院 系 数据科学与计算机学院 学 号 16337341 姓 名 朱志儒

【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。

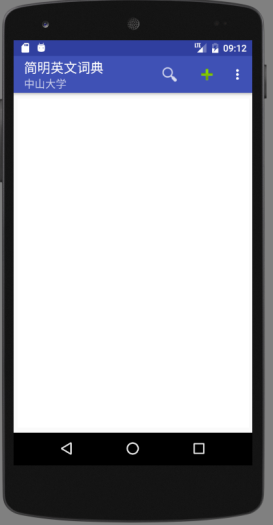
（d）可以先看一下老师的类设计（在本报告末尾）

【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**：

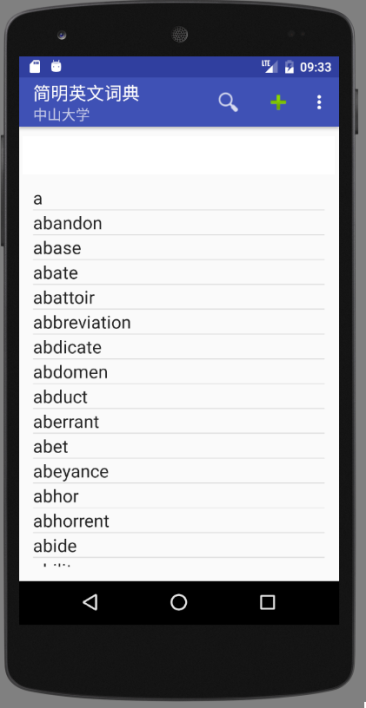
(1)AndroidManifest.xml中加入允许互联网访问：  
<**uses-permission android:name="android.permission.INTERNET"**/>

(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）

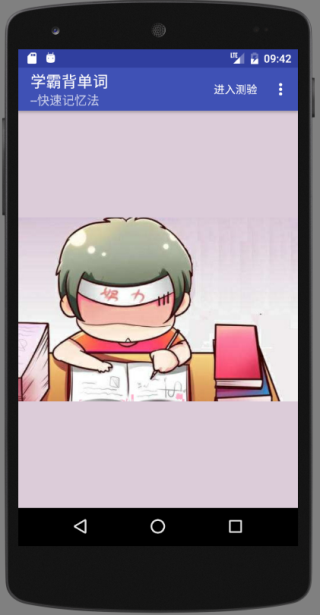
**参考源码**： SqliteCv、FirstProvider

**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

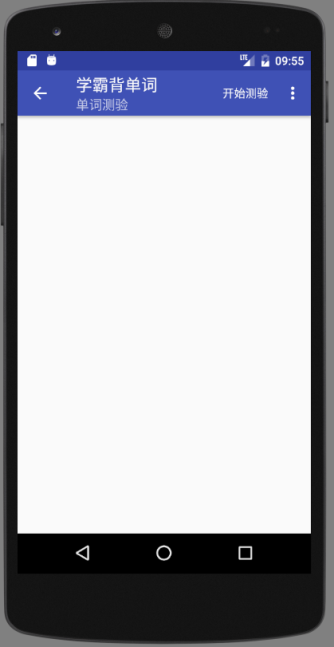
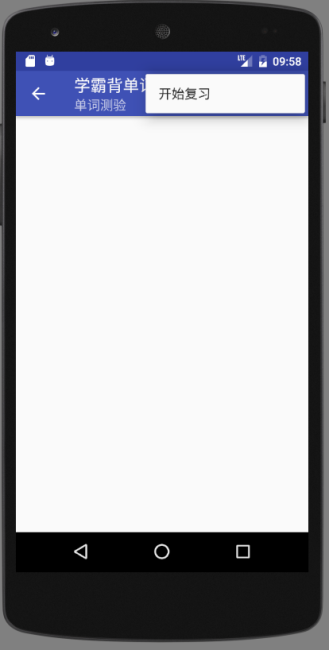
项目：GrandWordRemember

5、做一个主界面(MainActivity)

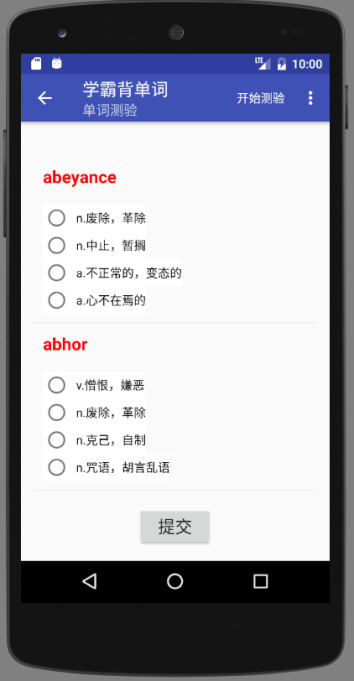
**参考资源**： rem.jpg

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。

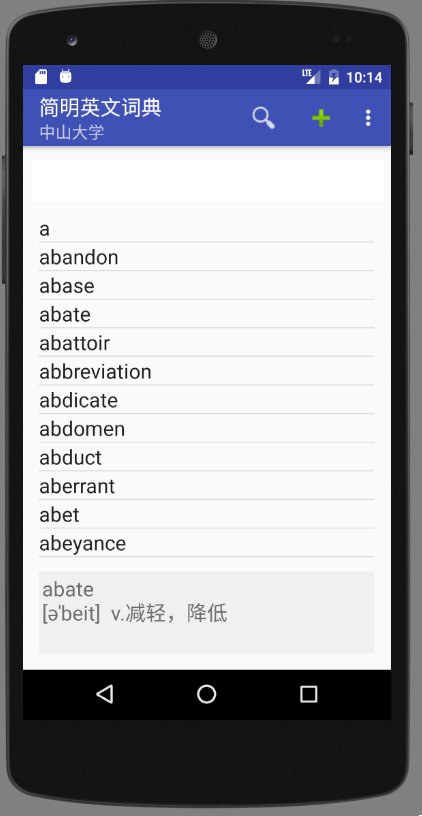
\* Context在Activity中Context可以用this或ActivityName.this取得，在事件处理程序中可以用ActivityName.this取得。

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

（三）词典扩展功能设计

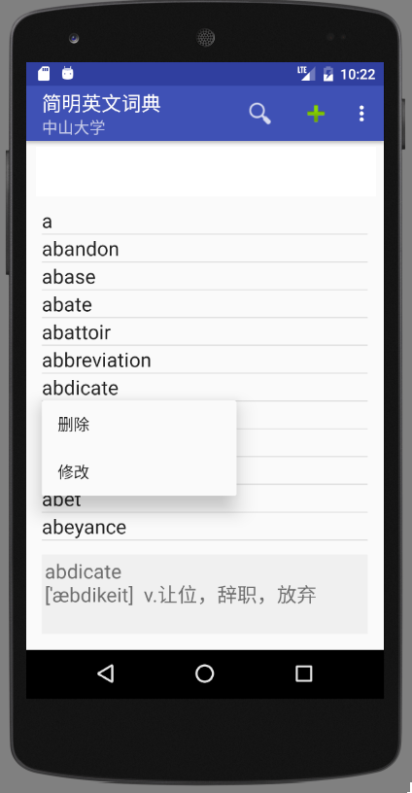
项目：GrandDictionary

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



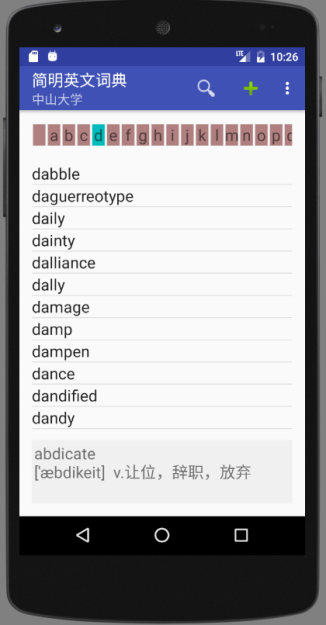
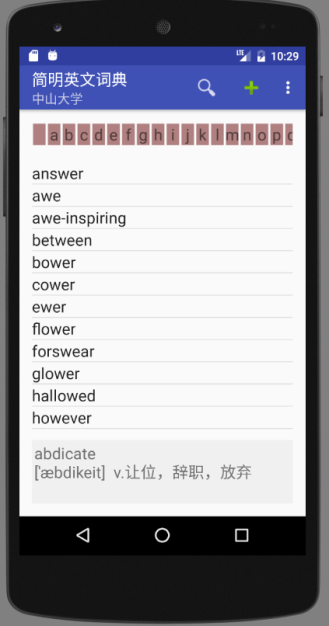
**参考源码**：ListViewSimple，SqliteCv

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

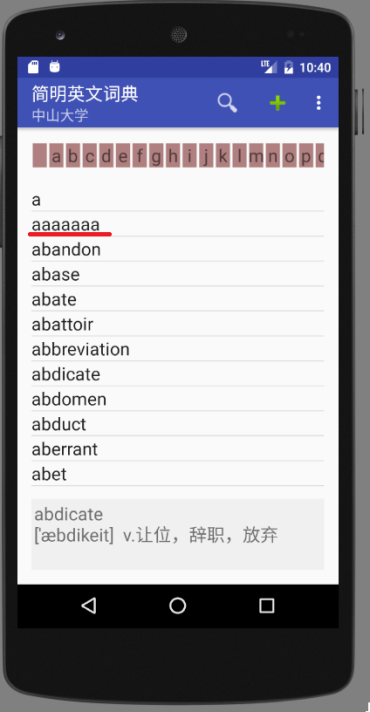
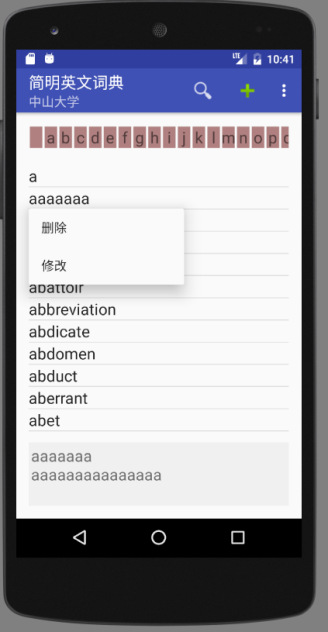
**功能说明**：在列表上方增加一个字母表，点击时只显示以这些字母开头的单词；点击查询图标可以查询包含输入子串的单词。

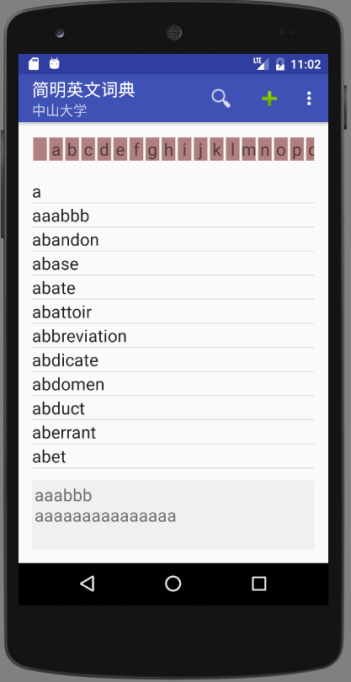
**参考源码**：DynamicView,HorizontalScrollView（自己上网查找使用方法）、SqliteCv

**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

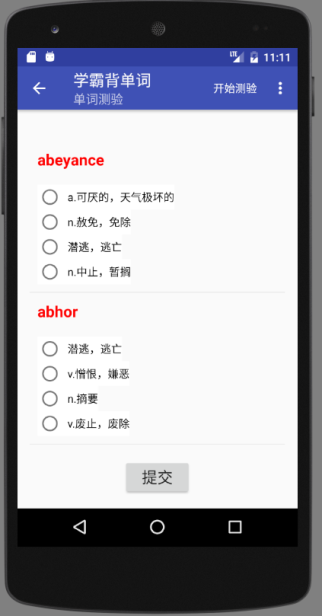
db.execSQL(**"CREATE TABLE words(\_id integer primary key autoincrement,word varchar(64) unique,level int default 0, test\_count int default 0, correct\_count int default 0,last\_test\_time timestamp)"**);

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。

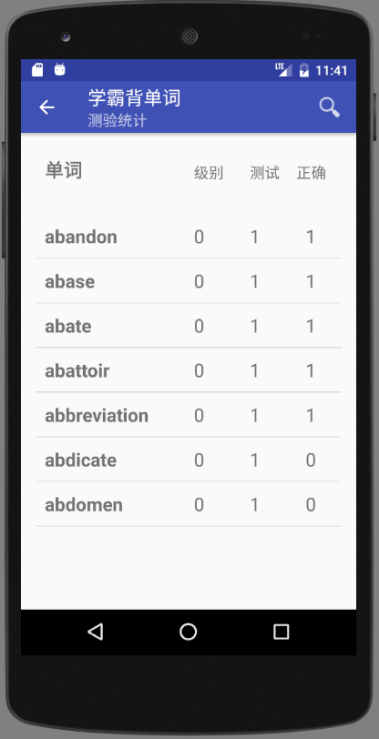
  

**参考源码**：SpinnerCustom（改为ListView），TestAdapter.java

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

<http://www.voidcn.com/article/p-xdrauljg-gn.html> *(*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

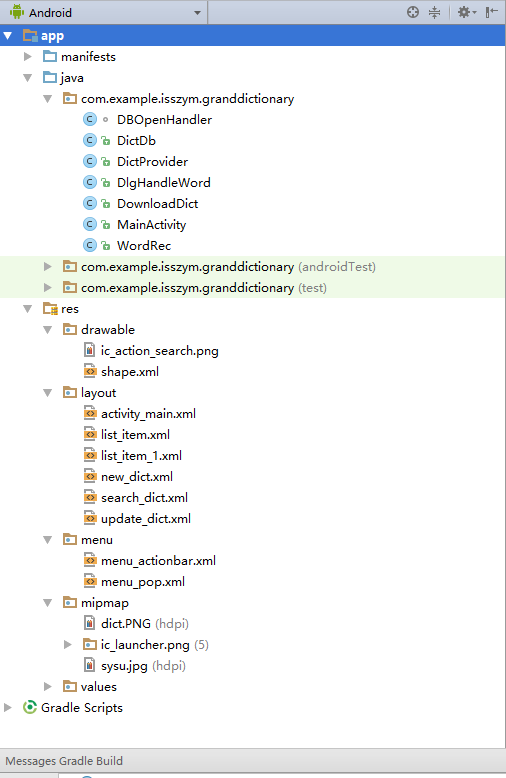
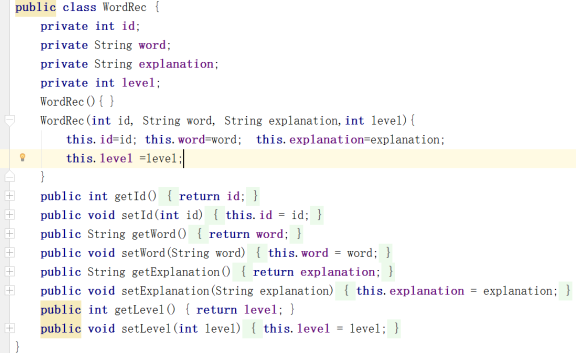
（e）利用中文解释，选择英文单词

（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)

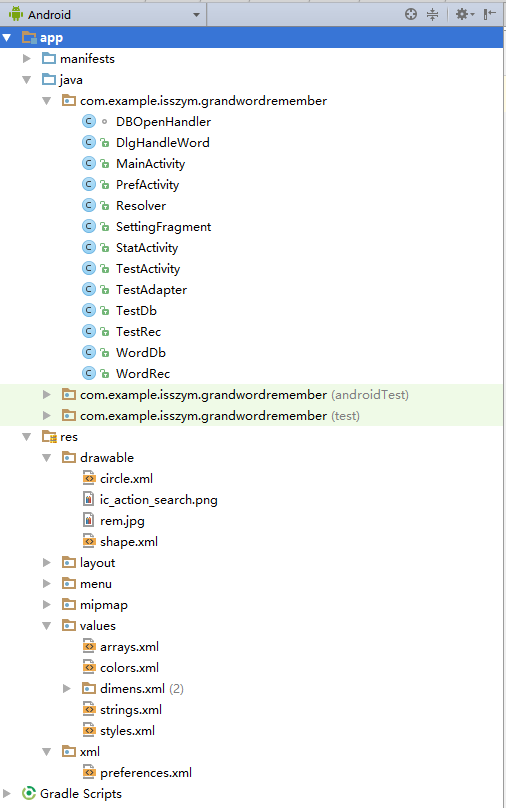
 

DbOpenHandler：打开数据库类 DictDb:所有数据库操作类

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）

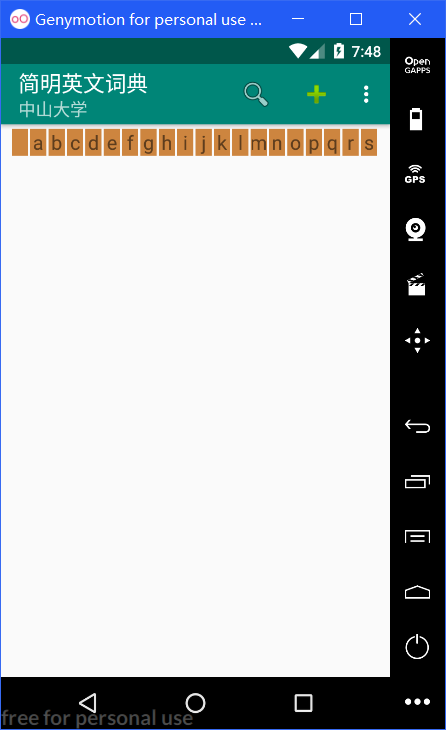
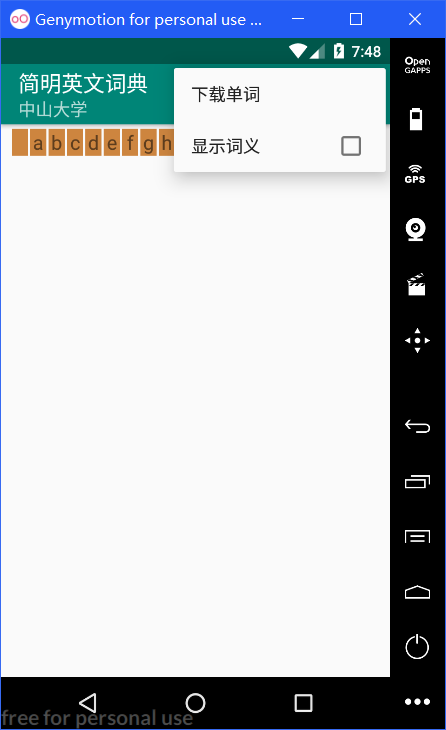
TestDb：本地数据库操作类 TestRec：本地统计记录类 WordDb：（远程）词典操作类

WordRec：单词记录类

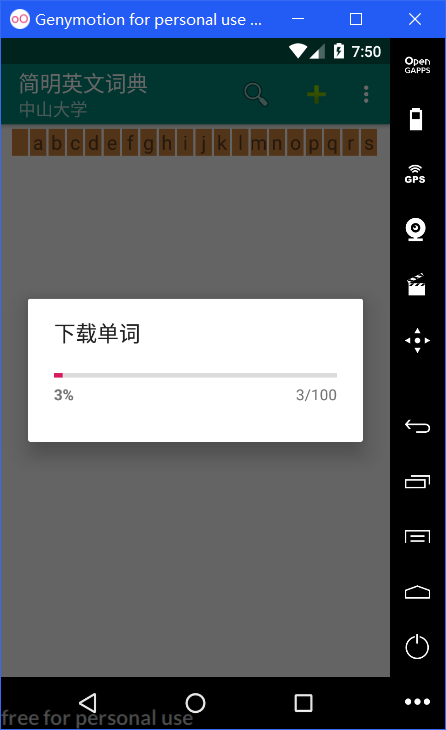
*完成后程序运行截屏及简要说明：*

【词典程序】

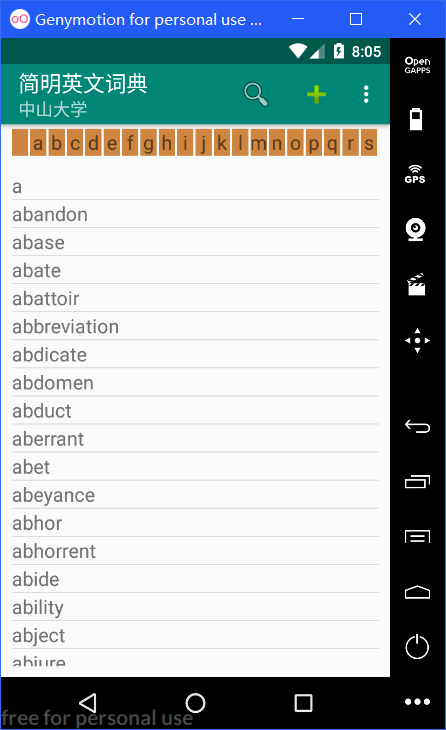
（1）左为第一次运行程序时的界面，右为点击右上角菜单后的界面

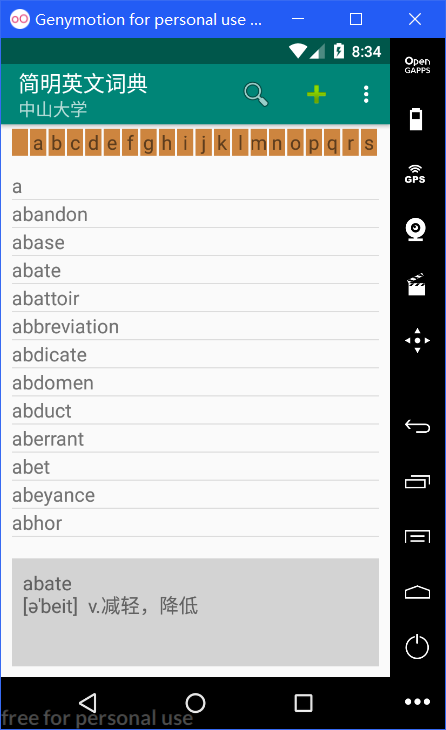
（2）点击下载菜单完成单词下载



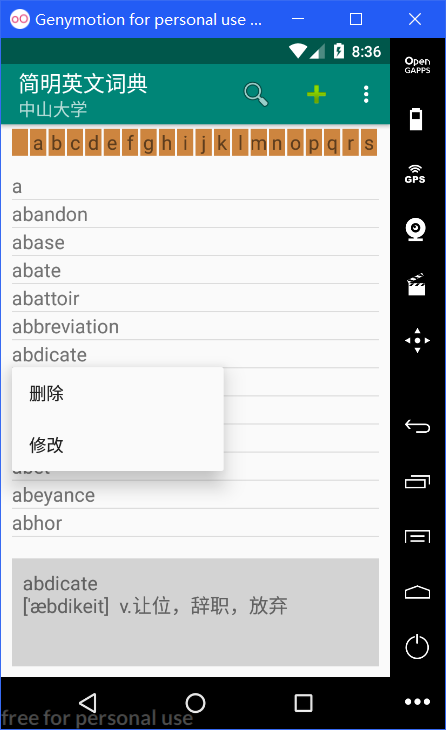
（3）下载完成后，把数据库中的数据显示出来



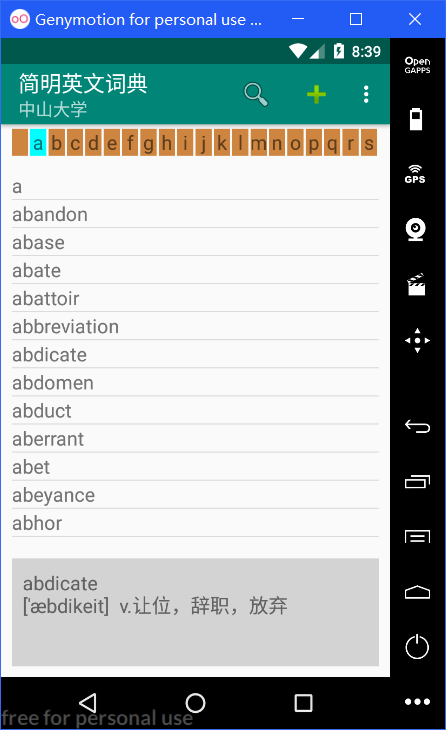
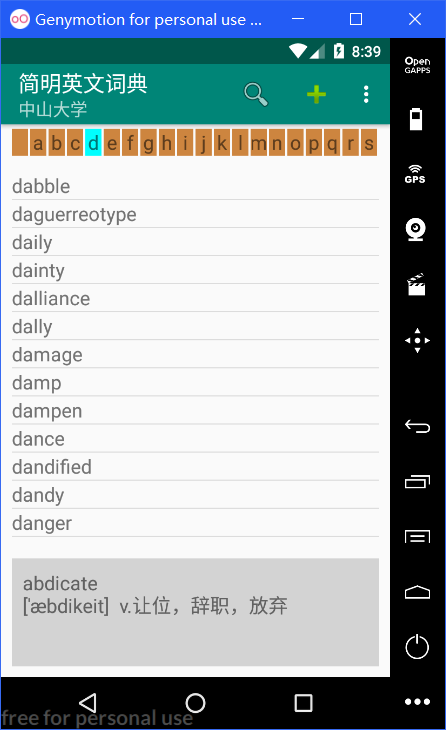
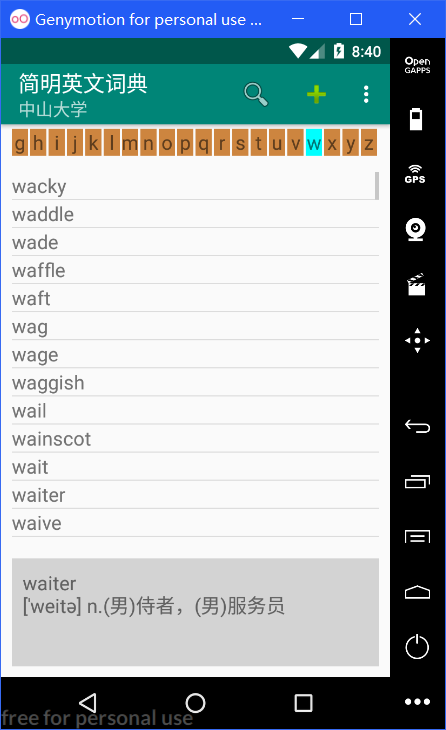
（4）增加信息显示栏，点击单词将显示含义，点击单词abate后显示其具体含义



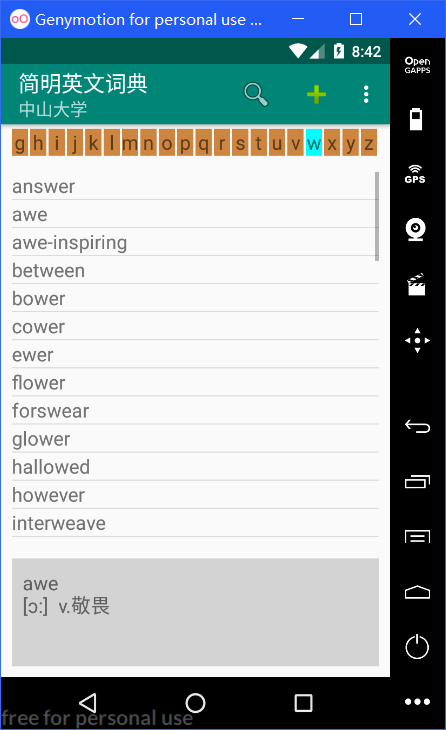
（5）长点击单词abdicate将显示弹出式菜单



（6）单词查询功能，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。

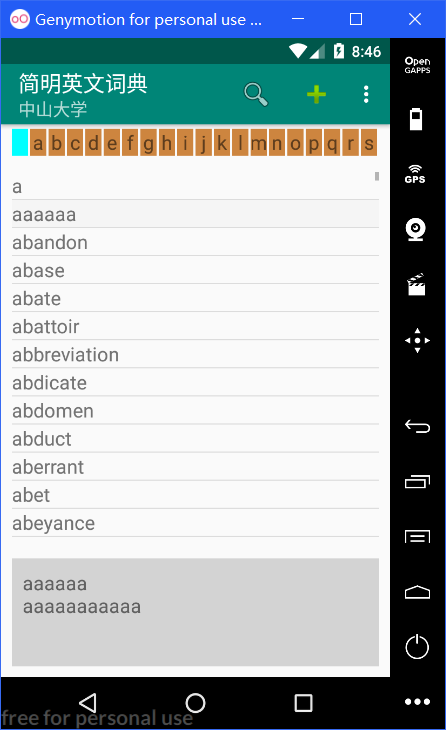
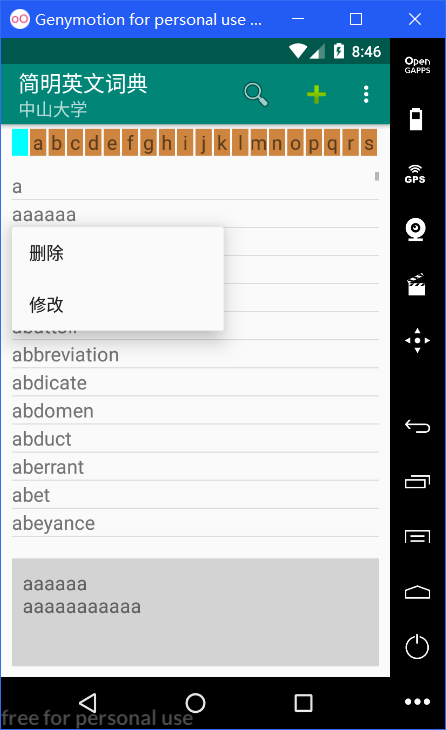
  

显示以a开头的单词 显示以d开头的单词 显示以w开头的单词

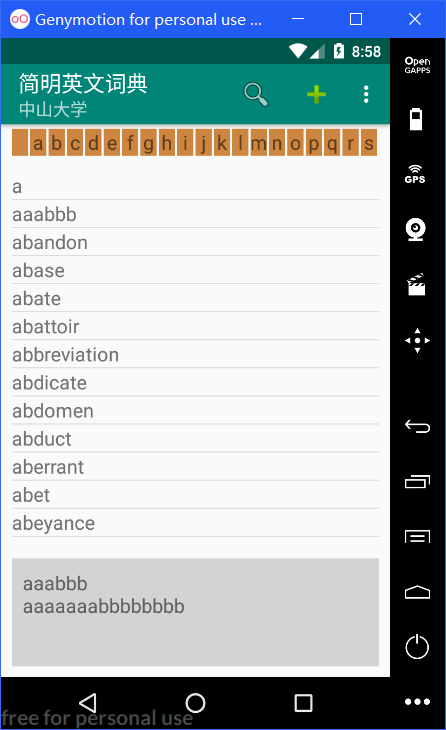
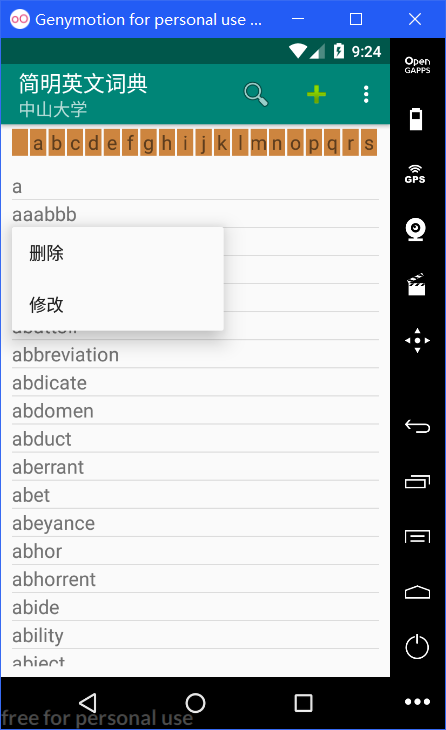
 

点击查询图标 查询包含输入子串的单词

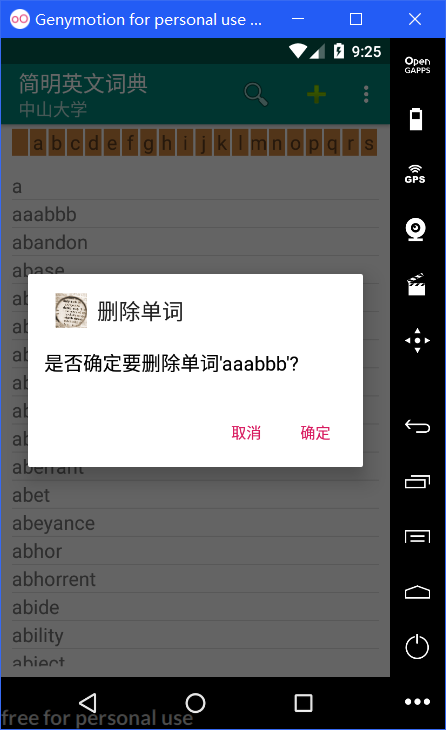
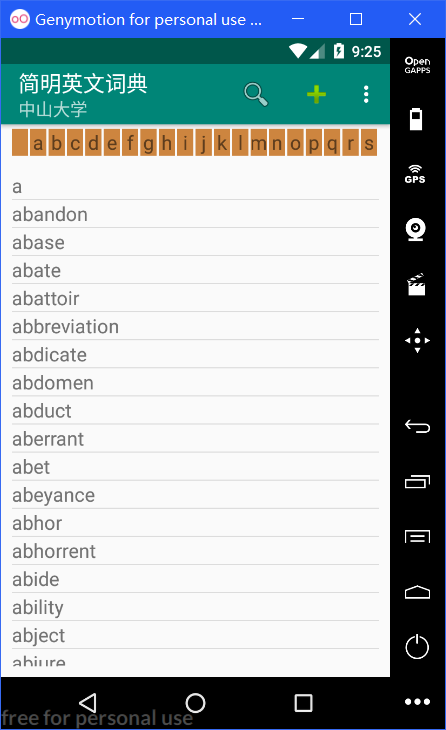
（7）通过图标和弹出式菜单用对话框方式实现增加、删除和修改单词

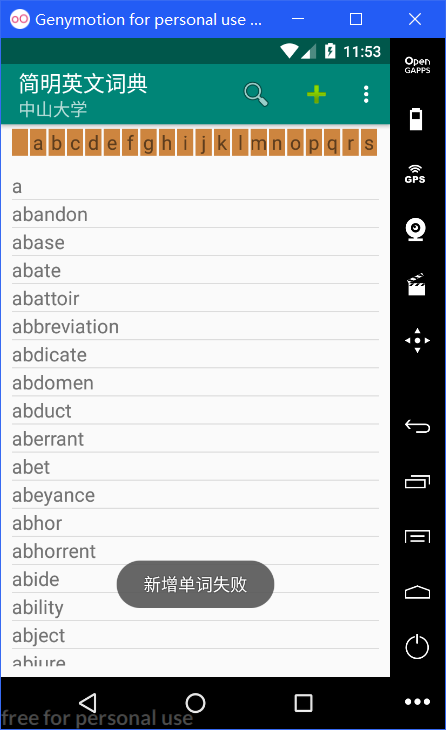
增加单词 增加后 长按单词aaaaaa

修改单词aaaaaa 修改后 长按单词aaabbb

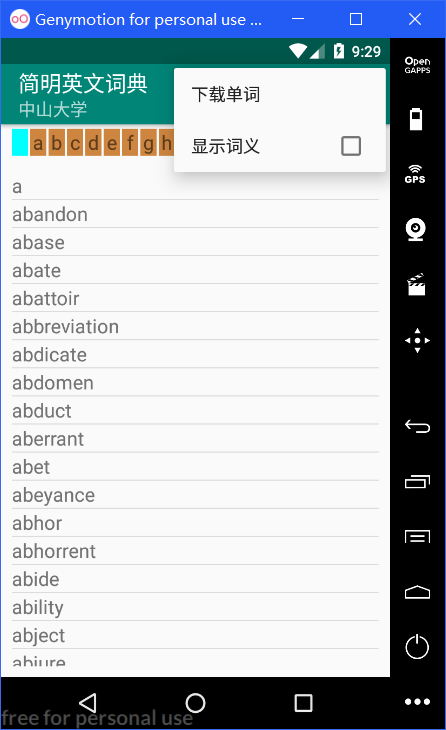
 

删除单词aaabbb 删除后

增加单词时选择不覆盖 显示失败信息

（8）通过菜单可以让词典直接显示语义

显示语义 不显示语义

**（9）在直接显示语义的情况下，还是可以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(新增功能)**

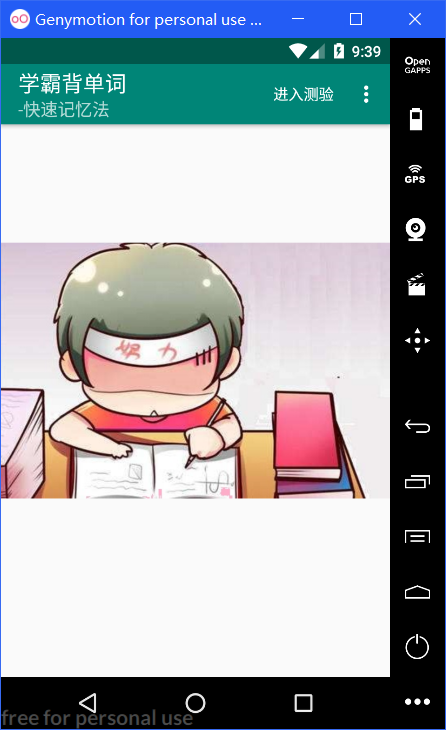
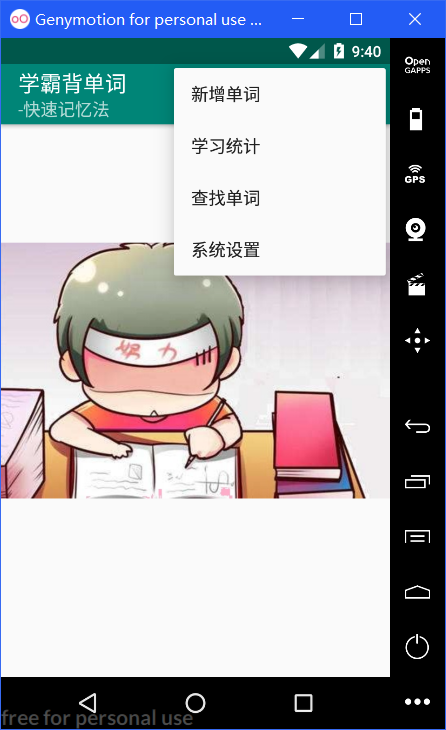
  

（10）下载和保存词典时增加一个进度显示

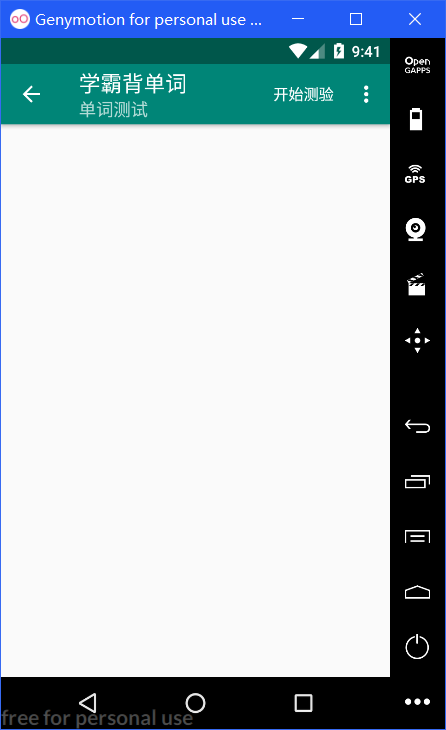
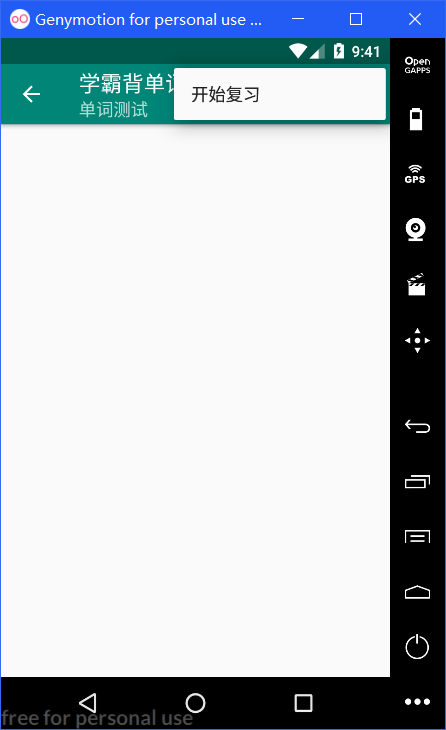


【背单词程序】

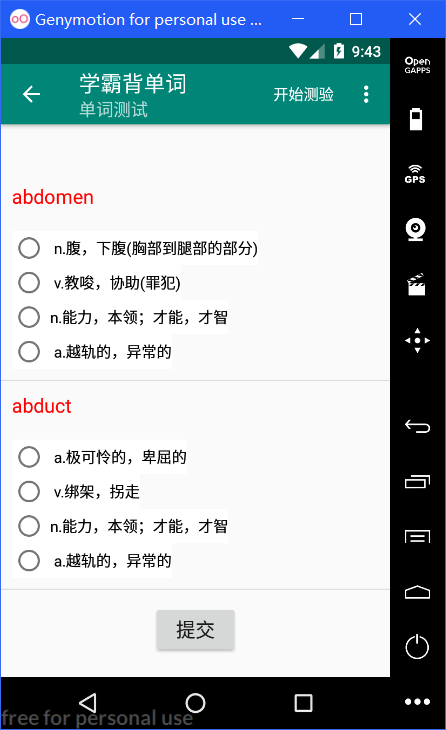
（1）做主界面

（2）点击“进入测验”，进入测验界面

（3）点击上面界面的开始测验，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个单词的解释中随机选取，并包含要测验的单词，顺序也要打乱



（4）点击菜单“新增单词”，通过对话框为词典程序增加单词

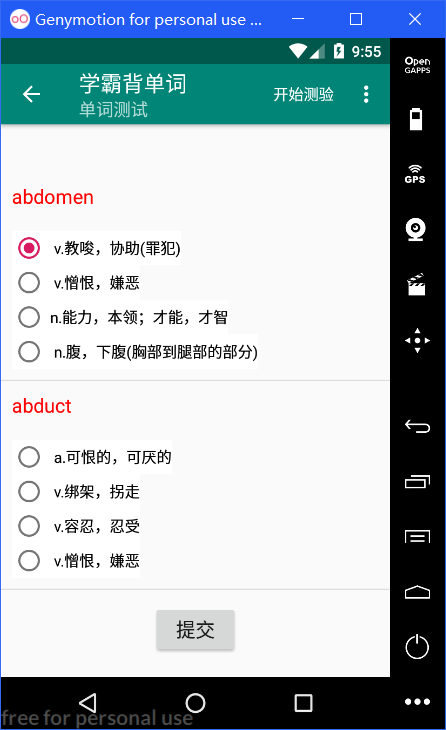
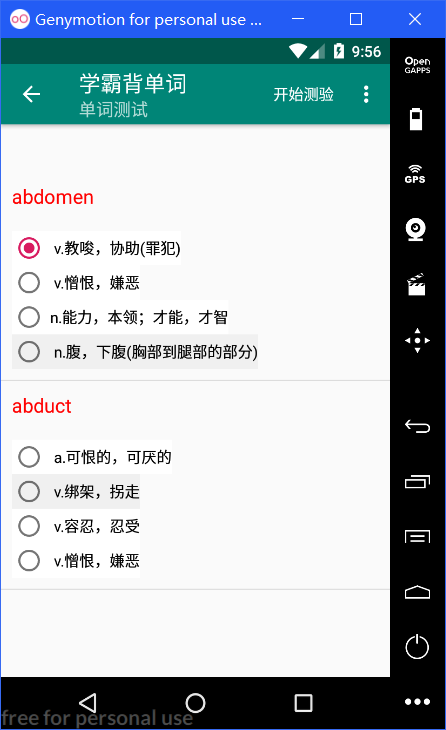
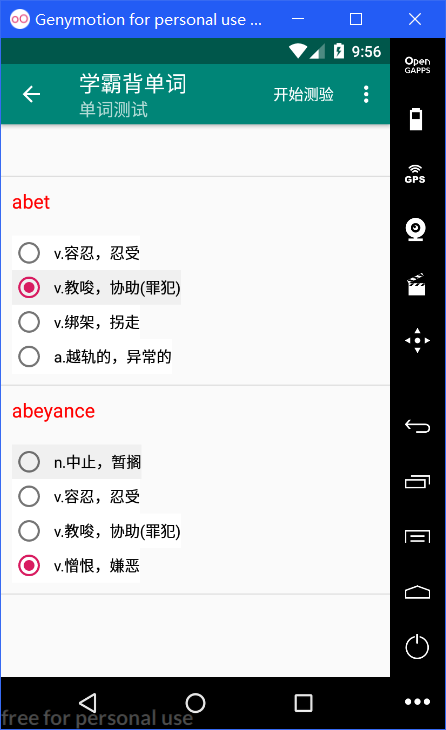
 

添加单词aabbccddee 在字典程序中添加成功

添加时选择不覆盖 添加失败

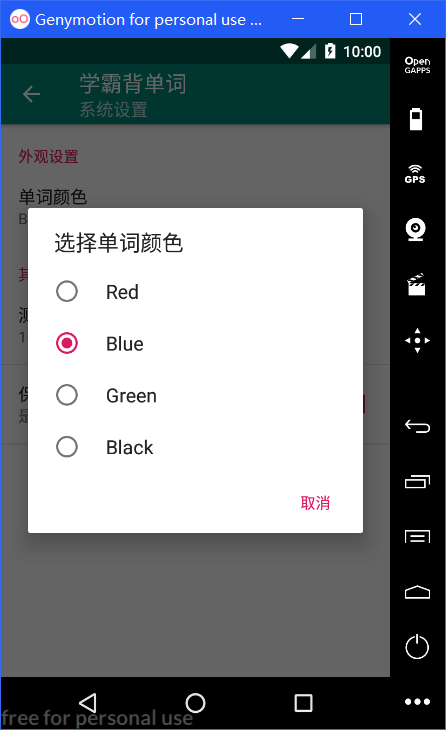
（5）提交后进行批改，看哪些题答对了，用背景色表示正确答案，提交后隐藏提交按钮

（6）显示学习统计数据



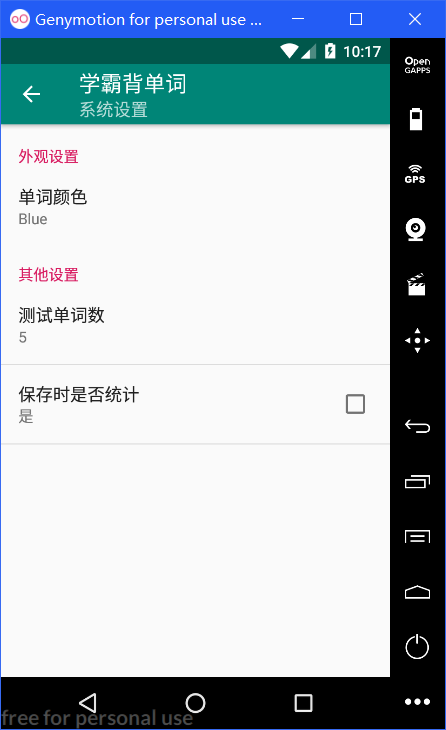
（7）增加“系统设置”功能，可以选择单词颜色，可以设置每次测试的单词数，以及保存时是否进行统计

设置单词颜色为Blue

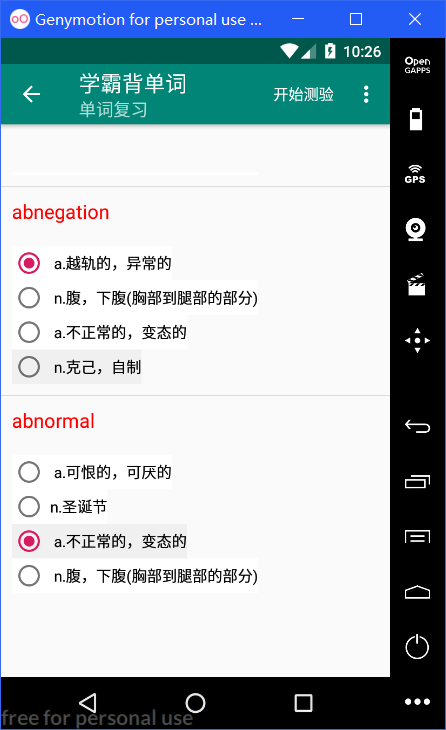
 

设置单词数为5

保存时不统计 测试的新单词 没有保存新单词记录

**（8）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（选做）**

复习已学过的单词 复习后的统计记录

**（9）背单词的查找单词，从词典中找出单词（选做）**

查找含we子串的单词 查找结果

**（10）背单词的统计加入首字母筛选查询（选做）**

查找前缀为abe的单词记录 查找结果

*完成后的主要源码(.java和.xml)：*

【词典程序】

**（1）AndroidManifest.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<manifest** xmlns:android="http://schemas.android.com/apk/res/android"
3. package="com.example.granddictionary"**>**
5. **<uses-permission** android:name="android.permission.INTERNET"**/>**
7. **<application**
8. android:allowBackup="true"
9. android:icon="@mipmap/ic\_launcher"
10. android:label="@string/app\_name"
11. android:roundIcon="@mipmap/ic\_launcher\_round"
12. android:supportsRtl="true"
13. android:theme="@style/AppTheme"**>**
14. **<activity**
15. android:name=".MainActivity"
16. android:label="@string/app\_name"
17. android:theme="@style/AppTheme.NoActionBar"**>**
18. **<intent-filter>**
19. **<action** android:name="android.intent.action.MAIN" **/>**
21. **<category** android:name="android.intent.category.LAUNCHER"**/>**
22. **</intent-filter>**
23. **</activity>**
24. **<provider**
25. android:authorities="com.example.providers.DictProvider"
26. android:name=".DictProvider"
27. android:exported="true"**/>**
28. **</application>**
30. **</manifest>**

**（2）MainActivity.java:**

1. **package** com.example.granddictionary;
3. **import** ...
5. **public** **class** MainActivity **extends** AppCompatActivity {
6. ListView listView;
7. Uri uri = Uri.parse("content://com.example.providers.DictProvider/");
8. ContentResolver resolver;
9. AlertDialog alertDialog;
10. AlertDialog change\_alertDialog;
11. ArrayList<Map<String, Object>> lists = **new** ArrayList<>();
12. SimpleAdapter adapter = **null**;
13. ProgressDialog progressDialog;
14. **int** last\_id = -999999, progressstatus;
15. **boolean** show\_explan = **false**;
17. Comparator<Map<String, Object>> comparator = **new** Comparator<Map<String, Object>>() {
18. @Override
19. **public** **int** compare(Map<String, Object> stringObjectMap, Map<String, Object> t1) {
20. **if** (stringObjectMap.get("word").toString().toLowerCase().compareTo(t1.get("word").toString().toLowerCase()) < 0)
21. **return** -1;
22. **else**
23. **return** 1;
24. }
25. };
26. Handler handler = **new** Handler() {
27. @Override
28. **public** **void** handleMessage(Message message) {
29. **if** (message.what == 0)
30. progressDialog.setProgress(progressstatus);
31. **else** **if** (message.what == 1) {
32. Collections.sort(lists, comparator);
33. adapter.notifyDataSetChanged();
34. }
35. }
36. };
38. @Override
39. **protected** **void** onCreate(Bundle savedInstanceState) {
40. **super**.onCreate(savedInstanceState);
41. setContentView(R.layout.activity\_main);
42. Toolbar toolbar = findViewById(R.id.toolbar);
43. setSupportActionBar(toolbar);
44. ActionBar actionBar = getSupportActionBar();
45. actionBar.setTitle("简明英文词典");
46. actionBar.setSubtitle("中山大学");
47. listView = (ListView) findViewById(R.id.listview);
48. resolver = getContentResolver();
49. display();
50. createchildlinearlayout();
51. }
53. @Override
54. **public** **boolean** onCreateOptionsMenu(Menu menu) {
55. getMenuInflater().inflate(R.menu.menu\_main, menu);
56. **return** **true**;
57. }
59. @Override
60. **public** **boolean** onOptionsItemSelected(MenuItem item) {
61. **switch** (item.getItemId()) {
62. **case** R.id.downloadwords:
63. getJson();
64. **break**;
65. **case** R.id.displaymeaning:
66. **if** (item.isChecked()) {
67. item.setChecked(**false**);
68. show\_explan = **false**;
69. setListView();
70. }
71. **else** {
72. TextView textView = (TextView) findViewById(R.id.word\_information);
73. textView.setVisibility(View.INVISIBLE);
74. ViewGroup.LayoutParams params = listView.getLayoutParams();
75. params.height = ViewGroup.LayoutParams.MATCH\_PARENT;
76. listView.setLayoutParams(params);
77. item.setChecked(**true**);
78. show\_explan = **true**;
79. setListView();
80. }
81. **break**;
82. **case** R.id.search\_button:
83. search\_word();
84. **break**;
85. **case** R.id.add\_button:
86. add\_word();
87. **break**;
88. }
89. **return** **super**.onOptionsItemSelected(item);
90. }
92. **public** **void** createchildlinearlayout() {
93. LinearLayout linearLayout = (LinearLayout) findViewById(R.id.linear);
94. **final** String[] strings = {"", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l",
95. "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"};
96. **for** (**int** i = 0; i < strings.length; ++i) {
97. **final** TextView tv = **new** TextView(**this**);
98. tv.setText(strings[i]);
99. tv.setTextSize(18);
100. tv.setPadding(2,0,2,0);
101. tv.setHeight(50);
102. tv.setWidth(30);
103. tv.setClickable(**true**);
104. tv.setId(i);
105. tv.setGravity(Gravity.CENTER);
106. LinearLayout.LayoutParams tvlp = **new** LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP\_CONTENT,
107. LinearLayout.LayoutParams.WRAP\_CONTENT);
108. tvlp.setMargins(0,0,4,0);
109. tv.setBackgroundColor(getResources().getColor(R.color.peru));
110. tv.setOnClickListener(**new** View.OnClickListener() {
111. @Override
112. **public** **void** onClick(View view) {
113. TextView textView;
114. **if** (last\_id != -999999) {
115. textView = (TextView) findViewById(last\_id);
116. textView.setBackgroundColor(getResources().getColor(R.color.peru));
117. }
118. last\_id = view.getId();
119. textView = (TextView) findViewById(view.getId());
120. textView.setBackgroundColor(getResources().getColor(R.color.cyan));
121. Cursor cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"},
122. "word like ?", **new** String[]{strings[view.getId()] + "%"},
123. "word");
124. change\_lists(cursor);
125. adapter.notifyDataSetChanged();
126. }
127. });
128. linearLayout.addView(tv, tvlp);
129. }
130. }
132. **public** **void** display() {
133. Cursor cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"},
134. **null**, **null**, "word");
135. change\_lists(cursor);
136. setListView();
137. }
139. **public** **void** change\_lists(Cursor cursor) {
140. lists.clear();
141. **while** (cursor.moveToNext()) {
142. Map<String, Object> map = **new** HashMap<String, Object>();
143. map.put("word", cursor.getString(cursor.getColumnIndex("word")));
144. map.put("explanation", cursor.getString(cursor.getColumnIndex("explanation")));
145. map.put("level", cursor.getString(cursor.getColumnIndex("level")));
146. lists.add(map);
147. }
148. Collections.sort(lists, comparator);
149. }

152. **public** **void** setListView() {
153. Collections.sort(lists, comparator);
154. **if** (show\_explan)
155. adapter = **new** SimpleAdapter(**this**, lists, R.layout.list\_meaning, **new** String[]{"word", "explanation"}, **new** **int**[]{R.id.word\_explanation, R.id.explanation});
156. **else**
157. adapter = **new** SimpleAdapter(**this**, lists, R.layout.list\_item, **new** String[]{"word"}, **new** **int**[]{R.id.word});
158. listView.setAdapter(adapter);
159. listView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {
160. @Override
161. **public** **void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {
162. **if** (show\_explan) **return**;
163. TextView textView = (TextView) findViewById(R.id.word\_information);
164. textView.setVisibility(View.VISIBLE);
165. ViewGroup.LayoutParams params = listView.getLayoutParams();
166. params.height = 680;
167. listView.setLayoutParams(params);
168. String str = lists.get(i).get("word").toString();
169. str += "\n" + lists.get(i).get("explanation").toString();
170. textView.setText(str);
171. }
172. });
173. listView.setOnItemLongClickListener(**new** AdapterView.OnItemLongClickListener() {
174. @Override
175. **public** **boolean** onItemLongClick(AdapterView<?> adapterView, **final** View view, **final** **int** i, **long** l) {
176. PopupMenu popupMenu = **new** PopupMenu(MainActivity.**this**, view);
177. popupMenu.getMenuInflater().inflate(R.menu.long\_click\_menu, popupMenu.getMenu());
178. popupMenu.setOnMenuItemClickListener(**new** PopupMenu.OnMenuItemClickListener() {
179. @Override
180. **public** **boolean** onMenuItemClick(MenuItem menuItem) {
181. **switch** (menuItem.getItemId()) {
182. **case** R.id.change:
183. change\_word(i);
184. **break**;
185. **case** R.id.delete:
186. delete\_word(i);
187. **break**;
188. }
189. **return** **true**;
190. }
191. });
192. popupMenu.show();
193. **return** **true**;
194. }
195. });
196. }
198. **public** **void** delete\_word(**final** **int** position) {
199. RelativeLayout word\_form = **new** RelativeLayout(**this**);
200. TextView textView = **new** TextView(**this**);
201. textView.setText("是否确定要删除单词'" + lists.get(position).get("word").toString() +"'?");
202. textView.setTextSize(18);
203. textView.setPadding(30, 40, 0, 0);
204. textView.setTextColor(Color.BLACK);
205. word\_form.addView(textView);
206. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
207. change\_alertDialog = builder.setIcon(R.drawable.dict)
208. .setTitle("删除单词")
209. .setView(word\_form)
210. .setNegativeButton("取消", **null**)
211. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
212. @Override
213. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
214. resolver.delete(uri, "word = ?", **new** String[]{lists.get(position).get("word").toString()});
215. lists.remove(position);
216. adapter.notifyDataSetChanged();
217. }
218. }).create();
219. change\_alertDialog.show();
220. }
222. **public** **void** change\_word(**final** **int** position) {
223. TableLayout word\_form = (TableLayout) getLayoutInflater().inflate(R.layout.change\_word, **null**);
224. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
225. change\_alertDialog = builder.setIcon(R.drawable.dict)
226. .setTitle("修改单词")
227. .setView(word\_form)
228. .setNegativeButton("取消", **null**)
229. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
230. @Override
231. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
232. EditText newword = (EditText) change\_alertDialog.findViewById(R.id.change\_word);
233. EditText newexplan = (EditText) change\_alertDialog.findViewById(R.id.change\_explanation);
234. EditText newlevel = (EditText) change\_alertDialog.findViewById(R.id.change\_level);
235. Cursor cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"}, "word = ?",
236. **new** String[]{lists.get(position).get("word").toString()}, **null**);
237. cursor.moveToFirst();
238. **int** index = cursor.getInt(cursor.getColumnIndex("\_id"));
239. ContentValues contentValues = **new** ContentValues();
240. contentValues.put("\_id", index);
241. contentValues.put("word", newword.getText().toString());
242. contentValues.put("explanation", newexplan.getText().toString());
243. contentValues.put("level", newlevel.getText().toString());
244. resolver.update(uri, contentValues, "\_id = ?", **new** String[]{Integer.toString(index)});
245. **for** (**int** j = 0; j < lists.size(); ++j) {
246. **if** (lists.get(j).get("word").toString().equals(cursor.getString(cursor.getColumnIndex("word")))) {
247. Map<String, Object> map = **new** HashMap<>();
248. map.put("word", newword.getText().toString());
249. map.put("explanation", newexplan.getText().toString());
250. map.put("level", newlevel.getText().toString());
251. map.put("\_id", index);
252. lists.remove(j);
253. lists.add(map);
254. **break**;
255. }
256. }
257. Collections.sort(lists, comparator);
258. adapter.notifyDataSetChanged();
259. }
260. }).create();
261. change\_alertDialog.show();
262. }
264. **public** **void** getJson() {
265. showProgress(100);
266. **new** Thread(**new** Runnable() {
267. @Override
268. **public** **void** run() {
269. **try** {
270. String address = "http://172.18.187.9:8080/dict/";
271. URL url = **new** URL(address);
272. URLConnection connection = url.openConnection();
273. InputStream inputStream = connection.getInputStream();
274. InputStreamReader inputStreamReader = **new** InputStreamReader(inputStream, "utf-8");
275. BufferedReader bufferedReader = **new** BufferedReader(inputStreamReader);
276. StringBuilder stringBuilder = **new** StringBuilder();
277. String line;
278. **while** ((line = bufferedReader.readLine()) != **null**)
279. stringBuilder.append(line);
280. String str = stringBuilder.toString();
281. JSONArray jsonArray = **new** JSONArray(str);
282. Cursor cursor = resolver.query(uri, **new** String[]{"word"}, **null**, **null**, **null**);
283. Set set = **new** HashSet();
284. **while** (cursor.moveToNext())
285. set.add(cursor.getString(cursor.getColumnIndex("word")));
286. lists.clear();
287. **for** (**int** i = 0; i < jsonArray.length(); ++i) {
288. WordRec wordRec = parserJSON(jsonArray.get(i).toString());
289. ContentValues values = **new** ContentValues();
290. values.put("\_id", i);
291. values.put("word", wordRec.getWord());
292. values.put("explanation", wordRec.getExplanation());
293. values.put("level", wordRec.getLevel());
294. Map<String, Object> map = **new** HashMap<>();
295. map.put("word", wordRec.getWord());
296. map.put("explanation", wordRec.getExplanation());
297. map.put("level", wordRec.getLevel());
298. map.put("\_id", i);
299. lists.add(map);
300. **if** (set.contains(wordRec.getWord()))
301. resolver.update(uri, values, "word = ?", **new** String[]{wordRec.getWord()});
302. **else**
303. resolver.insert(uri, values);
304. progressstatus = 100 \* (i + 1) / jsonArray.length();
305. handler.sendEmptyMessage(0);
306. }
307. progressDialog.dismiss();
308. handler.sendEmptyMessage(1);
309. } **catch** (Exception ex) {
310. ex.printStackTrace();
311. }
312. }
313. }).start();
314. }
316. **public** **void** showProgress(**int** max\_length) {
317. progressDialog = **new** ProgressDialog(MainActivity.**this**);
318. progressDialog.setMax(max\_length);
319. progressDialog.setTitle("下载单词");
320. progressDialog.setCancelable(**false**);
321. progressDialog.setProgressStyle(ProgressDialog.STYLE\_HORIZONTAL);
322. progressDialog.setIndeterminate(**false**);
323. progressDialog.show();
325. }
327. **public** **static** WordRec parserJSON(String str) {
328. JSONObject jsonObject = **null**;
329. **try** {
330. jsonObject = **new** JSONObject(str);
331. } **catch** (Exception ex) {
332. ex.printStackTrace();
333. }
334. WordRec wordRec = **new** WordRec();
335. wordRec.setWord(jsonObject.optString("word"));
336. wordRec.setExplanation(jsonObject.optString("explanation"));
337. wordRec.setLevel(Integer.parseInt(jsonObject.optString("level")));
338. **return** wordRec;
339. }
341. **public** **void** add\_word() {
342. TableLayout word\_form = (TableLayout) getLayoutInflater().inflate(R.layout.new\_word, **null**);
343. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
344. alertDialog = builder.setIcon(R.drawable.dict)
345. .setTitle("增加单词")
346. .setView(word\_form)
347. .setNegativeButton("取消", **null**)
348. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
349. @Override
350. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
351. EditText newword = (EditText) alertDialog.findViewById(R.id.new\_word);
352. EditText newexplan = (EditText) alertDialog.findViewById(R.id.new\_explanation);
353. EditText newlevel = (EditText) alertDialog.findViewById(R.id.new\_level);
354. CheckedTextView checkedTextView = (CheckedTextView) alertDialog.findViewById(R.id.check);
355. Cursor cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"}, "word = ?",
356. **new** String[]{newword.getText().toString()}, **null**);
357. **if** (cursor.getCount() == 0) {
358. Log.v("out:", "insert");
359. cursor = resolver.query(uri, **new** String[]{"count(\*)"}, **null**,
360. **null**, **null**);
361. cursor.moveToFirst();
362. **int** index = cursor.getInt(0);
363. ContentValues contentValues = **new** ContentValues();
364. contentValues.put("\_id", index + 1);
365. contentValues.put("word", newword.getText().toString());
366. contentValues.put("explanation", newexplan.getText().toString());
367. contentValues.put("level", newlevel.getText().toString());
368. resolver.insert(uri, contentValues);
369. Map<String, Object> map = **new** HashMap<>();
370. map.put("word", newword.getText().toString());
371. map.put("explanation", newexplan.getText().toString());
372. map.put("level", newlevel.getText().toString());
373. lists.add(map);
374. Collections.sort(lists, comparator);
375. adapter.notifyDataSetChanged();
376. }
377. **else** **if** (checkedTextView.isChecked()) {
378. cursor.moveToFirst();
379. **int** index = cursor.getInt(cursor.getColumnIndex("\_id"));
380. ContentValues contentValues = **new** ContentValues();
381. contentValues.put("\_id", index);
382. contentValues.put("word", newword.getText().toString());
383. contentValues.put("explanation", newexplan.getText().toString());
384. contentValues.put("level", newlevel.getText().toString());
385. resolver.update(uri, contentValues, "word = ?", **new** String[]{newword.getText().toString()});
386. **for** (**int** j = 0; j < lists.size(); ++j) {
387. **if** (lists.get(j).get("word").toString().equals(cursor.getString(cursor.getColumnIndex("word")))) {
388. Map<String, Object> map = **new** HashMap<>();
389. map.put("word", newword.getText().toString());
390. map.put("explanation", newexplan.getText().toString());
391. map.put("level", newlevel.getText().toString());
392. map.put("\_id", index);
393. lists.remove(j);
394. lists.add(map);
395. **break**;
396. }
397. }
398. Collections.sort(lists, comparator);
399. adapter.notifyDataSetChanged();
400. }
401. **else**
402. Toast.makeText(getApplicationContext(), "新增单词失败", Toast.LENGTH\_SHORT).show();
403. }
404. }).create();
405. alertDialog.show();
406. }
408. **public** **void** search\_word() {
409. TableLayout wordform = (TableLayout) getLayoutInflater().inflate(R.layout.search\_word, **null**);
410. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
411. alertDialog = builder.setIcon(R.drawable.dict)
412. .setTitle("查询单词")
413. .setView(wordform)
414. .setNegativeButton("取消", **null**)
415. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
416. @Override
417. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
418. EditText searchword = (EditText) alertDialog.findViewById(R.id.search\_word);
419. Cursor cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"},
420. "word like ?", **new** String[]{"%" + searchword.getText().toString() + "%"},
421. "word");
422. change\_lists(cursor);
423. adapter.notifyDataSetChanged();
424. }
425. }).create();
426. alertDialog.show();
427. }
429. **public** **void** toggle(View view) {
430. CheckedTextView checkedTextView = (CheckedTextView) view;
431. checkedTextView.toggle();
432. }
433. }

**（3）WordRec.java:**

1. **package** com.example.granddictionary;
3. **public** **class** WordRec {
4. **private** **int** id;
5. **private** String word;
6. **private** String explanation;
7. **private** **int** level;
8. WordRec() {}
9. WordRec(**int** id, String word, String explanation, **int** level) {
10. **this**.id = id;
11. **this**.word = word;
12. **this**.explanation = explanation;
13. **this**.level = level;
14. }
15. **public** **int** getId() { **return** id; }
16. **public** **void** setId(**int** id) { **this**.id = id; }
17. **public** String getWord() { **return** word; }
18. **public** **void** setWord(String word) { **this**.word = word; }
19. **public** String getExplanation() { **return** explanation; }
20. **public** **void** setExplanation(String explanation) { **this**.explanation = explanation; }
21. **public** **int** getLevel() { **return** level; }
22. **public** **void** setLevel(**int** level) { **this**.level = level; }
23. }

**（4）DBOpenHandler.java:**

1. **package** com.example.granddictionary;
3. **import** android.content.Context;
4. **import** android.database.sqlite.SQLiteDatabase;
5. **import** android.database.sqlite.SQLiteOpenHelper;
7. **public** **class** DBOpenHandler **extends** SQLiteOpenHelper {
8. **int** version;
9. **public** DBOpenHandler(Context context, String name, SQLiteDatabase.CursorFactory factory, **int** version) {
10. **super**(context, name, factory, version);
11. **this**.version = version;
12. }
13. @Override
14. **public** **void** onCreate(SQLiteDatabase db) {
15. db.execSQL("CREATE TABLE dict(\_id integer primary key autoincrement, word varchar(64) unique, explanation text, level int default 0, modified\_time timestamp)");
16. }
17. @Override
18. **public** **void** onUpgrade(SQLiteDatabase db, **int** oldversion, **int** newversion) {
19. db.execSQL("");
20. }
21. }

**（5）DictProvider.java:**

1. **package** com.example.granddictionary;
3. **import** ...
5. **public** **class** DictProvider **extends** ContentProvider {
6. **private** DBOpenHandler dbOpenHandler;
7. @Override
8. **public** **boolean** onCreate() {
9. **this**.dbOpenHandler = **new** DBOpenHandler(getContext(), "DICT", **null**, 1);
10. **return** **true**;
11. }
12. @Override
13. **public** String getType(Uri uri) {
14. **return** **null**;
15. }
16. @Override
17. **public** Uri insert(Uri uri, ContentValues contentValues) {
18. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
19. db.insert("dict", **null**, contentValues);
20. db.close();
21. **return** uri;
22. }
23. @Override
24. **public** **int** delete(Uri uri, String where, String[] whereargs) {
25. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
26. **int** ret = db.delete("dict", where, whereargs);
27. db.close();
28. **return** ret;
29. }
30. @Override
31. **public** **int** update(Uri uri, ContentValues contentValues, String where, String[] whereArgs) {
32. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
33. **int** num = db.update("dict", contentValues, where, whereArgs);
34. db.close();
35. getContext().getContentResolver().notifyChange(uri, **null**);
36. **return** num;
37. }
38. @Override
39. **public** Cursor query(Uri uri, String[] projection, String where, String[] whereArgs, String sortorder) {
40. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
41. Cursor cursor = db.query("dict", projection, where, whereArgs, **null**,
42. **null**, sortorder, **null**);
43. **return** cursor;
44. }
46. }

**（6）activity\_main.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout**
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. xmlns:app="http://schemas.android.com/apk/res-auto"
5. xmlns:tools="http://schemas.android.com/tools"
6. android:layout\_width="match\_parent"
7. android:layout\_height="match\_parent"
8. tools:context=".MainActivity"**>**
10. **<android.support.design.widget.AppBarLayout**
11. android:layout\_height="wrap\_content"
12. android:layout\_width="match\_parent"
13. android:theme="@style/AppTheme.AppBarOverlay"**>**
15. **<android.support.v7.widget.Toolbar**
16. android:id="@+id/toolbar"
17. android:layout\_width="match\_parent"
18. android:layout\_height="?attr/actionBarSize"
19. android:background="?attr/colorPrimary"
20. app:popupTheme="@style/AppTheme.PopupOverlay" **/>**
21. **</android.support.design.widget.AppBarLayout>**
23. **<HorizontalScrollView**
24. android:layout\_width="match\_parent"
25. android:layout\_height="wrap\_content"
26. android:id="@+id/scrollView"
27. android:scrollbarAlwaysDrawHorizontalTrack="false"
28. android:layout\_marginTop="60dp"
29. android:layout\_marginRight="10dp"
30. android:layout\_marginLeft="10dp"**>**
31. **<LinearLayout**
32. android:layout\_width="wrap\_content"
33. android:layout\_height="wrap\_content"
34. android:orientation="horizontal"
35. android:id="@+id/linear"**/>**
36. **</HorizontalScrollView>**
38. **<ListView**
39. android:layout\_width="match\_parent"
40. android:layout\_height="match\_parent"
41. android:layout\_marginTop="100dp"
42. android:layout\_marginBottom="10dp"
43. android:layout\_marginLeft="10dp"
44. android:layout\_marginRight="10dp"
45. android:id="@+id/listview"**/>**
47. **<TextView**
48. android:id="@+id/word\_information"
49. android:layout\_width="match\_parent"
50. android:layout\_height="100dp"
51. android:layout\_alignParentBottom="true"
52. android:layout\_marginTop="10dp"
53. android:visibility="invisible"
54. android:text="单词详情"
55. android:background="@color/grey"
56. android:textSize="18sp"
57. android:layout\_margin="10dp"
58. android:padding="10dp"**/>**
60. **</RelativeLayout>**

**（7）change\_word.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<TableLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:tools="http://schemas.android.com/tools"
4. android:layout\_width="match\_parent"
5. android:layout\_height="match\_parent"
6. android:id="@+id/changeform"
7. android:padding="12dp"**>**
8. **<TableRow** android:paddingBottom="10dp"
9. android:layout\_width="wrap\_content"
10. android:layout\_height="wrap\_content"**>**
11. **<TextView**
12. android:layout\_width="0px"
13. android:layout\_weight="0.2"
14. android:layout\_height="wrap\_content"
15. android:text="单词"
16. android:textSize="18sp" **/>**
17. **<EditText** android:id="@+id/change\_word"
18. android:layout\_height="wrap\_content"
19. android:layout\_width="0dp"
20. android:layout\_weight="0.8"
21. android:hint="请填写单词"
22. android:padding="10dp"
23. android:background="@drawable/edit\_border"
24. android:selectAllOnFocus="true"**/>**
25. **</TableRow>**
26. **<TableRow** android:paddingBottom="10dp"**>**
28. **<TextView**
29. android:layout\_width="0dp"
30. android:layout\_weight="0.2"
31. android:layout\_height="wrap\_content"
32. android:text="解释"
33. android:textSize="18sp" **/>**
34. **<EditText**
35. android:layout\_height="90dp"
36. android:layout\_width="0dp"
37. android:layout\_weight="0.8"
38. android:minLines="5"
39. android:gravity="top"
40. android:hint="请填写解释"
41. android:padding="10dp"
42. android:background="@drawable/edit\_border"
43. android:id="@+id/change\_explanation"**/>**
44. **</TableRow>**
45. **<TableRow** android:paddingBottom="10dp"**>**
46. **<TextView** android:text="级别"
47. android:textSize="18sp"
48. android:layout\_width="0dp"
49. android:layout\_weight="0.2"
50. android:layout\_height="wrap\_content"**/>**
51. **<EditText** android:id="@+id/change\_level"
52. android:layout\_height="wrap\_content"
53. android:layout\_width="0dp"
54. android:layout\_weight="0.8"
55. android:padding="10dp"
56. android:background="@drawable/edit\_border"
57. android:hint="请填写难度(0~8)"**/>**
58. **</TableRow>**
59. **</TableLayout>**

**（8）list\_item.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"**>**
5. **<TextView**
6. android:layout\_width="wrap\_content"
7. android:layout\_height="25dp"
8. android:id="@+id/word"
9. android:text="单词"
10. android:textSize="18sp"**/>**
11. **</RelativeLayout>**

**（9）new\_word.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<TableLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:tools="http://schemas.android.com/tools"
4. android:layout\_width="match\_parent"
5. android:layout\_height="match\_parent"
6. android:id="@+id/wordform"
7. android:padding="12dp"**>**
8. **<TableRow** android:paddingBottom="10dp"
9. android:layout\_width="wrap\_content"
10. android:layout\_height="wrap\_content"**>**
11. **<TextView**
12. android:layout\_width="0px"
13. android:layout\_weight="0.2"
14. android:layout\_height="wrap\_content"
15. android:text="单词"
16. android:textSize="18sp" **/>**
17. **<EditText** android:id="@+id/new\_word"
18. android:layout\_height="wrap\_content"
19. android:layout\_width="0dp"
20. android:layout\_weight="0.8"
21. android:hint="请填写单词"
22. android:padding="10dp"
23. android:background="@drawable/edit\_border"
24. android:selectAllOnFocus="true"**/>**
25. **</TableRow>**
26. **<TableRow** android:paddingBottom="10dp"**>**
28. **<TextView**
29. android:layout\_width="0dp"
30. android:layout\_weight="0.2"
31. android:layout\_height="wrap\_content"
32. android:text="解释"
33. android:textSize="18sp" **/>**
34. **<EditText**
35. android:layout\_height="90dp"
36. android:layout\_width="0dp"
37. android:layout\_weight="0.8"
38. android:minLines="5"
39. android:gravity="top"
40. android:hint="请填写解释"
41. android:padding="10dp"
42. android:background="@drawable/edit\_border"
43. android:id="@+id/new\_explanation"**/>**
44. **</TableRow>**
45. **<TableRow** android:paddingBottom="10dp"**>**
46. **<TextView** android:text="级别"
47. android:textSize="18sp"
48. android:layout\_width="0dp"
49. android:layout\_weight="0.2"
50. android:layout\_height="wrap\_content"**/>**
51. **<EditText** android:id="@+id/new\_level"
52. android:layout\_height="wrap\_content"
53. android:layout\_width="0dp"
54. android:layout\_weight="0.8"
55. android:padding="10dp"
56. android:background="@drawable/edit\_border"
57. android:hint="请填写难度(0~8)"**/>**
58. **</TableRow>**
59. **<TableRow>**
60. **<CheckedTextView** android:id="@+id/check"
61. android:layout\_width="wrap\_content"
62. android:layout\_height="match\_parent"
63. android:checkMark="?android:attr/listChoiceIndicatorMultiple"
64. android:checked="true"
65. android:clickable="true"
66. android:gravity="center"
67. android:text="是否覆盖"
68. android:textSize="18sp"
69. android:onClick="toggle"**/>**
70. **</TableRow>**
71. **</TableLayout>**

**（10）list\_meaning.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"**>**
5. **<LinearLayout**
6. android:layout\_width="match\_parent"
7. android:layout\_height="wrap\_content"
8. android:orientation="vertical"**>**
9. **<TextView**
10. android:layout\_width="wrap\_content"
11. android:layout\_height="wrap\_content"
12. android:id="@+id/word\_explanation"
13. android:gravity="center\_vertical"
14. android:text="单词"
15. android:textSize="18sp"**/>**
16. **<TextView**
17. android:layout\_width="wrap\_content"
18. android:layout\_height="wrap\_content"
19. android:id="@+id/explanation"
20. android:gravity="center\_vertical"
21. android:text="意思"
22. android:textSize="18sp"**/>**
23. **</LinearLayout>**
24. **</RelativeLayout>**

**（11）search\_word.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<TableLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"
5. android:id="@+id/wordform"
6. android:padding="12dp"**>**
7. **<TableRow** android:layout\_width="wrap\_content"
8. android:layout\_height="wrap\_content"**>**
9. **<TextView** android:text="单词"
10. android:layout\_width="0px"
11. android:layout\_weight="0.15"
12. android:layout\_height="wrap\_content"
13. android:textSize="18sp"**/>**
14. **<EditText** android:id="@+id/search\_word"
15. android:layout\_width="0px"
16. android:layout\_weight="0.85"
17. android:layout\_height="wrap\_content"
18. android:hint="请填写单词"
19. android:padding="10dp"
20. android:background="@drawable/edit\_border"
21. android:selectAllOnFocus="true"**/>**
22. **</TableRow>**
23. **</TableLayout>**

**（12）long\_click\_menu.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"**>**
3. **<item** android:id="@+id/delete"
4. android:title="删除"**/>**
5. **<item** android:id="@+id/change"
6. android:title="修改"**/>**
7. **</menu>**

**（13）menu\_main.xml:**

1. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"
2. xmlns:app="http://schemas.android.com/apk/res-auto"
3. xmlns:tools="http://schemas.android.com/tools"
4. tools:context="com.example.granddictionary.MainActivity" **>**
5. **<item** android:id="@+id/search\_button"
6. android:icon="@android:drawable/ic\_menu\_search"
7. android:title="search"
8. app:showAsAction="ifRoom"**/>**
9. **<item** android:id="@+id/add\_button"
10. android:icon="@android:drawable/ic\_input\_add"
11. android:title="add"
12. app:showAsAction="ifRoom"**/>**
13. **<item** android:id="@+id/downloadwords"
14. android:title="下载单词"
15. android:orderInCategory="0"
16. app:showAsAction="never" **/>**
17. **<item** android:id="@+id/displaymeaning"
18. android:title="显示词义"
19. android:orderInCategory="1"
20. android:checkable="true"
21. android:checked="false"**/>**
22. **</menu>**

【背单词程序】

**（1）AndroidManifest.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<manifest** xmlns:android="http://schemas.android.com/apk/res/android"
3. package="com.example.grandwordremember"**>**
5. **<application**
6. android:allowBackup="true"
7. android:icon="@mipmap/ic\_launcher"
8. android:label="@string/app\_name"
9. android:roundIcon="@mipmap/ic\_launcher\_round"
10. android:supportsRtl="true"
11. android:theme="@style/AppTheme"**>**
12. **<activity**
13. android:name=".MainActivity"
14. android:label="@string/app\_name"
15. android:theme="@style/AppTheme.NoActionBar"**>**
16. **<intent-filter>**
17. **<action** android:name="android.intent.action.MAIN" **/>**
18. **<category** android:name="android.intent.category.LAUNCHER" **/>**
19. **</intent-filter>**
20. **</activity>**
21. **<activity** android:name=".TestActivity"**>**
22. **<intent-filter>**
23. **<action** android:name="Test"**/>**
24. **<category** android:name="android.intent.category.DEFAULT"**/>**
25. **</intent-filter>**
26. **</activity>**
27. **<activity** android:name=".StatisticsAcitivity"**>**
28. **<intent-filter>**
29. **<action** android:name="Statistics"**/>**
30. **<category** android:name="android.intent.category.DEFAULT"**/>**
31. **</intent-filter>**
32. **</activity>**
33. **<activity** android:name=".SystemSettingAcitivity"**>**
34. **<intent-filter>**
35. **<action** android:name="SystemSetting"**/>**
36. **<category** android:name="android.intent.category.DEFAULT"**/>**
37. **</intent-filter>**
38. **</activity>**
39. **<provider**
40. android:authorities="com.example.grandwordremember.TestDB"
41. android:name=".TestDB"**/>**
42. **</application>**
44. **</manifest>**

**（2）MainActivity.java:**

1. **package** com.example.grandwordremember;
3. **import** ...
5. **public** **class** MainActivity **extends** AppCompatActivity {
6. AlertDialog alertDialog;
7. Uri uri = Uri.parse("content://com.example.providers.DictProvider/");
8. Uri test\_uri = Uri.parse("content://com.example.grandwordremember.TestDB/");
9. ContentResolver resolver;
10. ListView listView;
11. SimpleAdapter adapter;
12. ArrayList<Map<String, Object>> lists = **new** ArrayList<>();
14. Comparator<Map<String, Object>> comparator = **new** Comparator<Map<String, Object>>() {
15. @Override
16. **public** **int** compare(Map<String, Object> stringObjectMap, Map<String, Object> t1) {
17. **if** (stringObjectMap.get("word").toString().toLowerCase().compareTo(t1.get("word").toString().toLowerCase()) < 0)
18. **return** -1;
19. **else**
20. **return** 1;
21. }
22. };
24. @Override
25. **protected** **void** onCreate(Bundle savedInstanceState) {
26. **super**.onCreate(savedInstanceState);
27. setContentView(R.layout.activity\_main);
28. Toolbar toolbar = findViewById(R.id.toolbar);
29. setSupportActionBar(toolbar);
30. ActionBar actionBar = getSupportActionBar();
31. actionBar.setTitle("学霸背单词");
32. actionBar.setSubtitle("-快速记忆法");
33. resolver = getContentResolver();
34. listView = (ListView) findViewById(R.id.find\_list\_view);
35. listView.setVisibility(View.INVISIBLE);
36. }
38. @Override
39. **public** **boolean** onCreateOptionsMenu(Menu menu) {
40. getMenuInflater().inflate(R.menu.menu\_main, menu);
41. **return** **true**;
42. }
44. @Override
45. **public** **boolean** onOptionsItemSelected(MenuItem item) {
46. Intent intent;
47. **switch** (item.getItemId()) {
48. **case** R.id.test:
49. intent = **new** Intent(**this**, TestActivity.**class**);
50. startActivity(intent);
51. **break**;
52. **case** R.id.add\_word:
53. add\_new\_word();
54. **break**;
55. **case** R.id.learning\_statistics:
56. intent = **new** Intent(**this**, StatisticsAcitivity.**class**);
57. startActivity(intent);
58. **break**;
59. **case** R.id.system\_settings:
60. intent = **new** Intent(**this**, SystemSettingAcitivity.**class**);
61. startActivity(intent);
62. **break**;
63. **case** R.id.find\_word:
64. find\_word();
65. **break**;
66. }
67. **return** **true**;
68. }
70. **public** **void** add\_new\_word() {
71. TableLayout word\_form = (TableLayout) getLayoutInflater().inflate(R.layout.new\_word, **null**);
72. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
73. alertDialog = builder.setIcon(R.drawable.dict)
74. .setTitle("增加单词")
75. .setView(word\_form)
76. .setNegativeButton("取消", **null**)
77. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
78. @Override
79. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
80. EditText newword = (EditText) alertDialog.findViewById(R.id.new\_word);
81. EditText newexplan = (EditText) alertDialog.findViewById(R.id.new\_explanation);
82. EditText newlevel = (EditText) alertDialog.findViewById(R.id.new\_level);
83. CheckedTextView checkedTextView = (CheckedTextView) alertDialog.findViewById(R.id.check);
84. Cursor cursor = resolver.query(uri, **new** String[]{"\_id"}, "word = ?", **new** String[]{newword.getText().toString()}, **null**);
85. **if** (cursor.getCount() == 0) {
86. Log.v("out:", "insert");
87. cursor = resolver.query(uri, **new** String[]{"count(\*)"}, **null**, **null**, **null**);
88. cursor.moveToFirst();
89. **int** index = cursor.getInt(0);
90. ContentValues contentValues = **new** ContentValues();
91. contentValues.put("\_id", index + 1);
92. contentValues.put("word", newword.getText().toString());
93. contentValues.put("explanation", newexplan.getText().toString());
94. contentValues.put("level", newlevel.getText().toString());
95. resolver.insert(uri, contentValues);
96. }
97. **else** **if** (checkedTextView.isChecked()) {
98. cursor.moveToFirst();
99. **int** index = cursor.getInt(cursor.getColumnIndex("\_id"));
100. ContentValues contentValues = **new** ContentValues();
101. contentValues.put("\_id", index);
102. contentValues.put("word", newword.getText().toString());
103. contentValues.put("explanation", newexplan.getText().toString());
104. contentValues.put("level", newlevel.getText().toString());
105. resolver.update(uri, contentValues, "word = ?", **new** String[]{newword.getText().toString()});
106. }
107. **else**
108. Toast.makeText(getApplicationContext(), "新增单词失败", Toast.LENGTH\_SHORT).show();
109. }
110. }).create();
111. alertDialog.show();
112. }
114. **public** **void** find\_word() {
115. TableLayout wordform = (TableLayout) getLayoutInflater().inflate(R.layout.search\_word, **null**);
116. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
117. alertDialog = builder.setIcon(R.drawable.dict)
118. .setTitle("查询单词")
119. .setView(wordform)
120. .setNegativeButton("取消", **null**)
121. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
122. @Override
123. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
124. EditText searchword = (EditText) alertDialog.findViewById(R.id.search\_word);
125. Cursor cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"},
126. "word like ?", **new** String[]{"%" + searchword.getText().toString() + "%"},
127. "word");
128. display(cursor);
129. }
130. }).create();
131. alertDialog.show();
132. }
134. **public** **void** display(Cursor cursor) {
135. ImageView imageView = (ImageView) findViewById(R.id.image);
136. imageView.setVisibility(View.INVISIBLE);
137. listView.setVisibility(View.VISIBLE);
138. **while** (cursor.moveToNext()) {
139. Map<String, Object> map = **new** HashMap<String, Object>();
140. map.put("word", cursor.getString(cursor.getColumnIndex("word")));
141. map.put("explanation", cursor.getString(cursor.getColumnIndex("explanation")));
142. map.put("level", cursor.getString(cursor.getColumnIndex("level")));
143. lists.add(map);
144. }
145. Collections.sort(lists, comparator);
146. adapter = **new** SimpleAdapter(**this**, lists, R.layout.list\_meaning, **new** String[]{"word", "explanation"}, **new** **int**[]{R.id.word\_explanation, R.id.explanation});
147. listView.setAdapter(adapter);
148. }
150. **public** **void** toggle(View view) {
151. CheckedTextView checkedTextView = (CheckedTextView) view;
152. checkedTextView.toggle();
153. }
154. }

**（3）DBOpenHandler.java:**

1. **package** com.example.grandwordremember;
3. **import** android.content.Context;
4. **import** android.database.sqlite.SQLiteDatabase;
5. **import** android.database.sqlite.SQLiteOpenHelper;
7. **public** **class** DBOpenHandler **extends** SQLiteOpenHelper {
8. **int** version;
9. **public** DBOpenHandler(Context context, String name, SQLiteDatabase.CursorFactory factory, **int** version) {
10. **super**(context, name, factory, version);
11. **this**.version = version;
12. }
13. @Override
14. **public** **void** onCreate(SQLiteDatabase db) {
15. db.execSQL("CREATE TABLE words(word varchar(64) primary key, level int default 0, test\_count int default 0, correct\_count int default 0,last\_test\_time timestamp)");
16. }
17. @Override
18. **public** **void** onUpgrade(SQLiteDatabase db, **int** oldversion, **int** newversion) { db.execSQL(""); }
19. }

**（4）SettingFragment.java:**

1. **package** com.example.grandwordremember;
3. **import** android.content.SharedPreferences;
4. **import** android.os.Bundle;
5. **import** android.preference.CheckBoxPreference;
6. **import** android.preference.EditTextPreference;
7. **import** android.preference.ListPreference;
8. **import** android.preference.PreferenceFragment;
10. **public** **class** SettingFragment **extends** PreferenceFragment **implements** SharedPreferences.OnSharedPreferenceChangeListener {
11. **private** EditTextPreference editTextPreference;
12. **private** ListPreference listPreference;
13. **private** CheckBoxPreference checkBoxPreference;
15. @Override
16. **public** **void** onCreate(Bundle saved) {
17. **super**.onCreate(saved);
18. addPreferencesFromResource(R.xml.preferences);
19. editTextPreference = (EditTextPreference) findPreference("edittext\_key");
20. listPreference = (ListPreference) findPreference("list\_key");
21. checkBoxPreference = (CheckBoxPreference) findPreference("checkbox\_key");
22. }
24. @Override
25. **public** **void** onResume() {
26. **super**.onResume();
27. SharedPreferences sharedPreferences = getPreferenceScreen().getSharedPreferences();
28. listPreference.setSummary(sharedPreferences.getString("list\_key", ""));
29. editTextPreference.setSummary(sharedPreferences.getString("edittext\_key", ""));
30. sharedPreferences.registerOnSharedPreferenceChangeListener(**this**);
31. }
33. @Override
34. **public** **void** onPause() {
35. **super**.onPause();
36. getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(**this**);
37. }
39. @Override
40. **public** **void** onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
41. **if** (key.equals("edittext\_key"))
42. editTextPreference.setSummary(sharedPreferences.getString(key, "30"));
43. **else** **if** (key.equals("list\_key"))
44. listPreference.setSummary(sharedPreferences.getString(key, ""));
45. }
46. }

**（5）StatisticsActivity.java:**

1. **package** com.example.grandwordremember;
3. **import** ...
5. **public** **class** StatisticsAcitivity **extends** AppCompatActivity {
6. Uri testdb = Uri.parse("content://com.example.grandwordremember.TestDB/");
7. ContentResolver resolver;
8. ArrayList<Map<String, Object>> lists = **new** ArrayList<>();
9. SimpleAdapter adapter;
10. ListView listView;
11. AlertDialog alertDialog;
13. Comparator<Map<String, Object>> comparator = **new** Comparator<Map<String, Object>>() {
14. @Override
15. **public** **int** compare(Map<String, Object> stringObjectMap, Map<String, Object> t1) {
16. **if** (stringObjectMap.get("word").toString().toLowerCase().compareTo(t1.get("word").toString().toLowerCase()) < 0)
17. **return** -1;
18. **else**
19. **return** 1;
20. }
21. };
23. @Override
24. **protected** **void** onCreate(Bundle savedInstanceState) {
25. **super**.onCreate(savedInstanceState);
26. setContentView(R.layout.acitivity\_statistics);
27. android.support.v7.widget.Toolbar toolbar = findViewById(R.id.statistics\_toolbar);
28. setSupportActionBar(toolbar);
29. ActionBar actionBar = getSupportActionBar();
30. actionBar.setDisplayHomeAsUpEnabled(**true**);
31. actionBar.setTitle("学霸背单词");
32. actionBar.setSubtitle("测试统计");
33. resolver = getContentResolver();
34. listView = (ListView) findViewById(R.id.statistics\_listview);
35. display();
36. }
38. @Override
39. **public** **boolean** onCreateOptionsMenu(Menu menu) {
40. getMenuInflater().inflate(R.menu.menu\_statistics, menu);
41. **return** **true**;
42. }
44. @Override
45. **public** **boolean** onOptionsItemSelected(MenuItem item) {
46. **switch** (item.getItemId()) {
47. **case** R.id.statistics\_search:
48. search\_word();
49. **break**;
50. **case** android.R.id.home:
51. **this**.finish();
52. **break**;
53. }
54. **return** **true**;
55. }
57. **public** **void** display() {
58. Cursor cursor = resolver.query(testdb, **new** String[]{"word", "level", "test\_count", "correct\_count"},
59. **null**, **null**, "word");
60. change\_lists(cursor);
61. setListView();
62. }
64. **public** **void** change\_lists(Cursor cursor) {
65. lists.clear();
66. **while** (cursor.moveToNext()) {
67. Map<String, Object> map = **new** HashMap<>();
68. map.put("word", cursor.getString(cursor.getColumnIndex("word")));
69. map.put("level", cursor.getString(cursor.getColumnIndex("level")));
70. map.put("test\_count", cursor.getInt(cursor.getColumnIndex("test\_count")));
71. map.put("correct\_count", cursor.getInt(cursor.getColumnIndex("correct\_count")));
72. lists.add(map);
73. }
74. Collections.sort(lists, comparator);
75. }
77. **public** **void** setListView() {
78. adapter = **new** SimpleAdapter(**this**, lists, R.layout.statistics\_item, **new** String[]{"word", "level", "test\_count", "correct\_count"},
79. **new** **int**[]{R.id.statistics\_word, R.id.statistics\_level, R.id.test\_count, R.id.correct\_count});
80. listView.setAdapter(adapter);
81. }
83. **public** **void** search\_word() {
84. TableLayout wordform = (TableLayout) getLayoutInflater().inflate(R.layout.search\_word, **null**);
85. AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);
86. alertDialog = builder.setIcon(R.drawable.dict)
87. .setTitle("查询单词")
88. .setView(wordform)
89. .setNegativeButton("取消", **null**)
90. .setPositiveButton("确定", **new** DialogInterface.OnClickListener() {
91. @Override
92. **public** **void** onClick(DialogInterface dialogInterface, **int** i) {
93. EditText searchword = (EditText) alertDialog.findViewById(R.id.search\_word);
94. Cursor cursor = resolver.query(testdb, **new** String[]{"word", "level", "test\_count", "correct\_count"},
95. "word like ?", **new** String[]{searchword.getText().toString() + "%"},
96. "word");
97. change\_lists(cursor);
98. adapter.notifyDataSetChanged();
99. }
100. }).create();
101. alertDialog.show();
102. }
103. }

**（6）SystemSettingActivity.java:**

1. **package** com.example.grandwordremember;
3. **import** android.app.FragmentManager;
4. **import** android.app.FragmentTransaction;
5. **import** android.os.Bundle;
6. **import** android.support.v7.app.ActionBar;
7. **import** android.support.v7.app.AppCompatActivity;
8. **import** android.view.Menu;
9. **import** android.view.MenuItem;
11. **public** **class** SystemSettingAcitivity **extends** AppCompatActivity {
12. @Override
13. **protected** **void** onCreate(Bundle saved) {
14. **super**.onCreate(saved);
15. setContentView(R.layout.acitivity\_systemsetting);
16. android.support.v7.widget.Toolbar toolbar = findViewById(R.id.system\_settings\_toolbar);
17. setSupportActionBar(toolbar);
18. ActionBar actionBar = getSupportActionBar();
19. actionBar.setDisplayHomeAsUpEnabled(**true**);
20. actionBar.setTitle("学霸背单词");
21. actionBar.setSubtitle("系统设置");
22. FragmentManager fragmentManager = getFragmentManager();
23. FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
24. SettingFragment settingFragment = **new** SettingFragment();
25. fragmentTransaction.add(R.id.content\_layout, settingFragment);
26. fragmentTransaction.commit();
27. }
29. @Override
30. **public** **boolean** onCreateOptionsMenu(Menu menu) {
31. getMenuInflater().inflate(R.menu.menu\_systemsetting, menu);
32. **return** **true**;
33. }
35. @Override
36. **public** **boolean** onOptionsItemSelected(MenuItem item) {
37. **if** (item.getItemId() == android.R.id.home)
38. **this**.finish();
39. **return** **true**;
40. }
41. }

**（7）TestActivity.java:**

1. **package** com.example.grandwordremember;
3. **import** ...
5. **public** **class** TestActivity **extends** AppCompatActivity {
6. Uri uri = Uri.parse("content://com.example.providers.DictProvider/");
7. Uri testdb = Uri.parse("content://com.example.grandwordremember.TestDB/");
8. ContentResolver resolver;
9. ArrayList<WordRec> word\_lists;
10. TestAdapter adapter;
11. SharedPreferences settings;
13. @Override
14. **protected** **void** onCreate(Bundle savedInstanceState) {
15. **super**.onCreate(savedInstanceState);
16. setContentView(R.layout.activity\_test);
17. android.support.v7.widget.Toolbar toolbar = findViewById(R.id.test\_toolbar);
18. setSupportActionBar(toolbar);
19. ActionBar actionBar = getSupportActionBar();
20. actionBar.setDisplayHomeAsUpEnabled(**true**);
21. actionBar.setTitle("学霸背单词");
22. actionBar.setSubtitle("单词测试");
23. resolver = getContentResolver();
24. settings = PreferenceManager.getDefaultSharedPreferences(**this**);
25. }
27. @Override
28. **public** **boolean** onCreateOptionsMenu(Menu menu) {
29. getMenuInflater().inflate(R.menu.menu\_test, menu);
30. **return** **true**;
31. }
33. @Override
34. **public** **boolean** onOptionsItemSelected(MenuItem item) {
35. **switch** (item.getItemId()) {
36. **case** R.id.start\_test:
37. testing();
38. **break**;
39. **case** R.id.start\_review:
40. reviewing();
41. **break**;
42. **case** android.R.id.home:
43. **this**.finish();
44. **break**;
45. }
46. **return** **true**;
47. }
49. **public** **void** testing() {
50. Set set = **new** HashSet();
51. Cursor cursor = resolver.query(testdb, **new** String[]{"\_id", "word", "level"}, **null**, **null**, **null**);
52. **while** (cursor.moveToNext())
53. set.add(cursor.getString(cursor.getColumnIndex("word")));
54. cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"}, **null**, **null**, "word");
55. word\_lists = **new** ArrayList<>();
56. **int** count = 0;
57. Log.v("setting", settings.getString("edittext\_key", ""));
58. **while** (cursor.moveToNext() && count < Integer.parseInt(settings.getString("edittext\_key", ""))) {
59. **if** (!set.contains(cursor.getString(cursor.getColumnIndex("word")))) {
60. Log.v("count", count + "");
61. Log.v("size", word\_lists.size() + "");
62. WordRec wordRec = **new** WordRec();
63. wordRec.setId(cursor.getInt(cursor.getColumnIndex("\_id")));
64. wordRec.setWord(cursor.getString(cursor.getColumnIndex("word")));
65. String explan = cursor.getString(cursor.getColumnIndex("explanation"));
66. wordRec.setExplanation(explan.substring(explan.indexOf("]") + 1));
67. wordRec.setLevel(cursor.getInt(cursor.getColumnIndex("level")));
68. word\_lists.add(wordRec);
69. count++;
70. }
71. }
72. Log.v("word\_lists", word\_lists.toString());
73. adapter = **new** TestAdapter(word\_lists, **this**);
74. adapter.color = settings.getString("list\_key", "");
75. ListView listView = (ListView) findViewById(R.id.listview);
76. listView.setAdapter(adapter);
77. **final** Button button = (Button) findViewById(R.id.submit);
78. button.setVisibility(View.VISIBLE);
79. button.setOnClickListener(**new** Button.OnClickListener(){
80. @Override
81. **public** **void** onClick(View v) {
82. button.setVisibility(View.INVISIBLE);
83. adapter.show\_answer = **true**;
84. adapter.notifyDataSetChanged();
85. **if** (!settings.getBoolean("checkbox\_key", **true**))
86. **return**;
87. **for** (**int** i = 0; i < word\_lists.size(); ++i) {
88. String word = word\_lists.get(i).getWord();
89. ContentValues contentValues = **new** ContentValues();
90. contentValues.put("word", word);
91. contentValues.put("level", word\_lists.get(i).getLevel());
92. contentValues.put("test\_count", 1);
93. **if** (adapter.choose.keySet().contains(word) &&
94. word\_lists.get(i).getExplanation().equals(adapter.choose.get(word)))
95. contentValues.put("correct\_count", 1);
96. resolver.insert(testdb, contentValues);
97. Log.v("insert", contentValues.toString());
98. }
99. }
100. });
101. }
103. **public** **void** reviewing() {
104. ActionBar actionBar = getSupportActionBar();
105. actionBar.setSubtitle("单词复习");
106. Set set = **new** HashSet();
107. Cursor cursor = resolver.query(testdb, **new** String[]{"word"}, **null**, **null**, **null**);
108. **while** (cursor.moveToNext())
109. set.add(cursor.getString(cursor.getColumnIndex("word")));
110. cursor = resolver.query(uri, **new** String[]{"\_id", "word", "explanation", "level"}, **null**, **null**, "word");
111. word\_lists = **new** ArrayList<>();
112. **int** count = 0;
113. **while** (cursor.moveToNext() && count < set.size() && set.contains(cursor.getString(cursor.getColumnIndex("word")))) {
114. WordRec wordRec = **new** WordRec();
115. wordRec.setId(cursor.getInt(cursor.getColumnIndex("\_id")));
116. wordRec.setWord(cursor.getString(cursor.getColumnIndex("word")));
117. String explan = cursor.getString(cursor.getColumnIndex("explanation"));
118. wordRec.setExplanation(explan.substring(explan.indexOf("]") + 1));
119. wordRec.setLevel(cursor.getInt(cursor.getColumnIndex("level")));
120. word\_lists.add(wordRec);
121. count++;
122. }
123. adapter = **new** TestAdapter(word\_lists, **this**);
124. adapter.color = settings.getString("list\_key", "");
125. ListView listView = (ListView) findViewById(R.id.listview);
126. listView.setAdapter(adapter);
127. **final** Button button = (Button) findViewById(R.id.submit);
128. button.setVisibility(View.VISIBLE);
129. button.setOnClickListener(**new** Button.OnClickListener(){
130. @Override
131. **public** **void** onClick(View v) {
132. button.setVisibility(View.INVISIBLE);
133. adapter.show\_answer = **true**;
134. adapter.notifyDataSetChanged();
135. **if** (!settings.getBoolean("checkbox\_key", **true**))
136. **return**;
137. **for** (**int** i = 0; i < word\_lists.size(); ++i) {
138. String word = word\_lists.get(i).getWord();
139. ContentValues contentValues = **new** ContentValues();
140. contentValues.put("word", word);
141. contentValues.put("level", word\_lists.get(i).getLevel());
142. Cursor cs = resolver.query(testdb, **new** String[]{"test\_count", "correct\_count"},
143. "word = ?", **new** String[]{word}, **null**);
144. cs.moveToFirst();
145. contentValues.put("test\_count", cs.getInt(cs.getColumnIndex("test\_count")) + 1);
146. **if** (adapter.choose.keySet().contains(word) &&
147. word\_lists.get(i).getExplanation().equals(adapter.choose.get(word)))
148. contentValues.put("correct\_count", cs.getInt(cs.getColumnIndex("correct\_count")) + 1);
149. resolver.update(testdb, contentValues, "word = ?", **new** String[]{word});
150. Log.v("update", contentValues.toString());
151. }
152. }
153. });
154. }
155. }

**（8）TestAdapter.java:**

1. **package** com.example.grandwordremember;
3. **import** ...
4. **public** **class** TestAdapter **extends** BaseAdapter {
5. **private** ArrayList<WordRec> lists;
6. **private** Context context;
7. **private** SparseIntArray checked = **new** SparseIntArray();
8. LayoutInflater inflater;
9. Map<String, String> choose = **new** HashMap<>();
10. **boolean** show\_answer = **false**;
11. String color = "Black";
12. Map<Object, ArrayList<Object>> maps = **new** HashMap<>();
14. **public** TestAdapter(ArrayList<WordRec> lists, Context context) {
15. **super**();
16. **this**.lists = lists;
17. **this**.context = context;
18. inflater = LayoutInflater.from(context);
19. }
20. @Override
21. **public** **int** getCount() { **return** lists.size(); }
22. @Override
23. **public** Object getItem(**int** position) { **return** lists.get(position); }
24. @Override
25. **public** **long** getItemId(**int** position) { **return** position; }
26. **public** **class** ViewHolder{
27. TextView textView;
28. RadioGroup radioGroup;
29. RadioButton radioButton1;
30. RadioButton radioButton2;
31. RadioButton radioButton3;
32. RadioButton radioButton4;
33. }
34. @Override
35. **public** View getView(**final** **int** position, View convertView, ViewGroup parent) {
36. **final** ViewHolder holder;
37. **if** (convertView == **null**) {
38. convertView = inflater.inflate(R.layout.list\_item, **null**);
39. holder = **new** ViewHolder();
40. holder.textView = (TextView) convertView.findViewById(R.id.word);
41. holder.radioGroup = (RadioGroup) convertView.findViewById(R.id.radio\_group);
42. holder.radioButton1 = (RadioButton) convertView.findViewById(R.id.radio\_button1);
43. holder.radioButton2 = (RadioButton) convertView.findViewById(R.id.radio\_button2);
44. holder.radioButton3 = (RadioButton) convertView.findViewById(R.id.radio\_button3);
45. holder.radioButton4 = (RadioButton) convertView.findViewById(R.id.radio\_button4);
46. convertView.setTag(holder);
47. }
48. **else**
49. holder = (ViewHolder) convertView.getTag();
50. holder.textView.setText(lists.get(position).getWord());
51. **switch** (color) {
52. **case** "Black":
53. holder.textView.setTextColor(Color.BLACK);
54. **break**;
55. **case** "Red":
56. holder.textView.setTextColor(Color.RED);
57. **break**;
58. **case** "Blue":
59. holder.textView.setTextColor(Color.BLUE);
60. **break**;
61. **case** "Green":
62. holder.textView.setTextColor(Color.GREEN);
63. **break**;
64. }
65. ArrayList<Object> list = **new** ArrayList<>();
66. **if** (!maps.containsKey(position)) {
67. list.add(position);
68. **for** (**int** i = 0; i < 3; ++i) {
69. **int** num = position;
70. **while** (list.contains(num)) num = (**int**) (Math.random() \* lists.size());
71. list.add(num);
72. }
73. Collections.shuffle(list);
74. maps.put(position, list);
75. }
77. **else**
78. list = maps.get(position);
79. holder.radioButton1.setText(lists.get((**int**)list.get(0)).getExplanation());
80. holder.radioButton2.setText(lists.get((**int**)list.get(1)).getExplanation());
81. holder.radioButton3.setText(lists.get((**int**)list.get(2)).getExplanation());
82. holder.radioButton4.setText(lists.get((**int**)list.get(3)).getExplanation());
83. holder.radioButton1.setBackgroundColor(Color.WHITE);
84. holder.radioButton2.setBackgroundColor(Color.WHITE);
85. holder.radioButton3.setBackgroundColor(Color.WHITE);
86. holder.radioButton4.setBackgroundColor(Color.WHITE);
87. **if** (show\_answer)
88. **switch** (list.indexOf(position)) {
89. **case** 0:
90. holder.radioButton1.setBackgroundColor(Color.rgb(240, 240, 240));
91. **break**;
92. **case** 1:
93. holder.radioButton2.setBackgroundColor(Color.rgb(240, 240, 240));
94. **break**;
95. **case** 2:
96. holder.radioButton3.setBackgroundColor(Color.rgb(240, 240, 240));
97. **break**;
98. **case** 3:
99. holder.radioButton4.setBackgroundColor(Color.rgb(240, 240, 240));
100. **break**;
101. }
102. holder.radioGroup.setOnCheckedChangeListener(**null**);
103. holder.radioGroup.clearCheck();
104. **if** (checked.indexOfKey(position) > -1)
105. holder.radioGroup.check(checked.get(position));
106. **else**
107. holder.radioGroup.clearCheck();
108. holder.radioGroup.setOnCheckedChangeListener(**new** RadioGroup.OnCheckedChangeListener() {
109. @Override
110. **public** **void** onCheckedChanged(RadioGroup radioGroup, **int** i) {
111. **if** (i > -1) {
112. checked.put(position, i);
113. RadioButton radioButton = (RadioButton) radioGroup.findViewById(i);
114. choose.put(holder.textView.getText().toString(), radioButton.getText().toString());
115. }
116. **else** **if** (checked.indexOfKey(position) > -1)
117. checked.removeAt(checked.indexOfKey(position));
118. }
119. });
120. **return** convertView;
121. }
122. }

**（9）TestDB.java:**

1. **package** com.example.grandwordremember;
3. **import** android.content.ContentProvider;
4. **import** android.content.ContentValues;
5. **import** android.database.Cursor;
6. **import** android.database.sqlite.SQLiteDatabase;
7. **import** android.net.Uri;
9. **public** **class** TestDB **extends** ContentProvider {
10. **private** DBOpenHandler dbOpenHandler;
11. @Override
12. **public** **boolean** onCreate() {
13. **this**.dbOpenHandler = **new** DBOpenHandler(getContext(), "words", **null**, 1);
14. **return** **true**;
15. }
16. @Override
17. **public** String getType(Uri uri) { **return** **null**; }
18. @Override
19. **public** Uri insert(Uri uri, ContentValues contentValues) {
20. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
21. db.insert("words", **null**, contentValues);
22. db.close();
23. **return** uri;
24. }
25. @Override
26. **public** **int** delete(Uri uri, String where, String[] whereargs) {
27. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
28. **int** num = db.delete("words", where, whereargs);
29. db.close();
30. **return** num;
31. }
32. @Override
33. **public** **int** update(Uri uri, ContentValues contentValues, String where, String[] whereargs) {
34. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
35. **int** num = db.update("words", contentValues, where, whereargs);
36. db.close();
37. getContext().getContentResolver().notifyChange(uri, **null**);
38. **return** num;
39. }
40. @Override
41. **public** Cursor query(Uri uri, String[] projection, String where, String[] whereargs, String sortorder) {
42. SQLiteDatabase db = dbOpenHandler.getWritableDatabase();
43. **return** db.query("words", projection, where, whereargs, **null**, **null**, sortorder, **null**);
44. }
45. }

**（10）TestRec.java:**

1. **package** com.example.grandwordremember;
3. **public** **class** TestRec {
4. **private** **int** \_id;
5. **private** String word;
6. **private** **int** level;
7. **private** **int** test\_count;
8. **private** **int** correct\_count;
9. **public** TestRec(**int** id, String word, **int** level, **int** test\_count, **int** correct\_count) {
10. **this**.\_id = id;
11. **this**.word = word;
12. **this**.level = level;
13. **this**.test\_count = test\_count;
14. **this**.correct\_count = correct\_count;
15. }
16. **public** **int** get\_id() { **return** \_id; }
17. **public** **void** set\_id(**int** id) { **this**.\_id = id; }
18. **public** String getWord() { **return** word; }
19. **public** **void**  setWord(String word) { **this**.word = word; }
20. **public** **int** getLevel() { **return** level; }
21. **public** **void** setLevel(**int** level) { **this**.level = level; }
22. **public** **int** getTest\_count() { **return** test\_count; }
23. **public** **void** setTest\_count(**int** test\_count) { **this**.test\_count = test\_count; }
24. **public** **int** getCorrect\_count() { **return** correct\_count; }
25. **public** **void** setCorrect\_count(**int** correct\_count) { **this**.correct\_count = correct\_count; }
26. }

**（11）WordRec.java:**

1. **package** com.example.grandwordremember;
3. **public** **class** WordRec {
4. **private** **int** id;
5. **private** String word;
6. **private** String explanation;
7. **private** **int** level;
8. WordRec() {}
9. WordRec(**int** id, String word, String explanation, **int** level) {
10. **this**.id = id;
11. **this**.word = word;
12. **this**.explanation = explanation;
13. **this**.level = level;
14. }
15. **public** **int** getId() { **return** id; }
16. **public** **void** setId(**int** id) { **this**.id = id; }
17. **public** String getWord() { **return** word; }
18. **public** **void** setWord(String word) { **this**.word = word; }
19. **public** String getExplanation() { **return** explanation; }
20. **public** **void** setExplanation(String explanation) { **this**.explanation = explanation; }
21. **public** **int** getLevel() { **return** level; }
22. **public** **void** setLevel(**int** level) { **this**.level = level; }
23. }

**（12）activity\_statistics.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout**
3. xmlns:android="http://schemas.android.com/apk/res/android"
4. xmlns:app="http://schemas.android.com/apk/res-auto"
5. android:layout\_width="match\_parent"
6. android:layout\_height="match\_parent"**>**
7. **<android.support.design.widget.AppBarLayout**
8. android:layout\_width="match\_parent"
9. android:layout\_height="wrap\_content"
10. android:theme="@style/AppTheme.AppBarOverlay"**>**
11. **<android.support.v7.widget.Toolbar**
12. android:layout\_width="match\_parent"
13. android:layout\_height="?attr/actionBarSize"
14. android:id="@+id/statistics\_toolbar"
15. android:background="?attr/colorPrimary"
16. app:popupTheme="@style/AppTheme.PopupOverlay"**/>**
17. **</android.support.design.widget.AppBarLayout>**
18. **<LinearLayout**
19. android:layout\_width="match\_parent"
20. android:layout\_height="wrap\_content"
21. android:layout\_marginTop="150px"
22. android:layout\_marginLeft="10px"
23. android:layout\_marginRight="10px"**>**
24. **<TextView**
25. android:layout\_width="0dp"
26. android:layout\_height="wrap\_content"
27. android:layout\_weight="0.52"
28. android:text="单词"
29. android:id="@+id/statistics\_word"
30. android:textSize="25sp"
31. android:layout\_margin="5px"**/>**
32. **<TextView**
33. android:layout\_width="0dp"
34. android:layout\_height="wrap\_content"
35. android:layout\_weight="0.16"
36. android:text="级别"
37. android:id="@+id/statistics\_level"
38. android:textSize="18sp"
39. android:layout\_margin="5px"**/>**
40. **<TextView**
41. android:layout\_width="0dp"
42. android:layout\_height="wrap\_content"
43. android:layout\_weight="0.16"
44. android:text="测试"
45. android:id="@+id/test\_count"
46. android:textSize="18sp"
47. android:layout\_margin="5px"**/>**
48. **<TextView**
49. android:layout\_width="0dp"
50. android:layout\_height="wrap\_content"
51. android:layout\_weight="0.16"
52. android:text="正确"
53. android:id="@+id/correct\_count"
54. android:textSize="18sp"
55. android:layout\_margin="5px"**/>**
56. **</LinearLayout>**
57. **<ListView**
58. android:layout\_width="match\_parent"
59. android:layout\_height="740px"
60. android:layout\_marginTop="280px"
61. android:layout\_marginLeft="10px"
62. android:layout\_marginRight="10px"
63. android:id="@+id/statistics\_listview"**/>**
64. **</RelativeLayout>**

**（13）activity\_systemsetting.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. xmlns:app="http://schemas.android.com/apk/res-auto"
5. android:layout\_height="match\_parent"**>**
6. **<android.support.design.widget.AppBarLayout**
7. android:layout\_width="match\_parent"
8. android:layout\_height="wrap\_content"
9. android:theme="@style/AppTheme.AppBarOverlay"**>**
10. **<android.support.v7.widget.Toolbar**
11. android:layout\_width="match\_parent"
12. android:layout\_height="?attr/actionBarSize"
13. android:id="@+id/system\_settings\_toolbar"
14. android:background="?attr/colorPrimary"
15. app:popupTheme="@style/AppTheme.PopupOverlay"**/>**
16. **</android.support.design.widget.AppBarLayout>**
17. **<FrameLayout**
18. android:layout\_width="match\_parent"
19. android:layout\_height="match\_parent"
20. android:id="@+id/content\_layout"
21. android:layout\_marginTop="60dp"**/>**
22. **</RelativeLayout>**

**（14）activity\_main.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"
4. xmlns:tools="http://schemas.android.com/tools"
5. android:layout\_width="match\_parent"
6. android:layout\_height="match\_parent"
7. tools:context=".MainActivity"**>**
9. **<android.support.design.widget.AppBarLayout**
10. android:layout\_width="match\_parent"
11. android:layout\_height="wrap\_content"
12. android:theme="@style/AppTheme.AppBarOverlay"**>**
14. **<android.support.v7.widget.Toolbar**
15. android:id="@+id/toolbar"
16. android:layout\_width="match\_parent"
17. android:layout\_height="?attr/actionBarSize"
18. android:background="?attr/colorPrimary"
19. app:popupTheme="@style/AppTheme.PopupOverlay" **/>**
20. **</android.support.design.widget.AppBarLayout>**
22. **<ImageView**
23. android:layout\_width="match\_parent"
24. android:layout\_height="wrap\_content"
25. android:layout\_centerVertical="true"
26. android:id="@+id/image"
27. android:src="@drawable/rem"**/>**
29. **<ListView**
30. android:layout\_width="match\_parent"
31. android:layout\_height="match\_parent"
32. android:id="@+id/find\_list\_view"
33. android:layout\_marginTop="80dp"
34. android:layout\_marginLeft="10dp"
35. android:layout\_marginRight="10dp"
36. android:layout\_marginBottom="10dp"**/>**
38. **</RelativeLayout>**

**（15）activity\_test.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"
4. android:layout\_width="match\_parent"
5. android:layout\_height="match\_parent"**>**
7. **<android.support.design.widget.AppBarLayout**
8. android:layout\_width="match\_parent"
9. android:layout\_height="wrap\_content"
10. android:theme="@style/AppTheme.AppBarOverlay"**>**
12. **<android.support.v7.widget.Toolbar**
13. android:id="@+id/test\_toolbar"
14. android:layout\_width="match\_parent"
15. android:layout\_height="?attr/actionBarSize"
16. android:background="?attr/colorPrimary"
17. app:popupTheme="@style/AppTheme.PopupOverlay" **/>**
18. **</android.support.design.widget.AppBarLayout>**
20. **<ListView**
21. android:id="@+id/listview"
22. android:layout\_width="match\_parent"
23. android:layout\_height="400dp"
24. android:layout\_marginTop="100dp"
25. android:fastScrollEnabled="true" **/>**
26. **<Button**
27. android:layout\_width="80dp"
28. android:layout\_height="wrap\_content"
29. android:id="@+id/submit"
30. android:text="提交"
31. android:visibility="invisible"
32. android:textSize="18sp"
33. android:layout\_centerHorizontal="true"
34. android:gravity="center"
35. android:layout\_below="@id/listview"**/>**
36. **</RelativeLayout>**

**（16）list\_item.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"**>**
5. **<TextView**
6. android:layout\_width="wrap\_content"
7. android:layout\_height="wrap\_content"
8. android:id="@+id/word"
9. android:text="单词"
10. android:paddingLeft="10dp"
11. android:paddingTop="10dp"
12. android:paddingBottom="20dp"
13. android:textColor="@color/red"
14. android:textSize="18sp"**/>**
15. **<RadioGroup**
16. android:layout\_width="wrap\_content"
17. android:layout\_height="wrap\_content"
18. android:id="@+id/radio\_group"
19. android:paddingLeft="10dp"
20. android:paddingBottom="10dp"
21. android:layout\_below="@id/word"
22. android:orientation="vertical"**>**
23. **<RadioButton**
24. android:layout\_width="wrap\_content"
25. android:layout\_height="wrap\_content"
26. android:id="@+id/radio\_button1"
27. android:text="意思1"**/>**
28. **<RadioButton**
29. android:layout\_width="wrap\_content"
30. android:layout\_height="wrap\_content"
31. android:id="@+id/radio\_button2"
32. android:text="意思2"**/>**
33. **<RadioButton**
34. android:layout\_width="wrap\_content"
35. android:layout\_height="wrap\_content"
36. android:id="@+id/radio\_button3"
37. android:text="意思3"**/>**
38. **<RadioButton**
39. android:layout\_width="wrap\_content"
40. android:layout\_height="wrap\_content"
41. android:id="@+id/radio\_button4"
42. android:text="意思4"**/>**
43. **</RadioGroup>**
45. **</RelativeLayout>**

**（17）list\_meaning.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"**>**
5. **<LinearLayout**
6. android:layout\_width="match\_parent"
7. android:layout\_height="wrap\_content"
8. android:orientation="vertical"**>**
9. **<TextView**
10. android:layout\_width="wrap\_content"
11. android:layout\_height="wrap\_content"
12. android:id="@+id/word\_explanation"
13. android:gravity="center\_vertical"
14. android:text="单词"
15. android:textSize="18sp"**/>**
16. **<TextView**
17. android:layout\_width="wrap\_content"
18. android:layout\_height="wrap\_content"
19. android:id="@+id/explanation"
20. android:gravity="center\_vertical"
21. android:text="意思"
22. android:textSize="18sp"**/>**
23. **</LinearLayout>**
24. **</RelativeLayout>**

**（18）new\_word.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<TableLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:tools="http://schemas.android.com/tools"
4. android:layout\_width="match\_parent"
5. android:layout\_height="match\_parent"
6. android:id="@+id/wordform"
7. android:padding="12dp"**>**
8. **<TableRow** android:paddingBottom="10dp"
9. android:layout\_width="wrap\_content"
10. android:layout\_height="wrap\_content"**>**
11. **<TextView**
12. android:layout\_width="0px"
13. android:layout\_weight="0.2"
14. android:layout\_height="wrap\_content"
15. android:text="单词"
16. android:textSize="18sp" **/>**
17. **<EditText** android:id="@+id/new\_word"
18. android:layout\_height="wrap\_content"
19. android:layout\_width="0dp"
20. android:layout\_weight="0.8"
21. android:hint="请填写单词"
22. android:padding="10dp"
23. android:background="@drawable/edit\_border"
24. android:selectAllOnFocus="true"**/>**
25. **</TableRow>**
26. **<TableRow** android:paddingBottom="10dp"**>**
28. **<TextView**
29. android:layout\_width="0dp"
30. android:layout\_weight="0.2"
31. android:layout\_height="wrap\_content"
32. android:text="解释"
33. android:textSize="18sp" **/>**
34. **<EditText**
35. android:layout\_height="90dp"
36. android:layout\_width="0dp"
37. android:layout\_weight="0.8"
38. android:minLines="5"
39. android:gravity="top"
40. android:hint="请填写解释"
41. android:padding="10dp"
42. android:background="@drawable/edit\_border"
43. android:id="@+id/new\_explanation"**/>**
44. **</TableRow>**
45. **<TableRow** android:paddingBottom="10dp"**>**
46. **<TextView** android:text="级别"
47. android:textSize="18sp"
48. android:layout\_width="0dp"
49. android:layout\_weight="0.2"
50. android:layout\_height="wrap\_content"**/>**
51. **<EditText** android:id="@+id/new\_level"
52. android:layout\_height="wrap\_content"
53. android:layout\_width="0dp"
54. android:layout\_weight="0.8"
55. android:padding="10dp"
56. android:background="@drawable/edit\_border"
57. android:hint="请填写难度(0~8)"**/>**
58. **</TableRow>**
59. **<TableRow>**
60. **<CheckedTextView** android:id="@+id/check"
61. android:layout\_width="wrap\_content"
62. android:layout\_height="match\_parent"
63. android:checkMark="?android:attr/listChoiceIndicatorMultiple"
64. android:checked="true"
65. android:clickable="true"
66. android:gravity="center"
67. android:text="是否覆盖"
68. android:onClick="toggle"
69. android:textSize="18sp" **/>**
70. **</TableRow>**
71. **</TableLayout>**

**（19）search\_word.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<TableLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"
5. android:id="@+id/wordform"
6. android:padding="12dp"**>**
7. **<TableRow** android:layout\_width="wrap\_content"
8. android:layout\_height="wrap\_content"**>**
9. **<TextView** android:text="单词"
10. android:layout\_width="0px"
11. android:layout\_weight="0.15"
12. android:layout\_height="wrap\_content"
13. android:textSize="18sp"**/>**
14. **<EditText** android:id="@+id/search\_word"
15. android:layout\_width="0px"
16. android:layout\_weight="0.85"
17. android:layout\_height="wrap\_content"
18. android:hint="请填写单词"
19. android:padding="10dp"
20. android:background="@drawable/edit\_border"
21. android:selectAllOnFocus="true"**/>**
22. **</TableRow>**
23. **</TableLayout>**

**（20）statistics\_item.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<LinearLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"**>**
5. **<TextView**
6. android:layout\_width="0dp"
7. android:layout\_height="wrap\_content"
8. android:layout\_weight="0.52"
9. android:text="单词"
10. android:id="@+id/statistics\_word"
11. android:textSize="18sp"
12. android:layout\_margin="5px"**/>**
13. **<TextView**
14. android:layout\_width="0dp"
15. android:layout\_height="wrap\_content"
16. android:layout\_weight="0.16"
17. android:text="级别"
18. android:id="@+id/statistics\_level"
19. android:textSize="18sp"
20. android:layout\_margin="5px"**/>**
21. **<TextView**
22. android:layout\_width="0dp"
23. android:layout\_height="wrap\_content"
24. android:layout\_weight="0.16"
25. android:text="测试"
26. android:id="@+id/test\_count"
27. android:textSize="18sp"
28. android:layout\_margin="5px"**/>**
29. **<TextView**
30. android:layout\_width="0dp"
31. android:layout\_height="wrap\_content"
32. android:layout\_weight="0.16"
33. android:text="正确"
34. android:id="@+id/correct\_count"
35. android:textSize="18sp"
36. android:layout\_margin="5px"**/>**
37. **</LinearLayout>**

**（21）menu\_main.xml:**

1. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"
2. xmlns:app="http://schemas.android.com/apk/res-auto"
3. xmlns:tools="http://schemas.android.com/tools"
4. tools:context="com.example.grandwordremember.MainActivity"**>**
5. **<item** android:id="@+id/test"
6. android:title="进入测验"
7. app:showAsAction="ifRoom"**/>**
8. **<item** android:id="@+id/add\_word"
9. android:orderInCategory="0"
10. android:title="新增单词"
11. app:showAsAction="never" **/>**
12. **<item** android:id="@+id/learning\_statistics"
13. android:orderInCategory="1"
14. android:title="学习统计"
15. app:showAsAction="never"**/>**
16. **<item** android:id="@+id/find\_word"
17. android:orderInCategory="2"
18. android:title="查找单词"
19. app:showAsAction="never"**/>**
20. **<item** android:id="@+id/system\_settings"
21. android:orderInCategory="3"
22. android:title="系统设置"
23. app:showAsAction="never"**/>**
24. **</menu>**

**（22）menu\_statistics.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"**>**
4. **<item** android:id="@+id/statistics\_search"
5. android:icon="@android:drawable/ic\_menu\_search"
6. android:title="search"
7. app:showAsAction="ifRoom"**/>**
8. **</menu>**

**（23）menu\_systemsetting.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"**>**
3. **</menu>**

**（24）menu\_test.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<menu** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:app="http://schemas.android.com/apk/res-auto"**>**
4. **<item** android:id="@+id/start\_test"
5. android:title="开始测验"
6. app:showAsAction="ifRoom"**/>**
7. **<item** android:id="@+id/start\_review"
8. android:title="开始复习"
9. app:showAsAction="never"**/>**
10. **</menu>**

**（25）array.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<resources>**
3. **<string-array** name="list\_preference"**>**
4. **<item>**Red**</item>**
5. **<item>**Blue**</item>**
6. **<item>**Green**</item>**
7. **<item>**Black**</item>**
8. **</string-array>**
9. **</resources>**

**（26）preferences.xml:**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<PreferenceScreen**
3. xmlns:android="http://schemas.android.com/apk/res/android"**>**
4. **<PreferenceCategory** android:title="外观设置"**>**
5. **<ListPreference**
6. android:key="list\_key"
7. android:title="单词颜色"
8. android:summary="black"
9. android:defaultValue="black"
10. android:entries="@array/list\_preference"
11. android:entryValues="@array/list\_preference"
12. android:dialogTitle="选择单词颜色"**/>**
13. **</PreferenceCategory>**
14. **<PreferenceCategory** android:title="其他设置"**>**
15. **<EditTextPreference**
16. android:key="edittext\_key"
17. android:defaultValue="30"
18. android:summary="30"
19. android:numeric="integer"
20. android:title="测试单词数"**/>**
21. **<CheckBoxPreference**
22. android:key="checkbox\_key"
23. android:defaultValue="true"
24. android:summary="是"
25. android:title="保存时是否统计"**/>**
26. **</PreferenceCategory>**
27. **</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）背单词的查找单词，从词典中找出单词。 √

（3）背单词的统计加入首字母筛选查询。 √

【实验体会】

在第2步中，我尝试使用JSONObject.toBean()函数将Json格式的数据转成object，然而实际使用中并没有转换成功，所以我只好实现parserJSON函数将Json字符串转换成Object。然后将这个Object插入数据库中。

在第7步和第17步中，我遇到一个问题：选择第一个单词的词义后，滚动listview中的内容，发现后面单词的部分选项自动选中；点击提交后，滚动listview中的内容，发现部分单词的4个选项全部显示为正确选项。这就是listview中嵌套radiogroup出现的选中混乱问题。

解决方案：每次取消掉监听器，使用hashmap保存选中的radiobutton位置，再进行恢复就可以了。

在第8步和第12步中的“新增单词”功能中，“是否覆盖”选项在new\_word.xml文件中标签为CheckTextView，它的点击监听器需要手动设置，于是，我便在“新增单词”AlertDialog的PositiveButton的listener中创建“是否覆盖”的点击监听器。运行程序，测试其功能，发现点击“是否覆盖”后其选中状态并没有改变。Debug时，点击“是否覆盖”后程序并没有击中我在监听器中设置的断点，也就是说，这个监听器是无效的，我认为问题的原因可能是不能在监听器中创建一个新的监听器。

解决方案：在CheckTextView中增加属性onClick，值为toggle()函数，该函数的作用是改变“是否覆盖”的选中状态。

在第13步中，需要让词典直接显示语义，我最初的想法是调整第3步listview中item的高度，再增加单词词义的显示。然而，我遇到了一个问题：无法使用代码调整item的高度，它的高度在list\_item.xml文件中已确定无法修改。我在网上搜索具体的原因，原来item子控件的尺寸由父控件决定，并且子控件的尺寸信息存储在LayoutParams中，我调用View.inflate()方法传递的parent参数是null，所以导致item的layoutParams属性为null，也就是说无法通过设置item的layoutParams属性来调整item的尺寸。

解决方案：构建一个新的list\_meaning.xml文件来设置item的高度和item中的内容，构建一个新的adapter使用这个新的list\_meaning.xml文件来显示单词以及其词义。

在第17步中，我遇到一个问题：每次点击提交按钮后，每个单词的4个选项中的内容均会发生变动，这是因为在提交按钮上设置了一个监听器，每次点击这个按钮就会调用adapter.notifyDataSetChanged()函数来刷新listview以显示正确选项，在adapter中每个item会随机选择4个词义在选项中显示，也就是说，每点击提交按钮都会随机选择4个词义在选项中显示。

解决方案：在第一次显示选项时，使用hashmap保存每个单词对应的4个词义，在下次显示选项的词义时，根据hashmap进行恢复即可。

在第19步中，我遇到一个问题：当测试单词数设置为小于5的数时，进行单词测试，整个app将进入卡死状态。我推测它可能在某个地方进入死循环了，但是当我debug，一步一步运行程序时并没有找到它到底在哪个地方进入死循环了，我觉得这个错误很神奇，但又解决不了。不过，只要测试单词数大于5，整个app就会正常运行，所以我觉得这个问题应该无伤大雅。