



中南大學

CENTRAL SOUTH UNIVERSITY

# 本科毕业设计(论文) 调研报告

题    目: Android 多媒体在线管理  
          系统的设计

---

学生姓名: 李钊伟

---

指导教师: 彭春华

---

学    院: 信息科学与工程学院

---

专业班级: 通信工程 1301 班

---

本科生院制

2017 年 3 月

# Android 多媒体在线管理系统的设计

## 摘要

计算机、网络 and 多媒体等技术的广泛应用，给人们带来了内容形象生动的信息，使得其内容的展现更加丰富多彩的同时，也为多媒体资源的管理带来了困难，多媒体资源的数量庞大使得管理日益成为一个严峻的问题。本文针对毕业设计题目-Android 多媒体在线管理系统的现有产品情况和技术实现方案进行了调研，对当前市场环境和同类产品的发展状况进行调查，对此系统业务功能和技术实现的部分内容在文中进行详细阐述，此外，在系统的技术开发方面，对现有的 Web 应用程序(基于 JavaEE 开发)和移动客户端应用程序(基于 Android 开发)之间的技术方案进行探讨与优缺点比较，分析可行性，针对现有系统需求有选择性地选取特定的实现方法，最终总结归纳出此毕业设计的最优的具体技术实现方案。

**关键词：**Android Web 应用 多媒体

## 目录

前言.....	1
第一章 设计任务和研究背景及意义.....	2
1.1 设计任务 .....	2
1.2 研究背景 .....	2
1.3 国内外研究现状 .....	2
1.4 研究意义 .....	3
第二章 多媒体管理系统 Web 端开发.....	4
2.1 J2EE 的 Web 应用开发.....	4
2.2 MVC 框架模式.....	5
2.3 SSM 框架.....	6
2.4 Web 端需求内容.....	8
第三章 多媒体管理系统 Android 端开发.....	9
3.1 Activity .....	9
3.2 Service .....	11
3.3 BroadcastReceiver .....	12
3.4 MVP 框架模式.....	13
3.5 Android 端需求内容.....	14
第四章 结论.....	15
结束语.....	16
参考文献.....	17

## 前言

如今,互联网已经越来越多的介入人们的生活之中。早期的互联网只是一些简单文本的相互传输,用于科学家之间的信息交流。随着互联网的发展,互联网应用于越来越多的领域,人们对互联网也有着越来越多的要求,如信息检索,电子商务,信息交流,网上检索等等。人们已经不满足于以往的文字信息、静态信息的浏览,而是更希望能在在线或本地欣赏到视频、音频大量的多媒体资源。

Web 技术的出现和 Internet 应用的迅速普及为我们解决以上需求指出了一个新的方向。基于浏览器 / 服务器 (Browser/Server) 架构的 Web 技术中, Web 浏览器通过对超文本标注语言的解释执行可以有效地支持多种媒体; 通过图像与文本相结合的超链接技术可以为用户提供一个统一、友好的人机交互界面; 多媒体信息资源集中存放于 Web 服务器中, 分布在不同地域的客户端 Web 浏览器均可以通过 Internet 连接访问这些信息, 从而有效地实现了资源共享。可以说, 基于 Web 的多媒体管理信息系统将取代传统的管理信息系统而成为信息技术领域的一个新的研究和应用方向。

Android 是一种基于 Linux 的自由及开放源代码的操作系统, 主要使用于移动设备, 如智能手机和平板电脑, 由 Google 公司和开放手机联盟领导及开发。Android 操作系统的出现, 使得人类进入移动互联网时代, 通过手机智能终端, 可以轻松方便地浏览互联网上的资源与信息。

在此多媒体在线管理系统中, 管理员可通过 Web 应用对服务器上的音视频多媒体资源进行增添、修改、删除等操作, 并在 Android 智能终端处也可进行同样的操作, 二者相结合, 提高用户体验, 让用户可以更方便, 更快捷地维护多媒体管理系统中的资源, 此外, 在 Android 端, 用户可以市场上的同类软件产品那样, 选择下载至本地或在线播放多媒体管理系统中的音视频资源。

## 第一章 设计任务和研究背景及意义

### 1.1 设计任务

- (1) 通过毕业设计，培养独立查阅文献资料 and 进行科研工作的能力；
- (2) 查阅 Android 的有关资料，了解 Android 的发展现状及趋势，并熟悉和掌握 Android 和 Java 编程，完成课题设计调研报告；
- (3) 对多媒体管理系统进行需求分析，分解各模块具体的功能；
- (4) 熟悉 JavaEE Web 开发，编写前端 Html 页面、CSS 样式、JS 脚本，实现各模块功能页面，界面要求友好、交互、生动、形象、操作方便；
- (5) 开发 Web 后端不同模块的数据接口，完成和前台的交互协议，供 Android 客户端和 Web 前端调用。
- (6) 开发 Android 客户端，实现各模块需求。
- (7) 前后端调试：对管理系统进行前后端联调，进行单元测试，解决漏洞。
- (8) 撰写毕业设计论文。阐述系统设计方案和程序设计思想，将所做的工作进行归纳和总结。

### 1.2 研究背景

多媒体管理系统是建立在多媒体、网络、数据库和数字存储等先进技术基础上的一个对各种媒体及内容(如视/音频资料等)进行数字化存储、管理以及应用的总体解决方案，包括数字媒体的增加、编辑、管理、删除等环节。其主要是满足媒体资源拥有者收集、保存、查找、编辑、发布各种信息的要求，为媒体资源的使用者提供访问内容的便捷方法，实现对媒体资源的高效管理，大幅度提高媒体资源的价值。

### 1.3 国内外研究现状

暴风影音是北京暴风科技有限公司推出的一款多媒体播放器产品，包含 Windows、Android、iOS 版本。该播放器兼容大多数的视频和音频格式。其软件特点有：通过自动侦测用户的硬件配置；自动匹配相应的解码器、渲染链；自动调

整对硬件的支持。它提供和升级了系统对常见绝大多数影音文件和流的支持，包括：RealMedia、QuickTime、MPEG2、MPEG4(ASP/AVC)、VP3/6/7、Indeo、FLV 等流行视频格式；AC3/DTS/LPCM/AAC/OGG/MPC/APE/FLAC/TTA/WV 等流行音频格式；3GP/Matroska/MP4/OGM/PMP/XVD 等媒体封装及字幕支持等。配合 Windows Media Player 最新版本可完成当前大多数流行影音文件、流媒体、影碟等的播放而无需其他任何专用软件。暴风影音采用 NSIS 封装，为标准的 Windows 安装程序，特点是单文件多语种(简体中文 + 英文)，具有稳定灵活的安装、卸载、维护和修复功能，并对集成的解码器组合进行了尽可能的优化和兼容性调整，适合普通的大多数以多媒体欣赏或简单制作为主要使用需求的用户。

美国 FatTail 公司是提供在线广告优化软件的创新公司。FatTail 在 2007 年的时候与国内一家软件公司阿丁特合作开发了一套多媒体资源管理系统 Admanager<sup>[1]</sup>。该系统在美国倍受欢迎，FatTail 凭借该软件在美国获得 AlwaysOn OnMedia 100 大奖，获得了不菲的奖金。有了国外的成功案例后，阿丁特后来又根据国内媒体行业的特自研发了一套适合国内媒体资源管理系统 Clifford，并成功上线为用户所使用。

#### 1.4 研究意义

近些年来，多媒体行业有着快速的发展，看着大街上形形色色的广告方案以及多种多样的广告宣传方式便可清楚地知道这点。随着行业发展的迅速，那么竞争也就越来越激烈。就国内来说，在行业巨头的带领下出现了一大批的中小型媒体公司，这些媒体公司为了与巨头竞争，也会收购比自己小的媒体公司，并积极开发出自己的新的媒体，竞争过程也正如大鱼吃小鱼，小鱼吃虾米一般。媒体巨头因为有着强大的经济实力作为支撑，通常会创建一个研发团队，自研发一套适合自身使用的媒体资源管理系统，来管理所拥有的众多媒体资源，从而提高工作效率，满足客户需求来达到提升效益的目的。

## 第二章 多媒体管理系统 Web 端开发

Web 应用程序是一种可以通过 Web 访问的应用程序。Web 应用程序的一个最大好处是用户很容易访问应用程序。用户只需要有浏览器即可，不需要再安装其他软件。应用程序有两种模式 C/S 模式和 B/S 模式。C/S 是客户端/服务器端程序，也就是说这类程序一般独立运行。而 B/S 就是浏览器端/服务器端应用程序，这类应用程序一般借助 Chrome、IE 等浏览器来运行。WEB 应用程序一般是 B/S 模式。Web 应用程序首先是“应用程序”，和用标准的程序语言，如 C、C++ 等编写出来的程序没有什么本质上的不同。然而 Web 应用程序又有自己独特的地方，就是它是基于 Web 的，而不是采用传统方法运行的。换句话说，它是典型的浏览器/服务器架构的产物。

一个 Web 应用程序是由完成特定任务的各种 Web 组件 (web components) 构成的并通过 Web 服务器将服务展示给外界。在实际 JavaEE 应用的 Servlet 开发方式中，Web 应用程序是由多个 Servlet、JSP 页面、HTML 文件以及图像文件等组成。所有这些组件相互协调为用户提供一组完整的服务。

### 2.1 J2EE 的 Web 应用开发

本毕业设计基于 J2EE 开发，J2EE (Java 2 Platform, Enterprise Edition) 是一个为大企业主机级的计算类型而设计的 Java 平台。Sun 微系统 (与其工业伙伴一起，例如 IBM) 设计了 J2EE，以此来简化在客户级环境下的应用开发。由于创造了标准的可重用模块组件以及由于构建出能自动处理编程中多方面问题的等级结构，J2EE 简化了应用程序的开发，让开发人员更专注与业务逻辑实现，提高开发效率。

J2EE 平台由一整套服务 (SERVICES)、应用程序接口 (APIS) 和协议构成，它对开发基于 WEB 的多层应用提供了功能支持。

J2EE 使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据他们所在的层分布在不同的机器上。事实上，sun 设计 J2EE 的初衷正是为了解决两层模式 (client/server) 的弊端，在传统模式中，客户端担当了过多的角色而显得臃肿，在这种模式中，第一次部署的时候比较容易，但难于升级或



改进，可伸展性也不理想，而且经常基于某种专有的协议，通常是某种数据库协议。它使得重用业务逻辑和界面逻辑非常困难。现在 J2EE 的多层企业级应用模型将两层化模型中的不同层面切分成许多层。一个多层化应用能够为不同的每种服务提供一个独立的层，降低耦合。

## 2.2 MVC 框架模式

本毕业设计 Web 后端使用 MVC 框架模式，MVC 全名是 Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

MVC 是一种使用 MVC (Model View Controller 模型-视图-控制器) 设计创建 Web 应用程序的模式：

- (1)Model（模型）表示应用程序核心（比如数据库记录列表）。
- (2)View（视图）显示数据（数据库记录）。
- (3)Controller（控制器）处理输入（写入数据库记录）。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

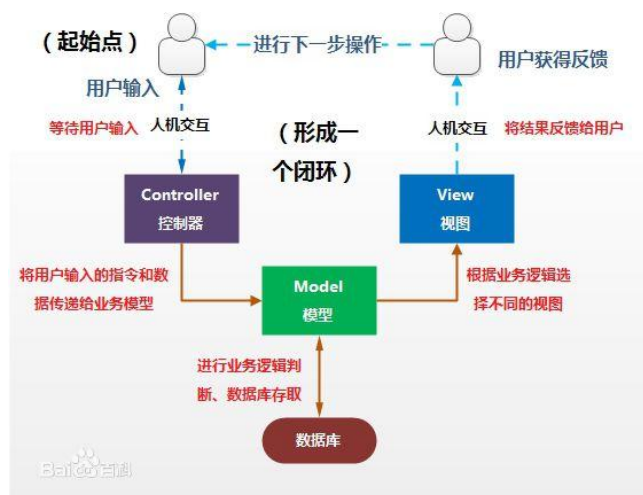


图 2-1 MVC 框架模式结构图



MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。

### 2.3 SSM 框架

本毕业设计使用SSM框架进行开发，SSM即Spring、SpringMVC、MyBatis三个开源框架的整合框架集。

Spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。其特点如下：

#### (1) 轻量

从大小与开销两方面而言Spring都是轻量的。完整的Spring框架可以在一个大小只有1MB多的JAR文件里发布。并且Spring所需的处理开销也是微不足道的。此外，Spring是非侵入式的：典型地，Spring应用中的对象不依赖于Spring的特定类。

#### (2) 控制反转

Spring通过一种控制反转（IoC）的技术促进了低耦合。当应用了IoC，一个对象依赖的其它对象会通过被动的方式传递进来，而不是这个对象自己创建或者查找依赖对象。你可以认为IoC与JNDI相反——不是对象从容器中查找依赖，而是容器在对象初始化时不等对象请求就主动将依赖传递给它。

#### (3) 面向切面

Spring提供了面向切面编程的丰富支持，允许通过分离应用的业务逻辑与系统级服务（例如审计（auditing）和事务（transaction）管理）进行内聚性的开发。应用对象只实现它们应该做的——完成业务逻辑——仅此而已。它们并不负责（甚至是意识）其它的系统级关注点，例如日志或事务支持。

#### (4) 容器

Spring包含并管理应用对象的配置和生命周期，在这个意义上它是一种容器，你可以配置你的每个bean如何被创建——基于一个可配置原型（prototype），你的bean可以创建一个单独的实例或者每次需要时都生成一个新的实例——以及它们是如何相互关联的。然而，Spring不应该被混同于传统的重量级的EJB容器，它们经常是庞大与笨重的，难以使用。

SpringMVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。Spring MVC的作用是当客户端发送请求，服务器控制器（由DispatcherServlet实现的）完成请求的转发，控制器调用一个用于映射的类HandlerMapping，该类用于将请求映射到对应的处理器来处理请求。

HandlerMapping 将请求映射到对应的处理器Controller（相当于Action）在Spring 当中如果写一些处理器组件，一般实现Controller 接口，在Controller 中就可以调用一些Service 或DAO 来进行数据操作 ModelAndView 用于存放从DAO 中取出的数据，还可以存放响应视图的一些数据。如果想将处理结果返回给用户，那么在Spring 框架中还提供一个视图组件ViewResolver，该组件根据Controller 返回的标示，找到对应的视图，将响应response 返回给用户。

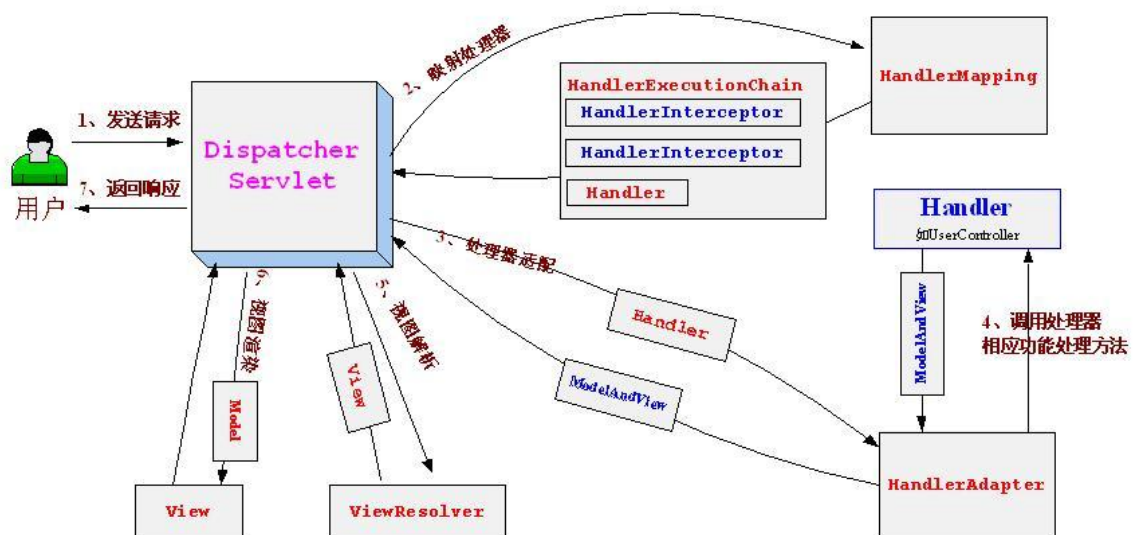


图 2-2 Spring MVC 框架结构图

MyBatis是一个支持普通SQL查询，存储过程和高级映射的优秀持久层框架。它消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML或注解用于配置和原始映射，将接口和 Java 的POJOs (Plain Old Java Objects, 普通的Java对象) 映射成数据库中的记录。每个MyBatis应用程序主要都是使用SqlSessionFactory实例的，一个SqlSessionFactory实例可以通过SqlSessionFactoryBuilder获得。SqlSessionFactoryBuilder可以从一个xml 配置文件或者一个预定义的配置类的实例获得。

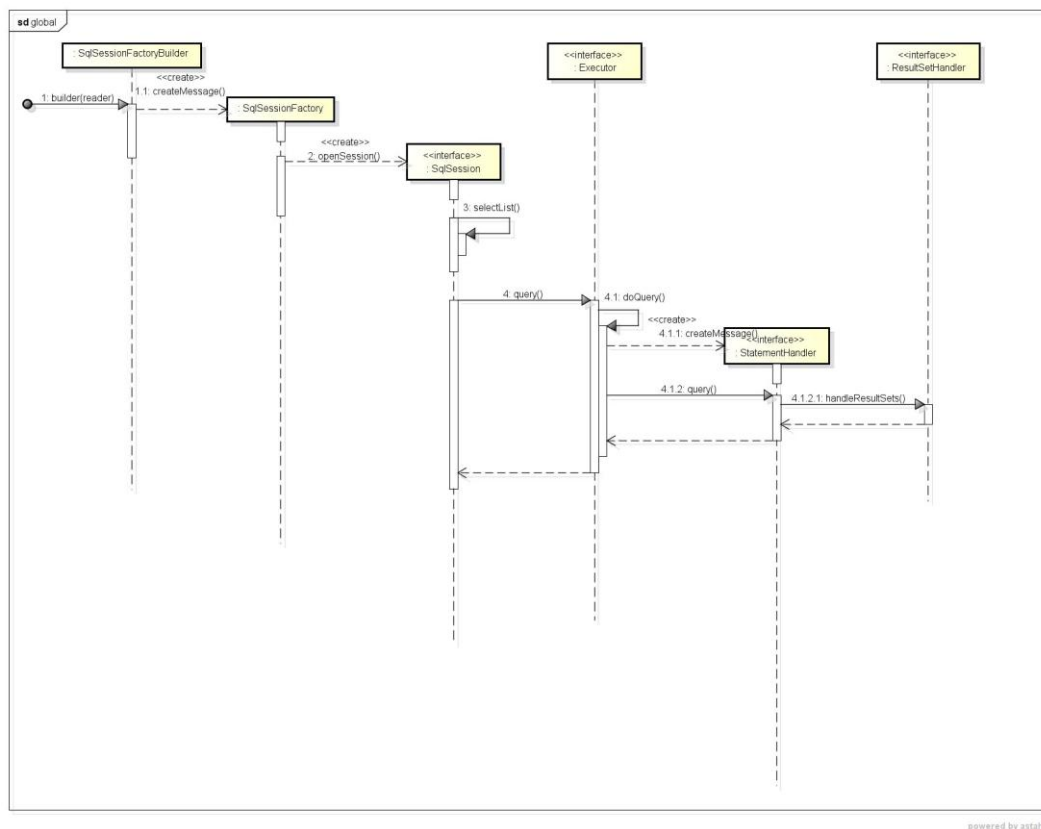


图 2-3 Mybaitis 框架执行流程图

## 2.4 Web 端需求内容

在 Web 端，本多媒体管理系统暂定的主要功能和需求如下：

### (1) 用户权限控制

系统用户分为两类，即普通用户和管理员，管理员具有更高权限。普通用户可以对服务器上的音视频资源文件进行查询、浏览等操作。管理员除此以外可以对服务器上的音视频资源进行增删、修改、分类等操作。

### (2) 音视频资源维护

系统可分类显示服务器上的音视频资源列表，可对每一个音频或视频文件的信息进行修改维护，管理员可以对服务器上的音视频文件进行的增加、删除等。

### (3) 音视频等大文件上传

系统应提供界面友好的上传页面，供管理员使用，同时对于视频等大文件的上传采用合适的网络协议，优化的用户体验。

### (4) Android 端所需接口实现及令牌鉴权

系统应实现 Android 客户端所需要的接口，提供数据。同时通过服务端下发 token 令牌来实现客户端业务的请求鉴权。

## 第三章 多媒体管理系统 Android 端开发

### 3.1 Activity

Activity 是一个应用程序组件<sup>[2]</sup>，提供一个屏幕，用户可以用来交互为了完成某项任务，它上面可以显示一些控件也可以监听并处理用户的事件做出响应。

在 Android 中，Activity 拥有四种基本状态：

#### (1)Active/Running

一个新 Activity 启动入栈后，它显示在屏幕最前端，处理是处于栈的最顶端（Activity 栈顶），此时它处于可见并可和用户交互的激活状态，叫做活动状态或者运行状态（active or running）。

#### (2)Paused

当 Activity 失去焦点，被一个新的非全屏的 Activity 或者一个透明的 Activity 被放置在栈顶，此时的状态叫做暂停状态（Paused）。此时它依然与窗口管理器保持连接，Activity 依然保持活力（保持所有的状态，成员信息，和窗口管理器保持连接），但是在系统内存极端低下的时候将被强行终止掉。所以它仍然可见，但已经失去了焦点故不可与用户进行交互。

#### (3)Stopped

如果一个 Activity 被另外的 Activity 完全覆盖掉，叫做停止状态（Stopped）。它依然保持所有状态和成员信息，但是它不再可见，所以它的窗口被隐藏，当系统内存需要被用在其他地方的时候，Stopped 的 Activity 将被强行终止掉。

#### (4)Killed

如果一个 Activity 是 Paused 或者 Stopped 状态，系统可以将该 Activity 从内存中删除，Android 系统采用两种方式进行删除，要么要求该 Activity 结束，要么直接终止它的进程。当该 Activity 再次显示给用户时，它必须重新开始和重置前面的状态。

当一个 Activity 实例被创建、销毁或者启动另外一个 Activity 时，它在这四种状态之间进行转换，这种转换的发生依赖于用户程序的动作。下图说明了

Activity 在不同状态间转换的时机和条件:

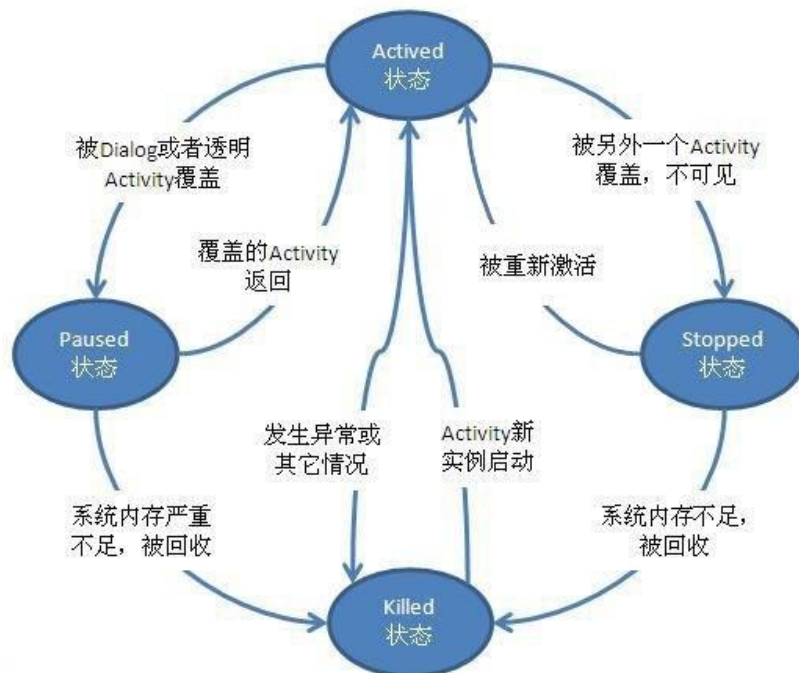


图 3-1 Activity 状态转移流程图

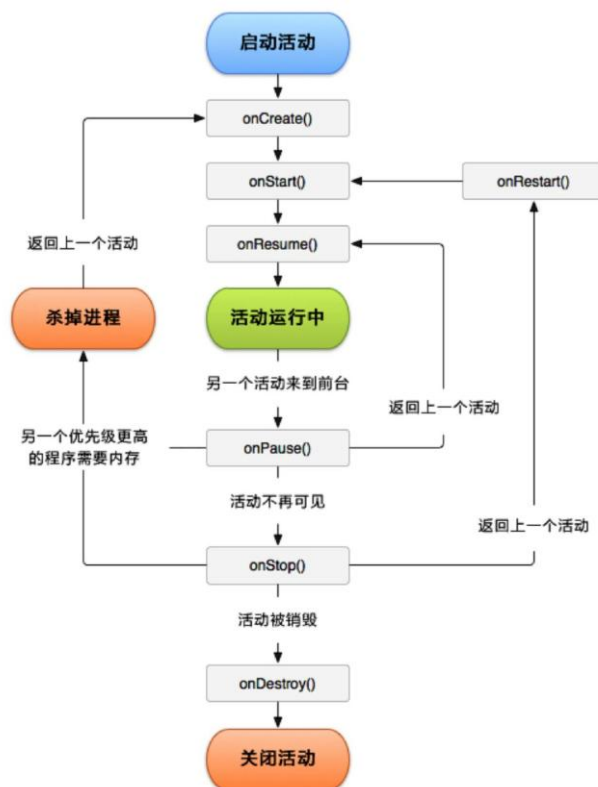


图 3-2 Activity 生命周期流程图

开发过程中，主要所关注的即为 Activity 生命周期的中的方法，通过继承 Activity 基类，重写 onCreate() 等方法，实现自己项目所需的业务逻辑，从而实现具体的需求。

在本项目中，Andoird 端的每一个用户界面，主要都通过 Activity 来开发与用户进行交互。

### 3.2 Service

Service 是一种长生命周期的，没有可视化界面，运行于后台的一种服务程序。它主要用于在后台处理一些耗时的逻辑，或者去执行某些需要长期运行的任务。必要的时候我们甚至可以在程序退出的情况下，让 Service 在后台继续保持运行状态。

Service 的生命周期，从它被创建开始，到它被销毁为止，可以有两条不同的路径, 如下图所示：

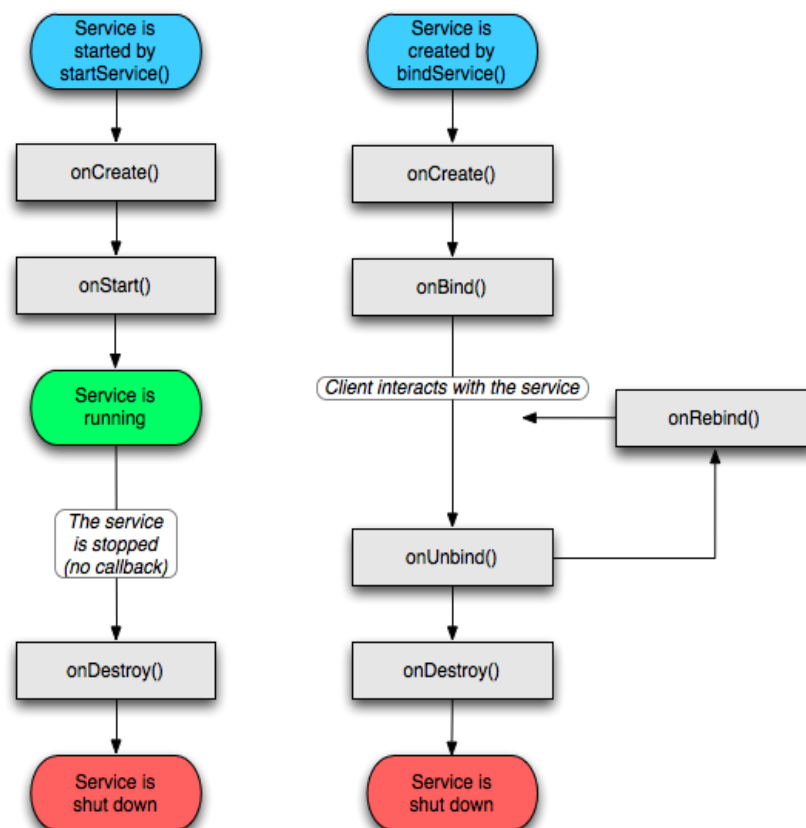


图 3-3 Service 生命周期流程图

### (1) A started service

被开启的 service 通过其他组件调用 `startService()` 被创建。这种 service 可以无限地运行下去，必须调用 `stopSelf()` 方法或者其他组件调用 `stopService()` 方法来停止它。当 service 被停止时，系统会销毁它。

(2) A bound service 被绑定的 service 是当其他组件（一个客户）调用 `bindService()` 来创建的。客户可以通过一个 `IBinder` 接口和 service 进行通信。客户可以通过 `unbindService()` 方法来关闭这种连接。一个 service 可以同时和多个客户绑定，当多个客户都解除绑定之后，系统会销毁 service。

在本项目中，一些需要耗时并且需要常驻后台的逻辑功能代码，需要通过 Service 来实现，如播放音频的过程，如果不通过 Service 实现，那么当播放页面的 Activity 被系统杀死后，或者用户返回上一级 Activity 后，音乐播放即会停止，产生不好的用户体验，不完善的播放功能。

## 3.3 BroadcastReceiver

BroadcastReceiver 即 Android 当中的广播接收者，用来实现组件当中相互的一对多通信。

Android 广播：系统在产生某个事件时发送广播，应用程序使用广播接收者接收这个广播，就知道系统产生了什么事件。Android 系统在运行的过程中，会产生很多事件，比如开机、电量改变、收发短信、拨打电话、屏幕解锁。

### 广播的两种类型

(1) 无序广播：所有跟广播的 intent 匹配的广播接收者都可以收到该广播，并且是没有先后顺序（同时收到）。

(2) 有序广播：所有跟广播的 intent 匹配的广播接收者都可以收到该广播，但是会按照广播接收者的优先级来决定接收的先后顺序。

最终接收者：所有广播接收者都接收到广播之后，它才接收，并且一定会接收。

使用广播接收者 (BroadcastReceiver) 时需要注册，在 Android 中，有两种注册方式，一种方式是在对应的 Activity 的 Java 代码中注册，另一种是在 xml 文件中注册。



在本项目中，当音频在后台 Service 中播放的时候，每一秒获取当前歌曲播放的进度，并通过 Service 中发送广播和 Activity 中注册 BroadcastReceiver 来实现组件中的通信，从而每秒在 Activity 的用户界面上更新音频播放进度条，当用户拖拽了 Activity 中的进度条时，在 Activity 中发送广播给 Service，Service 接收到后，更新用户拖拽的进度值，并设为当前音频播放的进度值，从而实现拖拽音频进度播放的功能。

### 3.4 MVP 框架模式

随着 UI 创建技术的功能日益增强，UI 层也履行着越来越多的职责。为了更好地细分视图(View)与模型(Model)的功能，让 View 专注于处理数据的可视化以及用户的交互，同时让 Model 只关系数据的处理，基于 MVC 概念的 MVP(Model-View-Presenter)模式应运而生。

在 MVP 模式里通常包含 4 个要素：

(1)View:负责绘制 UI 元素、与用户进行交互(在 Android 中体现为 Activity)。

(2)View interface:需要 View 实现的接口，View 通过 View interface 与 Presenter 进行交互，降低耦合，方便进行单元测试。

(3)Model:负责存储、检索、操纵数据。

(4)Presenter:作为 View 与 Model 交互的中间纽带，处理与用户交互的负责逻辑。

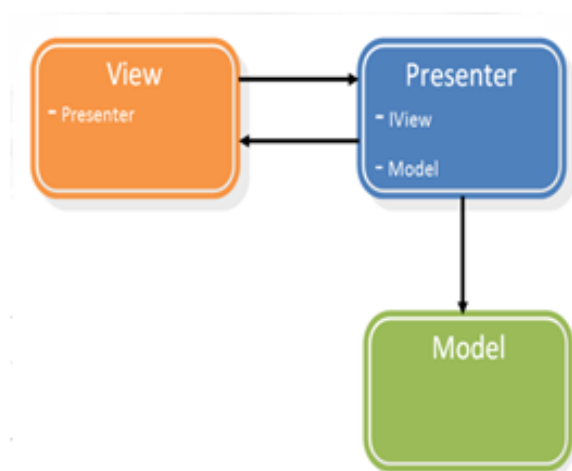


图 3-4 MVP 框架结构图

在 Android 开发中, Activity 并不是一个标准的 MVC 模式中的 Controller, 它的首要职责是加载应用的布局和初始化用户界面, 并接受并处理来自用户的操作请求, 进而作出响应。随着界面及其逻辑的复杂度不断提升, Activity 类的职责不断增加, 以致变得庞大臃肿。当我们将其中复杂的逻辑处理移至另外的一个类 (Presenter) 中时, Activity 其实就是 MVP 模式中 View, 它负责 UI 元素的初始化, 建立 UI 元素与 Presenter 的关联 (Listener 之类), 同时自己也会处理一些简单的逻辑 (复杂的逻辑交由 Presenter 处理)。

### 3.5 Android 端需求内容

在 Android 端, 本多媒体管理系统暂定的主要功能和需求如下:

#### (1) 音视频文件资源列表查询

通过 Web 后端提供的 API 接口, 发起 Http 请求, 获取返回的数据并解析, 以友好的用户界面呈现。

#### (2) 音视频文件播放

对于音频文件, 包含本地音频播放、在线音频播放 (边下边播)、暂停、停止、上一曲、下一曲、歌曲封面展示、音量调节等; 对于视频文件、包含本地视频播放、在线视频播放 (变下边播)、暂停、停止。

#### (3) 音视频文件下载

客户端应实现音视频文件的下载功能, 并提供友好的用户界面进行下载进度展示。

#### (4) 权限控制

客户端通过向服务端发送本地令牌 token 鉴权, 对于管理员, 可以在客户端对音视频资源服务器进行增加、删除等文件维护操作。对于普通用户, 可以再客户端进行播放、下载、查询等操作。

## 第四章 结论

对于本毕业设计-Android 多媒体在线管理系统, 针对现有的需求功能, Web 后端拟采用基于 JavaEE 中的 Spring MVC<sup>[3]</sup>、Spring<sup>[4]</sup>、Mybatis 框架<sup>[5]</sup>进行开发, 采用 MVC 框架模式, 通过 AOP(面向切面编程)、DI(依赖注入)等方法减少层间耦合, 利于后续扩展和代码维护, 通过三级缓存、SQL 调优等方法提高数据访问性能, 实现各模块业务逻辑, 为前端提供 API 数据接口, 创建出高性能、高内聚、低耦合的 Web 应用。Web 前端需要实现友好的网页界面, 可使用 Bootstrap、jQuery、Vue 等组件开发, 编写 HTML、CSS、JS 代码, 调用 Web 后端提供的数据 API 接口, 更好地在网页前端实现与用户的交互。Android 客户端使用 MVP 框架<sup>[6]</sup>模式, 结合 Android 系统提供的组件 Activity、Service、BroadcastReceiver 实现用户界面以及用户交互<sup>[7]</sup>, 使用 Web 后端提供的 API 接口进行业务逻辑请求, 实现项目需求内容。

## 结束语

本调研报告介绍了本毕业设计项目的研究意义与市场情况,通过对项目进行需求分析,以及技术方案选型与介绍,得出了针对本项目的最终解决方案,接下来将会根据上述技术方案进一步编写程序代码,实现项目功能,更好地完成此毕业设计项目。

## 参考文献

- [1] Darrell Woelk. Multimedia Information Management in an Object-Oriented Database System [EB/OL]. <http://vldb.org/conf/1987/P319.PDF>, 1987.
- [2] 郭霖. 第一行代码——Android[M]. 人民邮电出版社, 2014.
- [3] 计文柯. Spring技术内幕[M]. 机械工业出版社, 2012.
- [4] Marten Deinum, Koen Serneels, Colin Yates, Seth Ladd, Christophe Vanfleteren. Pro Spring MVC[M]. Apress, 2012.
- [5] Clinton Begin, Brandon Goodin, Larry Meadors. iBATIS 实战[M]. 人民邮电出版社, 2008.
- [6] 许瑾. 基于Android平台音乐播放器的设计与实现[D]. 北京邮电大学, 2011.
- [7] 李德平. 基于Android的移动多媒体播放器软件的设计与实现[J]. 河北软件职业技术学院学报, 2014, 16(1):52-56.
- [8] 胡启敏, 薛锦云, 钟林辉. 基于 Spring 框架的轻量级 J2EE 架构与应用[J]. 计算机工程与应用, 2008, 44(5):115-118.
- [9] 荣艳冬. 关于 Mybatis 持久层框架的应用研究[J]. 信息安全与技术, 2015, 6(12):86-88.
- [10] 张锦贤. 基于 bootstrap 框架的 web 设计开发研究[J]. 工程技术:文摘版, 2016(8):00089-00089.