



中山大學
SUN YAT-SEN UNIVERSITY

Android 移动应用开发期中项目报告

三国演义电子辞典

组号	2	联系邮箱	1026051192@qq.com
组长	15352458 庄博伟	成员 1	15352453 朱钱超
成员 2	15352454 朱睿	成员 3	15352229 刘俊材

目录

- 1 概述 2
- 2 背景 2
- 3 项目说明 (重点) 3
 - 3.1 功能模块 3
 - 3.2 逻辑流程 4
 - 3.3 技术说明 (重点) 5
 - 3.4 代码架构 9
- 4 应用效果 10
- 5 开发过程中遇到的问题及解决办法 16
- 6 思考及感想 19
- 7 小组分工 20
- 8 参考资料 21

1、概述

本次项目的题目是三国人物电子辞典。在完成了移动应用开发课程的前六个实验之后，我们已经有能力实现一个实现增删查改功能的类似通讯录的安卓应用，按照题目要求我们将通讯录的任务替换成为三国中各类特色鲜明的人物（添加了 54 人）。我们小组在实现项目基本要求（初始化人物信息、增删查改）的同时，还增加了数据库储存、背景音乐、文字转语音的功能。我们的电子辞典内容完备，人物齐全，唯一的遗憾是有些卡顿，希望在学习了之后的课程我们能够优化线程，内存，让 APP 变得更加流畅。

2、背景

鉴于三国人物绿林好汉颇多，且每个英雄的姓名、籍贯、主效势力等各种人物信息迥异，为了方便用户查找三国人物信息，本次项目的目的是做一个三国人物电子词典，实现对三国人物的个人信息和人物背景进行增删改查。同类应用中，大部分以网页为主，或者没有提供完整的人物系统的信息查询功能，抑或是与用户的互动比较少，用户体验比较差。

本次实验并且提供精致的界面和背景音乐提供用户体验，使用数据库系统，实现对三国人物信息的多种方式的快速查找，和可视化性质比较强的删除更改增加功能，动图效果等以增强与用户的互动；同时当面临大段的人物背景信息时，提供文本转语音阅读功能，增强与用户的互动，减少用户的视觉疲劳。该项目满足了用户对三国人物的查询功能，同时加入文本转语音阅读功能减少用户的视觉疲劳，兼具实用性和创新性。项目多处使用可视化和动画效果，从图标到提示都尽量以图代替文字，简单方便、易于操作、交互性强。

3、项目说明（重点）

3.1 功能模块

我们小组在实现三国演义电子辞典中注重基础功能的实现，并且增加了许多创新的功能。主要的功能如下：

- 1. 辞典欢迎界面（载入界面）：打开应用时会有一个欢迎界面，该界面保持 3 秒，背景图片随机选择。
- 2. 辞典主列表：带有 50 多个人物的基础信息项（头像，姓名，主效势力，生卒年月），且拥有类微信联系人列表右侧的首字母检索栏与词条查询框（输入人物关键字，动态筛选包含该关键字的条目）。在主列表中我们不希望用户能够删除基础的条目，因此只提供了词条的增加。
- 3. 辞典收藏列表：包含用户收藏过的人物，在该列表用户可以对收藏过的人物条目进行删除。
- 4. 人物详情：带有人物的详细信息项（头像，姓名，字号，性别，生卒年月，籍贯，主效势力，全身图片以及配图）。在该界面下方第二个按钮是显示人物更多信息，包含人物的生平简介以及一个简单的 gif 图来展示人物；第三个按钮可以对该人物信息进行修改。
- 5. 创新功能：数据库导入数据；对 UI 界面进行了美化；主界面按钮播放/停止背景音乐；人物更多信息中语音讲解人物生平等。

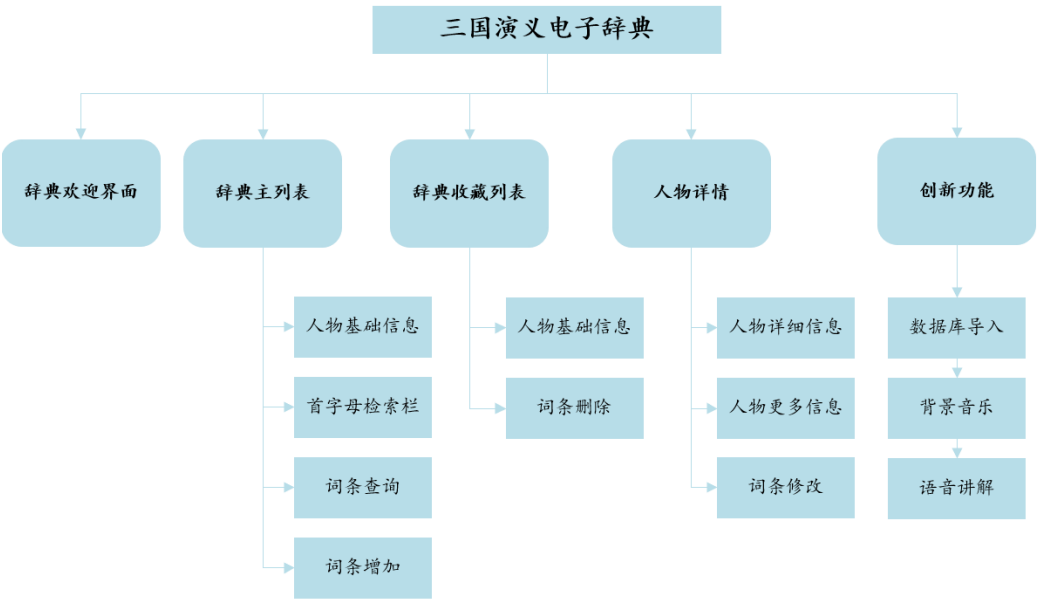


图 3.1 功能模块

3.2 逻辑流程

由于该应用属于辞典类应用，所以主要是呈现出更多的信息给大家。按钮的跳转一般都在界面的明显位置，所以操作起来什么简单。下列简单介绍本应用的使用逻辑流程：

首先我们像大多数 APP 一样采用了点击应用生成几秒钟的封面，封面下方是我们的应用名和 logo，而背景则采用了库中随机背景产生，然后跳转至主界面，主界面下方含有几个主要按钮：分别是默认的列表项、编辑人物界面、背景音乐按钮和收藏列表。

点击列表项任意英雄，会跳转到指定英雄的详细界面，可进行收藏和编辑操作，并且点击更多信息进行下滑可以查看生平事迹和查看 GIF 图。

而点击编辑按钮也会跳转到编辑界面，设置好相关信息后右滑点击完成即可完成添加。具体操作请看下面逻辑流程图。

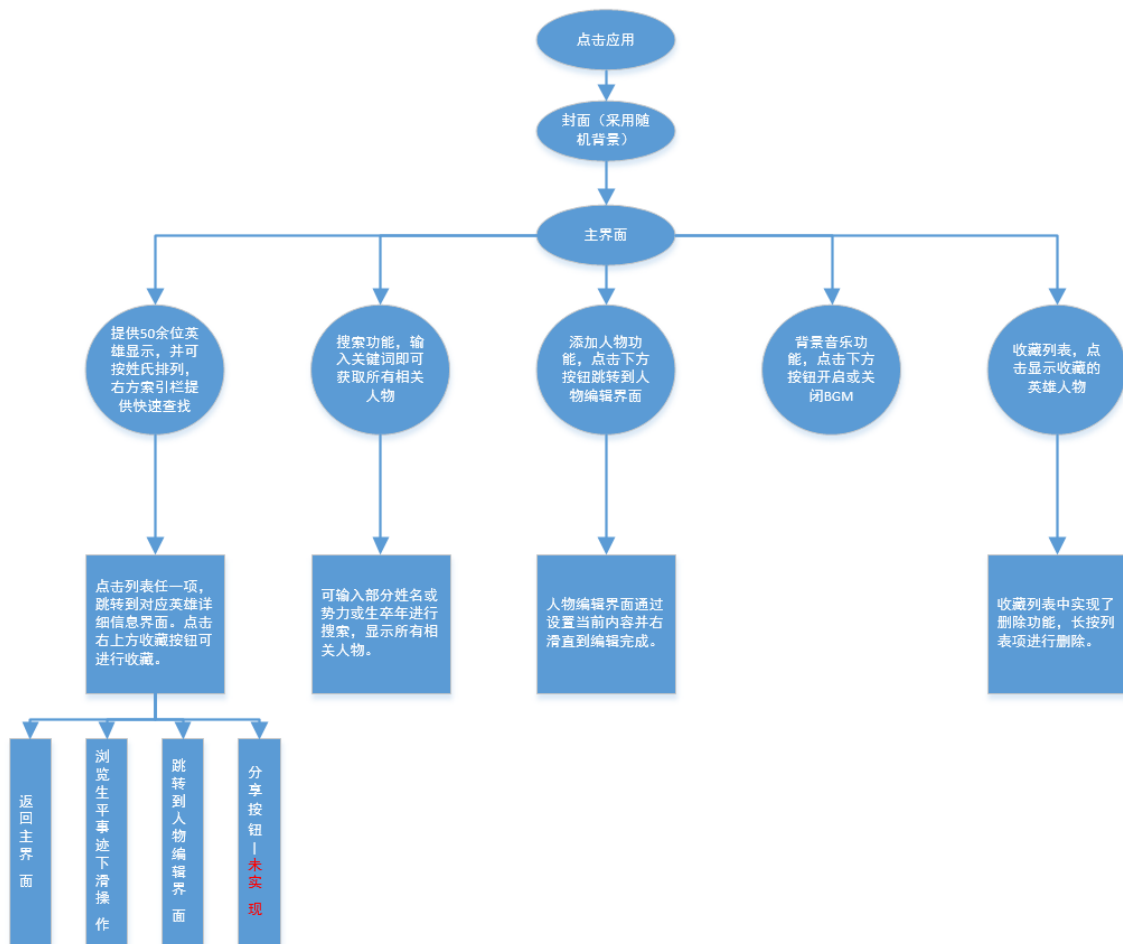


图 3.2 逻辑流程

3.3 技术说明 (重点)

3.3.1 布局

在整体布局与控件到的选择上面大量参考了 Github 上的开源项目，取精华而去糟粕，在实现整体词典框架的同时不断添加新鲜的小内容，充实词典的内容和可操作性。以下是本次布局方面的重点使用技术：

(1) 按拼音首字母排序的索引：我们在主列表界面的右侧加入了一个 26 字母的索引，在此过程还要引入 pinyin 的 jar 包，来得到每个英雄人物的中文首字母，并继续得到英文的首字母，将其按照字典序的方式分别放入各个对应的字母栏下方，主要用到了 StringHelper 和 ListviewAdapter 类，并在 mainactivity 中根据人物姓名的首字母信息进行的排序与右边索引栏中一一对应，基本实现了通过索引栏滑动快速寻找某项人物信息的功能。下面是功能实现模块图。

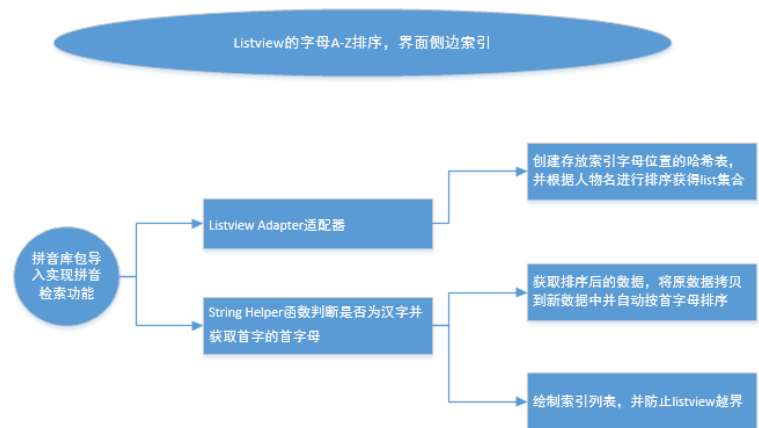


图 3.3.1 listview 实现右侧栏索引功能模块图

(2) 素材的处理：本次实验的大量人物图像素材如头像和全身图和相关配图均来源于三国无双游戏，但是网上的大多数图片都有背景，这样就不能很好地利用我们固定的背景来更换人物图，所以我们利用 photoshop 对所有的配图都采取了去背景的抠图处理，确实很困难。另外，诸如各种图标和背景素材的选取我们都做了大量地修图工作，确实，这些素材奠定了我们 UI 界面的整体画风可以更加自然。

(3) 繁体字体：在 assets 文件中导入了繁体字体的 ttf 文件，并添加依赖为相应的文本控件设置 typeface 才得以让繁体成功显示，这一部分也是 debug 了好久。

- (4) GIF 图的显示：在详细界面的最低端，我们设置了 gif 图的显示，这都依靠于我们耐心地在 Github 上不断地尝试各种显示 gif 图的方式才得以成功。
- (5) 详细信息界面下滑时的动画：详细信息界面我们在显示生平事迹的时候会 有一个伸缩的小动画效果，整体界面也除于一个很长的状态，这也都依靠于我们在 github 上不断搜寻优秀的 viewpage 才得以实现。
- (6) 编辑界面的排版与动画显示：由于我们添加人物和编辑人物信息均在同一个界面，所以该界面的设计就需要花费一些心思。这里我仍然采用了可不断右滑的 viewpage，并使每一页的控件可以不断地以动画形式滑动，但是改页面在多次滑动后可能会出现控件位置移动的情况，估计是优化效果做的不太好。

3.3.2 java 文件

Java 文件中的代码编写基本是根据功能逻辑来完成的，用的代码大部分是这半个学期安卓课上学到的东西，因此这里只挑一些关键的部分来讲。

- (1) WelcomeActivity.java：该文件对应的是应用欢迎界面，这里用到了随机函数 random 来选择随机的封面图片显示。然后通过使用 handler 的 sendMessageDelayed 的方法设置 3 秒中的延时，再跳转到应用主界面。
- (2) MainActivity.java：该文件中设置了主界面元素的属性，比如主列表与收藏列表的适配器配置，以及列表点击事件与下方按钮的点击事件等。这里需要说明的是上方搜索框的使用。我们使用的方法是根据 EditText 搜索框 ListView 动态显示数据。方法实现的比较简单，设置 EditText 的监听器，当监听到文本改变时，就用 Handler post 一个 Runnable 去做相应的改变：如果输入的内容与某条目所含有的信息相同，则该条目会被筛选出来，然后列表的适配器进行 notifyDataSetChanged()更新。注意当搜索框内容清零时，要对列表进行恢复。
- (3) 关于人物条目的添加与修改，我们使用了两个相似的 java 文件：edition.java 对应的是人物的修改，editionForMain.java 对应的是人物的增加。因为对人物的修改是针对某一个已有条目的信息进行修改，因此在布局文件中的信息会默认为该条目的信息；而对人物的增加是增加一个新的条目，因此编辑界面的默认值是“无”，“未知”之类的。

- (4) more.java：该文件中对应人物详细信息界面。我们在拓展此功能的时候是比较晚的时候，因此在这里面只添加了部分人物的详细信息，其他的统一使用默认的三国时期的概述模版与 gif 图。
- (5) 关于 Activity 之间的交流，我们充分使用了课堂上所学的 startActivityResult 以及 EventBus，主要是对主列表与收藏列表的条目更新。

3.3.3 数据库

平时见到的 android 数据库操作一般都是在程序开始时创建一个数据库类，然后在类里面初始化数据库，并且向这个数据库添加信息，然后在其他类里面使用这个数据库。虽然这样的操作能够保证在 APP 关掉再打开时能够保存增删查改的状态，但是我们考虑到这个 APP 是一个辞典，最根源的人物数据库是不允许用户接触的，但同时要满足用户的增删改，这很矛盾。于是我们想出一个解决办法：

在 AndroidStudio 之外将人物信息保存到数据库中，把数据库存在手机里，然后在启动 APP 的时候将人物信息从数据库中导出，加在到一个变量里面，允许用户对这个变量进行操作，而不是对数据库操作。

Android 系统下数据库应该存放在 /data/data/com.*.*(package name)/ 目录下，所以我们需要做的是把已有的数据库传入那个目录下，记得在 Manifest 里面注册权限。

把原数据库包括在项目源码的 res/raw 目录下，建立一个 DBManager 类

```
public class DBmanager {
    private final int BUFFER_SIZE=400000;
    public static final String DB_NAME = "sanguooo.db"; //保存的数据库名
    public static final String PACKAGE_NAME = "com.example.miracle.midproject";
    public static final String DB_PATH="/data"+ Environment.getDataDirectory().getAbsolutePath()+ "/" +PACKAGE_NAME; //在手机里存放数据库的位置
    private SQLiteDatabase database;
    private Context context;
```

图 3.3.3.1 DBmanager 类

用 FileInputStream 读取原数据库，再用 FileOutputStream 把读取到的东西写入到那个目录

```
private SQLiteDatabase openDatabase(String dbfile) {
    try {
        if (!(new File(dbfile).exists())) {
            InputStream is=this.context.getResources().openRawResource(R.raw.sanguooo);
            FileOutputStream fos=new FileOutputStream(dbfile);
            byte[] buffer =new byte[BUFFER_SIZE];
            int count=0;
            while((count=is.read(buffer))>0) {
                fos.write(buffer, 0, count);
            }
            fos.close();
            is.close();
        }
        SQLiteDatabase db=SQLiteDatabase.openOrCreateDatabase(dbfile, factory: null);
```


图 3.3.3.2 读取数据

打开 DeviceMonitor，检查是否导入成功：

com.example.miracle.midproject	2017-11-25	04:53	drwxr-x--x	
> cache	2017-11-21	00:13	drwxrwx--x	
lib	2017-11-25	04:53	lrwxrwxrwx	-> /data/ap...
sanguo.db	12288	2017-11-22	09:44	-rw-----
sanguo.db-journal	8720	2017-11-22	09:44	-rw-----
sanguo.dbf	12288	2017-11-22	05:26	-rw-----
sanguo.dbf-journal	8720	2017-11-22	05:26	-rw-----
sanguoo.db	69632	2017-11-22	15:34	-rw-----
sanguoo.db-journal	8720	2017-11-22	15:34	-rw-----
sanguooo.db	36864	2017-11-22	17:03	-rw-----
sanguooo.db-journal	8720	2017-11-22	17:03	-rw-----

图 3.3.3.3 导入数据

接着在 MainActivity 里面启动这个类，建立数据库，然后在需要设置人物信息的时候打开数据库：

```
private void setData() {  
    db=SQLiteDatabase.openOrCreateDatabase(path: DManager.DB_PATH + "/" + DManager.DB_NAME, factory: null);  
    System.out.println(db.getPath());  
    persons=getCity();  
}
```

图 3.3.3.4 建立数据库

使用 cursor，执行查询语句并且传递，

```
private ArrayList<Person> getCity() {  
    Cursor cur = db.rawQuery("SELECT name,sex,bddate,home,force,secname,picturea,pictureb,picturec FROM SanGuo", selectionArgs: null);  
    if (cur != null) {  
        int NUM_per = cur.getCount();  
    }  
}
```

图 3.3.3.4 执行查询语句

遍历所有数据库元组的通用语句：

```
If cursor.moveToFirst  
Do something  
while cursor.moveToNext
```

3.3.4 创新功能

- (1) 背景音乐：即在用户使用 APP 的同时，播放背景音乐，增强用户体验与互动性；这里使用了后台服务，使用 Binder 来实现 Activity 与 Service 之间的通信，并实现了动态获取文件读取权限。
- (2) 文本阅读功能：即将文本转化为语音阅读的方式，使得用户免去阅读大段人物背景信息文字的视觉疲劳；这里使用了讯飞科技的 TTS 的 API 接

口，把 API 里面的语音合成器封装成一个类，类里面加入了一些设置语音参数、设置语音中止播放的方法，方便主体 Activity 调用这个功能。

3.4 代码架构

关于代码架构，这里给出我们总体的代码架构图以及数据库使用的代码架构图，具体的代码可以参考附件的工程项目。

3.4.1 总体代码架构



图 3.4.1 总体代码架构

3.4.2 数据库代码架构

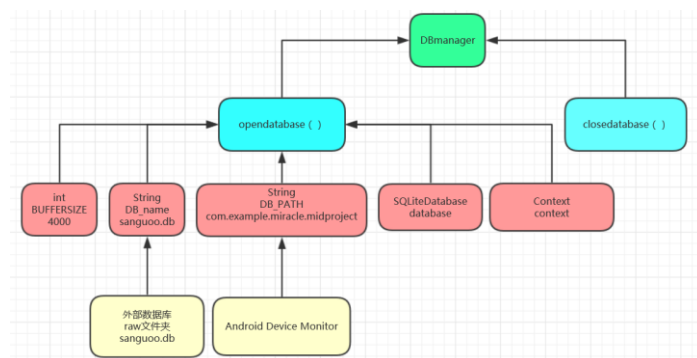


图 3.4.2.1 数据库代码架构

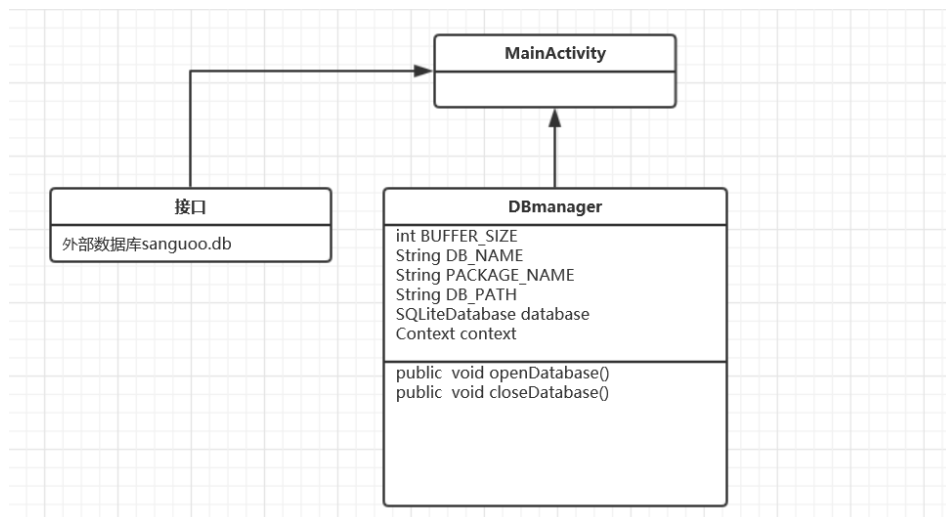


图 3.4.2.2 数据库类关系图

4、应用效果

首先打开应用，欢迎界面，背景图片是随机选择的：



图 4.1 应用欢迎界面

进入辞典主界面，显示的是带有首字母索引人物信息的 list，最上方有搜索功能。



图 4.2 应用主界面

搜索功能：输入人物信息，列表动态更新筛选关键信息。（因为虚拟机不能输入中文，此处输入的数字代表生卒年，当然输入名字与主效势力也是可以动态筛选的）



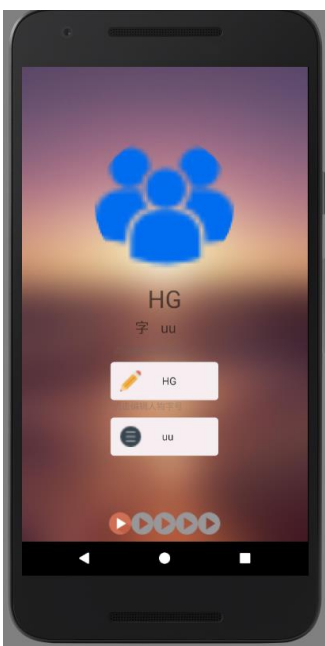
图 4.3 搜索功能

界面右边的仿微信字母跳转搜索功能：



图 4.4 首字母检索功能

下方第二个按钮是添加词条按钮，可以添加人物，共有五步如下：



(1)



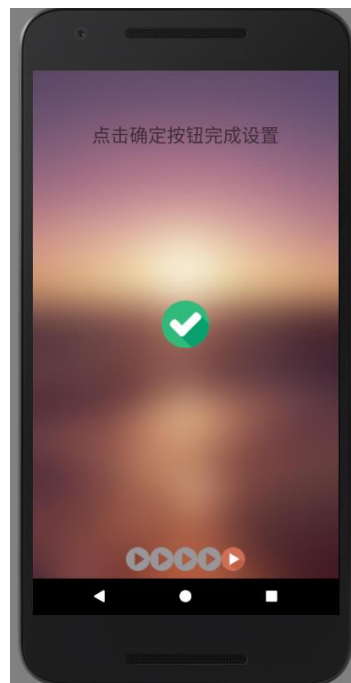
(2)



(3)



(4)



(5)

回到主界面可以在人物列表中看到已经新添加了 HG 这个人，打开以后发现信息和刚才设置的符合：



图 4.5 添加人物

点开一个人物查看具体信息，此处左上方与下方第一个按钮都是回到主界面。



图 4.6 查看人物具体信息

点开下方第二个按钮是查看人物的更新信息，可以看到人物的生平简介以及一张 gif 图展示。右上方的按钮实现了语音转文字的功能，点击可以自动阅读人物的详细信息。

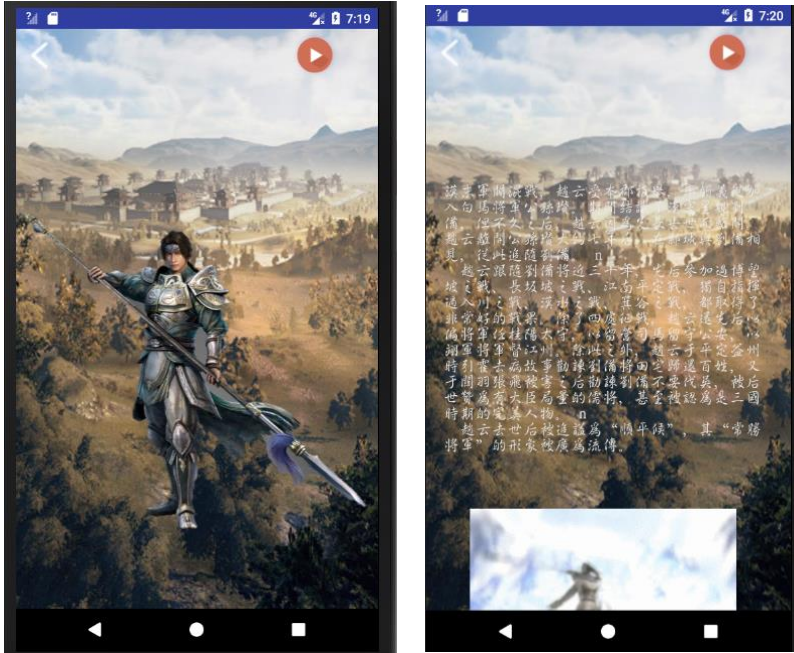


图 4.7 查看人物更多信息

点开下方第三个按钮是修改该人物的信息，步骤同上面的五步，下面对人物进行修改：（第四个按钮的分享功能最后未实现）



图 4.8 修改已有人物的信息

点击右上角的红心可以收藏人物，收藏后的人物在主界面收藏列表中可以

看到。

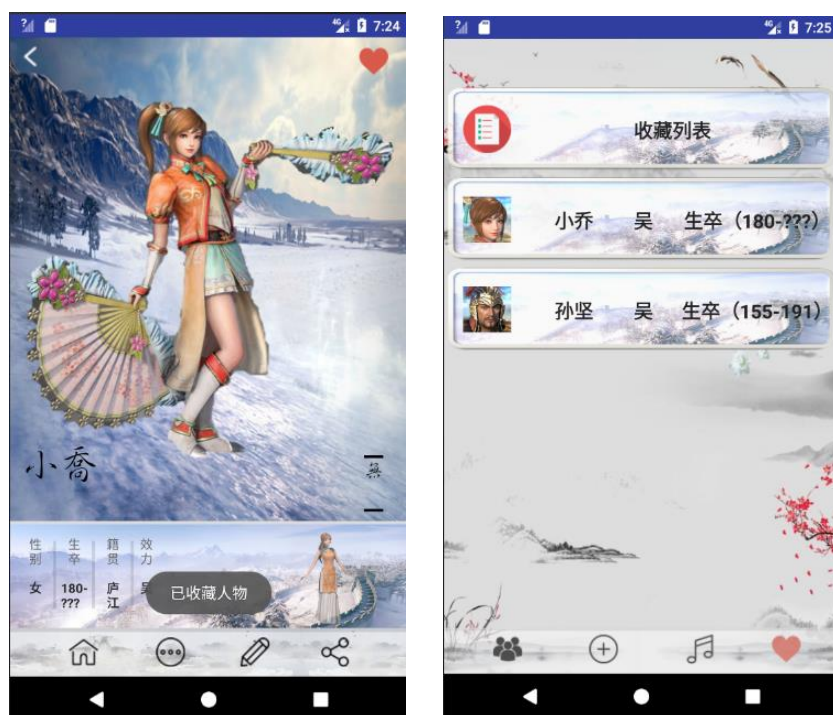


图 4.9 人物收藏

收藏人物列表中长按条目可以将人物移除出收藏夹：

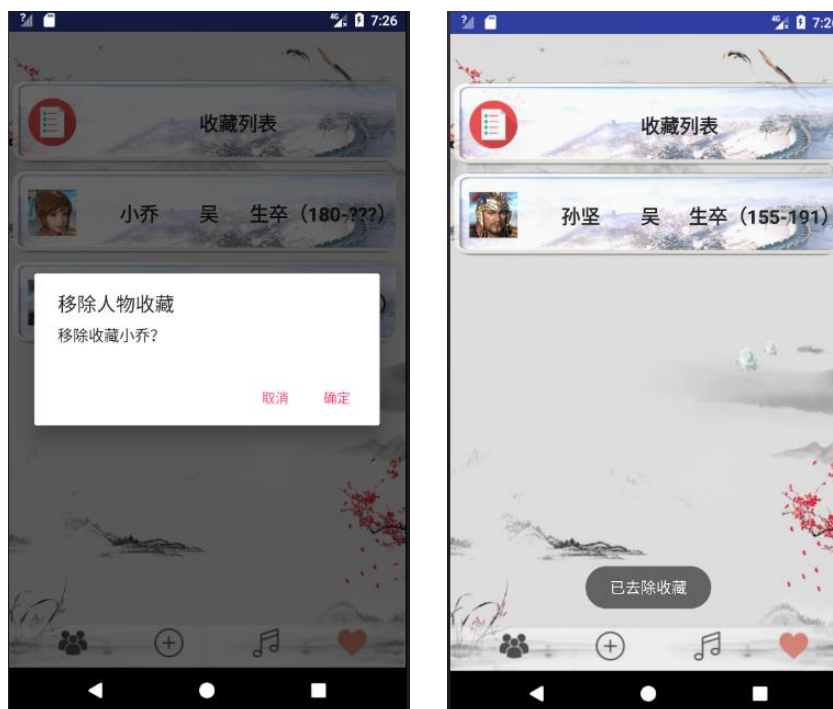


图 4.10 人物收藏删除

点击下方第三个按钮可以播放背景音乐：

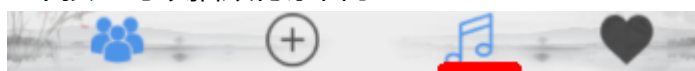


图 4.11 背景音乐播放

5、开发过程中遇到的问题及解决办法

• 问题 1：主列表右侧栏索引的实现

制作索引的时候虽然大量参考了网上的一些开源项目，但是实际嵌入我们的三国词典项目还是有一定难度的，首先就是对 jar 库包导入和使用的不熟悉，还有对一些排序容器和方法使用的陌生也让我们困扰了很久。

解决方法：在搜索了大量项目和进行实践后，才终于实现了该功能。其实最核心的函数还是通过导入的 pinyin4j 包获取人物名的姓的拼音，然后截取首字母，然后进行整体排序。

• 问题 2：素材的获取

由于本次是实现一个字典应用，所以主要是呈现更丰富的信息给用户。最开始不知道该如何选取相关的图片素材，选取的图片大部分带有背景，无法完美嵌入我们的布局。

解决方法：在选取了背景、各种图标与 logo、人物图和头像、编辑界面各种选择样式和 gif 图等等，这些素材都需要花费大量的时间从网上获取合乎整体画风的素材再进行加工才可以使用。开始大家对素材的抠图都不熟悉，后面在尝试了几张图片的加工慢慢也就熟悉了。

• 问题 3：布局空间调整与小动画的呈现

由于详细信息通常会比较长，再配合上图片与 gif 图的使用在普通的布局上会放不下。另外在添加与修改人物信息时的布局也是困扰了很久。

解决方法：在实现 more.xml 这个布局界面的时候，我们采用的整体下拉的方式来呈现生平事迹和 gif 图，还有 edition.xml 文件也实现了不断右滑的方式来呈现不同的设置内容，但是这两个布局的控件放置的位置都很讲究，由于在添加了小动画的情况下，界面预览不能使用，所以只有不断尝试修改各种边距再打开虚拟机进行调整，虽然没有技术难度，但是各种细节的调整也确实需要耗费大量时间来进行。

• 问题 4：搜索框的使用

不知道该如何设置搜索框对列表的筛选作用。

解决方法：通过在网上查阅资料后，学习到了一种比较简单的方法，是根据 EditText 搜索框 ListView 动态显示数据。方法实现的比较简单，设置 EditText 的监听器，当监听到文本改变时，就用 Handler post 一个 Runnable 去做相应的改变：如果输入的内容与某条目所含有的信息相同，则该条目会被筛选出来，然后列表的适配器进行 notifyDataSetChanged()更新。而在实际实现的时候发现当搜索框清零时，列表无法回到带有首字母索引的原始列表。经过调整后需要在搜索框内容为空的时候对列表进行恢复操作。

• 问题 5：数据库存放图片

不知道如何在数据库中存放需要用到的图片。

解决方法：开始想到用二进制存进数据库，论坛上的人说这样 APP 会很卡很卡，启动时间很长。后来考虑将数据库表放在服务器上，打开 APP 时用流量加载图片（大部分企业应该是这样做的吧）。最后我们是这样做的：比如把 R.drawable.zhangfei 这个 string 存进数据库，然后使用 JAVA 的反射机制，把这个字符串转换成 AndroidStudio 里的图片 int 类型。

• 问题 6：数据库文件问题

不知道数据库文件放在哪，如何读取数据库文件，以及外部数据库如何导出成.db 文件。

解决方法：Android 系统下数据库应该存放/data/data/com.*.*(package name)/ 目录下，所以我们需要做的是把已有的数据库传入那个目录下，记得在 Manifest 里面注册权限。

而关于读取数据库文件需要建立一个 DBManager 类，记录一些数据库位置信息，然后再用 FileInputStream 读取原数据库，再用 FileOutputStream 把读取到的数据库写入目录。

关于导出.db 文件，我们查了很多资料，最后使用 sqlserver2sqlite_converter 工具将数据库导出成.db 文件，网上可以下载到这个软件，很实用。

• 问题 7：getInstance()实现未有同步出现。

封装 TTS 的 API 时，getInstance()里面没有同步声明静态对象的空间时导致语音无法播放。

解决方法：在封装 API 时，因为想要把类封装成静态可以调用的，即封装成一个不需要声明实例即可通过 getInstance()链式调用的类，这时需要在类 Audio 里面声明一个 Audio 的静态对象，getInstance()返回该静态对象，但是该对象未分配空间时，需要使用同步 synchronized 的方法分配空间。

• 问题 8：模拟器上调试背景音乐。

因为使用模拟器调试背景音乐，与真机调试环境不一样，需要修改获取源文件的代码，否则无法获取文件播放音乐。

解决方法：在使用背景音乐这个功能时，因为在模拟器上测试，没有把音乐放入 SD 卡，而是放入资源文件里面，这时播放器获取文件源的方式也要改变，从 Media.GetDatasource()改为 Media.create()。

6、思考及感想

这次期中实验项目是做一个辞典应用，可以说是综合了课程内六次实验知识的成果。我们小组从讨论，分工开始，团队合作开发井井有条，大家都在指定的时间内完成了模块的实现，有困难的时候一起研究讨论，总的来说虽然碰到不少问题，但是通过强有力的团队能力，最后都一一解决并且实现了最终的应用成品。

首先是关于布局方面的内容，虽然说是编辑一些关于布局的 xml 文件，但任务远不止于这么简单。由于有些特殊的控件想要添加一些酷炫的动画效果，必须还得在 java 文件中添加函数实现，因此设计布局与编写 java 文件的人员需要配合好整体思路和代码风格，不然在实现各个部分的具体功能时候编辑代码就会变得很乱，所以在分工时候还是很讲究配合的。另外，我们在本次构思布局的过程中也确实学到了很多，如何在 GitHub 上寻找精美的 UI 布局为我们所用？如何添加 gif 控件？如何呈现繁体字的效果？如何实现类似通讯录的索引布局？如何实现一些简单的动画效果？所以我们认为程序中负责 UI 界面的人就掌握了整个应用的走向，所以更要注重每一个控件的细节。

另外，本次应用的素材选取也很讲究，我们没有采取参考网站上模糊不清的图像，而是参考了三国无双游戏，截取了该游戏中大量的高清图，并对图片进行了大量处理再进行使用的，另外，背景图片、图标 logo、人物配图等的选取都花费了大量时间，但是精工才能出精品。在最后看到成品的时候，还是会觉得很值得。

对于编写 java 文件代码的时候，总会遇到多多少少的 bug，比如说在运行测试的时候遇到了很多次程序崩溃的情况。这时候我们都会充分利用 AS 的 debug 功能，找到代码中引发程序崩溃的地方，通常能够很顺利的解决问题。而在设置 Activity 之间的跳转与交流的时候，通过复习半个学期这六次实验，我们熟练使用了 startActivityForResult 与 EventBus 等方法，在编写的时候需要高度的细心，保证每个获取的控件，传递的变量，设置的文本、图片信息无误。

关于数据库的使用，起初在做数据库的时候想了很多，数据库到底应该这么用在这里？与其他小组的同学讨论后，发现很多小组都是在一个数据库管家类里面初始化数据库，往数据库里面添加数据，然后在 MainActivity 或者其他类里面使用这个数据库。虽然这样能保存关掉 APP 后再人物的增删改，但是我们认为，一个辞典的内部数据库是不能允许用户接触的，比如说百词斩，能让

用户修改四六级词库吗？显然不能，用户只能对自己的收藏，对自己的背单词计划进行增删改。我们秉承了这个思想，自己在 AndroidStudio 外面利用 SQL SERVER 提前做好了一个人物信息数据库，使用转换工具将数据库表导出成为.db 文件，然后导入安卓手机数据库存放路径用，FileInputStream 读取原数据库，再用 FileOutputStream 把读取到的东西写入到那个目录。最后将数据库信息加载到一个新的人物对象 list 中，实现了对 list 的增删改，而对原本的数据库用户是没有权限接触的。

在实现拓展功能的时候，我们需要调用其它 API 时，需要详细阅读 API 的文档，并且自己实现一个小工程确定这块功能无误后，再把代码移植到项目代码里面。并且直接现成的 API 需要自己进行进一步的封装，才能方便我们的阅读和调用。

我们在本次实验中从开始的毫无头绪，到后面不断寻找开源项目和素材并不断尝试调整的过程中确实学到了不少东西，我们所有人都很享受这种不断学习的过程，但是美中不足的是最后没有时间对代码风格进行优化，使得我们的代码整体看来会很乱，没有进行必要的函数封装。并且最后的成品文件规模还是过大了，希望在以后的项目中可以进行调整。

7、小组分工

表 1：小组分工

组员	学号	负责内容	贡献百分比（%）
庄博伟	15352458	编写 Java 文件基本逻辑与各 Activity 之间的交流，人物信息素材收集，实验报告编写与汇总。	25
朱钱超	15352453	所有布局文件的设计，索引列表的设计，素材收集，gif 图与繁体字体的实现，实验报告编写与排版。	25
朱睿	15352454	数据库使用设计，实验报告编写。	25
刘俊材	15352229	拓展功能：语音转文字与背景音乐，实验报告编写	25

8、参考资料

- ① Github 优秀 UI 开源项目：<https://github.com/wasabeef/awesome-android-ui>
- ② 《真三国无双》系列全武将图片素材：
<http://www.gamersky.com/handbook/201605/750167.shtml>
- ③ 图标 logo：<https://www.flaticon.com>
- ④ Android Unleashed 安卓应用开发揭秘.杨丰盛.机械工业出版社 2008
- ⑤ 本学期往期实验内容