D{5}1A{N8WS26{M7`62$WXC copy

JIANGXI AGRICULTURAL UNIVERSITY

**本 科 毕 业 论 文（设 计）**



题目**：** **基于Spring多人烟花设计平台的开发与实现**

**学 院：** 软件学院

**姓 名：** 胡成

**学 号：** 20141923

**专 业：** 软件工程

**班 级：** 软件工程1409班

**指导教师：** 刘 毅 **职 称：** 助 教

**二0一八年 四 月**

目录

[摘要 I](#_Toc483904116)

[Abstract II](#_Toc483904117)

[1 绪论 3](#_Toc483904118)

[1.1 背景与意义 3](#_Toc483904119)

[1.2 现状和发展趋势 4](#_Toc483904120)

[1.3 研究方案 4](#_Toc483904121)

[2 可行性分析 6](#_Toc483904122)

[2.1 经济可行性 6](#_Toc483904123)

[2.2 技术可行性 6](#_Toc483904124)

[2.3 社会可行性 7](#_Toc483904125)

[3 需求分析 9](#_Toc483904126)

[3.1 系统目标 9](#_Toc483904127)

[3.2 功能分析 9](#_Toc483904128)

[3.2.1 用户信息管理 9](#_Toc483904129)

[3.2.2 用户认证 9](#_Toc483904130)

[3.2.3 主题模板 10](#_Toc483904131)

[3.2.4 切换多个主题 10](#_Toc483904132)

[3.2.5 导出 10](#_Toc483904133)

[3.2.6 发布 10](#_Toc483904134)

[3.3 系统功能结构 11](#_Toc483904135)

[3.4 系统性能要求 11](#_Toc483904136)

[3.5 功能描述和用例分析 12](#_Toc483904137)

[4 系统总体设计 24](#_Toc483904138)

[4.1 系统整体架构设计 24](#_Toc483904139)

[4.1.1 Nginx简介 25](#_Toc483904140)

[4.1.2 Redis简介 27](#_Toc483904141)

[4.1.3 Vue简介 27](#_Toc483904142)

[4.1.4 Zookeeper 27](#_Toc483904143)

[5 数据库设计 28](#_Toc483904144)

[5.1 数据库选择 28](#_Toc483904145)

[5.2 E-R模型图 28](#_Toc483904146)

[5.3 数据库表设计 29](#_Toc483904147)

[5.3.1 用户主题表userTheme 29](#_Toc483904148)

[5.3.2 模块表module 30](#_Toc483904149)

[5.3.3 模板样式表style 31](#_Toc483904150)

[5.3.4 初始资源表themeRes 31](#_Toc483904151)

[5.3.5 用户操作表userOperation 32](#_Toc483904152)

[6 详细设计 34](#_Toc483904153)

[6.1 控制层 34](#_Toc483904154)

[6.2 业务层 35](#_Toc483904155)

[6.3 持久层 35](#_Toc483904156)

[7 效果展示 36](#_Toc483904157)

[7.1 进入平台 36](#_Toc483904158)

[7.2 创建主题 36](#_Toc483904159)

[7.3 资源替换 37](#_Toc483904160)

[7.4 导出 38](#_Toc483904161)

[7.5 应用到手机 39](#_Toc483904162)

[参考文献 41](#_Toc483904163)

[致谢 42](#_Toc483904164)

# 摘要

小时候过年最开心的事情莫过于和家人一起看烟花，在爆竹声中迎接新年，和小伙伴们买些小爆 竹烟火一群玩耍。随着现在人们对环境卫生的越来越重视，烟花也放的越来越少，尽管这是一个主要原因，但烟花的‘花样’也就寥寥几种，这何其不是一个原因呢？就如今像在许多乡村，如我的家乡，一个小县城中，每有结婚，生日，新房做成等都爱放烟花。小时候也常常想，要是有那种可以自己设计出各种形状或者各种字的烟花形状该多好啊。随着现在社会的发展，也冒出了多种多样的小烟花，但也难以满足的喜爱，毕竟每个人的喜好不同，有些人希望自己买到能展现出自己想法的烟花。

正是这种情况下，我想开发出一个多人设计的烟花平台。其主要针对的用户是烟花的喜爱者，能够设计出更多受欢迎更好的视觉效果的烟花，由于该平台设计方式简便，能够通俗易懂的表达出自己的设计想法效果，所以一些游客也能够在该平台上注册后可以发布的自己的设计，让所有用户都可以参与进来设计。用户可以在浏览所有人设计的烟花爆竹的外形设计和想法，也可以在上面订做自己设计的烟花，设计好后拿到烟花释放就可以直接得到一个形状，放烟花后也能得到自己设计好的效果。

关键词：烟花；设计；平台网站。

# Abstract

Now the rapid development of mobile phone hardware, processor and memory don't seem to need the money by major manufacturers in the mobile phone mobile phone crazy, the price decreases, the performance also increased gradually. This is certainly a good thing, but it also allows users to produce new concerns. The mobile phone includes not only the appearance of mobile phone color value, also includes the UI page design of mobile phone, a cool cock fried days lock screen, a fresh theme, a love of the user desktop wallpaper can quickly attract users' attention. We have to ask, these topics come from their creation process the cumbersome and not cumbersome. Actually they are from some theme designer design. The process is quite complicated, the time period is very long, the designers of each month the theme of the creation is just a number. Of course, here we exclude some love copy and paste designer.

Consider from another angle, each side of the mobile phone users on the theme of the demand is great. According to the statistics of major mobile phone manufacturers the theme of business, Nikkatsu can reach millions, but the conversion rate is very low, indicating that the user demand is great, but to attract the user's theme is not much, users spend a lot of time to find their own love theme, but it is difficult to find. We cannot blame the designers do not good enough, say one thousand readers have one thousand Hamlett, and these users throughout all ages, their love is not the same, the aesthetic point of view is not the same. Behoove, designers cannot develop their love theme.

It is inefficient in the designer creative theme, a background user is also difficult to find their own love theme, our cloud platform design platform turned out. The user is based on the designer, can make everyone is a designer, because users can easily use the mouse. It is this the platform can not only make the designer efficient theme, also can make the ordinary users to easily create their own theme, for the user to save more time.

**Key words**: Cloud platform; mobile phone theme; designer; creative website

# 绪论

## 背景与意义

在我国历史上每当逢年过节，或大型庆典活动，人们都要燃放烟花以示庆祝， 这是中华民族的传统习俗。烟花的历史源远流长，既有美丽动人的传说，又有记载，有关烟花文化的诗、词、歌、赋、画历经千古至今仍传颂于世，例如《辛弃疾诗文选集》中的“蛾儿雪柳黄金缕，笑语盈盈暗香去。众里寻他千百度，蓦然 回首，那人却在，灯火阑珊处”。元人赵孟頫的千古名诗《赠放烟火者》中“人间巧艺夺天工，炼药燃灯清昼同。柳絮飞残铺地白”。可以说，从古到今，烟花艺术始终是人类不可离开的文化现象。

小时候过年最开心的事情莫过于和家人一起看烟花，在爆竹声中迎接新年，和小伙伴们买些小爆竹烟火一群玩耍。随着现在人们对环境卫生的越来越重视，烟花也放的越来越少，尽管这是一个主要原因，但烟花的‘花样’也就寥寥几种，这何其不是一个原因呢？就如今像在许多乡村，如我的家乡，一个小县城中，每有结婚，生日，新房做成等都爱放烟花。小时候也常常想，要是有那种可以自己设计出各种形状或者各种字的烟花形状该多好啊。随着现在社会的发展，也冒出了多种多样的小烟花，但也难以满足的喜爱，毕竟每个人的喜好不同，有些人希望自己买到能展现出自己想法的烟花。

正是这种情况下，我想开发出一个多人设计的烟花平台。其主要针对的用户是烟花的喜爱者，能够设计出更多受欢迎更好的视觉效果的烟花，由于该平台设计方式简便，能够通俗易懂的表达出自己的设计想法效果，所以一些游客也能够在该平台上注册后可以发布的自己的设计，让所有用户都可以参与进来设计。用户可以在浏览所有人设计的烟花爆竹的外形设计和想法，也可以在上面订做自己设计的烟花，设计好后拿到烟花释放就可以直接得到一个形状，放烟花后也能得到自己设计好的效果。

## 现状和发展趋势

烟花的形成和发展是历史的必然产物，烟花是一门综合艺术，站在时代的高度，结合汉文化从设计美学的角度对烟花艺术进行研究，对于提高我们艺术设计的素养，弘扬和创新现代设计和传统美学理论有着重要的作用。当今社会是一个更趋重于设计的时代，设计不仅成为一种满足实用性和功能性需要的行为，也是一种文化创造的行为，它在建构和美化人类的生存世界和生活世界中起着极为重要的作用。烟花设计作为设计文化的一个分支也不例外，纵观烟花设计的发展史， 充分展示着人们的智慧，映射着设计界的开拓博大。烟花是一种很成功的艺术化设计作品，对其设计美学特性进行系统的研究探索，不仅有利于对烟花进行更好的认识，同时将烟花这一独特的艺术提升到设计的高度，这对于促进烟花的发展， 促进设计的多元化具有重要作用。

烟花在人们精神生活中的重要作用是其他任何现代手段或方式所无法替代的，这不仅促进了烟花的生产和设计研究事业的蒸蒸日上，而且随着当今时代科学技术的蓬勃发展，烟花同样迎来了更加姹紫嫣红的新天地，它作为民族文化的化身和世代情感的寄托走过昨天，迈步今天，展望明天，以独特的魅力服务于社会和人民。

基于市场上的烟花种类，爆炸效果寥寥无几的现状，能够自己直接参与设计烟花的平台就应运而生。它主要是满足烟花爱好者能够在这个平台上进行设计，浏览他人的设计得到灵感，也能使所有游客看到他人的设计，从而加入进来一起设计。一千个读者眼里有一千个哈姆雷特，而且这些用户遍布各个年龄层，他们的爱好不一样，审美角度也不一样，所以也能更加促进烟花行业的发展。

平台不仅是对烟花爱好者提供服务，更希望引导用户自己设计出自己喜欢的烟花类型。

## 研究方案

本文主要研究的是基于java语言开发的烟花设计平台，主要是针对于用户进行设计自己喜欢的烟花类型，分为前台功能页面和后台管理系统两个模块。

前台功能系统主要是用户的登录和注册，作品的分类展示，作品详情页面的展示，用户设计作品，作品的删除，用户个人中心资料的修改、查看。

后台管理系统主要是对网站用户的统一管理，作品的上传、删除，作品单的管理，

用户信息的修改和删除等。

# 可行性分析

在如今互联网发展迅速的时代，我们可以发现大部分烟花销售的网站只是单纯的销售一些固定的样式。然而烟花的样式是烟花的呈现形式，必不可少的一部分，久而久之也会让用户产生审美疲劳，自己设计能提高用户体验。所有多人设计烟花类型的平台市场还是很有前景的。烟花爱好者或者一些购买烟花的用户可以自己设计出自己的喜欢的烟花类型，只需要一张设计好的图片和一段文字介绍，然后上传到一个平台供大家互相之间学习交流，也使其他人得到灵感，这样一个平台显得十分重要。

## 经济可行性

系统经济可行性，是从项目开发需要投入的人力物力以及时间成本出发，计算后期收回价值是否可以高出前期的投入价值，任何产品都需要保证自己的回收成本，才能保证后期提升产品的竞争力。

基于多人设计烟花类型平台，创作目标明确，旨在让烟花爱好者有一个平台可以在上面进行烟花类型的设计，在需求和功能明确的情况下，降低了项目更改需求的风险。

从开发角度分析，使用的是网上开源语言java，数据库myspl也是如此，工作量并不是很大，同时也不需要像游戏一样快速上线，所以，开发成本并不高。而从服务资源投入，该平台只需要几台性能较好的服务器，以及企业级的带宽，就能正常运作，所以前期投入成本不会很高。

## 技术可行性

从技术角度讲，该项目采用最常用的技术和架构。

开发语言选择的Java。Java作为当今世界最为热门的编程语言，与至今都被人追捧的C语言齐头并进，稳居TIOBE排行榜的第一，如图 2‑1所示。Java语言作为后起之秀，不仅吸收了C、C++的优点，摒弃了它们的缺点，同时也构建了自己的特点，在Web领域，占据着举重若轻的地位。

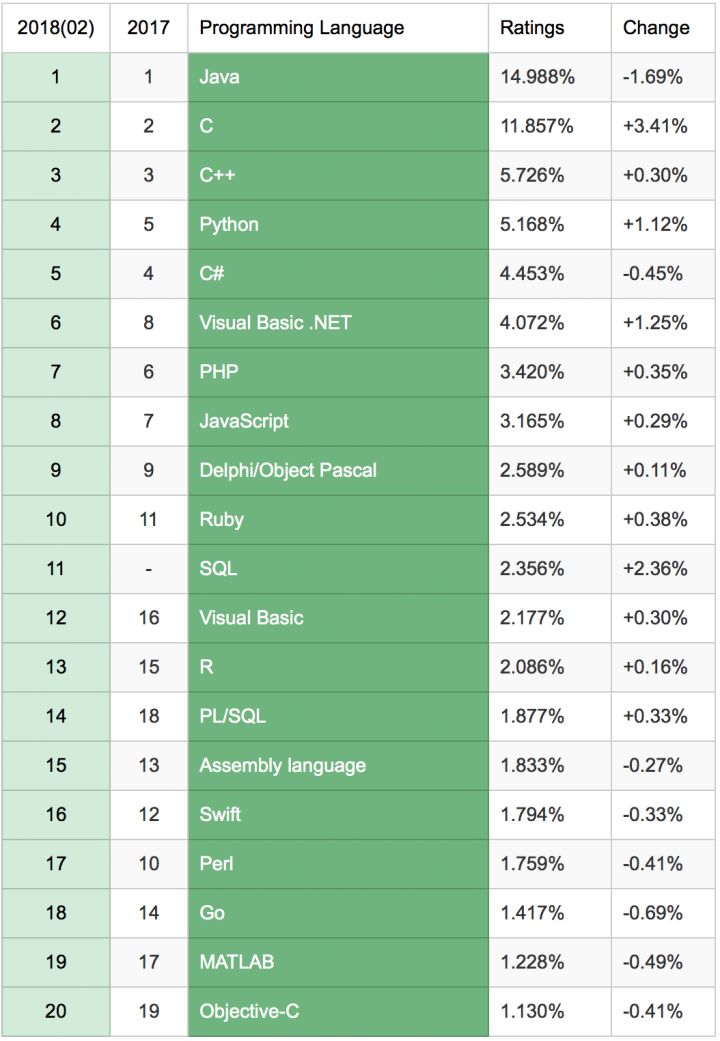


图 2‑1 TIOBE Index for February 2018

来源：TIOBE WEB SITE http://www.tiobe.com/tiobe-index/

控制层使用的框架是由Rod Johnson开发出的Spring框架。Spring是一个开放源代码的设计层面框架，他解决的是业务逻辑层和其他各层的松耦合问题，因此它将面向接口的编程思想贯穿整个系统应用。Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson创建。简单来说，Spring是一个分层的JavaSE/EE full-stack(一站式) 轻量级开源框架。

持久层使用的是MyBatis框架，MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，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。

数据库使用的是MySQL。MySQL是一个轻量级的数据库服务器，不像ORACLE、DB2那般笨重，性能却不输于它们，是中小型企业的最佳选择。MySQL上手简单，性能出众，不仅运行快速，稳定性也很高。从淘宝的数据库由ORACLE换为MySQL，相信MySQL的性能和实力毋庸置疑。

## 社会可行性

该平台主要是为了烟花爱好者能够设计出自己喜欢烟花类型，让用户能够在浏览他人作品的同时得到灵感进而创作。随着新的生活方式、生活观念形成新的审美价值观，新的文化意识导致新的流行与新的个性觉悟，在这样的文化语境下，烟花设计也在不断地更新换代，革故鼎新，以满足人们不断提升的审美心理。烟花是设计艺术反映社会生活的一种特殊形式，它不是纯精神产品的艺术形象，而是实用性和艺术性完美结合的艺术形象。烟花能调整人们的身心，使人焕发出新的活力，能凝聚丰富的感情，燃放烟花总能把人置于一种非功利性的身心愉悦之中, 人们的想象力和创造力得到了充分的发挥，烟花由此而具有了艺术与审美的品质，对于中国人来说，燃放烟花炮竹其实是一种非同寻常的文化活动，其存在是人们需求的折射，其作用是其他任何现代手段或方式所无法替代的。

该平台也能和一些出售烟花的商家合作，在他们的销售网站中加一个接口，通过这个接口，可以让用户在购买烟花时也可以进行烟花的设计或者浏览他人的烟花设计，从而达到定制烟花的作用。因为该平台操作简单，人人都可以成为烟花设计师，也能让商家和用户达到双赢。所有该平台很具有社会可行性。

# 需求分析

平常的设计平台需要很多繁琐的方式。用户前期需要花费大量时间学习设计，设计到中途的数据保存等问题，而该平台解决了这些问题。

## 系统目标

基于spring的多人烟花设计平台，满足多用户同时线上工作，进行烟花类型的设计，实时记录用户的作品，保存用户数据。

1. 页面美观且用户交互体验好。
2. 数据存储安全可靠。
3. 降低设计烟花的门槛，简化设计的变化，用户只关注主题的设计。
4. 实时保存用户作品，作品根据划分更便于寻找。
5. 用户无意的错误输入进行验证。
6. 平台的重塑性强，可进行功能的二次开发。

## 功能分析

基于spring的多人设计烟花平台具有高效，保存用户数据作品，用户操作简单快速，门槛低等特点。它具有以下功能。

### 用户信息管理

该功能是为了方便用户对个人信息的管理。用户成功登陆该平台后，可以查看自己的信息。所有信息只能自己看到，其他人无法查看，具有很好的隐私性。同时用户可以随时更改自己的信息，能够有效的保持用户信息的时效性和准确性。

### 用户认证

用户认证为平台提供了识别用户行为的基础。用户认证功能包含如下几点。第一提供用户注册功能，用户需要填写相关信息，注册成我们平台的用户，才可以开始他的创作之旅。第二提供用户登录验证功能，用户凭借账号与密码登录该平台，平台根据用户账号保存用户所有的信息和作品。

### 快捷设计

每个人都希望在刚接触到一个陌生的网站平台是很友好的，简单操作。基于这种情况，本平台简化了设计操作，用户可以事先准备好设计完的烟花图，也可以通过平台提供的一个简单画图的插件进行设计，用户只需要上传一张作品设计图和一段作品介绍和作品标题就完成了整个作品。

### 发布

该平台还提供发布功能，所有用户创作的作品设计都可以发布到该平台上，与他们共享。

## 系统功能结构

总的系统模块主要包括：用户信息管理模块，用户可以对自己的信息进行查看和修改；安全验证模块，用户可以进行注册，登录和切换账号等操作；管理员可以查看所有用户信息，用户的所有作品，增删改查所有用户信息，增删改查所有作品，查看作品的点赞浏览数量。详细模块图如图 3‑1所示。

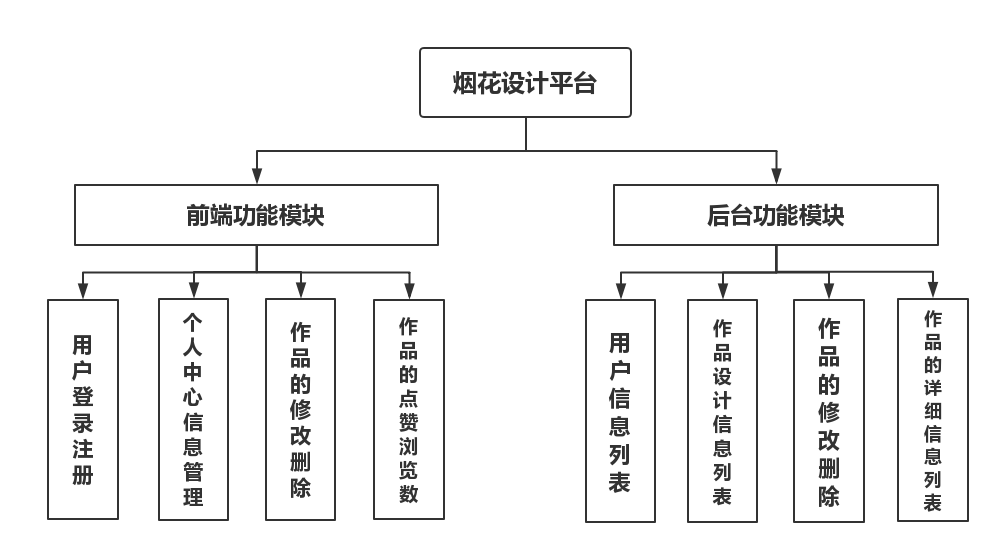


图 3‑1 系统模块图

## 系统性能要求

系统性能从平台本身和用户两个角度考虑。

从平台本身考虑，任何一个web站都需要做到安全性和稳定性。安全性着重强调的是对数据的保密性，用户的信息，用户的行为记录和用户个人数据都需要进行保密。同时加强自身站点的安全性，防止其他恶意入侵者入侵服务，盗取数据。防范XSS攻击，CSRF攻击，数据库注入等前端页面的攻击。同时，充分考虑系统容灾，只要不是服务全部挂掉，就会有服务对外提供服务。

从用户的角度出发，系统需达到快速响应，良好的体验两点。随着网速的增快，用户对网站的响应速度要求越来越高。平台需要做到常规请求下，响应速度不超过1秒钟。研究表明，用户对网站响应速度的容忍点是3秒，所以，对于费时需要大量资源的请求，最慢也要控制在3秒之内。良好的用户体验是站在使用者的立场，提供新手引导，指引用户使用。提供帮助手册，解决平台常见问题。布局友好，方便操作。

## 功能描述和用例分析

用例描述是需求分析时常用的与客户沟通交流的语言，成功解决客户不懂编程的专业术语，程序员不了解客户脑中所想的尴尬。用例描述了在某个特定的场景中，客户需要实现的功能，包含执行者和功能。用例描述可以大概描述出项目中的功能，以及这些功能的之前条件和产生的结果。

用例是是第二代面向对象技术的标志。用例是用来描述业务功能的，但用例图不仅仅是角色和用例的堆积，首先，用例是有层次概念的，一个大的用例可以用更小的多个用例来细化，直到无需再细分为止。根据分析下面列出本系统的系统用例图和各个用例的用例描述，项目总系统用例图如图 3‑2所示。

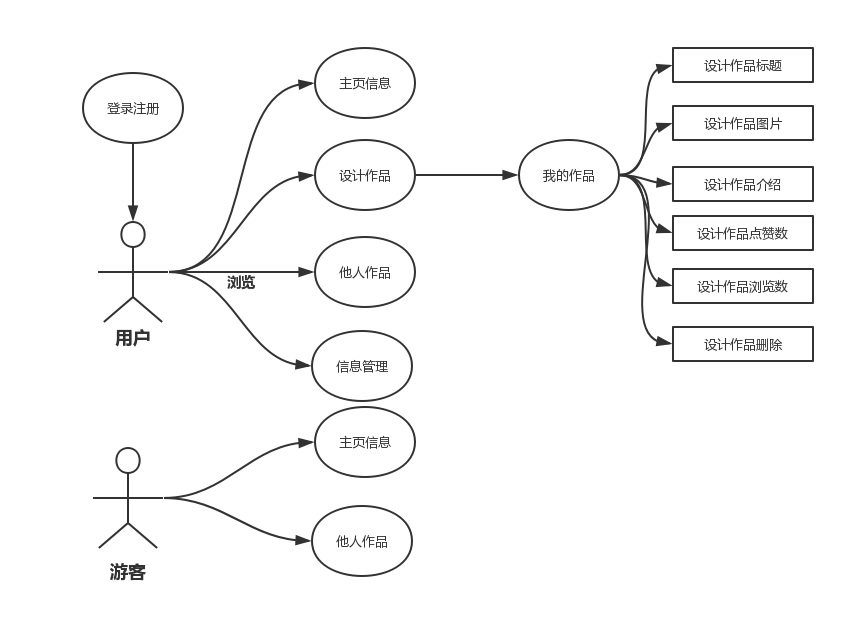


图 3‑2 系统总用例图

系统总用例图是对系统的总体概括，下面将会对每个模块进行详细介绍。信息管理模块包含两个功能，用户信息查看和用户信息修改，用例图如图 3‑3所示。

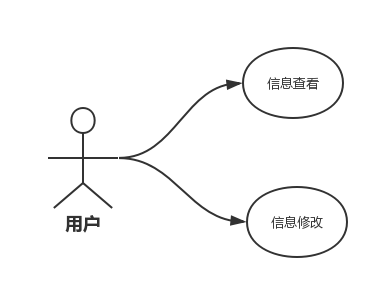


图 3‑3 用户信息管理用例图

信息查看功能提供入口给用户查看自己的信息。用户登录成功后，可以进入自己的详情页查看自己的信息，如自己的账号名称，手机号码，QQ号码，地址以及其他等隐私信息。该用例描述如表 3‑1所示。信息查看用例看似很简单却很重要，它是保证用户信息正确的关键。

表 3‑1 用户信息查看用例描述表

| 内容 | 说明 |
| --- | --- |
| 用例名称 | 用户信息管理。 |
| 用例说明 | 用户登录平台，进入个人主页，可以查看自己的信息。 |
| 参与者 | 用户。 |
| 前置条件 | 成功登入平台。 |
| 操作流程 | 1、用户成功登录进入系统，点击查看个人信息；  2、系统接收用户请求，根据用户唯一标识id查询用户信息：  1)、系统验证该账号id不存在(用户擅自修改浏览器中Cookie信息)，提示该账号不存在；  2)、系统检测账号存在，查询该账号信息，并返回到前台页面，用户跳转到个人信息详情页；  3)、系统内部出现错误，提示错误信息：“查询失败！系统内部出现错误！请联系管理员！”。 |
| 后置条件 | 成功返回用户的个人信息，并展示给用户。 |

信息修改功能提供入口给用户修改自己的信息。同样，该功能的前置条件也是用户已经成功登入到本平台，并且进入到个人信息详情页。用户点击信息内容，输入框便会显示出来，用户可以重新填写信息，并提交修改，平台会判断用户重新输入的信息内容是否符合要求，如果不符合要求，平台会返回错误信息，并显示在页面中。如果符合要求，则平台会更新数据库中用户的信息，并重新返回用户的信息。信息修改详细用例描述如表 3‑2所示。

表 3‑2 用户信息修改用例描述表

| 内容 | 说明 |
| --- | --- |
| 用例名称 | 用户信息修改用例。 |
| 用例说明 | 用户可以更新个人信息，或者发现信息有误进行修改。 |
| 参与者 | 用户。 |
| 前置条件 | 成功登入平台；进入个人信息详情页。 |
| 操作流程 | 1、用户成功登录到平台，并且已经进入到个人详情页；  2、用户点击自己想要修改的内容，被点击的地方会弹出输入框，用户可以重新填写信息，填写完成后点击修改按钮；  3、平台接收用户填写的数据，首先判断是否含有非法字符和敏感词汇，如果有则提示修改错误，并返回错误信息；其次判断内容是否合法，如判断手机号码的长度，邮箱的格式，如果格式不正确，依然提示修改错误，并返回错误信息。  4、平台更新数据库信息，如果更新失败，返回系统内部错误信息，如果更新成功，返回200状态码，以及成功提示。 |
| 后置条件 | 成功更新用户信息，并重新显示到页面。 |

安全验证模块是识别请求状态的基石。HTTP是一中无状态协议，服务端认为浏览器端每次请求都是一个新的请求，前一次请求与后一次请求没有任何关联，这导致服务器无法识别同一用户的多次请求。为解决上述问题，一般有两种方案：浏览器端采用Cookie和服务端采用Session；浏览器端采用Cookie，是将用户唯一标识放入浏览器中，服务端读取唯一标识，从而判断每次请求的用户。安全验证模块就是基于这个原理，平台会将标识用户的id和校验用户的Token放入浏览器Cookie中，从而维持用户的请求。安全验证模块用例图如图 3‑4所示，该模块主要提供登录、注册、退出3个功能。

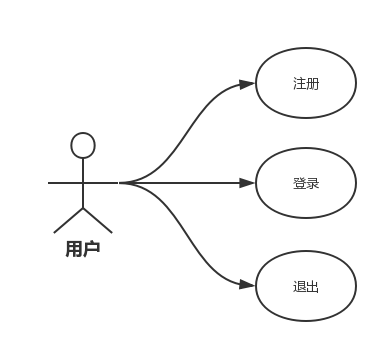


图 3‑4 安全验证模块用例图

注册功能是一个使普通用户成为能被该平台识别的用户的过程，其提供了一个完整的注册场景。用户需要填写注册账号，平台会检测该账号有没有被注册，如果被注册过，平台会提示用户改账号被注册过；如果没有被注册，用户需要填写验证码（用于防止程序恶意注册），然后点击下一步按钮，如果验证码错误，平台会提示验证码错误信息；如果正确则成功进入下一步，该步骤需要用户填写手机号码，然后平台会发送激活码（激活账号）到手机端；用户输入激活码，平台会验证激活码是否与发送给当前账号的激活码匹配，匹配成功则激活账号，如果不成功，则提示激活码错误信息。注册用例描述如表 3‑3所示。

表 3‑3 注册用例描述

| 内容 | 说明 |
| --- | --- |
| 用例名称 | 用户注册用例。 |
| 用例说明 | 注册用户。 |
| 参与者 | 用户。 |
| 前置条件 | 无 |
| 操作流程 | 1、用户输入注册账号，平台检测该账号是否被注册，被注册过，提示用户该账号被注册过，请输入其他账号名称。  2、用户输入验证码。  3、用户点击下一步，平台检测验证码，如果不匹配则提示错误。  4、用户输入手机号码，平台检测手机码号是否正确，错误则提示错误信息，正确则发送激活码到手机端。  5、用户填写接收到的激活码，平台取用户输入的激活码匹配平台中的激活码，如果匹配不成功，则提示激活码错误信息，如果匹配成功，则成功为用户注册该账号，并账号信息保存到数据库。 |
| 后置条件 | 注册成功，该账号有效；注册失败，返回失败信息。 |

登录功能是保持用户状态的前提，如上所述，用户未登陆状态下，服务端是无法识别当前请求是哪一个用户的，服务端也就无法完成与个人状态有关的请求。只有登陆成功后，每次请求中都带有唯一标识某个用户的id，服务端才有能力识别每次请求，详细的登录用例描述如表 3‑4所示。

表 3‑4 登录用例描述

| 内容 | 说明 |
| --- | --- |
| 用例名称 | 用户登录用例 |
| 用例说明 | 用户通过登录功能，让服务进行用户识别。 |
| 参与者 | 用户 |
| 前置条件 | 无 |
| 操作流程 | 1、用户填写账号和密码，点击登录按钮；  2、系统验证账号与密码是否匹配，首先系统验证该账号是否存在，如果该账号不存在，返回登录错误，并提示该账号不存在；  点击登录按钮  3、匹配账号和密码，如果不匹配，返回错误信息：用户账号与密码不匹配；如果匹配成功，将id、Token放入浏览器的Cookie中。 |
| 后置条件 | 成功进入平台主页，并将id与Token放入浏览器Cookie中。 |

退出用例与登录用例相反，当用户的请求不想被服务端识别可以使用退出，不过一般都不会这么做。退出是为了方便用户切换账号，在退出的情况下，用户可以重新登录其他账号。退出用例描述如表 3‑5所示。

表 3‑5 退出用例描述

| 内容 | 说明 |
| --- | --- |
| 用例名称 | 退出用例说明。 |
| 用例说明 | 用户退出系统，不在被系统识别。 |
| 参与者 | 用户 |
| 前置条件 | 登录成功 |
| 操作流程 | 1、用户点击退出按钮；  2、服务端清空浏览器中的cookie，如果失败，返回内部错误信息，如果删除成功，则跳转到登录页面。 |
| 后置条件 | 浏览器的Cookie信息被删除，跳转到登录页面。 |

烟花设计功能是方便用户设计出自己喜欢的烟花样式，当数量达到一定数量，还会分页显示。用例描述如表 3‑6所示。

表 3‑6 设计烟花用例描述

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例名称 | 主题列表用例说明 |
| 用例说明 | 用户查看主题列表 |
| 参与者 | 用户 |
| 前置条件 | 登录成功 |
| 操作流程 | 1、用户成功登入系统，点击主题列表按钮；  2、系统接收用户请求，获取Cookie中用户的id，查询该用户创建的主题：  1)、系统内部出现错误，提示错误信息：“主题列表查询失败！系统内部出现错误！请联系管理员！”；  2)、查询成功，返回数据，页面弹出主题列表信息浮层。 |
| 后置条件 | 返回主题列表信息 |

选择某个主题方便用户设计多个主题时相互切换。平台充分考虑到用户同时设计多个主题的情况，不会等到用户设计一个主题完成后，才能进行下一个主题设计。选择主题用例描述如表 3‑8所示。

表 3‑8 选择主题用例描述

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例名称 | 选择主题用例描述 |
| 用例说明 | 用户选择某个主题，切换到选择的主题环境中 |
| 参与者 | 用户 |
| 前置条件 | 登录成功 |
| 操作流程 | 1、用户成功登入系统，点击主题列表，系统会显示出所有的主题列表，用户双击某一个主题；  2、系统接收用户请求，获取主题的id，检测该主题是否存在：  1)、系统检测该主题不存在，提示错误信息：“选择主题失败！你选择的主题当前不存在！”；  2)、系统检测该主题存在，系统切换当前用户操作主题的状态，并提示选择主题成功信息。 |
| 后置条件 | 系统更新用户当前操作的主题信息 |

导出方便用户在设计主题时随时查看效果。传统的做法，用户需要整理主题包里的每个文件夹，并按照层级要求和命名要求统一规划，还需要手动逐层打包，相当繁琐。该系统将代替人工自动打包，并生成一张二维码供客服端下载。详细的导出用例描述如表 3‑9所示。

表 3‑9 导出用例描述

| 内容 | 说明 |
| --- | --- |
| 用例名称 | 导出用例描述 |
| 用例说明 | 导出主题 |
| 参与者 | 用户 |
| 前置条件 | 登录成功 |
| 操作流程 | 1、用户成功登入系统，用户点击导出按钮；  2、系统接收用户导出请求，系统根据Cookie中用户id查询用户当前制作主题的信息，包含主题名称，主题可打包状态，并弹出导出浮层：  1)、主题名称为空，导出浮层中主题名称显示默认主题名称-“我的主题”；  2)、主题名称不为空，导出浮层中显示主题的名称；  3)、主题可打包，浮层显示导出按钮；  4)、主题不可打包，浮层显示提供下载的二维码；  3、用户填写主题名称，点击导出按钮；  4、系统接收请求，判断主题名称是否合法，如果不合法，提示用户主题名称不合法，请重新填写。如果合法，则进行打包操作。  5、系统对当前主题进行打包，如果打包失败，则提示信息：“导出失败！主题打包过程出错！”；如果打包成功，则浮层会重新刷新，显示一张二维码，供用户下载。 |
| 后置条件 | 生成一张二维码 |

下载功能可以用户将制作中的主题下载到手机上，而不需要数据线。并且下载地址不需要手动记录，根据平台提供的二维码就能实现下载。下载用例描述如

表 3‑10所示。

表 3‑10 下载用例描述

|  |  |
| --- | --- |
| 内容 | 说明 |
| 用例名称 | 下载用例说明 |
| 用例说明 | 客户端下载主题包 |
| 参与者 | 客户端 |
| 前置条件 | 客户端连上互联网 |
| 操作流程 | 1、客户端扫描二维码；获取二维码中下载链接，发送下载请求；  2、系统接收客户端请求，根据请求中参数，查找相应的内容：  1)、请求内容不存在，提示错误信息：“下载失败！你请求的内容不存在！请更换下载内容！”  2)、请求内容存在，响应客户端，转成输出流，发送给客服端。 |
| 后置条件 | 客户端得到主题包 |

# 系统总体设计

## 系统整体架构设计

基于J2EE应用的B/S架构的三层结构，将系统划分为表示层、应用层(控制层与业务层)和数据层，如图 4‑1所示。

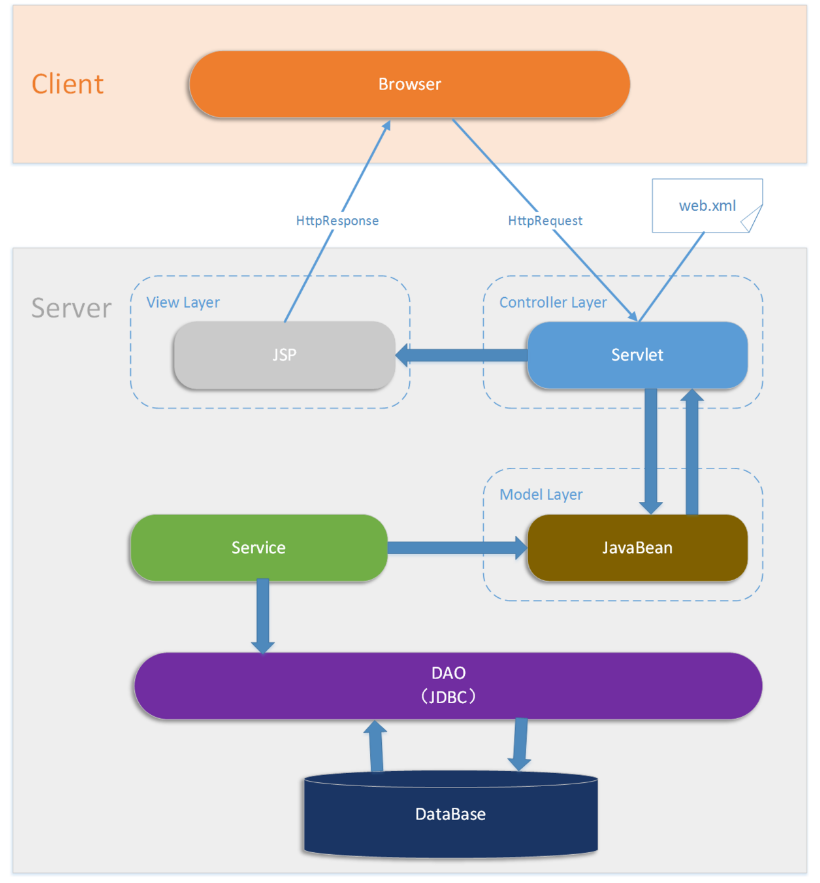


图 4‑1 J2EE体系MVC架构图

来源：https://www.debugrun.com/a/JnUxHgq.html

基于J2EE轻量级容器开发是对传统B/S的三层结构的延伸和拓展，它有优点也有缺点，如表 4‑1所示。

表 4‑1 J2EE优缺点比较表

|  |  |
| --- | --- |
| 优点 | 缺点 |
| 侵入性低 | 大吞吐量问题 |
| 对容器的依赖小 | 所有请求数据都需要经过数据库，增加了数据库的压力，也浪费很多不必要的时间 |
| 可配置 | Jsp页面与后台绑定，难以维护， |
| 通用性好 |  |

基于上述问题，该项目在传统的架构之上做了一些优化和改进。主要包含如下几点：在控制层之上(即服务接受到http请求时)，添加nginx；常用数据放入缓存中，降低数据库的压力和访问量，能大大提升服务的响应速度。这里我们选择的缓存服务是redis；采用前后台分离技术，前段页面也建立一套属于自己的服务，不依赖后台代码。通过接口进行连接。这里我们使用的是vue。以下我们将对这些技术或框架机型详细的介绍。

### Nginx简介

Nginx[5]是一款由俄罗斯人Igor Sysoev编写的轻量级Web服务器，性能卓越，安装简单，使用方便，可灵活配置。Nginx常用于反向代理和负载均衡。

首先谈谈正向代理，常用的正向代理如vpn，用户需要手动配置代理服务的地址，从而到达翻墙或者隐藏本地真实ip地址的目的，如图 4‑2所示。

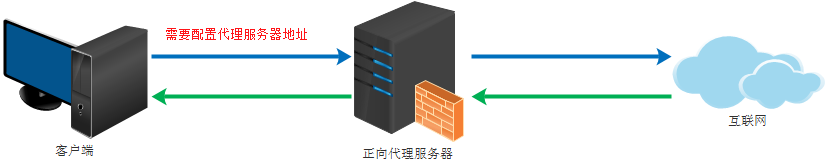


图 4‑2 正向代理示意图

反向代理与其相反，反向代理能隐藏服务器的真实地址，通过代理服务对外进行服务，这样可以起到安全的作用，如图 4‑3所示。

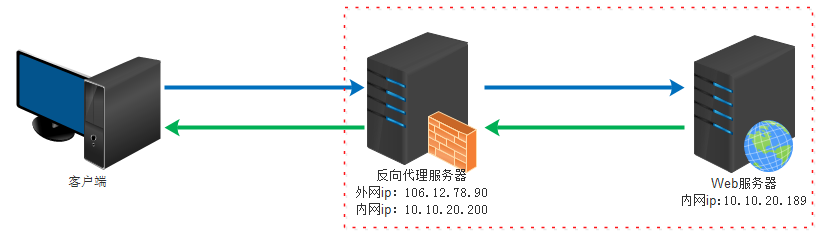


图 4‑3 反向代理示意图

负载均衡是在反向代理的机制之上进行优化，当服务器有大吞吐量是，Nginx通过算法将请求均匀分摊到多个服务器上，为单个服务器减压，负载均衡讲究的是均匀，避免单个服务器cpu占用很高，其他服务器cpu空闲。负载均衡使用图如图 4‑4所示。

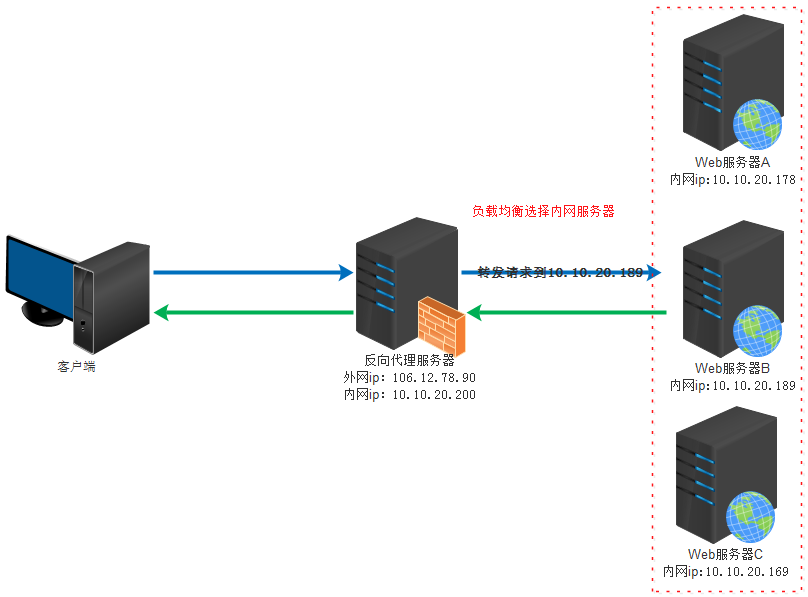


图 4‑4 负载均衡示意图

### Redis简介

Redis(Remote DIctionary Server)是一款Salvatore Sanfilippo编写的高性能Key-Value存储系统。和Memcached类似。Redis支持一些常用数据结构，如Lists、Sets、Hashes和Ordered Sets等。

Redis优点很多，性能高，每秒的读写频率超过100K+;丰富的数据类型，如Sets、Strings、Lists、Hashes等等；原子性，Redis的所有操作都是原子性，不可再分割；丰富的特性，支出publish/subscribe、key过期等；还支持不同方式的排序。

### Vue简介

Vue.js作为一款当前非常流行的前端框架，已经快速流入前端工程师手中，被各大公司使用。相比于Angular.js，Vue.js是一款更加简洁、更便于理解的前端语言，使开发人员更容易上手。Vue优点如下：

1. Vue支持局部CSS，只需要在<style>标签上增加scoped属性。并且不会像单纯的选择器嵌套那样渗透到它包含的子组件中。
2. 每一个Vue组件最终都被编译为纯粹的JavaScript模块，并且不需要任何浏览器polyfill即可支持到最低IE9。
3. <script>标签默认支持ES2015。
4. 随意在每一语言块中使用你想用的预处理器。

### Zookeeper

该项目的服务调用基于Zookeeper，Zookeeper是一款解决分布式的协调服务。分布式常见的问题就是数据的同步性，如服务的配置，服务的命名，如果其中一台更改的话，其他服务也必需更改，这对于维护很不便利。还有的是开发人员不能很好的使用同步锁，导致数据不同步，或者出错，这位程序带来很大的隐患，使用Zookeeper的目的便是解决这些问题。

# 数据库设计

数据库设计是在特定的项目环境中，设计满足该项目特点的数据结构建立相应的数据库系统，从而达到快速处理，高效存贮。数据库设计的最终目标是设计适合该系统的数据结构，性能最优的数据结构并不一定满足项目中的业务逻辑，满足项目业务逻辑的数据结构方便项目的开发，有利于数据的维护，有利于数据的分析。数据库设计是软件系统开发和建设的关键组成部分之一，所以过程也比较复杂，要想达到最佳效果不可能一蹴而就，需要设计者反复探索，不断优化。

## 数据库选择

数据库选择的MySQL，MySQL作为一款轻量级的数据库服务器，不像ORACLE、DB2那般笨重，性能却不输于它们，是中小型企业的最佳选择。

## E-R模型图

ER模型图是对数据库中实体类进行关系建模，使开发人员能一眼看出实体间的相互关系，便于数据库设计。E-R图由表示一个客观存在的事物实体，描述实体之间的关系的联系，以及对实体的特性描述的属性组成。

从交互的流程谈谈本项目中的实体。当用户从浏览器端发送请求时，服务器为了识别用户，所以需要设计一个用户实体；用户开始制作的第一步是选择一个主题模板，所以需要设计一个模板样式实体；当然这些样式里面有许多资源，并且这些资源对应不同的模块，所以还需要设计模块实体和资源实体；当选择完成后，便已经产生了一个原始的主题包啦，所以还需要设计一个主题实体；其次，用户的操作记录都需要记录，便于系统的维护，所以还需要设计一个记录实体。系统中还有一些其他的实体设计，只是不是主要的部分，便不一一介绍，这里只描述一下重要部分的E-R模型，其图如图 5‑1所示。

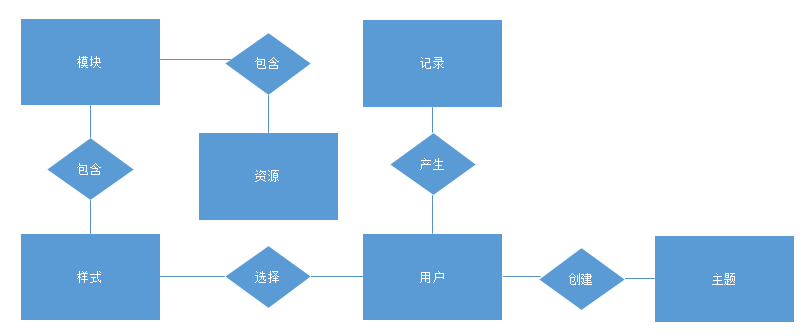


图 5‑1 基于云平台设计师创作平台E-R模型图

## 数据库表设计

数据库表有用户表、主题表、模块表、样式表、资源表、用户记录表、翻译表等等。这里贴出部分的表设计。

### 用户主题表userTheme

用户主题表用于记录用户创建的主题信息。所以需要用户编号和主题唯一编号和创建的时间，同时需要记录用户创建主题时关联的模板，所以需要styleId字段。详细表结构描述如表 5‑1所示。

表 5‑1 用户主题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 字段约束 | 对应中文名 |
| id | Bigint | 20 | Not null | 表id |
| userId | Bigint | 20 | Not null | 用户唯一编号 |
| themeId | Bnary | 16 | Not null | 主题唯一编号 |
| styleId | Binary | 16 | Not null | 模块编号 |
| createTime | Bigint | 20 | Not null | 主题创建时间 |
| activeMap | Varchar | 1100 | Not null | 模块激活状态 |
| skipMap | Varchar | 1100 | Not null | 模块跳过状态 |

### 模块表module

模块表用于对主题的细分。一个完整的主题是由许多个模块构成，这样使主题更有多样化。每个模块有唯一的编号和名称以及显示顺序，同时采用两级模块描述，使模块定为更加准确，所以还需要parentId记录模块的父子关系。详细的模块表结构描述如表 5‑2所示。

表 5‑2 模块表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 字段约束 | 对应中文名 |
| id | Int | 11 | Primary key | 表id |
| moduleId | Binary | 16 | Not null | 模块唯一编号 |
| modulePath | Varchar | 50 | Not null | 模块对应位置 |
| moduleName | Varchar | 50 | Not null | 模块名称 |
| moduleOrder | Int | 11 | Not null | 模块排序 |
| parentId | Bigint | 20 | Not null | 父模块id |

以下是建表语句：

CREATE TABLE `module` (

`id` int(11) NOT NULL DEFAULT '0',

`moduleId` binary(16) NOT NULL,

`modulePath` varchar(50) NOT NULL,

`parentId` bigint(20) DEFAULT NULL,

`moduleOrder` int(11) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `idx\_module\_id` (`moduleId`),

KEY `idx\_module\_path` (`modulePath`),

KEY `idx\_moduleOrder` (`moduleOrder`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

### 模板样式表style

模板样式表记录了木块的模板信息，比如一个锁屏模块有许多个样式，如上滑解锁，左滑解锁等。模板样式详细表结构描述如

表 5‑3所示。

表 5‑3 模板样式表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 字段约束 | 对应中文名 |
| id | Int | 10 | Primary key | 表id |
| styleId | Binary | 16 | Not null | 模板唯一编号 |
| moduleId | Int | 11 | Not null | 模块唯一编号 |
| styleUrl | Varchar | 255 | Not null | 模板路径 |

以下是建表语句：

CREATE TABLE `style` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`styleId` binary(16) NOT NULL,

`moduleId` int(10) unsigned NOT NULL,

`styleUrl` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

KEY `idx\_assembly\_url` (`styleUrl`),

KEY `idx\_assembly\_id` (`styleId`)

) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=utf8

### 初始资源表themeRes

初始资源表记录主题模板中原始的资源信息，需要记录资源文件的文件类型，大部分是图片，需要记录文件的长度和宽度，图片名称和路径，已经对应的模块和样式。详细的资源表结构描述如表 5‑4所示。

表 5‑4 资源表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 字段约束 | 对应中文名 |
| id | Bigit | 20 | Primary key | 表id |
| resourceId | Binay | 16 | Not null | 资源唯一编号 |
| styleId | Int | 11 | Not null | 模板编号 |
| picWidth | Varchar | 10 | Not null | 资源图片长度 |
| picHeight | Varchar | 10 | Not null | 资源图片高度 |
| picType | Varchar | 50 | Not null | 资源图片格式 |
| resourceUrl | Varchar | 255 | Not null | 资源路径 |
| resOrder | Tinyint | 3 | Not null | 显示位置排序 |
| moduleId | Int | 11 | Not null | 所属模块id |
| uploadTime | Bigint | 20 | Not null | 上传时间 |

以下是建表语句：

CREATE TABLE `themeRes` (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`resourceId` binary(16) NOT NULL,

`styleId` int(11) NOT NULL,

`picWidth` varchar(10) NOT NULL,

`picHeight` varchar(10) NOT NULL,

`picType` varchar(50) NOT NULL,

`resourceUrl` varchar(255) NOT NULL,

`resOrder` tinyint(3) NOT NULL,

`moduleId` int(11) DEFAULT NULL,

`uploadTime` bigint(20) NOT NULL,

PRIMARY KEY (`id`),

KEY `idx\_res\_url` (`resourceUrl`)

) ENGINE=InnoDB AUTO\_INCREMENT=158 DEFAULT CHARSET=utf8

### 用户操作表userOperation

用户操作表记录用户的操作记录，记录用户对哪些模块和哪些资源进行过操作，哪些没有进行操作。通过操作记录计算用户完成的模块和未完成的模块，计算用户的制作进度。详细的操作表结构描述如表 5‑5所示。

表 5‑5 用户操作表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 字段约束 | 中文名 |
| id | Bigit | 20 | Primary key | 表id |
| operationTime | Bigint | 20 | Not null | 操作时间 |
| themeId | Binary | 16 | Not null | 主题编号 |
| styleId | Binary | 16 | Not null | 模板样式编号 |
| resourceId | Binary | 16 | Not null | 资源编号 |
| replaceFlag | Int | 5 | Not null | 操作标记 |
| moduleId | Binary | 16 | Not null | 模块编号 |

以下是建表语句：

CREATE TABLE `userOperation` (

`id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`operationTime` bigint(20) NOT NULL,

`themeId` binary(16) NOT NULL,

`styleId` binary(16) NOT NULL,

`resourceId` binary(16) DEFAULT NULL,

`replaceFlag` int(5) DEFAULT '1',

`moduleId` binary(16) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `idx\_operation\_time` (`operationTime`),

KEY `idx\_theme\_id` (`themeId`),

KEY `idx\_assembly\_id` (`styleId`),

KEY `idx\_res\_id` (`resourceId`),

KEY `idx\_moduleId` (`moduleId`)

) ENGINE=InnoDB AUTO\_INCREMENT=796 DEFAULT CHARSET=utf8

对应数据库表的设计就介绍到这里。

# 详细设计

本项目采用的是基本的controller/service/dao项目结构。控制层调用业务层接口，业务层调用持久层接口，不能反向调用，也不能跨层调用。

## 控制层

控制层接收用户请求，然后调用相应的业务层接口，从而获得数据并返回给前端页面，它对应的是项目中的整个controller层。控制层类图设计如图 6‑1所示。

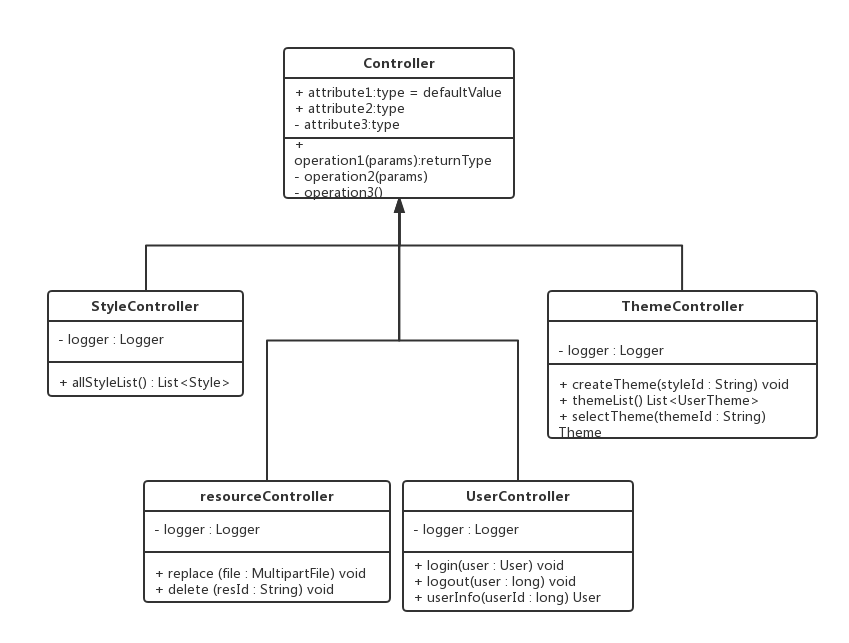


图 6‑1 控制层类图

## 业务层

业务层主要负责对数据的处理。业务层需要多次调用持久层的接口，将得到的数据重新组装成用户需要的数据。业务层类图如图 6‑2所示 。

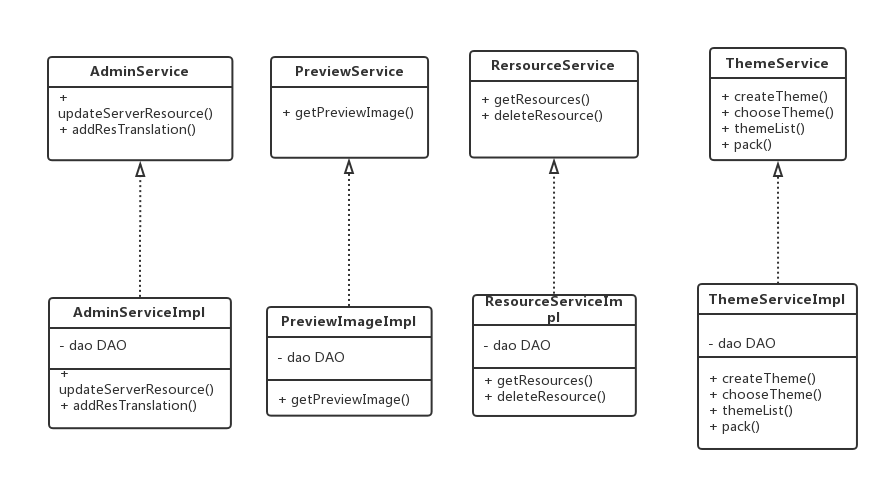


图 6‑2 业务层类图

## 持久层

持久层是负责向一个或多个数据存储器中存储或者获取数据的一组类和组件。这个层必须包括一个业务领域实体的模型。持久层类图如图 6‑3所示。

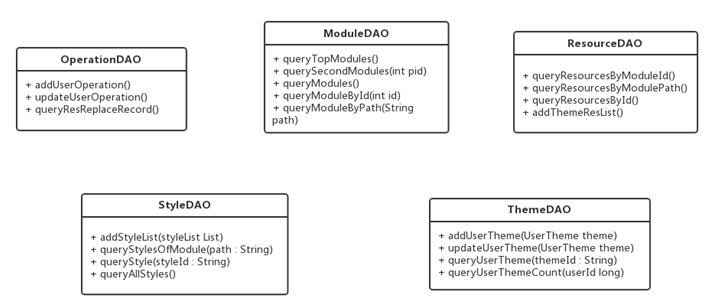


图 6‑3 持久层类图

# 效果展示

本章将从最终的效果展示出发，让大家对项目的功能有更清晰的认识，从进入平台、创建主题、替换资源、导出这样的一个操作顺序来演示系统的主要功能。

## 进入平台

用户进入平台主页见到的内容如图 7‑1所示。右上角有个登录按钮，点击登录，便跳转到登录页面，如果用户没有账号的话，在登录页面有注册账号的入口。用户输入用户名和密码，认证成功后，便会再次回到主页，同时右上角的登录按钮会消失不见，显示内容为当前用户的昵称和头像。左上角能切换语言，目前之前中文，繁体中文，英文等。而最中间则是开始制作入口，点击该按钮，便能进行主题制作啦。



图 7‑1 基于云平台创作平台首页

## 创建主题

在主页内，用户点击开始制作按钮便能进行主题创作啦。点击开始制作按钮，会为用户呈现系统提供的一些主题模板，模板有相应的描述。用户选择喜欢的模板，系统会将该模板的资源复制到用户的数据区。模板页面见图 7‑2所示。

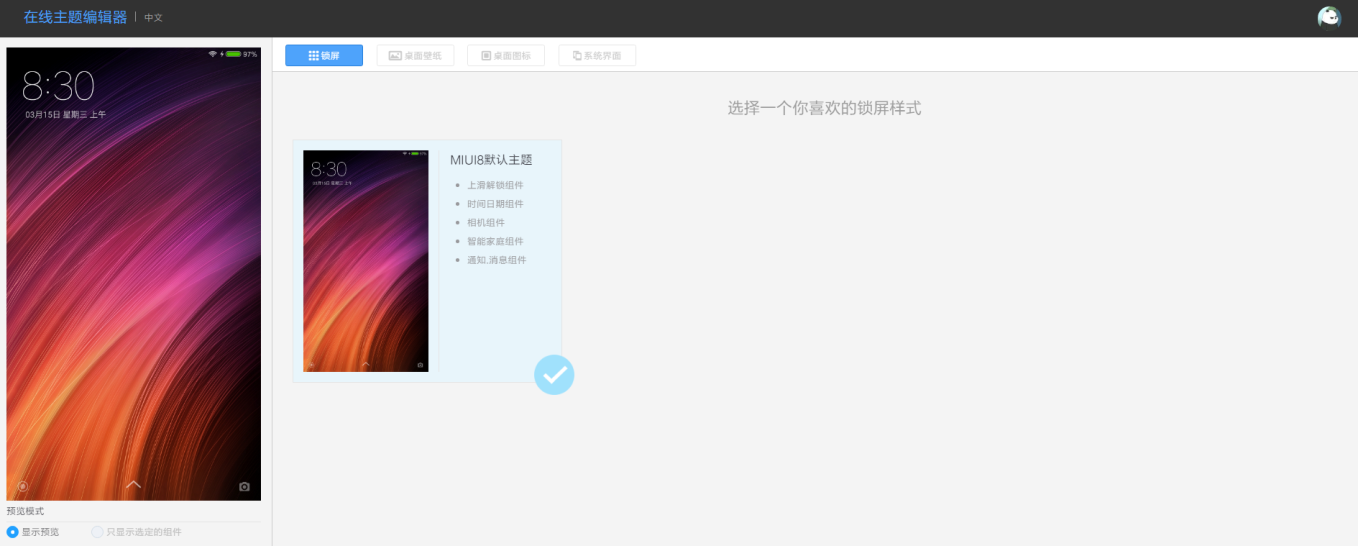


图 7‑2 模板页面图

## 资源替换

创建完主题后，便进入到资源的显示页面啦。如图 7‑3所示，模块下显示原始模板的初始资源图片，最左边显示的是资源的预览图，用户点击替换按钮，能弹出本地的文件框，供用户上传资源图片，而且我们支持用户用拖拽的方式进行上传图片！当用户成功上传资源图片后，相应的预览图更新，与用户的资源保持一样，方便用户看到最真实的效果，其效果图如图 7‑4所示，初始资源更改，预览更改。

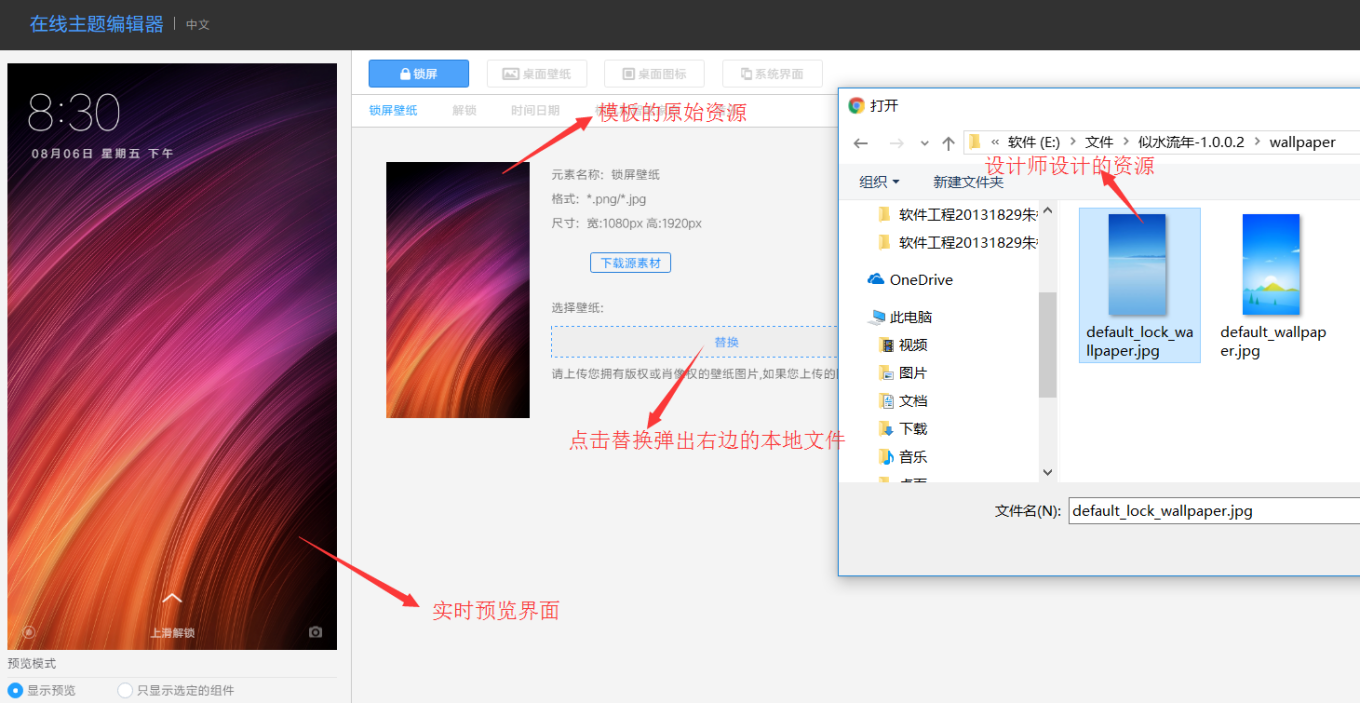


图 7‑3 资源显示页面图

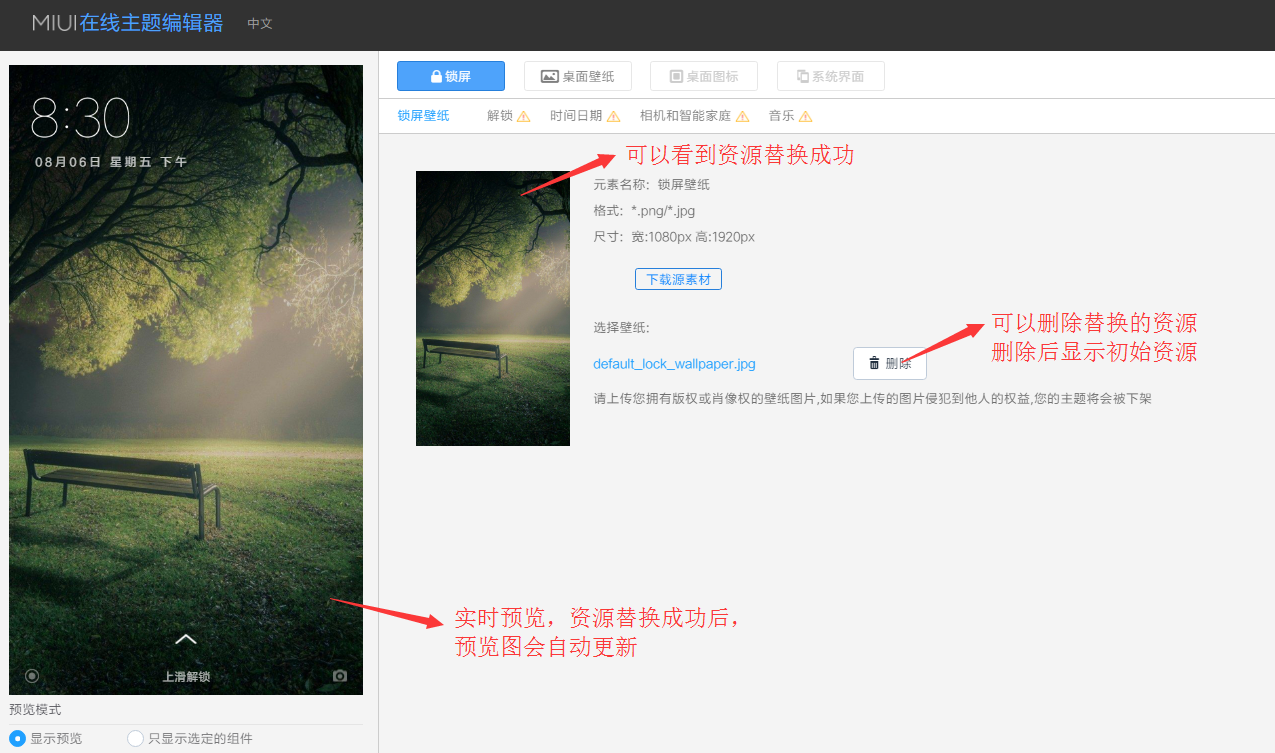


图 7‑4 替换成功展示图

从这两张图中可以看出，用户不仅很轻易的知道如何做，也很方便操作，这也是我们极力追求的目标。

## 导出

用户点击导出，会弹出一个打包的提示框，用户可以给主题包命名，打包完成生成一张二维码，用户只需要扫一扫二维码便能下载这个主题包，流程图如图 7‑5所示。



图 7‑5 打包流程展示图

## 应用到手机

接下来导入手机，应用该主题，看看效果，如图 7‑6所示。上边两张图是应用前的锁屏和解锁后的视图，下边两张图是应用后的视图。因为我们没有设计桌面图标，所以图标保持的默认的图标。



图 7‑6 应用主题效果图

从整个流程看，制作一个手机主题是如此的简单，根据平台的提示，一步一步往下做就能完成一个主题，这正是我们追求的目标。当然这也不仅仅是我们的目标，我们还会提供更有趣的制作方式，锁屏玩游戏互动，动态图标等等。

# 参考文献

1. 云计算的概念和内涵[J/OL]. http://www.chinacloud.cn/show.aspx?id=14668&cid=17.2014-02-24.
2. 十种简单方法有效保护云数据安全[J/OL].云创大数据.

http://www.cstor.cn/textdetail\_4934.html.2013-08-23.

1. Redis Documentation[J/OL]. https://redis.io/documentation.
2. 54chen.Rose使用手册[J/OL]. http://www.54chen.com/rose.html.
3. Nginx常见应用技术指南(Nginx Tips)[J/OL]. http://www.linuxtone.org/html/85/t-1685.html.

# 致谢

断断续续几个月过去了，我也终于迎来了本次毕业论文设计的尾声。因为工作的忙碌，因为第一次的缺乏经验，毕业设计的完成并没有想象中那么简单，其中的过程参和中痛苦与快乐。时常为word的排版所苦恼；时常为明知其意，却不知如何表达而烦恼；时常为项目中的bug烦恼；也时常为自己匮乏的知识而烦恼。但其中也有我们快乐的回忆，为自己解决问题，排出优雅整洁的文档格式而开心；为自己成功解决bug而喜悦；为自己在项目设计中学习到知识而开心。我深知成功完成这次毕业设计并不只是我一个人的功劳，它的背后有太多我需要感谢的人。

首先我得感谢我得母校，为我创造一个良好的学习环境，让我专心学习，陶冶情操；感谢我的老师对我的谆谆教导，不仅教给我们专业知识，还教给我们生活的道理，为我们排忧解难；感谢我的同学，我的同事，为我解决问题，不图回报；最后感谢我的指导老师刘毅，本次论文能够顺利完成，我的指导老师功不可没。是他的细心教导，教我如何一步步完成设计；是他犀利的眼神找出我的错误与不足，帮助我改进论文设计；是他不辞辛苦的为我解决问题，才会有这胜利的一天。

我知道，不管做什么事，都离不开他人的帮助。但他人的帮助并不是理所应当，我应该谨记这些给予我帮助的人，感谢他们。向他们学习，热心助人，常怀感恩之心。