# 幼儿园网站建设

# 1 绪论

## 1.1背景

随着网络正在不断深入我们身边生活，我们已经积累了关于网络的大量科学信息。而且孩子在开始被互联网所吸引，并与网络搜索引擎进行互动。当代的孩子们诞生了数字传递和互联网的世界之中，这种现象被一些作家描述为“网络世代”或“数字本土人”。有小朋友的家庭比普通家庭上网更加频繁，大约有 95% 左右的澳大利亚15岁以下儿童的家庭与互联网连接。研究表明，三岁以下的儿童具有使用电脑和互联网的技术能力。

我们虽然关注儿童的数码世界，但有限的研究探索了幼儿的互联网使用的应用场景。从指望、万方等搜索表现出，很少有研究报告宣布幼儿的网络技术和他们与网络场景的互动。互联网通过怎样的形式，来影响小朋友的日常生活。本文介绍了小朋友应用场景的其中一项内容，即记录关于小朋友在报名幼儿园时候的应用场景。

我们关注幼儿的教育问题。所以在中国家长总有这样的一种现象，即为孩子倾其所有。每一位家长总希望自己的孩子更够在基础教育阶段，获取更多的知识。虽不强求领先于其他小朋友，但也决不能输在起跑线上。可是许多家长都是第一次当爸爸妈妈，对于宝宝成长的哪几个个阶段，需要培养哪些需求是模糊不清的。小朋友在自己不同的年龄段，根据自己身心发展的规律和优点缺点所在（如记忆能力、图形能力、运算能力、空间想象能力等），制定不同的学习需求计划，以及对个人能力的培养的计划。家长、幼儿园和孩子想沟通结合，才能使孩子“德智体美劳”全面发展。

幼儿园教育最应该重视什么呢？重视的是小朋友的个体差异，为每一个幼儿提供他自己选择的，喜欢的发展方向，并在已经达到的成就上，得到进一步升华。我们通常来讲，幼儿教育内容指的是孩子在幼儿园所接受的适应性教育。幼儿园是中国基础教育之中极为重要的组成部分，其主要针对于入园的学龄前儿童（三周岁及以上），完成入学前的保育及教育任务。有的人认为，幼儿园教育的内容是空泛的、无用的。然而事实却是恰恰相反，幼儿教育虽然广泛但是绝不空泛，且每一个同时学前适应性教育的孩子，相比较于其他的孩子更具有适应能力和思考能力，故适应性教育是必要的，是重要的，更是我们必须重视的。幼儿应当完成的适应性教育应该有哪些具体的内容呢？在不同的国家和地区，可按当地的风土人情和文化根源划分出不同的范围。为绝大多数介意抽象为五个具体的方面健康、科学、社会、艺术、语言。我们所关注的，就是基础的适应性教育发展的是孩子的知识拓展，应用实践、技能技巧、理性感性等不同方面。

一所优秀的幼儿园应当具备哪些软件素质以及硬件设施呢？教育质量毫无疑问是适应性教育最为重要的一个因素。但提高幼儿园的教育质量是围绕着广大幼儿教育工作者，一直都在思考的问题。办好幼儿园，达到提高整体适应性教育质量，其核心和关键在于园长的前瞻性、正确性和执行能力。狭义的看，一个好的园长就代言了一所好的幼儿园，不仅仅是孩子的家长会啧啧称奇，连行内人也会对其敬佩有加。这就是说，园长便是决定性因素之一。在教育儿童过程中，园长处于管理层的领袖地位。他若在具有前瞻性、预见性的条件下，把握好幼儿园的整体发展方向，便可使得幼儿园蒸蒸日上。若反之则不然。

而另外的一个决定性便是环境设施。优秀的幼儿园不仅仅在硬件配备上应该满足儿童的愿望，还应该在软件上和看不见的地方（如安保、食堂卫生、监控等方面）下更大的力气。而网站建设，也属于基础建设的一部分。

网站是基础建设和环境建设的重要组成部分。针对于幼儿园来说，拥有自己的网站，一方面可以向外界传达更多的信息，便于展示自身的优势所在，退一步讲，网站也算是幼儿园硬件设施的一部分；另一方面可以让想要了解到更多信息的人增添一个获取信息的渠道，让大家对幼儿园的理解由消息的被动传达到主动的搜索信息。在互联网时代之前，人们想了解一个事物的好与坏是通过周边人的信息而得知的。其局限性在于获取信息的数量和内容较少。现在借助网络，保证信息的真实可靠情况下，我们可以用更简单的方式获取更多的信息。然而“这是一个最好的时代，也是一个最坏的时代”，互联网的内容参差不齐是我们获取信息的最大阻碍。能有效的辨识可靠信息，是我们浏览网页时需要考虑和注意的。

对于即将送孩子入园的爸爸妈妈来说，最想了解的是这个幼儿园具体的教学质量、安全卫生状况等关键信息。然而在网站道听途说不可轻信，那么来这个幼儿园的官方网站所获取到的信息无疑是最真实可靠的。如幼儿教师的资格证书展示，食品卫生安全检验标准，教室设施场景还原，器材使用，举办幼儿活动庆典等的信息，我们均可以在幼儿园网站中所获取到。这也是为什么，在如今的“互联网时代”和“数字时代”，幼儿园网站已经成为幼儿园自身的基础设施之一的原因所在。

我这次的课题名称是《幼儿园网站建设》 ，相信在过去很少有人涉足这方面的题材。相比较于大学课堂以及中小学的九年制义务教育，幼儿园的适应性教育虽然应该被人重视，但却一直没能落到实处。园长是一所幼儿园的灵魂和思想的集中体现，但是绝大多数的幼儿园园长还未将目光投入互联网。于是有一个很有趣的现象，即“懂教育的人不懂技术，懂技术的人不懂教育”。预计在未来15年左右，全国幼儿园会开始对网站渐渐有所要求，同时还会将数据库打通，将网络搜索一直到微信公众号、微信小程序等上面。如今我们立足于眼下，更是应该学会早所准备，未雨绸缪，也同时为家长和为幼儿工作者提供一个让他们施展才华的舞台。

我在搜索知网、万方数据等相关论文期刊数据库时发现，真正切合自己要求的文献资料不足5篇，这现象在“大数据”的时代显然是一件匪夷所思的事。毫无疑问，关注的和重视的人并不多。所以希望自己能够做出符合自己预期的项目，尽量用自己的想法填补一下这块空白的领域。

我的母亲在幼儿园工作，在无意中听闻关于在每年招生时，幼儿园小朋友报名的种种不便之处，如家长连夜排队等候报名，教师在报名时工作量巨大等。于是自己想尝试做出一个报名的系统，来减轻双方的压力。但仅仅一个报名系统过于薄弱，所以在其基础之上，便有了这次的项目——幼儿园网站建设。

幼儿园网站面向的是四种角色：管理层、教师、家长、游客。我们从这些角色依次入手分析。管理层拥有最高权限，可以管理所有小朋友、教师，修改园区公告，查看班级页面，开启招生系统和查看招生详细信息。教师作为班级负责人，可以发布班级公告，查看本班信息，已经对本班孩子进行修改。家长可以通过孩子的学号和密码，查看孩子所在班级主页，并且可以参加招生。游客仅仅可以浏览公共主页以及参加招生，但在填写信息时必须填写完整的验证信息（如身份证号加身份证家庭住址）。

## 1.2开发环境

### 1.2.1开发语言及技术

《幼儿园网站建设》项目可将其分为视觉层和交互层。视觉层主要表现为展示界面及其对应的加载文件。我们的视觉层首先以HTML的形式加载完成，后整合与Java Web项目中。其中使用到的技术主要有HTML5，jsp，CSS3，JavaScript（基于ES5标准，即w3school标准），Ajax异步传输，bootstrap框架，dataTable表格，jQuery等。

而后台交互层使用的是Java语言，使用框架Spring Boot + SpringMVC + MyBatis完成数据的传输及交互。Spring Boot是基于Spring的基础开发的，主要负责整合SpringMVC及MyBatis。SpringMVC主要负责与视觉层的交互以及数据传递，并对从数据库和前台传递的值进行简单的处理。MyBatis主要负责与数据库的交互，最大限度的保证数据库会话的稳定和持久，属于持久层。

为什么选择SpringBoot，而不是已经更加成熟的Spring呢？Spring Boot在继承了Spring的优点的同时，也增添了许多Spring没有的插件和框架：

1. SpringBoot是伴随着Spring4.0诞生的，一经推出，引起了巨大的反向；
2. 从字面理解，Boot是引导的意思，因此SpringBoot帮助开发者快速搭建Spring框架；
3. SpringBoot帮助开发者快速启动一个Web容器；
4. SpringBoot继承了原有Spring框架的优秀基因；
5. SpringBoot简化了使用Spring的过程；
6. Spring Boot提高了程序员开发软件时的效率，减少了代码书写量。但是其并没有令人耳目一新的新技术，运用的都是J2EE开发者常用的技术。

Spring Boot主要特性包括以下几点：

1. “习惯由于配置”：再发开软件项目的过程中，我们需要做到开发简单，多人协作和便于维护这几点。而便于开发的含义即减少配置文件的书写。Spring Boot相比较于Spring，在完成相同的功能上减少了配置文件的信息。而在绝大多数时候，我们还需要使用磨人的配置文件即可；
2. 项目快速搭建，可以无需配置的自动整合第三方的框架；
3. 可以完全不使用XML配置文件，只需要自动配置和Java Config；
4. 内嵌Servlet容器，降低了对环境的要求，可以使用命令直接执行项目，应用可用jar包执行：java -jar；
5. 提供了starter POM, 能够非常方便的进行包管理, 很大程度上减少了jar hell或者dependency hell；
6. 运行中应用状态的监控；
7. 对主流开发框架的无配置集成；
8. 与云计算的天然继承；

简而言之，Spring Boot使编码变简单，使配置变简单，使部署变简单，使监控变简单。

数据库连接我们使用的是MyBatis。MyBatis是通过XML描述符或注释将对象与存储过程或SQL语句相关联的Java持久性框架。其是在Apache许可证 2.0下分发的自由软件，是iBATIS 3.0的分支版本。其维护团队也包含iBATIS的初创成员。

MyBatis的映射关键是将Java方法与SQL语句相关联，而并非将Java对象和数据库表关联，这与Hibernate有很大的区别。由于可以让用户自由的书写SQL语句，MyBatis可以将数据库的功能完全开发，如表连接查询、数据库的专有特性、存储过程的一系列复杂的行为。所以当我们面对使用不规范数据库的项目是，选择MyBatis是非常适用的。

与JDBC相比。JDBC访问数据库是最快的，因为它是直接编译，而只要是使用框架的话一定会牺牲性能，但是带来的好处很多，复用性好、代码量少，MyBatis访问数据库，只要写好配置文件就行。框架的维护性也好，比较方便，因为都是写在xml或是yml的配置文件中。MyBatis提供的映射引擎，使得SQL语句和对象resultMap相互对应。而resultMap的使用又会关联我们的实体类，通过这样的映射使得MyBatis完成对数据库的操作。

MyBatis与Spring Framework和Google Guice集成，这使开发者免于依赖性问题。

MyBatis所使用的另外一种技术是声明式数据缓存（declarative data caching）。在使用浏览器的时候会常常听说缓存，数据缓存同理。当正在执行的SQL语句被标注后，其执行语句的结果和运行过程会被储存在cache当中。在本次运行中，若再有此条数据，可从缓存之中直接读取，而并非再次查找数据库。在提高访问速度的同时，也可以减轻数据库访问的压力。MyBatis的缓冲实现，是基于Java的HashMap基础。其被用作默认连接器，应用于OSCache、Ehcache等。简化了我们的开发。

在与视觉层页面交互时，我们选取的是SpringMVC框架。Spring MVC和Struts2都属于表现层的框架,它是Spring框架的一部分。

SpringMVC的核心架构工作流程如下表示：

1、 用户发送请求至前端控制器DispatcherServlet。DispatcherServlet在收到请求后转发给其他其他处理器。DispatcherServlet本身只用于调控全局，是的所有的请求入口统一，但不负责各个请求的具体处理；

2、 DispatcherServlet——>HandlerMapping， HandlerMapping将会把请求映射为HandlerExecutionChain对象（包含一个Handler处理器（页面控制器）对象、多个HandlerInterceptor拦截器）对象，通过这种策略模式，很容易添加新的映射策略；

3、 DispatcherServlet——>HandlerAdapter，HandlerAdapter将会把处理器包装为适配器，从而支持多种类型的处理器，即适配器设计模式的应用，从而很容易支持很多类型的处理器；

4、 HandlerAdapter——>处理器功能处理方法的调用，HandlerAdapter将会根据适配的结果调用真正的处理器的功能处理方法，完成功能处理；并返回一个ModelAndView对象（包含模型数据、逻辑视图名）；

5、 ModelAndView的逻辑视图名——> ViewResolver， ViewResolver将把逻辑视图名解析为具体的View，通过这种策略模式，很容易更换其他视图技术；

6、 View——>渲染，View会根据传进来的Model模型数据进行渲染，此处的Model实际是一个Map数据结构，因此很容易支持其他视图技术；

7、返回控制权给DispatcherServlet，由DispatcherServlet返回响应给用户，到此一个流程结束。

Spring MVC的核心组件说明如下：

1. DispatcherServlet：前端控制器。当用户的请求到达前端控制器，DispatcherServlet便成为整个控制流程的中心以及核心。用MVC模式中的介绍，属于Control层。通过DispatcherServlet调用其他组件的使用，使得各个组件之间的耦合度降低。
2. HandlerMapping：处理器映射器。HandlerMapping主要负责通过用户的请求寻找到正确的处理器（及Handler）。SpringMVC提供了多种映射方式供我们选择。如常见的有配置文件方式（..properties或.xml）、注解（@ Resource或@Autowired）等等。
3. Handler：处理器。Handler 是继DispatcherServlet前端控制器的后端控制器，在DispatcherServlet的控制下Handler对具体的用户请求进行处理。 由于Handler涉及到具体的用户业务请求，所以一般情况需要程序员根据业务需求开发Handler。
4. HandlAdapter：处理器适配器。通过HandlerAdapter对处理器进行执行，这是适配器模式的应用，通过扩展适配器可以对更多类型的处理器进行执行。
5. ViewResolver：视图解析器。负责将处理结果生成View视图，ViewResolver首先根据逻辑视图名解析成物理视图名即具体的页面地址，再生成View视图对象，最后对View进行渲染将处理结果通过页面展示给用户。 springmvc框架提供了很多的View视图类型，包括：jstlView、freemarkerView、pdfView等。

以下为各种技术所负责的内容：

SpringBoot：整合项目，依赖注入，控制反转，AOP，以及强大的集成插件；

MyBatis：Java Web项目与数据库的持久层；

SpringMVC：与前端页面交互并传递数据；

Maven：项目文件管理；

HTML5：前面的搭建和展示；

JSP：动态Java Web项目的前台页面展示，可内置Java语言；

CSS3：基于CSS基础之上，为前台页面的格式和布局提供更好的支持；

JavaScript：为前台页面的交互提供语言的支持；

Ajax：异步传输数据，即加载局部页面；

dataTable：表格插件，可用于构建各种表格；

jQuery：基于JS而开发，为前台页面的交互提供语言的支持；

bootstrap：前台框架，呈现更好的视觉效果。

### 1.2.2开发工具

本项目使用的开发工具和软件包括有：eclipse，IntelliJ IDEA，HBuilder，Adobe Photoshop CC 2015，Navicat Premium，PowerDesigner，MySql、码云、GitHub等。

Eclipse：Java软件开发环境（IDE），主程序运行环境；

IntelliJ IDEA：Java软件开发环境（IDE），主程序测试IDE；

HBuilder：HTML开发环境，建构前台页面；

Adobe Photoshop：页面所需图片修改及加工；

Navicat Premium：数据库可视化管理软件，主要用于连接数据库；

PowerDesigner：数据库设计软件，用于构思和生成数据库中的表；

MySql：存储数据；

码云及GitHub：代码托管平台。主要依赖码云，利用GitHub进行备份。

# 2 需求分析

## 2.1 可行性分析及风险预估

### 2.1.1可行性分析

市场可行性：幼儿园网站建设在全国并未开始推广。就山西太原而言，网站较为出众的仅仅是只有三家（五育之三）。而报名系统，更是闻所未闻。其具有巨大的市场潜力，值得我们去发掘。

技术可行性：

|  |  |  |  |
| --- | --- | --- | --- |
| 后台技术 | | 前端技术 | |
| 整体框架 | Spring Boot | 基础语言 | HTML、CSS、JS |
| 数据库交互 | MyBatis | 集成框架 | Bootsharp |
| 页面交互 | SpringMVC |  | JQuery |
| 数据库 | MySql |  |  |

资源可行性：

服务器平台要求：支持Intel、AMD等平台；CPU1.5GHz以上；内存2G以上；硬盘100G；

客户端平台要求：win7/8/10或Liunx等；多种浏览器支持；支持JavaScript；

收益可行性：

定量收益：此项目为学生公益项目，暂不需直接经济收益

非定量收益：幼儿园减少的人力物力资源等

法律可行性：

该产品没有侵权或抄袭等违法情况，未被申请过专利。无国家政策限制，无地方政府（或其他组织）的限制。

使用可行性：

该项目面向的角色绝大多数属30岁上下的年轻人，学习成本较低。且本系统使用方法简单易懂，无须专门培训。

### 2.1.2风险预估

## 2.2 数据词典

数据项条目

数据项名称：管理层

别名：manage

类型：String

描述：拥有最高权限的执行人员

数据项名称：教师

别名：teacher

类型：String

描述：拥有普通权限的执行人员

数据项名称：家长

别名：child

类型：String

描述：代替孩子浏览页面的角色

数据项名称：游客

别名：无注册

类型：String

描述：可浏览公共信息，可参与报名

数据项名称：角色管理

别名：role

类型：String

描述：管理员、教师、家长、游客的抽象类

数据项名称：公告

别名：Notice

类型：String

描述：向不同群体发送的不同信息展示

数据项名称：公告

别名：notice

类型：String

描述：向不同群体发送的不同信息

## 2.3功能需求分析

管理层：登录，公告发布，审批招生，人员管理（教师、学生），对账户赋予权限；

教师：仅可查询本班信息。并对本班孩子进行增删改查；

家长：浏览本班信息及公告；

游客：浏览信息，参加招生报名；

## 2.4 性能需求分析

* 响应时间小于 5 ms；
* 更新处理时间小于 5 ms；
* 数据的转换和传送时间小于 5 ms；

## 2.5 运行环境分析

服务器：

CPU：Intel Xeon E5-2682 1核以上；

内存：1G以上 DDR4 2133 MHz；

带宽：1M；

硬盘：20G SSD；

环境要求：Windows Server 2008及以上；Ubuntu（基于Liunx）；

终端：

CPU：Intel G4560 ；AMD A8-7650k及以上；

内存：1G以上 DDR3 1333 MHz；

带宽：1M；

环境要求：windows xp及以上；

浏览器：chrome；windows Edge；IE；

## 2.6 接口需求分析

相关的设备跟人们生活息息相关，因而网络系统必须要求用户界面友好，用户体验效果优良，最关键是使用户可以很容易掌握操作流程，同时要求实用、舒适和创意。

系统硬件接口：设备之间的接口。如STAT3，南桥北桥，M.2接口，PCI-E接口等

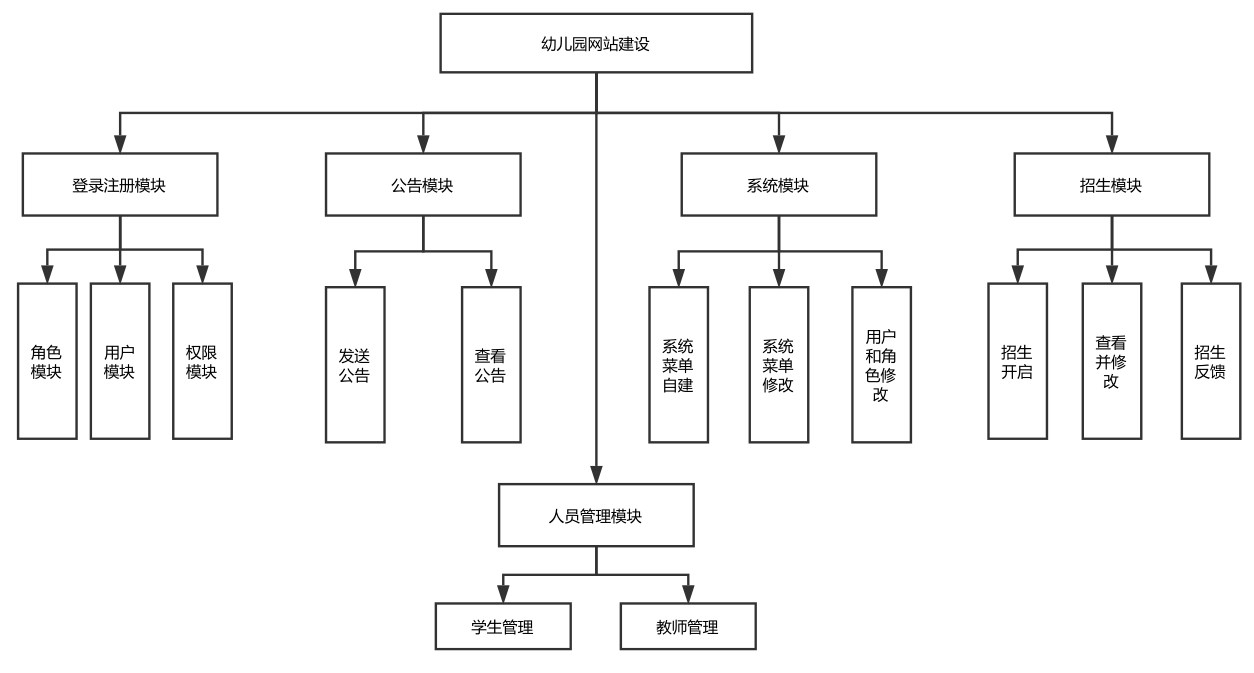
系统通信接口：本系统要求用户使用web服务器接入网路提供远程访问服务。具体就是可以接收信息，也可发送信息对设备进行控制。

## 2.7 非功能性需求

* 可用性：用户学习成本低，使用界面美观整洁，功能清晰，使人一目了然。
* 兼容性：用户可使用不同浏览器访问，均可辨识并反馈结果。
* 健壮性：系统在执行过程中处理错误，以及算法在遭遇输入、运算等异常时继续正常运行的能力要承担起考验。并对已知问题有所应对方案和解决办法。

# 3 总体设计

## 3.1 模块设计

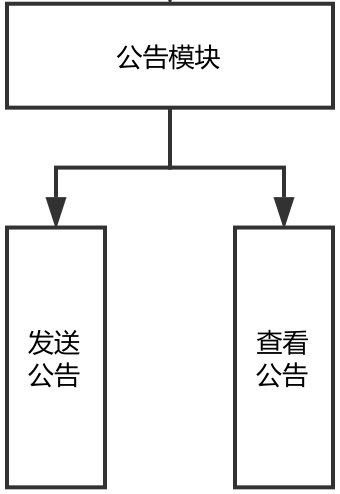


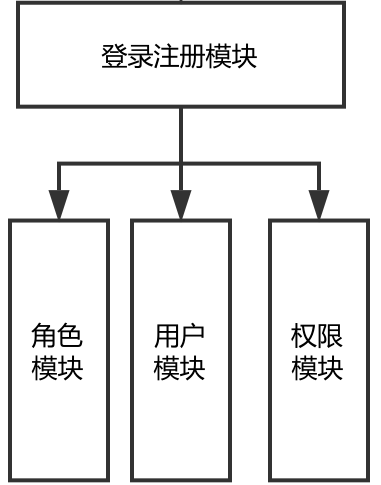
本项目共分为四个大模块，分别是：登录注册模块，公告模块，系统模块和招生模块。

登录注册模块：

登录注册模块里面具体确定了三项：用户模块，角色模块，权限模块等。

* 用户模块是用户登录时所需要的账号及密码。
* 角色模块是管理层、教师、孩子等的用户抽象的模块。角色模块拥有自己的权限，同时也是相同权限用户的集合类。
* 权限模块是不同类型角色具体所用的权限详细内容。



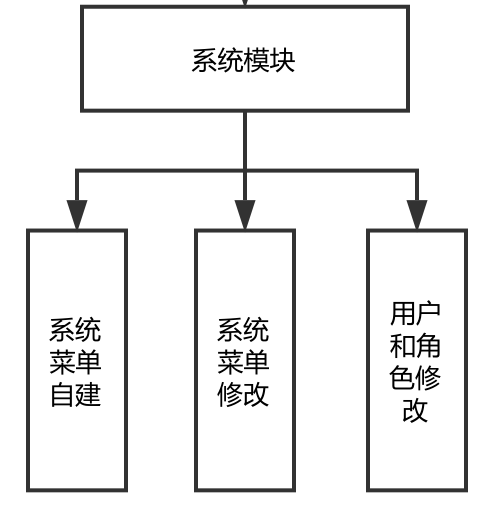


公告模块：

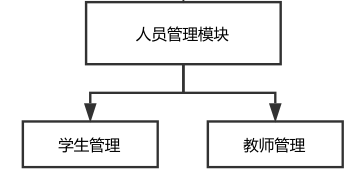
发送公告：园长有权限编写公告信息，并发布到主页可以被查看。

查看公告：在主页和公告栏上查看信息。

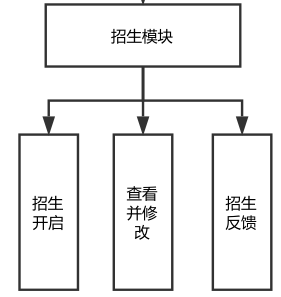
系统模块：



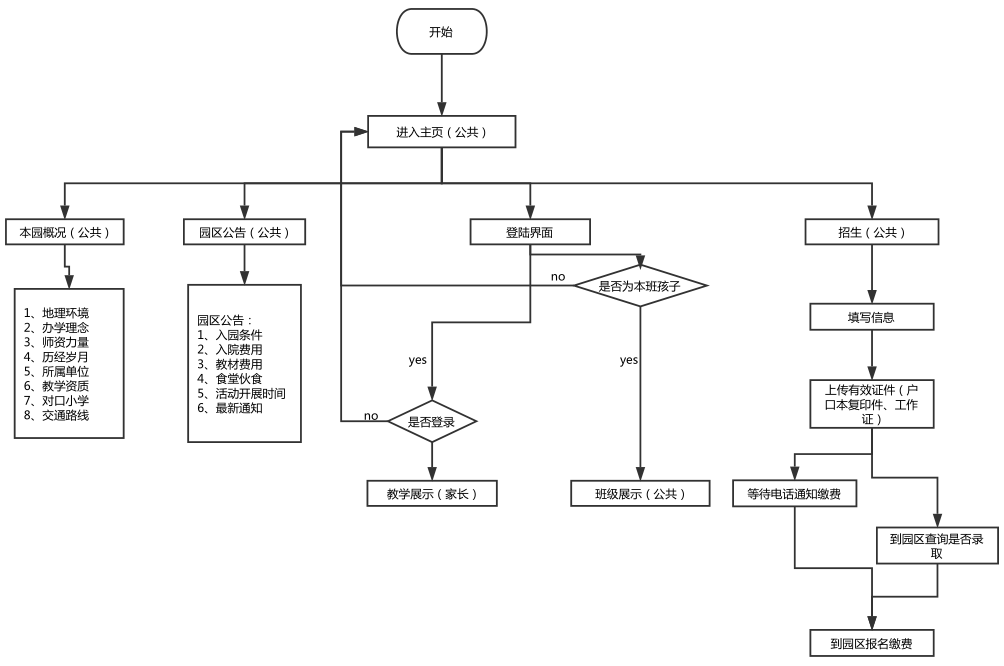
人员管理模块：



招生模块：



## 3.2 流程图设计

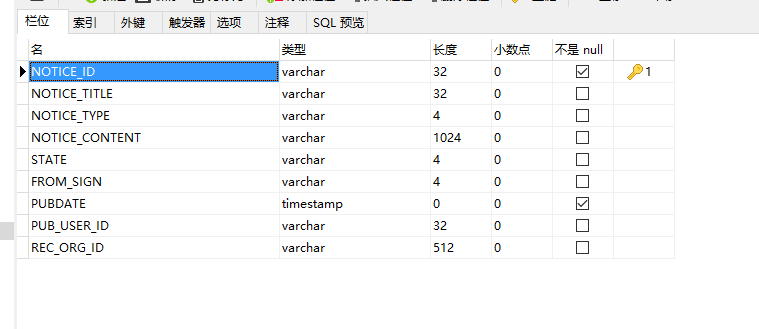


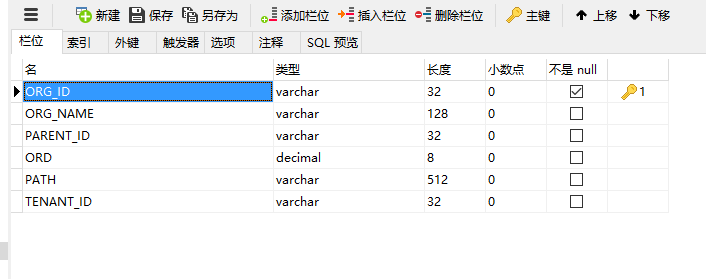
## 3.3 数据库设计

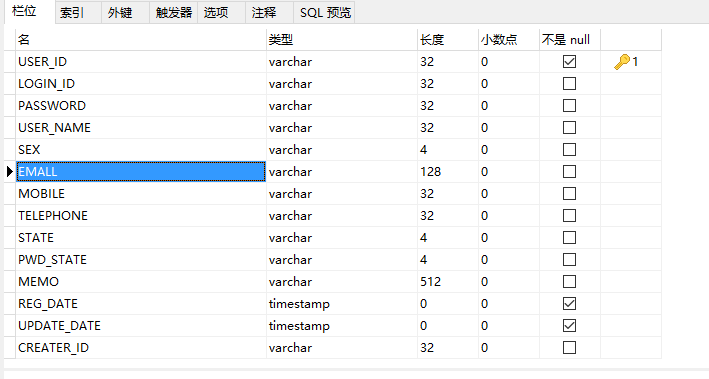


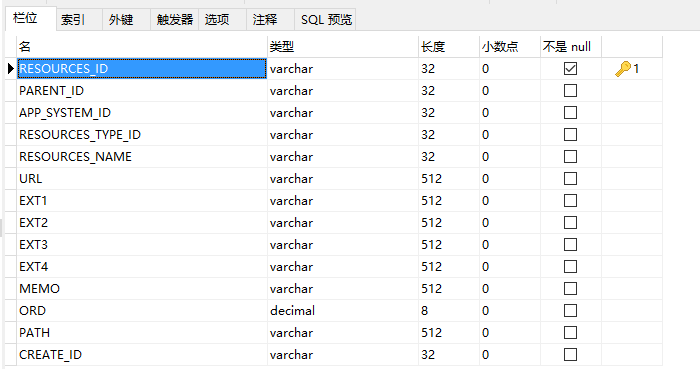
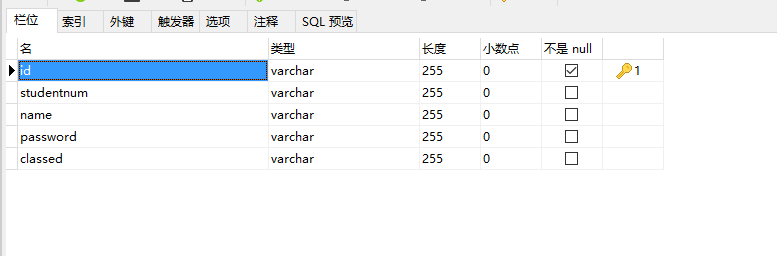
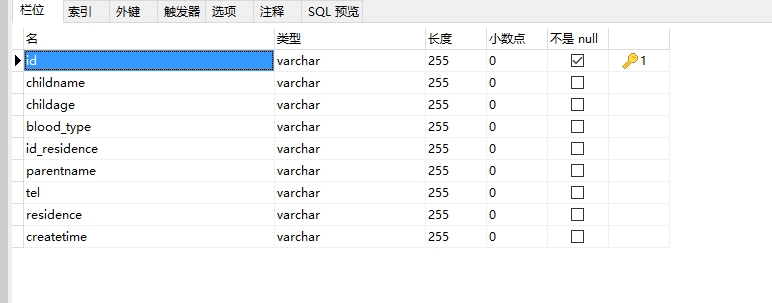
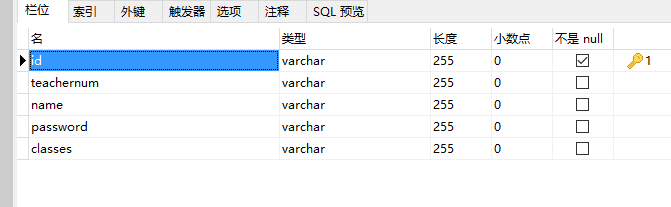
重要的表设计：

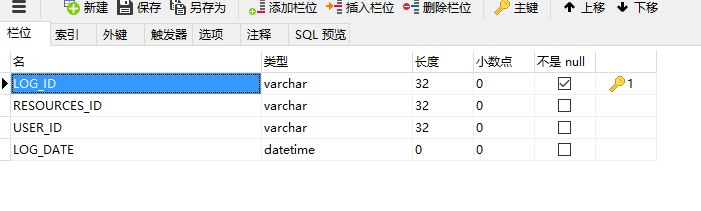
1. 公告设计表



1. 组织部门表设计
2. 角色设计表



1. 用户设计表
2. 学生设计表
3. 招生信息
4. 普通教师表
5. 用户登录记录表



# 4 详细设计

## 4.1实体类设计

根据项目的需要和数据库的衔接，共设计了11个实体类图。但是为满足设计需求和测试需求，也设计和模拟了一些虚拟类。以下罗列出了部分实体类。

### 4.1.1 管理层实体类（省略getting/setting方法以及构造方法）：

public class User implements Cloneable{

private String userId;

private String loginId;

private String password;

private String userName;

private String sex;

private String emall;

private String mobile;

private String telephone;

private String state;

private String pwdState;

private String memo;

private Date regDate;

private Date updateDate;

private String createrId;

private String orgId;

private String dataAuth;

private Date lockDate;

private long lockLoginTimes;

private long pwdValidState;

private long tenantAdmin;

private Date pwdUpdateDate;

private String tenantId;

private Date lastLoginDate;

private List<Orgnization> orgnization;

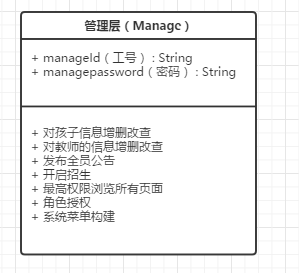
public User(){}

public User clone() {

return new User(this);

}

}



### 4.1.2 教师实体类

public class Teacher {

private String id;

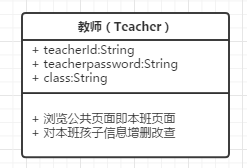
private String teachernum;

private String name;

private String password;

private String classes;

}



### 4.1.3 公告实体类

public class Notice {

private String noticeId;

private String noticeTitle;

private String noticeType;

private String noticeContent;

private String state;

private String fromSign;

private Date pubdate;

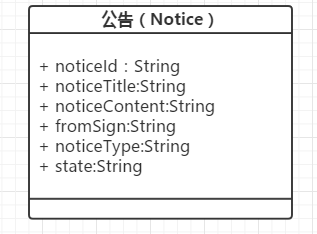
private String pubTime;

private String pubUserId;

private String recOrgId;

private String pubPerson;

}



### 4.1.4 组成类

public class Orgnization {

private String orgId;

private String orgName;

private String parentId;

private Long ord;

private String path;

private String tenantId;

private List<Orgnization> children;

private String text;

}

### 4.1.5 孩子实体类

public class Child {

private String id;

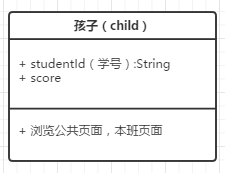
private String studentnum;

private String name;

private String password;

private String classed;

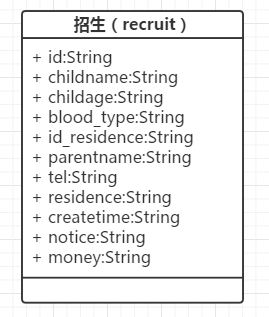
}



### 4.1.6 招生实体类

public class Recruit {

private String id;

 private String childname;

private String childage;

private String blood\_type;

private String id\_residence;

private String parentname;

private String tel;}

private String residence;

private String createtime;

}

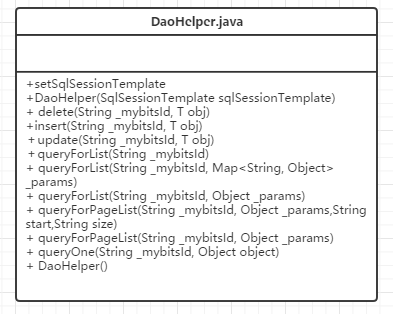
## 4.2接口设计与具体实现

### 4.2.1 DaoHelper

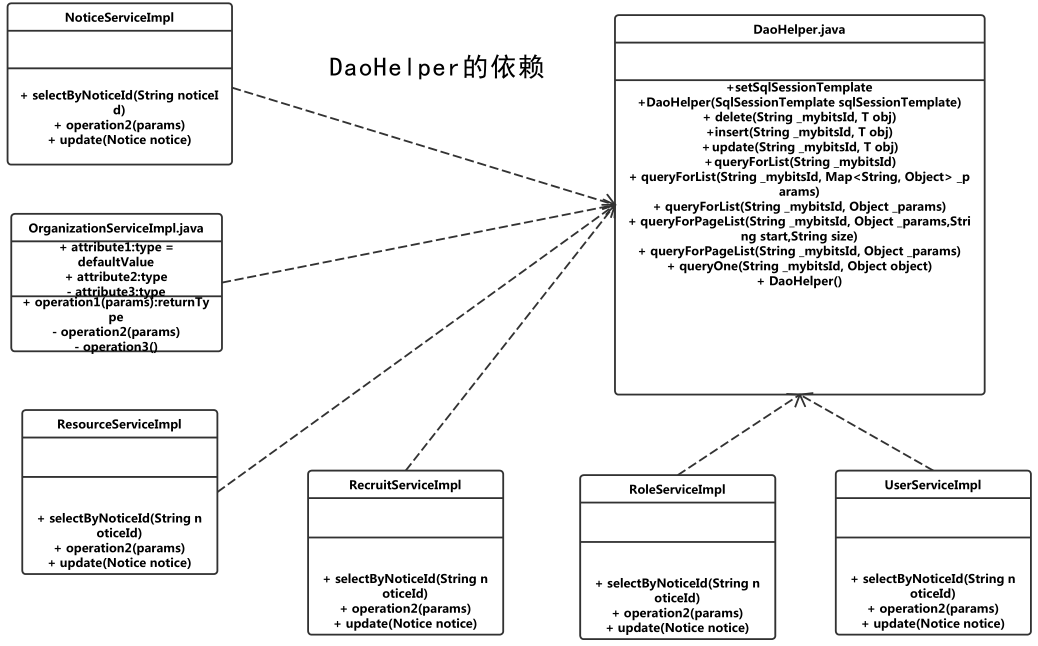
数据库的会话层展开，涉及到系统的实体类多，所以设计基础DaoHelper类，通过使用@Resource注解注入来实现基本的增删改查，对于DaoHelper尽量地封装完全，其他的实体类操作只要继承DaoHelper就可以有基本的数据操作方法，省略了大量重复地代码，提高了代码的有效率。在设计DaoHelper不仅包括了MyBatis持久化的基本增删改查，还尽量地把关联和条件查询等涉及到的方法用泛型来基本实现，使DaoHelper功能尽可能的强大和全面，可以实现一处封装，处处继承。

DaoHelper集成的方法包括了：

1. 使用SqlSessionTemplate打开数据库会话持久层；
2. 根据条件删除记录delete(String \_mybitsId, T obj){}；
3. 将指定的对象插入到数据库insert(String \_mybitsId, T obj)；
4. 更新数据库内数据update(String \_mybitsId, T obj)；
5. 无条件查询结果集，返回List<T>。方法名queryForList(String \_mybitsId)；
6. 查询结果集，查询参数为Map，返回List<T>。方法名queryForList(String \_mybitsId, Map<String, Object> \_params)；
7. 查询结果集，查询参数为Object，返回List<T>。方法名queryForList(String \_mybitsId, Object \_params)；
8. 分页查询。\_params 查询参数，start 从第几条数据开始查，size 每页显示的条数，Map<String, Object>--total：总条数，data：查询结果；
9. 查询一条记录。queryOne(String \_mybitsId, Object object)；



DaoHelper的实现



### 4.2.2 登录接口与方法的实现，见图4-2-2

图4-2-2

### 4.2.3 公告接口与实现，见图4-2-3

图4-2-3

### 4.2.4 组织接口与实现，见图4-2-4

图4-2-4

### 4.2.5 系统菜单接口与实现，见图4-2-5

图4-2-5

### 4.2.6 角色的接口与实现，见图4-2-6

图4-2-6

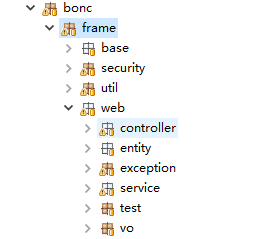
## 4.3具体功能模块设计

系统模块分为公告模块、登录模块、人员管理模块、系统页面、招生模块等。其中所有的方法均是属于重构的方法，通过引用DaoHelper实现重构最后实现。系统功能通过HTTP请求来实现，系统整体使用SpringMvc架构，通过Controller来处理请求，与页面进行交互，接受用户的输入并调用模型和视图去完成用户的需求，调用Service层，处理相关数据，决定调用对应的页面，来实现具体的功能。

负责打开数据库会话的是mybatis。其特点是可以自主定义和拼接sql语句，同时解除sql与程序代码的耦合。mybatis提供映射标签和提供对象关系映射标签，支持对象关系组建维护，便于我将sql与实体类和serviceImpl层相关联。使用xml标签，支持编写动态sql。让我们对代码和SQL语句拥有更大的控制权。

系统采用Maven项目管理工具，利用Maven可以很轻松的将我们需要的jar包加入项目。同时使用Maven使得项目的层次更加分明。当我们需要加入新的插件或者加入新的jar包是，只需在Maven项目的pom文件中导入相关的依赖就可以继承使用。

## 4.4项目模块介绍

模块设计分为五个部分，分别是common，实现主要工具类的封装；mybatis，实现数据库操作的封装；entity和model，实现实体类的设计与封装；web和controller，实现系统的请求处理，与客户端的交互，以及页面的静态生成，调用其它模块，通过处理客户端的请求，实现页面的功能。

项目实际封装分为：

### 4.4.1 Frame

Frame负责角色管理、资源授权、注册教师修改、公告模块等；见图4-4-1。

图4-4-1

### 4.4.2 Person

Person负责普通人员的管理。如不需要权限的老师及孩子。并对其进行数据库JDBC的操作。见图4-4-2。

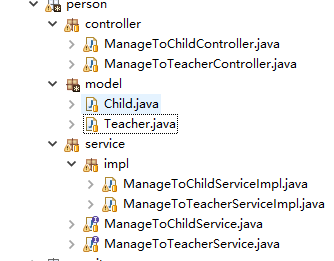


图4-4-2

### 4.4.3 Recruit

Recruit负责招生模块。管理员可以对招生信息完成反馈，并最终通过该模块查看孩子是否报名成功以及缴费情况。见图4-4-3。

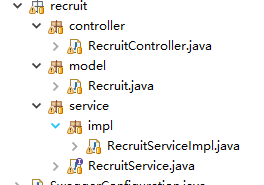
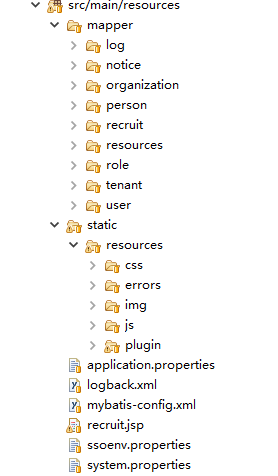


图4-4-3

### 4.4.4 资源模块

资源模块包括对数据库的操作、静态资源和配置文件的存放。Mapper、js、css、图片等均存放于此目录下。见图4-4-4。



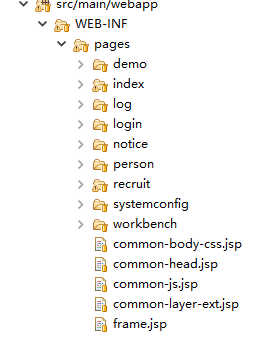


图4-4-5

图4-4-4

### 4.4.5 webapp

webapp包括对jsp页面的存放。见图4-4-5

# 5 代码实现及运行

## 5.1代码实现

### 5.1.1 MD5加密

private static String toHexString(byte[] b) {

StringBuilder sb = new StringBuilder(b.length \* 2);

for (int i = 0; i < b.length; i++) {

sb.append(HEX\_DIGITS[(b[i] & 0xf0) >>> 4]);

sb.append(HEX\_DIGITS[b[i] & 0x0f]);

}

return sb.toString();

}

public static String Bit32(String SourceString) throws NoSuchAlgorithmException {

MessageDigest digest = java.security.MessageDigest.getInstance("MD5");

digest.update(SourceString.getBytes());

byte messageDigest[] = digest.digest();

return toHexString(messageDigest);

}

### 5.1.2 id生成

private static short counter = (short) 0;

private static final int JVM = (int) (System.currentTimeMillis() >>> 8);

private final static String format(int intval) {

String formatted = Integer.toHexString(intval);

StringBuilder buf = new StringBuilder("00000000");

buf.replace(8 - formatted.length(), 8, formatted);

return buf.toString();

}

private final static String format(short shortval) {

String formatted = Integer.toHexString(shortval);

StringBuilder buf = new StringBuilder("0000");

buf.replace(4 - formatted.length(), 4, formatted);

return buf.toString();

}

private final static int getJVM() {

return JVM;

}

private final static short getCount() {

synchronized (IdUtil.class) {

if (counter < 0)

counter = 0;

return counter++;

}

}

### 5.1.3招生增删改查

@ResponseBody

@RequestMapping("/selectPage")

public Map selectPage(String start, String length, String jsonStr) {

System.out.println(start + " " + length + " " + jsonStr);

Map<String, Object> paramMap = JsonUtils.stringToCollect(jsonStr);

return recruitService.selectAll(start, length, paramMap);

}

@ResponseBody

@RequestMapping(value = "/delete", method = RequestMethod.POST)

public int delete(String id) {

return recruitService.deleteById(id);

}

@ResponseBody

@RequestMapping(value = "/insert", method = RequestMethod.POST)

public int insert(Recruit recruit) {

String temp\_str = "";

Date dt = new Date();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

temp\_str = sdf.format(dt);

recruit.setId(IdUtil.createId());

recruit.setCreatetime(temp\_str);

return recruitService.insert(recruit);

}

@ResponseBody

@RequestMapping("/getById")

public Recruit selectById(String id) {

Recruit recruit = recruitService.selectById(id);

return recruit;

}

@ResponseBody

@RequestMapping(value = "/update", method = RequestMethod.POST)

public int update(Recruit recruit) {

return recruitService.update(recruit);

}

### 5.1.4发送公告Ajax

var bindsendedBtnFun = function(){

$('.editNotice').unbind().click(function(){

var id = $(this).attr('data-idx');

window.open(webpath + '/notice/editNotice?fromSign=1&noticeId=' + id, '\_blank');

});

};

## 5.2 运行截图

### 5.2C:\Users\niu\Desktop\1494479659(1).jpg.1 项目主页截图

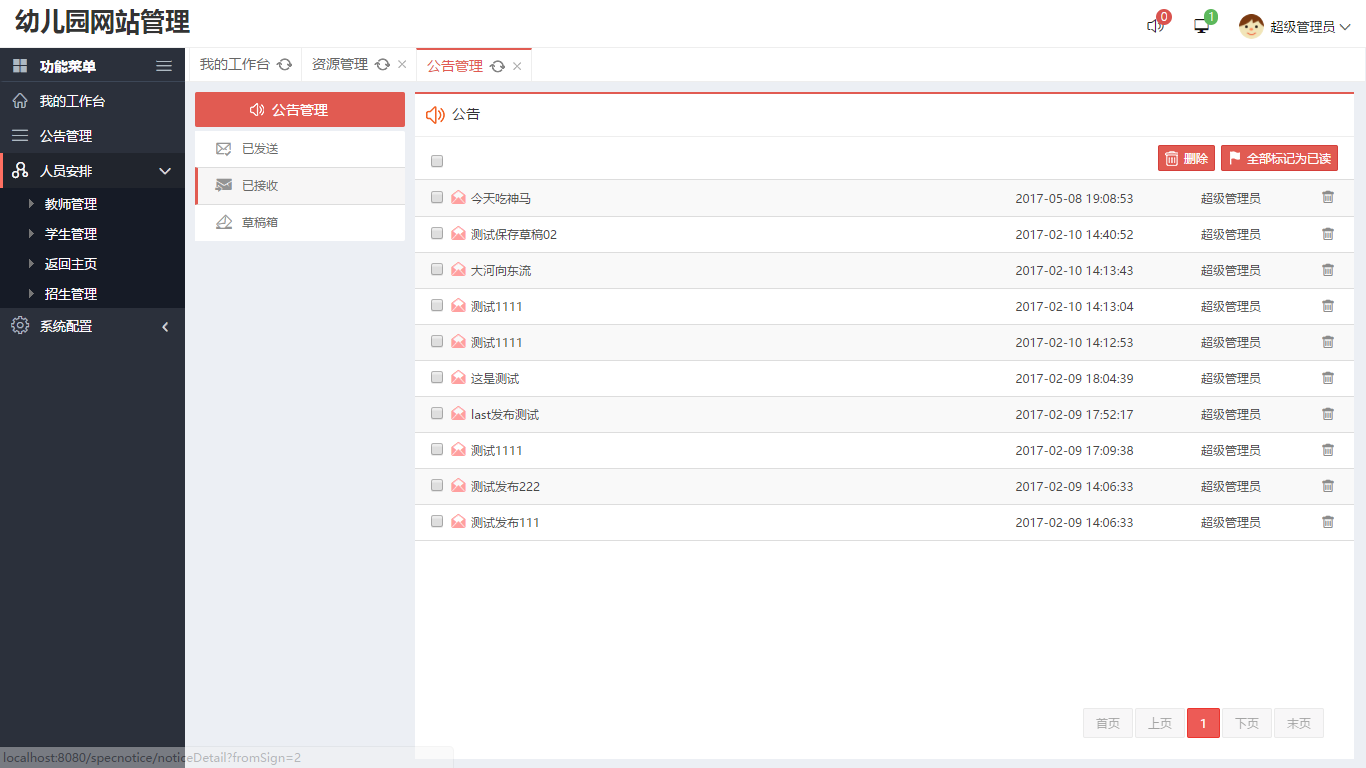


### 5.2.2 管理员主界面

### 5.2.3 登录界面

### 5.2.4 招生界面

### 5.2.5 公告发布界面



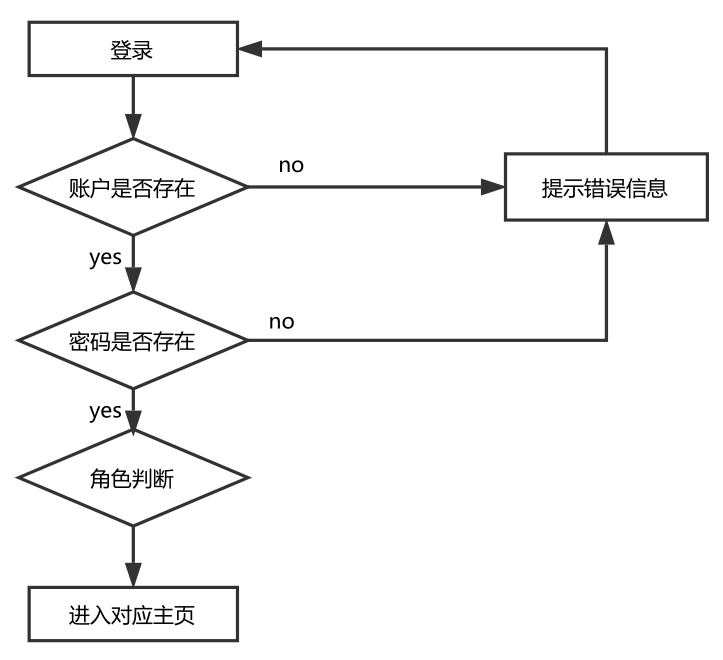
# 6 测试及维护

本系统测试以模块测试和验收测试两部分组成。分别对其使用白盒测试方法与黑盒测试方法。我们会在下面进行详细的说明介绍。

## 6.1模块测试

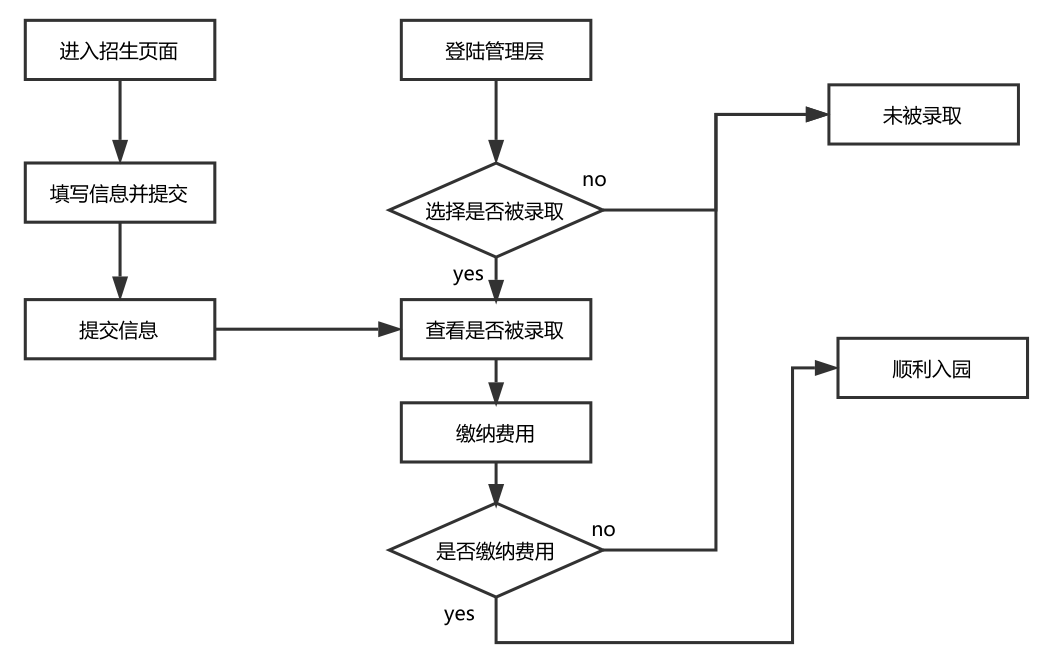
### 6.1.1黑盒—模块测试

登录模块：



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试对象 | 测试用例  （账号密码） | 预期结果 | 实际结果 |
| 1 | 后台登录 | admin / admin | 跳转显示页面 | 跳转显示页面 |
| 2 | 后台登录 | admin / null | 密码错误 | 密码错误 |
| 3 | 后台登录 | 123 / admin | 用户名错误 | 用户名错误 |
| 4 | 后台登录 | null / admin | 用户名不能为空 | 用户名不能为空 |
| 5 | 后台登录 | jingxiaowei / 123456 | 页面跳转 | 页面跳转 |

招生模块：



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试对象 | 测试用信息 | 预期结果 | 实际结果 |
| 1 | 招生测试 | 提交信息不完整（未提交电话） | 无法查看录用信息 | 无法查看录用信息 |
| 2 | 招生测试 | 提交信息不完整（提交电话） | 可查看信息 | 可查看信息 |
| 3 | 招生测试 | 未登录管理员账户 | 无招生模块 | 无招生模块 |
| 4 | 招生测试 | 登录管理员账户 | 查看并修改招生信息 | 查看并修改招生信息 |
| 5 | 招生测试 | 查找显示是否录取及缴费 | 显示正确的信息 | 显示正确的信息 |

### 6.1.2白盒—模块测试

通过白盒测试，也就是通常意义上的结构测试。我们将系统的登录模块，公告模块，招生模块均完成了分支测试。其与黑盒测试最大的不同之处在于，我们将全部的分支均覆盖到。并且专门对边界进行了测试，全部通过。

## 6.2系统测试

# 参考文献

1. 李莉，廖敏，基于ASP.NET的幼儿园网站系统，南京交通职业技术学院， 2013
2. 赵振国，我国幼儿园网站现状调查分析，陕西师范大学教育科学学院，2008
3. 严晓霞，幼儿园网站建设中的几点思考，扬州大学第一幼儿园，2012
4. 吴琼，幼儿园网站建设的几点思考，东北师范大学教育科学学院，2006
5. 黄涛，基于三层架构ASP.NET的幼儿园管理系统设计与实现，电子科技大学， 2014
6. 北京六一幼儿园网站，京ICP备16038046号，2016
7. Liu Fei，Subjective Evaluation of Built Environment for Yuhang Tiaoxi Kindergarten，[[D]](javascript:void(0);). Zhejiang University，2016
8. [Mizrap Bulunuz](http://scholar.cnki.net/result.aspx?q=%e4%bd%9c%e8%80%85%3a(Mizrap+Bulunuz))，Teaching science through play in kindergarten: does integrated play and science instruction build understanding?，[[J]](javascript:void(0);).[European Early Childhood Education Research Journal](http://scholar.cnki.net/result.aspx?q=%e6%9d%a5%e6%ba%90%3a(European+Early+Childhood+Education+Research+Journal))，2013
9. [Kevin McGowan](http://scholar.cnki.net/result.aspx?q=%e4%bd%9c%e8%80%85%3a(Kevin+McGowan))，The transition to kindergarten: One school's effort to build a new bridge，[[D]](javascript:void(0);).[State University of New York at Buffalo](http://scholar.cnki.net/result.aspx?q=%e6%9d%a5%e6%ba%90%3a(State+University+of+New+York+at+Buffalo))，2007
10. [Anonymous](http://scholar.cnki.net/result.aspx?q=%e4%bd%9c%e8%80%85%3aAnonymous)，Dorset nursery introduces unusual woody shrubs on website，[[J]](javascript:void(0);).[Horticulture Week](http://scholar.cnki.net/result.aspx?q=%e6%9d%a5%e6%ba%90%3a(Horticulture+Week))， 2008

# 致谢

至此以上是我毕业设计的所有成果展示，在大学的最后一段时光中，完成了全部的毕业设计，通过整体的毕业设计的分析和编码，对我的大学知识和专业知识有了又一次的巩固和学习，能真正的实现把大学课本上大知识运用到实践，运用到项目的开发，真正地懂得了在实践当中以专业理论知识做支撑才能更好的完成项目的开发，对我今后的工作中也有了更大的启发。

本次的毕业设计《幼儿园网站建设》。选题是我身边的事情。在学习网站设计之初，便有心愿可以帮助母亲，写出一个属于她们幼儿园网站。需求设计是自己在询问业内人士之后，不断地调整项目和思路，慢慢屡清楚了方向。数据库设计和项目架构是一次次推翻重写后的成果。不积跬步无以至千里，我们只要一步一个脚印的前行，最终都会完成我们最初的心愿。

在毕业设计的整体过程中，运用了java在web方向的基本框架，用最基本的MVC方式来进行开发，但在设计开发中以及最后论文的编写阶段还是遇到了很多问题，非常感谢两位指导老师的指导，认真的讲解了我可能会遇到的问题，应该怎样解决，现在的不足应该怎样去改正，老师们对于我们毕业设计的认真和耐心严谨，这也是我应该学习的，相信在今后的工作和生活中也一定会受益匪浅。

这次的毕业设计，有许许多多的人为我无偿的提供了帮助，让我受益匪浅，感触颇深。从选题到项目的最终，都我自己慢慢的雕琢出来的。也非常感谢我们的指导老师。在此向指导我的老师老师表示深深的感谢和崇高的敬意。

三个月毕业设计的结束，是我人生过程中一段收获颇丰的一段经历，也是我大学生活中一段难忘的记忆，大学生活也随着毕业设计的结束而接近尾声，在此感谢在大学生活中教导过我的老师和帮助过我的同学，大学时代满满的回忆和满满的收获，使我终生受益。