# 目 录

1. [**绪论 1**](#_bookmark0)

###### [研究的背景及意义 1](#_bookmark1)

###### [选题的背景 1](#_bookmark2)

###### [国内外研究现状 1](#_bookmark3)

###### [研究的意义 2](#_bookmark4)

###### [系统目标 3](#_bookmark5)

1. [**需求分析 4**](#_bookmark6)

###### [业务需求 4](#_bookmark7)

###### [功能需求 13](#_bookmark8)

###### [角色分析 13](#_bookmark9)

###### [业务功能 13](#_bookmark10)

###### [非功能需求 22](#_bookmark11)

###### [环境需求 22](#_bookmark12)

###### [性能需求 23](#_bookmark13)

###### [安全需求 23](#_bookmark14)

1. [**总体设计 24**](#_bookmark15)

###### [系统设计原则 24](#_bookmark16)

###### [体系结构设计 24](#_bookmark17)

###### [系统功能结构设计 26](#_bookmark18)

1. [**数据库设计 28**](#_bookmark19)

###### [概念结构设计 28](#_bookmark20)

###### [设计思路 28](#_bookmark21)

###### [4.1.2 E-R 图 28](#_bookmark22)

###### [逻辑结构设计 33](#_bookmark23)

###### [设计思路 33](#_bookmark24)

###### [逻辑模型 34](#_bookmark25)

###### [物理结构设计 34](#_bookmark26)

###### [存储方式 34](#_bookmark27)

###### [存储结构 35](#_bookmark28)

1. [**界面设计 39**](#_bookmark29)

[1](#_bookmark29)

* 1. [界面关系图 39](#_bookmark30)
  2. [界面设计成果 40](#_bookmark31)
     1. [主界面 40](#_bookmark32)
     2. [子界面 41](#_bookmark33)

1. [详细设计 45](#_bookmark34)
   1. [系统主要功能模块介绍 45](#_bookmark35)
   2. [报名信息管理模块 46](#_bookmark36)
      1. [功能结构设计 46](#_bookmark37)
      2. [类图设计 46](#_bookmark38)
      3. [顺序图设计 47](#_bookmark39)
      4. [核心处理流程设计 48](#_bookmark40)
   3. [统计分析模块 49](#_bookmark41)
      1. [功能结构设计 49](#_bookmark42)
      2. [类图设计 49](#_bookmark43)
      3. [顺序图设计 50](#_bookmark44)
      4. [核心处理流程设计 50](#_bookmark45)
   4. [用户管理模块 51](#_bookmark46)
      1. [功能结构设计 51](#_bookmark47)
      2. [类图设计 51](#_bookmark48)
      3. [顺序图设计 52](#_bookmark49)
      4. [核心处理流程设计 53](#_bookmark50)
   5. [课程管理模块 54](#_bookmark51)
      1. [功能结构设计 54](#_bookmark52)
      2. [类图设计 54](#_bookmark53)
      3. [顺序图设计 55](#_bookmark54)
      4. [核心处理流程设计 55](#_bookmark55)
   6. [公司信息管理模块 56](#_bookmark56)
      1. [功能结构设计 56](#_bookmark57)
      2. [类图设计 57](#_bookmark58)
      3. [顺序图设计 57](#_bookmark59)
      4. [核心处理流程设计 58](#_bookmark60)
   7. [就业管理模块 59](#_bookmark61)
      1. [功能结构设计 59](#_bookmark62)

[2](#_TOC_250001)

* + 1. [类图设计 59](#_bookmark63)
    2. [顺序图设计 60](#_bookmark64)
    3. [核心处理流程设计 61](#_bookmark65)

[7 编码 62](#_bookmark66)

* 1. [代码实现与核心算法 62](#_bookmark67)
     1. [表单模块 62](#_bookmark68)
     2. [用户模块 63](#_bookmark69)
     3. [就业管理模块 65](#_bookmark70)
     4. [公司信息模块 66](#_bookmark71)
     5. [课程模块 66](#_bookmark72)
     6. [统计分析模块 67](#_bookmark73)
  2. [代码优化分析 67](#_bookmark74)

[8 测试 68](#_bookmark75)

* 1. [测试方案设计 68](#_bookmark76)
     1. [测试策略 68](#_bookmark77)
     2. [测试进度安排 68](#_bookmark78)
     3. [测试资源 68](#_bookmark79)
     4. [关键测试点 69](#_bookmark80)
  2. [测试用例构建 69](#_bookmark81)
     1. [测试用例编写约定 69](#_bookmark82)
     2. [测试用例设计 69](#_bookmark83)
     3. [关键测试用例 70](#_bookmark84)
     4. [测试用例维护 73](#_bookmark85)

1. [总结与展望 74](#_bookmark86)
   1. [设计工作总结 74](#_bookmark87)
   2. [未来工作展望 75](#_bookmark88)

[谢辞 76](#_bookmark89)

[参考文献 77](#_bookmark90)

[附录 A 外文翻译—原文部分 79](#_bookmark91)

[附录 B 外文翻译—译文部分 86](#_bookmark92)

[附录 C 软件使用说明书 92](#_bookmark93)

[3](#_TOC_250000)

[附录 D 主要源代码 94](#_bookmark94)

4

# 1 绪论

### 研究的背景及意义

###### 选题的背景

随着企业信息化建设的不断发展，企业信息门户成为了一个新派生概念。在消费背景的前提下，人们对高质量教育的需求越来越旺盛，满足用户碎片化学习需求的在线教育用户规模持续增长。因此，在线教育市场需求大幅增长，为了紧随时代的步伐，为了适应在线教育的发展，提供更多、更好的教育资源，获得更多、更宽的生源，决定建设一套展示教育信息的门户网站。

###### 众所周知，信息化之于社会各领域最大的重要意义就在于其超快的反应速度，同时还有着极低的差错发生率，可以大大提高社会各领域的工作效率同时改进工作效果，在教育领域也不例外。现代信息技术以网络设备与技术为基础，在教育过程中加以运用，用此来改革教育，从而适应社会信息化这个新概念，其意义在于可以实施素质教育。

如今在线教育的师资力量匮乏，建立一个这样的网站有助于能力强的教师方便加入到此公司下，并且此网站能将在线教育渗透到三、四线教学设备相对薄弱的地区。为了能获得更多的生源以及给予更好的教育，本课题要开发出一个能展示教育信息的门户网站，吸引广大的想接触并学习这类行业的人员。往后，各种各样的人才将被送往全国各个地方进行信息类服务工作，更好的回报社会，同样可以拉动“新经济”的增长。

###### 国内外研究现状

公司门户网站经过多年的发展以及创新，国外许多国家的公司或者企业都经过信息化的熏陶从而建设了自己的门户网站，可以说如今这个社会的各个公司或企业的网站百花齐放，各有各的特点。IT 信息化推进了国民经济的发展，带来了巨大的经济效益。无论从哪个方面来说，当下企业或公司在Internet 经济下必须要具备信息化的技术手段。

###### 从 2001 年到 2016 年，每年新增的中国网站总数达到为前年的 27%，其中公司或企业

网站近年来增长了 23%左右，数量大概是从 20 万个左右增加到 600 万个。截至 2016 年底，

###### 据统计，中国的网页总数目前有 45 亿个左右，在线域名总数达到几百万个，这个数字的增长并不是没有原因的。信息技术行业由于受到了大规模和普遍的投入，实现了经济快速增长，以至于实现比其他行业更快更强的拉动经济作用。虽然中国企业网站数量每年都在快速增长，但面对过度的信息市场环境，大多数公司建设网站时候并没有明确的目标，未去做调研报告，从而没有做好需求分析，导致功能不全面，并且未找到自己的网站该往哪

1

###### 个方面进行服务，以至于很多网站没有达到应有的效果，除此之外，网站建设后，不进行创新和维护，以致网页内容更新慢，加载页面内容也缓慢，极大的不考虑企业在线上进行信息展示的效果。

回看过往，我国的网站建设市场经历了三个阶段：萌芽、粗放、细分。如今，国内的网站建设越来越迅速和规范。虽然建站水平成熟，但门户网站没有知名度，不能像 2345 网站、淘宝、京东一样做好宣传和推广工作，从而使网站的访问量极低。网站建设之后不能忽略对网站的推广，因为推广也是对于企业竞争的一个重要因素。

###### 当前，建设一个企业对外的信息窗口是十分重要并且非常急迫的，对于此类网站来说， 还有更多的创新和发展空间，比如负载能力、安全特性、性能等等。

随着网络的发展，教育信息化拥有了更为便捷的工具。如今更多的学生和老师通过网络上的各种资源来进行学习，这样的学习有很强的自主性。当前，某些知识的专题站在我国网站蓬勃发展条件下，成为了网络信息资源的重要组成部分。

###### 自主学习能力在教育信息化的过程中受到老师和家长越来越多的重视，他们会利用各种电子产品用以学习，提升自己的生活水平。当前，我国这类网站正在积极地寻求适应自己的发展模式来顺应这一趋势。

###### 研究的意义

###### 信息类大型门户网站对现在来说是一个非常好的势头，“新经济”的基础在于教育信息化，是促进社会信息全面发展的必然要求。

各种单位都深知内容管理的重要性。从信息化的角度来说，信息管理软件存在广大市场的相关因素主要有：

###### 信息展示是公司的“大门”。在Internet 交互过程中，绝大部分公司或企业都和信息交互有关，社会人员通过互联网来获取知识的时间越来越长。在教育信息的个性环境下， 游客可以直接通过在网站上获取公司或企业信息从而直接在网站上与其进行互动。

1. 公司信息化需求增长。随着各种单位信息化进行线上互动，内、外联网之间存在越来越多的信息交互，优秀的门户网站对公司内部来说，信息的收集、重复利用、增值利用等等通过这个才能做的更好。信息的真正交互式和协调性的内容才是公司外部最好的门面，只有外部的东西做好了，才能获取网站的浏览量，从而提升公司的知名度。

###### 信息的更新程度。无论在企业还是公司，信息的更新只会越来越快，各个单位信息的产生量也必然越来越大，趋势是成倍的增长，公司或企业招收人员更需要的是一个功能全面、安全性高、反馈迅速的信息管理技术用以信息的维护与更新，更重要的是保证信息的安全性。

2

### 系统目标

###### “”IT 教育信息网站前台的功能主要是实现网站信息展示，同时提供网上报名、在线浏览课程内容、在线咨询等功能。游客在浏览信息之后，可选择继续浏览其他信息或者获取更多课程信息，获取更多信息需进行网上报名或者在线咨询，网上报名需要填写基本信息与手机号，在线咨询需要咨询师进行回复。

网站后台的功能主要是用来维持网站运行重要的平台，主要是提供网站所需的各种信息，创建不同用户，对这些信息或用户进行新增、修改、删除、查找及管理。如管理用户， 需要新增用户的基本信息，并添加账号密码，赋予权限，测试用户登陆后台内容的显示是否正确，才能获得一个新的用户。

###### 本系统利用模块化开发方法，通过 UML 建模技术、Spring Boot 框架、MySQL 数据库、HTML5、CSS、Vue、Ajax 等技术实现一个教育信息门户网站。此系统包括前台模块与后台管理模块。前后台数据交互，界面美观，用户体验良好。

后台管理系统主要包括以下六个模块：

###### 用户管理：超级管理员能够新增用户，查找用户信息，查看用户的状态，后台用户拥有Token 时效，当Token 过期时，自动下线，以保证系统安全。

1. 表单管理：后台用户能够实现新增表单信息，查找表单信息，对表单信息进行修改或删除处理，自动生成处理表单的时间。

###### 表单分析：后台用户可以查看系统对表单某些字段进行分析处理后得到的结果。

1. 咨询信息管理：后台用户能够实现対咨询信息的检索，修改或者删除自动生成的咨询信息，咨询信息生成的时间为系统自动生成。

###### 就业管理：后台用户能够查看前台展示的就业信息，并可以对此信息进行操作。

1. 教资信息管理：后台用户能够查看前台展示的教资信息，并可以对此信息进行操作，点击详情可以查看对教师的详细介绍。

###### 公司信息管理：后台用户能查看与公司合作单位的信息以及对公司信息进行操作。

1. 课程管理：后台用户能对课程类别信息进行操作，并且可以查看课程信息，同样对课程信息进行操作，同时对课程修改后的信息会同步到前台进行展示。

3

# 需求分析

### 业务需求

###### 能够实现课程信息展示、教资信息展示、就业信息展示、在线咨询、网上报名等功能，对信息展示进行全面管理。

1. 实现对在线报名和在线咨询过程中产生的所有业务数据进行处理，如报名表单、报名记录、咨询信息的管理与统计。

###### 网上报名设置了不能重复报名规则，并且设置了姓名和手机号需符合格式且真实。

1. 在线咨询功能引入外部接口，链接到另一个页面上。

###### 系统后台功能齐全，能够对前台页面上的信息进行有效的管理，同时还可以对后台登陆用户信息进行管理，后台业务包括用户管理、表单管理等等。

1. 能够进行对表单系统处理的统计分析和咨询师处理表单的统计分析。

###### 创建用户时，需要对其分配权限以及角色，还需要手动输入其账号和密码。

1. 设置了页面的响应时间，采用了Token 验证方式进入后台，性能较好。

###### 后台系统有检索和分析功能，有利于信息的筛选。

主要业务流程：

###### 统计分析业务：登陆成功后，点击统计分析，查看统计分析数据。统计分析业务活动，如图 2-5 所示：

登陆成功

点击统计分析

查看统计分析

数据

图 2-5 统计分析活动图

4

###### 网上报名业务：选择网报浮窗，填写网报基本信息，专业，电话，填写完成后点击报名。若电话号码填写不正确，报名失败；若填写正确，报名成功，生成报名信息添加至表单系统，咨询师获得信息后，电话联系游客。网上报名业务活动，如图 2-1 所示：

点击网上报名

浮窗

信息填写

否

信息是否

符合要求

是

报名成功

图 2-1 网上报名活动图

5

###### 后台登陆业务：打开登陆页面，选择登陆角色，检测账号密码，若账号密码正确， 则根据登陆角色获得后台权限，登陆成功，不同账号显示的可用模块不同。若不正确，则返回登陆页面重新登陆。后台登陆业务活动，如图 2-2 所示：

访问登陆页面

输入账号密码

否

密码是否正确

是

判断账号权限

超级管理员登

录

普通管理员

咨询师

信息录入员

登录系统成功

图 2-2 后台系统登陆活动图

6

###### 查询业务：登陆成功后，在搜索栏填入搜索信息，查找成功，返回数据，查找失败，返回未查到所需信息。查询业务活动，如图 2-3 所示：

登陆成功

点击搜索框

输入信息

是否存在

所查找信息

是

否

数据不存在， 返回控制台

输出查询的信

息

图 2-3 查询业务活动图

7

###### 创建用户业务：进入用户管理，添加用户基本信息，输入账号密码，分配权限， 更新用户列表。创建用户业务活动，如图 2-4 所示：

是

否

登陆成功

是否显示正确的

用户权限内容

创建用户

更新用户列表

使用新用户登

录

是否验证新用

户合法性

是

退出系统

否

创建成功

图 2-4 创建用户操作活动图

8

###### 报名信息管理业务：登陆成功后，点击进入报名信息管理，查看信息并检索，若要删除信息，则点击删除按钮，若报名信息状态为已处理，则删除，若状态为其他类型， 则删除失败，提示该信息不允许删除，刷新报名信息列表。报名信息管理业务活动，如图2-6 所示：

登陆成功

点击进入报名 删除信息

信息管理

检索信息

信息状态是

否为已处理

刷新列表信息

是

否

不允许删除信息

图 2-6 报名信息管理活动图

9

###### 就业管理业务：登陆成功后，选择就业管理业务，查看就业信息，选择相应的信息，点击操作栏里的修改或删除按钮，随后根据系统提示完成操作。若操作成功，更新信息；若操作失败，返回控制台。就业管理业务活动，如图 2-7 所示：

登陆成功

选择就

业管理

否

对就业信息进

行操作

信息是否

符合要求

是

录入信息成功

刷新列表信息

图 2-7 就业管理业务活动图

10

###### 公司信息管理业务：登陆成功后，点击选择公司基本信息或公司概况信息，点击选择所要做出的操作，根据系统提示完成操作。若填写的信息符合规定，则操作成功，更新课程信息列表。若操作失败，则重新填写。公司信息管理业务活动，如图 2-8 所示：

登陆成功

选择公司

信息管理

返回控制台

对公司信息进行操作

信息是否 否 符合要求

是

录入信息成功

刷新列表信息

图 2-8 公司信息管理业务活动图

11

###### 课程信息管理业务：登陆成功后，点击选择课程管理或课程类别，点击选择所要做出的操作，包括信息的添加，修改或删除，若填写的信息符合规定，则操作成功，更新课程信息列表。若操作失败，则重新填写。课程信息管理业务活动，如图 2-9 所示：

登陆成功

选择课程

信息管理

对课程信息

进行操作

返回控制台

否

信息是否

符合要求

是

录入信息成功

刷新列表信息

图 2-9 课程信息操作活动图

12

### 功能需求

###### 角色分析

系统角色分析，如表 2-1 所示：

表 2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 超级管理员 | 管理用户系统，管理课程信息系统，管理公司信息系统，管理表单系  统，管理菜单栏系统，进行就业管理和咨询内容管理 |
| 普通管理员 | 管理菜单栏系统，管理表单系统，参与就业管理 |
| 咨询师 | 参与咨询内容管理，参与表单系统 |
| 信息录入员 | 管理课程信息系统，管理公司信息系统 |
| 游客 | 参与课程管理，查看公司基本信息、公司概况信息，参与咨询内容管  理，参与报名信息管理 |

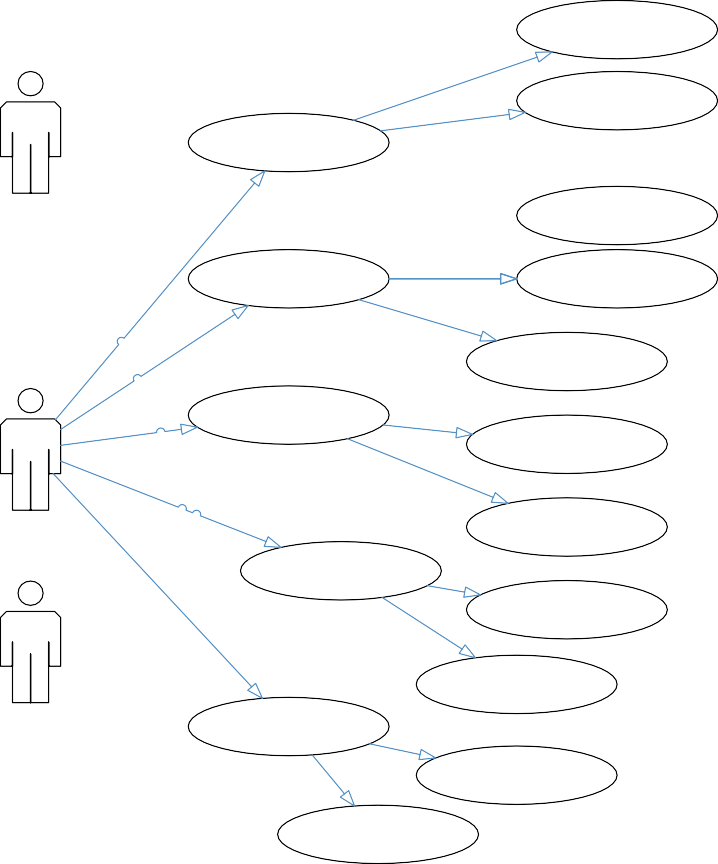
###### 业务功能

网站前台的功能主要是实现网站信息展示，同时提供网上报名、浏览课程信息、在线咨询等功能。游客在浏览信息之后，可选择继续浏览其他信息或者获取更多课程信息，获取更多课程信息需进行网上报名或者在线咨询，网上报名需要填写基本信息与手机号，在线咨询需要咨询师进行回复。

###### 网站后台的功能主要是用来维持网站运行重要的平台，主要是提供网站所需的各种信息，创建不同用户，对这些信息或用户进行新增、修改、删除、查找及管理。如管理用户， 需要新增用户的基本信息，并添加账号密码，赋予权限，测试用户登陆后台内容的显示是否正确，才能获得一个新的用户。

13

###### 系统中的总用例图，如图 2-10 所示：



就业管理

菜单栏目

栏目状态

菜单栏系统

普通管理员

用户管理

用户系统

角色管理

权限管理

课程信息系统

课程类别

超级管理员

课程管理

公司信息系统

公司基本信息

公司概况信息

信息录入员

表单系统

报名信息管理

游客

统计分析

咨询师

咨询内容管理

图 2-10 系统总体用例图

###### 用户管理子系统总用例，如图 2-11 所示：



登陆

新增用户

<<包含>>

修改用户信息

<<包含>>

用户管理

用户系统

<<包含>>

<<包含>> 删除用户信息

<<包含>>

查询用户

角色管理

超级管理员

<<包含>>

分配角色

权限管理

<<包含>>

权限分配

图 2-11 用户子系统用例图

14

###### 登陆，如表 2-2 所示：

表 2-2 登陆用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 登陆 |
| 用例描述 | 登陆是系统进行身份验证的功能模块，用户在提供有效的账号和密码后，发送系统后台进行身份验证，返回验证后的信息，根据返回信息  判断身份验证是否通过。 | | |
| 执行者 | 超级管理员、普通管理员、咨询师、信息录入员 | | |
| 前置条件 | 登陆后台前，拥有登陆后台的账号和密码 | | |
| 后置条件 | 登陆成功后显示正确的菜单栏 | | |
| 涉众利益 | 1.用户退出系统后能够记住账号 | | |
| 用例场景 | 系统登陆界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 填写账号、密码及验证码 2. 系统验证账号、密码及验证码是否填写完整 3. 如果验证不完整，提示用户按要求填写必要的信息 4. 将有效的用户账号、密码及验证码提交到服务器验证 5. 系统根据服务器验证结果进行相应操作 6. 如果验证是验证码错误，直接提示重新输入验证码 7. 如果验证通过，直接登陆到系统主界面。扩展流程（Extend Flow）   1a.登陆信息不充分：  1a1.系统提示登陆信息不充分  1a2.用例结束  2a.验证码错误：  2a1.提示重新输入  2a2.用例结束  字段列表（Filed List）  1. 用户登陆信息＝账号＋密码＋验证码业务规则（Business role）   1. 账号不能为空，超级管理员规定账号； 2. 密码不能为空，超级管理员规定密码； 3. 密码必须是加密； 4. 系统先验证验证码，后验证账号及密码。 | | |

15

续表 2-2

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 登陆 |
| 特殊需求  （Special Requirement） | 验证码刷新时间为 1min。 | | |
| 编写人 |  | | |

###### 分配角色，如表 2-3 所示：

表 2-3 分配角色用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 分配角色 |
| 用例描述 | 分配角色是在角色管理上给用户分配角色名称，让用户登陆后拥有不同  的权限。 | | |
| 执行者 | 超级管理员 | | |
| 前置条件 | 登陆后台前，拥有登陆后台的账号和密码 | | |
| 涉众利益 | 1. 管理者担心后台菜单栏显示与权限不符 2. 用户退出系统后能够记住账号 | | |
| 用例场景 | 控制台搜索框界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 点击角色管理 2. 点击修改角色 3. 选择角色名称 4. 确认修改并自动返回角色管理列表字段列表（Filed List）   1.角色信息=用户名+角色名+状态+角色拥有权限 | | |
| 编写人 |  | | |

###### 分配权限，如表 2-4 所示：

表 2-4 分配权限用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0003 | 用例名称 | 分配权限 |
| 用例描述 | 分配权限是在权限管理上给不同角色赋予不同权限。 | | |
| 执行者 | 超级管理员 | | |
| 前置条件 | 登陆后台前，拥有登陆后台的账号和密码 | | |
| 涉众利益 | 1. 管理者担心后台菜单栏显示与权限不符 | | |
| 用例场景 | 控制台搜索框界面 | | |

16

续表 2-4

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0003 | 用例名称 | 分配权限 |
| 基本事件流 | 基本流程（Base Flow）   1. 点击权限管理 2. 选择角色可以拥有的权限 3. 完成选择并返回权限管理列表字段列表（Filed List）   1.权限信息=权限名+角色名+状态 | | |
| 编写人 |  | | |

###### 表单管理子系统总用例，如图 2-12 所示：



查询报名信息

**<<**包含**>>**

普通管理员

添加报名信息

**<<**包含**>>**

报名信息管理 **<<**包含**>>** 修改报名信息

游客

**<<**包含**>>**

删除报名信息

表单系统

超级管理员

统计分析

**<<**包含**>>**

分析数据展示

咨询师

图 2-12 表单管理子系统用例图

###### 查询报名信息，如表 2-5 所示：

表 2-5 查询报名信息用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 查询报名信息 |
| 用例描述 | 查询报名信息是通过选择关键字来筛选所需要展示在后台上的报名信  息。 | | |
| 执行者 | 超级管理员、普通管理员、咨询师 | | |
| 前置条件 | 后台登陆成功 | | |
| 后置条件 | 登陆成功的权限有效 | | |
| 用例场景 | 系统表单管理界面 | | |

17

续表 2-5

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 查询报名信息 |
| 基本事件流 | 基本流程（Base Flow）   1. 登陆成功后，菜单栏会显示表单管理 2. 点击进入表单管理，可以查看报名信息 3. 选择关键字进行筛选 列表显示相关的报名信息   字段列表（Filed List）  1.报名信息＝姓名+电话+专业+创建时间+处理时间+处理人+处理结果+  状态+备注 | | |
| 特殊需求  （Special Requirement） | 1. 进行查询操作后，自动刷新列表时间为 3s 2. 刷新列表反应时间不超过 5s | | |
| 编写人 |  | | |

###### 添加报名信息，如表 2-6 所示：

表 2-6 添加报名信息用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 添加报名信息 |
| 用例描述 | 添加报名信息是游客在前台页面上进行网上报名或者后台用户进行手  录信息从而在列表上添加一条报名信息。 | | |
| 执行者 | 超级管理员、普通管理员、咨询师、游客 | | |
| 前置条件 | 登陆后台前，拥有后台用户的账号和密码 | | |
| 后置条件 | 登陆成功后存在此管理的权限 | | |
| 涉众利益 | 1.添加报名信息符合要求 | | |
| 用例场景 | 系统表单管理界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 打开相应界面 2. 输入相关信息，完成添加字段列表（Filed List）   1.报名信息＝姓名+电话+专业+创建时间+处理时间+处理人+处理结果+  状态+备注  业务规则（Business role）   1. 姓名只能是中文字符 2. 电话必须符合规则且不能重复 | | |
| 编写人 |  | | |

18

###### 就业管理总用例，如图 2-13 所示：



添加就业信息

**<<**包含**>>**

修改就业信息

普通管理员

**<<**包含**>>**

就业管理

**<<**包含**>>**

删除就业信息

游客

**<<**扩展**>>**

超级管理员

就业信息展示

图 2-13 就业管理总用例图

###### 删除就业信息，如表 2-7 所示：

表 2-7 删除就业信息用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 删除就业信息 |
| 用例描述 | 删除就业信息是根据实际要求删除就业信息，直接在就业管理界面点  击删除按钮。 | | |
| 执行者 | 超级管理员、普通管理员 | | |
| 前置条件 | 登陆后台前，拥有后台用户的账号和密码 | | |
| 后置条件 | 登陆成功后拥有此界面的权限 | | |
| 用例场景 | 系统就业管理界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 打开就业管理界面 2. 选择相应的报名信息 3. 点击删除按钮 4. 点击确认删除   字段列表（Filed List）  1.就业信息＝姓名+性别+公司+薪资+专业+毕业年份 | | |
| 编写人 |  | | |

###### 就业信息展示，如表 2-8 所示：

表 2-8 就业信息展示用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 就业信息展示 |
| 用例描述 | 就业信息展示是根据信息的状态将就业信息有选择的展示到前台页  面上。 | | |
| 执行者 | 所有用户 | | |

19

续表 2-8

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 就业信息展示 |
| 用例场景 | 系统前台界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 打开就业界面 2. 查看相应的就业信息   字段列表（Filed List）  1.就业信息＝姓名+性别+公司+薪资+专业+毕业年份 | | |
| 编写人 |  | | |

###### 课程信息子系统总用例，如图 2-14 所示：



删除课程类别

**<<**包含**>>**

修改课程类别

课程类别

**<<**包含**>>**

**<<**包含**>>**

课程信息系统

添加课程类别

超级管理员

课程管理

**<<**包含**>>**

**<<**包含**>>**

课程信息展示

游客

信息录入员

**<<**包含**>>** 删除课程信息

**<<**包含**>>**

修改课程信息

添加课程信息

图 2-14 课程信息子系统总用例图

###### 1) 添加课程信息，如表 2-9 所示：

表 2-9 添加课程信息用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 添加课程信息 |
| 用例描述 | 添加课程信息是在后台添加相关的课程信息并在前台进行展示。 | | |
| 执行者 | 超级管理员、信息录入员 | | |
| 前置条件 | 登陆后台前，拥有后台用户的账号和密码 | | |
| 后置条件 | 登陆成功后存在此管理的权限 | | |
| 涉众利益 | 1.添加课程信息符合要求 | | |
| 用例场景 | 系统课程管理界面 | | |

20

续表 2-9

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 添加课程信息 |
| 基本事件流 | 基本流程（Base Flow）   1. 打开课程管理界面 2. 点击添加按钮 3. 进入管理界面 4. 输入相应信息 5. 完成添加自动返回列表 字段列表（Filed List） 6. 课程信息＝课程名称+相关专业+时长+发展方向+是否免费+介绍+所属课程类别+图片路径 7. 课程类别=类别名称+状态+图片路径 | | |
| 编写人 |  | | |

###### 公司信息子系统总用例，如图 2-15 所示：



合作单位信息展示

**<<**包含**>>**

合作单位信息 **<<**包含**>>** 添加合作单位信息

**<<**包含**>>**

**<<**包含**>>**

超级管理员

修改合作单位信息

删除合作单位

公司信息系统

游客

公司概况信息

**<<**包含**>>** 公司概况信息展示

**<<**包含**>>**

信息录入员

**<<**包含**>>**

**<<**包含**>>**

修改公司概况信息

添加公司概况信息

删除公司概况信息

图 2-15 公司信息子系统总用例图

###### 1) 修改公司信息，如表 2-10 所示：

表 2-10 修改公司信息用例描述表

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 修改公司信息 |
| 用例描述 | 修改公司信息是修改相关公司的信息，从而在前台进行更新展示。 | | |
| 执行者 | 超级管理员、信息录入员 | | |
| 前置条件 | 登陆后台前，拥有后台用户的账号和密码 | | |
| 后置条件 | 登陆成功后存在此管理的权限 | | |

21

续表 2-10

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 修改公司信息 |
| 用例场景 | 系统合作单位信息界面或公司概况信息界面 | | |
| 基本事件流 | 基本流程（Base Flow）   1. 打开公司信息界面，点击修改按钮 2. 进入修改界面后根据相关提示修改 3. 完成修改   字段列表（Filed List）   1. 合作单位＝合作单位名称+状态+图片路径+所属类型 2. 公司概况信息=名称+地址+联系电话+邮箱+在岗人数+成立年份 | | |
| 编写人 |  | | |

### 非功能需求

###### 环境需求

服务端环境，如表 2-11 所示：

表 2-11 服务端环境表

|  |  |
| --- | --- |
| 操作系统 | Windows 10 |
| Web 服务器 | Tomcat 7.0 |
| 数据库 | SQL Server 2008 |

###### 客户端环境，如表 2-12 所示：

表 2-12 客户端环境表

|  |  |
| --- | --- |
| 操作系统 | Windows 7 或其以上版本操作系统 |
| 浏览器 | IE 6.0 以及其以上版本（IE7.0、IE8.0） |
| 分辨率 | 最佳效果为 1366×768 像素 |

###### 用户接口：系统的总体结构采用菜单调用窗体，利用窗体来对信息进行管理、数据加工、信息。界面菜单需要统一的菜单风格，包括菜单图片、颜色，菜单栏目的划分，各种功能的菜单标题等。操作方式为鼠标操作加键盘操作。色彩方案设计为清新。不需要导航方式。

硬件接口：考虑到数据的备份等要求，需要外部存储设备，如管理员电脑的硬盘等， 这较易实现。

22

###### 软件接口：这里，主要考虑软件与操作系统、数据库管理系统的接口，以及局域网和互联网软件之间的数据交换。考虑到文档处理时有可能需要较常用的办公软件。例如 office 系列，所以应尽量实现他们之间的数据格式的自动转换。

通信接口：数据不管是在企业内部之间传输，还是公司与分公司之间进行远程数据传输时，防止数据被不法分析任意的修改和破坏，只有对信息解密的人员才能最终读取数据信息。这样才能最大程度的防止数据在传输的过程的安全保密性。

###### 性能需求

系统应具有快速响应的特性，用户打开界面和提交表单信息的平均响应时间应该低于3s。用户进行在线实时查询业务操作的数据处理时间应低于 5s。系统应具有高可靠性，客户端以及服务器正常运行的概率较高。目标系统用户界面操作简洁、灵活，系统用户帮助文档齐备，易于进行软件的使用。系统需要高吞吐量，因其表单信息的提交量巨大，硬件设备要求较高。

###### 安全需求

为了保障用户的信息安全，本系统对所有用户的密码采用 MD5 加密技术用来提升系统的保密性，并使用Token 认证方式，前端封装自定义 Axios 请求，在请求头携带Token， 后端进行请求拦截，验证Token 有效性，以防止用户信息泄露。前端编写路由拦截器，在路由跳转前验证是否获取到用户信息，若获取到正确的用户信息，即可进入系统后台，否则返回登陆页面。这体现了系统的保密性。

###### 系统运行过程中，在特定的运行条件与运行环境下，不能出现预期以外的系统故障导致系统崩溃。除此之外，系统还会对自身可能产生的一些的错误，拥有一定的容错机制， 当系统出现一些故障时，并不影响系统的正常运行。这体现了系统的可靠性与容错性。

23

# 总体设计

### 系统设计原则

###### 系统的整体构架需要以技术先进，安全保密、产品导向、设计合理、低成本以及设计规范等基本建设原则来进行规划。

1. 先进性：在产品设计方面，整个系统的软硬件设备设计符合高新技术发展趋势， 解压缩、传输等许多关键设备处于国际领先技术水平。系统设计在满足大部分功能的前提下拥有着前瞻性，同时在今后的一些时期里也保持着技术先进性。

###### 安全性：系统采用全面的安全防护措施，具有防病毒以及防黑客措施，同时在自然不可抗力因素下，具有很强的安全性。系统对接入的设备和用户采取严格的接入认真机制，来保证接入的安全性。

1. 合理性：在初始设计阶段，就充分的考虑系统扩容的要求，以便系统对运行环境具有非常好的匹配性，不依赖于单一固定的计算机设备。
2. 规范性：系统采用的各种工具类符合技术使用规范，对于编解码各种协议也符合国家标准。系统存在良好的互通性。

### 体系结构设计

###### 目前的系统设计中，主要采用的体系结构由 B/S 结构、C/S 结构以及 B/S 和 C/S 混合结构，本系统采用的体系结构设计是B/S 架构，因为B/S 具有以下优点：

其一，B/S 最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件，只要有一台能上网的电脑就能使用，[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)零安装、零维护。系统的扩展非常容易。其二， 由需求推动了 AJAX 技术的发展，它的程序也能在[客户端](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%E7%AB%AF)电脑上进行部分处理，从而大大的减轻了[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)的负担；并增加了交互性，能进行局部实时刷新。其三，B/S 结构主要利用了不断成熟的 Web 浏览器技术：结合浏览器的多种脚本语言和 ActiveX 技术，用通用浏览器实现原来需要复杂专用软件才能实现的强大功能，节约了开发成本。

###### 除此之外，B/S 架构应用广泛，在许多应用都实现，因此可以去借鉴，从而利用 B/S

架构更加简便。

24

表单管理 就业管理

用户管理 信息管理 咨询管理

日志记录

数据层

业务层

###### 系统技术架构设计，如图 3-2 所示：

前端UI

Html Css jQuery 图片

Ajax交互 Post请求 Get请求

储存过程 数据缓存

自定义函数

事务

读写数据库

权限控制

模块引擎渲染

展示层

数据库

MySQL MySQL

运行环境

操作系统

开发平台

Web服务器

图 3-2 技术架构设计图

###### 本系统结构一共分为六个层面：前端 UI、展示层、业务层、数据层、数据库、运行环境。下面是对其详细介绍：

前端 UI 主要包括 Html、CSS、jQuery 等网络技术来渲染前端数据，是系统界面展示的主要部分。展示层主要包括 html 等页面，显示系统数据，为用户提供前端操作 UI 界面。业务层主要包括业务接口开发与业务实体类开发，对数据层进行具体的操作，连接了整个系统结构。数据层主要包括数据访问层实体类开发、Mapper 文件的配置与数据访问层接口开发，其主要功能是访问数据库。数据库使用的是 MySQL 数据库，用以存储项目中的各种数据与信息。运行环境包括了 web 服务器、开发平台、操作系统，是项目测试的重点环境配置。日志记录主要是对业务层以及数据层里各种数据交互的消息进行记录，保存到一个xml 文件中。

###### 模块引擎渲染采用Spring Boot 以及Vue 框架，实现数据与页面分离，业务代码与逻辑代码的分离，这就大大提升了开发效率。对于 Ajax 交互来说，其包括了 Post、Get 等请求。Ajax 在浏览器与Tomcat 服务器之间使用异步传输请求，能使网页从服务器请求一些信息， 而不是整个页面。

25

### 系统功能结构设计

###### 本系统包括菜单栏系统、用户系统、课程信息系统、公司信息系统等等，如图 3-3 所示。以下是对其详细介绍。

其他管理系统

公司信息系统

课程信息系统

用户系统

表单系统

菜单栏系统

教育信息门户网站系统

图 3-3 教育信息门户网站系统功能结构设计图

###### 菜单栏目是对其他子系统的汇总，进行一些子系统的图标设计以及查看这些子系统的设计方法。菜单状态是对其他子系统显示状态的设置，例如对后台说明的信息可以隐藏。菜单栏子系统功能结构设计，如图 3-4 所示：

菜单状态

菜单栏目

菜单栏系统

图 3-4 菜单栏子系统功能结构设计图

###### 报名信息管理是对网页上提交的报名信息进行汇总，还可以对这些信息进行修改或删除操作，或者手动增加一条报名信息。统计分析是对报名信息进行汇总后分析，可以根据汇总的信息进行地理位置分析，专业分析等等。表单子系统功能结构设计，如图 3-5 所示：

报名信息管理

统计分析

表单系统

图 3-5 表单子系统功能结构设计图

###### 用户管理是对所有用户进行信息维护以及对用户信息的概览。角色管理是对用户进行角色分配以及查看用户角色的合理性。权限管理是对用户进行权限分配，不同的权限的用户登陆后台后显示的菜单栏栏目不同。用户子系统功能结构设计，如图 3-6 所示：

权限管理

角色管理

用户管理

用户系统

图 3-6 用户子系统功能结构设计图

26

###### 课程管理是对前台页面上展示课程信息的管理，包括对课程信息的添加、修改、删除等操作。课程类别根据不同的课程信息进行分类，并在前台页面上可以显示不同课程类别信息。课程信息管理子系统功能结构设计，如图 3-7 所示：

课程管理

课程类别

课程信息系统

图 3-7 课程信息子系统功能结构设计图

###### 公司基本信息是对公司的名称、成立年份、教资人数等基本信息进行管理，可以在前台页面中显示。公司概况信息是对员工的一些制度与规定，以及包括招聘员工所需要的信息，对后台进行管理说明等等。公司信息管理子系统功能结构设计，如图 3-8 所示：

公司概况信息

公司基本信息

公司信息系统

图 3-8 公司信息子系统功能结构设计图

###### 咨询内容管理是对前台页面游客咨询关于教育门户网站的其他信息的汇总，记录咨询信息的时间以及内容，同时可以对咨询信息状态做出反馈。就业管理是对公司一些毕业学员就业后所在企业的薪资以及公司进行前台页面展示以及后台管理。其他管理子系统功能结构设计，如图 3-9 所示：

就业管理

咨询内容管理

其他管理子系统

图 3-9 其他管理子系统功能结构设计图

27

# 数据库设计

### 概念结构设计

###### 设计思路

E-R 图模型的组成是由实体，属性和联系。用[菱形](https://baike.baidu.com/item/%E8%8F%B1%E5%BD%A2)框表示实体间的联系，联系以适当的含义命名，在菱形框中写名字，将参加联系的实体矩形框根据实际需要与菱形框用无向连线相连，并在连线上标明联系的类型，即 1-1、1-N 或 M-N。从数据需求分析中分析出系统的实体属性图，对实体之间的依赖关系进行了整合，得出系统 E-R 图。

###### 本系统的数据库设计以实际功能需求出发，考虑到实体之间的联系。数据库主要包括用户表、表单信息表、就业表、在线咨询表、教资信息表、公司信息表等用例表的设计。用户中超级管理员可以管理多个用户，后台用户可以查看多条不同种类信息内容，游客可以查看前台信息，即查看多条就业信息、教资信息、公司信息等等。

###### E-R 图

###### 系统实体关系，如图 4-1 所示：

|  |  |
| --- | --- |
| 菜单栏 | |
|  | n |

1

n

1

n

操作

n

n

咨询师

信息录入员

权限

普通管理员

超级管理员

教资信息

角色信息

n

操作

n

就业信息

1

n

1 1 操作

n

n

公司信息

操作

回复

n n n

课程类别

1 1

咨询信息

1 浏览 n

游客

n 1 操作

报名信息

n 1

课程信息

n

属于

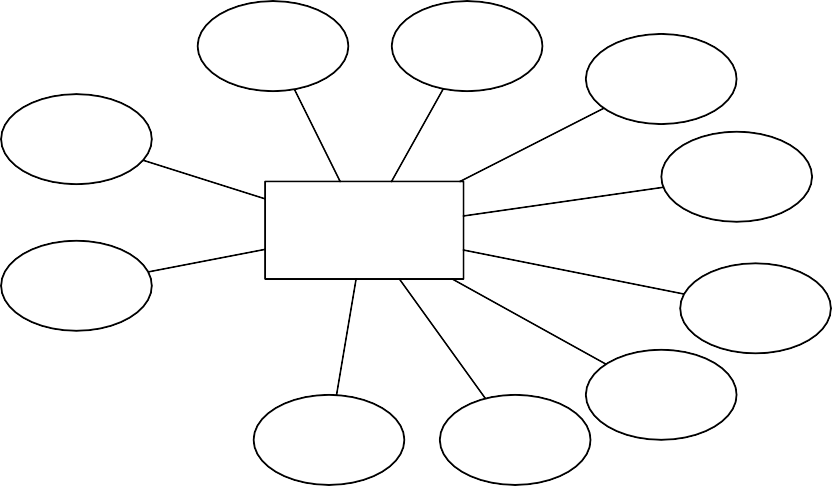
图 4-1 系统总 E-R 图

28

###### 用户信息的实体，如图 4-2 所示： 实体：用户

属性：用户编号，姓名，帐名，密码，创建时间，最后登陆时间，在职状态，权限组

###### id，是否在线，活跃时间。



密码

姓名

创建时间

账号

最后登录

时间

用户

用户编号

所属权限

在职状态

活跃时间

是否在线

图 4-2 用户实体图

###### 角色信息的实体，如图 4-3 所示： 实体：角色

属性：角色编号，角色名，用户组。

角色编号

用户组

图 4-3 角色实体图

角色名

角色

###### 权限信息的实体，如图 4-4 所示： 实体：权限

属性：权限编号，权限名，审批，用户组。

权限编号

权限名

审批

图 4-4 权限实体图

用户组

权限

29

###### 菜单栏目信息的实体，如图 4-5 所示： 实体：菜单栏目

属性：栏目名称，图标，图标路径，使用方法，栏目状态，所属权限。

图标

图标路径

使用方法

栏目状态

图 4-5 菜单栏目实体图

栏目名称

所属权限

菜单栏目

###### 栏目状态信息的实体，如图 4-6 所示： 实体：栏目状态

属性：栏目状态，栏目名称，备注。

栏目名称

栏目状态

备注

栏目状态

图 4-6 栏目状态实体图

###### 课程类别信息的实体，如图 4-7 所示： 实体：课程类别

属性：类别编号，类别名称，类别图片链接，状态。

类别图片链接

状态

类别编号

类别名称

课程类别

图 4-7 课程类别实体图

###### 课程信息的实体，如图 4-8 所示： 实体：课程

属性：课程编号，所属课程类别，课程名称，职称，是否免费，课时时长，所属专业， 图片链接，简介。

30

课程名称

课程编号

课程

图 4-8 课程实体图

所属专业

职称

课程时长

简介

是否免费

所属课程

类别

图片链接

###### 公司基本信息的实体，如图 4-9 所示： 实体：公司基本信息

属性：公司名称，公司地址，公司联系方式，教资人数，成立年份，公司前景。

公司地址

公司联系

电话

教资人数

成立年份

公司名称

公司前景

公司基本信息

图 4-9 公司基本信息实体图

###### 公司概况信息的实体，如图 4-10 所示： 实体：公司概况信息

属性：合作单位编号，合作类型，图片链接，合作单位名称，状态。

合作类型

图片链接

状态

合作单位

名称

合作单位

编号

公司概况信息

图 4-10 公司概况信息实体图

###### 报名信息的实体，如图 4-11 所示： 实体：报名信息

属性：报名信息编号，姓名，电话，专业，创建时间，处理人，处理时间，状态。

31

专业

处理人

状态

处理时间

报名信息

图 4-11 报名信息实体图

创建时间

电话

姓名

报名信息

编号

###### 统计分析的实体，如图 4-12 所示： 实体：统计分析

属性：报名信息编号，姓名，电话，专业。

姓名

报名信息

编号

联系电话

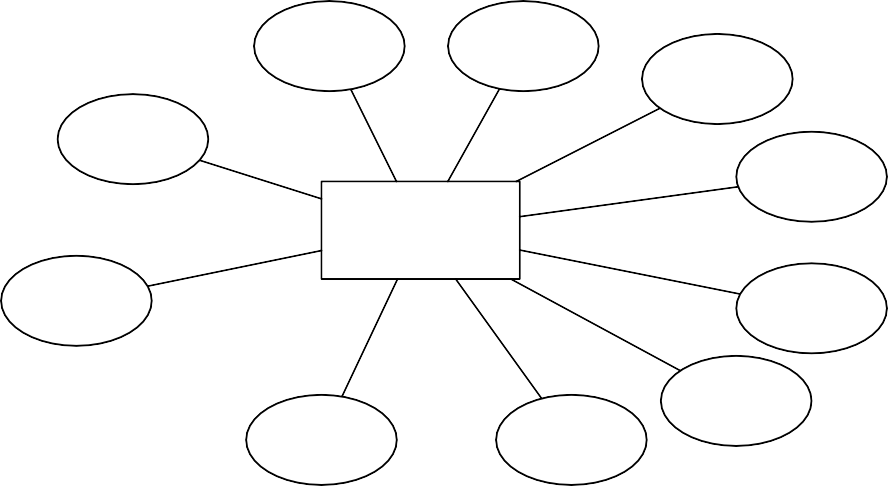
专业

统计分析

图 4-12 统计分析实体图

###### 咨询信息的实体，如图 4-13 所示： 实体：咨询

属性：咨询编号，开始时间，介绍时间，咨询内容，回复对象，回复内容，关键字， 来源，回访次数，状态。



开始时间

结束时间

咨询内容

咨询编号

回复对象

咨询

状态

回复内容

关键字

回访次数

来源

图 4-13 咨询信息实体图

32

###### 就业信息的实体，如图 4-14 所示： 实体：就业信息

属性：就业信息编号，姓名，入职公司，薪资，照片链接，毕业年份，备注，状态。

薪资

毕业年份

入职公司

照片链接

姓名

就业信息

编号

状态

就业信息

图 4-14 就业信息实体图

###### 教资信息的实体，如图 4-15 所示： 实体：教资信息

属性：教资信息编号，姓名，职称，介绍，照片链接，专业类型，状态。

介绍

职称

姓名

专业类型

教资信息

图 4-15 教资信息实体图

教资信息编号

照片链接

状态

### 逻辑结构设计

###### 设计思路

本系统使用的数据库是关系型数据库。逻辑结构设计是将E-R 图转换成关系模型。在确定本系统关系模式的属性后，将它们之间的联系转化成关系模型

###### 在确定关系模型之后，关系数据模型需要通过规范化理论来优化。通过消除冗余联系， 确定数据的依赖，对关系模型进行一定的分解等方法来优化逻辑模型。

33

###### 逻辑模型

1. 用户表：（编号、姓名、账号、密码、创建时间、最后登陆时间、在职状态、权限组id、是否在线、活跃时间）

###### 角色表：（编号、角色名、用户组）

1. 权限表：（编号、权限名、审批、用户组）

###### 菜单栏目表：（编号，栏目名称，图标，图标路径，使用方法，栏目状态，所属权限）

1. 栏目状态表：（编号，栏目状态，栏目名称，备注）

###### 课程类别表：（编号，类别名称，类别图片链接，状态）

1. 课程信息表：（编号，课程名称，所属专业，课时时长，职称，是否免费，图片链接，简介，所属课程类别）

###### 公司基本信息表：（编号，公司名称，公司地址，公司联系方式，教资人数，成立年份，公司前景）

1. 公司概况信息表：（编号，合作类型，图片链接，合作单位名称，状态）

###### 报名信息表：（编号，姓名，电话，专业，创建时间，处理人，处理时间，状态）

1. 咨询信息表：（编号，开始时间，结束时间，咨询内容，回复对象，回复内容， 关键字，来源，回访次数，状态）

###### 就业信息表：（编号，姓名，入职公司，薪资，照片链接，毕业年份，备注，状态）

1. 教资信息表：（编号，姓名，职称，简介，照片链接，专业类型，状态） 备注：含下划线字段为该表的主键。

### 物理结构设计

###### 存储方式

在 MySQL 数据库，使用的是默认的数据库存储引擎 InnoDB。在 InnoDB 中存在着缓冲管理，通过缓冲池，将索引和数据全部缓存起来，加快查询的速度对于 InnoDB 类型的表，其数据的物理组织形式是聚簇表。所有的数据按照主键来组织。数据和索引放在一块， 都位于 B+数的叶子节点上。

34

###### 存储结构

1. 用户表admin，如表 4-1 所示：

表 4-1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | user\_name | 姓名 | varchar | 10 |  |
| 3 | name | 账号 | varchar | 30 |  |
| 4 | pwd | 密码 | varchar | 32 |  |
| 5 | creat\_time | 创建时间 | varchar | 255 |  |
| 6 | last\_time | 最后登陆时间 | varchar | 255 |  |
| 7 | state | 在职状态 | tinyint | 1 |  |
| 8 | group\_id | 权限组 id | tinyint | 4 |  |
| 9 | on\_line | 是否在线 | tinyint | 1 |  |
| 10 | active | 活跃时间 | varchar | 255 |  |

###### 角色表auth\_group，如表 4-2 所示：

表 4-2 角色表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | mediumint | 8 | 主键 |
| 2 | uid | 角色名 | mediumint | 8 |  |
| 3 | group\_user | 用户组 | mediumint | 8 |  |

###### 权限表auth\_group\_access，如表 4-3 所示：

表 4-3 权限表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | mediumint | 8 | 主键 |
| 2 | title | 权限名 | char | 100 |  |
| 3 | status | 状态 | tinyint | 1 |  |
| 4 | rules | 审批 | varchar | 255 |  |

###### 菜单栏目表menu，如表 4-4 所示：

表 4-4 菜单栏目表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 8 | 主键 |
| 2 | name | 栏目名称 | varchar | 20 |  |

35

续表 4-4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 3 | picture | 图标 | varchar | 255 |  |
| 4 | url | 图标路径 | varchar | 255 |  |
| 5 | function | 使用方法 | varchar | 40 |  |
| 6 | status | 栏目状态 | tinyint | 1 |  |
| 7 | group\_user | 所属权限 | tinyint | 8 |  |

###### 栏目状态表menu\_status，如表 4-5 所示：

表 4-5 栏目状态表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 8 | 主键 |
| 2 | status | 栏目状态 | tinyint | 1 |  |
| 3 | name | 栏目名称 | varchar | 255 |  |
| 4 | ps | 备注 | varchar | 255 |  |

###### 课程类别表course\_type，如表 4-6 所示：

表 4-6 课程类别表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 类别名称 | char | 30 |  |
| 3 | url | 类别图片链接 | mediumint | 0 |  |
| 4 | status | 状态 | tinyint | 1 |  |

###### 课程表course，如表 4-7 所示：

表 4-7 课程表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 课程名称 | varchar | 30 |  |
| 3 | related\_majors | 所属专业 | varchar | 10 |  |
| 4 | time | 课程时长 | varchar | 255 |  |
| 5 | career | 职称 | varchar | 30 |  |
| 6 | is\_free | 是否免费 | tinyint | 1 |  |
| 7 | url | 图片链接 | varchar | 255 |  |
| 8 | introduce | 简介 | varchar | 255 |  |
| 9 | c\_name | 所属课程类别 | varchar | 30 |  |

36

###### 公司基本信息表company，如表 4-8 所示：

表 4-8 公司基本信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 公司名称 | varchar | 30 |  |
| 3 | address | 公司地址 | varchar | 50 |  |
| 4 | tel | 公司联系方式 | varchar | 11 |  |
| 5 | people | 教资人数 | varchar | 30 |  |
| 6 | time | 成立年份 | varchar | 10 |  |
| 7 | future | 公司前景 | varchar | 255 |  |
| 8 | picture | 公司照片 | varchar | 255 |  |

###### 公司概况信息表cooperation，如表 4-9 所示：

表 4-9 公司概况信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 合作单位名称 | varchar | 50 |  |
| 3 | type | 合作类型 | varchar | 10 |  |
| 4 | url | 图片链接 | varchar | 255 |  |
| 5 | status | 状态 | tinyint | 1 |  |

###### 报名信息表enroll，如表 4-10 所示：

表 4-10 报名信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 姓名 | varchar | 9 |  |
| 3 | tel | 电话 | varchar | 11 |  |
| 4 | professional | 专业 | varchar | 255 |  |
| 5 | creat\_time | 创建时间 | varchar | 255 |  |
| 6 | deal\_name | 处理人 | varchar | 255 |  |
| 7 | deal\_time | 处理时间 | varchar | 255 |  |
| 8 | status | 状态 | tinyint | 5 |  |

37

###### 咨询表consult，如表 4-11 所示：

表 4-11 咨询表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | start\_time | 开始时间 | varchar | 255 |  |
| 3 | end\_time | 结束时间 | varchar | 255 |  |
| 4 | context | 咨询内容 | varchar | 255 |  |
| 5 | deal\_name | 回复对象 | varchar | 10 |  |
| 6 | deal\_context | 回复内容 | varchar | 255 |  |
| 7 | words | 关键字 | varchar | 30 |  |
| 9 | number | 回访次数 | tinyint | 10 |  |
| 10 | status | 状态 | tinyint | 1 |  |

###### 就业信息表employment，如表 4-12 所示：

表 4-12 就业信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 姓名 | varchar | 10 |  |
| 3 | company | 入职公司 | varchar | 255 |  |
| 4 | salary | 薪资 | varchar | 20 |  |
| 5 | url | 照片链接 | varchar | 255 |  |
| 6 | graduation | 毕业年份 | varchar | 255 |  |
| 7 | ps | 备注 | varchar | 255 |  |
| 8 | status | 状态 | int | 3 |  |

###### 教资信息表teacher，如表 4-13 所示：

表 4-13 教资信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段含义 | 数据类型 | 字段长度 | 备注 |
| 1 | id | 编号 | tinyint | 3 | 主键 |
| 2 | name | 姓名 | varchar | 5 |  |
| 3 | professional | 职称 | varchar | 20 |  |
| 4 | introduce | 简介 | varchar | 255 |  |
| 5 | url | 照片链接 | varchar | 255 |  |
| 6 | type | 专业类型 | varchar | 10 |  |
| 7 | status | 状态 | int | 3 |  |

38

# 界面设计

### 界面关系图

###### 本系统界面设计的原则是注重系统用户的交互体验，为系统用户提供美观、便捷的 UI

操作界面。本系统UI 界面设计的主要流程如下：

###### 页面视觉设计：视觉设计阶段根据信息网站建设需求，设置网上报名模块，并且便于用户进行前台页面浏览设置一个置顶的浮窗，并且设计出实用、美观的界面图形。

1. 前端界面开发：在视觉设计的基础上，进行前台页面的编码工作。主要利用了

###### HTML5、CSS 等技术，开发出前端的UI 界面。

系统开始打开的第一个页面为登陆页面，登录页面登陆成功后进入后台主页，主页左边有用户管理等信息管理栏目，进入各个栏目里对相关信息进行所需要的操作，操作完成后结束。系统界面关系，如图 5-1 所示：

登陆失败

后台用户

登录界面

登陆成功

主页

进入

就业管理

课程分类管理

修

查询

用户管理

修改

添加

用户添加页面

用户修改页面

查询

表单管理

修改

添加

表单添加页面

表单修改页面

|  |  |  |  |
| --- | --- | --- | --- |
|  | 公司信息管理 | |  |
| 改 | | 添加 | |

信息修改页面

信息添加页面

就业详情页面

分类详情页面

图 5-1 系统界面关系图

39

### 界面设计成果

###### 主界面

1. 前台主界面主要包括首页，课程，师资力量，就业学子，网上报名，关于，在线咨询等功能按钮，展示了关于此公司各种信息。前台主界面，如图 5-2 所示：

图 5-2 前台主界面图

40

###### 后台主界面：后台管理主界面包括表单系统，公司概况信息管理，教师信息管理， 用户管理，课程管理，课程类别管理，就业管理等功能按钮。后台主界面，如图 5-3 所示：

图 5-3 后台主界面图

###### 子界面

1. 前台子界面

###### 前台课程中Java 开发子界面，如图 5-4 所示：

图 5-4 Java 开发子界面图

41

###### 前台师资力量子界面，如图 5-5 所示：

图 5-5 师资力量子界面图

###### 前台网上报名子界面，如图 5-6 所示：

图 5-6 网上报名子界面图

42

###### 后台子界面

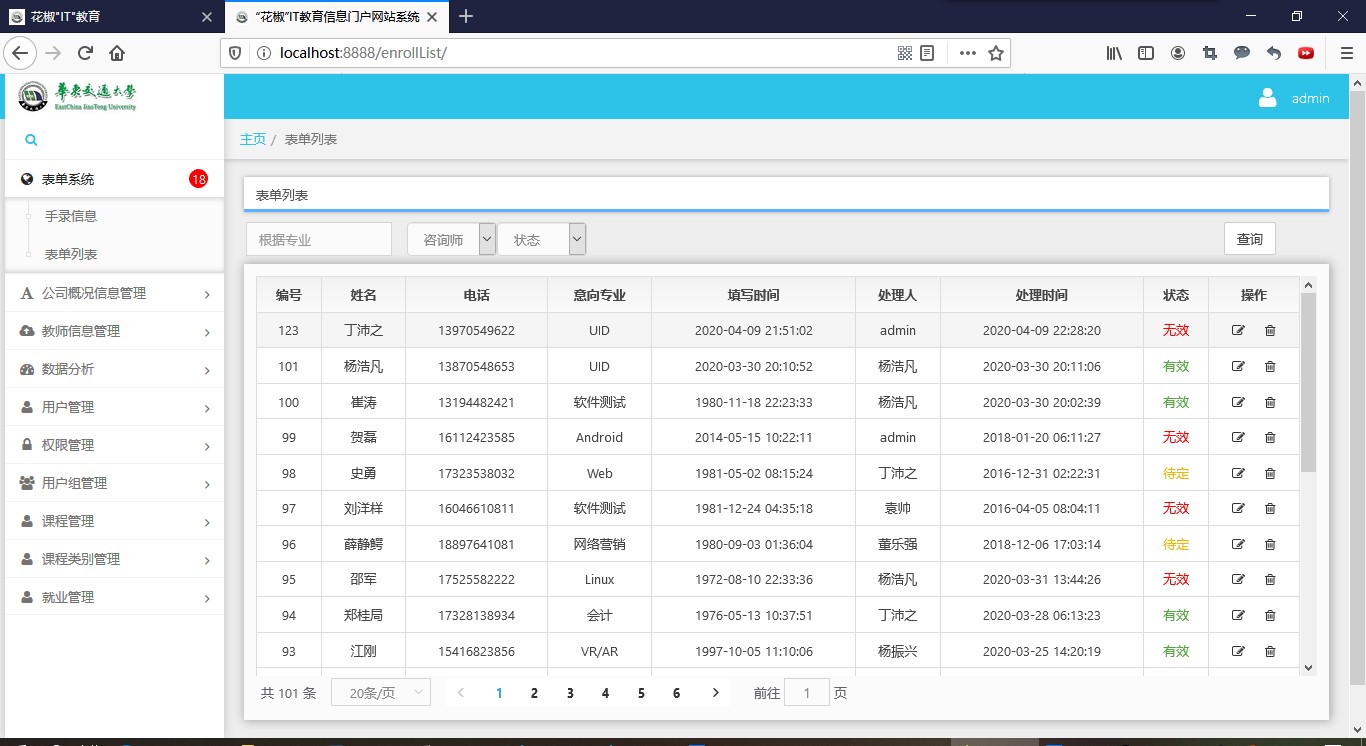
1. 后台表单系统子界面，如图 5-7 所示：

图 5-7 表单子界面图

###### 后台用户管理子界面，如图 5-8 所示：

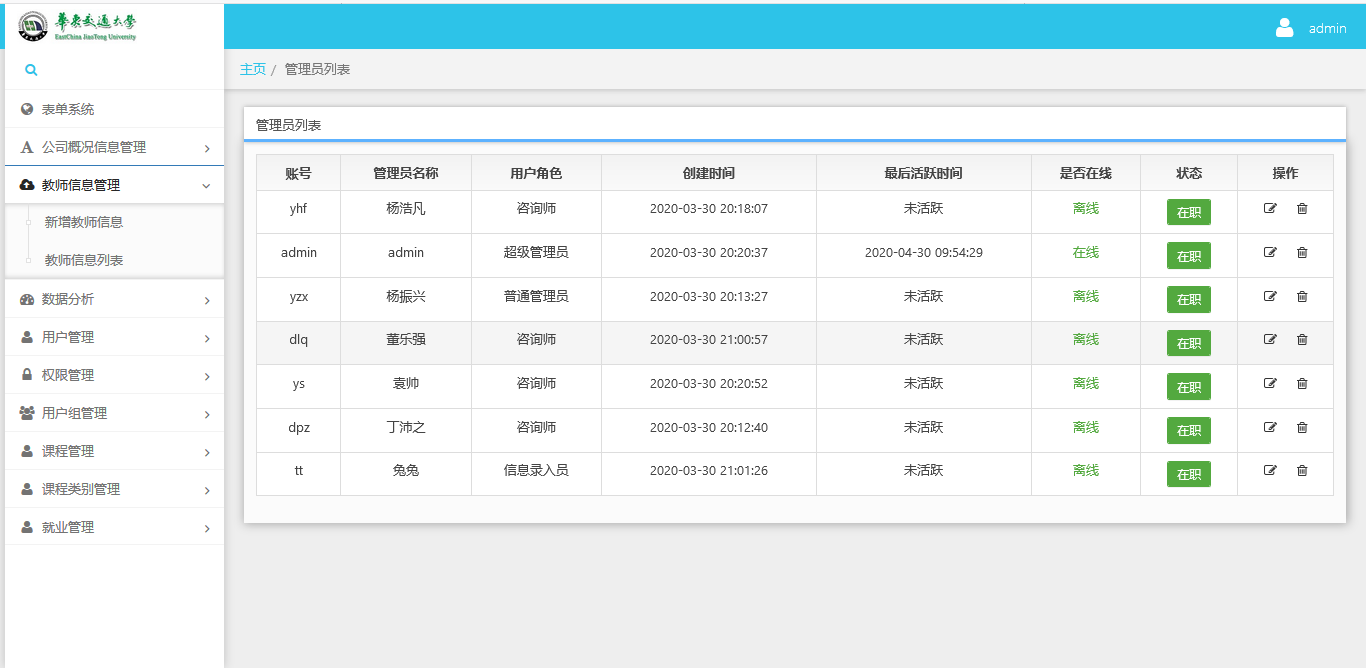


图 5-8 用户管理子界面图

43

###### 后台统计分析子界面，如图 5-9 所示：



图 5-9 统计分析子界面图

###### 后台教师信息子界面，如图 5-10 所示：



图 5-10 教师信息子界面图

44

# 详细设计

### 系统主要功能模块介绍

###### 系统分为菜单栏系统，表单系统，用户系统，课程信息系统，公司信息系统，咨询内容管理，就业管理等模块，如图 6-1 所示。

后台模块

就业管理

图 6-1 功能模块结构图

菜单栏系统

栏目状态

菜单栏目

表单系统

报名信息管理

统计分析

用户系统

权限管理

角色管理

用户管理

课程信息系统

课程管理

课程类别

公司信息系统

公司概况信息

公司基本信息

咨询内容管理

###### 菜单栏系统包括菜单栏目，栏目状态等功能，超级管理员根据用户权限以及菜单栏状态来设置用户可用栏目。

1. 表单系统包括统计分析，报名信息管理等功能，报名信息管理是根据前台页面上提交表单进行管理，根据表单的信息进行数据分析之后形成表格。

###### 用户系统的功能有权限、角色管理与用户管理，用户管理包括了创建用户、登陆、退出登陆、用户信息的增删改查等等，角色管理包括对用户进行角色选择，权限管理包括对用户角色进行权限分配。

1. 课程信息系统包括课程管理，课程类别等功能，课程管理是对页面上课程信息展示来进行管理，包括对其信息的增删改查，课程类别是根据不同类别对课程进行分类，并在页面上展示不同类别。

45

###### 公司信息系统包括公司基本信息，公司概况信息，公司基本信息包括对公司的名称、成立年份、教师资历等信息的展示以及增删改查，公司概况信息包括对员工的制度信息，对成员的管理制度以及对招聘教师的信息管理，管理分为增加、修改、删除此类信息。

1. 咨询内容管理包括对咨询信息进行记录，包括咨询的时间，时长，地址，咨询师的记录，也可以设置咨询内容的状态。
2. 就业管理就是对毕业走上社会岗位的优秀学员进行汇总以及展示的管理。

### 报名信息管理模块

###### 功能结构设计

报名信息管理模块结构设计，如图 6-2 所示。修改报名信息状态包括处理、未处理等状态，根据报名信息状态来进行删除报名信息。查询报名信息需要输入相关信息进行查询。报名信息审批是对所有的报名信息进行检验，比对数据库，查看是否有重复的数据。

报名信息管理

修改报名信息

删除报名信息

查询报名信息

报名信息审批

手录报名信息

图 6-2 报名信息管理模块结构设计图

###### 类图设计

报名信息管理模块具有新增，修改，删除，查询等功能，其中删除通过 id 来进行删除， 查询通过id 进行查询报名信息。报名信息管理模块类图设计，如图 6-3 所示：

46

|  |  |
| --- | --- |
| 报名信息控制类 | |
| PK | |
| addId( ) delId( ) updateId( )  SelectId( ) | |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 报名信息实体类 | | |  |
| PK | | |  |
| id: int name: string tel: string  professional: string  create\_time: string  deal\_time:  string  status: int | | |  |
| setter/getter( PK  ) | | |  |
|  | |  | |
|  | 报名信息数据库处理类 | | |
|  | PK | | |
|  | add( ) update( ) deleteById( )  SelectById( ) | | |

图 6-3 报名信息管理模块类图

add( )

del( ) Update( ) Select( )

PK

报名信息服务类

###### 顺序图设计

用户通过登陆页面进入后台系统，进入报名信息管理模块，进行增删改查的相关处理， 进入数据库处理并更新结果后，弹出提示信息，返回列表。报名信息管理模块顺序图设计， 如图 6-4 所示：

47

超级管理员

登录系统

报名信息管理模块

相关处理

数据库模块

**1.**输入用户名和密码

**2.**提交用户名

1. 显示报名信息处理信息
2. 提交增删改操作的报名信息
3. 数据库处理并更新结果
4. 更新成功

**7.**提示操作成功信息

图 6-4 报名信息管理模块时序图

###### 核心处理流程设计

用户登陆成功后，进入报名信息管理模块，可以选择删除或者检索信息，根据系统提示来完成处理流程。报名信息管理模块核心处理流程设计，如图 6-5 所示：

登陆成功

点击进入报名

信息管理

删除信息

检索信息

信息状态是

否为已处理

是

刷新列表信息

否

不允许删除信

息

图 6-5 报名信息管理模块活动图

48

### 统计分析模块

###### 功能结构设计

统计分析模块，如图 6-6 所示。汇总数据主要是刷新报名信息数据，直接点击汇总按钮。按关键词分析是通过不同的关键词来分析报名信息数据，再返回到数据台上。

统计分析

咨询师分析

汇总数据

专业分析

图 6-6 统计分析模块结构设计图

###### 类图设计

统计分析模块具有检索并分析数据功能，主要是对表单系统以及咨询师处理表单信息的检索。统计分析模块类图设计，如图 6-7 所示：



add( )

del( ) Update( ) Select( )

PK

统计分析服务类

addId( ) delId( ) updateId( )

SelectId( )

PK

统计分析控制类



|  |
| --- |
| 统计分析实体类 |
| PK |
| id: string  name: string professional: string phone: string |
| PK setter/getter( ) |

|  |  |
| --- | --- |
|  |  |
| 统计分析数据库处理类 | |
| PK | |
| add( ) update( ) deleteById( )  SelectById( ) | |

图 6-7 统计分析模块类图

49

###### 顺序图设计

用户通过登陆页面进入后台系统，进入统计分析模块，进行检索信息，进入数据库查询并更新结果后，弹出提示信息，返回列表。统计分析模块顺序图设计，如图 6-8 所示：



超级管理员

登录系统

统计分析模块

相关处理

**1.**输入用户名和密码

**2.**提交用户名

1. 显示统计分析表格
2. 生成统计分析数据
3. 提示分析数据已生成

图 6-8 统计分析模块时序图

###### 核心处理流程设计

用户登陆成功后，直接点击统计分析模块，点击汇总数据以查看处理结果。统计分析模块核心处理流程设计，如图 6-9 所示：

登陆成功

点击统计分析

查看统计分析

数据

图 6-9 统计分析模块活动图

50

### 用户管理模块

###### 功能结构设计

登陆是进入后台必要的步骤，在登陆页面输入账号密码以及验证码之后，点击登陆按钮，页面上的参数与数据库数据进行比对，若符合，则可以登陆后台，否则登陆失败。新增用户信息是添加一个后台用户，并且添加用户的所有信息，包括其登陆后台的账号密码。修改用户信息是修改用户相关信息，点击修改按钮，选择要修改的相应栏目，点击完成修改即可。删除用户信息可直接点击删除按钮。用户状态排序是点击排序按钮，数据会根据用户状态来进行排序。用户管理模块，如图 6-10 所示：

用户管理

登录

修改用户信息

新增用户信息

删除用户信息

用户状态修改

图 6-10 用户管理模块结构设计图

###### 类图设计

用户管理模块具有新增，修改，删除，查询等功能，其中删除通过 id 来进行删除，查询通过id 进行查询用户信息。用户管理模块类图设计，如图 6-11 所示：

51





|  |  |
| --- | --- |
| 用户控制类 | |
| PK | |
| adduser( ) deluser( ) updateuser( )  Selectuser( ) | |
|  |  |

|  |  |
| --- | --- |
| 用户实体类 | |
| PK | |
| id: string | |
| name: string | |
| pwd: string | |
| state：string | |
| on\_line：int | |
| active：string | |
| PK | setter/getter( ) |

|  |
| --- |
| 用户服务类 |
| PK |
| add( )  del( ) Update( ) Select( ) |

|  |  |
| --- | --- |
|  |  |
| 用户数据库处理类 | |
| PK | |
| add( ) update( ) deleteByid( )  SelectByid( ) | |

图 6-11 用户管理模块类图



###### 顺序图设计

用户通过登陆页面进入后台系统，进入用户管理模块，进行增删改查的相关处理，进入数据库处理并更新结果后，弹出提示信息，返回列表。用户管理模块顺序图设计，如图6-12 所示：

52

超级管理员

登录系统

用户管理模块

相关处理

数据库模块

**1.**输入用户名和密码

**2.**提交用户名

1. 显示用户管理处理信息
2. 提交增删改操作的用户信息
3. 数据库处理并更新结果
4. 更新成功

**7.**提示操作成功信息

图 6-12 用户管理模块时序图

###### 核心处理流程设计

用户登陆成功后，进入用户管理模块，创建一个新的用户，给其分配权限与角色，在退出系统，使用新用户登陆来验证新用户的合法性。用户管理模块核心处理流程设计，如图 6-13 所示：

登陆成功

点击用户列表

更新用户列表

是否验证

新用户合法性

创建账号

分配权限

是

否

否

是否显示正确的

用户权限内容

使用新用户登 退出系统录

是

创建成功

图 6-13 用户管理模块活动图

53

### 课程管理模块

###### 功能结构设计

课程管理模块，如图 6-14 所示。新增课程类别是输入课程类别名称，添加此课程。删除课程类别是选择不需要的课程类别直接点击删除按钮。按课程类别排序可以更好的整理相同类别的课程信息，以便管理。

课程类别管理

删除课程类别

专业排序

图 6-14 课程管理模块结构设计图

新增课程类别

###### 类图设计

课程管理模块具有新增，修改，删除，查询等功能，其中删除通过 id 来进行删除，查询通过id 进行查询课程信息。课程管理模块类图设计，如图 6-15 所示：



add( )

del( ) Update( ) Select( )

PK

课程类别服务类

|  |  |
| --- | --- |
| 课程类别控制类 | |
| PK | |
| addName( ) delName( ) updateName( )  SelectName( ) | |
|  |  |

|  |  |
| --- | --- |
| 课程类别实体类 | |
| PK | |
| id：int | |
| name：string | |
| status：int | |
| url：string | |
| PK | setter/getter( ) |

|  |  |
| --- | --- |
|  |  |
| 课程类别数据库处理类 | |
| PK | |
| add( ) update( ) deleteByid( )  SelectByid( ) | |

图 6-15 课程管理模块类图

54

###### 顺序图设计

用户通过登陆页面进入后台系统，进入课程管理模块，进行增删改查的相关处理，进入数据库处理并更新结果后，弹出提示信息。课程管理模块顺序图设计，如图 6-16 所示：

超级管理员

登录系统

课程类别模块

相关处理

数据库模块

**1.**输入用户名和密码

**2.**提交用户名

1. 显示课程类别处理信息
2. 提交增删改操作的课程类别
3. 数据库处理并更新结果
4. 更新成功

**7.**提示操作成功信息

图 6-16 课程管理模块时序图

###### 核心处理流程设计

用户登陆成功后，进入课程管理模块，可以选择增删改信息，根据系统提示来完成处理流程。课程管理模块核心处理流程设计，如图 6-17 所示：

55

返回控制台

添加信息

否

登陆成功

选择课程类别

管理

信息是否符

合要求

修改信息

是

删除信息

录入信息成功

刷新列表信息

图 6-17 课程管理模块活动图

### 公司信息管理模块

###### 功能结构设计

公司信息管理模块，如图 6-18 所示：新增基本信息是点击新增按钮，输入公司的基本信息，完成信息的新增。删除基本信息是删除不需要使用的信息，直接删除。修改基本信息是基本信息有错误或者与实际不符，则需要修改其信息。基本信息审批是管理公司基本信息在前台页面的展示情况，包括隐藏和显示两个选择。

公司基本信息

新增基本信息

删除基本信息

修改基本信息

基本信息审批

图 6-18 公司信息管理模块结构设计图

56

###### 类图设计

公司信息管理模块具有新增，修改，删除，查询等功能，其中删除通过 id 来进行删除， 查询通过id 进行查询公司信息。公司信息管理模块类图设计，如图 6-19 所示：



add( )

del( ) Update( ) Select( )

add( )

update( ) deleteByname( ) SelectByname( )

PK

PK

公司基本信息数据库处理类

公司基本信息服务类

|  |  |
| --- | --- |
| 公司基本信息控制类 | |
| PK | |
| addName( ) delName( ) updateName( )  SelectName( ) | |
|  |  |

|  |  |
| --- | --- |
| 公司基本信息实体类 | |
| PK | |
| name: string | |
| address: string | |
| phone: string | |
| future: string | |
| number: string | |
| time: string | |
| PK | setter/getter( ) |

图 6-19 公司信息管理模块类图

###### 顺序图设计

用户通过登陆页面进入后台系统，进入公司信息管理模块，进行增删改查的相关处理， 进入数据库处理并更新结果后，弹出提示信息，返回列表。公司信息管理模块顺序图设计， 如图 6-20 所示：

57

超级管理员

登录系统

公司基本信息模块

相关处理

数据库模块

**1.**输入用户名和密码

**2.**提交用户名

1. 显示公司基本信息处理
2. 提交增删改操作的公司基本信息
3. 数据库处理并更新结果
4. 更新成功

**7.**提示操作成功信息

图 6-20 公司信息管理模块时序图

###### 核心处理流程设计

用户登陆成功后，进入公司信息管理，可以选择增删改信息，根据系统提示来完成处理流程。公司信息管理模块核心处理流程设计，如图 6-21 所示：

返回控制台

添加信息

否

登陆成功

选择公司

信息管理

信息是否

符合要求

修改信息

是

删除信息

录入信息成功

刷新列表信息

图 6-21 公司信息管理模块活动图

58

### 就业管理模块

###### 功能结构设计

就业管理模块，如图 6-22 所示：就业信息审批是对前台页面信息的展示选择，将某些就业信息隐藏或展示。新增就业信息是输入新的学员就业信息，完成信息的新增。删除就业信息是将不需要的信息，直接点击删除按钮。修改就业信息是将需要修改的信息进行处理，从而将信息更加准确。

就业管理

就业信息审批

新增就业信息

修改就业信息

概况就业审批

图 6-22 就业管理模块结构设计图

###### 类图设计

就业管理模块具有新增，修改，删除，查询等功能，其中删除通过 id 来进行删除，查询通过id 进行查询就业信息。就业管理模块类图设计，如图 6-23 所示：

59

|  |  |
| --- | --- |
| 就业实体类 | |
| PK | |
| id: string | |
| namet: string | |
| company: string | |
| time: string | |
| illustrate: string | |
| picture: string | |
| money: string | |
| PK | setter/getter( ) |

图 6-23 就业管理模块类图



就业数据库处理类

PK

add( ) update( ) deleteById( )

SelectById( )

add( )

del( ) Update( ) Select( )

PK

就业服务类

addId( ) delId( ) updateId( )

SelectId( )

PK

就业控制类

###### 顺序图设计

用户通过登陆页面进入后台系统，进入就业管理模块，进行增删改查的相关处理，进入数据库处理并更新结果后，弹出提示信息，返回列表。就业管理模块顺序图设计，如图6-24 所示：

60

超级管理员

登录系统

就业管理模块

相关处理

数据库模块

**1.**输入用户名和密码

**2.**提交用户名

1. 显示就业处理信息
2. 提交增删改操作的就业信息
3. 数据库处理并更新结果
4. 更新成功

**7.**提示操作成功信息

图 6-24 就业管理模块时序图

###### 核心处理流程设计

用户登陆成功后，进入公司信息管理，可以选择增删改信息，根据系统提示来完成处理流程。就业管理模块核心处理流程设计，如图 6-25 所示：

返回控制台

添加就业信息

否

选择就

业管理

信息是否

符合要求

修改就业信息

是

登陆成功

删除就业信息

录入信息成功

刷新列表信息

图 6-25 就业管理模块活动图

61

# 编码

### 代码实现与核心算法

###### 表单模块

（1） 获取表单：获取数据库所有表单信息，前端通过此接口获取数据信息。除此之外， 还包括了表单列表的分页查询以及模糊查询，前端通过此接口查询信息。

###### @PostMapping("/getEnrollList")

public RespEntity getEnrollList(@RequestParam("professional") String professional,

###### @RequestParam("deal\_name") String deal\_name, @RequestParam("status") String status, @RequestParam("pageSize") String pageSize, @RequestParam("pageIndex") String pageIndex) {

Map<String,Object> condition = new HashMap<>(); condition.put("professional",professional); condition.put("deal\_name",deal\_name); condition.put("status",status);

###### condition.put("start", (Integer.parseInt(pageIndex)-1) \* Integer.parseInt(pageSize)); condition.put("pageSize", Integer.parseInt(pageSize));

Integer totalSize = enrollService.getTotalSize(condition); List<Enroll> enrollList = enrollService.getEnrollList(condition); Map<String, Object> result = new HashMap<>(); result.put("totalSize", totalSize);

###### result.put("enrollList", enrollList);

return new RespEntity(RespCode.SUCCESS, result);

###### }

62

###### 用户模块

1. 登陆：先判断验证码的正确性，再来判断用户的账号以及密码是否正确，若错误， 则会提示相关信息以便纠正。

@PostMapping("/login")

public RespEntity login(@RequestParam("name") String name,

@RequestParam("pwd") String pwd, @RequestParam("code") String code, @RequestParam("last\_ip") String last\_ip) {

String sessionCode = codeString;

if(!StringUtils.equalsIgnoreCase(code, sessionCode)) { //忽略验证码大小写return new RespEntity(RespCode.ERROR, "哦豁！验证码错了呀！");

}

String MD5\_PWD = MD5.encodeByMD5(pwd); System.out.println(name);

Admin = adminService.login(name); if(admin == null) {

return new RespEntity(RespCode.ERROR, "哦吼！猜错了，木有这个人啊！");

}else {

if(!admin.getPwd().equals(MD5\_PWD)) {

return new RespEntity(RespCode.ERROR, "哎呦！猜错了！现在输入正确的密码！");

}else if(admin.getState() != 1){

return new RespEntity(RespCode.ERROR, "流着泪说分手，你已不在这干了！");

}else {

//登陆成功，生成 token,将 token 返回给客户端String token = JwtUtil.sign(admin.getId(), name, pwd); if(token != null) {

admin.setLast\_ip(last\_ip); admin.setOn\_line(1);

admin.setActive(Long.toString(new Date().getTime())); Integer result = adminService.updateUser(admin); if(result == 1) {

Token tokenObj = new Token(token);

return new RespEntity(RespCode.SUCCESS, tokenObj);

}

}

}

}

return new RespEntity(RespCode.FAIL);

}

63

###### 获取登陆用户信息：获取数据库登录用户所有信息，先获取 Token 验证其登陆的有效性，前端通过此接口获取数据信息。

@GetMapping("/getUserInfo")

public RespEntity getUserInfo(HttpServletRequest request) { String token = request.getHeader("token");

Integer id = JwtUtil.getUserId(token); if(id != null) {

Admin admin = adminService.getUserInfo(id);

return new RespEntity(RespCode.SUCCESS, admin);

}

return new RespEntity(RespCode.ERROR);

}

1. 退出登陆：通过id 来实现对用户进行退出系统的功能。

@PostMapping("/logout")

public RespEntity logout(@RequestParam("id") Integer id) { Admin admin = new Admin();

admin.setId(id); admin.setOn\_line(0);

Integer result = adminService.updateUser(admin); if(result == 1) {

return new RespEntity(RespCode.SUCCESS, "退出成功");

}

return new RespEntity(RespCode.ERROR, "退出失败");

}

###### 修改密码：修改的密码在数据库中保存的样式为一串字符，原因是密码使用 MD5

加密方式。

@PostMapping("/resetPassword")

public RespEntity resetPassword(@RequestParam("id") String id, @RequestParam("pwd") String pwd) { String MD5\_PWD = MD5.encodeByMD5(pwd);

Admin admin = new Admin(); admin.setId(Integer.parseInt(id)); admin.setPwd(MD5\_PWD);

Integer result = adminService.updateUser(admin); if(result == 1) {

return new RespEntity(RespCode.SUCCESS);

}else {

return new RespEntity(RespCode.ERROR, "修改密码失败");

}

}

64

###### 验证码：登路页面自动生成字母加数字的验证码。

@GetMapping("/validateCode")

public void validateCode(HttpServletResponse response) throws IOException {

//设置响应的类型格式为图片格式response.setContentType("image/jpeg");

//禁止图片缓存response.setHeader("Pragma", "no-cache");

response.setHeader("Cache-Control", "no-cache"); response.setDateHeader("Expires", 0);

VerificationCodeImg vCode = new VerificationCodeImg(130, 34, 4, 50); codeString = vCode.getCode(); vCode.write(response.getOutputStream());

}

* + 1. 就业管理模块

###### （1） 添加就业信息：在就业管理模块新增一条就业信息并存入数据库。

@PostMapping("/addEmployment")

public RespEntity addEmployment(@RequestParam("name") String name,

@RequestParam("company") String company, @RequestParam("salary") String salary, @RequestParam("url") String url, @RequestParam("graduation") String graduation, @RequestParam("ps") String ps, @RequestParam("status") String status){

Employment employment = new Employment(); employment.setName(name); employment.setCompany(company); employment.setSalary(salary); employment.setUrl(url); employment.setGraduation(graduation); employment.setPs(ps); employment.setStatus(Integer.parseInt(status));

Integer result = employmentService.addEmployment(employment); if(result == 1) {

return new RespEntity(RespCode.SUCCESS, "添加成功");

}else {

return new RespEntity(RespCode.ERROR, "添加失败");

}

}

65

###### 公司信息模块

（1） 删除公司信息：通过id 来进行删除一条公司信息。

@PostMapping("/deleteCooperation")

public RespEntity deleteCooperation(@RequestParam("id") String id){

Integer result=cooperationService.deleteCooperation(Integer.parseInt(id)); if(result==1){

return new RespEntity(RespCode.SUCCESS,"删除成功");

}else{

return new RespEntity(RespCode.ERROR,"删除失败");

}

}

###### 课程模块

（1） 修改课程信息：动态修改课程信息，当一条信息中某个数据未修改时，保留其原始数据。

@PostMapping("/updateCourseType")

public RespEntity updateCourseType(@RequestParam("id") String id, @RequestParam("name") String name, @RequestParam("status") String status, @RequestParam("url") String url){

CourseType courseType = new CourseType(); courseType.setId(Integer.parseInt(id)); if(name != null && name != "") {

courseType.setName(name);

}

if(status != null && status != "") { courseType.setStatus(Integer.parseInt(status));

}

if(url != null && url != "") { courseType.setUrl(url);

}

Integer result = courseTypeService.updateCourseType(courseType); if(result == 1) {

return new RespEntity(RespCode.SUCCESS, "修改成功");

}else {

return new RespEntity(RespCode.ERROR, "修改失败");

}

}

66

###### 统计分析模块

（1） 获取统计分析数据：获取表单数据，前台通过其接口来分析数据并渲染。

@GetMapping("/getAnalysis")

public RespEntity getConsultantList(){

List<Admin> consultantList = analysisService.getConsultantList(); if (consultantList != null){

return new RespEntity(RespCode.SUCCESS, consultantList);

}

return new RespEntity(RespCode.ERROR,"获取失败");

}

### 代码优化分析

###### 为了保障系统用户的信息安全，系统对用户的登陆密码采用了MD5 加密技术。字段加密代码下：



用户信息加密效果，如图 7-1 所示：

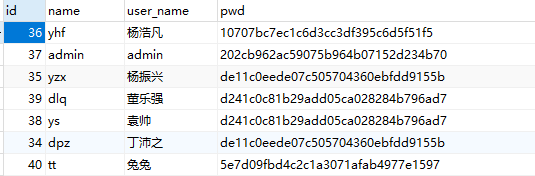


图 7-1 用户信息加密效果图

###### 由于系统采用前后端分离方式，为避免浏览器同源策略，采用 Token 作为身份令牌，服务端在登陆时产生 Token，并将 Token 返回给前端，前端需在每次请求前加上 Token 头以验证身份的有效性，并设置Token 有效时间。Token 生成使用 JWT 插件，代码如下：



Token 失效效果，如图 7-2 所示：



图 7-2 Token 失效效果图

67

# 测试

### 测试方案设计

###### 测试策略

1. 边开发边测试：在开发过程中，对系统功能进行测试调节。

###### 黑盒测试：根据系统的功能需求，对系统功能的完成度进行测试。

* + 1. 测试进度安排

###### 测试进度安排，如表 8-1 所示：

表 8-1 测试进度安排表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试阶段 | 任务目标 | 预计时间 | 测试人员 | 起止时间 |
| 第一阶段  准备阶段 | 编写测试用例 | 2 日 |  | 2020.04.10-2020.04.12 |
| 第二阶段  功能测试 | 对系统的各个功能模  块进行测试 | 3 日 |  | 2020.04.12-2020.04.15 |
| 第三阶段  兼容测试 | 测试系统在各种游览  器上的运行情况 | 1 日 |  | 2020.04.15-2020.04.17 |

###### 测试资源

测试资源，如表 8-2 所示：

表 8-2 测试资源表

|  |  |
| --- | --- |
| 操作系统 | Windows 10 |
| 浏览器 | 谷歌、火狐、IE 等主流浏览器 |
| 数据库 | MySQL |
| 服务器 | apache-tomcat 9.0 |
| Java 开发包 | jdk1.8 |
| CPU | Intel Core i5 处理器 |
| 显卡 | 2G 核心显卡 |
| 内存 | 8G |

68

###### 关键测试点

1. 登陆功能测试。

###### 修改密码功能测试。

1. 表单查询功能测试。

###### 网上报名功能测试。

1. 就业信息修改功能测试。

###### 新增用户功能测试。

1. 教资信息修改功能测试。
2. 显示在各种游览器系统是否正常。

### 测试用例构建

###### 测试用例编写约定

1. 保证测试用到的数据真实有效。

###### 确保系统最终实现的功能与系统功能需求吻合。

1. 完善系统在不同游览器上的重复性测试。

###### 增强本系统的可信任度。

* + 1. 测试用例设计

###### 测试用例设计，如表 8-3 所示：

表 8-3 测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 内容和目的 | 测试结果 |
| 1 | 用户登陆 | 输入正确的账号密码 | 登陆成功 |
| 账号密码不正确 | 提示错误 |
| 2 | 用户修改密码 | 输入原密码正确，确认新密码 | 修改成功 |
| 输入原密码不正确 | 修改失败 |
| 3 | 退出登陆 | 点击退出 | 返回登陆页面 |
| 4 | 表单查询 | 输入关键词 | 查询成功 |
| 选择关键词 | 查询成功 |
| 5 | 统计分析 | 点击汇总数据 | 查询成功 |
| 6 | 手录表单信息 | 各种信息符合格式 | 添加成功 |
| 格式不符 | 添加失败 |

69

续表 8-3

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 内容和目的 | 测试结果 |
| 7 | 新增用户 | 输入正确信息 | 新增成功 |
| 输入错误信息 | 新增失败 |
| 8 | 处理表单信息 | 点击处理，修改状态 | 处理成功 |
| 9 | 网上报名 | 填写姓名、手机号、专业 | 报名成功 |
| 手机号重复 | 报名失败 |
| 专业不存在 | 报名失败 |
| 10 | 新增教资信息 | 输入正确教资信息 | 操作成功 |
| 输入错误信息 | 数据库异常 |
| 11 | 新增合作单位 | 输入正确合作单位信息 | 操作成功 |
| 输入错误信息 | 数据库异常 |

###### 关键测试用例

1. 用户登陆测试

###### 内容：用户输入账号、密码、验证码，是否能成功登陆到系统主页面操作：点击登陆按钮

结果：测试成功

###### 提示登陆成功用例，如图 8-1 所示：



图 8-1 提示登陆成功用例图

###### 登陆成功后个人信息界面显示用例，如图 8-2 所示：

图 8-2 个人信息显示成功用例图

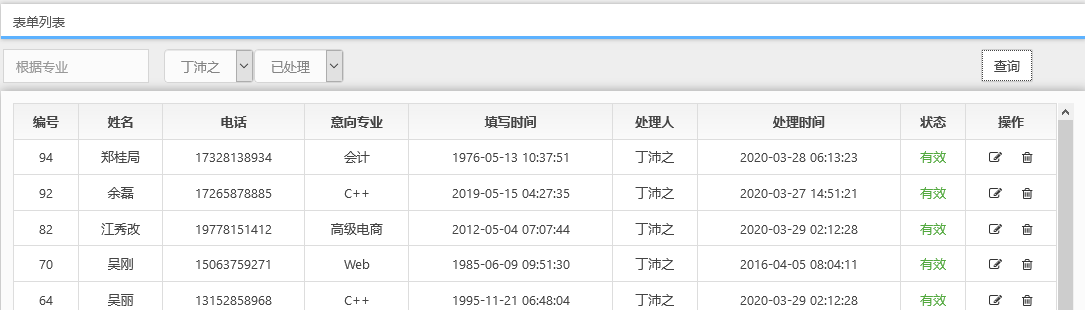
70

###### 表单查询测试

内容：管理员对表单信息的查询操作

###### 操作：输入内容、选择状态、选择处理人，满足其中一个或多个便可查询结果：测试成功

提示查询成功用例，如图 8-3 所示：



###### 网上报名测试

图 8-3 查询成功用例图

###### 内容：游客进行输入正确的信息，点击网上报名，后台新增此条报名信息操作：输入正确报名信息后，点击网上报名按钮

结果：测试成功

###### 提示报名成功用例，如图 8-4 所示：



图 8-4 提示报名成功用例图

###### 报名信息记录查询成功用例如图 8-5 所示：



1. 新增用户测试

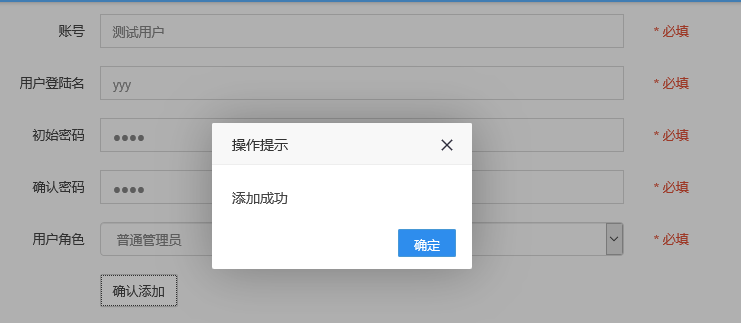
图 8-5 报名信息查询成功用例图

###### 内容：新增用户用以登陆后台系统

操作：超级管理员给用户新增账号，密码以及各种信息结果：测试成功

###### 新增用户成功测试用例，如图 8-6 所示：

71



###### 前台展示测试

图 8-6 新增用户成功测试用例图

###### 内容：前台展示各种信息

操作：修改信息状态，使得前台页面不展示。结果：测试成功

###### 修改状态前效果，如图 8-7 所示：



图 8-7 修改状态前效果图

72

###### 修改状态前效果，如图 8-8 所示：



图 8-8 修改状态后效果图

###### 测试用例维护

1. 根据系统新增的需求，添加新的测试用例，如加入搜索全局信息需求，需增加一项新的功能，故必须要进行测试；

###### 删除不需要的表单功能，存在某些信息系统会自动生成而不需要手动添加，故将手动添加信息功能删除；

1. 当登陆功能需要验证其Token 有效期，需要对此Token 进行记录时间；

###### 当使用者反应系统存在某些问题，应及时修正不完善的用例；

1. 增加、修改、删除各种管理信息日常维护用例。

73

# 总结与展望

### 设计工作总结

###### 在设计系统前，对此网站做了大量的需求分析，通过查阅资料以及老师的指导确定了网站应该如何开始设计。网站页面结构的展现以及网站颜色与文字的搭配都是网站建设的一个比较重要的方面，故此系统借鉴了多个教育信息门户网站的设计页面，混合搭配而成。

在系统进行需求分析的时候，没有深入的去了解各个用户应该具有哪些功能，导致需求分析出现遗漏。经过指导后，弥补了之前未分析到的需求，除此之外，还分析了一个隐形的需求，比如说后台用户需要对表单系统的数据进行分析。在总体设计方面，确定了网站系统采用Spring Boot 以及Vue 框架进行搭建，数据信息采用MySQL 数据库进行存储， 能够清晰地展现网站后台系统的功能模块。此系统主要包括表单管理，用户管理等模块功能。进行数据库设计时，确定了各个实体之间的关系，确定了各个实体的各个属性，并将它们用图或表的形式展现出来。

###### 在详细设计阶段，主要是对网站后台系统进行模块化的设计与分析，将后台系统中某个模块具体实现哪些功能，实现功能流程等等以 UML 建模方式说明。详细设计阶段在需求分析与编码阶段之间相当于一个交互的作用。

在编码阶段，系统刚开始进行搭建时需用到很多工具包，比如 MD5。实现的第一个模块是用户模块，这个模块是需要花大量的时间去实现的，因为此模块关系到网站前后台的交互，不仅仅是用户对后台的使用。当然，用户的安全也是极为重要的，因此，在登陆功能上需要用到动态验证码的检验以及Token 验证才能进行登陆。对于其他模块，由于采用模块化开发，功能的实现都大致相同，功能包括对信息的修改，删除信息等等。对于统计分析模块，是将表单系统中咨询师处理的信息进行分析，此功能大部分是前端来进行实现。

###### 系统完成之后，收尾便是进行测试阶段了。测试的时候，发现页面上有些图片未显示， 原因是路径不对，于是修改了系统的默认路径，使得图片可以完全的显示出来。还发现了用户的修改密码有问题，会连带其他用户的密码一起修改，发现问题是后台传值出现问题， 修改后可以正确的修改登陆用户的密码。当然测试中还有一些小问题，都已修改完成，不需要进行详细的说明，比如说多余的弹窗。

总之，系统设计阶段做了明确详尽的系统规划，划分系统模块，确定功能符合需求调研记录；编码阶段体现了 Spring Boot 以及Vue 框架的应用与实现。网站的颜色搭配以及字体的选择还有待处理，才可以使网站版面更加美观以及清新。此网站需要不断的去更新信息，从而使网站能够更加的活跃。

74

### 未来工作展望

###### 在网站首页方面，可以加入在线视频播放模块以及论坛模块，在线播放视频能更好的传播网站的有效信息，论坛可以增加游客访问此网站后对各类信息的一些讨论以及发表意见，论坛同咨询信息一样需要经常进行维护。若要加入这两类模块，需要使用更加快捷的技术去实现，才能使维护信息变得更加方便与准确。“”IT 教育信息门户网站的设计与实现涉及多方面的理论、方法和技术，本系统还有许多新的问题需要解决，需要在实际应用中不断积累和完善。当前，未对在线咨询等拓展功能进行更为深入的研究，使得咨询信息管理的接口不够完善，往后需要对其进行完善。

在开始编码阶段，使用的是单页面开发应用技术，由于先开发了后台部分，再去开发前台页面的话，编码文件会显得很混乱，加大了开发难度，于是将系统分为了两个项目， 前台项目与后台项目，当然在实际运用中不会有太大问题，未来若要重新建设此系统，可将两个项目放回一起，使用的访问链接为同一个，此系统就整合到一起了。

###### 相关的 web 技术在该系统的开发过程中应用范围还不够，还有提高的余地。比如各个栏目或页面上的特效，若加入一些特效，会将网站变得更加活跃，更加的有氛围。最好是能直接引入 JS 文件，直接改变某些元素的特效。

在咨询栏目中，未来在实际应用中需要对其进行激活服务，就可以进行人工回复，当然，这需要一定的资金。对于网上报名栏目，需要新增更加便携的回访功能，以便能更快的通知以及回访客户。登陆界面还需新增一个忘记密码链接，通过手机验证方式来找回密码。对于后台的首页，应该加入一个表格来直接展示咨询师处理的信息而不需要进入表单系统管理来进行查看。

###### 总的来说，系统还有很大的完善空间，未来需要使用更多的技术来进行实现。

75

# 谢辞

76

# 参考文献

* + 1. 高云艳.基于 Java EE 的校园门户网站设计与实现[J].建材技术与应用,2018(05):46-48.
    2. 黄安.基于 PHP+Mysql 技术的网站设计与实现——以美食网站系统的设计为例[J].轻纺工业与技术,2019,48(07):168-170.
    3. 李杰,陈惠敏.基于 SSM 的在线学习平台设计与实现[J].电脑编程技巧与维护,2019(11):53-54+66. [4]李鹏楼.高性能门户网站及其信息发布系统的设计与实现[D].北京邮电大学,2019.

1. 杨嘉诚, 柯海丰. 基于 HTML5 和 JavaScript 的信息学学习网站的设计与实现[J]. 计算机时代,2019(11):32-34+37.
2. 刘艳.基于 UML 的在线投稿系统的设计与实现[J].电子技术与软件工程,2018(21):47-49.
3. 于敞.基于微服务架构的接口自动化测试服务与测试报告的设计与实现[D].内蒙古大学,2019. [8]朱建华.基于 Spring 的 Web 自动化测试平台设计与实现[D].大连理工大学,2019.
4. 胡畔.软件测试信息管理系统的设计与实现[D].电子科技大学,2018.
5. 吕宇琛.SpringBoot 框架在web 应用开发中的探讨[J].科技创新导报,2018,15(08):168+173.
6. 肖文娟,王加胜.基于 Vue 和 Spring Boot 的校园记录管理 Web App 的设计与实现[J].计算机应用与软件,2020,37(04):25-30+88.
7. 颜治平. 基于 SpringBoot 和 Vue 框架的教代会提案系统的设计与实现[J]. 科技创新与应用,2020(03):91-93+95.
8. 陈 倩 怡 , 何军 .Vue+Springboot+MyBatis 技 术 应 用 解 析 [J]. 电 脑 编 程 技 巧 与 维护,2020(01):14-15+28.
9. 夏朋举.网站设计及网页制作技巧[J].电脑编程技巧与维护,2020(01):171-173.
10. 熊永平.基于 SpringBoot 框架应用开发技术的分析与研究[J].电脑知识与技术,2019,15(36):76-77. [16]张振球.基于 SpringBoot 的校园文章发布系统设计与实现[J].无线互联科技,2019,16(23):41-43. [17]罗路腾,王贵鑫.基于 Springboot 的博客网站的设计与实现[J].科学技术创新,2019(33):64-66.
11. 王巍,王倩,汤二子.信息时代中国企业应该建设网站吗？[J].生产力研究,2019(11):78-82.
12. 朱运乔. 基于 SpringBoot+SSM 框架的 Web 应用系统搭建与实现[J]. 电脑编程技巧与维护,2019(10):23-25.
13. 王悦,张雷.基于 SpringBoot 技术的高职信息技术核心课教学改革研究[J].湖南邮电职业技术学院学报,2019,18(02):27-29.
14. 贺紫珺. 基于 SpringBoot 和 Vue 框架的第三方医疗器械供应链平台的设计与实现[D].东华大学,2019.
15. 何军 , 陈 倩 怡 .Vue+Springboot+Mybatis 开 发 消 费 管 理 系 统 [J]. 电 脑 编 程 技 巧 与 维

护,2019(02):87-88+102.

77

1. Shufan Liu. Design and Implementation of Communication Base Station Survey System based on Springboot[C]. 香港新世纪文化出版社（Hongkong New Century Cultural Publishing House）.2018 年智慧教育与人工智能发展学术会议论文集（第一部分）.香港新世纪文化出版社（Hongkong New Century Cultural Publishing House）:香港新世纪文化出版社有限公司,2018:34-36.
2. 焦鹏珲. 基于 SpringBoot 和 Vue 框架的电子招投标系统的设计与实现[D].南京大学,2018. [25]Engr. Oboshior Joel(MNSE),Samuel Egwunatun (MNIQS),Prof. (ARC.) J. E. Ahianba. Design and

Implementation of Student Grade Analysis System Based on Spring Boot Microservice Framework[J]. MA Healthcare Ltd.,2014,1(4).

1. Moussa Leblouba,Sami W.Tabsh,Samer Barakat.Reliability-based design of corrugated web steel girders in shear as per AASHTO LRFD[J].Journal of Constructional Steel Research,2020,169.
2. Soumaya Rebai,Marouane Kessentini,Hanzhang Wang,Bruce Maxim.Web service design defects detection:A bi-level multi-objective approach[J].Information and Software Technology,2020,121.

[28]M.Yamamoto.Non-parametric optimization of railway wheel web shape based on fatigue design criteria[J].International Journal of Fatigue,2020,134.

1. Kan Ji. Design and Implementation of Teaching Quality Evaluation System Based on SpringBoot[C]. 西南石油大学（Southwest Petroleum University）.第七届计算与信息科学国际学术会议论文集.西南石油大学（Southwest Petroleum University）:西南石油大学计算机科学学院,2019:444-452.
2. Shufan Liu. Design and Implementation of Communication Base Station Survey System based on Springboot[C]. 香港新世纪文化出版社（Hongkong New Century Cultural Publishing House）.2018 年智慧教育与人工智能发展学术会议论文集（第一部分）.香港新世纪文化出版社（Hongkong New Century Cultural Publishing House）:香港新世纪文化出版社有限公司,2018:34-36.
3. 马昭凤,黄萍,文智丽,尹子重,詹飞.旅游政务网站可视化信息类型与公众接受偏好分析[J].旅游纵览(下半月),2020(03):23-24+27.

78

## 附录A 外文翻译—原文部分

外文原文出处：Praveen.Gupta et al. / International Journal of Engineering Science and Technology Vol.

2(6), 2010, 1684-1689

##### Spring Web MVC Framework for Rapid Open Source J2EE Application Development: A Case Study

Abstract— Today it is the highly competitive for the development of Web application, it is the need of the

time to develop the application accurately, economically, and efficiently. We are interested to increase productivity and decrease complexity. This has been an underlying theme in a movement to change the way programmers approach developing Java 2 Platform, Enterprise Edition (J2EE) Web applications. Our focus is how to create J2EE-compliant software without using Enterprise Java Beans (EJB). The one of the best alternative is the Spring framework, which provides less services but it is much less intrusive than EJB. The driving force behind this shift is the need for greater productivity and reduced complexity in the area of Web application software development and implementation. In this paper, we briefly describe spring underlying architecture and present a case study using Spring web MVC Framework.

***Index Terma:* MVC, Spring, XML**

**I** INTRODUCTION

Web is the very complex issues these days. Since the desire of the companies and organizations are increasing so the complexity and the performance of the web programming matters. Complexity with the different types of communication devices is increasing. The business is demanding applications using the web and many communication devices. So with the increase load of the data on the internet we have to take care of the architecture issue. Let us discuss how it works fast using spring web mvc framework the rapid application development while maintaining the Model View Architecture of the application.

Spring frameworks comes with rich set of features, let us discuss these features in brief.

1. Inversion Of Control: Inversion of Control or IoC is one of the techniques used to wire services or components to an application program. The IoC is “A software design pattern and set of associated programming techniques in which the flow of control of a system is inverted in comparison to the traditional interaction mode.” In IoC instead of an application calling the framework, it is the framework that calls the components specified by the application. The IoC can be explained as "Injection of required resources or dependency at run-time into the dependent resource" which is also known as Dependency Injection. The org.springframework.beans.factory.BeanFactory is the actual representation of the Spring IoC container which is responsible for containing and managing the beans. The BeanFactory interface is the central IoC container interface in Spring. A bean is simply an object that is instantiated and managed by a Spring IoC container. These beans and the dependencies between them are reflected in the configuration metadata used by a container.
2. Constructor Dependency Injection: we can use the java class constructor to load the bean values. A java Class is defined with a constructor of single field. Details.xml file provides the value to be passes to

79

the constructor. Now another java loads the xml file using the BeanFactory Method. This uses the xml file to load values in the constructor of the java file. This is used to pass values to the constructor.

1. Setter Dependency Injection: With every bean we defined the getters and setters. We can also use setters method to set the values in the beans. setters method overrides the values loaded from the beans.
2. Interface: we can define the interface class in spring. To implement this we will import interface to the java program. Now we can use methods defined in interface using spring and xml.
3. Inheritance: One java class can acquired the properties of another class just like a java program. There are three sub types of it. 1. Abstract: Beans declared abstract cannot be inherited in the springs. 2. Parent Child: we can define hierarchy like parent child. 3. Parent - Child - Sub Child Relationship: in this we can define hierarchy for 3 or more classes.
4. Autowiring: Autowiring is used to map the property name, values in xml file with java file. There are four types to integrate it. byName, byType, constructor, autodetect. If nothing is defined about it then byName is the default.
5. Scope of Beans: All beans defined in spring are having scope of four values prototype, session, request, singleton, global-session. This is used to control the access of the beans.
6. Reference Beans: One bean in the xml file can be assigned values from the other bean. This is used to read values from one bean and assign to another bean.

**II** MAJOR SPRING COMPONENTS

In the spring we also follow the principals of the MVC. It has been designed more for the desktop and internet based applications. Spring consist of three core collaborating components.

1. Controller: Handles navigation logic and interacts with the Service tier for business logic
2. Model: The contract between the Controller and the View Contains the data needed to render the View Populated by the Controller
3. View: Renders the response to the request Pulls data from the model.

Core components in the spring MVC are as follows.

1. DispatcherServlet: this is the spring’s front controller implementation. Web.xml receives the request and transfer it to the DispatchServlet. This is the first controller which interacts to the requests. It is also known as implementation of the Servlet. It controls the complete flow of the application and navigates the flow of application.
2. Controller: this is the user created component for handling requests. It encapsulates the navigation logic with it. Controller delegates the services for the service object.
3. View: view is responsible for rendering output. Different views can be selected for the different types of output bases on the results and the viewing device, communication devices.
4. ModelAndView: ModelAndView is the core part of the spring framework. It implements the business logic of the application. It is created by the controller. It associates the view to the request. It stores the

80

business logic and Model data. A controller calls it and it will execute. On execution it will return the data and name ofview.

1. ViewResolver: How the output is to be displayed depends on the result received from ModelAndView. It is used to map logical view names to actual view implementations. This part identifies and implement what is the output media and how to display it.
2. HandlerMapping: Strategy interface used by DispatcherServlet for mapping incoming requests to individual Controllers. It identifies the request and calls the respective handler to provide the services. Handler will call to controller.

**III** SPRING ARCHITECTURE

The Spring framework provides a full-featured MVC module for building Web applications. with spring’s pluggable MVC architecture. It is configurable with multiple view technologies Ex Java Server Pages, Velocity, Tiles, iText etc. Spring MVC separates the roles of the controller, model object, dispatcher Servlet and the handler object. Clear separation of objects and controllers makes them easier to customize. The figure 01 shows the view of the execution flow.

Delegate

**Front**

**Contr**

Return

**model**

Delegate

response

**model**

rendering

Render

**View template**

The following figure 02 shows the sequence diagram of the spring model. In this the dispatcher Servlet is the entry point for the application. As soon the Dispatch Servlet get the request for the services and it will decides the handler. All handlers are mapped with the Servlet. Handler will come in action and will call the respective controller and the pass the request parameters to it.

Now controller comes in action, it contains business logic and a ModelAndView is associated with the controller. on execution it will return the ModelAndView to the Dispatch Servlet. This ModelAndView contains the data and view name.

Dispatcher Servlet gets the ModelAndView from the controller. It contains the data and view name. Servlet will call the view resolver. View resolver will identify the name of the view through which data is to be presented. Finally it will present the data to the respective and appropriate format to the user.

81



View

Reso

getHandler

Controller

handleRequest()

ModelAndView

resolveViewName

View

render()

Contro ller

Handler Mappin

Dispatch er Servlet

Figure 2: Sequence flow of application in the spring framework.

**IV** SPRING AND XML

Xml is widely used in the spring framework. It simplify the development process and saves time. xml is used to store the data, which is used during the execution of application.

web.xml is the entry point in the application. It will tell you the further path of navigation. It loads the application context class and the tells the name of the dispatcher Servlet xml file.

**web.xml:-** <?xml version="1.0" encoding="UTF-8"?>

<web-app version="1.0" xmlns="http//java.sun.com/xml/ns/javaee" xmlns:xs[i="h](http://www.w3.org/2001/XMLSchema-instance)ttp[://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:schem[alLocation="htt](http://java.sun.com/xml/ns/javaee)p://[java.sun.](http://java.sun.com/xml/ns/javaee)com/xm[l/ns/javaee](http://java.sun.com/xml/ns/javaee)

[http://java.sun.com/xml/ns/javaee/web-app-2\_5.xsd">](http://java.sun.com/xml/ns/javaee/web-app-2_5.xsd)

<listener><listener-Class>org.springframework.web.context.ContextLoaderListener</listener-Class></listener

>

<servlet>

<servlet-name>dispatcher</servlet-name><servlet- class>org.springframework.web.servlet.DispatcherServlet</servlet- class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/send/\*</url-pattern>

82

</servlet-mapping>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list></web-app>

The above web.xml .xml will interact with the incoming request on the server. This file defines the name of the Servlet which is dispatcher, ApplicationContext and the index.jsp as welcome page. url pattern defined as \*.\* means it will all types of incoming request.

**ApplicationContext.xml:-** The ApplicationContext is build on top of the BeanFactory. It provides an easy integration with Springs AOP features, message resource handling, event propagation. The BeanFactory provides the configuration framework and basic functionality. ApplicationContext adds enhanced capabilities to application. While building applications in a J2EE-environment ApplicationContext must be used.

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/beans"](http://www.springframework.org/schema/beans) xmlns:xs[i="h](http://www.w3.org/2001/XMLSchema-instance)ttp[://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xmlns:p=["http://www.springfra](http://www.springframework.org/schema/p)m[ework.org/schema/p](http://www.springframework.org/schema/p)" xmlns:[aop="ht](http://www.springframework.org/schema/aop)tp:/[/www.springframework.org/schema/aop"](http://www.springframework.org/schema/aop) xm[lns:tx="http://www.s](http://www.springframework.org/schema/tx)pringfr[amework.org/schema/tx"](http://www.springframework.org/schema/tx) xsi:schemaLocation=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/](http://www.springframework.org/schema/beans) [beans](http://www.springframework.org/schema/beans)

[http://www.springframework.org/schema/beans/spring-beans-2.](http://www.springframework.org/schema/beans/spring-beans-2.5.xsd) [5.xsd](http://www.springframework.org/schema/beans/spring-beans-2.5.xsd) <http://www.springframework.org/schema/aop> [http://www.springframework.org/schema/aop/spring-beans-2.5.x](http://www.springframework.org/schema/aop/spring-beans-2.5.xsd) [sd](http://www.springframework.org/schema/aop/spring-beans-2.5.xsd) <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx-2.5.xsd>">

<bean id="superClass" class="packagename.SuperClass" />

<bean id ="subClass" class=" packagename.SubClass"></bean>

<property name="superClass" ref="superClass"/>

</beans>

The above file loads the bean of the SuperClass.java and SubClass.java. It also defines the reference class. Dispatcher-servlet.xml:-

Spring's web MVC framework is a request driven web MVC framework, it is designed around a servlet that dispatches requests to controllers and provides much functionality for handling the applications. DispatcherServlet is completely integrated with the Spring ApplicationContext and allows you to use feature of springs.

83

Dispatcher Servlet is the central controlling unit for the working of the application. It is used to define the view resolver, beans, handlers and their mapping of the application.

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/beans"](http://www.springframework.org/schema/beans)xmlns: xs[i="h](http://www.w3.org/2001/XMLSchema-instance)ttp[://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xmlns:p=["http://www.springfra](http://www.springframework.org/schema/p)m[ework.org/schema/p](http://www.springframework.org/schema/p)" xmlns:[aop="ht](http://www.springframework.org/schema/aop)tp:/[/www.springframework.org/schema/aop"](http://www.springframework.org/schema/aop) xm[lns:tx="http://www.s](http://www.springframework.org/schema/tx)pringfr[amework.org/schema/tx"](http://www.springframework.org/schema/tx) xsi:schemaLocation=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/](http://www.springframework.org/schema/beans) [beans](http://www.springframework.org/schema/beans)

[http://www.springframework.org/schema/beans/spring-beans-2.](http://www.springframework.org/schema/beans/spring-beans-2.5.xsd) [5.xsd](http://www.springframework.org/schema/beans/spring-beans-2.5.xsd) <http://www.springframework.org/schema/aop> [http://www.springframework.org/schema/aop/spring-beans-2.5.x](http://www.springframework.org/schema/aop/spring-beans-2.5.xsd) [sd](http://www.springframework.org/schema/aop/spring-beans-2.5.xsd) <http://www.springframework.org/schema/tx> <http://www.springframework.org/schema/tx/spring-tx-2.5.xsd>">

<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix">

<value>/WEB-INF/views/</value></property>

<property name="suffix"><value>.jsp</value></property>

</bean>

<bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">

<property name="mappings">

<props><prop key ="/\*">dispatchController</prop></props>

</property></bean>

<bean id="dispatchController" class=" packagename.DispatchController"></bean></beans>

**V** HOW IT WORKSRAD

Rapid Application Development is the requirement of the industry since a long time. There are many development tools which have helped it. But gradually technology and the requirements goes on increasing so e need the tools and the in fact architecture which can handle the growing size of the application. Tools are different from the architecture. Spring is the architecture in MVC which can support the large applications. In this technology and architecture once implemented it is easy to inheritance the application without touching the existing code. It’s use of the xml files helps us to add the new mappings, requests, java beans etc to the application.

84

**VI** ARCHITECTURAL BENEFIT

let's look at some of the Architectural benefits spring web MVC Framework can bring to a project.

* + Spring effectively organize your middle tier objects, EJB doesn't affect it. The configuration management services can be used in any architectural layer and in any runtime environment.
  + The Spring Web MVC Framework is a robust, flexible, and well-designed framework for rapidly developing web applications using the MVC design pattern.
  + Spring eliminate the proliferation of Singletons. This is a major problem, reducing testability and object orientation.
  + Clear separation of roles: Spring MVC nicely separates the roles played by the various components that make up this web framework. All components like controllers, command objects, and valuators’ each component plays a distinct role.
  + Adaptable controllers: If your application does not require an HTML form, you can write a simpler version of a Spring controller that does need all the extra components required for form controllers. Spring provides several types of controllers, each serving a different purpose.
  + Spring eliminate the need to use a variety of custom properties file formats, by handling configuration in a consistent way throughout applications and projects.
  + Spring provides good programming practice by reducing the cost of programming to interfaces, rather thanclasses.
  + Applications built with it depend on as few of its APIs. Most business objects in spring applications have no dependency on spring.
  + Applications built using spring are very easy to unit test.
  + Spring can make the use of EJB an implementation choice, rather than the determinant of application architecture.
  + You can choose to implement business interfaces as POJOs or local EJBs without affecting calling code.
  + Spring provide an alternative to EJB that's appropriate for many applications. It can use AOP to deliver declarative transaction management without using an EJB container.
  + Spring provides a consistent framework for data access, whether using JDBC or an O/R mapping, Hibernate. It provides a consistent and simple programming model in areas like JDBC, JMS, JavaMail, JNDI and many APIs which makes it an ideal architectural.
  + This is the framework which builds applications using POJOs. It also conceals complexity from the developer.
  + While using JDBC it solve the problem of connection leak, we need to write only necessary SQL, it also solves the problems of error returned from database.

85

## 附录B 外文翻译—译文部分

##### 用于快速开源 J2EE 应用程序开发的

Spring Web MVC 框架：一个案例研究

摘要—如今，对于 Web 应用程序的开发来说，它具有很高的竞争力，需要时间来准确，经济，高效地开发应用程序。我们有兴趣提高生产率并降低复杂性。这是改变程序员开发 Java 2 平台企业版

（J2EE）Web 应用程序的方式的一个基本主题。我们的重点是如何在不使用企业 Java Bean（EJB）的情况下创建符合 J2EE 的软件。最好的替代方法之一是 Spring 框架，该框架提供较少的服务，但与 EJB 相比，侵入性小得多。这种转变背后的驱动力是在 Web 应用程序软件开发和实现方面需要更高的生产率和降低的复杂性。在本文中，我们简要描述了 Spring 底层体系结构，并提出了一个使用 Spring Web MVC Framework 的案例研究。

索引术语：MVC，Spring，XML

I 引言

如今，Web 是非常复杂的问题。由于公司和组织的需求在增加，因此 Web 编程的复杂性和性能至关重要。不同类型的通信设备的复杂性正在增加。业务要求使用 Web 和许多通信设备的应用程序。因此，随着 Internet 上数据负载的增加，我们必须解决体系结构问题。让我们讨论一下如何使用 Spring Web MVC 框架快速进行工作，以快速进行应用程序开发，同时维护应用程序的模型视图体系结构。

Spring 框架带有丰富的功能集，让我们简要地讨论这些功能。

1. 控制反转：控制反转或 IoC 是用于将服务或组件连接到服务器的技术之一。应用程序。 IoC 是“一种软件设计模式和一组关联的编程技术，其中，与传统的交互模式相比，系统的控制流程是相反的。” 在 IoC 中，不是框架调用应用程序，而是框架调用应用程序指定的组件。 IoC 可以解释为“在运 行 时 将 所 需 资 源 或 依 赖 项 注 入 到 依 赖 资 源 中 ” ， 也 称 为 依 赖 项 注 入 。org.springframework.beans.factory.BeanFactory 是 Spring IoC 容器的实际表示形式，该容器负责包含和管理 Bean。 BeanFactory 接口是 Spring 中的中央 IoC 容器接口。 Bean 只是一个由 Spring IoC 容器实例化和管理的对象。这些 Bean 及其之间的依赖关系反映在容器使用的配置元数据中
2. Constructor Dependency Injection：我们可以使用 java 类的构造函数来加载bean 的值。 Java 类是使用单个字段的构造函数定义的。 Details.xml 文件提供了要传递给构造函数的值。现在，另一个 Java 使用 BeanFactory 方法加载 xml 文件。这使用 xml 文件将值加载到 java 文件的构造函数中。这用于将值传递给构造函数。
3. Setter 依赖注入：我们为每个 bean 定义了 getter 和 setter。我们还可以使用 setters 方法来设置

bean 中的值。 setters 方法将覆盖从 Bean 加载的值。

1. Interface：我们可以在 spring 中定义接口类。为了实现这一点，我们将导入 java 程序的接口。现在我们可以使用 spring 和 xml 在接口中定义的方法。
2. 继承：一个 Java 类可以像 Java 程序一样获取另一个类的属性。它分为三种子类型。 1.单一： 声明为摘要的 Bean 不能在 Spring 中继承。 2.多级：我们可以像父级子级一样定义层次结构。 3.多层： 在此我们可以为 3 个或更多类定义层次结构。
3. Autowiring：Autowiring 用于映射属性名称，xml 文件中的值与 java 文件。有四种类型可以集

86

成它。 byName，byType，构造函数，自动检测。如果未定义任何内容，则默认为byName。

1. Beans 的范围：在 spring 中定义的所有 bean 都具有四个值的范围：原型，会话，请求，单例， 全局会话。这用于控制 Bean 的访问。
2. Reference Beans：可以为 xml 文件中的一个bean 分配来自另一个bean 的值。这用于从一个 bean

读取值并分配给另一个 bean。

II spring 主要组件

在 spring 中，我们还将遵循 MVC 的原则。它针对桌面和基于 Internet 的应用程序进行了更多设计。

Spring 由三个核心协作组件组成。

1. 控制器：处理导航逻辑并与服务层进行业务逻辑交互
2. 模型：控制器和视图之间的协定包含呈现由控制器填充的视图所需的数据。
3. 视图：呈现对请求的响应从模型中提取数据。

Spring MVC 中的核心组件如下：

1. DispatcherServlet ：这是 Spring 的前端控制器实现。 Web.xml 接收请求并将其传输到DispatchServlet。这是第一个与请求进行交互的控制器。它也称为 Servlet 的实现。它控制应用程序的完整流程并导航应用程序的流程。
2. Controller：这是用户创建的用于处理请求的组件。它封装了导航逻辑。控制器将服务委托给服务对象。
3. View：视图负责渲染输出。基于结果和查看设备，通信设备，可以为不同类型的输出选择不同的视图。
4. ModelAndView：ModelAndView 是 spring 框架的核心部分。它实现了应用程序的业务逻辑。它由控制器创建。它将视图与请求关联。它存储业务逻辑和模型数据。控制器调用它，它将执行。在执行时，它将返回数据和视图名称。
5. ViewResolver：输出的显示方式取决于从 ModelAndView 收到的结果。它用于将逻辑视图名称映射到实际的视图实现。此部分标识并实现什么是输出媒体以及如何显示它。
6. HandlerMapping：DispatcherServlet 使用的策略接口，用于将传入请求映射到各个 Controller。它识别请求并调用相应的处理程序以提供服务。处理程序将调用控制器。

III SPRING 架构

Spring 框架提供了用于构建 Web 应用程序的功能齐全的 MVC 模块。带有 spring 的可插拔 MVC 架构。它可以使用多种视图技术进行配置，例如 Java Server Pages，Velocity，Tiles，iText 等。Spring MVC 分离了控制器，模型对象，调度程序 Servlet 和处理程序对象的角色。对象和控制器的清晰分隔使它们

更易于自定义。下图 01 显示了执行流程的视图。

Return

Delegate

**Front Contr**

**model**

response

**model**

Render

Delegate rendering

**View template** 87

下图 02 显示了弹簧模型的顺序图。在这种情况下，调度程序 Servlet 是应用程序的入口点。一旦Dispatch Servlet 收到了对服务的请求，它将决定处理程序。所有处理程序都与 Servlet 映射。处理程序将起作用，并将调用相应的控制器，并将请求参数传递给它。

现在控制器开始起作用，它包含业务逻辑，并且 ModelAndView 与控制器关联。在执行时，它将把

ModelAndView 返回给Dispatch Servlet。此 ModelAndView 包含数据和视图名称。



View

Reso

getHandler

Controller

handleRequest()

ModelAndView

resolveViewName

View

render()

Contro

ller

Handler Mappin

Dispatch er Servlet

图 2: spring 框架中应用程序的序列流程

分派器 Servlet 从控制器获取 ModelAndView。它包含数据和视图名称。 Servlet 将调用视图解析器。视图解析器将识别要通过其呈现数据的视图的名称。最后，它将以相应的适当格式将数据呈现给用户。

IV SPRING 和 XML

Xml 在 spring 框架中被广泛使用。 它简化了开发过程并节省了时间。 xml 用于存储数据，该数据在应用程序执行期间使用。

web.xml 是应用程序中的入口点。 它将告诉您导航的进一步路径。 它加载应用程序上下文类，并告诉调度程序 Servlet xml 文件的名称。

web.xml:- <?xml version="1.0" encoding="UTF-8"?>

<web-app version="1.0" xmlns="http//java.sun.com/xml/ns/javaee" xmlns:xs[i="h](http://www.w3.org/2001/XMLSchema-instance)ttp[://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:schem[alLocation="htt](http://java.sun.com/xml/ns/javaee)p://[java.sun.](http://java.sun.com/xml/ns/javaee)com/xm[l/ns/javaee](http://java.sun.com/xml/ns/javaee) [http://java.sun.com/xml/ns/javaee/web-app-2\_5.xsd">](http://java.sun.com/xml/ns/javaee/web-app-2_5.xsd)

<listener>

<listener-Class>org.springframework.web.context.ContextLoaderListener</listener-Class></listener>

<servlet>

<servlet-name>dispatcher</servlet-name><servlet- class>org.springframework.web.servlet.DispatcherServlet

88

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/send/\*</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

</web-app>

上面的 web.xml .xml 将与服务器上的传入请求进行交互。该文件定义了 Servlet 的名称（即调度程序），ApplicationContext 和 index.jsp 作为欢迎页面。定义为\*。\*的 url 模式意味着它将处理所有类型的传入请求。 ApplicationContext.xml：-ApplicationContext 构建在 BeanFactory 之上。它提供了与 Springs AOP 功能， 消息资源处理， 事件传播的轻松集成。 BeanFactory 提供配置框架和基本功能。ApplicationContext 为应用程序添加了增强的功能。

在 J2EE 环境中构建应用程序时，必须使用 ApplicationContext。

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/beans"](http://www.springframework.org/schema/beans) xmlns:xs[i="h](http://www.w3.org/2001/XMLSchema-instance)ttp[://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xmlns:p=["http://www.springfra](http://www.springframework.org/schema/p)m[ework.org/schema/p](http://www.springframework.org/schema/p)" xmlns:[aop="ht](http://www.springframework.org/schema/aop)tp:/[/www.springframework.org/schema/aop"](http://www.springframework.org/schema/aop) xm[lns:tx="http://www.s](http://www.springframework.org/schema/tx)pringfr[amework.org/schema/tx"](http://www.springframework.org/schema/tx) xsi:schemaLocation=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/beans](http://www.springframework.org/schema/beans) <http://www.springframework.org/schema/beans/spring-beans-2.5.xsd> <http://www.springframework.org/schema/aop>

<http://www.springframework.org/schema/aop/spring-beans-2.5.xsd> <http://www.springframework.org/schema/tx><http://www.springframework.org/schema/tx/spring-tx-2.5.xsd>">

<bean id="superClass" class="packagename.SuperClass" />

<bean id ="subClass" class=" packagename.SubClass"></bean>

<property name="superClass" ref="superClass"/>

</beans>

上面的文件加载了 SuperClass.java 和 SubClass.java 的bean。

89

它还定义了引用类。Dispatcher-servlet.xml：- Spring 的 Web MVC 框架是一个请求驱动的 Web MVC 框架，它围绕一个 Servlet 设计，该 Servlet 将请求分发给控制器，并提供了许多用于处理应用程序的功能。 DispatcherServlet 与 Spring ApplicationContext 完全集成，并允许您使用 spring 的功能。

Dispatcher Servlet 是应用程序工作的中央控制单元。 它用于定义视图解析器，Bean，处理程序及其对应用程序的映射。

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/beans"](http://www.springframework.org/schema/beans)xmlns:xs[i="h](http://www.w3.org/2001/XMLSchema-instance)ttp[://www.w3.org/2001/XMLSchema-in](http://www.w3.org/2001/XMLSchema-instance) [stance"](http://www.w3.org/2001/XMLSchema-instance) xmlns:p=["http://www.springfra](http://www.springframework.org/schema/p)m[ework.org/schema/p](http://www.springframework.org/schema/p)" xmlns:[aop="ht](http://www.springframework.org/schema/aop)tp:/[/www.springframework.org/schema/aop"](http://www.springframework.org/schema/aop) xm[lns:tx="http://www.s](http://www.springframework.org/schema/tx)pringfr[amework.org/schema/tx"](http://www.springframework.org/schema/tx) xsi:schemaLocation=["http://www.spring](http://www.springframework.org/schema/beans)f[ramework.org/schema/beans](http://www.springframework.org/schema/beans) <http://www.springframework.org/schema/beans/spring-beans-2.5.xsd> <http://www.springframework.org/schema/aop>

<http://www.springframework.org/schema/aop/spring-beans-2.5.xsd> <http://www.springframework.org/schema/tx><http://www.springframework.org/schema/tx/spring-tx-2.5.xsd>">

<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix">

<value>/WEB-INF/views/</value></property>

<property name="suffix"><value>.jsp</value></property>

</bean>

<bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">

<property name="mappings">

<props><prop key ="/\*">dispatchController</prop>

</props>

</property></bean>

<bean id="dispatchController" class=" packagename.DispatchController"></bean></beans>

V RAD 如何运作

长期以来，快速开发应用程序是行业的要求。有许多开发工具对此有所帮助。但是随着技术和需求的逐渐增加，因此需要能够处理不断增长的应用程序的工具和实际上的体系结构。工具与体系结构不同。Spring 是 MVC 中的体系结构，可以支持大型应用程序。在实施该技术和体系结构后，无需触摸现有代码即可轻松继承应用程序。使用 xml 文件可以帮助我们将新的映射，请求，java bean 等添加到应用程序中。

VI 网络效益

让我们看一下 Spring Web MVC Framework 可以为项目带来的一些架构上的好处。

* Spring 可以有效地组织您的中间层对象，而 EJB 不会对其产生影响。可以在任何体系结构层和任何运行时环境中使用配置管理服务。

90

* Spring Web MVC Framework 是一个健壮，灵活且设计良好的框架，用于使用 MVC 设计模式快速开发 Web 应用程序。
* Spring 消除了单例的扩散。这是一个主要问题，降低了可测试性和面向对象。清晰的角色分离：Spring MVC 很好地分离了组成此 Web 框架的各个组件所扮演的角色。所有组件（如控制器，命令对象和评估器）均扮演着不同的角色。
* 自适应控制器：如果您的应用程序不需要HTML 表单，则可以编写 Spring 控制器的简单版本，该版本确实需要表单控制器所需的所有其他组件。 Spring 提供了几种类型的控制器，每种都有不同的用途。
* 通过在整个应用程序和项目中以一致的方式处理配置，Spring 消除了使用各种自定义属性文件格式的需求。Spring 通过减少接口（而不是类）的编程成本来提供良好的编程实践。
* 使用它构建的应用程序仅依赖于其很少的 API。 Spring 应用程序中的大多数业务对象都不依赖于

Spring。

* 使用 spring 构建的应用程序很容易进行单元测试。
* Spring 可以将EJB 的使用作为实现选择，而不是应用程序体系结构的决定因素。
* 您可以选择将业务接口实现为 POJO 或本地 EJB，而不会影响调用代码。
* Spring 提供了适用于许多应用程序的 EJB 替代方案。它可以使用AOP 来提供声明性事务管理，而无需使用 EJB 容器。
* Spring 提供了一个一致的数据访问框架，无论使用 JDBC 还是 O / R 映射 Hibernate。它在 JDBC， JMS，JavaMail，JNDI 和许多API 等领域提供了一致且简单的编程模型，这使其成为理想的体系结构。
* 这是使用 POJO 构建应用程序的框架。它还向开发人员隐藏了复杂性。
* 虽然使用 JDBC 解决了连接泄漏的问题，但我们只需要编写必要的 SQL，也解决了从数据库返回的错误的问题。

91

## 附录C 软件使用说明书

#### 一、系统运行环境

###### 操作系统：Win10

1. 编译器：IntelliJ IDEA 2019.3.1
2. 成功搭建JAVA 环境，Vue 脚手架、Mysql 数据库

#### 二、系统使用步骤

###### 网站后台使用步骤

1. 部署运行本项目，登陆后进入系统后台，后台如图C-1 所示：



图 C-1 系统首页图

###### 点击各种信息栏目，处理相关展示信息。

1. 点击搜索栏，输入关键字，查询信息。

###### 点击右上角用户按钮，退出后台或者修改密码。

1. 点击统计分析栏目，即可查看分析的数据信息。

###### 点击信息列表操作栏中的删除或修改按钮，可实现对信息的修改和删除操作。

92

###### 网站前台使用步骤

1. 在浏览器搜索框中输入localhost：8889，进入前台首页，如图C-2 所示：



图 C-2 系统前台首页图

###### 点击课程，查看网站课程信息内容。

1. 点击网上报名栏目，跳转到首页底部，填写报名信息。

###### 点击关于，点击合作单位，跳转到相应位置。

1. 点击在线咨询，进入在线咨询页面。

93

## 附录D 主要源代码

###### 用户mapper 层：

@Mapper

public interface AdminMapper{

//登陆

@Select("select id,name,pwd,state from admin where name=#{name}") public Admin login(String name);

//获取用户信息

@Select("select id,name,user\_name,creat\_time,last\_time,state,group\_id,last\_ip,on\_line,active from admin where id=#{id}")

public Admin getUserInfo(Integer id);

//查询所有用户信息@Select("select\*from admin") public List<Admin>getUserList();

//添加用户信息

@Insert("insert into

admin(name,user\_name,pwd,creat\_time,group\_id)values(#{name},#{user\_name},#{pwd},#{creat\_time},#{gr oup\_id})")

public Integer addUser(Admin admin);

//删除用户

@Delete("delete from admin where id=#{id}") public Integer deleteUser(Integer id);

//动态更新用户@Update("<script>"+ "update admin set id=#{id}"+

"<if test='name!=null'>,name=#{name}</if>"+

"<if test='user\_name!=null'>,user\_name=#{user\_name}</if>"+ "<if test='pwd!=null'>,pwd=#{pwd}</if>"+

"<if test='last\_time!=null'>,last\_time=#{last\_time}</if>"+ "<if test='state!=null'>,state=#{state}</if>"+

"<if test='group\_id!=null'>,group\_id=#{group\_id}</if>"+ "<if test='last\_ip!=null'>,last\_ip=#{last\_ip}</if>"+

94

"<if test='on\_line!=null'>,on\_line=#{on\_line}</if>"+ "<if test='active!=null'>,active=#{active}</if>"+ "where id=#{id}"+

"</script>")

public Integer updateUser(Admin admin);

}

###### 用户service 层：

@Service

public class AdminServiceImpl implements AdminService{ @Autowired

private AdminMapper adminMapper;

public AdminServiceImpl(AdminMapper adminMapper){ this.adminMapper=adminMapper;

}

@Override

public Admin login(String name){ return adminMapper.login(name);

}

@Override

public Admin getUserInfo(Integer id){return adminMapper.getUserInfo(id);}

@Override

public List<Admin>getUserList(){ return adminMapper.getUserList();

}

@Override

public Integer addUser(Admin admin){ return adminMapper.addUser(admin);

}

@Override

public Integer deleteUser(Integer id){ return adminMapper.deleteUser(id);

95

}

@Override

public Integer updateUser(Admin admin){ return adminMapper.updateUser(admin);

}

}

###### 用户controller 层：

@RestController

public class AdminController{

@Autowired

private AdminService adminService;

//存放全局验证码 private String codeString;

/\*\*

\*登陆

\*@param name

\*@param pwd

\*@param code

\*@return json 数据

\*/ @PostMapping("/login")

public RespEntity login(@RequestParam("name")String name,@RequestParam("pwd")String pwd,@RequestParam("code")String code,@RequestParam("last\_ip")String last\_ip){

String sessionCode=codeString; if(!StringUtils.equalsIgnoreCase(code,sessionCode)){//忽略验证码大小写return new RespEntity(RespCode.ERROR,"哦豁！验证码错了呀！");

}

String MD5\_PWD=MD5.encodeByMD5(pwd); System.out.println(name);

Admin admin=adminService.login(name); if(admin==null){

return new RespEntity(RespCode.ERROR,"哦吼！猜错了，木有这个人啊！");

96

}else{ if(!admin.getPwd().equals(MD5\_PWD)){

return new RespEntity(RespCode.ERROR,"哎呦！猜错了！现在输入正确的密码！");

}else if(admin.getState()!=1){

return new RespEntity(RespCode.ERROR,"流着泪说分手，你已不在这干了！");

}else{

//登陆成功，生成 token,将 token 返回给客户端String token=JwtUtil.sign(admin.getId(),name,pwd); System.out.println(token);

if(token!=null){ admin.setLast\_ip(last\_ip); admin.setOn\_line(1);

admin.setActive(Long.toString(new Date().getTime())); Integer result=adminService.updateUser(admin); if(result==1){

Token tokenObj=new Token(token);

return new RespEntity(RespCode.SUCCESS,tokenObj);

}

}

}

}

return new RespEntity(RespCode.FAIL);

}

@GetMapping("/getUserInfo")

public RespEntity getUserInfo(HttpServletRequest request){ String token=request.getHeader("token");

Integer id=JwtUtil.getUserId(token); if(id!=null){

Admin admin=adminService.getUserInfo(id);

return new RespEntity(RespCode.SUCCESS,admin);

}

return new RespEntity(RespCode.ERROR);

}

/\*\*

\*

\*@return

\*/

97

@PostMapping("/logout")

public RespEntity logout(@RequestParam("id")Integer id){ Admin admin=new Admin();

admin.setId(id); admin.setOn\_line(0);

Integer result=adminService.updateUser(admin); if(result==1){

return new RespEntity(RespCode.SUCCESS,"退出成功");

}

return new RespEntity(RespCode.ERROR,"退出失败");

}

/\*\*

\*修改密码

\*@param id

\*@param pwd

\*@return

\*/ @PostMapping("/resetPassword")

public RespEntity resetPassword(@RequestParam("id")String id,@RequestParam("pwd")String pwd){ String MD5\_PWD=MD5.encodeByMD5(pwd);

Admin admin=new Admin(); admin.setId(Integer.parseInt(id)); admin.setPwd(MD5\_PWD);

Integer result=adminService.updateUser(admin); if(result==1){

return new RespEntity(RespCode.SUCCESS);

}else{

return new RespEntity(RespCode.ERROR,"修改密码失败");

}

}

/\*\*

\*生成验证码

\*@throws IOException

\*/ @GetMapping("/validateCode")

public void validateCode(HttpServletResponse response)throws IOException{

//设置响应的类型格式为图片格式

98

response.setContentType("image/jpeg");

//禁止图片缓存response.setHeader("Pragma","no-cache"); response.setHeader("Cache-Control","no-cache"); response.setDateHeader("Expires",0);

VerificationCodeImg vCode=new VerificationCodeImg(130,34,4,50); codeString=vCode.getCode(); vCode.write(response.getOutputStream());

}

@GetMapping("/getUserList") public RespEntity getUserList(){

List<Admin>list=adminService.getUserList(); if(list!=null){

return new RespEntity(RespCode.SUCCESS,list);

}

return new RespEntity(RespCode.ERROR,"获取用户列表失败");

}

@PostMapping("/addUser")

public RespEntity addUser(@RequestParam("name")String name, @RequestParam("user\_name")String user\_name, @RequestParam("pwd")String pwd, @RequestParam("creat\_time")String creat\_time, @RequestParam("group\_id")String group\_id){

Admin admin=new Admin(); admin.setName(name); admin.setName(user\_name); admin.setPwd(MD5.encodeByMD5(pwd)); admin.setCreat\_time(creat\_time); admin.setGroup\_id(Integer.parseInt(group\_id)); Integer result=adminService.addUser(admin); if(result==1){

return new RespEntity(RespCode.SUCCESS,"添加成功");

}else{

return new RespEntity(RespCode.ERROR,"添加失败");

}

}

99

@PostMapping("/deleteUser")

public RespEntity deleteUser(@RequestParam("id")String id){ Integer result=adminService.deleteUser(Integer.parseInt(id)); if(result==1){

return new RespEntity(RespCode.SUCCESS,"删除成功");

}else{

return new RespEntity(RespCode.ERROR,"删除失败");

}

}

@PostMapping("/updateUser")

public RespEntity updateUser(@RequestParam("id")String id, @RequestParam("name")String name, @RequestParam("user\_name")String user\_name, @RequestParam("pwd")String pwd, @RequestParam("group\_id")String group\_id){

Admin admin=new Admin(); admin.setId(Integer.parseInt(id)); admin.setName(name); admin.setName(user\_name); admin.setPwd(MD5.encodeByMD5(pwd)); admin.setGroup\_id(Integer.parseInt(group\_id)); Integer result=adminService.updateUser(admin); if(result==1){

return new RespEntity(RespCode.SUCCESS,"修改成功");

}else{

return new RespEntity(RespCode.ERROR,"修改失败");

}

}

}

###### 用户前端接口：

import ajax from"@/utils/ajax.js";

//登陆

export function login(params){ return ajax({

url:'/login', method:'post',

100

data:params

})

}

//获取用户信息

export function getUserInfo(){ return ajax({ url:'/getUserInfo',

method:'get'

})

}

//登出

export function logout(params){ return ajax({

url:'/logout', method:'post', data:params

})

}

//修改密码

export function resetPassword(params){ return ajax({

url:'/resetPassword', method:'post', data:params

})

}

//获取验证码

export function validateCode(){ return ajax({ url:'/validateCode',

method:'get'

})

}

//获取用户列表

101

export function getUserList(){ return ajax({ url:'/getUserList',

method:'get'

})

}

//添加用户

export function addUser(params){ return ajax({

url:'/addUser', method:'post', data:params

})

}

//删除用户

export function deleteUser(params){ return ajax({

url:'/deleteUser', method:'post', data:params

})

}

//更新用户

export function updateUser(params){ return ajax({

url:'/updateUser', method:'post', data:params

})

}

102