.content.ContentValues;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.net.Uri;  
**import** android.support.constraint.ConstraintLayout;  
**import** android.support.v7.app.ActionBar;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.Log;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.CheckBox;  
**import** android.widget.EditText;  
**import** android.widget.TableLayout;  
**import** android.widget.Toast;  
  
**import** java.io.File;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** Toolbar **mToolbar**;  
 **private** ActionBar **actionBar**;  
 **private** AlertDialog **alertDialog** = **null**;  
 **private** AlertDialog.Builder **dialogBuilder** = **null**;  
 **private** ContentResolver **resolver**;  
 **private** ContentValues **cv**;  
 **private final** Uri **uri** = Uri.*parse*(**"content://com.africamonkey.granddict/"**);  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_main***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 Intent intent\_home = **new** Intent(**this**,MainActivity.**class**);  
 startActivity(intent\_home);  
 **break**;  
 **case** R.id.***enterTest***:  
 Intent intent\_test = **new** Intent(**this**,TestActivity.**class**);  
 startActivity(intent\_test);  
 **break**;  
 **case** R.id.***addWord***:  
 ConstraintLayout addWordForm = (ConstraintLayout) getLayoutInflater()  
 .inflate(R.layout.***add\_word\_dialog***, **null**);  
 **dialogBuilder** = **new** AlertDialog.Builder(MainActivity.**this**);  
 **alertDialog** = **dialogBuilder** .setTitle(**"添加单词"**)  
 .setView(addWordForm)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 }  
 })  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 String word = ((EditText)**alertDialog**.findViewById(R.id.***editWord***)).getText().toString();  
 String explanation = ((EditText)**alertDialog**.findViewById(R.id.***editExplanation***)).getText().toString();  
 String level = ((EditText)**alertDialog**.findViewById(R.id.***editLevel***)).getText().toString();  
 **boolean** overwrite = ((CheckBox)**alertDialog**.findViewById(R.id.***checkOverwrite***)).isChecked();  
 **cv**.clear();  
 **cv**.put(**"word"**, word);  
 **cv**.put(**"explanation"**, explanation);  
 **cv**.put(**"level"**, level);  
 **cv**.put(**"overwrite"**, overwrite);  
 Uri newUri = **resolver**.insert(**uri**, **cv**); *// null--错误* **if** (newUri == **null**) {  
 Toast.*makeText*(MainActivity.**this**, **"Failed"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** {  
 Toast.*makeText*(MainActivity.**this**, **"Success"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 })  
 .create();  
 **alertDialog**.show();  
 **break**;  
 **case** R.id.***studyCount***:  
 Intent intent\_stat = **new** Intent(**this**,StatActivity.**class**);  
 startActivity(intent\_stat);  
 **break**;  
 **case** R.id.***findWord***:  
 **break**;  
 **case** R.id.***preference***:  
 **break**;  
 }  
 **return true**;  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *//We have to tell the activity where the toolbar is* **mToolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**mToolbar**);  
 **actionBar** = getSupportActionBar();  
 *//Display home with the "up" arrow indicator* **actionBar**.setHomeButtonEnabled(**false**);  
 *//actionBar.setDisplayHomeAsUpEnabled(true);* **resolver** = getContentResolver();  
 **cv** = **new** ContentValues();  
 }  
}

1. TestActivity.java

**package** com.example.grandwordremember;  
  
**import** android.content.ContentResolver;  
**import** android.content.ContentValues;  
**import** android.content.Intent;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.net.Uri;  
**import** android.support.v7.app.ActionBar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.support.v7.widget.Toolbar;  
**import** android.util.Log;  
**import** android.view.ContextMenu;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
  
**import** java.io.File;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** TestActivity **extends** AppCompatActivity {  
 **private** Toolbar **mToolbar**;  
 **private** ActionBar **actionBar**;  
 **private** ContentResolver **resolver**;  
 **private final** Uri **uri** = Uri.*parse*(**"content://com.africamonkey.granddict/"**);  
 **private int word\_to\_test**;  
 **private** ArrayList<String> **word**, **explanation**;  
 **private** ArrayList<Integer> **level**;  
 **private** TestListViewAdapter **adapter**;  
 **private** ListView **listView**;  
 **private** DBOpenHandler **dbOpenHandler**;  
 **private** SQLiteDatabase **db**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_test***);  
 **resolver** = getContentResolver();  
 **word\_to\_test** = 10;  
  
 *//We have to tell the activity where the toolbar is* **mToolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**mToolbar**);  
 **actionBar** = getSupportActionBar();  
 *//Display home with the "up" arrow indicator* **actionBar**.setHomeButtonEnabled(**true**);  
 **actionBar**.setDisplayHomeAsUpEnabled(**true**);  
  
 File file = **new** File(TestActivity.**this**.getFilesDir(), **"words.db3"**);  
 **dbOpenHandler** = **new** DBOpenHandler(TestActivity.**this**, file.getAbsolutePath(), **null**, 1);  
 **db** = **dbOpenHandler**.getReadableDatabase();  
 **final** String create\_table\_sql = **"CREATE TABLE if not exists 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)"**;  
 **db**.execSQL(create\_table\_sql);  
  
 **adapter** = **new** TestListViewAdapter(TestActivity.**this**, **db**);  
 **listView** = (ListView)findViewById(R.id.***testListView***);  
 **listView**.setAdapter(**adapter**);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 Intent intent\_home = **new** Intent(**this**,MainActivity.**class**);  
 startActivity(intent\_home);  
 **break**;  
 **case** R.id.***startTest***:  
 Cursor local = **db**.query(**"words"**, **new** String[] {**"max(word)"**}, **""**, **new** String[] {}, **""**, **""**, **""**);  
 String nowmax = **""**;  
 **if** (local != **null** && local.moveToNext() && local.getString(0) != **null**) {  
 nowmax = local.getString(0);  
 }  
 Cursor c = **resolver**.query(**uri**, **new** String[] {**"word"**, **"explanation"**, **"level"**}, **"word > ?"**, **new** String[] {nowmax}, **"word limit "** + String.*valueOf*(**word\_to\_test**));  
 **word** = **new** ArrayList<String>();  
 **explanation** = **new** ArrayList<String>();  
 **level** = **new** ArrayList<Integer>();  
 **adapter**.clear();  
 **while** (c.moveToNext()) {  
 **word**.add(c.getString(0));  
 **explanation**.add(c.getString(1));  
 **level**.add(Integer.*valueOf*(c.getInt(2)));  
 }  
 **int** size = **word**.size();  
 **if** (size < 10) {  
 Toast.*makeText*(**this**, **"请添加至少10个单词，并在偏好设置中设置每次测试个数至少为10。"**, Toast.***LENGTH\_SHORT***).show();  
 **break**;  
 }  
 **for** (**int** i = 0; i < size; ++i) {  
 **int** a1 = i, b1 = i, c1 = i, d1 = i, temp = -1;  
 **int** rnd = (**int**)(Math.*random*() \* 4);  
 **while** (b1 == a1) b1 = (**int**)(Math.*random*() \* size);  
 **while** (c1 == a1 || c1 == b1) c1 = (**int**)(Math.*random*() \* size);  
 **while** (d1 == a1 || d1 == b1 || d1 == c1) d1 = (**int**)(Math.*random*() \* size);  
 **switch** (rnd) {  
 **case** 0:  
 temp = 0;  
 **break**;  
 **case** 1:  
 temp = a1;  
 a1 = b1;  
 b1 = temp;  
 temp = 1;  
 **break**;  
 **case** 2:  
 temp = a1;  
 a1 = c1;  
 c1 = temp;  
 temp = 2;  
 **break**;  
 **case** 3:  
 temp = a1;  
 a1 = d1;  
 d1 = temp;  
 temp = 3;  
 **break**;  
 **default**:  
 temp = 0;  
 **break**;  
 }  
 **adapter**.addData(**new** Problem(**word**.get(i), **level**.get(i), **new** String[] {**explanation**.get(a1), **explanation**.get(b1), **explanation**.get(c1), **explanation**.get(d1)}, temp));  
 }  
 **adapter**.notifyDataSetChanged();  
 **break**;  
 **default**:  
 **break**;  
 }  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_test***, menu);  
 **return true**;  
 }  
}

1. StatActivity.java

**package** com.example.grandwordremember;  
  
**import** android.content.Intent;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.support.v7.app.ActionBar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.TableLayout;  
**import** android.widget.TextView;  
  
**import** java.io.File;  
  
**public class** StatActivity **extends** AppCompatActivity {  
 **private** Toolbar **mToolbar**;  
 **private** ActionBar **actionBar**;  
 **private** TableLayout **tableLayout**;  
 **private** DBOpenHandler **dbOpenHandler**;  
 **private** SQLiteDatabase **db**;  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 Intent intent\_home = **new** Intent(**this**,MainActivity.**class**);  
 startActivity(intent\_home);  
 **break**;  
 **default**:  
 **break**;  
 }  
 **return true**;  
 }  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_stat***);  
  
 *//We have to tell the activity where the toolbar is* **mToolbar** = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(**mToolbar**);  
 **actionBar** = getSupportActionBar();  
 *//Display home with the "up" arrow indicator* **actionBar**.setHomeButtonEnabled(**true**);  
 **actionBar**.setDisplayHomeAsUpEnabled(**true**);  
  
 File file = **new** File(StatActivity.**this**.getFilesDir(), **"words.db3"**);  
 **dbOpenHandler** = **new** DBOpenHandler(StatActivity.**this**, file.getAbsolutePath(), **null**, 1);  
 **db** = **dbOpenHandler**.getReadableDatabase();  
 **final** String create\_table\_sql = **"CREATE TABLE if not exists 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)"**;  
 **db**.execSQL(create\_table\_sql);  
  
 **tableLayout** = (TableLayout)findViewById(R.id.***tableLayout***);  
 Cursor c = **db**.query(**"words"**, **new** String[] {**"word"**, **"level"**, **"test\_count"**, **"correct\_count"**}, **""**, **new** String[] {}, **""**, **""**, **"word"**);  
 **while** (c.moveToNext()) {  
 View view = View.*inflate*(StatActivity.**this**, R.layout.***stat\_table\_row***, **null**);  
 TextView t1 = (TextView)view.findViewById(R.id.***content1***);  
 TextView t2 = (TextView)view.findViewById(R.id.***content2***);  
 TextView t3 = (TextView)view.findViewById(R.id.***content3***);  
 TextView t4 = (TextView)view.findViewById(R.id.***content4***);  
 **if** (c.getString(0) != **null**) {  
 t1.setText(c.getString(0));  
 t2.setText(String.*valueOf*(c.getInt(1)));  
 t3.setText(String.*valueOf*(c.getInt(2)));  
 t4.setText(String.*valueOf*(c.getInt(3)));  
 **tableLayout**.addView(view);  
 }  
 }  
 }  
}

1. TestListViewAdapter.java

**package** com.example.grandwordremember;  
  
**import** android.content.ContentValues;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.graphics.Color;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.Button;  
**import** android.widget.RadioButton;  
**import** android.widget.RadioGroup;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
  
**public class** TestListViewAdapter **extends** BaseAdapter {  
 **private** ArrayList<Problem> **data**;  
 **private** Context **context**;  
 **private boolean display\_answer**;  
 **private** SQLiteDatabase **db**;  
  
 **public** TestListViewAdapter(Context context, SQLiteDatabase db) {  
 **super**();  
 **this**.**context** = context;  
 **data** = **new** ArrayList<Problem>();  
 **display\_answer** = **false**;  
 **this**.**db** = db;  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 **if** (position < **data**.size()) {  
 convertView = View.*inflate*(**context**, R.layout.***list\_item***, **null**);  
 ((TextView) convertView.findViewById(R.id.***word***)).setText(**data**.get(position).**word**);  
 RadioGroup group = convertView.findViewById(R.id.***radio\_group***);  
 group.setTag(position);  
 group.setOnCheckedChangeListener(**new** RadioGroup.OnCheckedChangeListener() {  
 @Override  
 **public void** onCheckedChanged(RadioGroup group, **int** checkedId) {  
 Problem tmp = **data**.get((**int**)group.getTag());  
 **if** (checkedId != -1) {  
 **switch** (checkedId) {  
 **case** R.id.***radioButton\_A***:  
 tmp.**selected** = 0;  
 **break**;  
 **case** R.id.***radioButton\_B***:  
 tmp.**selected** = 1;  
 **break**;  
 **case** R.id.***radioButton\_C***:  
 tmp.**selected** = 2;  
 **break**;  
 **case** R.id.***radioButton\_D***:  
 tmp.**selected** = 3;  
 **break**;  
 }  
 }  
 **data**.set((**int**)group.getTag(), tmp);  
 }  
 });  
 **switch** (**data**.get(position).**selected**) {  
 **case** 0:  
 group.check(R.id.***radioButton\_A***);  
 **break**;  
 **case** 1:  
 group.check(R.id.***radioButton\_B***);  
 **break**;  
 **case** 2:  
 group.check(R.id.***radioButton\_C***);  
 **break**;  
 **case** 3:  
 group.check(R.id.***radioButton\_D***);  
 **break**;  
 }  
 **if** (**display\_answer**) {  
 **switch** (**data**.get(position).**answer**) {  
 **case** 0:  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_A***)).setBackgroundResource(R.color.***colorYellow***);  
 **break**;  
 **case** 1:  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_B***)).setBackgroundResource(R.color.***colorYellow***);  
 **break**;  
 **case** 2:  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_C***)).setBackgroundResource(R.color.***colorYellow***);  
 **break**;  
 **case** 3:  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_D***)).setBackgroundResource(R.color.***colorYellow***);  
 **break**;  
 }  
 }  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_A***)).setText(**data**.get(position).**options**[0]);  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_B***)).setText(**data**.get(position).**options**[1]);  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_C***)).setText(**data**.get(position).**options**[2]);  
 ((RadioButton) convertView.findViewById(R.id.***radioButton\_D***)).setText(**data**.get(position).**options**[3]);  
 } **else** {  
 convertView = View.*inflate*(**context**, R.layout.***list\_submit***, **null**);  
 Button btn = convertView.findViewById(R.id.***submit***);  
 **if** (**display\_answer**) btn.setVisibility(View.***INVISIBLE***);  
 **else** btn.setVisibility(View.***VISIBLE***);  
 btn.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **for** (**int** i = 0; i < **data**.size(); ++i) {  
 Cursor c = **db**.query(**"words"**, **new** String[] {**"test\_count"**, **"correct\_count"**}, **"word = ?"**, **new** String[] {**data**.get(i).**word**}, **null**, **null**, **null**);  
 **int** test\_count = 1, correct\_count = **data**.get(i).**selected** == **data**.get(i).**answer** ? 1 : 0;  
 **if** (c != **null** && c.moveToNext()) {  
 test\_count += c.getInt(0);  
 correct\_count += c.getInt(1);  
 }  
 ContentValues cv = **new** ContentValues();  
 cv.put(**"word"**, **data**.get(i).**word**);  
 cv.put(**"level"**, **data**.get(i).**level**);  
 cv.put(**"test\_count"**, test\_count);  
 cv.put(**"correct\_count"**, correct\_count);  
 cv.put(**"last\_test\_time"**, **"2019-01-01 12:00:00"**);  
 **db**.insert(**"words"**, **null**, cv);  
 **db**.update(**"words"**, cv, **"word = ?"**, **new** String[] {**data**.get(i).**word**});  
 }  
 **display\_answer** = **true**;  
 notifyDataSetChanged();  
 }  
 });  
 }  
 **return** convertView;  
 }  
  
 @Override  
 **public long** getItemId(**int** position) {  
 **return** position;  
 }  
  
 @Override  
 **public** Object getItem(**int** position) {  
 **if** (position < **data**.size())  
 **return data**.get(position);  
 **else  
 return null**;  
 }  
  
 @Override  
 **public int** getCount() {  
 **if** (**data**.size() == 0) **return** 0;  
 **return data**.size() + 1;  
 }  
  
 **public void** addData(Problem p) {  
 **data**.add(p);  
 }  
  
 **public void** clear() {  
 **data**.clear();  
 }  
}

1. Activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout 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"  
 tools:context=".MainActivity"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/colorPrimary"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="?attr/actionBarTheme"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:subtitle="@string/fastRememberMethod"  
 app:title="@string/studyBullyRememberWords"** />  
  
 <**View  
 android:id="@+id/view"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:background="@color/colorBackground"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/toolbar"** />  
  
 <**ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="@id/view"  
 app:layout\_constraintEnd\_toEndOf="@id/view"  
 app:layout\_constraintStart\_toStartOf="@id/view"  
 app:layout\_constraintTop\_toTopOf="@id/view"  
 app:srcCompat="@drawable/rem"** />  
  
</**android.support.constraint.ConstraintLayout**>

1. Activity\_test.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout 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"  
 tools:context=".TestActivity"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/colorPrimary"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="?attr/actionBarTheme"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:subtitle="@string/wordTest"  
 app:title="@string/studyBullyRememberWords"** />  
  
 <**ListView  
 android:id="@+id/testListView"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/toolbar"** />  
</**android.support.constraint.ConstraintLayout**>

1. Activity\_stat.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout 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"  
 tools:context=".StatActivity"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/colorPrimary"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="?attr/actionBarTheme"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**ScrollView  
 android:id="@+id/scrollView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/toolbar"**>  
  
 <**TableLayout  
 android:id="@+id/tableLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_margin="20dp"  
 android:stretchColumns="\*"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/scrollView"**>  
  
 <**TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**TextView  
 android:id="@+id/head1"  
 android:text="单词"  
 android:textColor="@color/colorBlack"  
 android:textSize="18sp"  
 android:textStyle="bold"**></**TextView**>  
  
 <**TextView  
 android:id="@+id/head2"  
 android:text="级别"  
 android:textColor="@color/colorBlack"  
 android:textSize="18sp"  
 android:textStyle="bold"**></**TextView**>  
  
 <**TextView  
 android:id="@+id/head3"  
 android:text="测试"  
 android:textColor="@color/colorBlack"  
 android:textSize="18sp"  
 android:textStyle="bold"**></**TextView**>  
  
 <**TextView  
 android:id="@+id/head4"  
 android:text="正确"  
 android:textColor="@color/colorBlack"  
 android:textSize="18sp"  
 android:textStyle="bold"**></**TextView**>  
 </**TableRow**>  
  
 </**TableLayout**>  
  
 </**ScrollView**>  
</**android.support.constraint.ConstraintLayout**>

…

【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：  
无

…

【实验体会】

为了实现进度条功能，我特意去网上搜了一下怎么做，他们用安卓的AsyncTask就能实现。我在实现的过程中也遇到了很多坑，比如在doInBackground中执行adapter.notifyDataChange。经查，这实际上是执行更新UI的过程，安卓为了线程安全，不允许非主线程更新UI。AsyncTask还提供了3个可以更新UI（即在主线程中执行）的函数，onPreExecute，onProgressUpdate和onPostExecute。在这三个函数中，我们可以在里面更新UI。后来我尝试在doInBackground中添加数据，onProgressUpdate中执行notifyDataChange（ListView的UI更新），由于执行notifyDataChange时会检查缓存元素个数是否等于列表个数，不一样就报错，所以添加数据和UI更新必须是同步的，不能是异步的。最终我选择在doInBackground中添加数据，数据全部添加完后，在onPostExecute中执行adapter.notifyDataChange。

另外，我们希望下载完字典就立刻将字典加载到ListView，那我们要监控下载线程什么时候结束啊！事实上，我们可以创建一个Listener，下载线程执行到onPostExecute时调用一下回调函数，我们就能在主线程知道它下载完了，然后执行加载ListView这个任务。

【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)