三国说

安卓期中项目

组号：90

组长：蔡烨 15352010

邮箱：caiy29@mail2.sysu.edu.cn

QQ：973044852

组员：蔡丽芝 15352006

曾何萌 15352019

曾玉莹 15352024

目录

一、项目概述

《三国演义》是我国优秀古典小说，此书人物众多，虽各具特色但初读者难免有混淆或者记不清人物的情况。因此，我们实现了三国人物的电子辞典，此应用支持人物的增删改查功能，并提供了人物的详情。数据库的使用使得整个项目更加规范；拍照以及本地选择图片上传作为人物图像的功能使得本应用在不同性格的人手中展现出不同的个人特色;在用户交互界面的设计中,我们采用了颇具中国风格的水墨主题;除此之外，我们还增加了背景音乐，凸显了中国式的韵味。

二、项目背景

在当今社会能静下心来阅读《三国演义》的人越来越少，虽然基于三国的游戏并不鲜见，但是像三国电子辞典这一类的应用却迟迟没有出现，这对真正的文学爱好者来说是非常遗憾的事情，于是“三国说”这一应用诞生了。在设计这个应用的时候，我们特意加入了一点三国元素以及中国风格。我们所使用的应用图标上的三个字是:“临江仙”，这是《三国演义》开篇中出现的杨慎的词作的词牌名，而打开应用时出现的闪屏则是这首词作的书法作品。字体以及背景也尽力加入了一些中国古典元素。在完成这个项目的时候，我们不仅着力于实现应用的功能完整性，更希望能体现出对古典文学的尊敬。

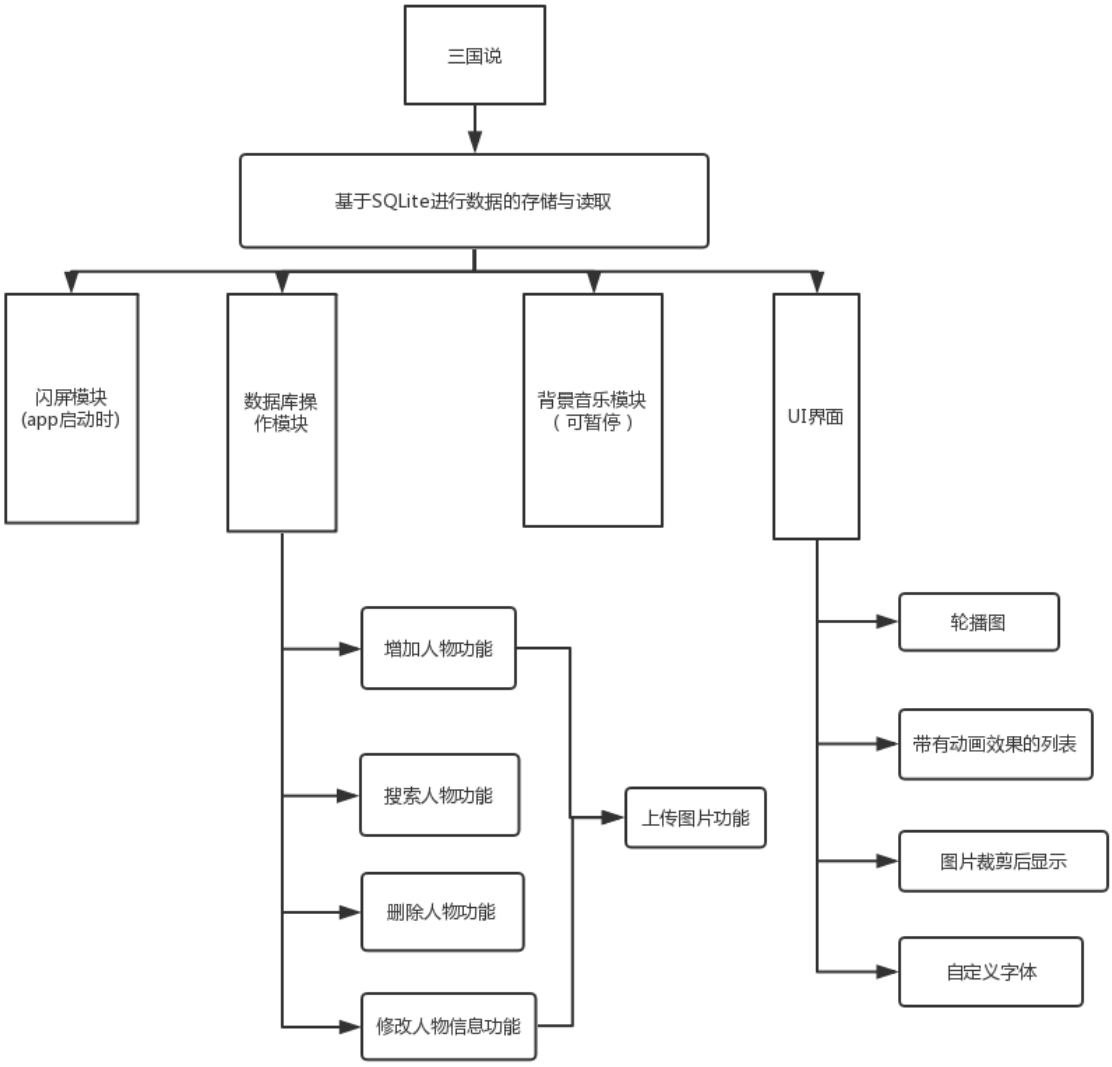
在应用功能方面，“三国说”使用了数据库进行人物信息的管理，并且在数据库的基础上实现了增删改查的功能，使得整个应用的逻辑清晰；用户可以修改人物原本的图片，选择自己喜欢的图片上传，在增加人物的时候也可以自行选择图片，尊重了用户的个性需求。

在用户交互界面方面，“三国说”的基调是中国风，采用了水墨画作为背景；使用《临江仙》这一书法作品作为闪屏界面；增加了轮播图展示部分人物和场景；选择了自行导入的具有中国风的字体。

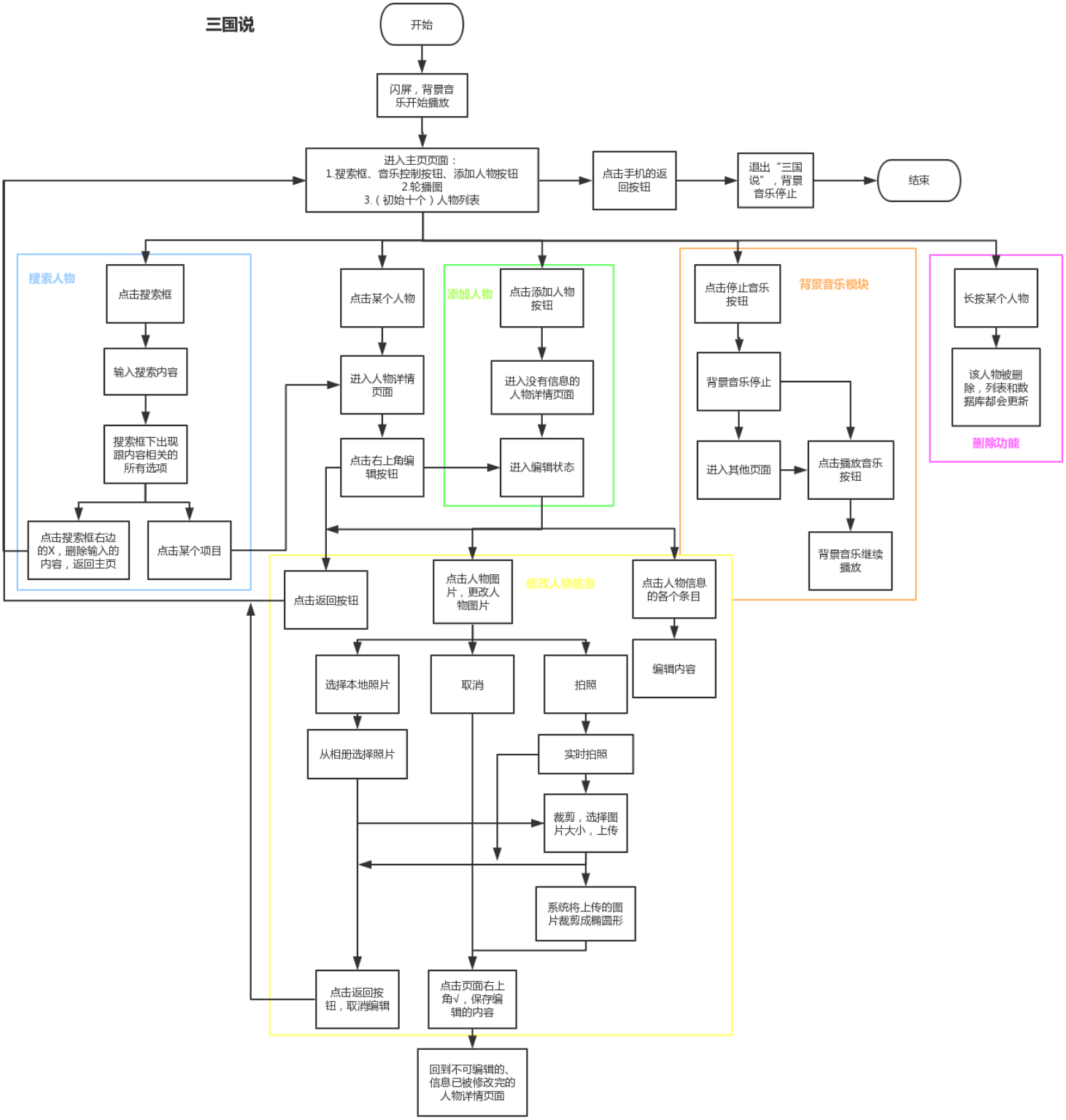
在用户体验方面，应用运行稳定，功能清晰明了易于上手。于此同时，“三国说”具有背景音乐播放器，播放的音乐是我们精心挑选的中国风轻音乐，用户也可以通过音乐播放器的标识自行选择关闭或者打开音乐播放器。

三、项目说明

1. 功能模块


三国说
基于SQLite进行数据的存储与读取
闪屏模块
(app启动时)
数据库操
作模块
背景音乐模块
（可暂停）
UI界面
增加人物功能
搜索人物功能
删除人物功能
修改人物信息功能
上传图片功能
轮播图
带有动画效果的列表
图片裁剪后显示
自定义字体

2. 逻辑流程


三国说
开始
闪屏，背景音乐开始播放
进入主页页面：
1.搜索框、音乐控制按钮、添加人物按钮
2.轮播图
3.（初始十个）人物列表
点击手机的返回按钮
退出“三国说”，背景音乐停止
结束
搜索人物
点击搜索框
输入搜索内容
搜索框下出现跟内容相关的所有选项
点击搜索框右边的X，删除输入的内容，返回主页
点击某个项目
点击某个人物
进入人物详情页面
点击右上角编辑按钮
添加人物
点击添加人物按钮
进入没有信息的人物详情页面
进入编辑状态
背景音乐模块
点击停止音乐按钮
背景音乐停止
进入其他页面
点击播放音乐按钮
背景音乐继续播放
长按某个人物
该人物被删除，列表和数据库都会更新
删除功能
点击返回按钮
点击人物图片，更改人物图片
修改人物信息
点击人物信息的各个条目
编辑内容
选择本地照片
从相册选择照片
取消
拍照
实时拍照
裁剪，选择图片大小，上传
系统将上传的图片裁剪成椭圆形
点击返回按钮，取消编辑
点击页面右上角√，保存编辑的内容
回到不可编辑的、信息已被修改完的人物详情页面

3. 技术说明

1. SQLite。

本次项目采用轻量级的 SQLite 作为数据库，在之前安装 AndroidStudio 时下载的 SDK（\platform-tools 目录下）就已经包含了该 SQLite。该项目用到 SQLiteDadaBas 和 SQLiteOpenHelper 以及自定义的 FigureRepo 类来实现三国人物的增删改查功能。

2 闪屏：

当应用打开时，首先跳到闪屏界面，通过一个子线程计时 5 秒后回到主界面。闪屏所用的图片通过 AlphaAnimation 设置动画，即改变透明度达到逐渐清晰的效果。闪屏出现时，按下菜单键可跳过闪屏，按下返回键可直接退出应用。

```
//设置闪屏界面的动画效果，即从全透明到完全不透明
LinearLayout ll = (LinearLayout) findViewById(R.id.sp);
AlphaAnimation alphaAnimation = new AlphaAnimation(0.0f, 1.0f);
alphaAnimation.setDuration(5000);
ll.startAnimation(alphaAnimation);
```

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    super.onKeyDown(keyCode, event);
    //按下menu键可跳过闪屏
    switch(keyCode){
        case KeyEvent.KEYCODE_MENU:
            mSplashActive = false;break;
        case KeyEvent.KEYCODE_BACK:
            android.os.Process.killProcess(android.os.Process.myPid());break;
        default:break;
    }
    return true;
}
```

3. 图片上传与（系统）裁剪（图库选择与拍照）。

通过调用系统功能：

（1）获取文件 Intent.ACTION_GET_CONTENT(android.intent.action.GET_CONTENT)

（2）拍照 Media.ACTION_IMAGE_CAPTURE(android.media.action.IMAGE_CAPTURE)

（3）图片裁剪（指定裁剪比例与宽高）com.android.camera.action.CROP

获取文件和拍照（首先要获取权限然后指定一个临时位置保存图片）返回的都是图片的 Uri，将其图片保存到 app 的私有目录下（数据库中保存的是图片的路径），裁剪之后得到 Bitmap 类型的图片然后显示。

4. 图片裁剪为圆形：

```
Bitmap bmp = BitmapFactory.decodeResource(res, figure.getPic());
bmp = ImageUtils.toRoundBitmap(bmp);
pic_iv.setImageBitmap(bmp);
```

从数据库中获取人物图片并转化为 btimap 格式，调用 ImageUtils 类中的 toRoundBitmap 函数，裁剪图片为圆形。

在这里使用到了一个新的类 ImageUtils，在这个类中有对关于处理图片的各种功能函数，就比如我们需要用到的 toRoundBitmap 函数，将 bitmap 格式的图片裁剪为圆形的。toRoundBitmap 函数的实现代码解释：

```
public static Bitmap toRoundBitmap(Bitmap bitmap) {
```

传入 toRoundBitmap 函数里的参数对象必须是 Bitmap 类型的，因此我们在调用这个函数的时候，必须将图片转化为 Bitmap 格式

```
Canvas canvas = new Canvas(output);
final int color = 0xff424242;
final Paint paint = new Paint();
final Rect src = new Rect((int)left, (int)top, (int)right, (int)bottom);
final Rect dst = new Rect((int)dst_left, (int)dst_top, (int)dst_right, (int)dst_bottom);
final RectF rectF = new RectF(dst);
paint.setAntiAlias(true);
canvas.drawARGB(0, 0, 0, 0);
paint.setColor(color);
canvas.drawRoundRect(rectF, roundPx, roundPx, paint);
paint.setXfermode(new PorterDuffXfermode(Mode.SRC_IN));
canvas.drawBitmap(bitmap, src, dst, paint);
return output;
```

使用 Canvas 类，调用 canvas.drawCircle 函数绘制一个圆形区域，其中 rectF 表示的是 RectF 对象，roundPx 表示的是在 x 方向上的圆角半径，第三个参数是 y 方向上的圆角半径，paint 表示绘制时所使用的画笔。最后调用 canvas.drawBitmap 函数，将图片裁剪放入到上面绘制的圆形区域中，就可以得到一个圆形的图片了。

5. 使用 RollPagerView 的轮播图：

在 activity_main.xml 的文件里，写上一个 rollPagerView 控件。在 MainActivity.java 中，获取该 RollPagerView，设置每一张图的播放时间，以及设配器等参数。该设配器是自己定义的一个继承自 StaticPagerAdapter 的类，里面定义了轮播的所有图片，重写了 getView 方法和 getCount 方法。getView 就是每次切换图片时，都会从这里获取新的图片，这个方法里面，设置了图片的 id，图片的 scaleType，以及图片的布局参数。

```
//关于轮播图的设置
private class TestNormalAdapter extends StaticPagerAdapter {
    private int[] imgs = {
            R.mipmap.bk1,
            R.mipmap.bk2,
            R.mipmap.bk3
    };
    @Override
    public View getView(ViewGroup container, int position) {
        ImageView view = new ImageView(container.getContext());
        view.setImageResource(imgs[position]);
        view.setScaleType(ImageView.ScaleType.FIT_XY);
        view.setLayoutParams(new ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT));
        return view;
    }
    @Override
    public int getCount(){ return imgs.length; }
}
```

6. 音乐播放器：

定义一个 MusicService，继承自 service 类。除了重写 service 的 onCreate、onStart、onDestroy、onBind 方法外，自定义了类 MusicBinder，函数 isPlay，playMusic，用于跟

activity 绑定时，可直接对 MusicService 里的 mediaPlayer 进行状态查询和控制。然后，在 mainActivity 里，绑定并启动 MusicService：

```
mMusic = (ImageButton) findViewById(R.id.music);
//启动音乐播放器的service
final Intent intentService = new Intent(MainActivity.this,MusicService.class);
ServiceConnection connection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        MusicService.MusicBinder binder = (MusicService.MusicBinder) service;
        musicService = binder.getService();
        startService(intentService);
    }
    @Override
    public void onServiceDisconnected(ComponentName name) {}
};
bindService(intentService,connection, Context.BIND_AUTO_CREATE);
```

在 mainActivity 和 FigureDetails 两个 java 文件里，对控制音乐播放/暂停的按钮注册监听事件。由于音乐可以在任何界面被改变播放状态，导致其他界面的图标没有及时更新，因此，在 mainActiviy.java 的 onActivityResult 函数里，加入对 MusicService 中音乐播放器状态的判断，从而来设定图标：

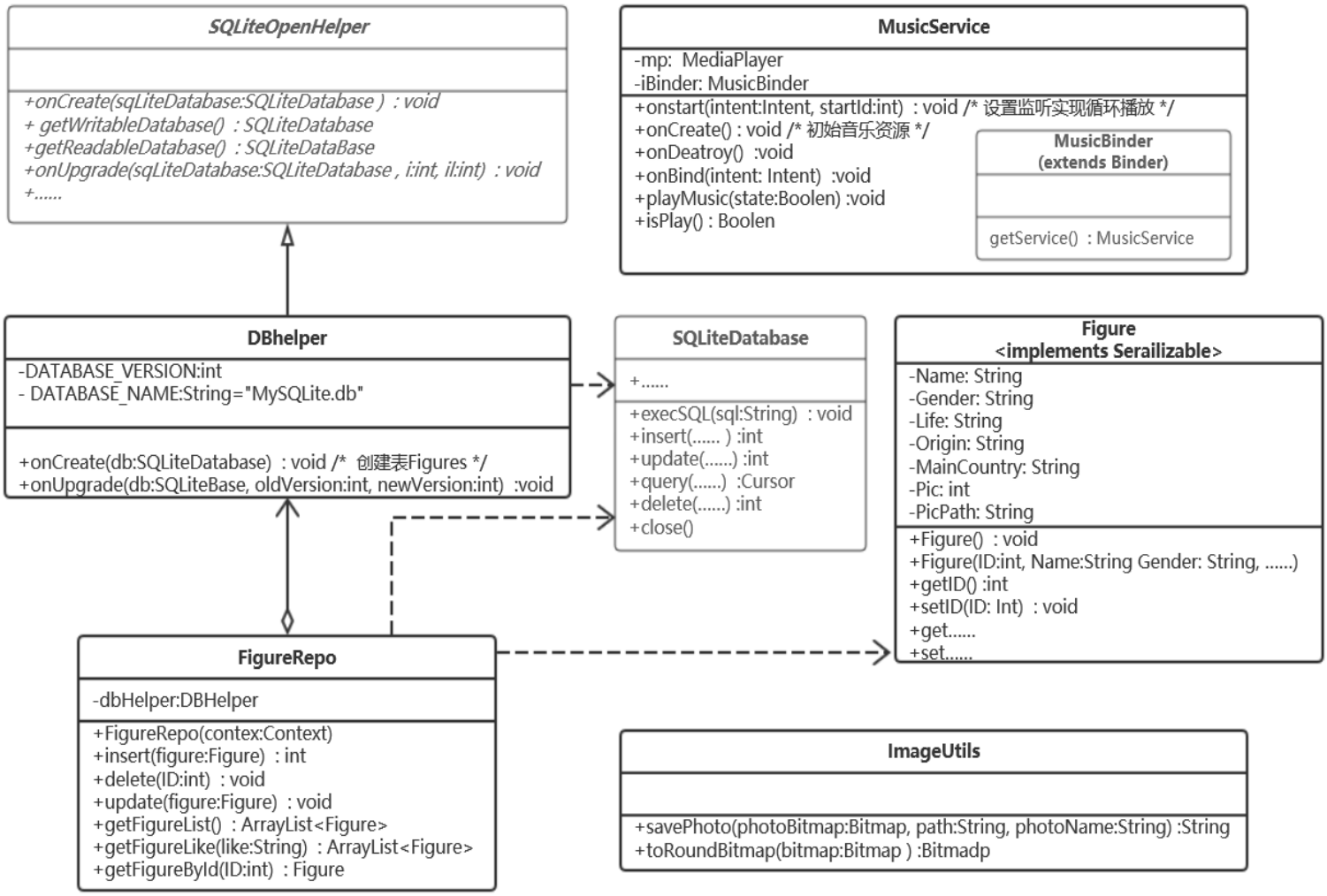
```
//判断音乐播放器的状态，设置图标
if(musicService.isPlay()){
    mMusic.setBackgroundResource(R.mipmap.stop);
} else {
    mMusic.setBackgroundResource(R.mipmap.play);
}
```

而在 FigureDetails.java 中，接收来自 mainActivity 传过来的 intent 中关于音乐播放状态的布尔参数，进而设置图标。

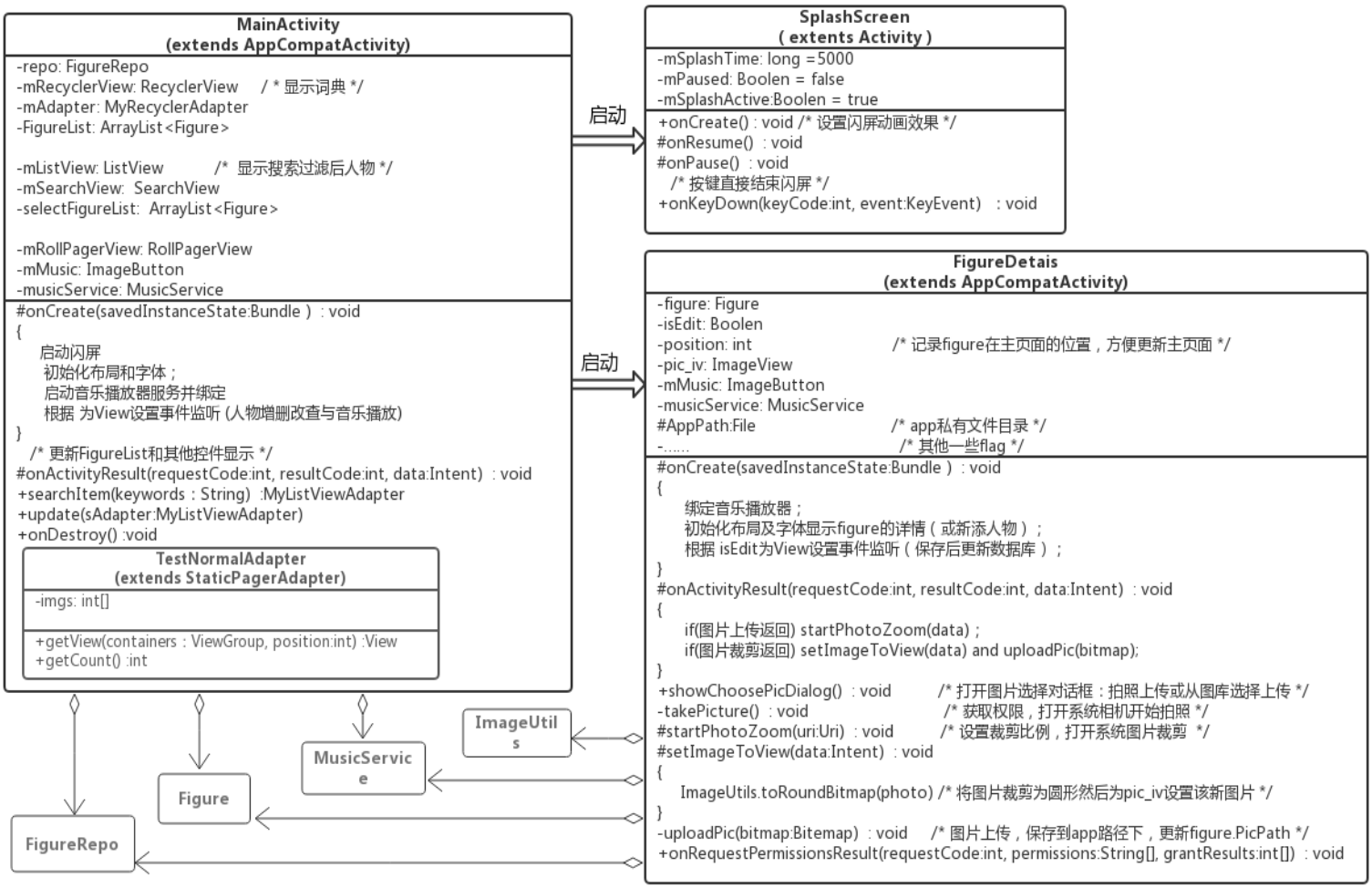
```
//判断音乐的播放状态来设置图标
if(intent.getBooleanExtra("music",false)==false){
    mMusic.setBackgroundResource(R.mipmap.play);
} else if(intent.getBooleanExtra("music",true)==true){
    mMusic.setBackgroundResource(R.mipmap.stop);
}
```

4. 代码架构

功能类(数据库、音乐播放服务、图片处理)实现：



活动类实现（跳转）与关联：



四、应用效果

UI 界面：

图 1 启动时闪屏

图 2 主页面

图三人物详情页

轮播图显示：

图 1 第一张轮播图

图 2 切换到第二张轮播图

图 3 切换到第三张轮播图

功能实现之增加人物功能

图 1 点击红框处的增添按钮

图 2 跳转到人物详情页面进入编辑状态

图 3 点击+按钮进行图片上传

图 4 选择本地照片上传照片

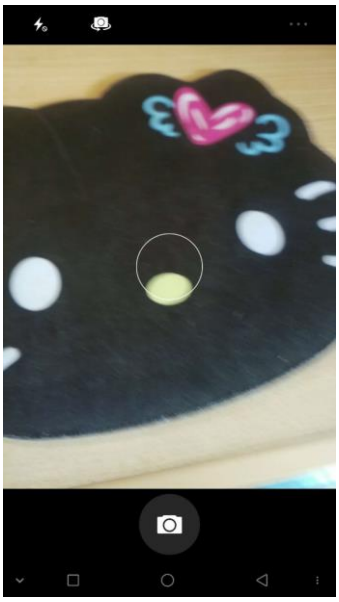


图 5 选择拍照进行拍照

图 6 最终新增人物信息

图 7 主页面更新

图 8 人物详情页更新

功能实现之删除人物功能

图 1 主页面中长按列表选项进行删除

功能实现之查找人物功能

图 1 在搜索框中输入查找信息
点击查找的结果

图 2 跳转到对应的人物详情

功能实现之修改人物信息功能

图 1 点击顶部栏右上角的编辑按钮

图 2 进入编辑状态，出现光标，进行修改

点击顶部栏的 √ 进行保存

图 3 主页面更新

图 4 详情页更新

功能实现之播放音乐功能：

图 1 点击红框处，可播放音乐

图 2 点击红框处，可停止播放音乐

图 3 返回主页面，播放状态与详情页面处状态保持一致

图 4 点击红框处按钮，停止播放

五、开发过程中遇到的问题及解决方法

问题 1：保存人物信息时，主页面未更新。

无论是增加人物还是修改人物信息的时候，当点击了保存按钮后，返回主页面查看人物列表

或者重新回到详情页的时候，人物信息并没有进行更改，需要重新开启 app 后才能看到修改的结果

解决方法：出现上述问题的原因是因为，虽然我们修改了人物信息，但是主页面中存放的人物列表并没有进行更新，因此我们需要在保存人物信息的同时创建一个 intent，存放修改的哪个人物，并在结束当前详情 activity 时，返回这个 intent 和一个 resultCode 为 1,然后在 mainActivity 中调用 notifyDataSetChanged 函数进行更新。

问题 2： 不知道如何将详情页的图片形状变为圆形。

在优化人物详情 UI 界面的时候，为了页面整体上的协调，希望将图片的形状变为椭圆形，然而遇到了不少困难

解决方法：一开始我的想法是通过自定义 ImageView 控件的形状，也上网查找了不少资料，后来在写代码的过程中，发现小组成员的上传照片的代码中有一个函数，能够将图片裁剪为圆形形状的，于是我直接调用了这个函数，就很好地解决了问题。

问题 3：搜索人物时搜索列表无法自动更新。

解决方法：可以使用 SearchView 自带的过滤器，但是这个过滤器在我们输入搜索内容的时候会有黑色的输入提示弹出，非常不美观，而且，使用过滤器来过滤姓名和国家是非常麻烦的，所以自己写了函数去调用数据库的查找功能，并且写了更新查询列表的函数，使得查询列表得以实时更新。

问题 4：搜索框的提示文字显示为黑色，在.xml 文件中也无法修改字体颜色。

解决方法：可以先使用函数获取搜索框的提示文字，并且将提示文字的颜色设置成自己想要的颜色。与设置颜色的方式类似，当我们使用自己导入的字体时，也可以使用这样的方法。

问题 5：数据库的创建。

解决方法：往 Figures 表中插入某一列之后在重新执行，发现查询语句中该列 notFound，但是创建表时，的确有该列。后来某次安装好的 app 发现数据库不是初始的几个人物，而是保留了之前的修改，是因为每次（直接覆盖）安装 app 时，数据库不会重新创建，除非卸载原来的 app。

问题 6： 在 activity 中无法获取音乐播放器的状态。

解决方法：在类 MusicService 中，写一个返回播放状态的函数 isPlay()，如果正在播放，返回 true，否则返回 false。再写一个 public 类型的 palyMusic(boolean)的函数，这样就可以在 activity 中通过调用该函数来直接控制播放器的暂停和播放。

问题 7：音乐播放器的控制按钮无法实现不同 activity 都能同步。

解决方法：在 mainActivity 的 onActivityResult 函数中加入判断音乐播放器状态的语句，从而对控制按钮的图标进行更新。因为除了打开 APP 时，其他时候进入 mainActivity 都会调用这个函数，因此在这个函数中设置。而每次跳转到详情页面时（搜索进入、增加人物进入、查看人物进入），都在 intent 中 put 进表示音乐播放器状态的变量，从而实现不同 activity 的音乐播放控制图标同步。

问题 8：点击自定义的返回按钮时可以正常返回，而手机自带的按钮则不行。

解决方法：在人物详情页面跳转到主页面时，原先是把要传输的 intent 放到自定义的返回按

钮的监听事件中。而实际上是，这个函数只需要负责结束当前 activity 就可以了，于是把传输 intent 的代码挪到保存按钮的监听事件中。这样子，无论是按了自定义的返回按钮，还是手机的返回按钮，都可以正常传输（如果有修改则需要传输）且结束当前 activity 回到主页面。

问题 9：增加人物时，复用修改信息的代码出现了冲突。
解决方法：进入详情页面时，增加一个判断，用来判断是从主页列表点击进来的，还是点击增加按钮进来的。如果是后者，那么页面设置为可编辑状态。点击保存按钮时，也要进行判断，如果是后者，那么把已经编辑好信息的新的人物插入数据库，拿到插入时的位置，如果是前者，则直接更新数据库的内容。

六、思考及感想

曾玉莹

这是我们第一次用 Github 完成整个项目，因为还不是特别熟悉 Github 的使用，所以 commit 和 push 的时候并不是很规范，不过经过这一次之后，真切体会到了 Github 的便利之处，希望下次做项目可以更好地使用它。

实现应用的功能遇到问题时可以在网上查阅资料解决，但是 UI 的设计真的很考验审美以及找图片的能力。本次项目的大部分图片都是我在堆糖网站上一遍一遍筛选下来的，非常遗憾没能找到三国人物的水墨图，要设计出让自己和别人满意的应用真的不容易，也庆幸队友之间总是可以不断提出建议，才可以让我们的应用慢慢变好。虽然它不完美，但是也算是我们宝贵的心血了。

蔡丽芝

在实现修改功能的过程中，学习到了如何更好地使用 EditView 控件，通过设置控件的 focusable 属性可以控制 EditView 的编辑状态，当 focusable 属性为 true 的时候，此时的 editView 控件可进行编辑，因此就可以通过给一个编辑按钮添加点击事件，当点击编辑按钮的时候就改变 editView 的 fousable 值，从而控制控件是否可以编辑，总体上来说，修改和增添的代码基本是一致，只是调用数据库的接口不同而已。在开发的过程中，首次使用 github 进行团队的合作开发，一开始由于对 github 使用不熟悉等各种原因，出现了不少小问题，当然合作的过程中组员之间也偶尔会出现意见不合的小问题，不过后来通过沟通和协调都解决了这些问题。

曾何萌

这次项目让我体会到了团队的力量，只要分工明确，实时沟通，有 bug 一起 de，没什么问题是解决不了的，做出来很美观的 app。“三国说”的功能很简单，主要是数据库的 CRUD 和界面的跳转与显示更新，但实现的过程中因为考虑不周多次出现问题，所以以后开发项目的时候，一定要先建立好一个框架（类似与结构图或者更详细的代码框架），理清思路然后在动手。最后，UI 真的很重要，队友真的很严格。

蔡烨

这是第一次使用 GitHub 进行团队的工程开发，一开始有许多不懂的，觉得很难用，甚至出现了 pull 之后自己写的代码不见了的情况。熟悉使用方法之后，明显感受到这种工作方

式的好处，多人协作，方便、快捷、准确。

其次就是对 activity 之间的信息传输，activity 和 service 之间的信息传输都有了进一步的了解。比如 activity 之间通过 intent 来传输，在对方的 onCreate 函数进行接收，而回来后是在自己的 onActivityResult 函数进行信息更新，诸如此类。

再者就是进一步学会了科学使用搜索引擎，遇到的 bug、不懂的知识点，都可以在网上查找，跟你遇到一样情况的大有人在，因此很多时候通过这种方式成功 debug（比如在 musicService 的注册中去掉 android:process=":remote"才能正常运行）。甚至是一个全新的没碰到过的知识点，也可以在网上的教程学会（比如轮播图的实现）。网络资源如此强大，确实要学会好好利用。

团队协作难免会出现意见不和，但沟通能够解决很多问题。虽然我们使用 GitHub，但是我们还是尽可能保持线下联系，聚在一起写代码，遇到问题及时讨论、解决，效率更高。感谢队友，一个项目也让我收获很多。

七、小组分工

学号	姓名	分工
15352010	蔡烨	UI，人物添加，音乐播放器，轮播图，实验报告
15352006	蔡丽芝	UI，人物信息修改，图片裁剪，实验报告
15352019	曾何萌	数据库实现，图片上传，实验报告
15352024	曾玉莹	UI，人物搜索及删除，闪屏，实验报告

八、参考资料

- Android Studio 下 SQLite 的配置与使用：
  https://www.cnblogs.com/icyhusky/p/5959535.html
- UML 的 9 种图例分析：
  http://blog.csdn.net/fatherican/article/details/44966891
- Android 文件系统说明与图片存储：
  http://blog.csdn.net/mrlixirong/article/details/6800585
  http://blog.csdn.net/ydxlt/article/details/48024017
- Android 服务的应用，在 activity 中实现背景阴郁额播放：
  http://blog.csdn.net/imyang2007/article/details/7597040
- Android 闪屏设计：
  http://www.cnblogs.com/loveliying/p/3144574.html
- Android UI 设计——闪屏效果：
  http://blog.csdn.net/haogaoming123/article/details/8837737
- Android Studio 中添加自定义字体的方法：
  http://blog.csdn.net/ldld1717/article/details/51983997

- 图片来源：堆糖
  https://www.duitang.com/
- 图标来源：iconfont
  http://www.iconfont.cn/

附项目 GitHub 地址：
https://github.com/zyy-7/CCZZ