



南京理工大学  
NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

# 毕业设计说明书

作 者： 郭肇强 学 号： 913106840546

学 院： 计算机科学与工程学院

专业(方向)： 软件工程（方向三一卓工）

题 目： 校园招聘平台的设计与开发

指导者： 王晓峰 副教授  
(姓 名) (专业技术职务)

叶峰 技术副总监  
(姓 名) (专业技术职务)

评阅者：  
(姓 名) (专业技术职务)

2017 年 6 月

## 声 明

我声明，本毕业设计说明书及其研究工作和所取得的成果是本人在导师的指导下独立完成的。研究过程中利用的所有资料均已在参考文献中列出，其他人员或机构对本毕业设计工作做出的贡献也已在致谢部分说明。

本毕业设计说明书不涉及任何秘密，南京理工大学有权保存其电子和纸质文档，可以借阅或网上公布其部分或全部内容，可以向有关部门或机构送交并授权保存、借阅或网上公布其部分或全部内容。

学生签名：

年 月 日

指导教师签名：

年 月 日

# 毕业设计说明书中文摘要

由于互联网技术的成熟和便利，人们越来越习惯于利用互联网这一新工具来解决旧问题。其中基于 B/S 的 WEB 应用的开发为是最突出的表现，几乎占据了整个新型市场。WEB 应用的流行得益于它的开发模式简单、通用和平台无关的特性，可以使用户随时随地享受到应用带来的便利。

传统的招聘行业要求公司和学生同时到指定场所进行求职应聘方面的咨询，极为消耗人力物力。本次校园招聘项目的研发旨在方便学生对招聘信息的获取和为公司接收简历提供便利，这样可以极大节省双方的精力从而提高效率。

本文主要介绍了项目的设计与开发过程，包括背景研究、平台搭建、数据库设计和系统功能设计介绍等各方面的内容，使读者对整个校园招聘平台的研发有一个整体的认识。

**关键词** 校园招聘 WEB 开发 PHP 数据库 Laravel

# 毕业设计说明书外文摘要

**Title**                    The Design and Development of Campus  
                                 Recruitment Platform

## **Abstract**

Due to the maturity and convenience of Internet technology, people are becoming more and more accustomed to using the Internet as a new tool to solve the old problems. Among them, the development of WEB application based on B/S is one of the most outstanding performance, which almost occupied the whole new market. The popularity of WEB applications thanks to its development model is simple, generic and platform independent features, allowing users to enjoy the convenience of applications anytime, anywhere.

Traditional recruitment industry requires companies and students at the same time to the designated location for job counseling, Causing great waste for human and material resources. The campus recruitment project research and development is designed to facilitate the recruitment of students to obtain information and to facilitate the company to receive resumes, which can greatly save the energy of both sides to improve efficiency.

This paper mainly introduces the design and development process of the project, including the research background, platform construction, database design and system function design and other aspects of the content, so that readers of the development of campus recruitment platform has an overall understanding.

**Keywords**   Campus Recruitment   WEB   PHP   Database   Laravel

## 目 次

1	绪论	1
1.1	选题背景	1
1.2	研究意义	1
1.3	实现目标	2
2	开发环境及关键技术	3
2.1	WEB 服务器 Apache	3
2.2	数据库管理系统 MySQL	3
2.3	编程语言 PHP	3
2.4	开发框架 Laravel	4
2.5	组件引用	4
3	需求分析	5
3.1	系统可行性分析	5
3.2	功能需求	5
3.2.1	学生需求	5
3.2.2	校招单位需求	6
3.2.3	管理员需求	6
4	总体设计	7
4.1	系统功能结构	7
4.2	MVC 模式的实现	8
4.2.1	数据层	8
4.2.2	控制层	9
4.2.3	视图层	9
5	数据库设计	11
5.1	关系定义及说明	11
5.2	关联关系及消除冗余	21
5.3	E/R 图	21
6	功能模块设计	23
6.1	用户认证模块	24
6.2	列表管理模块	25
6.3	内容展示模块	27
6.4	表单提交模块	29
6.5	异步通讯模块	31

---

6 . 6	消 息 管 理 模 块 .....	3 2
6 . 7	在 线 交 流 模 块 .....	3 3
6 . 8	上 传 下 载 模 块 .....	3 5
结 论	.....	3 6
致 谢	.....	3 7
参 考 文 献	.....	3 8

## 1 绪论

### 1.1 选题背景

#### 1.1.1 互联网的产生和发展

农业文明迈入工业文明的步伐是产生世界物理联系的动力。从工业文明到信息文明，则实现了物理联系到信息联系的转变<sup>[1]</sup>。催生这种变化的主角便是日臻完善的互联网技术。它的诞生使我们处在一个信息爆炸的时代，正在无形地推动经济形态不断发生演变。

我们的世界正在发生巨变。人们更乐意选择在网上办公、购物、交流、娱乐等等。互联网的普及可以使一家微博网站一天内发布的信息超过纽约时报辛勤工作 60 年的工作量，全球最大的视频网站一天上传的影像可以连续播放 98 年<sup>[2]</sup>。可以肯定的是，人类进步的身影中将时时刻刻伴随着互联网的发展。生活在互联网时代的我们，拥有着前人从未有过的如此巨大的信息资源和丰富的物质便利条件。在我国，上网人数每年都在快速增长，人们的生活已经无法离开互联网的支撑。传统的行业若是不积极向互联网方向转变，随时都有被淘汰的可能。

#### 1.1.2 WEB 技术的产生、发展和应用

互联网能够对人们的生活产生如此巨大的影响得益于 WEB 开发技术的快速发展和广泛应用。WEB 技术是建立在互联网平台上的一种典型的网络服务技术。它对用户使用互联网技能的要求较低，极大地推动互联网的在大众中的普及。

Web 平台能够盛行的另一个很潜在因素就在于它是对用户友好的。它可以在页面上同时显示绚丽多彩的多媒体产品和样式丰富文本，这使得用户能够轻松获取自己想要的信息。在 Web 技术产生之前的因特网上只有纯文本的信息，这导致网络技术尽在专业人士之间使用。此外，相比较其它类型的互联网应用，它具有平台无关的特性。WEB 应用可以利用其分布式的特点为用户提供超大规模的数据量，足以满足用户对信息的要求。WEB 页面的交互性是它的又一亮点之一，交互性保证了 WEB 站点信息的持续更新。

WEB 技术的发展历程影射互联网的迅猛发展。WEB 技术的广泛应用使人们真切体会到了互联网所带来的便利<sup>[3]</sup>。现如今，大多数网络应用通过 WEB 页面为人们提供服务，只要有一个互联网终端或者一台个人电脑，就可以在联网的情况下随时随地获取所需的服务。每天有成百上千万个页面被浏览。在未来，将会有更多的行业采用基于 WEB 技术互联网形式为它的用户提供服务。

### 1.2 研究意义

在互联网发展如火如荼的今天，传统行业已经收到极大地打压，摆在在它们面前的道路只有两条，要么向互联网方向转变，要么被淘汰。

传统的招聘行业需要公司到人才市场招聘。而应聘者需要每日到人才市场排长队等待应聘者的简历接收、面试、笔试等流程。非常耗费人力物力。双方没有信息交互的平台，效率十分低下。求职者应聘时目标性较小，成功率较低。由于网络技术不断发展，传统的招聘行业可以采用 WEB 平台来发布信息，应聘者在该平台上收集并筛选信息，提前做好响应准备，不仅节省人力物力，而且对于双方的要求都有进一步的满足。

### **1.3 实现目标**

本项目旨在完成学生的求职就业问题和公司的人才招聘问题。

本项目的实现目标是完成一个 WEB 网站，学生可以通过该网站查找职位信息，公司信息；校招单位可以通过网站搜索简历，接收学生的职位申请。网站有管理员来专门进行相关的审核工作，并且能够发布招聘会信息扩大招聘途径。



## 2 开发环境及关键技术

项目开发的首要任务是搭建好开发环境，根据需求选择好合适的开发语言，针对不同的问题选择相适应的框架和容器，这些工作做好之后才能进行之后程序的设计与开发工作。

### 2.1 WEB 服务器 Apache

WEB 服务器是运行整个项目的容器。该容器内存储着项目的页面和数据为用户提供。它负责接收从客户端传送来的各种请求。返回给用户的相应是各种 WEB 文档和数据，如 HTML 页面、JPEG 图形文件和下载文件等等。目前有三种 WEB 服务器较为流行，分别是 Apache、Nginx 和 IIS。

目前在世界上最受人们热捧的服务器便是 Apache。因为它具有跨平台的特性，几乎可以运行在任何计算机平台之上<sup>[4]</sup>。经过多方便的修补漏洞和完善，它的性能和安全性已经得到极大保障。基于这些优势，本项目拟采用 Apache 作为整个站点的服务器。

### 2.2 数据库管理系统 MySQL

数据库管理系统是用来存储整个项目中数据的一种较为基础的软件。用它来管理数据是必不可少的，因为在大量数据面前，简单地将数据存储到文件中是不可靠的，因为我们无法知道数据之间的结构和关联，更不能随心所欲地操作这些数据。数据库管理系统帮助我们解决了这些难题，可以使我们对数据的操作变得简单，从而提高安全性和便捷性。

MySQL 是一种开放源码的数据库，这是它普及的重要原因。因为它的体积小、速度快、成本低，一般来说，该数据库在中小型项目中应用十分广泛。本项目由于采用 PHP 作为开发语言，对 MySQL 表现出操作友好性，非常适合用 MySQL 作为其数据库。

### 2.3 编程语言 PHP

编程语言的选择对于一个项目的开发十分重要，当今各种语言种类繁多，其侧重点是各部相同，对于 WEB 开发者来说，选择的语言要不仅效率要高，还要能够跨平台，PHP 作为一种服务器端脚本语言，一开始它便是为专注于 WEB 应用而设计，因而是开发 WEB 应用程序的不二之选，由于它可以将程序嵌入到 HTML 文件中，运行时的效率超出纯 HTML 和 CGI 很多。PHP 的程序开发周期相对较快，这得益于它的简单易学，它的语法风格和 C、Java 类似，对于程序开发人员来说极易上手。PHP 对系统资源的消耗极少，可以大幅提升应用的效率。

PHP 的高效、便捷、实用等优势使得它在 WEB 开发中脱颖而出，越来越成为人们的首选。它可以被嵌入超文本标记语言到 HTML 中，在 WEB 开发中会减少开发人员的难度。它的语法中没有

严格的类型限制，因此十分适合非数值型应用开发，故作为本项目的开发语言。

## 2.4 开发框架 Laravel

一个好的开发框架是高效开发的保证，Laravel 框架便是这样的 MVC 架构，它以优雅而著称，开发人员只需专注于逻辑设计与开发，由框架来完成其它的工作。

Laravel 对 WEB 开发中需要考虑的流程都封装好了，在需要时尽管使用便是。它提供给使用者的是一个更加富有表现力的优雅接口，所有代码开源展示，用户完全可以自我定制。组件化的开发思想在这个框架的设计中体现的淋漓尽致，本身作为组件的它同时又是基于其它组件实现的，在使用该框架的时候，你完全可以体会到局部组件的完美设计会成就整体系统的完美展现。扩展包是它的又一大特性之一，除去基本的 MVC 功能模块，它在扩展包中实现了要在 WEB 开发中考虑的各种功能，例如加密机制、文件上传下载、单元测试、自动分页等。最值得一提的是，它的高质量的文档和强大社区的支持使得开发成为一件乐事。

## 2.5 组件引用

组件引用的意义和框架一样，可以让开发人员专心设计应用逻辑，不必考虑其它繁琐的底层操作。其实，Laravel 框架便是众多组件的集合，但是任何框架在设计时不可能面面俱到，因此还需引入其它的组件以实现更加完善的功能。

本项目中引用的组件包括以下几部分：

（1）前端基础组件。作为与用户直接打交道的视图部分，良好的使用体验会使得系统迅速获得大量用户。本项目中引用 jQuery 作为前端基础的 js 交互组件，bootstrap 在其基础上进一步封装后用来改善页面的观感。

（2）输入组件。如果手动填写时间，会造成许多问题，如格式十分复杂会使用户体验降低，此外还会造成大量错误。一个好的输入组件可以解决这个问题，它让用户选择内容代替手动输入，既方便又准确。这些组件包括有日期、城市和大学等。

### 3 需求分析

搭建好程序开发的环境之后，准备工作便已经妥当，但是还需要做一件重要的事情才能够开始开发，那便是项目的需求分析，没有分析便不能准确把握系统设计中各个方面的要点。做好事前分析工作是整个项目开发的前奏。

#### 3.1 系统可行性分析

一个系统是否可以完整实现，可行性分析十分关键。尽管这部分没有实际的项目内容，却是不可或缺的一步。做好可行性分析之后，在实际情况允许的条件下，才可以开始进行后续相关工作。

主要在以下几方面展开分析内容：

(1) 发展可行性。是否具有发展潜力是衡量一件产品的重要指标。本项目是要开发基于 WEB 的应用程序，当今是 WEB 程序发展势头良好的时代，由于 WEB 技术的发展，其优越性逐渐展示出来，大量的应用程序开始着眼于转向 WEB 开发，在未来的时代，WEB 产品必当占据软件行业的主导市场，其发展前途不可估量。因而其发展可行性已经具备。

(2) 技术可行性。WEB 技术从产生到现在，已经发展得越来越好，前端技术如 HTML、CSS 和 JavaScript 已经发展到趋于完善的程度。后端技术如 JSP、PHP 和 ASP 等都在不断地更新换代以适应新的需求，如今大多数基于信息发布的应用都可以以 WEB 的形式开发，并且具有优越的性能，因此本项目的技术可行性是具备的。

(3) 周期可行性。应用程序的研发周期可能会影响到产品的发布与市场。在竞争激烈的软件研发行业，能够在短期内研发出高质量的产品无疑是十分关键的。每个软件工程师都在于时间赛跑，能够在研发周期上占得先机，将会对产品和企业的发展提供巨大的机遇。本项目拟采用 Laravel 框架作为基础进行开发，开发者只需研究业务细节而无需考虑底层实现，极大地节省开发时间和精力，故而周期可行性也已具备。

#### 3.2 功能需求

功能需求是对整个系统功能的一种把握。要在完全搞清楚整个系统功能需求之后来完成其预期的任务。本系统中的有三种类型的用户，他们分别有不同的功能目标。下面详细探讨他们各自的功能需求。

##### 3.2.1 学生需求

学生对校招平台的总体需求是通过平台获取招聘信息，找到合适的工作岗位以完成自己的求职应聘。主要包括：

(1) 简历管理。学生在平台上制作自己的简历，可制作多份供应聘不同的职位，在完成之后将其投放人才库中供公司检索或者直接投递给自己心仪的公司岗位以获取面试和笔试的机会。

(2) 信息内容管理。学生能够查询和浏览各类招聘相关信息，如招聘会信息、公司动态。校园宣讲等等。并且具备在相关信息中做出反馈的要求，如关注一家公司，收藏某个职位等。

(3) 交流管理。学生能够在平台上收到公司发送给自己的消息，并且可主动发送消息给相应公司进行招聘事宜的交流。

### 3.2.2 校招单位需求

校招单位对平台的总体需求是通过平台找到适合自己岗位的毕业生。主要包括：

(1) 职位管理。公司发布和修改本公司需要招聘的岗位内容，提供给应聘者获取其信息；

(2) 信息管理。公司浏览系统人才库中的学生简历，对其满意者可以主动联系学生前来参加应聘；

(3) 招聘宣讲管理。公司可以发布自己的宣讲会信息，使学生有机会更进一步深入了解公司的情况；与此同时，公司可以参加各省市即将举办的招聘会来扩大自己的招聘渠道；

(4) 交流管理。公司能够在平台上收到学生的操作信息，并且可以通过这些信息收集到学生对于公司的看法和意见并针对具体问题做出相应调整。

### 3.2.3 管理员需求

管理员的需求是信息发布和对平台进行维护。主要包括：

(1) 认证管理。对公司注册的信息进行真伪认证，以防止虚假的公司通过该平台进行不法行为；

(2) 招聘管理。招聘会相关信息由管理员直接进行发布和维护。公司可以报名参加招聘会进行招聘，学生通过浏览招聘会信息进而参加应聘指定的公司。

4 总体设计

4.1 系统功能结构

本系统的用户分为三种类型，因此将分别展示每种用户的具体功能结构。

(1) 学生用户如图 4.1 所示：

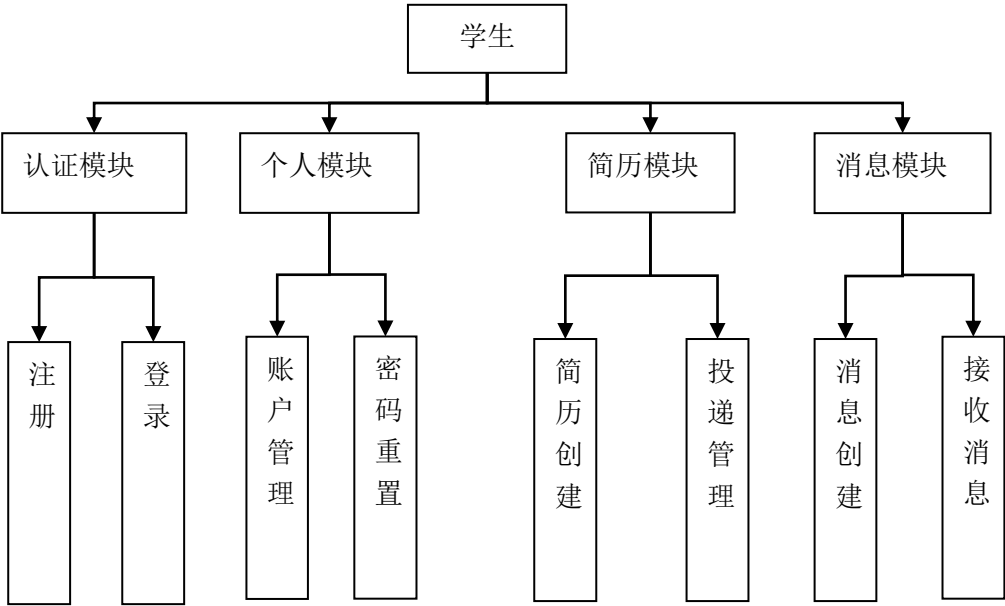


图 4.1 学生功能图

(2) 校招单位用户如图 4.2 所示：

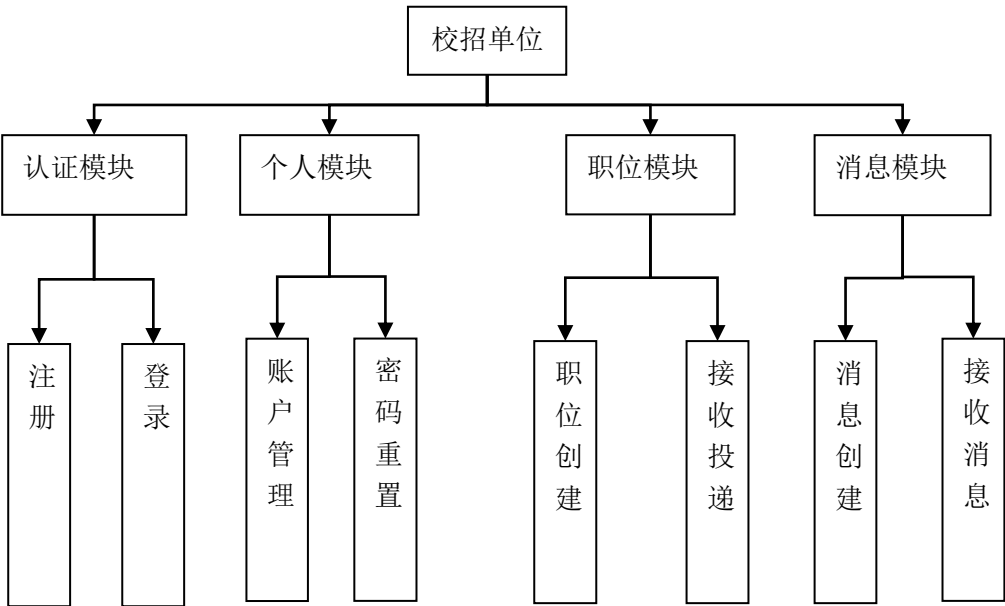


图 4.2 校招单位功能图

(3) 管理员用户如图 4.3 所示：

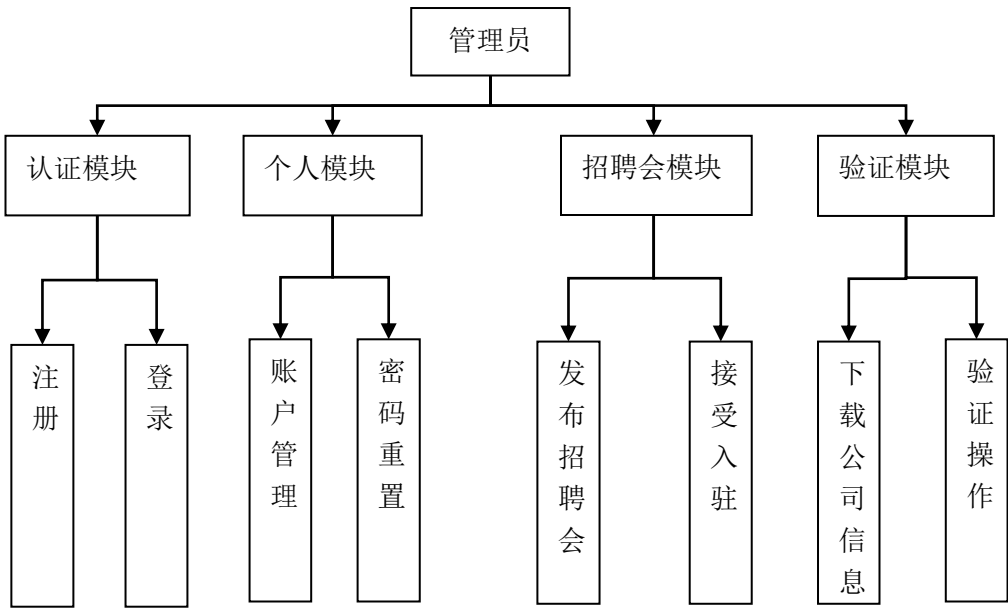


图 4.3 管理员功能图

4.2 MVC 模式的实现

MVC 设计模式是一种开发性能和使用性能俱佳的 WEB 代码模式。它的核心理念就是分离和重组，即是组件化开发的扩展和延伸。他将数据、视图、控制分离在三个模块中，各个部分在逻辑上相互独立，其内部相互之间没有任何关联，成功地实现了解耦，使得程序来发人员可以各司其职，只要遵守接口规范，即可实现每个模块的交互，从而将部分整合为一个完整的系统。

Laravel 框架正是基于这个理念设计出来的。它对每个子系统进行详细地划分，在整体上实现了 MVC 的模式。此外，它也是对组件友好型的开发框架，其中内嵌了各种功能的子模块共使用者调用，极尽一切可能简化开发者的步骤，给开发人员留下充足的精力实现其详尽的业务逻辑。

4.2.1 数据层

在一个完整的系统或者项目中，最宝贵的资源莫过于其中的数据。数据是程序的血肉，它贯通程序执行的始终，没有数据的程序开发是毫无意义的。对数据的操作是程序中至关重要的，它不容许有丝毫的差错和模棱两可；同时，操作数据是繁琐的和令人头疼的，是开发人员最关系的部分。因此，对数据进行分层包装显得异常重要。这不仅可以使得操作数据变

得轻松畅快，更重要的是，由于分层后的数据由模块内部的代码直接进行验证和操作，数据的安全性得到极大保障。

在 Laravel 框架中，每种数据模型对应一个继承了 Model 的类，通过类与数据库的映射来进行操作数据。通过底层的封装，我们可以通过调用该类中的方法实现对数据的操作而不必再与数据库打交道，完全降低了程序开发的冗余度，使得极速开发成为可能。

在本项目中，所有的 Model 模型类均存放在 `\app\Models` 目录下。

#### 4.2.2 控制层

一个高效简洁的系统必须要有一个灵活强大的大脑，那就是控制器。如果说数据层是程序的血肉，那么，控制层则当之无愧地成为程序运行的大脑，有了大脑的正确指挥，整个程序才能有条不紊地运行。在正常情况下，控制器会支撑程序执行，一旦遇到异常，他就会调用紧急预案才解决问题以免程序崩溃。试想一个没有正确控制流程的程序是不可能算出预期的结果的。

在 Laravel 框架中，控制器均封装在 Controller 子类中，开发者可以定义各个模块的控制器来分别指挥不同的工作，进一步实现分离，这样做的好处不言而喻，可以使系统逻辑更加清晰。控制层连接视图层和数据层的纽带。

#### 4.2.3 视图层

同样一个形象的类比，对于一个人来说，有了丰富的血肉和精明的大脑，如果不通过它的言谈举止等外观表现出它的内涵，对于其他人来说，我们仍无法感受它的魅力。视图层正是这样的输出系统，它将一个完美的系统活灵活现地展现在用户面前，用户不必精通它那复杂的逻辑实现和数据处理，只需要通过视图简单操作即可实现系统的运行，按需获取其服务。

在 Laravel 框架中，所有展现给用户的视图均是一个 php 页面，这部分代码均存放在 `/app/resources/views/` 目录下。框架中的 blade 模板简化了视图的设计冗余度，简化开发和维护难度。

视图层来传递和接收数据，控制层来分发任务和数据，数据层来处理数据。它们彼此相互独立、互不干扰。在三个层面的完美配合下，系统便能够有条不紊地运行。详细如图 4.4 所示：

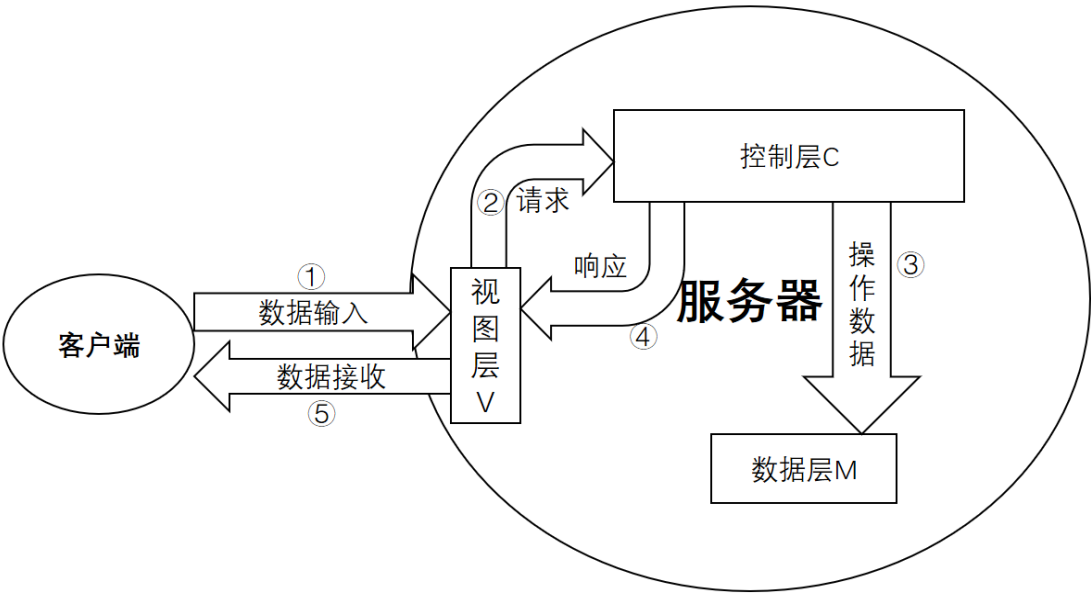


图 4.4 程序请求响应模式图



## 5 数据库设计

除一些全局变量之外，程序运行的数据绝大部分存储在数据库中，故而应当重视数据库的相关设计。选取数据库管理系统是十分关键的一步，要考虑多方面因素，如数据量、数据类型、结构模式和价格成本等。本项目中的数据均为关系型数据，适合采用关系型数据库进行存储，数据量不是很大，作为中小型数据库的 MySQL 比较合适，这个数据库既不会造成大量的浪费，也能够出色地完成任务。在确定采用的数据库之后，应当认真考虑关系设计。关系的冗余程度直接影响着系统的性能，而关系的贴切程度则会决定系统功能的完善。

### 5.1 关系定义及说明

通过对需求的详细分析，本项目中设计了共计 23 张关系表，其具体描述如下。

#### (1) 用户信息表 users

此表描述了用户的信息，用来标识用户的类型，使用户可以进入系统中进行求职或者招聘。其结构见表 5.1：

表 5.1 用户信息表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
name	varchar(255)	NO		NULL		昵称
email	varchar(255)	NO	UNI	NULL		邮箱
password	varchar(255)	NO		NULL		密码
usertype	int	NO		NULL		用户类型
avatar	varchar(255)	YES		NULL		头像地址
remember_token	varchar(100)	YES		NULL		口令
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

#### (2) 公司信息表 companies

此表描述了公司详细信息，用来进行公司注册认证和发布职位信息。学生和管理员可以查看该表中的记录来了解公司的各种信息。其结构见表 5.2：

表 5.2 公司信息表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
uid	int	NO		NULL		用户ID
name	varchar(255)	NO		NULL		公司名称
scale	int	NO		NULL		公司规模
property	int	NO		NULL		公司性质
industry	Int	NO		NULL		所属行业
website	varchar(255)	NO		NULL		公司主页
city	varchar(255)	NO		NULL		城市
address	varchar(255)	NO		NULL		公司地址
abstract	varchar(255)	NO		NULL		公司介绍
focus	int	NO		NULL		专注人数
job_number	int	NO		NULL		职位数目
auth	int	NO		NULL		是否认证
phone	varchar(255)	NO		NULL		电话
email	varchar(255)	NO		NULL		邮箱
auth_file_path	varchar(255)	NO		NULL		文件路径
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(3) 职位详情表 positions

此表描述了公司发布的职位信息，学生和管理员可以查看此表了解公司的职位详情。其结构见表 5.3：

表 5.3 职位详情表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
cid	int	NO		NULL		公司ID
name	varchar(255)	NO		NULL		职位名称
type	int	NO		NULL		职位类型
salary	int	NO		NULL		薪资
abstract	varchar(255)	NO		NULL		职位介绍
recruit_number	varchar(100)	NO		NULL		招募人数
post_number	int	NO		NULL		投递人数
collection_number	int	NO		NULL		收藏人数
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(4) 简历信息表 resumes

此表描述了学生的一份简历信息，通过这张表可以了解到这份简历的一些基本信息如完整度和求职意向。其结构见表 5.4:

表 5.4 简历信息表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
uid	int	NO		NULL		学生ID
name	varchar(255)	NO		NULL		应聘职位
complete	int	NO		NULL		完整度
is_post	int	NO		NULL		是否已读
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

#### (5) 简历条目—基本信息表 item\_basics

此表描述了简历中的基本信息模块，公司可以通过查看该模块而快速了解到应聘者的姓名、年龄等基本信息，对应聘者有一个初步的了解，每份简历最多有一条基本信息。其结构见表 5.5:

表 5.5 简历条目—基本信息表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
name	varchar(255)	NO		NULL		姓名
gender	int	NO		NULL		性别
height	int	NO		NULL		身高
weight	int	NO		NULL		体重
birthday	varchar(255)	NO		NULL		生日
idnumber	int	NO		NULL		证件号码
nation	int	NO		NULL		民族
health	int	NO		NULL		健康
status	int	NO		NULL		政治面貌
marriage	int	NO		NULL		婚姻状况
origin	varchar(255)	NO		NULL		生源地
permanen	varchar(255)	NO		NULL		户口所在地
photo	varchar(255)	NO		NULL		照片
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

#### (6) 简历条目—教育经历表 item\_educations

此表描述了简历中的教育经历模块，公司可以在该模块中了解到应聘者的学历、教育经历等信息，查看是否满足职位的基本要求，每份简历可以有多条教育经历。其结构见表 5.6：

表 5.6 简历条目—教育经历表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
startdate	date	NO		NULL		开始时间
enddate	date	NO		NULL		结束时间
school	varchar(255)	NO		NULL		学校
degree	int	NO		NULL		学位
qualification	varchar(100)	NO		NULL		学历
type	int	NO		NULL		培养类型
campus	varchar(255)	NO		NULL		学院
major	varchar(255)	NO		NULL		专业
class	varchar(255)	NO		NULL		课程
rank	int	NO		NULL		排名
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(7) 简历条目—联系放式表 item\_contacts

此表描述了简历中的联系放式模块，公司若是对应聘者的简历较为满意就可以通过这些联系途径通过对方前来应聘。其结构见表 5.7：

表 5.7 简历条目—联系方式表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
phone	varchar(255)	NO		NULL		电话
emergency	varchar(255)	NO		NULL		紧急联系人
email	varchar(255)	NO		NULL		邮箱
address	varchar(255)	NO		NULL		地址
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(8) 简历条目—项目经验表 item\_projects

此表描述了简历中的项目经验模块，公司一般更中意于有动手能力的应聘者，若是应聘者有这方面的经历会受到公司对的特别青睐。其结构见表 5.8：

表 5.8 简历条目—项目经验表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
name	varchar(255)	NO		NULL		项目名称
startdate	date	NO		NULL		开始时间
enddate	date	NO		NULL		结束时间
content	varchar(255)	NO		NULL		项目内容
harvest	varchar(255)	NO		NULL		收获
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

## (9) 简历条目—工作经历表 item\_works

此表描述了简历中的工作经验模块，与上一模块类似，这个模块对应聘者的要求有进一步的提高。其结构见表 5.9：

表 5.9 简历条目—工作经历表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
startdate	date	NO		NULL		开始时间
enddate	date	NO		NULL		结束时间
company	varchar(255)	NO		NULL		工作单位
position	varchar(255)	NO		NULL		工作岗位
content	varchar(255)	NO		NULL		项目内容
harvest	varchar(255)	NO		NULL		收获
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

## (10) 简历条目—所获奖励表 item\_awards

此表描述简历中的获奖信息模块，学生通过完成此模块，可以展现自己的优秀一面，为自己赢得好评。其结构见表 5.10：

表 5.10 简历条目—所获奖励表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
name	varchar(255)	NO		NULL		奖项名称
level	int	NO		NULL		等级
time	date	NO		NULL		获奖时间
details	varchar(255)	NO		NULL		详细内容
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(11) 简历条目—求职意向表 item\_jobs

此表描述简历中的求职意向模块，公司通过查看此模块，会更加了解求职者的心理期许，从而因人而异进行面试。其结构见表 5.11：

表 5.11 简历条目—求职意向表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
property	int	NO		NULL		公司性质
city	varchar(255)	NO		NULL		工作城市
state	int	NO		NULL		目前状态
salary	int	NO		NULL		期望薪资
arrival	date	NO		NULL		到岗时间
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(12) 简历条目—其它技能表 item\_skills

此表描述简历中的技能模块，一些额外的技能也是表现自己的重要因素，同时也会成为招聘的通过瓶颈，如英语等级要求。其结构见表 5.12：

表 5.12 简历条目—技能表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
type	int	NO		NULL		技能编号
level	varchar(255)	NO		NULL		技能水平
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

## (13) 简历条目—自我评价表 item\_comments

此表描述简历中的自我评价模块，HR 通过应聘者对自己的评语可以看出此人的认知水平和自我定位，更好地为 HR 留下深刻印象。其结构见表 5.13：

表 5.13 简历条目—自我评价表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
details	varchar(1000)	NO		NULL		自我评价
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

## (14) 简历条目—附件信息表 item\_attaces

此表描述简历中的附件信息，通过上传一些自己的附件来说明额外的情况，如一些证书材料，图片等等，可以更加真实地表现自己。其结构见表 5.14：

表 5.14 简历条目—附件信息表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
name	varchar(255)	NO		NULL		简历名称
address	varchar(255)	NO		NULL		存储地址
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

## (15) 职位收藏表 collections

此表描述学生对某个职位的收藏信息，学生可以在自己的账户中快速定位自己偏爱的职位，了解职位的详情。其结构见表 5.15：

表 5.15 职位收藏表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
uid	int	NO		NULL		学生ID
pid	int	NO		NULL		职位ID
is_read	int	NO		NULL		是否已读
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(16) 公司关注表 follows

此表描述公司在学生中的关注情况，对于关注本公司的用户，他们可以看到被关注者的相关动态，对自己心仪的公司会有更加深入的了解。其结构见表 5.16：

表 5.16 公司关注表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
uid	int	NO		NULL		学生ID
cid	int	NO		NULL		公司ID
is_read	int	NO		NULL		是否已读
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(17) 简历投递表 delivers

此表描述学生与公司之间通过简历的联系，学生投递简历给公司以应聘职位，公司接收简历并可以回馈学生相关信息。其结构见表 5.17：

表 5.17 简历投递表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
rid	int	NO		NULL		简历ID
pid	int	NO		NULL		职位ID
is_read	int	NO		NULL		是否已读
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(18) 信息交流表 information

此表描述的是不同种类用户之间的相互交流，通过生成此表的数据来记录交流信息。其



结构见表 5.18:

表 5.18 信息交流表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
sender	int	NO		NULL		发送者ID
receiver	int	NO		NULL		接收者ID
content	varchar(500)	NO		NULL		消息内容
is_read	int	NO		NULL		是否已读
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(19) 宣讲会表 career\_talks

此表描述公司发布的宣讲会信息, 学生可以查看此表的记录来选择相应的地点前往参加。

其结构见表 5.19:

表 5.19 宣讲会表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
cid	int	NO		NULL		公司ID
day	date	NO		NULL		宣讲日期
city	varchar(256)	NO		NULL		宣讲城市
college	varchar(256)	NO		NULL		宣讲学校
address	varchar(256)	NO		NULL		详细地址
details	varchar(2000)	NO		NULL		宣讲说明
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(20) 招聘会表 job\_fairs

此表描述管理员发布的招聘会信息, 学生和公司均可查看该表, 并且公司可以参与该招聘会, 以扩大其招聘规模。其结构见表 5.20:

表 5.20 招聘会表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
aid	int	NO		NULL		管理员ID
name	varchar(256)	NO		NULL		招聘会名称
time	date	NO		NULL		举办时间
host_address	varchar(256)	NO		NULL		主办地点
total_number	int	NO		NULL		总席位数
used_number	int	NO		NULL		使用席位
details	varchar(2000)	NO		NULL		详细说明
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(21) 入驻招聘会表 joins

此表描述的是公司参加招聘会的信息，通过查看招聘会中有哪些公司参加，学生会自行选择相应公司进行应聘。其结构见表 5. 21：

表 5.21 入驻招聘会表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
jid	int	NO		NULL		招聘会ID
cid	int	NO		NULL		公司ID
is_read	int	NO		NULL		是否已读
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

(22) 公司动态表 dynamics

此表描述的是公司在有新职位发布或者公司信息由修改要提醒关注该公司的学生相关变化的通知，学生也可自行查看相关公司的动态。其结构见表 5. 22：

表 5.22 公司动态表

属性	类型	可否为空	键	默认值	其它	备注
id	int	NO	PRI	NULL	自增	主键
cid	int	NO		NULL		公司ID
content	varchar(500)	NO		NULL		动态内容
created_at	timestamp	YES		NULL		创建时间
updated_at	timestamp	YES		NULL		更新时间

5.2 关联关系及消除冗余

在本项目中有以下几个模块需要进行关系关联以降低冗余度：

（1）简历避免冗余

本项目中简历关系的设计充分考虑了会产生冗余的情况，若是简历中的所有信息之存放在一张表中则会产生两种，其一是并非每张简历的内容都是等长的，如张三有工作经历却没有项目经验和获奖情况，而李四则除了没有工作经历之外其他各方面都有涉及，它们并非在某个条目上都是一样的，最后结果就是张三存放 8 个条目，李四存放 9 个条目，但是它们却 都占据了 10 个条目的空间，会产生大量内部碎片，造成空间浪费。其二是信息表达不完整，如张三曾经有过 3 份工作，却只能填写一份经历，这对于能力的展现是有所缺失的，对于应聘动手能力较强的岗位，公司会比较看重一个人的工作经历时，这些信息是十分有用的，一张固定大小的表完全不能表达出这些内容。

我特意将一张表分离成 11 张表，其中一张总表描述简历的一些全局参数，如求职意向和完整度等。其余 10 张表是简历的具体各项内容的详细展示，比如基本信息、教育经历、联系方式等。在一张主表有一个 id 作为其他 10 张条目信息表的外键，并在其中以 rid 的形式存放，每张条目表通过 rid 找到对应的主表，这样就实现了简历的消除冗余设计。

（2）消息关系关联

在每种类型的用户之间进行交流时，如何将信息的发送方关联起来才能使得通信双方都能收到反馈呢？每位用户均可产生多条与之相关的消息。解决办法之一便是在消息表中增加两个外键，其一是 senderId 来标识发送方，另外一个 receiverId 来标识接收方，通过两个外键便可将一条消息的收发双方进行定位，从而实现消息的发送定位及接收。类似消息关联的还有关注关系、收藏关系、动态消息关系等。

5.3 E/R 图

每张表的设计是有相关的关系的，通过 E/R 图来展示她们之间的联系。鉴于布局限制，

每个实体只显示了部分属性信息。其结构如图 5.1 所示：

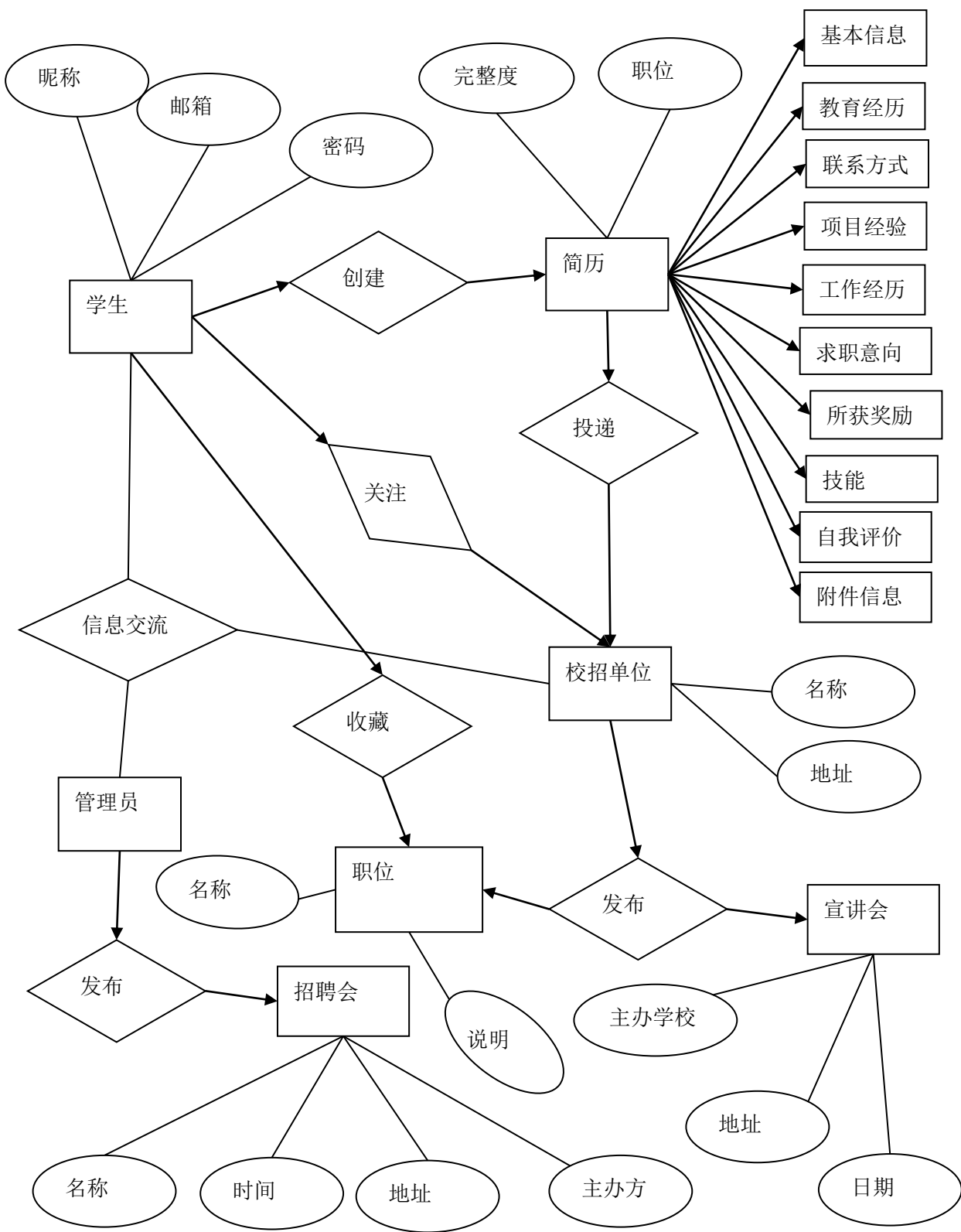


图 5.1 E/R 关系图

## 6 功能模块设计

本项目中功能较为杂乱，在本文中不能做到讲解所有的功能。根据各功能之间的共性可以具体划分为八个模块，每个模块负责完成一类具体任务。将每个模块分类处理，将会大幅简化程序开发难度，提高应用开发速度从而缩短周期。也便于展开后期维护和升级工作。

本项目的目录结构如图 6.1 所示，其中 `recruit/app/Http` 中存放的是控制器类和一些中间件验证类，`recruit/app/Models` 目录下存放的是数据库映射类，`recruit/resources/views` 目录存放全部的视图文件，而 `recruit/public` 目录下存放的是一些资源文件如 `js`、`css` 和图片等和用户上传的文件。在 `recruit/routes/web.php` 中存放的是项目的路由地址转换文件。

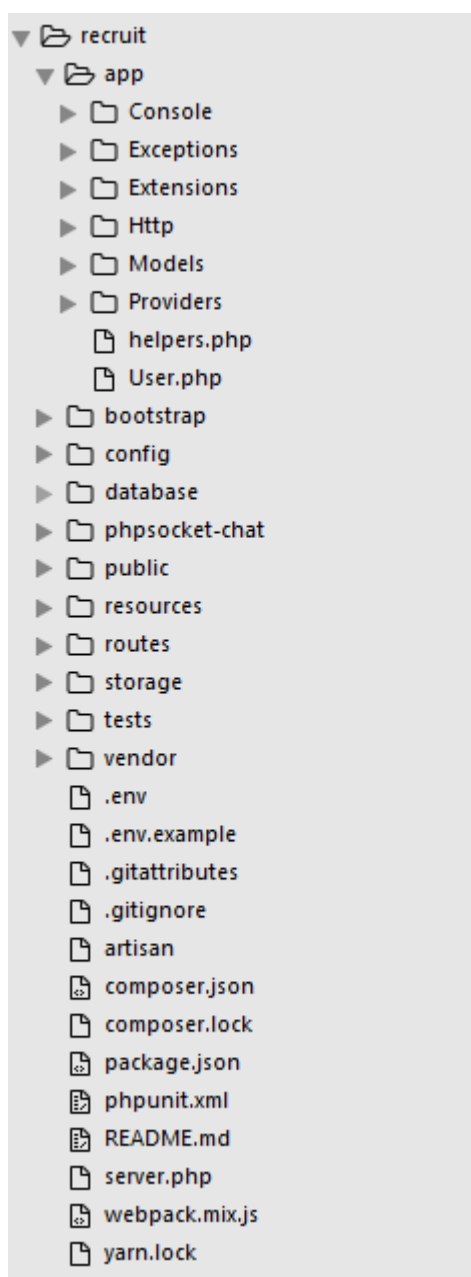


图 6.1 项目文件目录

## 6.1 用户认证模块

目前大多数系统都具有用户数据，只有通过认证的用户才能使用其大部分功能。本模块包括用户注册、登录和用户的划分。在实现时这一模块有赖于框架的支持，Laravel 自带的 auth 门面提供了良好的用户认证支持，既方便又安全，通过命令 `make::auth` 即可实现。对用户类型进行区别时是通过设置 users 表中的 `usertype` 属性来进行区分。

页面展示核心代码见图 6.2:

```
<form class="form-horizontal" role="form" method="POST" action="{{ route('login') }}">
    {{ csrf_field() }}
    <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
        <label for="email" class="col-md-2 control-label">用户</label>
        <div class="col-md-8">
            <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}" required
                autofocus placeholder="请输入用户邮箱"><!-- 用户邮箱输入框-->
            @if ($errors->has('email'))
                <span class="help-block">
                    <strong>{{ $errors->first('email') }}</strong>
                </span>
            @endif
        </div>
    </div>
    <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
        <label for="password" class="col-md-2 control-label">密码</label>
        <div class="col-md-8"><!-- 密码输入框-->
            <input id="password" type="password" class="form-control" name="password" required>
            @if ($errors->has('password'))
                <span class="help-block">
                    <strong>{{ $errors->first('password') }}</strong>
                </span>
            @endif
        </div>
    </div>
    <input type="hidden" name="usertype" value="1"><!-- 隐藏域代表用户类型-->
    <div class="form-group">
        <div class="col-md-6 col-md-offset-4">
            <div class="checkbox">
                <label><input type="checkbox" name="remember" {{ old('remember') ? 'checked' : '' }}> 记住我</label><br>
                <span class="error hide text-danger">用户类型不匹配,重新输入用户名</span>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-8 col-md-offset-4"><!-- 登录按钮-->
            <button type="submit" id="login" class="btn btn-primary">
                登录
            </button>
        </div>
    </div>
</form>
```

图 6.2 登录页面代码

页面展示效果见图 6.3:



图 6.3 登录界面

## 6.2 列表管理模块

列表管理是展示批量信息时用到的组件。用户在搜索到大量相关信息之后不可能完全阅读全部内容。他们一般会更关注信息的标题部分从而进一步筛选信息。本项目中的列表包括职位信息列表、招聘信息列表、宣讲信息列表、简历列表、交流消息列表、投递信息列表。

举例说明获取公司列表数组的代码。如图 6.4 所示，在这个方法中，通过\$companies 变量查找到公司信息的分页列表，之后将数据发送到 student/company\_info\_list 页面显示。

```
//获取公司列表
public function get_companies_list()
{
    //从Company表中获取所有记录并且进行分页，每页显示10条数据
    $companies = Company::paginate(10);
    //返回列表视图，并且将获取的参数传递到前端
    return view('student.company_info_list')->with('companies', $companies);
}
```

图 6.4 获取列表数据代码

页面展示代码如图 6.5 所示，在该页面中，用逐元循环来遍历后台发送的公司信息，并且分页显示。

```
<div class="list-group">
    @foreach( $companies as $company ) <!-- 循环遍历整个数组，逐条显示每一个对象 -->

        <div class="list-group-item">
            <div class="row">
                <div class="col-md-3">
                    <h4 class="list-group-item-heading">
                        <a href="/guest/goto_company_info/{{ $company->id }}">{{ $company->name }}</a>
                    </h4>
                    @if( $company->auth == 1 )
                        <span class="badge"> {{ CompanyInfo::getAuth($company->auth) }} </span>
                    @endif

                </div>
                <div class="col-md-5">

                    <h4>
                        <i class="icon icon-map-marker"></i>
                        {{ $company->city }}
                    </h4>
                </div>
                <div class="col-md-2">
                    <h4><span class=" badge">{{ $company->focus }}</span>人</h4>关注
                </div>
            </div>

            <div class="row">
                <div class="col-md-3">
                    <p class="list-group-item-text">
                        规模: {{ CompanyInfo::getScale($company->scale) }}
                    </p>
                </div>
            </div>
        </div>
    @endforeach

    {{ $companies->links() }} <!-- 显示页面跳转链接 -->
</div>
```

图 6.5 列表展示代码

展示效果见图 6.6 所示：



校园招聘 Campus Recruit

我的简历

公司活动

输入职位信息

搜索

找工作

消息动态 2

聊天室

naplus

有15家公司符合要求

湖南计算机科技销售部

已认证

长沙市

537 人

关注

规模：200~500人

广东计算机技术销售部

重庆市

3230 人

关注

规模：500~2000人

四川计算机管理科研所

济南市

2332 人

关注

规模：50人以下

陕西计算机制造研发中心

成都市

2932 人

关注

规模：50人以下

河北计算机批发有限责任公司

已认证

杭州市

2487 人

关注

规模：500~2000人

陕西计算机科技开发岗

已认证

郑州市

377 人

关注

规模：50人以下

海南计算机处理开发岗

已认证

长沙市

3686 人

关注

规模：50人以下

图 6.6 列表展示效果

### 6.3 内容展示模块

内容展示模块的主要功能是为用户呈现每个列表项中具体内容。它的信息量比列表大，能够提供给用户需求的信息量。用户可以通过点击列表进入该模块查看其详细信息。内容模块包括公司说明、职位具体信息、招聘会详情、宣讲会安排、简历条目、消息详情等。

公司内容展示部分代码如图 6.7 所示，其中\$company\_info 是具体对象，双花括号是blade 模板的书写方式。通过\$company\_info 的各个属性将整个公司的信息呈现给用户。

```
<div class="row">
  <div class="col-md-3">
    <div class="panel panel-warning">
      <div class="panel-heading">
        <h3>
          <i class="icon icon-group icon-lg"></i>
          {{ CompanyInfo::getScale($company_info->scale) }}
        </h3>
      </div>
      <div class="panel-footer text-center">
        <h4>公司规模</h4>
      </div>
    </div>
  </div>
  <div class="col-md-3">
    <div class="panel panel-danger">
      <div class="panel-heading">
        <h3>
          <i class="icon icon-tags icon-lg"></i>
          {{ CompanyInfo::getCompanyProperty($company_info->property) }}
        </h3>
      </div>
      <div class="panel-footer text-center">
        <h4>公司性质</h4>
      </div>
    </div>
  </div>
  <div class="col-md-3">
    <div class="panel panel-info">
      <div class="panel-heading">
        <h2>
          <i class="icon icon-calendar icon-lg"></i>
          {{ $company_info->job_number }}
        </h2>
      </div>
      <div class="panel-footer text-center">
        <h4>职位数目</h4>
      </div>
    </div>
  </div>
</div>
```

图 6.7 内容展示部分代码

页面展示效果如图 6.8 所示:



图 6.8 内容展示

6.4 表单提交模块

表单提交模块是用户与系统交互的重要手段。大量的数据是通过填写表单来提交给系统，然后存入数据库中。本项目中各种表单主要用于信息提交并更新数据。

表单展示代码如图 6.9 所示；

```
<form id="form_award" name="form_award" action="/student/post_resume_award" method="POST">
  <input type="hidden" name="_token" value="{{csrf_token()}}"/>
  <input type="hidden" name="rid" value="{{ $rid }}">
  <input type="hidden" name="method" value="{{ $method }}">
  <input type="hidden" name="id" value="{{ $id }}">
  <div class="row">
    <div class="col-md-4 col-md-offset-2">
      <div class="form-group">
        <label for="name">奖项名称</label>
        <input type="text" class="form-control" id="name" name="name" value="{{ $award->name or '' }}"
          placeholder="请输入奖项名称" required>
      </div>
    </div>
    <div class="col-md-4 col-md-offset-0">
      <div class="form-group">
        <label for="level">等级</label>
        <select class="form-control" id="level" name="level">
          <option value="0">--请选择</option>
          <option value="1">国家级</option>
          <option value="2">省级</option>
          <option value="3">校级</option>
        </select>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-8 col-md-offset-2">
      <div class="form-group">
        <label for="time">获奖时间</label>
        <input type="text" class="form-control" id="time" name="time" placeholder="获奖时间" required>
      </div>
      <div class="form-group">
        <label for="details">获奖内容</label>
        <textarea class="form-control" id="details" name="details" rows="5" placeholder="描述获奖内容"
          required="{{ $award->details or '' }}"></textarea>
      </div>
      <button type="submit" class="btn btn-primary" id="submit_award" name="submit_award">确认</button>
    </div>
  </div>
</form>
```

图 6.9 表单提交页面代码

展示效果如图 6.10:



图 6.10 表单提交页面

## 6.5 异步通讯模块

异步通讯模块是一项提升系统响应性能的关键技术。在客户端与服务器端忙碌传递信息而无需重载页面，用户几乎感觉不到数据传送的存在。用到此项技术的部分包括消息的加载、关注信息。

本项目在学生关注公司和收藏职位时进行了 Ajax 的应用，当用户点击关注或收藏按钮时，系统会在后台将数据发送给服务器处理，然后改变按钮的状态以提示数据提交成功。在加载用户新消息时也会用到 Ajax 技术，在每次新加载页面的时候在后台将用户收到的新消息加载到系统前端在导航栏出显示。

通信模块部分代码示例如图 6.11 和图 6.12 所示：

```
//点击关注按钮之后的Ajax操作
function followUp()
{
    $.get('/student/follow/' + $('#cid').val(), //url地址
        {},
        function(data) //通信成功之后的回调函数，进行前端样式调整
        {
            $('#follow_number').text(data.follow_number); //关注数目
            $('#isFollow').text('已关注'); //关注信息
            $('#follow').removeClass('btn-success');
            $('#follow').addClass('btn-default');
            $('#follow').val('follow');
            $('#f_value').text('取消关注');
        }
    );
}
```

图 6.11 Ajax 部分前端代码

```
//关注该公司 ,公司ID ,AJAX
public function follow($cid) //参数为公司的ID
{
    $senderId = Auth::user()->id; //发送者ID
    $receiverId = Company::find($cid)->owner->id; //接收者ID

    $content = ' ';
    $content .= Auth::user()->name . '关注了贵公司';

    if(Follow::isExist($senderId, $receiverId))
        return 'not add';
    Follow::store($senderId, $receiverId); //存储一条关注信息
    $follow_number = Company::followUp($cid); //计数+1

    Information::newInfo($senderId, $receiverId, $content); //生成新消息

    $data = array('follow_number' => $follow_number);
    return $data; //将数据打包发送给前端
}
```

图 6.12 Ajax 部分后端代码

### 6.6 消息管理模块

一个系统的交互性是评价系统功能的重要指标，用户总是希望系统能够对自己的操作做出反馈，一个没有反馈的系统是不友好的。在本系统中，用户之间的交互式通过消息管理模块实现的，主要借助数据库中的 information、collections、follows 和 delivers 表来存储响应的信息。

页面加载完后加载消息代码如图 6.13 所示：

```
<script type="text/javascript">
$(document).ready(function() {
    getNewInfos(); // 获取新消息
});

function getNewInfos()
{
    $.get(
        '/student/get_new_infos/{{$auth->id }}',
        {},
        function(data)
        {
            var str = '';
            if( data.length > 0 )
            {
                $('#info_number').text(data.length);
                for(var i = 0 ; i < data.length ; i++)
                {
                    str += '<li class="list-group-item info-list"><a href="">' + data[i].content + '</a></li>';
                    str += '<li class="list-group-item text-center"><a href="/student/get_unread_info_list">查看全部消息</a></li>';
                }
            }
            else
            {
                str += '<li class="list-group-item text-center">暂无新消息<a href="/student/get_unread_info_list">查看历史消息</a></li>';
            }

            $('#nav-info-list').html(str);
        }
    );
}
</script>
```

图 6.13 加载消息代码

页面展示如图 6.14 所示：



图 6.14 动态消息页面

## 6.7 在线交流模块

在线交流需要用到 HTML5 的新标准，在这个部分用到 socket.io 的组件库对 WebSocket 的封装。该组件会在服务器端进行监听，当收到来自客户端发送的消息时，就通知响应消息处理程序进行广播消息，用以通知其它用户接收显示消息，进而实现多人聊天的功能。

群聊效果如下图 6.15 所示：

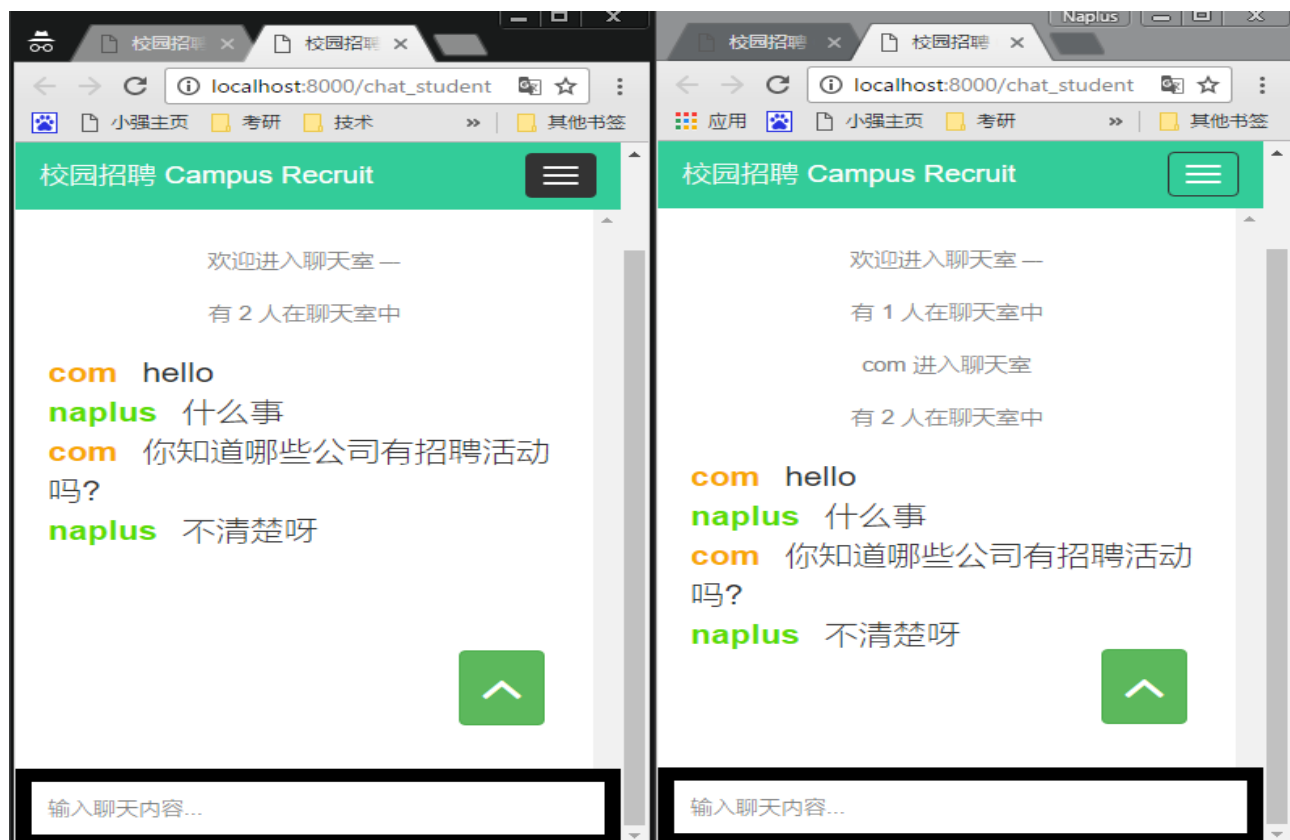


图 6.15 聊天效果显示

实现代码如图 6.15 所示, 通过代码 `$web = new WebServer('http://0.0.0.0:2022');`, 服务器在 2022 端口监听来自客户端的连接消息, socketIO 在 2020 端口处理请求。来自客户端的数据有用户名和消息内容。

```
$io = new SocketIO(2020);
$io->on('connection', function($socket){
    $socket->addedUser = false;

    // when the client emits 'new message', this listens and executes
    // 当客户端发出'new message', 此监听器监听并执行
    $socket->on('new message', function ($data) use($socket){
        // we tell the client to execute 'new message'
        $socket->broadcast->emit('new message', array(
            'username'=> $socket->username,
            'message'=> $data
        ));
    });

    // when the client emits 'add user', this listens and executes
    $socket->on('add user', function ($username) use($socket){
        global $usernames, $numUsers;
        // we store the username in the socket session for this client
        $socket->username = $username;
        // add the client's username to the global list
        $usernames[$username] = $username;
        ++$numUsers;
        $socket->addedUser = true;
        $socket->emit('login', array(
            'numUsers' => $numUsers
        ));
        // echo globally (all clients) that a person has connected
        $socket->broadcast->emit('user joined', array(
            'username' => $socket->username,
            'numUsers' => $numUsers
        ));
    });

    // when the client emits 'typing', we broadcast it to others
    $socket->on('typing', function () use($socket) {
        $socket->broadcast->emit('typing', array(
            'username' => $socket->username
        ));
    });
});
```

图 6.16 聊天效果显示



## 6.8 上传下载模块

本项目中涉及到文件上传的地方有三处，一处是用户头像的上传，一处是公司认证材料的上传，另外一处就是学生在简历编辑时上传的附件信息。在完成这些文件的上传时，我用到了 Larval 框架自己封装的代码，实现起来较为方便，将上传之后的文件存放在 /public/uploads/ 目录下，数据库中存放着上传文件的名称和对应的存放地址以便前台调用显示。上传的业务逻辑代码如图 6.17 所示：

```
//上传头像
public function profile_avatar_post(Request $request)
{
    $path = Auth::user()->avatar;
    Storage::delete($path); //删除旧文件
    //保存头像，存放地址为/uploads/images/ 目录下
    $path = $request->file('avatar')->store('/images');

    $user = Auth::user(); //更新数据库信息
    $user->avatar = $path;
    $user->save();
    return redirect()->intended('home'); //返回用户主页
}
```

图 6.17 上传处理逻辑

上传头像的页面展示如图 6.18 所示

上传头像

照片

Choose files To Upload

Choose Files

提交

图 6.18 上传头像页面

## 结 论

在这次毕业项目的设计和开发中，对自己的知识的巩固和能力的提升有很大帮助。

此次校园招聘平台的设计总体上完成了预期目标，实现了其基本功能。该系统可以同时为学生和招聘单位提供服务，学生能够在该系统上搜索和浏览相应的公司职位信息；公司也可以在系统上发布自己的招聘信息，与学生相互交流；同时，管理员可以在系统上验证注册的公司和发布招聘会等操作。

我在这次系统开发中受益颇多，不但巩固了自己所学习的 WEB 开发的技能如 HTML、PHP、JavaScript 和 CSS，而且在开发的过程中学习了不少新的技术，如采用 Ajax 异步通讯请求技术来实现页面无刷新的更新；采用 WebSocket 技术来实现在线聊天室的功能；引用 bootstrap 插件来实现具有丰富交互特性的动态前端和友好视图的丰富界面。

经过这次的开发，我懂得了要完成一个完整的项目，需要做好与 BUG 作斗争的准备。面对困难时要迎难而上，做出成绩时也要谦虚谨慎。

作为一名即将毕业的本科生，经过毕业设计的锻炼，我发现了自己的不足和优势。不足之处在于缺乏经验，对技术的应用和扩展比较生疏。优点在于勤奋好学，能够快速入门。我相信自己有能力掌握和应用自己所学习的专业知识和技能。在今后的学习和工作生活之中，我都会吸取这次设计的教训和经验，在发展中取长补短，适应这个技术不断变化革新的社会。

## 致 谢

时间如白驹过隙，从开始选择毕业设计选题到完成毕业论文的撰写完成，我的本科毕业设计总共历时约六个月。这期间前半期因为考研的耽搁，是在导师的帮助下完成了选题和开题报告的撰写，后期开始忙碌的代码设计和编写阶段，同学经常帮助我查看代码中的漏洞，老师很负责地给我提供意见和建议。在不懈努力下终于要迎来最后的曙光了，在此十分感谢帮助过我的同学和老师。

在毕业设计的开发期间，对我帮助最大的我的导师王晓峰老师。她总是很负责地为我每个阶段的工作作出详细地计划，按时检查我的工作进度，并且不厌其烦地为我在开发过程中所遇到的问题提供宝贵的建议。在老师的帮助下，我的开发过程比较轻松，在这里对王老师表达我衷心地感谢。

## 参 考 文 献

- [1]陈昊等. Laravel 框架关键技术解析[M]. 北京: 电子工业出版社, 2016.
- [2]Cristian Darie 等. Ajax 与 PHP Web 开发[M]. 北京: 人民邮电出版社, 2007
- [3]Russ Unger 等. UX 设计之道[M]. 北京: 人民邮电出版社, 2010
- [4]Brett McLaughlin 等. 深入浅出 Ajax[M]. 北京: 东南大学出版社, 2008
- [5]Robert Daigneau 等. 服务设计模式[M]. 北京: 机械工业出版社, 2013
- [6]Andy Clarke 等. 超越 CSS:Web 设计艺术精髓[M]. 北京:人民邮电出版社, 2012
- [7]Eric Freeman 等. Head First HTML5 Programming[M]. 北京:中国电力出版社, 2012
- [8]Jay Bryant 等. 高性能 HTML5[M]. 北京:电子工业出版社, 2014
- [9]Benjamin LaGrone 等. 响应式 Web 设计[M]. 北京:机械工业出版社, 2014
- [10]Ralph Steyer 等. jQuery 应用开发[M]. 北京:机械工业出版社, 2014
- [11]William Sanders 等. PHP 设计模式[M]. 北京:中国电力出版社, 2014
- [12]Vijay Joshi 等. PHP 与 jQuery 开发实例[M]. 北京:人民邮电出版社, 2011
- [13]Josh Lockhart 等. Modern PHP[M]. 北京: 中国电力出版社, 2015
- [14]Peter B. MacIntyre 等. PHP 语言精粹[M]. 北京:电子工业出版社, 2012
- [15]Matt Zandstra 等. 深入 PHP 面向对象、模式与实践[M]. 北京: 人民邮电出版社, 2011