

# 中南民族大学

## 毕业论文(设计)

学院：  计算机科学学院  

专业：  软件工程   年级：  2014  

题目：基于 Android 和 Flask 的最炫  
广场舞 APP 设计与开发

学生姓名：  柳真   学号：  201421092073  

指导教师姓名：  孙翀   职称：  讲师  

2018 年 5 月 26 日

## 中南民族大学本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：                    年    月    日

目 录

摘 要..... 1

Abstract.....1

1 前言..... 2

    1.1 背景介绍.....2

    1.2 系统架构方案.....2

    1.3 系统开发工具及部署环境.....3

2 需求分析..... 3

    2.1 可行性研究..... 3

    2.2 功能需求.....4

        2.2.1 功能划分.....4

        2.2.2 功能描述.....4

    2.3 用例图.....6

        2.3.1 用例模型.....6

        2.3.2 用例描述.....8

    2.4 外部配置及接口需求.....12

        2.4.1 移动端 APP 配置需求..... 12

        2.4.2 软件接口.....12

    2.5 性能需求.....12

        2.5.1 时间特性要求.....12

        2.5.2 灵活性及精度.....13

3 系统设计.....13

    3.1 概要设计.....13

        3.1.1 设计目标.....13

        3.1.2 系统技术框架介绍.....13

        3.1.3 系统技术架构.....15

    3.2 详细设计.....17

        3.2.1 系统主要功能模块设计与实现.....17

3.2.2 系统功能界面设计.....	26
3.2.1 系统后端接口设计.....	34
3.3 数据库（存储）设计.....	39
3.3.1 集合.....	39
3.3.2 文档数据模型.....	40
3.3.3 具体代码映射实现.....	40
4 调试与分析.....	41
4.1 测试准则.....	41
4.2 测试目标.....	42
4.3 测试及结果.....	42
4.3.1 舞群定位和周边舞群模块.....	42
4.3.2 舞蹈视频上传模块.....	42
4.3.3 舞蹈播放源信息及评论信息模块.....	43
4.3.4 时间服务功能模块.....	43
结论.....	43
致谢.....	43
参考文献.....	44

# 基于 Android 和 Flask 的最炫广场舞 APP 设计与开发

**摘要：**广场舞在国内越来越受到广大中老年群体的喜爱，舞群组织不规范导致扰民问题时有发生、舞蹈教学视频缺乏导致广场舞爱好者训练难等问题。为了解决上述问题，设计和研发了最炫广场舞 APP。本系统设计下列功能需求：1) 提供周边舞群功能，规范市民周边舞群活动地点，并且做好登记和管理工作；2) 提供当前热门舞蹈教学视频以及用户视频上传分享功能；3) 提供周边舞具商品推荐功能；4) 提供舞友交流功能。移动端基于 Android 实现，后台管理功能基于 Python 的 Flask 框架实现，数据库采用非关系型数据库 MongoDB 来搭建。系统所需图片、视频等文件信息存放在 Apache 服务器上，后端具体服务和文件存放部署在阿里云服务器上。全文由四部分组成：1) 前言，介绍系统的相关背景；2) 需求分析，介绍系统的具体功能需求模块及性能需求；3) 系统设计，介绍系统的概要设计、详细设计和数据库设计；4) 调试与分析，介绍系统基本功能调试结果。

**关键词：**广场舞；移动应用；安卓；WEB 服务器；非关系数据库

## Design and Development of the most Dazzling Square Dance APP based on Android and Flask

**Abstract:** Majority of middle-aged people are more like to go to square dance. The irregular organization of dance groups leads to frequent disturbances, and many people who like dancing square dance have difficulty participating in training and other issues. In order to solve the above issues, The most dazzling square dance APP has been designed and developed. This system makes the following functional requirements: 1) Provide the function of surrounding dance group to regulate the activities of the surrounding dance groups, and complete registration and do a good job in management; 2) Provide the function of watching popular dance teaching videos, uploading and sharing videos by users; 3) Provide the function of surrounding dance product recommended; 4) Provide the function of communicating with friends who like dancing. Mobile terminal is based on Android implementation, background management functions based on Flask framework of Python, Database is built by MongoDB which is a non-relational database. Images, video, and other file information which are required by system are stored on the Apache server, backend specific services and file storage deployed on Alibaba Cloud server. The full text consists of four

parts:1)Foreword,introduce system related background;2)Demand analysis,describe the specific functional requirements and performance;3)System design,introduce the system's outline design, detailed design and database design;4)Commissioning and analysis,describing the results that basic functions are testd in this system.

**Keywords:** Square Dance;Mobile Applications;Android;WEB Server;Non-relational Database

## 1 前言

### 1.1 背景介绍

互联网行业的高速发展，智能手机产业迭代式更新，日常生活与互联网的联系也变得更加密切。人们的生活方式在不断地发生着重大的变化，由以往的传统通信方式转变为使用手机解决日常问题和通信交流。广场舞在国内逐渐普及，越来越多的广场舞爱好者投入学习大潮，但是与之相应而产生了扰民和广场舞视频教学资源匮乏等问题。市场上相关产品软件鱼龙混杂，并不能很好的解决这些核心问题。针对国内广大广场舞爱好者，以他们的日常训练和交流并结合社会大众对广场舞扰民的看法为基本出发点，作出精准需求设计的软件还很少，或者说还没有达到普及和成熟的标准。因此，最炫广场舞 APP 就孕育而生了。

### 1.2 系统架构方案

客户端和服务端是本系统架构的核心模块。客户端包含基于移动的 Andorid 用户端和基于 Web 的网页管理端，服务端采用基于 Python 的 Flask 框架和非关系型 MongoDB 数据库搭建。系统架构如下图 1-1 所示：

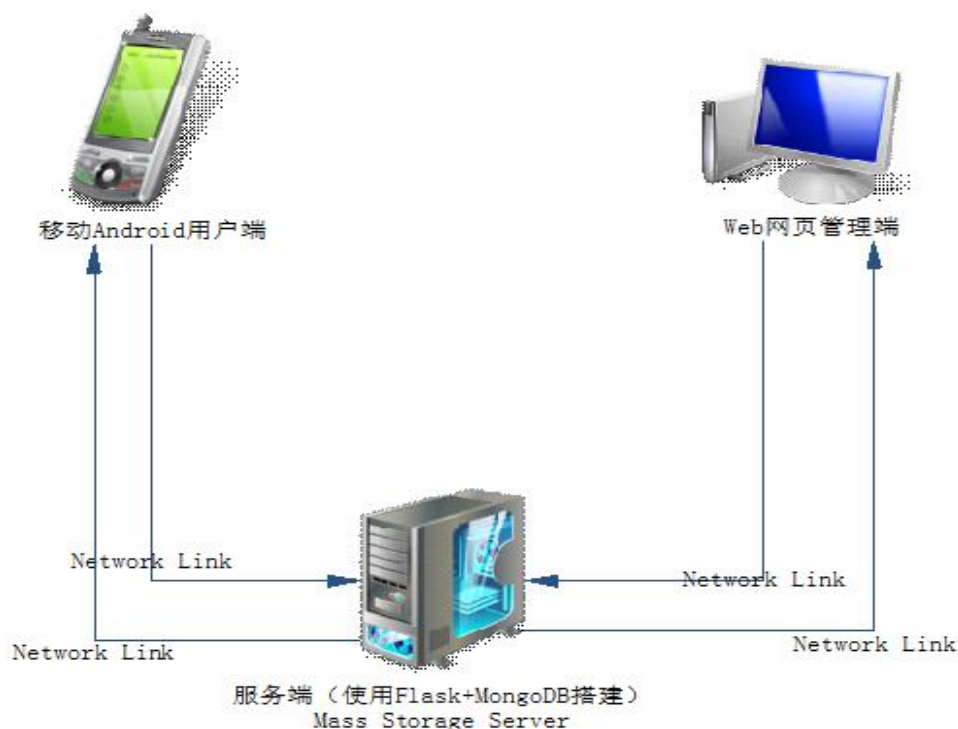


图 1-1 最炫广场舞系统架构方案图

### 1.3 系统开发工具及部署环境

#### (1) 客户端

分为用户端和管理端。

- 用户端采用 Android 实现。开发工具：AndroidStudio；操作系统：Andorid5.0 及以上系统。
- 管理端采用 BootStrap+JavaScript+CSS 实现。开发工具：PyCharm；操作系统：Windows Server 2008 或者更新版本或者 Ubuntu。

#### (2) 服务端

采用 Flask 框架和 MongoDB 数据库实现。开发工具：PyCharm；操作系统：Windows Server 2008 或更新版本。

## 2 需求分析

### 2.1 可行性研究

国内市场已有与广场舞题材相关的 APP，它们的核心功能点都是关于教学视频这一需求。经过实际调查，目前手机应用市场上已上架恰恰广场舞和糖豆广场舞等 APP。

与现有的广场舞 APP 相比，最炫广场舞这款 APP 不仅涵盖了市场已有 APP 的所有功能，并结合实际情形，主推精心设计的核心功能服务模块。国内的已有广场舞 APP 都从广场舞教学视频这个需要入手，不仅结构单一化，而且没有切实的观察过广大广场舞爱好者的其他困难和现实需求。最炫广场舞 APP 是通过走访广大广场舞爱好者，并深入了解他们的需求和困难后形成的，能够全面解决现实存在的诸多问题。最炫广场舞与市场已有 APP 功能对比如下表 2-1 所示：

表 2-1 最炫广场舞与市场已有 APP 功能对比表

最炫广场舞与市场已有 APP 功能对比结果						
APP 功能	视频下载	视频教学	舞友交流	周边商家信息	周边地图及导航	时间服务及备忘提醒
APP 名称						
就爱广场舞	√	√	√			
糖豆广场舞	√	√				
全民广场舞	√	√				
恰恰广场舞	√	√	√			

天天广场舞	√	√				√
播视广场舞	√	√				
最炫广场舞	√	√	√	√	√	√

广场舞已成为广大群众无法回避的一个问题，但却鲜有互联网方向从业人士关注该问题。最炫广场舞正是为广场舞爱好者定向设计开发并且适合他们的一款 APP，可以为他们解决无法找到大组织从而三五成群影响居民生活的问题，减少摩擦矛盾，增添日常生活的情趣。

## 2.2 功能需求

### 2.2.1 功能划分

#### (1) 用户端

实现功能分为四大模块：舞群导航、视频聚合、舞具商城和个人中心。

#### (2) 管理端

实现功能分为：管理员登陆、注册以及用户后台数据的增删查改操作。

### 2.2.2 功能描述

#### (1) 用户端

具体功能描述：

- 舞群导航：显示用户当前位置以及周边舞群位置，用户选择其中一个舞群后实现线路规划导航。
- 视频聚合：定期向用户推送当前最热门的广场舞视频。用户设置开启摇一摇功能后，可以实现摇一摇刷新当前推送内容。提供分类教学模块，用户选择其中一个分类后，即可查看相应教学视频。相应视频下方可点赞、评论、收藏和分享。
- 舞具商城：用户指定当前所在地区，系统将智能推荐周边售卖有关广场舞音响、服装等舞具商店。
- 个人中心：我的信息（可查看、修改），我的舞友（保存联系电话，可电话、短信联系），我的收藏（包括视频收藏和商城收藏），我的上传（我拍摄的广场舞分享视频），我的下载（视频下载，用于离线观看），申请舞群（向系统提交新的舞群地点和个人认证信息，管理员审核通过后，即可在舞群导航模块搜索到），时间服务（用于设置提醒闹钟），开启摇一摇（用于刷新定期推送视频信息），退出登陆。

系统用户端功能结构如下图 2-1 所示：



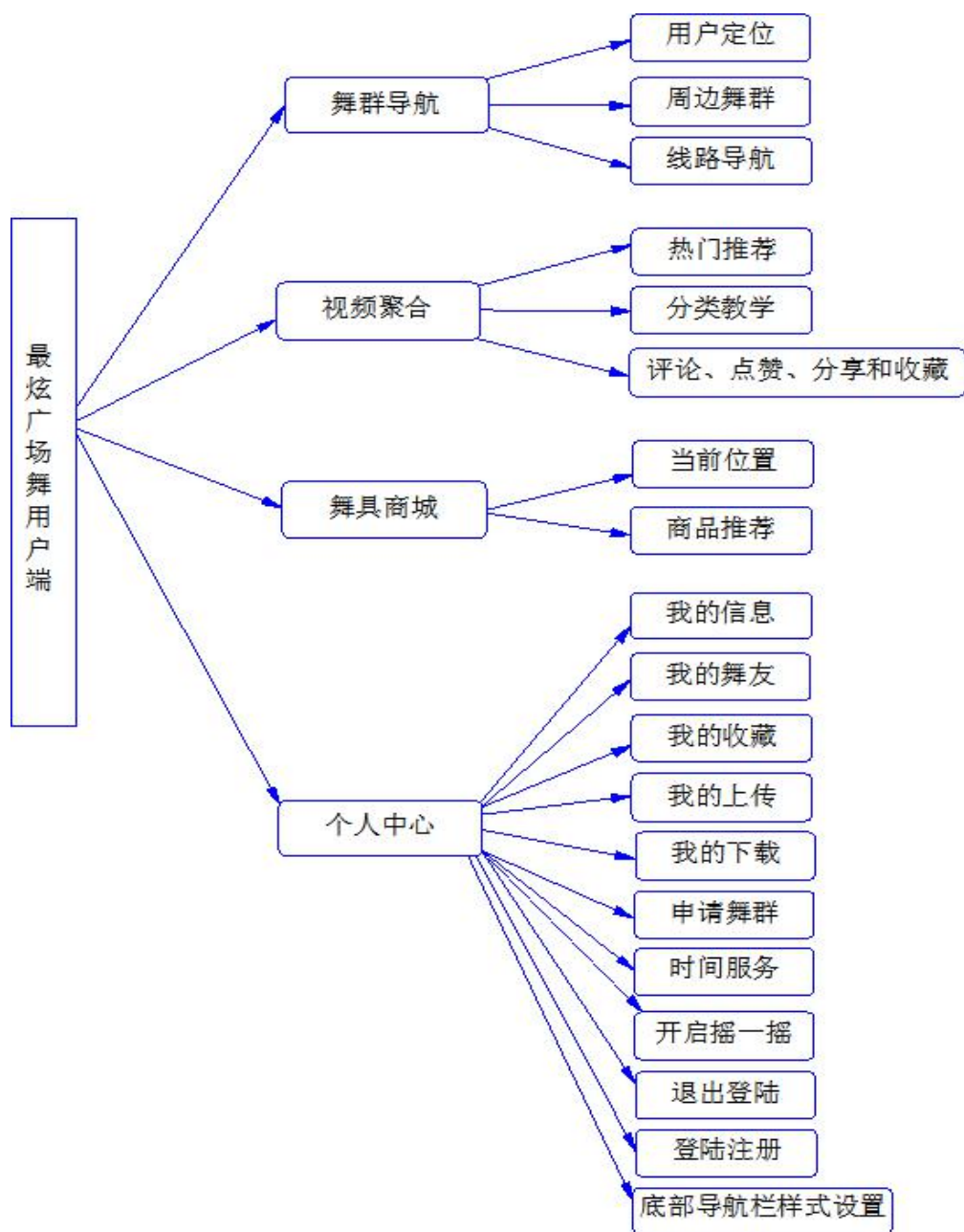


图 2-1 最炫广场舞用户端功能结构图

## （2）管理端

管理员登陆和注册数据单独存放在一个数据集合中。其中对用户端数据进行增删查改操作的集合共分为五个。

- 用户信息集合。
- 舞群信息集合。
- 视频信息集合。
- 评论信息集合。
- 商品信息集合。

系统管理端功能结构如下图 2-2 所示：

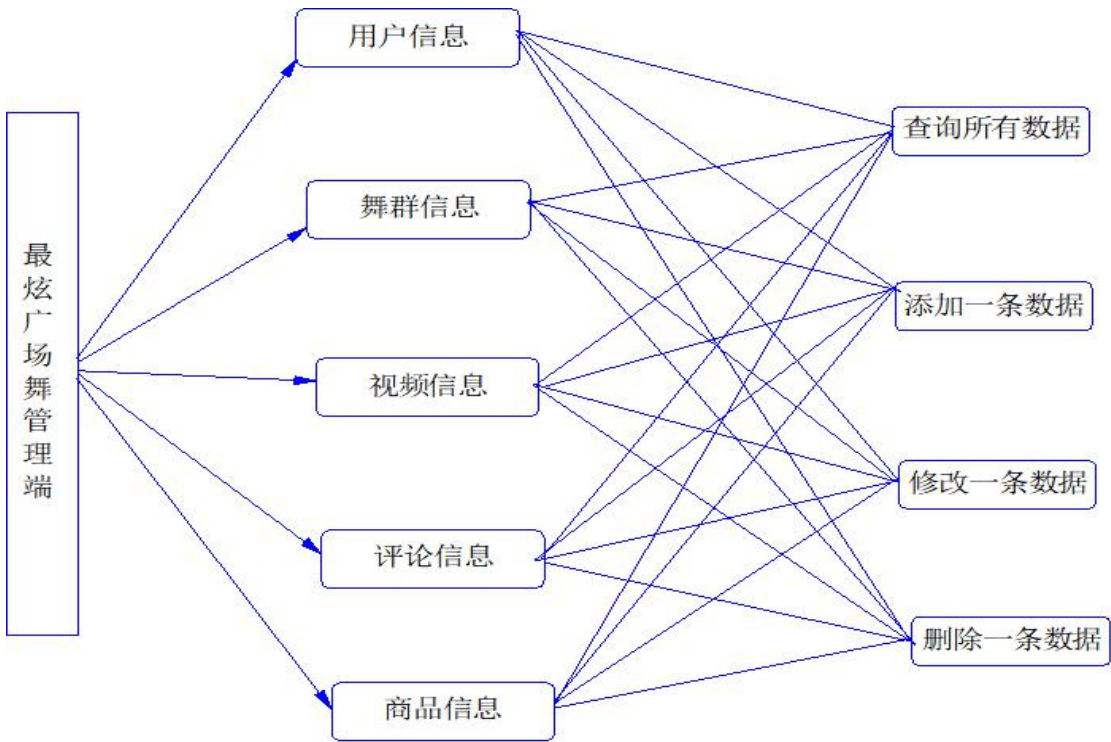


图 2-2 最炫广场舞管理端功能结构图

## 2.3 用例图

### 2.3.1 用例模型

本系统参与者包括用户和管理员，具体模型板块分为四大类：舞群导航、视频聚合、舞具商城和个人中心。

#### （1）舞群导航板块

系统舞群导航用例图如下图 2-3 所示：

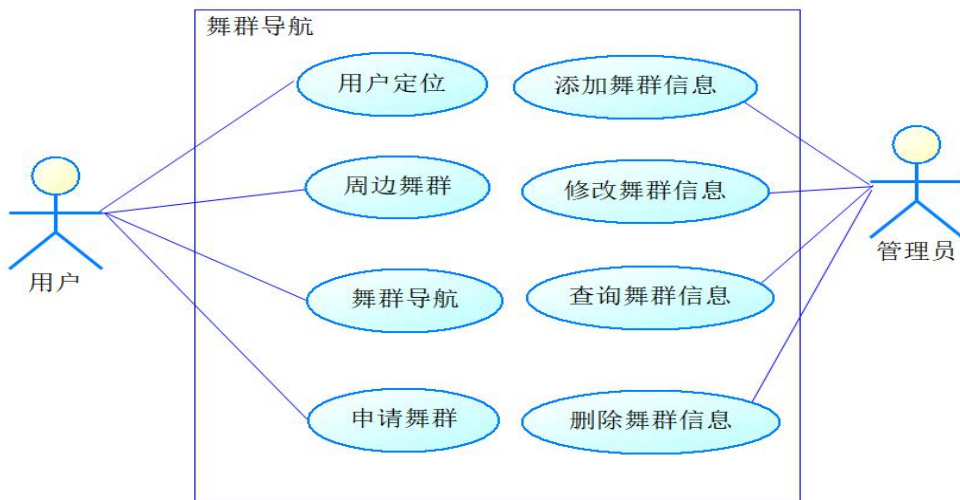


图 2-3 舞群导航板块用例图

## (2) 视频聚合板块

系统视频聚合用例图如下图 2-4 所示：

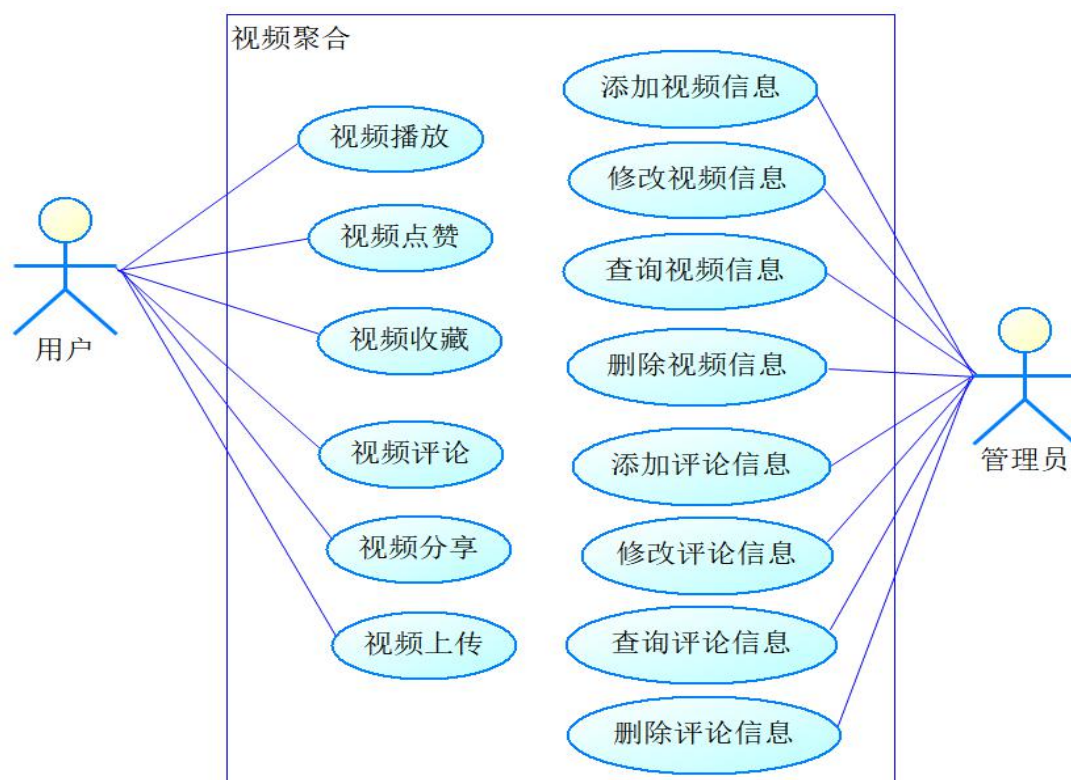


图 2-4 视频聚合板块用例图

## (3) 舞具商城板块

系统舞具商城用例图如下图 2-5 所示：

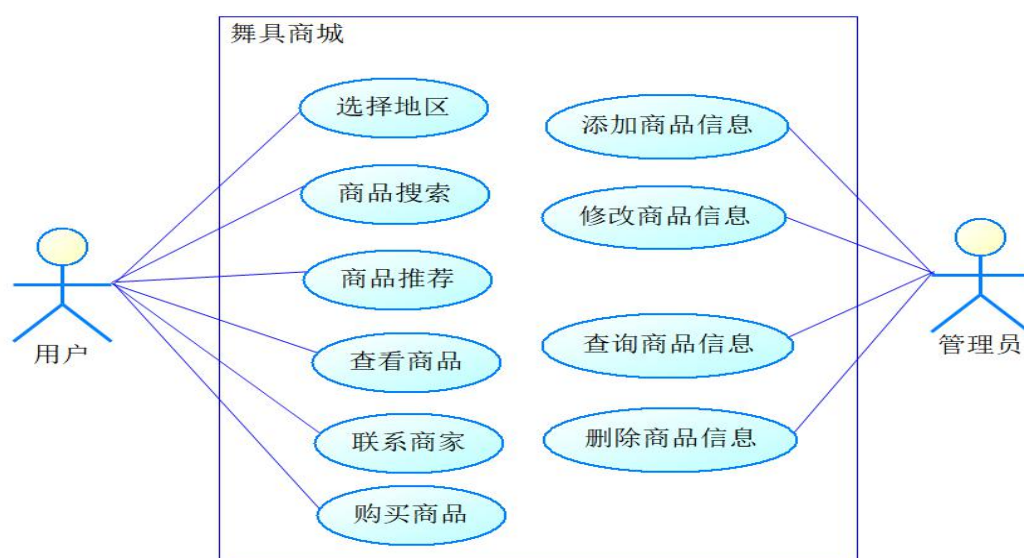


图 2-5 舞具商城板块用例图

(4) 个人中心板块

系统个人中心用例图如下图 2-6 所示：

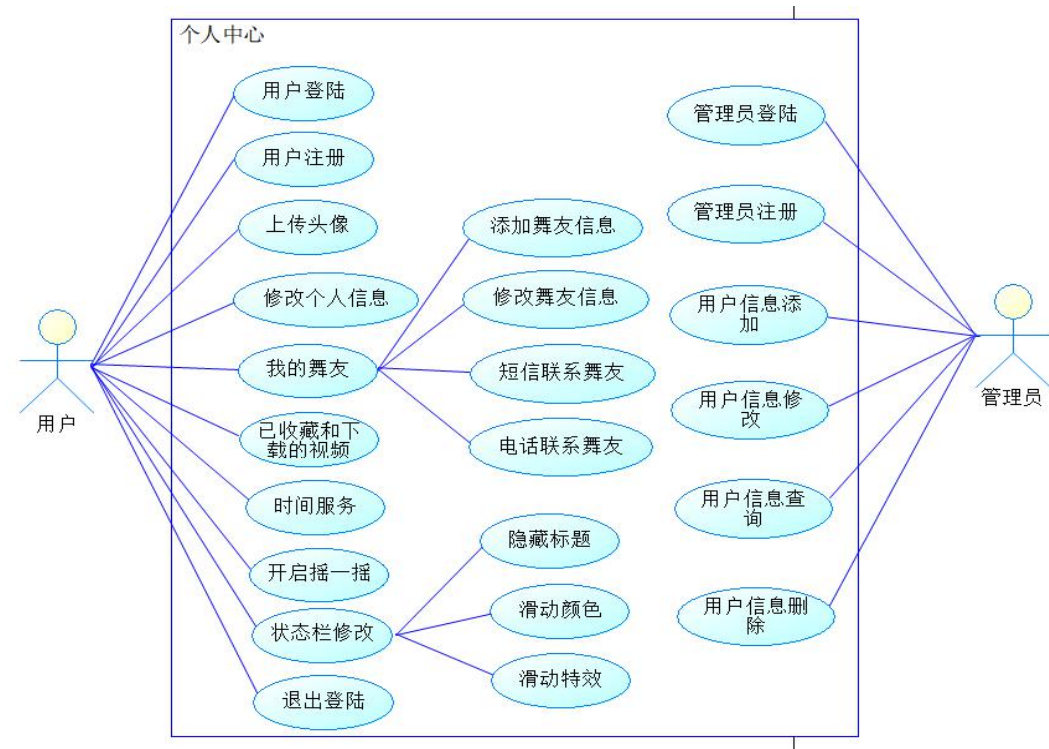


图 2-6 个人中心板块用例图

2.3.2 用例描述

(1) 登陆用例

表 2-2 登录用例描述

登陆用例	
参与者	用户和管理员。
触发条件	用户点击 Android 端登录按钮，管理员点击 Web 端登陆按钮。
前置条件	用户打开了 APP，管理员打开了后台管理系统。
后置条件	用户登录到 APP 主页，管理员进入后台数据管理主页。
正常流程	1) 用户打开 APP，管理员打开后台管理系统； 2) 输入账号密码开始登录； 3) 登录成功则进入主页面。
扩展流程	2a.输入账号或者密码为空

---

1) 系统提示账号或密码不能为空, 请重新输入。

3a.输入账号或者密码错误

1) 系统提示已输入账号和密码不匹配, 请仔细检查再重新输入。

---

(2) 注册用例

表 2-3 注册用例描述

注册用例	
参与者	用户和管理员。
触发条件	用户点击 <b>Android</b> 端注册按钮, 管理员点击 <b>Web</b> 端注册按钮。
前置条件	用户打开了 <b>APP</b> , 管理员打开了后台管理系统。
后置条件	用户和管理员注册成功。
正常流程	<ol style="list-style-type: none"><li>1) 用户和管理员分别打开各自系统端;</li><li>2) 点击注册按钮, 进入注册页面;</li><li>3) 输入需要注册的信息;</li><li>4) 注册成功。</li></ol>
扩展流程	<p>2a.输入注册信息为空</p> <ol style="list-style-type: none"><li>1) 系统提示需注册信息为必填项, 请重新输入, 不能留空。</li></ol> <p>3a.输入账号重复</p> <ol style="list-style-type: none"><li>1) 系统提示该账号已被注册, 请换一个的账号输入。</li></ol>

---

(3) 申请舞群用例

表 2-4 申请舞群用例描述

申请舞群用例	
参与者	用户和管理员。
触发条件	用户进入 <b>Android</b> 端申请舞群界面。
前置条件	舞群信息需要经过管理员在后台管理系统中审核通过。

---

后置条件	舞群申请成功，并可在用户 Android 端界面地图中查看。
正常流程	<ol style="list-style-type: none"> <li>1) 用户进入申请舞群界面，填写舞群名称、具体地点和负责人信息；</li> <li>2) 管理员在后台管理系统查看申请信息，并进行资料审核；</li> <li>3) 审核通过，可在 Android 端地图中查看。</li> </ol>
扩展流程	<p>2a.申请舞群信息出现为空</p> <ol style="list-style-type: none"> <li>1) 系统提示申请舞蹈信息不能为空，请按要求全部填写完毕。</li> </ol> <p>3a.输入舞群名称重复</p> <ol style="list-style-type: none"> <li>1) 系统提示该舞群名称已被申请，请换一个名称申请。</li> </ol>

#### (4) 视频点赞评论收藏分享用例

表 2-5 视频点赞评论收藏分享用例描述

视频点赞评论收藏分享用例	
参与者	用户。
触发条件	用户点击点赞评论收藏或者分享功能按钮。
前置条件	用户登陆成功，管理端后台已经上传并发布当前最新视频信息。
后置条件	相关视频点赞评论收藏数据信息更新。
正常流程	<ol style="list-style-type: none"> <li>1) 用户进入 Android 端视频聚合功能界面；</li> <li>2) 点击播放查看相关最新视频；</li> <li>3) 在播放的视频底部可选择进行点赞、评论、收藏和分享。</li> </ol>
扩展流程	<p>2a.该视频信息不存在</p> <ol style="list-style-type: none"> <li>1) 系统提示该视频信息在管理端已被删除，请刷新客户端页面信息。</li> </ol>

#### (5) 商品信息查看购买用例

表 2-6 商品信息查看购买用例描述

商品信息查看购买用例	
参与者	用户。

触发条件	用户点击商品信息查看或者联系商家购买功能按钮。
前置条件	用户登陆成功。
后置条件	商品信息正常浏览，或者购买成功。
正常流程	<ol style="list-style-type: none"> <li>1) 用户进入 <b>Android</b> 端舞具商城功能界面；</li> <li>2) 点击相应商品信息推荐栏目图标；</li> <li>3) 查看相应具体商品信息；</li> <li>4) 点击联系商家购买商品功能按钮；</li> <li>5) 商品购买成功。</li> </ol>
扩展流程	暂无

(6) 舞友信息添加修改查看用例

表 2-7 舞友信息添加修改查看用例描述

舞友信息添加修改查看用例	
参与者	用户。
触发条件	用户点击舞友查看、添加和修改功能按钮。
前置条件	用户登陆成功。
后置条件	舞友信息更新。
正常流程	<ol style="list-style-type: none"> <li>1) 用户进入 <b>Android</b> 端个人中心我的舞友功能界面；</li> <li>2) 查看所有舞友姓名、性别信息；</li> <li>3) 选择界面右下方添加按钮，进行舞友信息添加；</li> <li>4) 点击查看界面中舞友具体姓名，进入舞友具体信息界面；</li> <li>5) 选择电话或者短信联系舞友，也可点击页面右上方按钮，修改舞友信息。</li> </ol>
扩展流程	<p>2a.添加舞友信息已存在</p> <ol style="list-style-type: none"> <li>1) 系统提示该舞友姓名已存在，不需要重新添加，或者换一个姓名添加。</li> </ol>

## （7）时间服务用例

表 2-8 时间服务用例描述

时间服务用例	
参与者	用户。
触发条件	用户选择提醒时间，可提醒的次数以及铃声方式。
前置条件	用户登陆成功。
后置条件	到达用户设置时间后，系统自动提醒用户。
正常流程	<ol style="list-style-type: none"><li>1) 用户进入 Android 端个人中心时间服务功能界面；</li><li>2) 选择提醒时间，设置提醒次数和铃声方式；</li><li>3) 系统提醒用户指定时间已到达；</li></ol>
扩展流程	暂无

## 2.4 外部配置及接口需求

### 2.4.1 移动端 APP 配置需求

Android 系统的智能手机、RAM：1G 以上、支持 GPS、线性加速度和重力传感器、摄像头。

### 2.4.2 软件接口

- 操作系统：Android5.0 系统及以上。
- 网络协议：HTTP1.1。

## 2.5 性能需求

### 2.5.1 时间特性要求

Android 移动端运行的响应时间要求不能超过 2 秒。

- 响应时间：用户界面间跳转间隙不超过 1 秒。
- 数据转换与传输时间：数据转换主要是涉及到与服务器相关数据请求的问题。为了减少一次显示过多的记录会消耗太多的时间，在设计时让数据尽量分页获取和显示。传输时间和服务器的性能设置以及无线网络的信号强弱有关。
- 服务运行时间：如非特殊原因应保持 24 小时开通，即保持服务端一直处于运行状态。



### 2.5.2 灵活性及精度

(1) 用户使用 Android 端在进行周边舞群功能相关操作时，确保周边舞群信息地址定位的准确度，并且方便用户选择相应舞群，进行线路导航操作。

(2) 舞蹈教学视频信息保持每天更新一次，并且管理员进行相关内容审核，确保上传视频的质量。

(3) 舞具商品信息要确保定期更新，并确保商家信息来源合法，用户可以随时和商家进行沟通。

## 3 系统设计

### 3.1 概要设计

#### 3.1.1 设计目标

随着中国地区热爱广场舞人数的迅速增加，人口老龄化不断上升，一款实用便利的广场舞 APP 的设计研发成功后，在给喜爱广场舞的人群带来便利的同时，也能够为商家带来新的商机。本系统设计要突出用户人群要求的舞群导航和视频观看下载功能。在此基础上提高用户数量，并结合用户动态发展周边商家功能。

本系统提供的核心功能：

- 提供周边地图定位及舞群定向导航功能。
- 提供当前最新、最热门的广场舞视频，并支持上传功能。
- 针对用户的活跃度，智能推荐周边热门广场舞商品。
- 用户间进行分享或互动，提高用户间的交流，例如拨打电话和短信功能。
- 提供时间服务功能，当在户外活动时，注意相关时长，以免忘记要事。

在上述功能基础上，优化用户界面和系统性能，提高用户体验。此外，本系统设计希望可以达到以下目标：

- 系统运行稳定，安全可靠。
- 界面设计简洁大气，界面交互友好。
- 信息查询和发布方便，数据存储安全性能高。
- 定位功能能够克服信号弱，通过数据信号、wifi 信号和 GPS 信号多途径定位。
- 视频功能做到及时更新。
- 系统后台管理方便，并提供界面化操作。

#### 3.1.2 系统技术框架介绍

##### (1) 客户端

采用 Android 设计。目前市场上，采用 Android 操作系统的手机占用的市场份额达到了百分之八十。Android 系统提供了界面化操作，系统版本定期更新，用户体验也变得越来越好，使用基于 Android 系统的手机要求也在逐步降低。

本系统的客户端界面设计部分调用了 Android 系统自带的应用层，用于给用户

展示相关功能界面，方便用户进行相关功能操作<sup>[1]</sup>。在用户进行功能操作的时候，客户端会调用 Android 系统自带的应用框架层，提供相关界面 UI 展示和功能操作的 API。除此之外，客户端还会调用 Android 系统自带的 SQLite 数据库，并调用 Android 系统自带的系统运行层。本系统需要用户使用 Android 手机才可以使用，整个 Android 系统的底层是基于 Linux 内核层，为相关 APP 提供硬件支持和驱动<sup>[2][3][4]</sup>。

（2）数据存储端

本系统数据存储端采用非关系型 MongoDB 数据库<sup>[5]</sup>。MongoDB 属于 NoSQL 分支，舍弃了复杂的关系和实体间的关联。MongoDB 也可以被称为是一种文档型数据库，类似于我们日常写文档，不用担心文档的具体内容，可以随意存放在指定文件夹。MongoDB 中存入和读取的数据格式均为 BSON 格式，BSON 是一种类似 JSON 格式的数据。

表 3-1 MySQL 与 MongoDB 名词对照表

MySQL 和 MongoDB 名词对照	
MySQL（关系型数据库）	MongoDB（非关系型数据库）
database（数据库）	database（数据库）
table（表）	collection(集合)
row(行)	document（文档）
column（列）	field(域)
index（索引）	index（索引）
table joins(表关系)	无
primary key(主键)	自动将_id 字段设置为主键

（3）管理端

后台管理员端采用 Bootstrap、JavaScript 和 CSS 实现。

Bootstrap 是一个用于快速搭建 Web 网站页面的框架。Bootstrap 框架具有的特点：

- 方便后端开发人员学习，学习成本较低。
- 为开发人员定义好相关应用控件。
- 该框架是开源的。

- 制作的界面简单，耐看。

JavaScript<sup>[6]</sup>和 C++、Java 一样具有面向对象特性，并具有安全性能的脚本语言，一般通过在 HTML 语言中进行嵌入实现特定功能。Javascript 也是一种平台无关性的语言，它依赖于浏览器本身，与操作环境无关，只要计算机能够运行浏览器，并支持 JavaScript，就可以正确运行<sup>[7]</sup>。

CSS(英文全称：Cascading Style Sheets)中文意思为指层叠样式表，是一种用来表现 HTML 或 XML 等文件样式的计算机语言。CSS 一般可以用来美化装饰静态网页，还能够和相关脚本语言对网页中动态控件进行特定的修饰。

#### (4) Python 后台服务器端

本系统后台采用 Python 语言<sup>[8]</sup>编写，使用基于 Python 的 Flask 框架来完成相关后台接口编写<sup>[9][10]</sup>，并完成用户端与 MongoDB 数据库的交互。

Flask 是一个基于 Python 的开源“微框架”，很多功能既可以开发者自定义实现，也可以使用社区共享的模块。Flask 框架的核心包含路由、Web 服务器网管接口、调试和模板系统，Werkzeug 给 Web 网关子系统提供核心技术支持，Jinja2 给模板系统提供核心技术支持<sup>[11]</sup>。除此之外，Flask 支持所有的关系型数据库<sup>[12][13][14]</sup>，对非关系型数据库也支持，与 MongoDB 数据库结合开发非常完美。

后台接口功能代码部署在阿里云服务器上，采用的操作系统是 Windows Server 2008 标准版 SP2 32 位中文版。后台接口功能部署的服务器采用了 Flask 框架自带的服务器和 Apache 服务器，其中 Flask 框架自带的服务器负责运行功能接口代码和连接 MongoDB 数据库，Apache 服务器用于存放 Android 用户端所需的相关图片、视频资源。

### 3.1.3 系统技术架构

本系统技术架构采用前后端分离模式，符合标准的 MVC 软件设计模式。MVC 是一种当前软件设计领域比较流行的设计模式，这种模式将应用程序实现过程分为模型（Model）、视图（View）和控制器（Controller）三个部分，主要特点是把显示和数据分离，增加了软件的可维护性和重用性。

在本系统技术架构设计中，代表 MVC 设计模式的三大模块分别如下：

- 视图层（View）：用于获取具体数据并提供给 Android 用户端，具体实现以接口来表示。
- 控制器层（Controller）：将具体的模型层与视图层连接在一起，完成后端接口业务逻辑的具体实现和相关性能优化。
- 模型层（Model）：一般是和数据库完成映射，使得控制器层能够较轻松的对数据库进行增删改查，此处数据存储采用 MongoDB 数据库。

视图层使用 RESTful 方式实现接口。接口一般会有 Post、Get、Put 和 Delete 等方式请求，具体表示发送、获取、修改和删除数据。使用一个 URL 链接进行数据传输，一般是基于 Http1.1 协议来实现具体的数据传输。

RESTful 架构四个基本原则：

- 使用 GET、POST、PUT 和 DELETE 四个 HTTP 动词。
- 无状态连接，每个请求都是独立的，服务器端不应保存过多上下文状态。
- 为每个资源设置 URI。
- 使用 XML 或者 JSON 进行数据传递。

RESTful 架构的特性：

- 互联网环境下，可以快速理解任何应用的架构和 API。
- 分布式环境下，可以发送任何请求到任意服务器。
- 异构环境下，统一访问和使用任何资源。

最后，下面的示意图详细说明了本系统实际部署的流程以及数据请求存储的过程，能够更加清晰的介绍本系统的技术架构，具体如下图 3-1 所示：

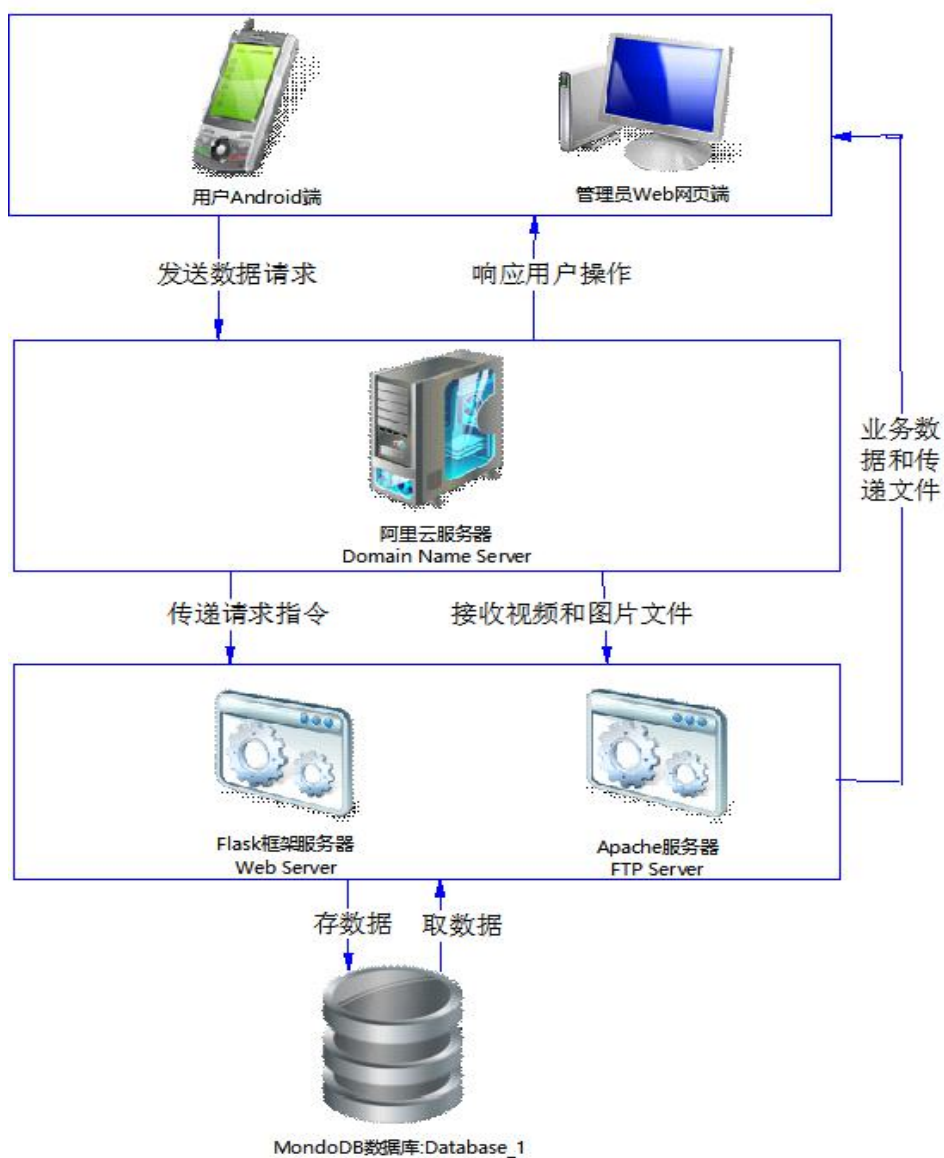


图 3-1 系统技术部署架构图

## 3.2 详细设计

### 3.2.1 系统主要功能模块设计与实现

#### (1) 登陆和注册

分为用户 Android 端和管理员网页端，后端使用 MongoDB 数据库进行存储用户名和密码。用户账号和管理员账号分别存储在不用的集合中，管理员可以增删查改用户集合中的数据。

用户 Android 端采用 OkHttp 网络数据操作框架对后端数据进行访问和请求,登陆和注册功能分别要用到后端登陆接口和注册接口。登陆接口使用 GET 方式进行数据请求,注册接口使用 POST 方式进行数据请求。管理员端是直接连接后端 MongoDB 数据库进行相应功能实现，两者实现方式大致相同。实现流程如下图 3-2 所示：

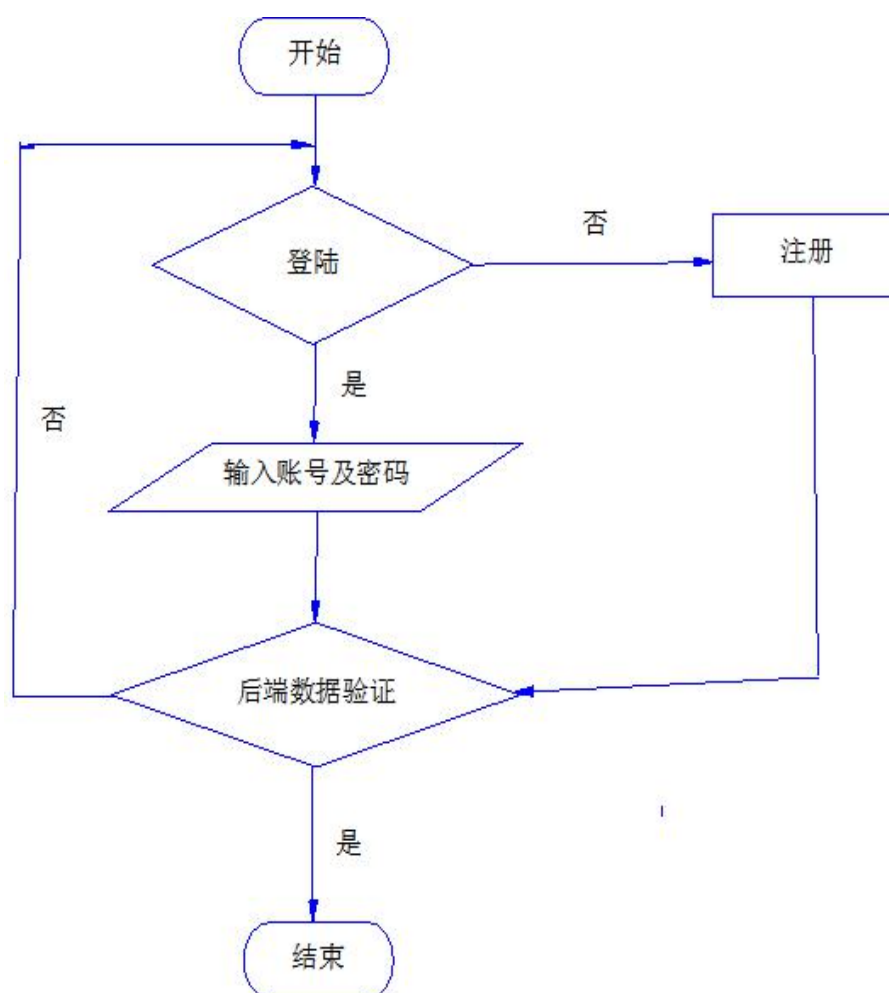


图 3-2 系统登陆注册功能逻辑图

OkHttp 库 Get 和 Post 方式与服务器端交互代码，其中 Post 请求方式实现过程和 Get 方式类似。

下面代码是使用 Get 方式获取服务器上数据：

```

1 public static void sendOkHttpRequest(final String address, final okhttp3.Callback callback)
2 {
3     OkHttpClient client = new OkHttpClient();
4     Request request = new Request.Builder().url(address) .build();
5     client.newCall(request).enqueue(callback);
6 }

```

用户 Android 端进行登陆和注册验证判断核心代码，其中注册核心代码和登陆实现过程类似。

登陆部分核心代码：

```

1 username = (EditText)findViewById(R.id.login_edt_username);
2 password = (EditText)findViewById(R.id.login_edt_pwd);
3 Button btn_login = (Button)findViewById(R.id.login_btn_login);
4 btn_login.setOnClickListener(new View.OnClickListener() {
5     public void onClick(View v) {
6         if(username.getText().toString().equals("") || password.getText().toString().equals(""))
7             Toast.makeText(LoginActivity.this, "用户名或者密码不能为空，请重新登陆！",
Toast.LENGTH_LONG).show();
8         else {
9             HttpUtil.sendOkHttpRequest("http://120.79.82.151/login", new okhttp3.Callback()
10             {
11                 @Override
12                 public void onResponse(Call call, Response response) throws IOException {
13                     showResponse(response.body().string());
14                 }
15                 @Override
16                 public void onFailure(Call call, IOException e) {
17                     }
18             });
19         }
20 }

```

管理员网页端登陆和注册核心代码，注册部分实现核心代码和登陆类似。

管理员登陆部分核心代码：

```

1 @auth.route('/login', methods=['GET', 'POST'])
2 def login():
3     form = LoginForm()
4     if form.validate_on_submit():
5         useradmin = mongo.db.userAdmin.find({"email":form.email.data})
6         edit_admin = {'email': u"待赋值", 'password': u"待赋值"}
7         useradminarray = UserAdminArray(... ...)
8         for s in useradmin:
9             useradminarray.id = s['_id']
10            useradminarray.email = s['email']
11            useradminarray.username = s['username']
12            useradminarray.user_tel = s['user_tel']
13            useradminarray.user_address = s['user_address']
14            useradminarray.password = s['password']
15        for s in useradmin:
16            edit_admin['email'] = s['email']
17            edit_admin['password'] = s['password']
18        if (useradminarray.email != u"待赋值") & (useradminarray.password ==
19        form.password.data):
20            login_user(useradminarray)
21            return redirect(url_for('home.admin_dashboard'))
22        # when login details are incorrect
23    else:
24        # login_user(useradminarray)
25        flash(u'邮箱无效或密码错误')
26    return render_template('auth/login.html', form=form, title='Login')

```

## （2）搜索显示周边舞群

周边舞群功能在地图上显示用户当前所在地附件的舞群地点信息，并在地图上标识具体位置信息。舞群信息通过 **OKHttp** 网络库从后端数据库中获取舞群的具体地点数据，然后通过调用百度地图的地理位置解析接口解析出该位置的具体经纬度数据，通过得到的经纬度数据把该舞群位置标识在地图上。实现流程如下图 3-3 所示：

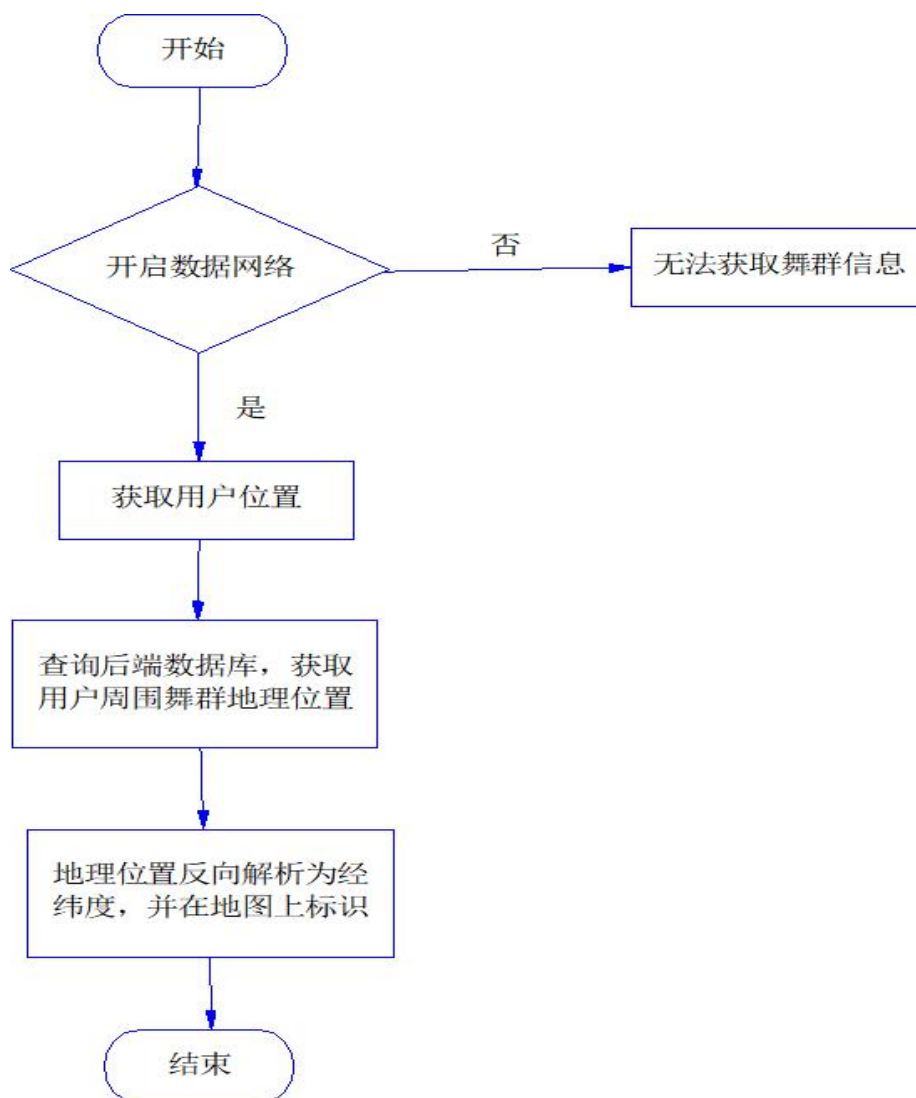


图 3-3 系统搜索显示周边舞群功能逻辑图

把具体地理位置解析成经纬度数据的实现代码：

```

1 public GeoPoint getGeoPointBystr(String str) {
2     GeoPoint gpGeoPoint = null;
3     if (str!=null) {
4         Geocoder gc = new Geocoder(getActivity(), Locale.CHINA);
5         List<Address> addressList = null;
6         try {
7             addressList = gc.getFromLocationName(str, 1);
8             if (!addressList.isEmpty()) {
9                 Address address_temp = addressList.get(0);

```



```

10         double Latitude=address_temp.getLatitude()*1E6;
11         double Longitude=address_temp.getLongitude()*1E6;
12         gpGeoPoint = new GeoPoint((int)Latitude, (int)Longitude);
13     }
14 } catch (IOException e) {
15     e.printStackTrace();
16 }
17 }
18 return gpGeoPoint;
19 }

```

把得到的经纬度坐标标识在地图上核心代码：

```

1 private void initGEO() {
2     mGeoSearch = GeoCoder.newInstance();
3     mGeoSearch.setOnGetGeoCodeResultListener(this);
4     mBaiduMap.setOnMapClickListener(new BaiduMap.OnMapClickListener() {
5         @Override
6         public void onMapClick(LatLng latLng) {
7             final LatLng lat = latLng;
8             mGeoSearch.reverseGeoCode(new ReverseGeoCodeOption().location(lat));
9         }
10        @Override
11        public boolean onMapPoiClick(MapPoi mapPoi) {
12            return false;
13        }
14    });
15 }

```

### （3）视频上传和下载

运用 **OKHttp** 库调用后端的视频上传和下载接口，视频上传到 **Apache** 服务器指定存放文件的目录下，并把该目录地址写入 **MongoDB** 数据库中，其中上传的视频文件命名统一采用上传成功那刻获取的系统时间经过特殊处理得到的一串数字命名。视频文件下载，获取数据库中视频文件在阿里云服务器中具体的存放地址，然后调

用相关方法从服务器上下载到用户手机本地。实现流程如下图 3-4 所示：

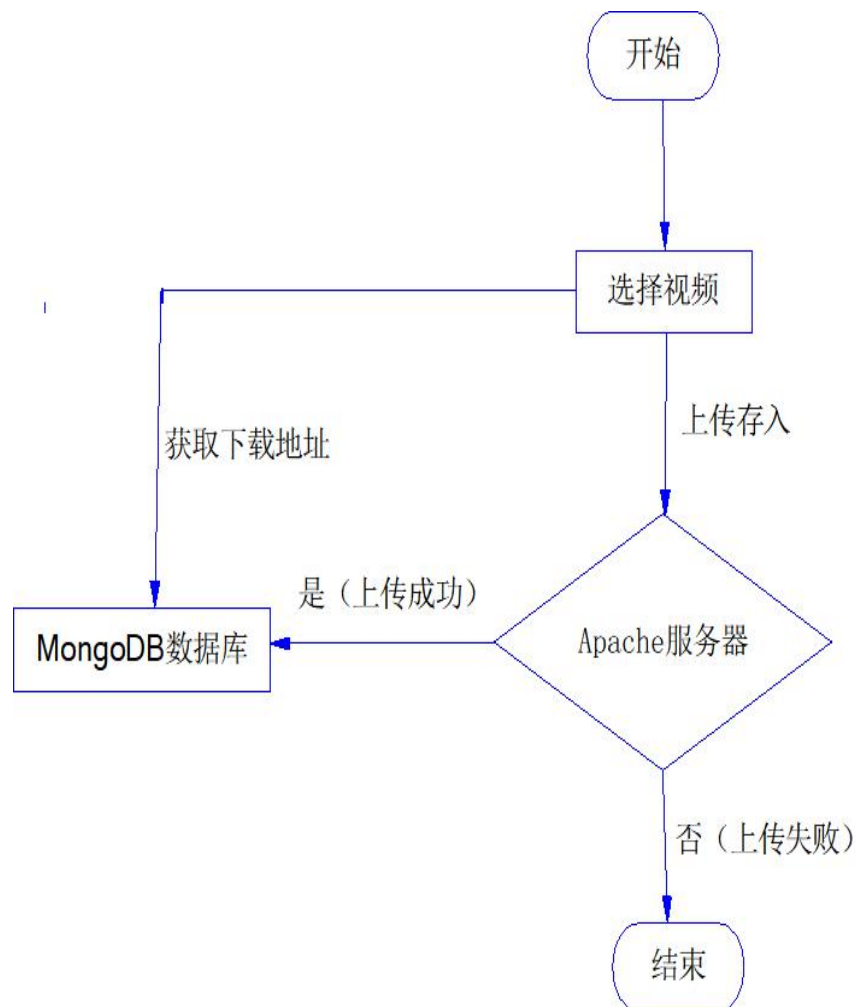


图 3-4 视频上传和下载功能逻辑图

OKHttp 库上传文件到服务端核心代码：

```
1 public static void postFile(String url, final ProgressListener listener, okhttp3.Callback
2     callback, File...files){
3     MultipartBody.Builder builder = new MultipartBody.Builder();
4     builder.setType(MultipartBody.FORM);
5     builder.addFormDataPart("myfile",files[0].getName(),
6         RequestBody.create(MediaType.parse("application/octet-stream"),files[0]));
7     MultipartBody multipartBody = builder.build();
8     Request request = new Request.Builder().url(url).post(new
9         ProgressRequestBody(multipartBody,listener)).build();
10    okHttpClient.newCall(request).enqueue(callback);
```

```
11 }
```

服务器端下载文件核心代码：

```
1 def download(filename):  
2     if request.method=="GET":  
3         if os.path.isfile(os.path.join('upload', filename)):  
4             return send_from_directory('upload',filename,as_attachment=True)  
5         abort(404)
```

#### （4）商城查看和购买

用户在舞具商城可以点击查看具体商品的信息。用户购买具体商品的操作方式为用户电话联系商家，然后双方商定具体的交货方式，本系统只提供一个展示功能，不提供具体支付和送货信息服务等功能。实现流程如下图 3-5 所示：

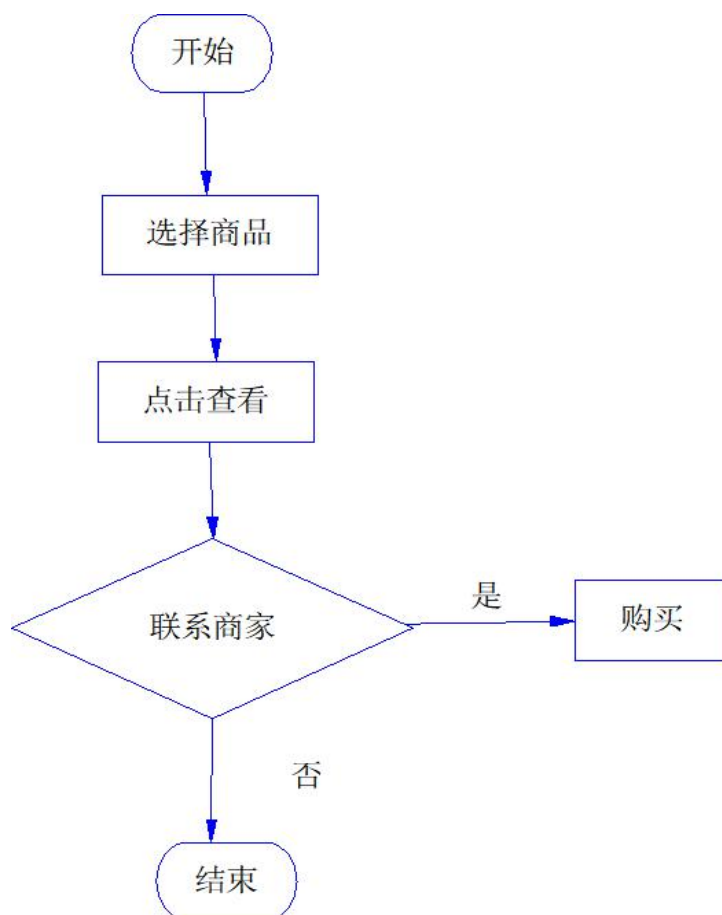


图 3-5 商品查看和购买功能逻辑图

#### （5）我的舞友

广场舞是一个群体性活动，除了日常教学视频和集合地点信息之外，用户之间还需要进行相关交流。我的舞友提供添加修改舞友姓名、电话、性别信息和拨打电话以及编辑发送短信功能。实现流程如下图 3-6 所示：

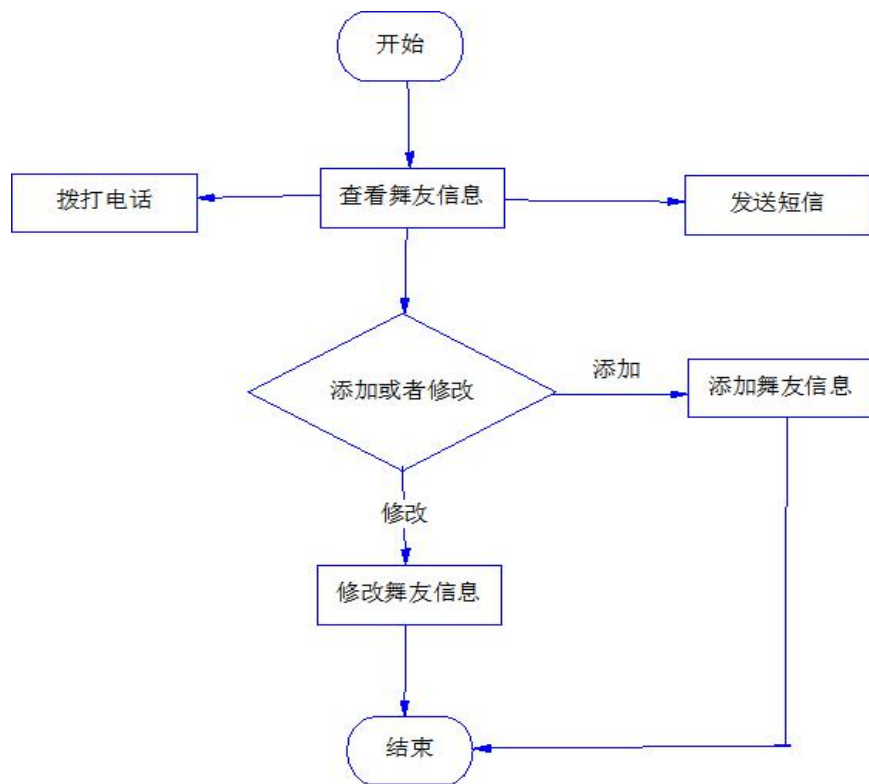


图 3-6 我的舞友功能逻辑图

拨打电话核心代码：

```

1 private void dialPhoneNumber(String phoneNumber) {
2     Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:" + phoneNumber));
3     startActivity(intent);
4 }

```

发送短信核心代码：

```

1 send_btn.setOnClickListener(new View.OnClickListener() {
2     public void onClick(View v) {
3         SmsManager sManager = SmsManager.getDefault();
4         PendingIntent pi = PendingIntent.getActivity(MessageEditActivity.this, 0, new
5             Intent(), 0);
6         String message = "";
7         if(send_mesg.getText().toString().equals("") ||
8             send_mesg.getText().toString().equals(null))
9             message = "你好呀，今晚大家一起去跳舞哟";
10        else

```

```

11         message = send_mesg.getText().toString();
12         sManager.sendTextMessage("15927175720", null, message, pi, null);
13         Toast.makeText(MessageEditActivity.this,"短信发送成功!
14         ",Toast.LENGTH_SHORT).show();
15     }
16 });

```

#### （6）时间服务

喜爱跳广场舞的群体，如果事先不约定好时间，会很难把控好具体活动的时间。时间服务提供的功能类似一般的闹钟功能，唯一不同的是此处由手机提供并且整合到了本系统中来，是一个方便喜爱广场舞用户的智能小工具。实现流程如下图 3-7 所示：

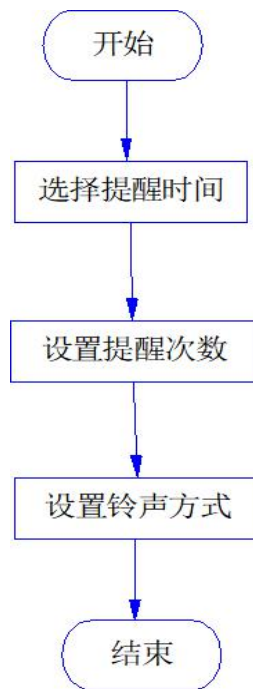


图 3-7 时间服务功能逻辑图

时间服务到时提醒核心代码：

```

1 private void setClock() {
2     if (time != null && time.length() > 0) {
3         String[] times = time.split(":");
4         if (cycle == 0) {
5             AlarmManagerUtil.setAlarm(this, 0, Integer.parseInt(times[0]), Integer.parseInt
6             (times[1]), 0, 0, "闹钟响了", ring);

```

```

7      } if(cycle == -1){
8          AlarmManagerUtil.setAlarm(this, 1, Integer.parseInt(times[0]), Integer.parseInt
9          (times[1]), 0, 0, "闹钟响了", ring);
10     } else {
11         String weeksStr = parseRepeat(cycle, 1);
12         String[] weeks = weeksStr.split(",");
13         for (int i = 0; i < weeks.length; i++) {
14             AlarmManagerUtil.setAlarm(this, 2, Integer.parseInt(times[0]), Integer
15             .parseInt(times[1]), i, Integer.parseInt(weeks[i]), "闹钟响了", ring);
16         }
17     }
18     Toast.makeText(this, "闹钟设置成功", Toast.LENGTH_LONG).show();
19 }
20 }

```

### 3.2.2 系统功能界面设计

系统功能界面设计分为用户 Android 端<sup>[15]</sup>和管理员后台网页端两大部分。

#### (1) 用户端界面

系统启动的首页，以及登陆注册设计图。用户点击最炫广场舞 APP 图标，首先会出现下图最左边的欢迎界面，大约 2 秒后自动跳转到登陆界面；然后，用户可点击登陆界面下方的立即注册按钮，系统会跳转到注册页面<sup>[16]</sup>。具体设计如下图 3-8 所示：

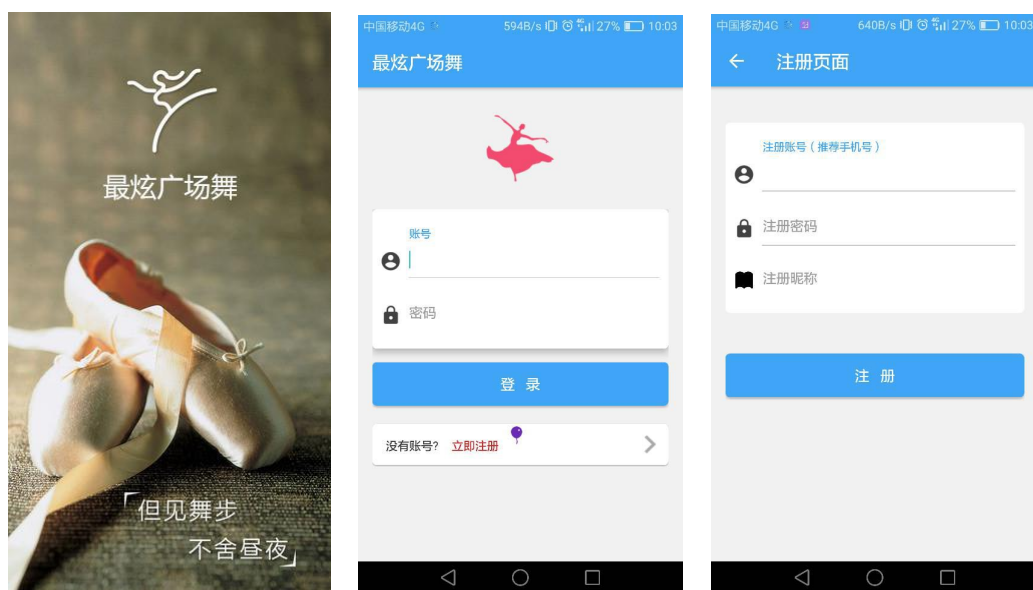


图 3-8 系统启动首页以及登陆注册功能模块界面

舞群导航：分为我的位置、周边舞群和线路导航三大模块。用户登陆成功后，显示界面便是下图最左方的我的位置界面。此时，用户和左右滑动选择周边舞群和线路导航功能界面。具体设计如下图 3-9 所示：

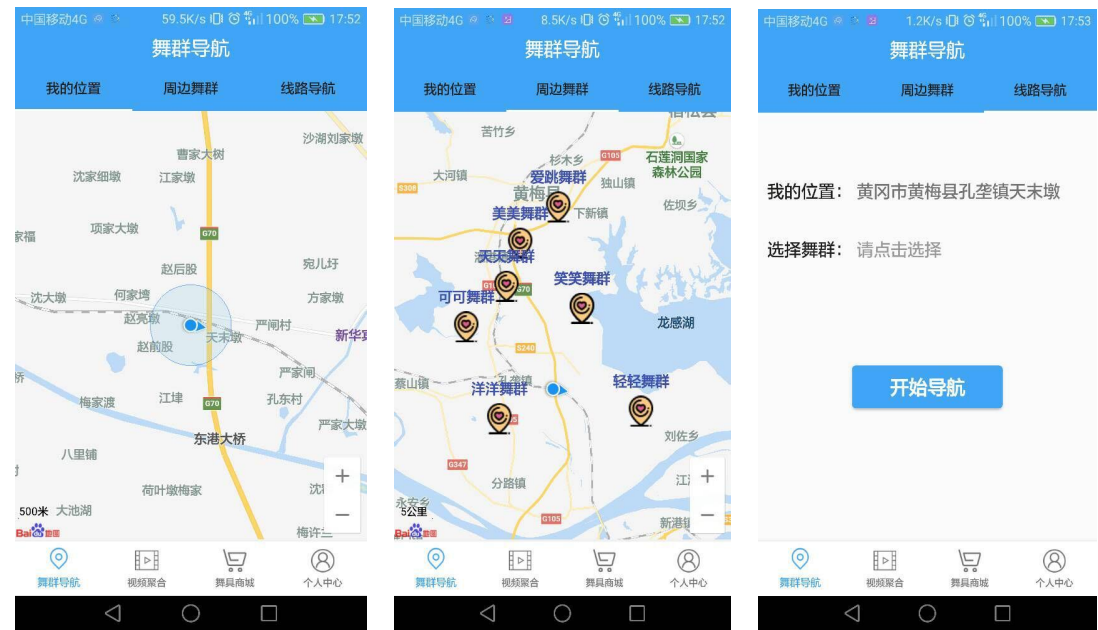


图 3-9 舞群导航功能模块界面

视频聚合：分为热门推荐和分类教学两大模块。可以进行视频播放、点赞、收藏、评论、分享和上传以及下载。用户点击底部导航栏中视频聚合可看到下图最左边的热门推荐界面，向右滑动看到分类教学界面，点击此界面中查看更多按钮，可看到舞群的更多教学视频。具体设计如下图 3-10 所示：



图 3-10 视频聚合功能模块主体界面



点击热门推荐中分享图标会出现下图中左边界面<sup>[7]</sup>；点击热门推荐中评论图标会出现下图的中间界面；点击视频聚合整体界面中的右下方+号按钮界面会出现下图最右边的界面。具体设计如下图 3-11 所示：

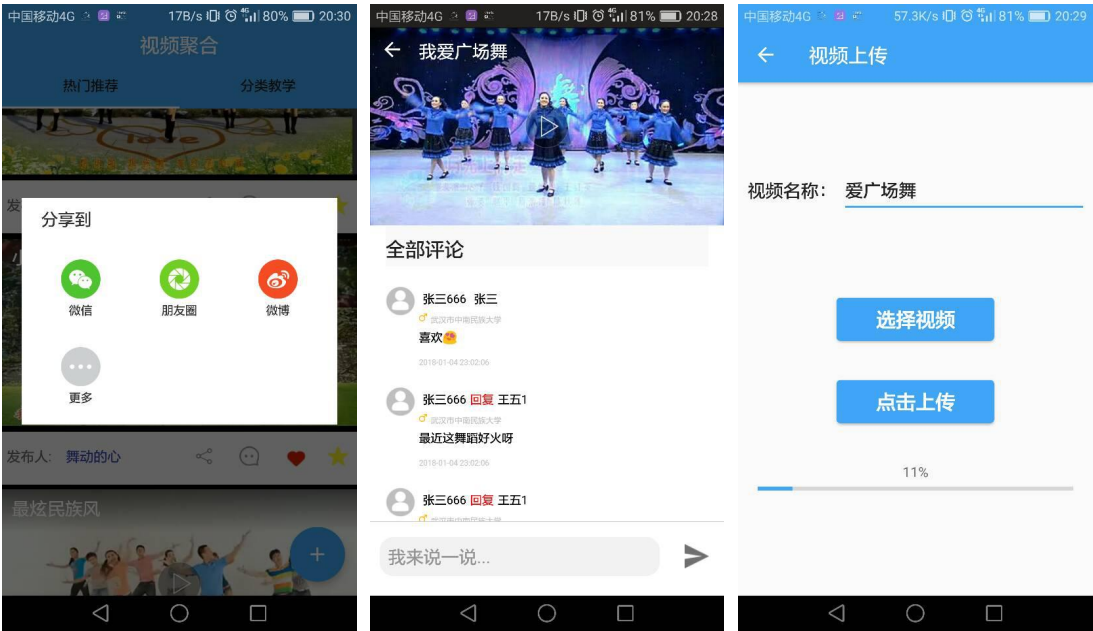


图 3-11 视频聚合模块分享评论和上传功能界面

舞具商城：用户点击界面左上方所在位置，可以选择相关地区，系统会确定出周边商城，并展示相关商品信息，具体如下图左边两个界面。用户点击界面首页搜索框，可以选择搜索指定商品信息，具体如下图最右边界面。具体设计如下图 3-12 所示：

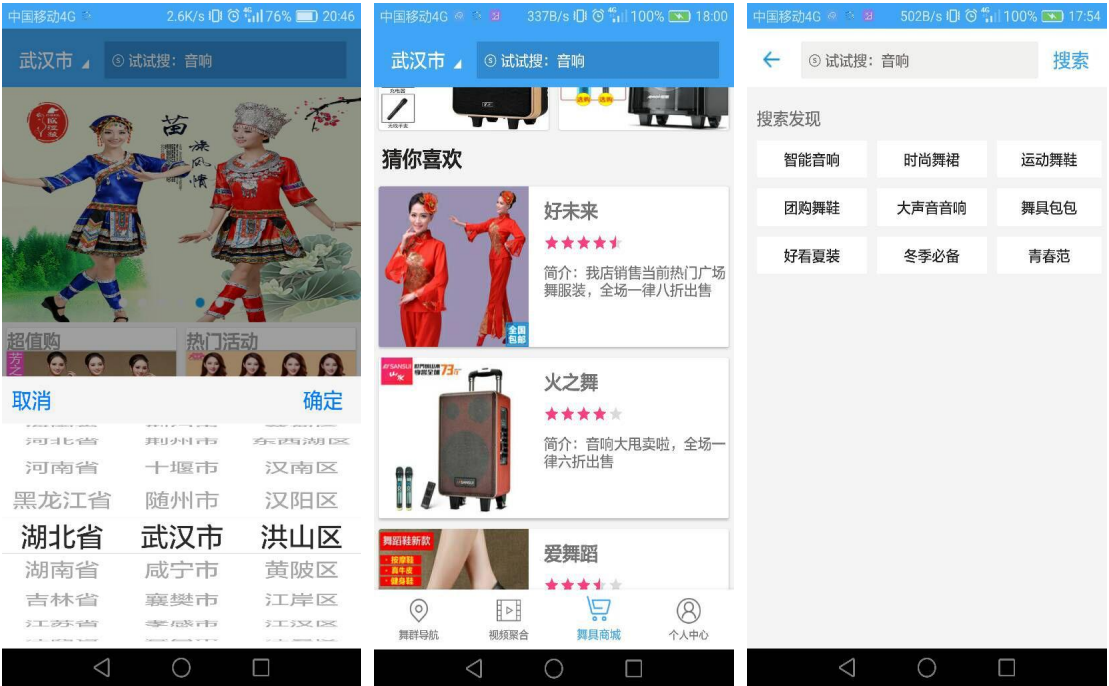


图 3-12 舞具商城模块选择位置及搜索功能界面



用户在搜索框中输入音响，点击搜索按钮后会出现下图最左边的界面；点击舞具商城模块中超值购商品模块，会出现下图中间界面；点击舞具商城首页好未来店铺，会出现最右边的界面。具体设计如下图 3-13 所示：

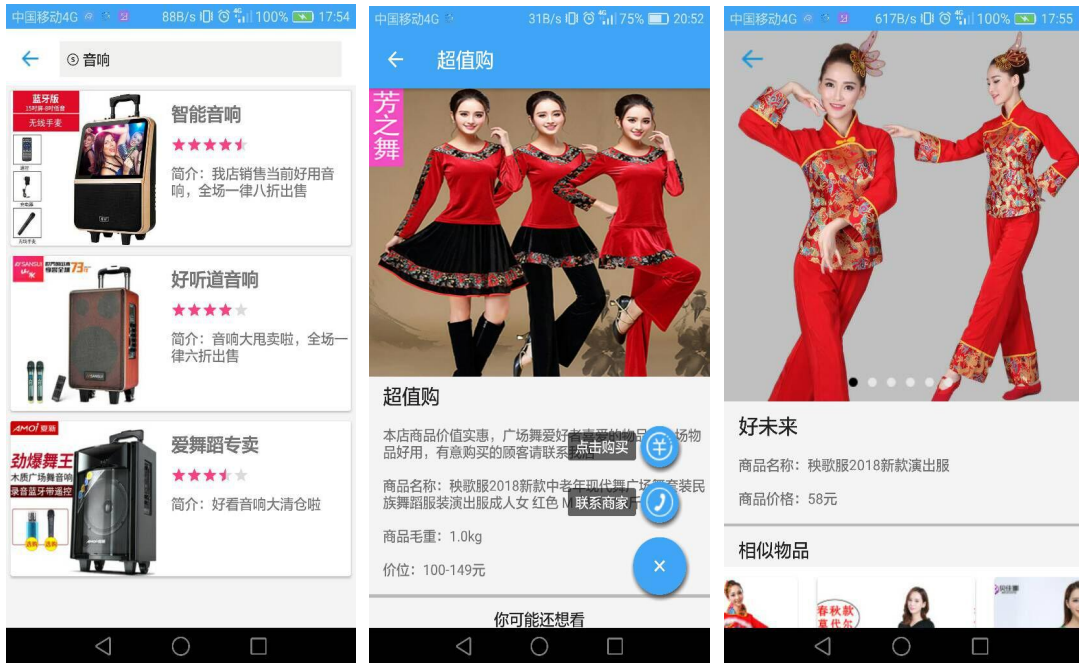


图 3-13 舞具商城模块搜索结果及查看商品信息功能界面

个人中心：当前模块主要提供用户信息账号管理功能，以及我的舞友、我的收藏、我的上传、我的下载，还包括申请舞群、时间服务及相关导航栏 UI 设定。下面最左边图是个人中心首页界面，中间的图是用户点击账号管理查看个人信息，并点击头像按钮进行头像上传修改操作；右边的图是用户点击个人信息界面的更多按钮，出现的结果。具体设计如下图 3-14 所示：

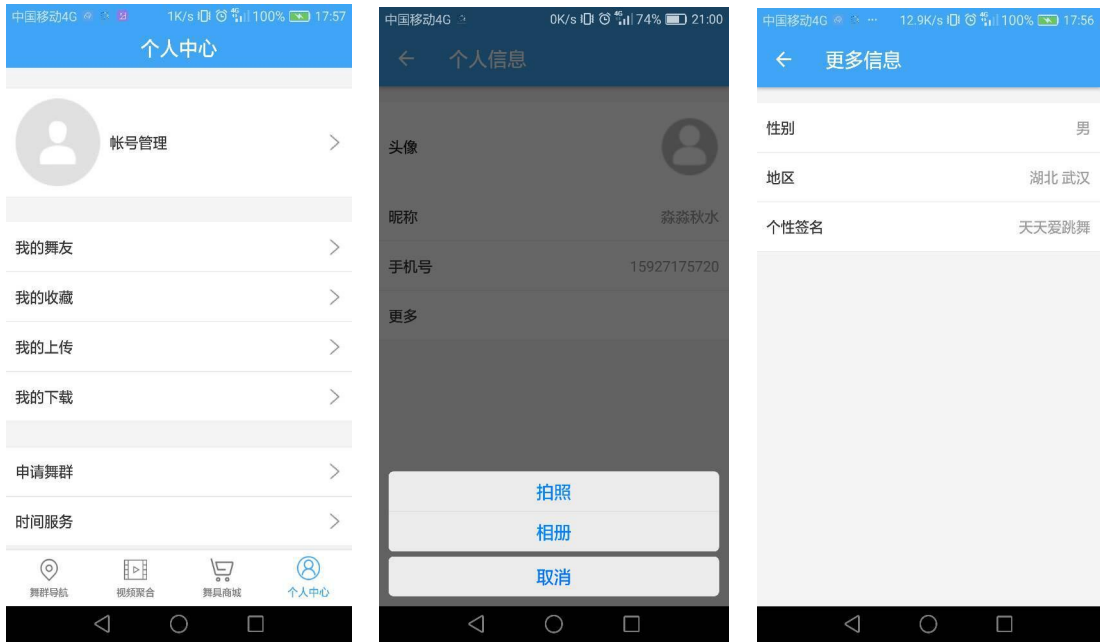


图 3-14 个人中心模块主体设计及用户具体信息展示界面

下面左边图是用户点击我的舞友一栏出现的首页界面；中间的图是点击首页群发按钮出现的界面；右边的图是点击首页舞友白水水出现的详情界面<sup>[18]</sup>。具体设计如下图 3-15 所示：

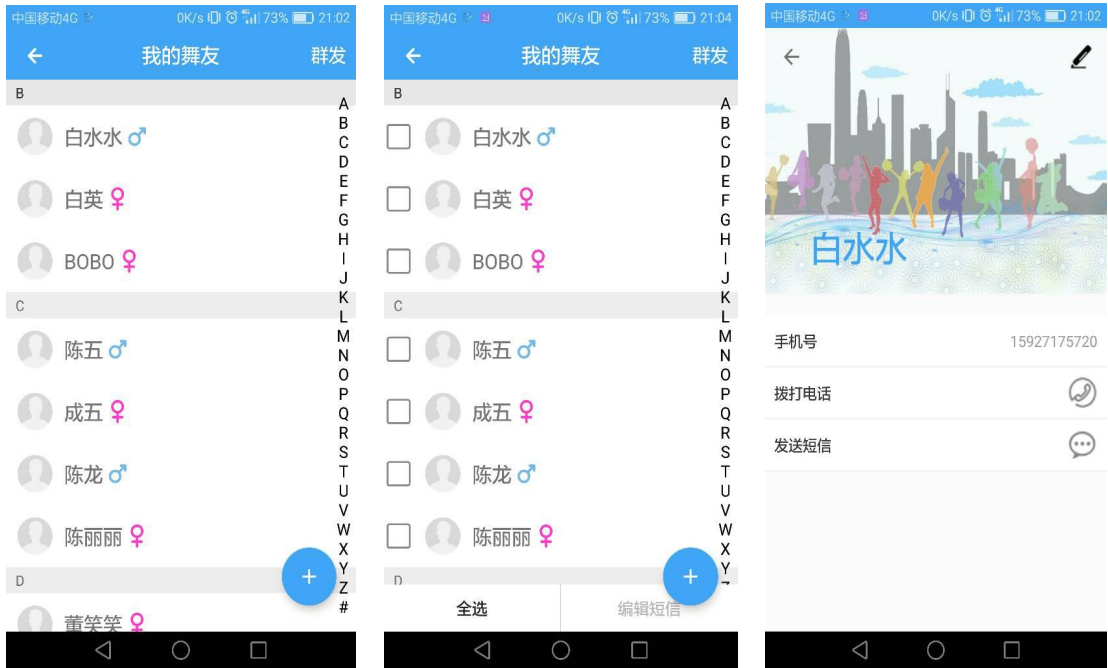


图 3-15 个人中心模块我的舞友信息查看设计界面

左边的图是点击我的舞友首页右下方+号按钮出现的界面；中间的图是点击舞友信息详情界面右上方修改按钮出现的界面；右边的图是点击舞友信息详情界面发送短信功能跳转出现的界面。具体设计如下图 3-16 所示：

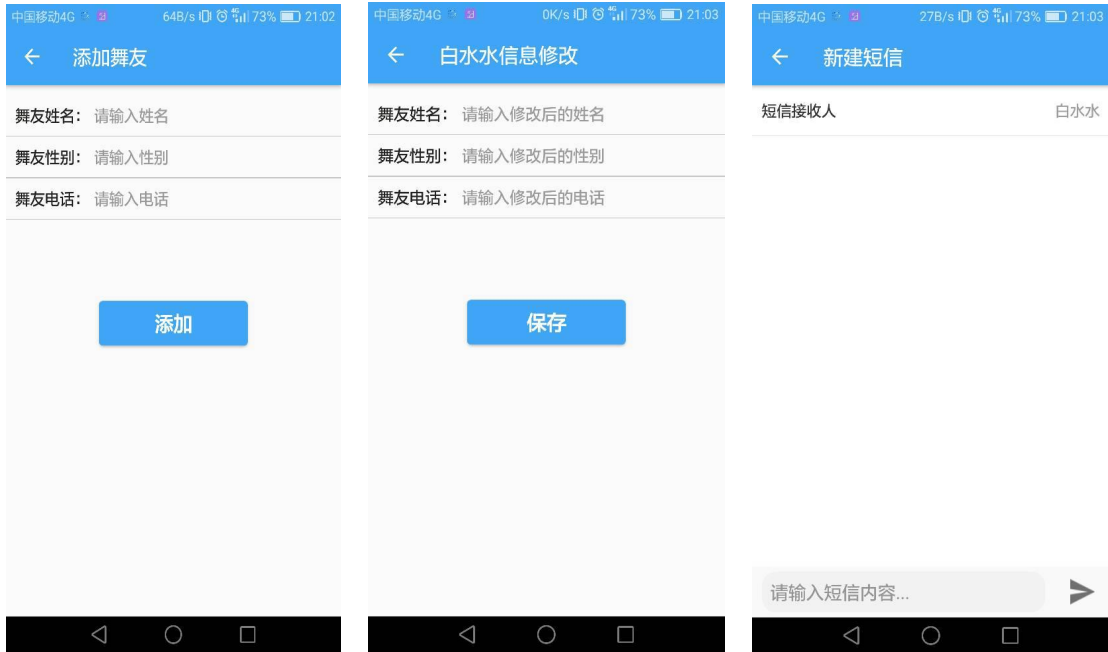


图 3-16 个人中心模块我的舞友添加修改信息以及编辑发送短信功能设计界面

下面左边的图是点击个人中心首页我的收藏出现的界面；中间的图是点击个人中心首页我的上传出现的界面；右边的图是点击个人中心首页我的下载出现的界面。具体设计如下图

3-17 所示:



图 3-17 个人中心模块我的收藏、上传和下载功能设计界面

下面左边的图是点击个人中心首页申请舞群出现的界面;右边的图是点击个人中心首页时间服务<sup>[19]</sup>出现的界面。具体设计如下图 3-18 所示:

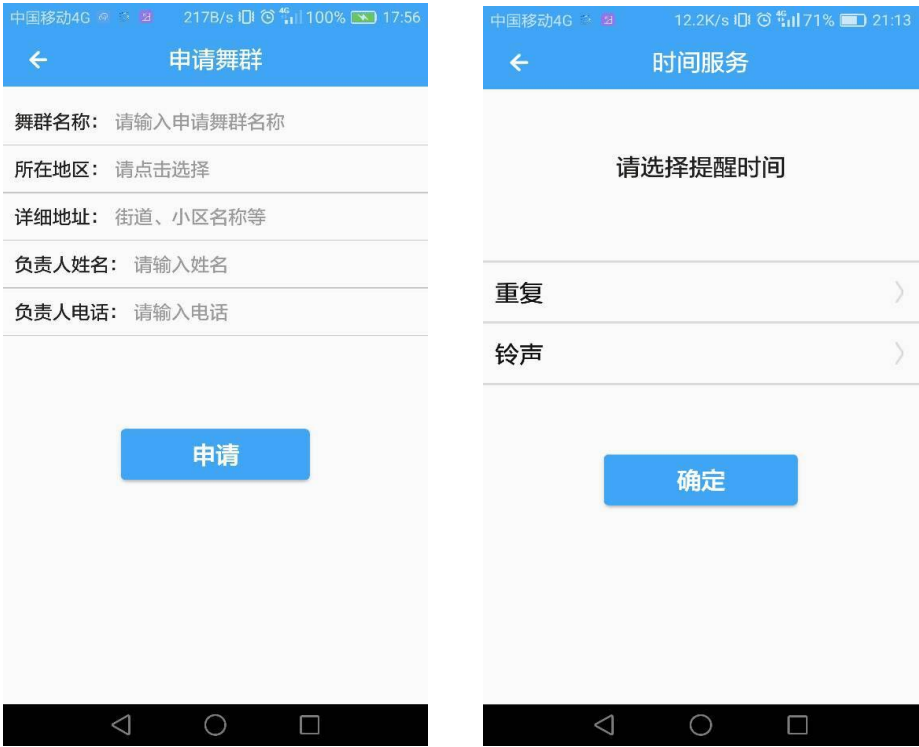


图 3-18 个人中心模块申请舞群和时间服务功能设计界面

下方左边的图是击图中隐藏标题按钮，会发现底部导航栏文体信息已经被隐藏；中间的图是点击途中滑动颜色按钮，会发现导航栏颜色已经发生改变；右边的图是点击途中滑动特效按钮，会发现导航栏当前图标变大，并且其它图标的标题已经被隐藏<sup>[20]</sup>。具体设计如下图 3-19 所示：

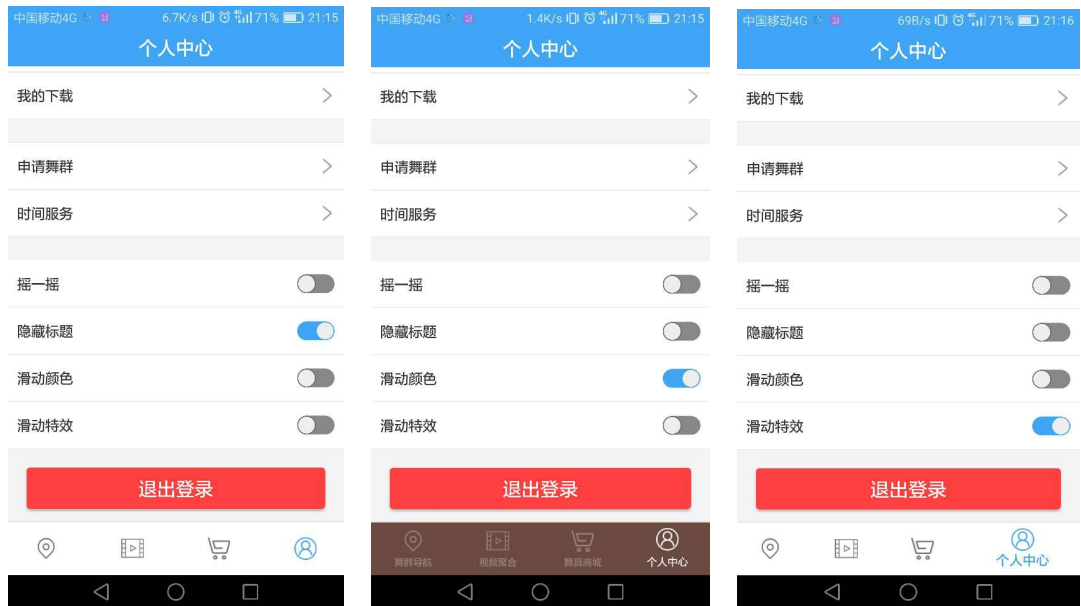


图 3-19 个人中心模块调节底部导航栏状态功能设计界面

(2) 管理员端界面

后台管理员端是基于 Web 端的系统，包括管理员登陆、注册和用户所需数据的五个集合增删改查功能的界面设计，其中进行数据的增删改查界面风格设计一致，下面只列出其中一个集合的添加的功能界面，其它集合信息界面设计类似<sup>[21]</sup>。具体设计如下图 3-20 所示：



图 3-20 管理员端首页功能设计界面

管理员点击系统首页右上方的注册按钮便可出现如下所示的界面。具体设计如下图 3-21 所示：

最新广场舞管理员端

首页注册登陆

欢迎进入注册界面

邮箱

用户名

联系电话

联系地址

密码

确认密码

注册

图 3-21 管理员端注册功能设计界面

管理员点击系统首页右上方的登陆按钮便可出现如下所示的界面。具体设计如下图 3-22 所示：

最新广场舞管理员端

首页注册登陆

欢迎进入登陆界面

邮箱

密码

登陆

图 3-22 管理员端登陆功能设计界面

管理登陆成功后，并且点击界面上方的评论信息按钮，会出现如下所示的界面。具体设计如下图 3-23 所示：



图 3-23 管理员端登陆成功查看评论信功能设计界面

管理员登陆成功后，点击首页舞群信息按钮，并点击该界面中添加舞群信息按钮，便可以出现如下所示的界面。具体设计如下图 3-24 所示：



图 3-24 管理员端登陆成功添加舞群信息功能设计界面

### 3.2.1 系统后端接口设计

本系统常用的几个功能接口设计，包括登陆、周边舞群信息和视频信息接口。此外，还包括注册、申请舞群、商品信息查看等功能接口设计与上述类似，限于篇幅就不详细说明。

### (1) 登陆接口

功能描述：用于用户端登陆系统。

调用方式：

表 3-2 登陆接口信息

登陆接口	
Key	Value
请求 URL	http://120.79.82.151/login
HTTP 请求方式	GET
支持格式	JSON
是否需要登录	是

返回值说明：

表 3-3 登陆接口返回值说明

登陆接口返回值说明		
参数名	类型	描述
name	String	返回信息
password	String	返回信息

返回值示例：

```
[
  {
    "name": "12350@qq.com",
    "password": "123456"
  },
  {
    "name": "dsff50@qq.com",
```

```

        "password": "dsaf"
    }
]

```

具体实现核心代码：

```

1 @app.route('/login', methods=['GET'])
2 def get_all_users():
3     userinfos = mongo.db.userInfo.find()
4     output = []
5     for s in userinfos:
6         output.append({'name' : s['name'], 'password' : s['password']})
7     return jsonify(output)

```

## （2）周边舞群信息接口

功能描述：用于在用户端地图上标注周边舞群具体信息，并且方便用户在线路导航功能中选择周边舞群。

调用方式：

表 3-4 周边舞群信息接口

周边舞群信息接口	
Key	Value
请求 URL	http://120.79.82.151/dancegroup_get_all
HTTP 请求方式	GET
支持格式	JSON
是否需要登录	是

返回值说明：

表 3-5 周边舞群信息接口返回值说明

周边舞群信息接口返回值说明
---------------



参数名	是否必选	类型	描述
dance_name	是	String	舞群名称
dance_region	是	String	舞群所在地区
detailed_address	是	String	详细地址

返回值示例：

```
{
  "dance_name": "美美舞群",
  "dance_region": "湖北省武汉市",
  "detailed_address": "洪山区中南民族大学南区操场",
}
```

具体实现核心代码：

```
1 @app.route('/dancegroup_get_all', methods=['GET'])
2 def get_all_dancegroups():
3     dancegroups = mongo.db.danceGroup.find()
4     output = []
5     for s in dancegroups:
6         output.append({'dance_name': s['dance_name'], 'dance_region': s['dance_region'],
7             'detailed_address': s['detailed_address']})
8     return jsonify({'result': output})
```

### （3）视频信息接口

功能描述：用于用户端热门推荐和分类教学具体视频信息内容展示，并且获取用户评论信息、点赞信息和收藏信息。

调用方式：

表 3-6 视频信息接口

视频信息接口	
Key	Value

请求 URL	http://120.79.82.151/api/upload
HTTP 请求方式	POST
支持格式	文件
是否需要登录	是

返回值说明：

表 3-7 视频信息接口返回值说明

视频信息接口返回值说明			
参数名	是否必选	类型	描述
myfile	是	文件类型	上传具体文件

返回值示例：

```
{
  "error": 0,
  "errmsg": "上传成功",
  "token": "MTUyMDQ4MzUxMy5tcDQ="
}
```

具体实现核心代码：

```
1 @app.route('/api/upload', methods=['POST'], strict_slashes=False)
2 def api_upload():
3     file_dir = os.path.join(basedir, app.config['UPLOAD_FOLDER'])
4     if not os.path.exists(file_dir):
5         os.makedirs(file_dir)
6     f = request.files['myfile']
7     if f and allowed_file(f.filename):
8         fname = f.filename
9         ext = fname.rsplit('.', 1)[1]
10        unix_time = int(time.time())
```

```

11         new_filename = str(unix_time) + '.' + ext
12         f.save(os.path.join(file_dir, new_filename))
13         token = base64.b64encode(new_filename)
14         return jsonify({"errno": 0, "errmsg": "上传成功", "token": token})
15     else:
16         return jsonify({"errno": 1001, "errmsg": "上传失败"})

```

### 3.3 数据库（存储）设计

MongoDB 是一种非关系型数据库，另一种比较贴切说法是文档型数据库，相对于关系型数据库，其在小型系统数据库设计和使用上很简单。一般情况下，只需要将所需数据分类存储在数据库中，不需要考虑实体间的关系。

在后台建立数据库和集合的过程中，不需要单独的建立数据库和集合的代码，只需要写出具体的查询、添加、删除和修改的操作语句，相关集合和数据库就会自动在 MongoDB 存储系统中被建立。

本系统中涉及了一个我的舞友功能，要求存储用户的朋友联系电话和姓名私密信息，在这个功能处理上，采用了 Andorid 系统自带的 SQLite 数据库进行存储。另外，关于舞具商城，用于定位用户当前所在地的地点信息，采用 XML 文件存储，在地点信息 XML 文件中保存了全国所有省市县级地区信息。

本系统结合本地存储和服务器存储相结合的方式，提高相关功能实用性的同时，也保证了用户的隐私安全。

#### 3.3.1 集合

##### （1）用户集合

包含域：编号（MongoDB 数据库默认自动生成，唯一确定一条文档记录），用户名，密码，昵称，手机号，头像，性别，地区，个性签名，权限。

##### （2）舞群集合

包含域：编号（MongoDB 数据库默认自动生成，唯一确定一条文档记录），舞群名称，所在地区，详细地址，负责人姓名，负责人电话，所在地经度，所在地纬度。

##### （3）视频集合

包含域：编号（MongoDB 数据库默认自动生成，唯一确定一条文档记录），视频名称，视频链接地址，视频首页图片，发布人姓名，获点赞数量，收藏用户名，上传者用户名，下载人用户名。

##### （4）评论集合

包含域：编号（MongoDB 数据库默认自动生成，唯一确定一条文档记录），评论视频名称，评论人昵称，回复人昵称，评论内容，评论人性别，评论人所在地，评论发布时间。

##### （5）商品集合

包含域：编号（MongoDB 数据库默认自动生成，唯一确定一条文档记录），店铺介绍，商品名称，商品重量，商品价位，商家联系电话，商家所在地，商品图片。

### 3.3.2 文档数据模型

MongoDB 数据库中集合包含多个文档，而文档中的信息，一般是参照集合包含的域信息来进行设计。即文档中，可以漏掉相关域信息，也可以添加新的域信息。下面的文档设计是按照 3.3.1 集合设计的域信息，刚好完全符合要求进行设计。文档数据的模型是 BSON 格式，和 JSON 格式类似。具体模型如下图 3-25 所示：



图 3-25 系统后端数据库文档数据模型图

### 3.3.3 具体代码映射实现

MongoDB 数据库集合中域不用完全一致，在一个集合中，不同文档的域名可以不一致，只要大体相同即可。另外，对于 MongoDB 数据库，其集合的建立和文档数据的插入也不需要特定的代码实现，只需要在运用时设定好文档的域名，并指定写入某个集合中就符合要求。若指定的集合不存在，系统会自动建立这一个集合，文档的信息写入也是同样的道理。

集合在管理端页面进行增删改查操作的具体域值建立映射关系,统一某一集合域名信息的同时,也方便了对数据的相关操作。具体参照如下表 3-8 所示:

表 3-8 MongoDB 数据库集合与模型层代码映射对照表

MongoDB 数据库集合与模型层代码映射对照	
集合名称	映射类名
用户集合	UserInfoForm
舞群集合	DanceGroupForm
视频集合	VideoForm
商品集合	GoodsForm
评论集合	CommentForm

限于篇幅,以下仅给出用户集合模型层映射代码示例,其它集合映射代码类似,就不在作描述。

```
1 class UserInfoForm(FlaskForm):
2     username = StringField(u'用户名', validators=[DataRequired()])
3     password = StringField(u'密码', validators=[DataRequired()])
4     nickname = StringField(u'昵称', validators=[DataRequired()])
5     telnumber = StringField(u'手机号', validators=[DataRequired()])
6     picture = StringField(u'头像', validators=[DataRequired()])
7     gender = StringField(u'性别', validators=[DataRequired()])
8     area = StringField(u'地区', validators=[DataRequired()])
9     signature = StringField(u'个性签名', validators=[DataRequired()])
10    submit = SubmitField(u'提交')
```

4 调试与分析

4.1 测试准则

(1) 确保系统的安全性

系统的安全性一般包括:数据机密性、数据存储和传输过程中的加密、以及防止数据窃取和窃听。该系统必须确保用户的访问控制:防止未经授权的资源使用或以未经授权的方式使用资源。身份真实性:确保被确认身份的身份已被验证。可用

性、可靠性：当系统退化和破坏时，系统可以继续完成其功能，从而在不利条件下尽可能少地受到侵略者的破坏。

## （2）确保系统的易用性

系统应具有良好的易用性。客户端具有良好的用户界面，方便用户使用查询系统。此外，系统需提供了一套周到的功能，以方便用户的良好互动。所有操作界面和方法均需同时具有简单的操作方式。只要用户输入相应的操作信息，就可以得到相应的结果，节省了时间，也满足了初始用户。同时，对于不同操作界面的所有输入要求都有不同的颜色提示、文本提示等。

## （3）确保系统的性能良好

为了使用户能够快速检索所需的信息，系统必须确保良好的响应时间和吞吐量。一方面，系统在处理每个用户的请求时需要得到保证。由于超时检查可能导致用户体验下降，并导致用户不信任客户端。另一方面，系统需要确保一定的吞吐量，并确保系统在高峰时段能够满足用户的需求。

## （4）确保严重性错误为零

系统中的严重错误不仅会大大降低用户体验，还会导致正在进行的项目出现重大损失。为了防止这种严重的错误发生，有必要在系统上线前对系统的核心功能模块进行严格的测试，并对相关的错误进行等级分类。一旦发现错误级别严重，在解决错误后，才能考虑系统的上线和维护功能需求。

# 4.2 测试目标

- （1）提高用户定位的准确性。
- （2）提高周边舞群搜索结果的精确度。
- （3）确保舞蹈视频上传功能稳定。
- （4）确保视频播放源和评论信息及时更新。
- （5）提高时间服务功能的准确度。

# 4.3 测试及结果

## 4.3.1 舞群定位和周边舞群模块

舞群定位功能实际使用过 50 组不同位置进行定位，经过和实际具体位置对比，误差范围在 30 米以内，定位性能服务功能设计的前期需求。网络信号程度保持 4G 百分之五十及以上程度，就可以正常返回定位数据，否则可能无法正常定位。

周边舞群显示图标功能，经过 50 组不同位置返回的舞群数据对比，显示都是以市级单位为限制，超出市级单位则不显示具体舞群地标信息，舞群位置地点和图标显示正常。但是一般的舞群实际地点和地图上标注的位置，一般有 300 米以内的误差，误差范围在周边舞群地位的允许范围内。

## 4.3.2 舞蹈视频上传模块

在 4G 一半信号程度、满格信号程度以及 Wifi 网络，并测试上传 50 组不同大小的视频。

测试结果显示，视频上传大小不能超过 100M，4G 满格信号和 Wiki 信号强的区域，视频都能达到快速上传效果；4G 一半信号程度或者 Wifi 信息程度较低，如果视频大小超过 50M，可能会因为网络原因可上传失败。此处视频上传模块性能，符合初步需求设计。

#### 4.3.3 舞蹈播放源信息及评论信息模块

用户登陆成功后，使用下拉刷新功能，可以成功得到刷新后的视频信息及相关评论信息。用户输入评论内容后，可以实时显示评论结果。此处测试结果达到预期目标。

#### 4.3.4 时间服务功能模块

使用过 50 组不同时间进行提醒时间和铃声方式设置，测试结果显示系统一般会时间到达后 30 秒之内进行提示，出现个别情况出现延迟 1 分钟时间才出现提示信息，铃声提示方式正常运行。此处功能准确度满足预期的目标。

## 结论

互联网的快速发展，Android 智能手机的广大普及和相关系统版本的迭代更新，都说明了我们的生活已经与互联网形成了密不可分的关系。本文结合当前国内广场舞大范围普及的情况，进行相关功能调研和设计，在为相关喜爱广场舞用户提供需求功能之外，也期望达到解决相关社会问题的目标。

为了满足预期目标，本系统重点在于设计实现周边舞群搜索和导航核心功能，以及舞蹈视频及时更新和上传下载功能，舞具商城、我的舞友以及时间服务功能。在实现的过程中，对于如何调用百度地图 SDK 实现用户定位和相关舞群地址反向解析，并显示在用户端手机地图上遇到了不小困难，但是最后通过反复查阅专业书籍和网上博客文章，成功的解决了这一难题。在实现时间服务功能模块，由于 Android 系统的升级对闹钟 API 的安全性进行了限定，导致相关方法调用在到了指定时间后不会出现提醒，这也是本次系统开发中遇到的问题，最后通过参考 GitHub 上相关优秀项目示例，最终也成功解决了这一问题。

最后，本文基本上实现了系统立项时预定的所有需求功能。由于时间原因，除了系统核心功能外，存在部分功能实现的效果可能不符合预期要求。但是，后续时间充裕时，我还会继续丰富完善本系统相关功能。

## 致谢

时光眨眼即逝，从刚入学时的懵懵懂懂，到现在对专业知识的熟悉运用，也让我深深喜欢上软件工程这一专业。

技术的学习是枯燥的，但是每当收到老师和同学们热心的帮助时，都会让我对技术的学习重新充满激情。很庆幸能够在大一下学期加入到我们学院新思路团队，不仅引导我进入专业技术学习的快车道，还让我认识了我现在的毕业设计指导老师孙翀老师，还认识了很多志同道合的同学，使得我的大学生活充满了许多乐趣。本文设计开发的系统的初步想法正是来自孙老师的提议，当时大二下学期我们数据库组通过孙老师的提议进行项目立项，并且以此报名参加计算机设计大赛，来驱动项目的进度，同时驱动组内同学进行相关技术方向的深入学习和动手实现。虽然，最后我们没能在计算机设计大赛中取得好成绩，但是通过这个项目

的立项，驱动着我进入了 Android 深入学习，也有了本文的初步模型。在大二下备赛期间，孙老师在百忙之中总是抽出时间来盯着我们的项目进度，并且还对我们的技术选型和功能设计提出了很多建设性建议，让我受益颇多。在本次毕业设计技术选择上，孙老师建议我采用基于 Python 的 Flask 框架进行后端接口服务搭建，让我深深感受到了 Python 语言的魅力，对我今后的技术选型有很大的帮助。在此，向可敬可爱的孙老师表示深深的感谢。

我还要感谢帖军老师以及在实验室帮助过我的老师和学长学姐，在我进入新思路实验室时给予的热心指导和帮助，也让我深深的喜欢上了新思路实验室，让我感受到了家一样的温暖。

最后，感谢一直默默支持我的父母、姐姐和期间帮助过我的同学和朋友。

## 参考文献

- [1]李刚.疯狂 Android 讲义（第 3 版）[M].北京：电子工业出版社，2015.
- [2]郭霖.第一行代码（第 2 版）[M].北京：人民邮电出版社，2016.
- [3]刘国柱，杜国威.Android 程序设计与开发[M].北京：清华大学出版社，2017:395-420.
- [4]李瑞奇.Android 开发实战从学习到产品[M].北京：清华大学出版社，2017:21-47.
- [5] Kristina Cbodorow.MongoDB 权威指南[M].北京：人民邮电出版社，2014.
- [6] Jon Duckett. JavaScript & jQuery 交互式 Web 前端开发[M].北京：清华大学出版社，2015.
- [7] Axel Rauschmayer.深入理解 JavaScript[M].北京：人民邮电出版社，2016:53-66.
- [8] Magnus Lie Hetland.Python 基础教程（第 2 版）[M].北京：人民邮电出版社，2014:26-88.
- [9]李勇，王文强.Python Web 开发学习实录[M].北京：清华大学出版社，2011:273-293.
- [10] Jack Stouffer.深入理解 Flask[M].北京：电子工业出版社，2016:96-126.
- [11] Miguel Grinberg. Flask Web 开发：基于 Python 的 Web 应用开发实战[M].北京：人民邮电出版社，2015.
- [12]谢英俊.Python 全栈开发实践入门[M].北京：电子工业出版社，2017:211-214.
- [13]刘长龙.Python 高效开发实战 Django Tornado Flask Twisted [M].北京：电子工业出版社，2016:270-326.
- [14]董伟明.Python Web 开发实战[M].北京：电子工业出版社，2016:28-88.
- [15] Ian G. Clifton.基于 Material Design 的 Android 用户界面设计[M].北京：电子工业出版社，2016.
- [16] Android 登陆页面 [EB/OL] . [2018-01-15] . <https://github.com/nanchen2251/AiYaSchool> Push.
- [17] Android 分享控件 [EB/OL] . [2018-02-01] . <https://github.com/xyzlf/ShareSDK>.
- [18] Android 通讯录设计 [EB/OL] . [2018-02-08] . <https://github.com/jiang111/SortList>.
- [19] Android 闹钟设计 [EB/OL] . [2018-02-10] . <https://github.com/loonggg/Android-AlarmManagerClock>.
- [20]Android 底部导航栏设计 [EB/OL] . [2018-01-08] . <https://github.com/aurelhubert/ahbottomnavigation>.
- [21]系统管理网页端页面设计 [EB/OL] . [2018-02-13] . <https://github.com/scotch-io/python-flask-crud-1>.