手机应用程序设计-课程期末综合报告

**锁屏背单词设计**

**学号：2017010840**

**姓名：郭知军**

**目 录**

1. **开发背景**
2. **系统功能设计**

**第一节系统功能结构设计**

**第二节业务流程设计**

1. **开发环境搭建**
2. **资源导入方法**
3. **图片资源导入**
4. **数据库资源导入**
5. **语言资源导入**
6. **Greendao资源导入**
7. **Jar包的导入**
8. **界面设计**
9. **锁屏界面**
10. **控件设计**
11. **同步系统时间**

**第四节数据库文件获取**

**第五节手势时间处理**

1. **复习界面设计**
2. **复习界面布局设计**
3. **复习节目功能设计**
4. **设置界面设计**
5. **开关按钮设计**
6. **开关按钮功能**
7. **设置界面布局**
8. **设置界面功能**

**第八章错词本界面设计**

**第一节错词本界面布局**

**第二节错词本界面功能**

**第九章主界面设计**

1. **主界面布局**
2. **锁屏状态监听**
3. **主界面功能**

**第十章功能展示**

**总结和心得**

**第一章**

**开发背景**

随着社会的不断发展与进步，越来越多的人平均每天使用手机的时间超过6小时，智能手机发展的越来越迅速，现如今已经可以通过感应而控制手机屏幕的亮灭，而人们使用手机的频率增高，但是现如今对于手机的依赖也是越来越高据统计，每个人每天平均花费在手机上的时间大于睡觉和休闲娱乐以及学习的时间由此可见手机对我们来说仿佛就像鱼离不开水一样。

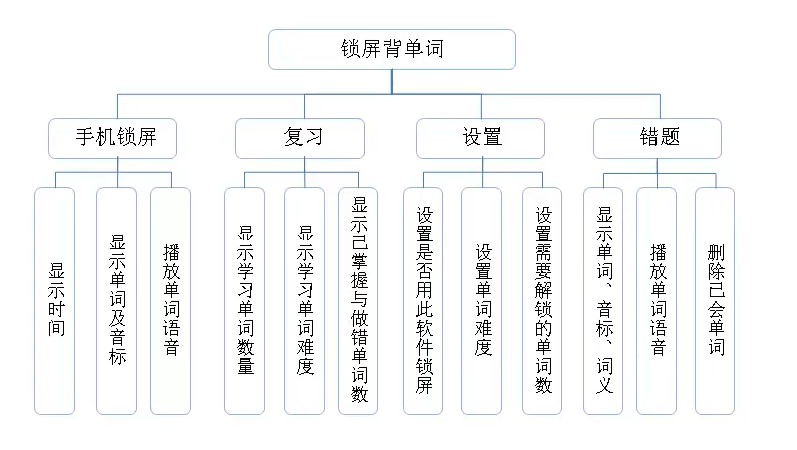
因此手机锁屏背单词就更适用于现在社会的发展了，锁屏背单词不仅适合现在的青少年同样适用于上班族等，在看手机的时候，通过锁屏可以学习到单词，每个单词用到的时间不超过30秒，就可以看一个单词，由此计算下来，半个小时可以看非常多的单词，不仅提高了学习的效率，更是符合现在的提倡“利用碎片时间学习”的原则，当人们拿起手机的时候，就可以看到不同的单词，根据单词选择意思，同时还可以听读音，大大的增加了学习的效果，走在路上，坐在车上、休闲娱乐的时候都可以拿起手机看上几个单词，再看单词的时候系统也会帮助你记录自己不会的和已经掌握的单词。这样的锁屏背单词，不仅让家长不用担心孩子因为玩手机导致成绩下降，同时也可以激发学生突破传统的用书本背单词的方式，拿起手机随时随地的看单词，增加了学生的主动学习的兴趣，同时也能随时掌握自己的学习成果可谓一举两得。

巧妙利用零散时间，让单词不再成为学习路上的障碍；从锁屏背单词开始，让解锁变得更有意义。

**第二章系统功能设计**

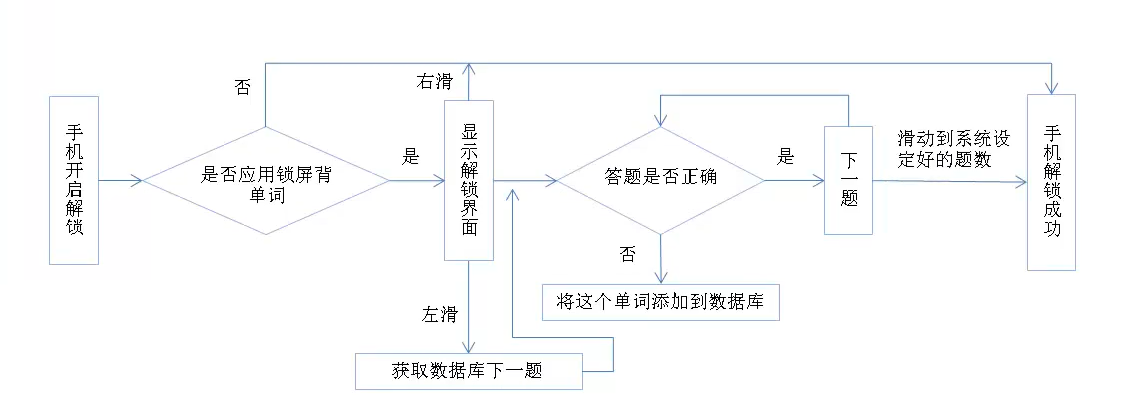
**第一节系统功能结构设计**

**2.1.1功能结构图**



**第二节业务流程设计**

**2.2.1业务流程图**

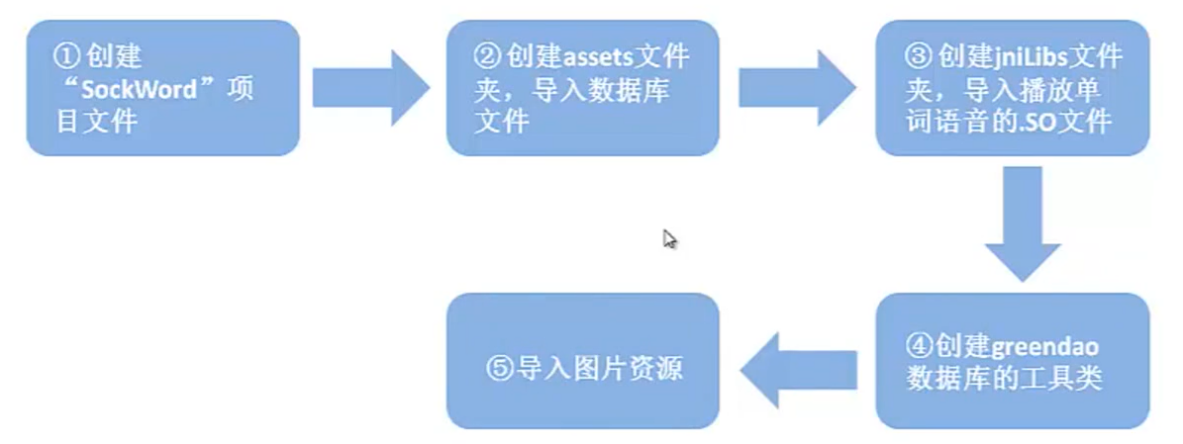


**第三章开发环境的搭建**

**3.1系统开发环境要求**



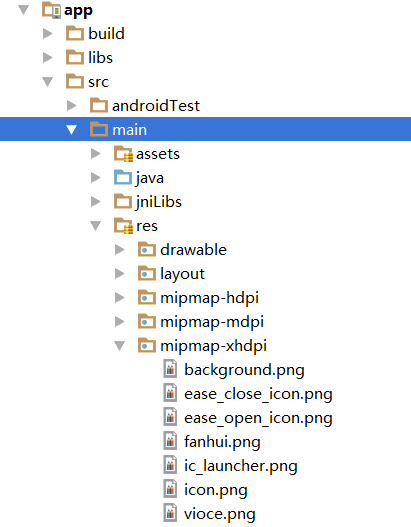
**3.2开发步骤**



**第四章资源导入方法**

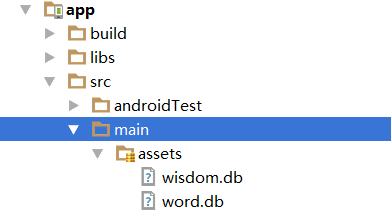
**第一节图片资源导入**

将所需要的图片复制粘贴到app/src/main/res/mipmap-xhdpi目录下



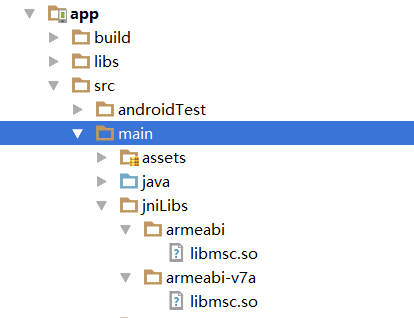
**第二节数据库资源导入**

在app/src/main目录下新建一个assets文件夹，复制数据库资源到该文件夹下。



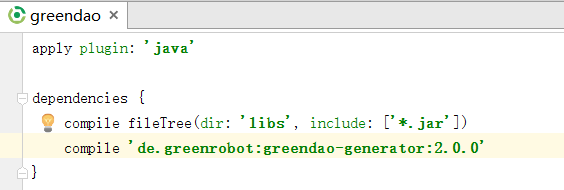
**第三节语言资源导入**

在app/src/main目录下新建一个jniLibs文件夹，复制语言文件到该文件夹下。

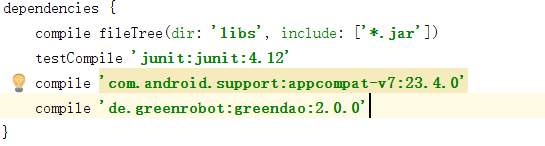


**第四节greendao资源导入**

在project目录下新建一个Module，命名为greendao，并在其build文件下设置依赖。



再在app目录下的build文件下同样设置依赖，最后进行同步。



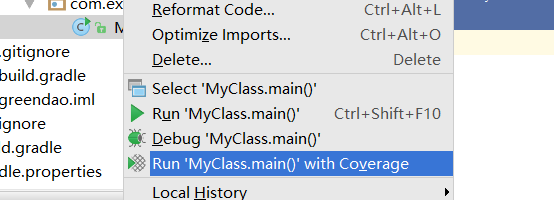
但两种依赖不一样，如果给app目录下的依赖设置为和greendao目录下一样，就会发生错误，myclass生成的文件找不到对应的包。

在greendao文件夹下编写myclass文件，使其生成数据库的工具类。

Myclass.java:

**package** com.example;  
  
**import** de.greenrobot.daogenerator.DaoGenerator;  
**import** de.greenrobot.daogenerator.Entity;  
**import** de.greenrobot.daogenerator.Schema;  
  
**public class** MyClass {  
 **public static void** main(String[] args) {  
 *// 创建Schema对象  
 // 构造方法第一个参数为数据库版本号  
 // 第二个参数为自动生成的实体类将要存放的位置,前面为我的Android Module的包名* Schema schema = **new** Schema(1000, **"com.li.greendao.entity.greendao"**);  
 *// 添加需要创建的实体类信息  
 addNote*(schema);  
 **try** {  
 *// 创建实体类.第二个参数填Android Module的路径* **new** DaoGenerator().generateAll(schema, **"./app/src/main/java"**);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* 添加将要创建的实体类的信息,会根据类名生成数据库的表,属性名生成数据库的字段<p>  
 \* 当然如果想建多张表,可以创建多个Entity对象  
 \*  
 \** ***@param schema*** *\*/* **private static void** addNote(Schema schema) {  
 *// 指定需要生成实体类的类名,类名确定了那么表名也是根据这个类名来自动命名的,例如下面这个,生成的表名叫做WisdomEntity* Entity entity = schema.addEntity(**"WisdomEntity"**);  
 *// 指定自增长主键* entity.addIdProperty().autoincrement().primaryKey();  
 *// 添加类的属性,根据属性生成数据库表中的字段* entity.addStringProperty(**"english"**);  
 entity.addStringProperty(**"china"**);  
  
 *// 指定需要生成实体类的类名,类名确定了那么表名也是根据这个类名来自动命名的* Entity entity1 = schema.addEntity(**"CET4Entity"**);  
 *// 指定自增长主键* entity1.addIdProperty().autoincrement().primaryKey();  
 *// 添加类的属性,根据属性生成数据库表中的字段* entity1.addStringProperty(**"word"**);  
 entity1.addStringProperty(**"english"**);  
 entity1.addStringProperty(**"china"**);  
 entity1.addStringProperty(**"sign"**);  
 }  
}

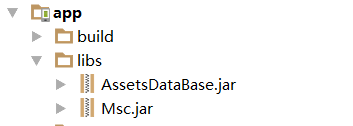
最后点击运行，在app/src/main/com.li目录下生成数据库工具类。





**第五节jar包的导入**

在app/libs文件夹下导入资源jar包，再进行同步。



注意:资源文件导入后需要同步，不然运行会导致错误。

**第五章界面设计**

**第一节锁屏界面**

**5.1.1设计样式**

****

**第二节控件设计**

**5.2.1activity\_main**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.li.sockwork.MainActivity"  
 android:orientation="vertical"  
 android:background="@mipmap/background"**>  
  
 <**TextView  
 android:id="@+id/time\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:gravity="center\_horizontal"  
 android:text="12:00"  
 android:textColor="@android:color/blue"  
 android:textSize="50dp"** />  
  
  
  
 <**TextView  
 android:id="@+id/date\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp"  
 android:gravity="center\_horizontal"  
 android:text="5月20日 星期三"  
 android:textColor="@android:color/blue"  
  
 android:textSize="17dp"** />  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="85dp"  
 android:layout\_marginTop="40dp"**>  
  
 <**TextView  
 android:id="@+id/word\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:text="parent"  
 android:textColor="@android:color/blue"  
 android:textSize="50dp"** />  
  
 <**ImageView  
 android:id="@+id/play\_voice"  
 android:layout\_width="27dp"  
 android:layout\_height="27dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="60dp"  
 android:background="@mipmap/vioce"** />  
 </**RelativeLayout**>  
  
 <**TextView  
 android:id="@+id/english\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:gravity="center\_horizontal"  
 android:text="[perent]"  
 android:textColor="@android:color/blue"  
 android:textSize="20dp"** />  
  
 <**RadioGroup  
 android:id="@+id/choose\_group"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:layout\_marginTop="40dp"  
 android:orientation="vertical"**>  
  
 <**RadioButton  
 android:id="@+id/choose\_btn\_one"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="A:兄弟"  
 android:button="@null"  
 android:textColor="@android:color/blue"  
 android:textSize="20sp"** />  
  
 <**RadioButton  
 android:id="@+id/choose\_btn\_two"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="B:姐妹"  
 android:button="@null"  
 android:textColor="@android:color/blue"  
 android:textSize="20sp"** />  
  
 <**RadioButton  
 android:id="@+id/choose\_btn\_three"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="C:父母"  
 android:button="@null"  
 android:textColor="@android:color/blue"  
 android:textSize="20sp"** />  
 </**RadioGroup**>  
</**LinearLayout**>

之后在MainActivity.java中声明这些控件

**private** TextView **timeText**, **dateText**, **wordText**, **englishText**;*//用来显示单词和音标的***private** ImageView **playVioce**; *//播放单词声音***private** String **mMonth**, **mDay**, **mWay**, **mHours**, **mMinute**; *// 用来显示时间***private** RadioGroup **radioGroup**; *//用于加载单词的三个选项  
//锁屏***private** KeyguardManager **km**;  
**private** KeyguardManager.KeyguardLock **kl**;  
**private** RadioButton **radioOne**, **radioTwo**, **radioThree**; *// 单词意思的三个选项***private** SpeechSynthesizer **speechSynthesizer**; *//合成对象 用于加载播放单词的朗读***private** SharedPreferences **sharedPreferences**; *//定义轻量级数据库*SharedPreferences.Editor **editor** = **null**; *//编辑数据库*

**第三节同步系统时间**

MainActivity中onstar方法

@Override  
**protected void** onStart() {  
 **super**.onStart();  
 */\*\*  
 \* 获取系统日期 并设置显示出来  
 \* \*/* Calendar calendar = Calendar.*getInstance*();  
 **mMonth** = String.*valueOf*(calendar.get(Calendar.***MONTH***) + 1); *//获取日期的月* **mDay** = String.*valueOf*(calendar.get(Calendar.***DAY\_OF\_MONTH***)); *//获取日期的天* **mWay** = String.*valueOf*(calendar.get(Calendar.***DAY\_OF\_WEEK***)); *//获取日期的星期  
  
 /\*\*  
 \* 如果小时是个位数  
 \*  
 \*则在前面价格“0”  
 \* \*/* **if** (calendar.get(Calendar.***HOUR***) < 10) {  
 **mHours** = **"0"** + calendar.get(Calendar.***HOUR***);  
 } **else** {  
 **mHours** = String.*valueOf*(calendar.get(Calendar.***HOUR***));  
 }  
 */\*\*  
 \* 如果分钟是个位数  
 \*  
 \*则在前面价格“0”  
 \* \*/* **if** (calendar.get(Calendar.***MINUTE***) < 10) {  
 **mMinute** = **"0"** + calendar.get(Calendar.***MINUTE***);  
 } **else** {  
 **mMinute** = String.*valueOf*(calendar.get(Calendar.***MINUTE***));  
 }  
  
 */\*\*  
 \* 获取星期  
 \* 并设置出来  
 \* \*/* **if** (**"1"**.equals(**mWay**)) {  
 **mWay** = **"天"**;  
 } **else if** (**"2"**.equals(**mWay**)) {  
 **mWay** = **"一"**;  
 } **else if** (**"3"**.equals(**mWay**)) {  
 **mWay** = **"二"**;  
 } **else if** (**"4"**.equals(**mWay**)) {  
 **mWay** = **"三"**;  
 } **else if** (**"5"**.equals(**mWay**)) {  
 **mWay** = **"四"**;  
 } **else if** (**"6"**.equals(**mWay**)) {  
 **mWay** = **"五"**;  
 } **else if** (**"7"**.equals(**mWay**)) {  
 **mWay** = **"六"**;  
 }  
 **timeText**.setText(**mHours** + **":"** + **mMinute**);  
 **dateText**.setText(**mMonth** + **"月"** + **mDay** + **"日"** + **" "** + **"星期"** + **mWay**);  
 BaseApplication.*addDestroyActiivty*(**this**,**"mainActivity"**);  
  
}

**第四节数据库文件获取**

**5.4.1定义轻量级数据库**

**private** SharedPreferences **sharedPreferences**; *//定义轻量级数据库*SharedPreferences.Editor **editor** = **null**; *//编辑数据库***int j** = 0; *//用于记录答了几道题*List<Integer> **list**; *//判断题的数目*List<CET4Entity> **datas**; *//用于从数据库独处相应的词库***int k**;  
  
**private** SQLiteDatabase **db**; *// 创建数据库***private** DaoMaster **mDaoMaster**, **dbMaster**; *// 管理者***private** DaoSession **mDaoSession**, **dbSession**; *// 和数据库进行会话***private** CET4EntityDao **questionDao**, **dbDao**;*// 对应的表,由java代码生成的,对数据库内相应的表操作使用此对象*

**5.4.2初始化控件**

*/\*\*  
 \* 初始化控件  
 \*/***public void** init() {  
 *//初始化轻量级数据库* **sharedPreferences** = getSharedPreferences(**"share"**, Context.***MODE\_PRIVATE***); *//初始化轻量级数据库* **editor** = **sharedPreferences**.edit(); *//初始化轻量级数据库编辑器  
 //appid换成自己申请的，播放语音* SpeechUser.*getUser*().login(MainActivity.**this**, **null**, **null**, **"appid=573a7bf0"**, **listener**);  
 **list** = **new** ArrayList<Integer>(); *//初始化list  
 /\*\*  
 \* 添加一个10个10以内随机数  
 \* \*/* Random r = **new** Random();  
 **int** i;  
 **while** (**list**.size() < 10) {  
 i = r.nextInt(20);  
 **if** (!**list**.contains(i)) {  
 **list**.add(i);  
 }  
 }  
 */\*\*  
 \* 得到键盘锁管理对象  
 \* \*/* **km** = (KeyguardManager) getSystemService(Context.***KEYGUARD\_SERVICE***);  
 **kl** = **km**.newKeyguardLock(**"unLock"**);  
 *// 初始化，只需要调用一次* AssetsDatabaseManager.*initManager*(**this**);  
 *// 获取管理对象，因为数据库需要通过管理对象才能够获取* AssetsDatabaseManager mg = AssetsDatabaseManager.*getManager*();  
 *// 通过管理对象获取数据库* SQLiteDatabase db1 = mg.getDatabase(**"word.db"**);  
 *// 对数据库进行操作* **mDaoMaster** = **new** DaoMaster(db1);  
 **mDaoSession** = **mDaoMaster**.newSession();  
 **questionDao** = **mDaoSession**.getCET4EntityDao();  
 *// 此DevOpenHelper类继承自SQLiteOpenHelper,第一个参数Context,第二个参数数据库名字,第三个参数CursorFactory* DaoMaster.DevOpenHelper helper = **new** DaoMaster.DevOpenHelper(**this**, **"wrong.db"**, **null**);  
 */\*\*  
 \* 初始化数据库  
 \* \*/* **db** = helper.getWritableDatabase();  
 **dbMaster** = **new** DaoMaster(**db**);  
 **dbSession** = **dbMaster**.newSession();  
 **dbDao** = **dbSession**.getCET4EntityDao();  
  
  
 **timeText** = (TextView) findViewById(R.id.***time\_text***); *//用于显示分钟绑定id* **dateText** = (TextView) findViewById(R.id.***date\_text***); *//用于显示日期绑定id* **wordText** = (TextView) findViewById(R.id.***word\_text***); *//用于显示单词绑定id* **englishText** = (TextView) findViewById(R.id.***english\_text***); *//用于显示音标绑定id* **playVioce** = (ImageView) findViewById(R.id.***play\_voice***); *//用于播放单词的按钮绑定id* **playVioce**.setOnClickListener(**this**); *//给播放单词按钮进行监听* **radioGroup** = (RadioGroup) findViewById(R.id.***choose\_group***); *//给加载单词三个选项绑定id* **radioOne** = (RadioButton) findViewById(R.id.***choose\_btn\_one***); *//给第一个选项绑定id* **radioTwo** = (RadioButton) findViewById(R.id.***choose\_btn\_two***); *//给第二个选项绑定id* **radioThree** = (RadioButton) findViewById(R.id.***choose\_btn\_three***); *//给第三个选项绑定id* **radioGroup**.setOnCheckedChangeListener(**this**); *//给加载单词三个选项设置监听事件* setParam();  
 SpeechUser.*getUser*().login(MainActivity.**this**, **null**, **null**, **"appid=573a7bf0"**, **listener**);  
 getDBData();  
}

**5.4.3回答错误保存到数据库**

**private void** saveWrongData() {  
 String word = **datas**.get(**k**).getWord(); *//获取答错这道题的单词* String english = **datas**.get(**k**).getEnglish(); *//获取答错这道题的音标* String china = **datas**.get(**k**).getChina(); *//获取答错这道题的汉语意思* String sign = **datas**.get(**k**).getSign(); *//获取答错这道题的标记* CET4Entity data = **new** CET4Entity(Long.*valueOf*(**dbDao**.count()),  
 word, english, china, sign);  
 **dbDao**.insertOrReplace(data); *//把这些字段存到数据库*}

**5.4.4获取数据**

*/\*\*  
 \* 设置选项  
 \*/***private void** setChina(List<CET4Entity> datas, **int** j) {  
 */\*\*  
 \* 随机产生几个随机数， 这里面产生几个随机数 是用于解锁单词  
 \* 因为此demo输入数据库里面20个单词， 所以产生的随机数是20以内的  
 \* \*/* Random r = **new** Random();  
 List<Integer> listInt = **new** ArrayList<>();  
 **int** i;  
 **while** (listInt.size() < 4) {  
 i = r.nextInt(20);  
 **if** (!listInt.contains(i)) {  
 listInt.add(i);  
 }  
 }  
 */\*\*  
 \* 以下的判断是给这个单词设置三个选项，设置单词选项是有规律的  
 \*三个选项，分别是正确的、正确的前一个、正确的的后一个  
 \*将这三个解释设置到单词的选项上，以下为实现逻辑  
 \* \*/* **if** (listInt.get(0) < 7) {  
 **radioOne**.setText(**"A: "** + datas.get(**k**).getChina());  
 **if** (**k** - 1 >= 0) {  
 **radioTwo**.setText(**"B: "** + datas.get(**k** - 1).getChina());  
 } **else** {  
 **radioTwo**.setText(**"B: "** + datas.get(**k** + 2).getChina());  
 }  
 **if** (**k** + 1 < 20) {  
 **radioThree**.setText(**"C: "** + datas.get(**k** + 1).getChina());  
 } **else** {  
 **radioThree**.setText(**"C: "** + datas.get(**k** - 1).getChina());  
 }  
 } **else if** (listInt.get(0) < 14) {  
 **radioTwo**.setText(**"B: "** + datas.get(**k**).getChina());  
 **if** (**k** - 1 >= 0) {  
 **radioOne**.setText(**"A: "** + datas.get(**k** - 1).getChina());  
 } **else** {  
 **radioOne**.setText(**"A: "** + datas.get(**k** + 2).getChina());  
 }  
 **if** (**k** + 1 < 20) {  
 **radioThree**.setText(**"C: "** + datas.get(**k** + 1).getChina());  
 } **else** {  
 **radioThree**.setText(**"C: "** + datas.get(**k** - 1).getChina());  
 }  
 } **else** {  
 **radioThree**.setText(**"C: "** + datas.get(**k**).getChina());  
 **if** (**k** - 1 >= 0) {  
 **radioTwo**.setText(**"B: "** + datas.get(**k** - 1).getChina());  
 } **else** {  
 **radioTwo**.setText(**"B: "** + datas.get(**k** + 2).getChina());  
 }  
 **if** (**k** + 1 < 20) {  
 **radioOne**.setText(**"A: "** + datas.get(**k** + 1).getChina());  
 } **else** {  
 **radioOne**.setText(**"A: "** + datas.get(**k** - 1).getChina());  
 }  
 }  
  
}  
  
*/\*\*  
 \* 获取数据  
 \*/***private void** getDBData() {  
 **datas** = **questionDao**.queryBuilder().list(); *//把词库里面的单词读出来* **k** = **list**.get(**j**);  
 **wordText**.setText(**datas**.get(**k**).getWord()); *//设置单词* **englishText**.setText(**datas**.get(**k**).getEnglish()); *//设置音标* setChina(**datas**, **k**); *//设置单词的三个选项*}  
  
*//滑取下一道题***private void** getNextData() {  
 **j**++; *//当前已做题数量* **int** i = **sharedPreferences**.getInt(**"allNum"**, 2); *//默认解锁题目为两道* **if** (i > **j**) { *//判断设定的解锁题目与当前所做题数量的大小关系* getDBData(); *//获取数据* setTextColor(); *//设置颜色  
 //已经学到的单词数量加一* **int** num = **sharedPreferences**.getInt(**"alreadyStudy"**, 0) + 1;  
 **editor**.putInt(**"alreadyStudy"**, num);  
 **editor**.commit(); *//存到数据库里* } **else** {  
 unlocked(); *//解锁* }  
}

**5.4.5音频获取**

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 */\*\*  
 \* 这句话是把此页面显示到手机屏幕的最上层  
 \* \*/* getWindow().addFlags(WindowManager.LayoutParams.***FLAG\_SHOW\_WHEN\_LOCKED*** | WindowManager.LayoutParams.***FLAG\_TURN\_SCREEN\_ON***);  
 getWindow().addFlags(WindowManager.LayoutParams.***FLAG\_FULLSCREEN***);  
 setContentView(R.layout.***activity\_main***);  
 init();*//初始化控件* setParam(); *//初始化语音播报*}

*/\*\*  
 \* 点击事件  
 \*/* @Override  
 **public void** onClick(View v) {  
 *//* ***TODO Auto-generated method stub* switch** (v.getId()) {  
 **case** R.id.***play\_voice***:*//播放单词声音* String text = **wordText**.getText().toString(); *//把单词提取出来* **speechSynthesizer**.startSpeaking(text, **this**); *//讯飞 播放声音* **break**;  
 }  
 }  
  
 @Override  
 **public void** onSpeakBegin() {  
  
 }  
  
 @Override  
 **public void** onBufferProgress(**int** i, **int** i1, **int** i2, String s) {  
  
 }  
  
 @Override  
 **public void** onSpeakPaused() {  
  
 }  
  
 @Override  
 **public void** onSpeakResumed() {  
  
 }  
  
 @Override  
 **public void** onSpeakProgress(**int** i, **int** i1, **int** i2) {  
  
 }  
  
 @Override  
 **public void** onCompleted(SpeechError speechError) {  
  
 }  
  
  
 */\*\*  
 \* 通用回调接口  
 \*/* **private** SpeechListener **listener** = **new** SpeechListener() {  
  
 *//消息回调* @Override  
 **public void** onEvent(**int** arg0, Bundle arg1) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//数据回调* @Override  
 **public void** onData(**byte**[] arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//结束回调（没有错误）* @Override  
 **public void** onCompleted(SpeechError arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 };  
  
 */\*\*  
 \* 初始化语音播报  
 \*/***public void** setParam() {  
 **speechSynthesizer** = SpeechSynthesizer.*createSynthesizer*(**this**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***VOICE\_NAME***, **"xiaoyan"**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***SPEED***, **"50"**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***VOLUME***, **"50"**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***PITCH***, **"50"**);  
 }

**5.4.6选项卡**

*/\*\*  
 \* 选项的点击事件  
 \*/*@Override  
**public void** onCheckedChanged(RadioGroup group, **int** checkedId) {  
 **radioGroup**.setClickable(**false**); *//默认选项未被选中* **switch** (checkedId) { *//选项的点击事件* **case** R.id.***choose\_btn\_one***: *//点击“A”选项* String msg = **radioOne**.getText().toString().substring(3); *//截取字符串* btnGetText(msg, **radioOne**); *//将参数传入到对应的方法里* **break**;  
 **case** R.id.***choose\_btn\_two***: *//点击“B”选项* String msg1 = **radioTwo**.getText().toString().substring(3); *//截取字符串* btnGetText(msg1, **radioTwo**); *//将参数传入到对应的方法里* **break**;  
 **case** R.id.***choose\_btn\_three***: *//点击“C”选项* String msg2 = **radioThree**.getText().toString().substring(3); *//截取字符串* btnGetText(msg2, **radioThree**); *//将参数传入到对应的方法里* **break**;  
 }  
}  
  
*/\*\*  
 \* 还原单词与选项的颜色  
 \*/***private void** setTextColor() {  
 *//还原单词选项的颜色* **radioOne**.setChecked(**false**); *//默认不被点击* **radioTwo**.setChecked(**false**); *//默认不被点击* **radioThree**.setChecked(**false**); *//默认不被点击* **radioOne**.setTextColor(Color.*parseColor*(**"#FFFFFF"**)); *//将选项的按钮设置为白色* **radioTwo**.setTextColor(Color.*parseColor*(**"#FFFFFF"**)); *//将选项的按钮设置为白色* **radioThree**.setTextColor(Color.*parseColor*(**"#FFFFFF"**)); *//将选项的按钮设置为白色* **wordText**.setTextColor(Color.*parseColor*(**"#FFFFFF"**)); *//将单词设置为白色* **englishText**.setTextColor(Color.*parseColor*(**"#FFFFFF"**)); *//将音标设置为白色*}  
  
*/\*\*  
 \* 解锁  
 \*/***private void** unlocked() {  
 Intent intent1 = **new** Intent(Intent.***ACTION\_MAIN***); *//界面跳转* intent1.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 intent1.addCategory(Intent.***CATEGORY\_HOME***); *//进入到手机桌面* startActivity(intent1); *//启动* **kl**.disableKeyguard(); *//解锁* finish(); *//销毁当前activity*}

**第五节手势时间处理**

**5.5.1定义变量**

*/\*\*  
 \* 手指按下的点为（x1,y1）  
 \* 手指离开屏幕的点为（x2,y2）  
 \*/***float x1** = 0;  
**float y1** = 0;  
**float x2** = 0;  
**float y2** = 0;

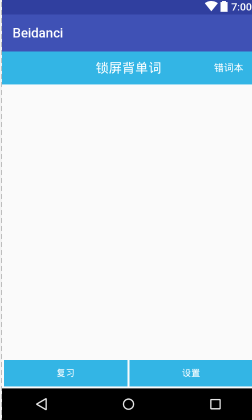
**5.5.2手势处理**

*/\*\*  
 \* 复写activity的onTouch方法  
 \* 监听滑动事件  
 \*/*@Override  
**public boolean** onTouchEvent(MotionEvent event) {  
 **if** (event.getAction() == MotionEvent.***ACTION\_DOWN***) {  
 *//当手指按下的时候* **x1** = event.getX();  
 **y1** = event.getY();  
  
 }  
 **if** (event.getAction() == MotionEvent.***ACTION\_MOVE***){  
 **x2** = event.getX();  
 **y2** = event.getY();  
 *//当手指离开的时候\* **if** (Math.*abs*(**x2** - **x1**) <10){  
  
 }  
 }  
 **if** (event.getAction() == MotionEvent.***ACTION\_UP***) {  
  
  
 **x2** = event.getX();  
 **y2** = event.getY();  
 **if** (**y1** - **y2** > 200) {*//向上滑  
 /\*\*  
 \* 已掌握单词数量加一  
 \* \*/* **radioOne**.setClickable(**false**);  
 **radioTwo**.setClickable(**false**);  
 **radioThree**.setClickable(**false**);  
 **int** num = **sharedPreferences**.getInt(**"alreadyMastered"**, 0) + 1;  
 **editor**.putInt(**"alreadyMastered"**, num);  
 **editor**.commit();  
 Toast.*makeText*(**this**, **"已掌握"**, Toast.***LENGTH\_SHORT***).show();  
 getNextData();*//获取下一条数据* } **else if** (**y2** - **y1** > 800) {*//向下滑* **radioOne**.setClickable(**false**);  
 **radioTwo**.setClickable(**false**);  
 **radioThree**.setClickable(**false**);  
 *// Toast.makeText(this, "待加功能......", Toast.LENGTH\_SHORT).show();* } **else if** (**x1** - **x2** > 200) {*//向左滑* **radioOne**.setClickable(**false**);  
 **radioTwo**.setClickable(**false**);  
 **radioThree**.setClickable(**false**);  
 getNextData();*//获取下一条数据* } **else if** (**x2** - **x1** > 200) {*//向右滑* **radioOne**.setClickable(**false**);  
 **radioTwo**.setClickable(**false**);  
 **radioThree**.setClickable(**false**);  
 finish();*//解锁* }**else** {  
 **radioOne**.setClickable(**true**);  
 **radioTwo**.setClickable(**true**);  
 **radioThree**.setClickable(**true**);  
 }  
  
 }  
 **return super**.onTouchEvent(event);  
}

**第六章复习界面设计**

**第一节复习界面布局设计**

**6.1.1页面预览**

****

**6.1.2布局文件home\_layout**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:background="@android:color/holo\_blue\_light"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:text="锁屏背单词"  
 android:textColor="@android:color/white"  
 android:textSize="20sp"** />  
  
 <**Button  
 android:id="@+id/wrong\_btn"  
 android:layout\_width="60dp"  
 android:layout\_height="30dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerVertical="true"  
 android:layout\_marginRight="10dp"  
 android:padding="3dp"  
 android:text="错词本"  
 android:textColor="@android:color/white"  
 android:textSize="15sp"  
 android:background="@drawable/btn\_kuang"**/>  
 </**RelativeLayout**>  
  
 <**FrameLayout  
 android:id="@+id/frame\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginBottom="40dp"  
 android:layout\_marginTop="50dp"**></**FrameLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginBottom="3dp"  
 android:orientation="horizontal"**>  
  
 <**Button  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="3dp"  
 android:layout\_weight="1"  
 android:background="@android:color/holo\_blue\_light"  
 android:onClick="study"  
 android:text="复习"  
 android:textColor="#FFFFFF"** />  
  
 <**Button  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="3dp"  
 android:layout\_marginRight="3dp"  
 android:layout\_weight="1"  
 android:background="@android:color/holo\_blue\_light"  
 android:onClick="set"  
 android:text="设置"  
 android:textColor="#FFFFFF"** />  
 </**LinearLayout**>  
  
</**RelativeLayout**>

**第二节复习界面功能设计**

**6.2.1实现跳转HomeActivity.java**  
*/\*\*  
 \* Created by 86185 on 2020/5/7.  
 \*/***public class** HomeActivity **extends** AppCompatActivity **implements** View.OnClickListener {  
 **private** ScreenListener **screenListener**; *//手机屏幕状态的监听* **private** SharedPreferences **sharedPreferences**; *//定义一个轻量级数据库* **private** FragmentTransaction **transaction**; *//用于加载设置界面和复习界面的fragment* **private** StudyFragment **studyFragment**; *//定义复习界面的fragment* **private** SetFrament **setFragment**; *//定义设置界面的fragment* **private** Button **wrongBtn**; *//用于跳转到错题界面* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 getWindow().addFlags(WindowManager.LayoutParams.***FLAG\_FULLSCREEN***);  
 setContentView(R.layout.***home\_layout***); *//绑定布局文件* **sharedPreferences** = getSharedPreferences(**"share"**, Context.***MODE\_PRIVATE***); *//初始化数据库* **wrongBtn** = (Button) findViewById(R.id.***wrong\_btn***); *//绑定id* **wrongBtn**.setOnClickListener(**this**); *//对按钮设置监听事件* **final** SharedPreferences.Editor editor = **sharedPreferences**.edit(); *//设置editer用于网数据库里面添加数据和修改数据* **screenListener** = **new** ScreenListener(**this**); *//屏幕状态进行监听* **screenListener**.begin(**new** ScreenListener.ScreenStateListener() {  
 @Override  
 **public void** onScreenOn() { *//手机已点亮屏幕的操作* **if** (**sharedPreferences**.getBoolean(**"btnTf"**,**false**)){ *//判断是否在设置界面开启了锁屏按钮* **if** (**sharedPreferences**.getBoolean(**"tf"**, **false**)) { *//判断屏幕是否解锁* Intent intent = **new** Intent(HomeActivity.**this**, MainActivity.**class**); *//启动锁屏页面* startActivity(intent); *//开始跳转* }  
 }  
 }  
  
 @Override  
 **public void** onScreenOff() { *//手机已锁屏的操作  
 /\*\*  
 \* 如果手机已经锁了  
 \* 就把数据库面的tf字段改成true  
 \* \*/* editor.putBoolean(**"tf"**, **true**);  
 editor.commit();  
 BaseApplication.*destroyActivity*(**"lockActivity"**); *//销毁锁屏页面的额activity* }  
  
 @Override  
 **public void** onUserPresent() { *//手机已解锁的操作  
 /\*\*  
 \* 如果手机已经解锁了  
 \* 就把数据库面的tf字段改成false  
 \* \*/* editor.putBoolean(**"tf"**, **false**);  
 editor.commit();  
 }  
 });  
 **studyFragment** = **new** StudyFragment(); *//当此页面加载的时候 显示复习界面的fragment* setFragment(**studyFragment**);*//设置不同的fragment* }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
  
 }  
  
 */\*\*  
 \* 点击不同的按钮显示不同的fragment  
 \* \*/* **public void** setFragment(Fragment fragment) {  
 **transaction** = getFragmentManager().beginTransaction();*//初始化transaction* **transaction**.replace(R.id.***frame\_layout***, fragment); *//绑定id* **transaction**.commit();  
 }  
  
 *//点击进入复习页面* **public void** study(View v) {  
 **if** (**studyFragment** == **null**) {  
 **studyFragment** = **new** StudyFragment();  
 }  
 setFragment(**studyFragment**);  
 }  
  
 *//点击进入设置界面* **public void** set(View v) {  
 **if** (**setFragment** == **null**) {  
 **setFragment** = **new** SetFrament();  
 }  
 setFragment(**setFragment**);  
 }  
  
  
 @Override  
 **protected void** onStop() {  
 **super**.onStop();  
 }  
  
 */\*\*  
 \* 解除广播监听  
 \*/* @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 **screenListener**.unregisterListener();  
 }  
  
 */\*\*  
 \* 点击事件  
 \* \*/* @Override  
 **public void** onClick(View v) {  
  
 **switch** (v.getId()){  
 */\*\*  
 \* 跳转到错题页面  
 \* \*/* **case** R.id.***wrong\_btn***:  
 Intent i = **new** Intent(**this**,WrongActivity.**class**); *//启动WrongActivity* startActivity(i);  
 **break**;  
 }  
 }  
}

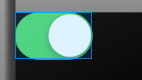
**6.2.2显示内容StudyFragment**

**public class** StudyFragment **extends** Fragment {  
 **private** TextView **difficultyTv**, *//学习的难度* **wisdomEnglish**, *//名人名句的英语意思* **wisdomChina**, *//名人名句的汉语意思* **alreadyStudyText**, *//已经学习题数* **alreadyMasteredText**, *//已经掌握题数* **wrongText**; *//答错题数* **private** SharedPreferences **sharedPreferences**; *//定义轻量级数据库  
 //数据库  
// private SQLiteDatabase db;* **private** DaoMaster **mDaoMaster**; *// 数据库管理者* **private** DaoSession **mDaoSession**; *// 与数据库进行会话* **private** WisdomEntityDao **questionDao**; *// 对应的表,由java代码生成的,对数据库内相应的表操作使用此对象* @Nullable  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***study\_fragment\_layout***, **null**); *//绑定布局文件* **sharedPreferences** = getActivity().getSharedPreferences(**"share"**, Context.***MODE\_PRIVATE***); *//初始化数据库* **difficultyTv** = (TextView) view.findViewById(R.id.***difficulty\_text***); *//学习难度绑定id* **wisdomEnglish** = (TextView) view.findViewById(R.id.***wisdom\_english***); *//名人名句的英语绑定id* **wisdomChina** = (TextView) view.findViewById(R.id.***wisdom\_china***); *//名人名句的汉语绑定id* **alreadyMasteredText** = (TextView) view.findViewById(R.id.***already\_mastered***); *//已经掌握题数绑定id* **alreadyStudyText** = (TextView) view.findViewById(R.id.***already\_study***); *//已经学习题数绑定id* **wrongText** = (TextView) view.findViewById(R.id.***wrong\_text***); *//倒错错数绑定id  
 // 初始化，只需要调用一次* AssetsDatabaseManager.*initManager*(getActivity());  
 *// 获取管理对象，因为数据库需要通过管理对象才能够获取* AssetsDatabaseManager mg = AssetsDatabaseManager.*getManager*();  
 *// 通过管理对象获取数据库* SQLiteDatabase db1 = mg.getDatabase(**"wisdom.db"**);  
 *// 对数据库进行操作* **mDaoMaster** = **new** DaoMaster(db1); *//初始化管理者* **mDaoSession** = **mDaoMaster**.newSession(); *//初始化会话对象* **questionDao** = **mDaoSession**.getWisdomEntityDao(); *//获取数据* **return** view;  
 }  
  
 @Override  
 **public void** onStart() {  
 **super**.onStart();  
 **difficultyTv**.setText(**sharedPreferences**.getString(**"difficulty"**, **"四级"**) + **"英语"**); *//默认设置难度为四级* List<WisdomEntity> datas = **questionDao**.queryBuilder().list(); *//货物数据 集合* Random random = **new** Random(); *//产生随机数* **int** i = random.nextInt(10); *//随机10以内的一个随机数* **wisdomEnglish**.setText(datas.get(i).getEnglish()); *//从数据库里面获取到这条数据的的英语* **wisdomChina**.setText(datas.get(i).getChina()); *//从数据库里面获取到这条数据的汉语* setText(); *//设置文字* }  
  
 **private void** setText() {  
 **alreadyMasteredText**.setText(**sharedPreferences**.getInt(**"alreadyMastered"**,0)+**""**); *//设置已经复习的题数（数据库读取）* **alreadyStudyText**.setText(**sharedPreferences**.getInt(**"alreadyStudy"**,0)+**""**); *//设置已经学习的题数（数据库读取）* **wrongText**.setText(**sharedPreferences**.getInt(**"wrong"**,0)+**""**); *//设置错题题数（数据库读取）* }  
  
}

**第七章设置界面设计**

**第一节开关按钮设计**

**7.1.1开关按钮预览**



**7.1.2布局文件switch\_button**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**>  
  
 <**ImageView  
 android:id="@+id/iv\_switch\_open"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@mipmap/ease\_open\_icon"  
 android:visibility="visible"** />  
  
 <**ImageView  
 android:id="@+id/iv\_switch\_close"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@mipmap/ease\_close\_icon"  
 android:visibility="invisible"** />  
  
</**FrameLayout**>

**第二节开关按钮功能**

**7.2.1SwitchButton.java**

*/\*\*  
 \* Created by 86185 on 2020/5/7.  
 \*/  
/\*\*  
 \* <p/>  
 \* 开关 自定义button  
 \*/***public class** SwitchButton **extends** FrameLayout {  
 **private** ImageView **openImage**; *//打开按钮的图片图片* **private** ImageView **closeImage**; *//关闭按钮的图片  
  
 /\*\*  
 \* 联系上下文  
 \* \*/* **public** SwitchButton(Context context) {  
 **this**(context, **null**);  
 }  
  
 */\*\*  
 \* 构造方法  
 \* \*/* **public** SwitchButton(Context context, AttributeSet attrs, **int** defStyleAttr) {  
 **this**(context, attrs);  
 }  
  
 **public** SwitchButton(Context context, AttributeSet attrs) {  
 **super**(context, attrs);  
 *//context通过调用obtainStyledAttributes方法来获取一个TypeArray，然后由该TypeArray来对属性进行设置* TypedArray typedArray = context.obtainStyledAttributes(attrs, R.styleable.***SwitchButton***);  
  
 *//画出开关为打开的状态* Drawable openDrawable = typedArray.getDrawable(R.styleable.***SwitchButton\_switchOpenImage***);  
  
 *//画出开关为关闭的状态* Drawable closeDrawable = typedArray.getDrawable(R.styleable.***SwitchButton\_switchCloseImage***);  
 **int** switchStatus = typedArray.getInt(R.styleable.***SwitchButton\_switchStatus***, 0);  
 *//调用结束后务必调用recycle()方法，否则这次的设定会对下次的使用造成影响* typedArray.recycle();  
  
 LayoutInflater.*from*(context).inflate(R.layout.***switch\_button***, **this**);*//绑定布局文件* **openImage** = (ImageView) findViewById(R.id.***iv\_switch\_open***); *//绑定id* **closeImage** = (ImageView) findViewById(R.id.***iv\_switch\_close***); *//绑定id* **if** (openDrawable != **null**) { *//如果是打开状态* **openImage**.setImageDrawable(openDrawable); *//设置显示图片* }  
 **if** (closeDrawable != **null**) { *//如果是关闭状态* **closeImage**.setImageDrawable(closeDrawable); *//设置关闭图片* }  
 **if** (switchStatus == 1) { *//判断开关的状态* closeSwitch(); *//执行关闭的方法* }  
  
 }  
  
 */\*\*  
 \* 判断开关的状态  
 \*/* **public boolean** isSwitchOpen() {  
 **return openImage**.getVisibility() == View.***VISIBLE***;  
 }  
  
 */\*\*  
 \* 打开开关  
 \*/* **public void** openSwitch() {  
 **openImage**.setVisibility(View.***VISIBLE***); *//显示打开图片* **closeImage**.setVisibility(View.***INVISIBLE***); *//隐藏关闭图片* }  
  
 */\*\*  
 \* 关闭开关  
 \*/* **public void** closeSwitch() {  
 **openImage**.setVisibility(View.***INVISIBLE***); *//隐藏打开图片* **closeImage**.setVisibility(View.***VISIBLE***); *//显示关闭图片* }  
}

**第三节设置界面布局**

**7.3.1设置界面预览**

****

**7.3.2布局文件set\_frsgment\_layout**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="5dp"  
 android:background="@android:color/white"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:padding="10dp"  
 android:text="开启单词锁屏"  
 android:textSize="17sp"** />  
  
 <**com.li.sockwork.SwitchButton  
 android:id="@+id/switch\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="28dp"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_marginRight="15dp"**>  
  
 </**com.li.sockwork.SwitchButton**>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="5dp"  
 android:background="@android:color/white"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="8"  
 android:padding="10dp"  
 android:text="开锁单词难度"  
 android:textSize="17sp"** />  
  
 <**Spinner  
 android:id="@+id/spinner\_difficulty"  
 android:layout\_width="0dp"  
 android:layout\_height="28dp"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_marginRight="15dp"  
 android:layout\_weight="2"  
 android:background="@null"**>  
  
 </**Spinner**>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="5dp"  
 android:background="@android:color/white"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="8"  
 android:padding="10dp"  
 android:text="需要解锁题的个数"  
 android:textSize="17sp"** />  
  
 <**Spinner  
 android:id="@+id/spinner\_all\_number"  
 android:layout\_width="0dp"  
 android:layout\_height="28dp"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_marginRight="15dp"  
 android:layout\_weight="2"  
 android:background="@null"**>  
  
 </**Spinner**>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="5dp"  
 android:background="@android:color/white"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="8"  
 android:padding="10dp"  
 android:text="每日新题的个数"  
 android:textSize="17sp"** />  
  
 <**Spinner  
 android:id="@+id/spinner\_new\_number"  
 android:layout\_width="0dp"  
 android:layout\_height="28dp"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_marginRight="15dp"  
 android:layout\_weight="2"  
 android:background="@null"**>  
  
 </**Spinner**>  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="5dp"  
 android:background="@android:color/white"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="8"  
 android:padding="10dp"  
 android:text="每日复习题的个数"  
 android:textSize="17sp"** />  
  
 <**Spinner  
 android:id="@+id/spinner\_revise\_number"  
 android:layout\_width="0dp"  
 android:layout\_height="28dp"  
 android:layout\_gravity="center\_vertical"  
 android:layout\_marginRight="15dp"  
 android:layout\_weight="2"  
 android:background="@null"**>  
  
 </**Spinner**>  
 </**LinearLayout**>  
</**LinearLayout**>

**第四节设置功能**

**7.4.1SetFragment.java**

*/\*\*  
 \* Created by 86185 on 2020/5/7.  
 \*/***public class** SetFrament **extends** Fragment **implements** View.OnClickListener {  
 **private** SharedPreferences **sharedPreferences**; *//定义一个轻量级数据库* **private** SwitchButton **switchButton**; *//开关按钮* **private** Spinner **spinnerDifficulty**; *//定义选择难度的下拉框* **private** Spinner **spinnerAllNum**; *//定义解锁题目的下拉框* **private** Spinner **spinnerNewNum**; *//定义新题目的下拉框* **private** Spinner **spinnerReviewNum**; *//定义复习题的下拉框* **private** ArrayAdapter<String> **adapterDifficulty**, **adapterAllNum**, **adapterNewNum**, **adapterReviewNUm**; *//定义下拉框的适配器* String[] **difficulty** = **new** String[]{**"小学"**, **"初中"**, **"高中"**, **"四级"**, **"六级"**}; *//选择难度下拉框里面的选项nr* String[] **allNum** = **new** String[]{**"2道"**, **"4道"**, **"6道"**, **"8道"**}; *//解锁题目下拉框的选项内容* String[] **newNum** = **new** String[]{**"10"**, **"30"**, **"50"**, **"100"**}; *//新题目下拉框的选项内容* String[] **revicwNum** = **new** String[]{**"10"**, **"30"**, **"50"**, **"100"**}; *//复习题目下拉框的选项内容* SharedPreferences.Editor **editor** = **null**; *//定义数据库的编辑器* **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***set\_fragment\_layout***,**null**); *//绑定布局文件* init(view);  
 **return** view;  
 }  
  
 */\*\*  
 \* 初始化控件  
 \*/* **private void** init(View view) {  
 **sharedPreferences** = getActivity().getSharedPreferences(**"share"**, Context.***MODE\_PRIVATE***); *//初始化适配器* **editor** = **sharedPreferences**.edit(); *//初始化编辑器* **switchButton** = (SwitchButton) view.findViewById(R.id.***switch\_btn***); *//开关按钮绑定id* **switchButton**.setOnClickListener(**this**); *//开关按钮设置监听事件* **spinnerDifficulty** = (Spinner) view.findViewById(R.id.***spinner\_difficulty***); *//选择难度下拉框绑定id* **spinnerAllNum** = (Spinner) view.findViewById(R.id.***spinner\_all\_number***); *//解锁题目下拉框绑定id* **spinnerNewNum** = (Spinner) view.findViewById(R.id.***spinner\_new\_number***); *//新题目下拉框绑定id* **spinnerReviewNum** = (Spinner) view.findViewById(R.id.***spinner\_revise\_number***); *//复习题下拉框绑定id* **adapterDifficulty** = **new** ArrayAdapter<String>(getActivity(),  
 android.R.layout.***simple\_selectable\_list\_item***, **difficulty**); *//初始化选择难度下拉框的适配器* **spinnerDifficulty**.setAdapter(**adapterDifficulty**); *//给选择难度下拉框设置适配器* setSpinnerItemSelectedByValue(**spinnerDifficulty**, **sharedPreferences**.getString(**"difficulty"**, **"四级"**)); *//定义选择难度下拉框的默认选项* **this**.**spinnerDifficulty**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() { *//设置选择难度的下拉框的监听事件* @Override  
 **public void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  
 String msg = parent.getItemAtPosition(position).toString(); *//获取到选择的内容* **editor**.putString(**"difficulty"**, msg); *//写到数据库里面* **editor**.commit(); *//保存* }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
 */\*\*  
 \* 同上面的选择难度的选项框的原理一样  
 \* \*/* **adapterAllNum** = **new** ArrayAdapter<String>(getActivity(), android.R.layout.***simple\_selectable\_list\_item***, **allNum**);  
 **spinnerAllNum**.setAdapter(**adapterAllNum**);  
 setSpinnerItemSelectedByValue(**spinnerAllNum**, **sharedPreferences**.getInt(**"allNum"**, 2) + **"道"**);  
 **this**.**spinnerAllNum**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  
 String msg = parent.getItemAtPosition(position).toString();  
 **int** i = Integer.*parseInt*(msg.substring(0, 1));  
 **editor**.putInt(**"allNum"**, i);  
 **editor**.commit();  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
  
 */\*\*  
 \* 同上面的选择难度的选项框的原理一样  
 \* \*/* **adapterNewNum** = **new** ArrayAdapter<String>(getActivity(), android.R.layout.***simple\_selectable\_list\_item***, **newNum**);  
 **spinnerNewNum**.setAdapter(**adapterNewNum**);  
 setSpinnerItemSelectedByValue(**spinnerNewNum**, **sharedPreferences**.getString(**"newNum"**, **"10"**));  
 **this**.**spinnerNewNum**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  
 String msg = parent.getItemAtPosition(position).toString();  
 **editor**.putString(**"newNum"**, msg);  
 **editor**.commit();  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
  
 */\*\*  
 \* 同上面的选择难度的选项框的原理一样  
 \* \*/* **adapterReviewNUm** = **new** ArrayAdapter<String>(getActivity(), android.R.layout.***simple\_selectable\_list\_item***, **revicwNum**);  
 **spinnerReviewNum**.setAdapter(**adapterReviewNUm**);  
 setSpinnerItemSelectedByValue(**spinnerReviewNum**, **sharedPreferences**.getString(**"reviewNum"**, **"10"**));  
 **this**.**spinnerReviewNum**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> parent, View view, **int** position, **long** id) {  
 String msg = parent.getItemAtPosition(position).toString();  
 **editor**.putString(**"reviewNum"**, msg);  
 **editor**.commit();  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> parent) {  
  
 }  
 });  
 }  
  
  
 @Override  
 **public void** onStart() {  
 **super**.onStart();  
 */\*\*  
 \* 从数据库获取开关按钮的状态  
 \* \*/* **if** (**sharedPreferences**.getBoolean(**"btnTf"**, **false**)) {  
 **switchButton**.openSwitch();  
 } **else** {  
 **switchButton**.closeSwitch();  
 }  
 }  
  
 */\*\*  
 \* 点击的监听事件  
 \*/* @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***switch\_btn***: *//点击开关按钮* **if** (**switchButton**.isSwitchOpen()) { *//如果开关按钮为打开状态* **switchButton**.closeSwitch(); *//则关闭按钮* **editor**.putBoolean(**"btnTf"**, **false**); *//写入数据库开关按钮状态* } **else** { *//否则为关闭状态* **switchButton**.openSwitch(); *//打开开关按钮* **editor**.putBoolean(**"btnTf"**, **true**); *//写入数据库状态* }  
 **editor**.commit(); *//进行保存* **break**;  
 }  
 }  
  
 */\*\*  
 \* 设置下拉框默认选项的方法  
 \*/* **public void** setSpinnerItemSelectedByValue(Spinner spinner, String value) {  
 SpinnerAdapter apsAdapter = spinner.getAdapter(); *//得到SpinnerAdapter对象* **int** k = apsAdapter.getCount();  
 **for** (**int** i = 0; i < k; i++) {  
 **if** (value.equals(apsAdapter.getItem(i).toString())) {  
 spinner.setSelection(i, **true**);*// 默认选中项* }  
 }  
 }  
}

**第八章错词本界面设计**

**第一节错词本界面布局**

**8.1.1错词本界面预览**



**8.1.2布局文件wrong\_layout**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:background="@android:color/holo\_blue\_light"  
 android:orientation="horizontal"**>  
 <**ImageButton  
 android:id="@+id/back\_btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@mipmap/fanhui"  
 android:layout\_centerVertical="true"  
 android:layout\_marginLeft="15dp"**/>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 android:text="锁屏背单词"  
 android:textColor="@android:color/white"  
 android:textSize="20sp"** />  
 <**Button  
 android:id="@+id/i\_know\_btn"  
 android:layout\_width="60dp"  
 android:layout\_height="30dp"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerVertical="true"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/btn\_kuang"  
 android:padding="3dp"  
 android:text="我会了"  
 android:textColor="@android:color/white"  
 android:textSize="15sp"** />  
 </**RelativeLayout**>  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="85dp"  
 android:layout\_marginTop="80dp"** >  
  
 <**TextView  
 android:id="@+id/word\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:text="parent"  
 android:textColor="@android:color/holo\_blue\_light"  
 android:textSize="50dp"** />  
  
 <**ImageView  
 android:id="@+id/play\_vioce"  
 android:layout\_width="27dp"  
 android:layout\_height="27dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginRight="60dp"  
 android:background="@mipmap/vioce"** />  
 </**RelativeLayout**>  
  
 <**TextView  
 android:text="[perent]"  
 android:id="@+id/english\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="5dp"  
 android:gravity="center\_horizontal"  
 android:textColor="@android:color/holo\_blue\_light"  
 android:textSize="20dp"** />  
  
 <**TextView  
 android:id="@+id/china\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:gravity="center\_horizontal"  
 android:text="父母"  
 android:textColor="@android:color/holo\_blue\_light"  
 android:textSize="20dp"** />  
  
</**LinearLayout**>

**第二节错词本界面功能**

**8.2.1WrongActivity.java**

*/\*\*  
 \* Created by 86185 on 2020/5/7.  
 \*/***public class** WrongActivity **extends** AppCompatActivity **implements** View.OnClickListener,SynthesizerListener{  
 **private** TextView **chinaText**, **wordText**, **englishText**; *//用来显示单词和音标的* **private** SharedPreferences **sharedPreferences**; *//定义轻量级数据库* **private** SharedPreferences.Editor **editor**; *//数据库编辑器* **private** ImageView **playVioce**;  
 **private** ImageView **backBtn**; *//播放声音* **private** Button **iKnowBtn**; *//“我会了”按钮* **private** SpeechSynthesizer **speechSynthesizer**; *//合成对象* **private** SQLiteDatabase **db**; *//定义数据库* **private** DaoMaster **mDaoMaster**; *// 数据库管理者* **private** DaoSession **mDaoSession**; *// 与数据库进行会话  
 // 对应的表,由java代码生成的,对数据库内相应的表操作使用此对象* **private** CET4EntityDao **questionDao**;  
  
 */\*\*  
 \* 手指按下的点为（x1,y1）  
 \* 手指离开屏幕的点为（x2,y2）  
 \*/* **float x1** = 0;  
 **float y1** = 0;  
 **float x2** = 0;  
 **float y2** = 0;  
  
 List<CET4Entity> **datas**;  
 List<CET4Entity> **wrongData**; *//定义一个list泛型为CET4Entity* **int wrongNum** = 0; *//定义一个int型数据* **int**[] **wrongArry**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 getWindow().addFlags(WindowManager.LayoutParams.***FLAG\_FULLSCREEN***);  
 setContentView(R.layout.***wrong\_layout***); *//绑定布局文件* **sharedPreferences** = getSharedPreferences(**"share"**, Context.***MODE\_PRIVATE***); *//初始化数据库* **editor** = **sharedPreferences**.edit(); *//初始化编辑器* **chinaText** = (TextView) findViewById(R.id.***china\_text***); *//汉语绑定id* **englishText** = (TextView) findViewById(R.id.***english\_text***); *//音标绑定id* **wordText** = (TextView) findViewById(R.id.***word\_text***); *//单词绑定id* **playVioce** = (ImageView) findViewById(R.id.***play\_vioce***); *//播放声音按钮绑定id* **playVioce**.setOnClickListener(**this**); *// 播放声音设置监听事件* **iKnowBtn** = (Button) findViewById(R.id.***i\_know\_btn***); *//“我会了”按钮绑定id* **iKnowBtn**.setOnClickListener(**this**); *//“我会了”按钮设置监听事件* **backBtn** = (ImageView) findViewById(R.id.***back\_btn***); *//返回按钮绑定id* **backBtn**.setOnClickListener(**this**); *//返回按钮设置监听事件  
 // 通过管理对象获取数据库  
 // 对数据库进行操作  
 // 此DevOpenHelper类继承自SQLiteOpenHelper,第一个参数Context,第二个参数数据库名字,第三个参数CursorFactory* DaoMaster.DevOpenHelper helper = **new** DaoMaster.DevOpenHelper(**this**, **"wrong.db"**, **null**);  
 **db** = helper.getWritableDatabase();  
 **mDaoMaster** = **new** DaoMaster(**db**);  
 **mDaoSession** = **mDaoMaster**.newSession();  
 **questionDao** = **mDaoSession**.getCET4EntityDao();  
 setData(**wrongNum**); *//设置错题  
// setWrongData();* setParam(); *//初始化语音播报* }

*/\*\*  
 \* 初始化语音播报  
 \* \*/* **public void** setParam() {  
 **speechSynthesizer** = SpeechSynthesizer.*createSynthesizer*(**this**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***VOICE\_NAME***, **"xiaoyan"**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***SPEED***, **"50"**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***VOLUME***, **"50"**);  
 **speechSynthesizer**.setParameter(SpeechConstant.***PITCH***, **"50"**);  
 }  
  
 */\*\*  
 \* 设置错题  
 \* \*/* **private void** setData(**int** j) {  
 **iKnowBtn**.setVisibility(View.***VISIBLE***); *//“我会了”按钮显示出来* **wrongData** = **new** ArrayList<>(); *//初始化list* **if** (**questionDao**.queryBuilder().list() != **null** && **questionDao**.queryBuilder().list().size() > 0  
 && j < **questionDao**.queryBuilder().list().size()  
 && j >= 0) { *//判断如果数据库不为空* **for** (**int** i = 0; i < **questionDao**.queryBuilder().list().size(); i++) {  
 **wrongData**.add(i, **questionDao**.queryBuilder().list().get(i)); *//把数据循环加到list里面* }  
  
 */\*\*  
 \* 分别将list里面的数据取出第j条数据设置单词音标以及汉语  
 \* \*/* **wordText**.setText(**wrongData**.get(j).getWord());  
 **englishText**.setText(**wrongData**.get(j).getEnglish());  
 **chinaText**.setText(**wrongData**.get(j).getChina());  
  
 } **else** {  
 */\*\*  
 \* 如果数据库为空  
 \* 隐藏“我会了”按钮  
 \* \*/* **wordText**.setText(**"好厉害"**);  
 **englishText**.setText(**"[没有]"**);  
 **chinaText**.setText(**"不会的单词了"**);  
 **iKnowBtn**.setVisibility(View.***GONE***);  
 }  
  
 }  
  
 */\*\*  
 \* 复写activity的onTouch方法  
 \* 监听滑动事件  
 \*/* @Override  
 **public boolean** onTouchEvent(MotionEvent event) {  
 **if** (event.getAction() == MotionEvent.***ACTION\_DOWN***) {  
 *//当手指按下的时候* **x1** = event.getX();  
 **y1** = event.getY();  
 }  
 **if** (event.getAction() == MotionEvent.***ACTION\_UP***) {  
 *//当手指离开的时候* **x2** = event.getX();  
 **y2** = event.getY();  
 **if** (**x1** - **x2** > 200) { *//向左滑* **if** (**wrongNum** + 2 > **questionDao**.queryBuilder().list().size()) { *//判断如果数据库里面没有数据了* Toast.*makeText*(**this**, **"没有更多的数据了"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** { *//否则有数据进行数据设置* **wrongNum**++; *//定义的int行数据+1* setData(**wrongNum**); *//设置数据* }  
  
 } **else if** (**x2** - **x1** > 200) { *//向右滑* **if** (**wrongNum** - 1 < 0) { *//判断是不是第一条数据* Toast.*makeText*(**this**, **"没有更多的数据了"**, Toast.***LENGTH\_SHORT***).show();  
 } **else** { *// 如果不是* **wrongNum**--; *//定义的int数据-1* setData(**wrongNum**); *//设置数据* }  
 }  
 }  
 **return super**.onTouchEvent(event);  
 }  
  
  
 */\*\*  
 \* 点击事件  
 \*/* @Override  
 **public void** onClick(View v) {  
 *//* ***TODO Auto-generated method stub* switch** (v.getId()) {  
 **case** R.id.***play\_vioce***: *//播放单词声音* String text = **wordText**.getText().toString(); *//获取文本* **speechSynthesizer**.startSpeaking(text, **this**); *//传给后台* **break**;  
 **case** R.id.***i\_know\_btn***: *//我会了 按钮的点击操作* **if** (**wrongNum** == 0) {  
 **questionDao**.deleteByKey(**questionDao**.queryBuilder().list().get(**wrongNum**).getId()); *//从数据库删除该条数据* setData(**wrongNum**++); *//刷新数据* } **else if** (**wrongNum** == **questionDao**.queryBuilder().list().size()) {  
 **questionDao**.deleteByKey(**questionDao**.queryBuilder().list().get(**wrongNum**).getId()); *//从数据库删除该条数据* setData(**wrongNum**--); *//刷新数据* } **else** {  
 **questionDao**.deleteByKey(**questionDao**.queryBuilder().list().get(**wrongNum**).getId()); *//从数据库删除该条数据* setData(**wrongNum**--); *//刷新数据* }  
 **break**;  
 **case** R.id.***back\_btn***: *//返回按钮* finish(); *//返回上一个页面* **break**;  
  
 }  
 }  
  
 *//缓冲进度回调通知* @Override  
 **public void** onBufferProgress(**int** arg0, **int** arg1, **int** arg2, String arg3) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//结束回调* @Override  
 **public void** onCompleted(SpeechError arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//开始播放* @Override  
 **public void** onSpeakBegin() {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//暂停播放* @Override  
 **public void** onSpeakPaused() {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//播放进度* @Override  
 **public void** onSpeakProgress(**int** arg0, **int** arg1, **int** arg2) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//继续播放* @Override  
 **public void** onSpeakResumed() {  
 *//* ***TODO Auto-generated method stub*** }  
  
 */\*\*  
 \* 通用回调接口  
 \*/* **private** SpeechListener **listener** = **new** SpeechListener() {  
  
 *//消息回调* @Override  
 **public void** onEvent(**int** arg0, Bundle arg1) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//数据回调* @Override  
 **public void** onData(**byte**[] arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 *//结束回调（没有错误）* @Override  
 **public void** onCompleted(SpeechError arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 };  
  
 @Override  
 **protected void** onStop() {  
 **super**.onStop();  
 }  
}

**第九章主页面设计**

**第一节主页面布局**

**主页面布局为study\_fragment\_layout和home\_layout组成，其代码我在之前第五章的时候就写过，这里不再重复赘述。**

**第二节锁屏状态监听**

**9.2.1ScreenListener.java**

*/\*\*  
 \* Created by 86185 on 2020/5/7.  
 \*/***public class** ScreenListener {  
 **private** Context **context**; *//联系上下文* **private** ScreenBroadcastReceiver **mScreenReceiver**; *//定义一个广播* **private** ScreenStateListener **mScreenStateListener**; *//定义个内部接口  
 /\*\*  
 \* 初始化  
 \* \*/* **public** ScreenListener(Context context) {  
 **this**.**context** = context;  
 **mScreenReceiver** =**new** ScreenBroadcastReceiver();*//初始化广播* }  
  
  
 */\*\*  
 \* 自定义接口  
 \* \*/* **public interface** ScreenStateListener{  
  
 **void** onScreenOn(); *//手机屏幕点亮* **void** onScreenOff(); *//手机屏幕关闭* **void** onUserPresent(); *//手机屏幕解锁* }  
  
 */\*\*  
 \* 获取screen的状态  
 \* \*/* **private void** getScreenState() {  
 PowerManager manager = (PowerManager) **context**.getSystemService(Context.***POWER\_SERVICE***); *//初始化powerManager* **if** (manager.isScreenOn()){ *//如果监听已经开始* **if** (**mScreenStateListener** != **null**){  
 **mScreenStateListener**.onScreenOn();  
 }  
 }**else** {  
 **if** (**mScreenStateListener** != **null**){  
 **mScreenStateListener**.onScreenOff();  
 }  
 }  
 }  
  
 */\*\*  
 \* 写一个内部的广播  
 \* \*/* **private class** ScreenBroadcastReceiver **extends** BroadcastReceiver {  
  
 **private** String **action** = **null**;  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 **action** = intent.getAction();  
 **if** (Intent.***ACTION\_SCREEN\_ON***.equals(**action**)){ *//开屏操作* **mScreenStateListener**.onScreenOn();  
 }**else if** (Intent.***ACTION\_SCREEN\_OFF***.equals(**action**)){ *//锁屏操作* **mScreenStateListener**.onScreenOff();  
 }**else if** (Intent.***ACTION\_USER\_PRESENT***.equals(**action**)) { *//解锁操作* **mScreenStateListener**.onUserPresent();  
 }  
  
 }  
 }  
  
 */\*\*  
 \* 开始监听广播状态  
 \* \*/* **public void** begin(ScreenStateListener listener){  
 **mScreenStateListener** = listener;  
 registerListener(); *//动态注册广播监听* getScreenState(); *//获取监听* }  
  
  
 */\*\*  
 \* 启动广播接收器  
 \* \*/* **private void** registerListener() {  
 IntentFilter filter = **new** IntentFilter();  
 filter.addAction(Intent.***ACTION\_SCREEN\_ON***);  
 filter.addAction(Intent.***ACTION\_SCREEN\_OFF***);  
 filter.addAction(Intent.***ACTION\_USER\_PRESENT***);  
 **context**.registerReceiver(**mScreenReceiver**, filter); *//发送广播* }  
  
  
  
 */\*\*  
 \* 解除广播  
 \* \*/* **public void** unregisterListener(){  
 **context**.unregisterReceiver(**mScreenReceiver**); *//注销广播* }  
}

**第三节主页面功能**

**主页面功能的文件为StudyFragment.java和HomeFragment.java组成，其代码我在之前第五章的时候就写过，这里不再重复赘述。**

**第十章功能展示**

锁屏界面：



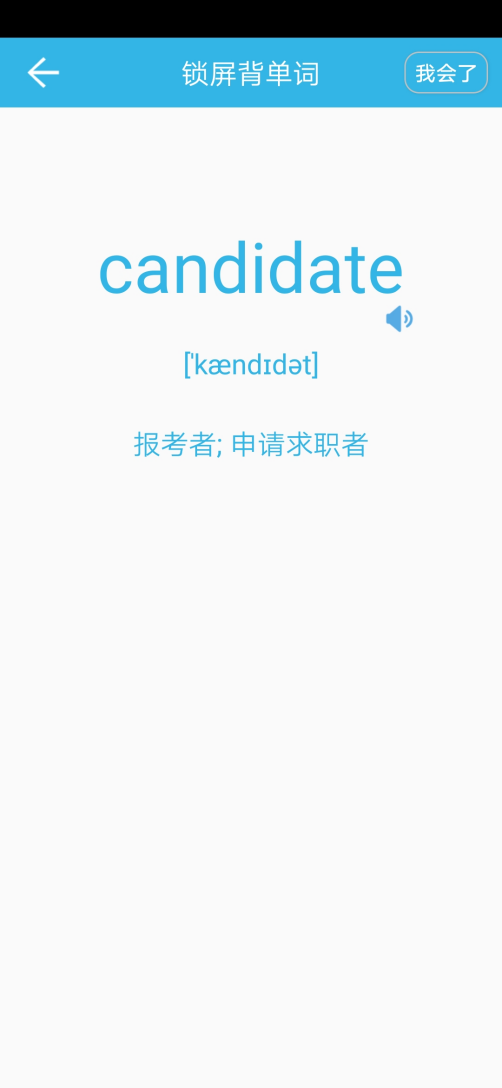
复习界面：



设置界面：



错词本界面：



**总结和心得**

这次实验主要绘制了一次锁屏界面的设计，让我更加充分了解到了Android的开发过程，具体每一步都需要细致思考，对于布局要合理，争取做到完美。