编号：



毕业设计说明书

题 目：**基于HTML5的培训机构**

**排课系统的设计与实现**

学 院：计算机与信息安全学院

专 业： 软件工程

学生姓名： 梁艺可

学 号： 1600300522

指导教师： 潘海玉

职 称： 副教授

**题目类型：**  理论研究 实验研究 工程设计 工程技术研究 ☑软件开发

2020年 05月 05日

摘 要

新中国成立以后，人民生活水平等各方面的质量都有了很大的改善。加之教育一直以来都是被中国人民所注重的，因此也越来越多的人接受教育，越来越多的人进入教育行业，教学课程也日益变得多样化，各方面资源数量的快速增长，也让传统的手工排课再也不太适用于当下的教育环境，即使可以通过人工去排课也会导致这其中付出很多的精力，工作量巨大，况且很多时候还需要考虑个别教师的上课喜好，尽量人性化。为了解决人工排课的困难，本文通过对人工智能领域的遗传算法进行研究分析，开发出了一套基于该算法的培训机构的排课系统，该系统针对于高中阶段教育的培训机构，学校教务处方面首先需要给某一个需要开设课程的学期分配好课程计划，即：年级编号，班级编号，教师编号，课程编号，上课周数，每周课时数，课程是否固定上课时间，课程属性等基本信息；在应用到算法实现上的时候，即可将课程计划的各项信息，即以上的信息编号进行编码，编码规则为：是否固定上课时间+年级编号+班级编号+教师编号+课程编号+课程属性+上课时间进行编码，即字符串的拼接操作，在完成编码以后即可得到初始的基因编码，这些基因编码的集合，我们将其看作初始种群，每一条基因编码就看作是种群中的一个个体，随后将每一个初始基因编码为其随机分配一个时间片，再将这些个体经过不断地进行交叉、变异操作，这两步都是对时间编码进行操作，到达指定遗传代数后算法结束，随后开始给这些经过变异操作的基因编码分配教室，判断“时间-教室”条件(即同一时间同一教室只能上一门课)不冲突最终即可得到每一门课程的上课时间以及上课所在的教室。系统采用前后端分离的模式进行开发，技术实现上，使用了当前最为流行的Spring框架进行开发，前端使用了Vue框架整合Element UI组件实现，利用Npm包管理工具实现对前端项目的依赖进行管理；后端使用SpringBoot框架搭建，编程语言使用Java1.8，数据库为MySQL 8.x版本；目前，很多排课软件都是采用了C/S架构(即客户端/服务器)进行开发，这样的话只有安装了其客户端才可以使用；本次为了用户更方便采用了B/S架构(即浏览器/服务器)，用户只需要一个浏览器即可使用该系统，带来了极大的便利。

**关键词**：遗传算法；排课；基因；个体；种群；编码；培训机构

**Abstract**

Since the founding of New China, the quality of people's living standards and other aspects have greatly improved. In addition, education has always been valued by the Chinese people, so more and more people are receiving education, more and more people are entering the education industry, and teaching courses are becoming more and more diverse. The rapid increase in the number of resources in all aspects makes traditional manual courses less suitable for the current educational environment. Even if it is possible to arrange lessons manually, it will result in a lot of energy and huge workload; moreover, it is often necessary to consider the preferences of individual teachers in class and implement humanized teaching as much as possible. In order to solve the difficulty of manual scheduling, this paper has developed a set of scheduling system for training institutions based on the algorithm through research and analysis of genetic algorithms in the field of artificial intelligence. This system is aimed at training institutions for high school education. The specific method is that the school's Academic Affairs Office first needs to assign a course plan to a semester that needs to open a course, that is: grade number, class number, teacher number, course number, class week, class hours per week, and whether the course is fixed. Basic information such as course attributes; when applied to the implementation of the algorithm, you can encode the information number of the information of the above course plan; the coding rule is: whether the class time is fixed + grade number + class number + teacher number + Course number + Course attribute + Class time encoding, that is, string concatenation operation, you can get the initial gene code after completing the coding; these gene code collection, we regard it as the initial population, each gene code is Seen as an individual in the population, then each initial gene encoding was randomly assigned to a time slice, and then these individuals were continuously subjected to crossover and mutation operations. These two steps are operations on time encoding to reach the designated The algorithm ends after genetic algebra. Afterwards, we began to assign classrooms to these genetically modified genetic codes to judge the "time-classroom" conditions (that is, only one course can be taken in the same classroom at the same time) without conflict. Eventually, we can get the class time of each course and the class classroom. The system is developed using a separate front-end and back-end mode. For technical implementation, the most popular Spring framework is used for development. The front-end uses the Vue framework and Element UI components for implementation. The Npm package management tool is used to rely on front-end projects. Management; the back end is built using the SpringBoot framework, the programming language is Java, the version is JDK1.8, and the database is MySQL 8.x version; at present, many scheduling software is using C / S architecture (ie client / server). Development, in this case, only the client can be used; this time, for the convenience of users, the B / S architecture (that is, browser / server) is adopted. The user only needs a browser to use the system, which gives the user Brought great convenience.

**Key words:** genetic algorithm；class scheduling；gene；individual；population；coding；training institution

目 录

# 引言

## 选题背景

20世纪中后期到现在的21世纪，我国在各方面的成绩突飞猛进，尤其是在经济发展水平和科学技术上，取得了世界瞩目的成就，这也直接使得国民生活水平质量得到了巨大的提升，同时中国社会人才短缺的问题也逐渐暴露出来，对人才的需求实在是迫在眉睫，国民对教育的重要性有了新的认识，意识到了教育的重要性，只有加大对教育的投资，加强对教育的重视才可以弥补职业教育不足带来的人才短缺问题。加之20世纪中后期我国提出的“科教兴国”，“人才强国”战略的提出，更是凸显出了国家对于教育的重视程度以及人才需求的迫切性，因而在这之后，我国在教育教学工作上也是做出了巨大的投入，这也随之促使了教育教学等资源的极速增长，一方面，意识形态的改变，让许多家长非常重视其子女的教育，甚至家长本身也对自己知识水平有了更高的要求，另一方面，社会发展需要更多的人才，“人才强国”战略也督促着人们想要在社会上有着比较好的成就，能够有更好的生活，也必须学习，必须接受教育，毕竟传统里“知识改变命运”也是不变的道理。基于以上各种因素，我国从那之后接受教育的人数、群体都呈现出只增不减的状态，国家加大对教育的投资，对学生的各种有利政策，学校扩招，除了公立学校之外，越来越多的人也参与到了教育行业中，为人才建设贡献一份力量，例如许多私立学校的创办，培训机构的设立都是教育行业蒸蒸日上的真实写照，教育教学资源的多元化，教育机构的大规模化，也寓意着在教育教学开展中也添加了更多的元素。另外，家长们为了提高孩子们的成绩，也更多地让自己的子女参加课外的一些培训课程，在这样的状况下，开设的课程必然是越来越多的，由此而来，教学安排中的排课事项肯定是不能像传统那样子通过人工手动去排课的，人工手动排课必将花费大力的人力物力；如今科学技术如此发达，计算机发展也相当可观，硬件成本也不像以前那么昂贵，计算机的处理能力、运算能力空前强大，我们应该合理地利用计算机去解决生产中的问题，本文我们将通过计算机编程去实现用遗传算法解决排课问题，大力减少了人力物力的损耗，提高了工作效率。

## 发展状况

上个世纪50年代末人们就已经开始着手对课程编排问题进行了研究并尝试在计算机上解决该问题，针对该问题的解决方案，他们也曾在解决排课问题的数学模型，是否存在对排课问题的解等方面上进行了许多的研究，但都没有很好的解决方案；直到上个世纪70年代课程编排问题被证明是一个NP完全问题，也就是说算法的运算时间是呈现出指数增长的样式的，在数学领域是没有一个通用的算法能够很好地解决该问题的，从此人们才开始将视线从数学模型的研究转移到解决该问题的实用算法探索上。

早在最开始的时候，人们就是通过在黑板上拿着笔去填写表格，边填写格子边观看上面的课程安排，不合理的地方就擦掉重新安排的，直到设定的课程都能够安排上为止，这要求编排者需要有非常严谨的思维能力，这样子的方式也只能适用于待安排的课程比较少的情况下，才可以在不耗费大功夫的情况下通过人工去手动编排，否则将会给人带来巨大的困难，导致人力物力等成本的增加，降低工作效率。

到了20世纪80年代的时候，随着计算机得到发展以及普及，许多人也能运用上这一拥有强大计算能力的机器，达到这个时候，人们在此之前已经对算法有了许多的了解，例如常用与解决问题的贪心算法、分治算法、动态规划法；因而人们这时候也想到了尝试通过使用计算机编程算法去解决课程安排问题，以让教育工作者提升工作效率，这个阶段，人们通常采用的是贪心算法、分治算法以及回溯算法进行处理排课问题完成排课工作。这几种常用的算法也在排课实现中有着不错的表现，但是它们也有着自己的缺点，例如贪心算法虽然没有迭代，时间复杂度相对较低，但是其贪心策略也常常导致只能得到局部最优解而非全局最优解；回溯算法时间复杂度会随着问题规模的增长而快速增大，因为该算法需要不断地进行迭代。

到了近些年，科技进一步发展，计算机领域的发展更是惊人，人工智能这一名词也常常闻于耳，许多算法也在人工智能领域被深度学习等方式进一步优化，例如本次我们使用的遗传算法，该算法在20世纪70年代就已经被提出，遗传算法是根据达尔文的进化论得来，即是模拟生物进化，适者生存，不适者淘汰的自然规律的一种全局搜索最优解算法，原理就是利用数学的方式，通过计算机去仿真模拟生物在进化过程中出现的选择、交叉、变异再到淘汰的过程，不断地进行代数的遗传，达到终止条件时算法就会结束。其在解决组合优化问题上表现出了优异的成绩，因而也一直被人们所喜爱，发展至今，遗传算法已经可以很好地处理课程安排问题，我们这次也毅然决定使用该算法进行尝试实现排课系统。

## 行业前景

之前我们所述，已经知道了随着国家综合实力的增强，重视教育已经促进了各式各样的教育机构遍地开花，许多研究数据也表明排课软件所占的市场份额到了2019年已经超过了亿元级别大关，从这些数据中我们应该也可以看到，在未来教育行业更加蓬勃的时代，人们唯有通过排课软件去提升课程安排工作的效率，因而排课软件在未来的市场份额不但一点也不比现在差，反而可能会出现比较大幅度的增长趋势。

## 研究意义

通过对排课系统的研究、设计与实现，利用计算机去解决传统中手工完成的工作，将传统通过人工手动去处理的计算问题交由每秒钟计算次数以亿为单位计量的处理器去处理，这无疑在很大程度上提升了教育工作者的工作效率,摆脱了传统手工操作给人带来的束缚，提升了用户体验也促进了行业的发展以及加快了行业的发展速度。

## 本文组织结构

本文先通过摘要对文章总体做一个概括，摘要之后有相关的关键词，让读者对文章的描述内容以及条理也比较清晰，从中也知道了本文是关于排课问题的研究；随后通过引言引出了本文的相关背景，通过背景的描述，教育教学资源的增加，科技的进步，让我们有了更好的办法去解决排课问题而不是停留在传统的人工去实现排课，从而得到本文的编写的缘由；再而对排课研究的发展状况进行了一个简要的描述，从而了解到前人在排课问题上所作的努力及其取得的成果，再而道出了本文所进行的研究的意义。

接下来开始分析本文所研究的排课系统在经济、政策、技术上的可行性；在可行性通过之后开始对系统进行一个比较常规的需求分析，这也是软件开发中所必须经历的流程；在明确我们的需求之后，那么就大体上知道我们要做什么了，那么怎么做呢？那么先来个大概要怎么做，此时，我们就需要进入到了总体设计，总体上有一个框架了，接下来开始对系统进行详细的设计了；有了设计的思路了，还需要编码实现，因而有了我们接下来的系统实现，这也是排课系统中比较关键的部分；系统实现了之后呢，需要对系统的功能等各方面进行一个测试，最起码的就是对功能进行测试，只有需求实现了，再谈及其它性能方面的内容才是正确的。最后得到本文研究的结论；致谢，向指导老师以及参考资料的作者表示感谢。

# 可行性分析

## 经济可行性

前期分析，针对排课算法实现排课系统的这一需求，一方面，在软件开发过程中所需成本不高，目前作为个人开发者，只需要拥有自己的计算机即可，在计算机上安装好所需的开发环境，另外计算机本身在开发阶段也可以作为部署项目的服务器，如果后续需要使用到专业的服务器，针对于该排课系统，并发量极少，因而说在服务器费用支出部分也不会需要过多的费用，因为租赁相对较为便宜的服务器即可运行该系统并完成常规下的任务。另一方面，前期系统由单人进行开发，不需要额外请人导致雇佣费用的支出，在人力成本上也并非很高，都是在可以接受的范围内有序进行开发。

后期分析如果后续系统涉及到商业应用，那么对于此情况下，整体上来说，需要对该系统进行定制化的改造，因为通常来说，针对不同的教育机构，不同的学校，其教学生态体系都是有着不少的区别的，因而针对他们的模式也常常需要进行定制，或者说在系统的原有基础上进行功能的扩展，使得系统可以应对多种生态的教学模式，针对大部分的情况都可以进行排课，满足学校的课程安排问题，如果在这方面做得到，推广得好，使用效果，实用性，可靠性等方面都基本上可以满足学校的需求，我相信这套软件系统也可以依靠其可靠性取得比较良好的经济效益的。

综合以上分析，即使软件不进行商业化，其开发所花费的经济成本也不高，都是在一个人可以承受的范围内；而若进行商业化并得到一定的青睐，那么软件系统也可以带来经济收入，那么相对于前期的支出，软件系统是可以获得收益的。

## 社会可行性

这套基于遗传算法编程实现的排课系统是个人通过Java自主编程实现，在当今社会的教育发展现状下，也恰巧是需要更多这么实用性的软件系统的，况且当下环境中，虽然有很多教育机构都有他们自己的定制软件，亦或者说有很多专注排课软件研发的公司，针对不同的教学生态、教学模式下研发的不同的软件，而后根据不同机构的需求将其软件授权给这些机构使用，但是这授权费用也是相对较高高，这也通常导致很多小型的机构，创业型机构对于这笔费用的支出处于犹豫状态；而我们提供的排课系统前期完全免费并开源，可以实现二次开发，实现定制化。

对于社会法律、政策等条件，我们的软件也是完全没有任何敏感的元素会触犯相关的法律法规或者是政策条文，反而在当下的教育行业蒸蒸日上的社会环境中，有着推动行业发展的作用，虽然其作用很微小。但其更大的作用还是给教育工作者带来了很大的便利，大大提高了教育机构安排课程的效率，让教育教学工作进行得更加有条不紊。

在社会推广方面，软件本身得益于其实用性以及高效性，也将不但不触犯任何法律条文，反而其可靠性、实用性也将会得到人们的青睐；软件由于其自主性也不惧怕任何侵权相关的问题。

综合以上各方面的社会因素分析，排课系统的开发在社会可行性上行得通。

## 技术可行性

该排课系统使用了当前软件行业中比较主流的技术架构，即B/S架构，也就是浏览器/服务器架构，使用这种架构极大地提升了用户体验，不再像传统那样需要使用某一个系统还需要安装一个客户端(Client)才可以，如今不管是手机还是计算机都有浏览器这个软件，只要有浏览器我们就可以使用系统。

系统使用了摘要中所述的框架进行搭建，这些框架可以快速成型项目结构，不需要像传统的项目一样写很多繁琐的配置文件，可以简化开发，提高开发效率，在计划的时间内可以完成软件系统的编写。

系统在主流框架的整合下，变得更加容易维护，相对来说系统的稳定性是很强的，但是系统稳定运行的条件通常取决于其所部署的环境，因此我们只要选择比较稳定的服务器，并且选择的服务器能支撑多数情况下系统的并发量，通常来说系统也是稳定运行；软件质量上，在规范编码的加持下，系统的实现代码逻辑也有条有理，符合逻辑，并且在系统完成后也进行了很多的测试，因此软件质量是比较可靠的。

之前所提到的，在当下社会科技比较发达的情况下，计算机硬件的配置也相当高，价格也比较便宜，人们通常也可以支付得起这一笔费用，因而说只要软件运行了，那么也无法吃满当下常见的计算机资源，由于代码的执行效率高，因此在实际生产环境中，该系统可以完全可以应对人们的需求，给人以良好的体验，生产效率也相当优秀。

综上所述，系统开发技术可行。

# 需求分析

## 任务概述

### 目标

科技发展，社会进步以及“科教兴国、人才强国”战略的推动作用，使得如今的教育教学资源直线上升，在课程安排的工作上也因为课程增加，个性化条件增加带来了许多的排课困难，不再像传统那样可以通过人工手动去实现课程编排，即使可以，那么也将付出巨大的代价。

因此我们本次需要研发出来一款软件来实现排课工作，帮助教育工作者提高工作效率，不让他们在课程安排工作中花费太多本来不应该花费的时间。软件系统的最基本功能应该是能够实现排课，这也是最核心的部分。其次我们希望该系统可以实现信息管理功能，比如知道学校学生的一些基本信息以及对这些信息的修改，这些信息管理包含教室管理，讲师管理，学生管理，课程管理，以上为必须实现的基本需求；除此之外我们系统仍可以在开发过程中根据实际情况进行相应地添加一些实用的功能。

### 假定和约束

对该系统的开发而言，暂时只对其做出相应的约束即可，限定系统的开发语言为Java 1.8；使用MySQL 8.x的数据库；后端框架采用当前主流的Spring Boot框架；前端通过Element UI组件与Vue框架整合搭建前端页面，如果有文件需要云存储则选择七牛云或者阿里云的云空间来存储；数据库持久层框架使用Mybatis-Plus，其与Spring Boot框架将可以完美地结合使用，实现对数据库的操作；数据可视化使用百度官方的Echarts实现对系统数据的查看；Excel文件的导入导出使用Easypoi实现。

### 用户特点

该系统的目标用户为培训机构中的师生，预设的用户为学生、讲师、教务处主任，这些用户对于计算机软件了解程度通常来说比较低，他们所关注的层面只是如何去使用该系统来满足他们的需求。当然也可以用于其它教学生态与系统设定的符合的教育机构，即使如此，系统的用户依然是属于教育工作者以及学生。

## 需求规定

### 对功能的规定-未完成

系统应该拥有起码3种用户，学生、讲师和管理员(教务处主任)，教务处主任拥有最高权限，拥有课程管理、学生管理、排课管理、教学设施管理、教材管理；讲师拥有学生管理、教学管理、查看自己的上课计划以及部分课程管理的功能；学生拥有查看课表，查看作业这些权限。

系统经过权限分明限制系统用户的权限后，用户无法越过自己的权力范围去操作别的数据，这样可在很大幅度上增强系统的安全性，最好还可以通过加密算法对用户的密码加密后再存入数据库中。

任何用户在未登录的情况下都应该无法进入系统，学生通过自行注册申请账号并以学号作为账户登录进入系统；管理员账号由系统工程师提前分配好，在这之后管理员可凭自己的账号登录系统；讲师账号经由管理员创建，因为讲师注册流程需要提交相关的资质材料给校方相关人员审核通过之后再创建账户给讲师，此后讲师才拥有系统的账户。

在系统管理员端可以现实系统的数据方便管理员查看，作为可视化的一部分，系统数据展示也将会提升用户体验。

在应用软件支持上，尽可能地兼容老版本的浏览器，以应付部分机构设备落后带来的尴尬局面。

### 对非功能的规定

当下IT行高速发展，很多应用软件系统已经不再像过去那样界面丑陋，因为有很多开源的项目使得如今项目开发过程中可以选择的UI元素更多，因而一方面要求系统的UI界面美观简洁，能直观体现出每一个功能模块及其功能。在系统界面足够美观的情况下，用户使用的感受也会良好许多；另一方面，系统的点击响应流畅不会发生卡顿，用户在使用过程中若系统出现小差错应该以人性化的提示展示给用户。

### 对性能的规定

当前使用的技术栈下，系统的稳定性更多地取决于系统所交付的机构方使用的服务器，服务器性能越好，当然是越能够体现出系统的流畅性，但该软件也基本上是每一家教育机构自己运行在自己购买的服务器下的，因此，系统的并发量也不会太高，因此这些机构并不需要购买很贵的服务器，只要系统能稳定运行即可。

而若是系统部署在我方，用户对象为多家教育机构，此时性能的考量就需要评估每一家机构可能会出现访问系统时候的并发量再进行服务器配置的评估；当然也可以根据实际需求等常常无法满足使用需求的时候再进行增购服务器以实现性能的提升。

唯有拥有了足够的带宽，配置足够的服务器来部署系统，才可以在请求响应时间上做到用户满意；此外，在系统安全上，也需要在服务器端进行高级的安全等级设置，以保证系统不被非法入侵，从而导致客户方的数据出现安全隐患。

## 系统层次方框图

## 系统用例图

## 运行设备规定

### 设备

系统需要部署在服务器上运行之后才可以给用户正常访问使用，服务器可以选择云服务器或者本地服务器；部署在Linux系统服务器上，也可以直接部署在人们常用的Windows系统的服务器上，只需要服务器上安装了相关的运行环境并做相关的部署配置即可。

### 支持的软件

如今每一台计算机或者手机都有一个上网浏览器，我们只需要浏览器即可使用该软件系统，这也是B / S架构开发下的便利之处，而不再需要再去单独安装一个软件使用系统。此外，为了兼容更多老版本浏览器，系统开发过程中也做出了努力，能支持到老版本的IE6，6以下不支持。

### 控制

系统部署后，日常维护还是需要工程师帮助的，工程师通过终端工具连接到所在服务器即可实现对项目的维护操作，数据库也部署在了项目所在的服务器，工程师也可以直接在当前服务器对数据库进行操作；用户端只关注系统的使用。

# 总体设计

## 系统全局层次图

## 系统模块层次图

## 系统包图

## 系统类图

## 系统配置图

## 接口设计

### 学生接口

### 管理员接口

### 讲师接口

## 数据库设计

### 数据库E-R图

### 数据表设计

学生表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 自增 | 可空 | 备注 |
| id | int | 11 | 是 | 是 | 否 | 学生id |
| student\_no | varchar | 36 | 否 | 否 | 否 | 学号 |
| username | varchar | 50 | 否 | 否 | 是 | 昵称 |
| password | varchar | 36 | 否 | 否 | 否 | 密码 |
| realname | varchar | 36 | 否 | 否 | 否 | 真实姓名 |
| grade | varchar | 2 | 否 | 否 | 是 | 年级编号 |
| class\_no | varchar | 36 | 否 | 否 | 是 | 班级编号 |
| age | int | 3 | 否 | 否 | 是 | 年龄 |
| address | varchar | 255 | 否 | 否 | 是 | 地址 |
| telephone | varchar | 50 | 否 | 否 | 是 | 联系电话 |
| avatar | varchar | 500 | 否 | 否 | 是 | 头像地址 |
| email | varchar | 50 | 否 | 否 | 是 | 邮件地址 |
| description | varchar | 100 | 否 | 否 | 是 | 个人签名 |
| status | int | 1 | 否 | 否 | 否 | 账号状态，默认0为正常，1异常 |
| deleted | int | 1 | 否 | 否 | 否 | 逻辑删除默认0不删，1删除 |
| create\_time | datetime | 0 | 否 | 否 | 是 | 创建时间 |
| update\_time | datetime | 0 | 否 | 否 | 是 | 更新时间 |

讲师表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 自增 | 可空 | 备注 |
| id | int | 11 | 是 | 是 | 否 | 讲师id |
| teacher\_no | varchar | 36 | 否 | 否 | 否 | 讲师编号 |
| username | varchar | 50 | 否 | 否 | 是 | 昵称 |
| password | varchar | 36 | 否 | 否 | 否 | 密码 |
| realname | varchar | 36 | 否 | 否 | 否 | 真实姓名 |
| jobtitle | varchar | 20 | 否 | 否 | 是 | 职称 |
| grade\_no | varchar | 2 | 否 | 否 | 是 | 教授年级 |
| license | varchar | 500 | 否 | 否 | 是 | 证件照地址 |
| teach | varchar | 10 | 否 | 否 | 是 | 教授科目 |
| address | varchar | 255 | 否 | 否 | 是 | 地址 |
| telephone | varchar | 50 | 否 | 否 | 是 | 联系电话 |
| age | int | 3 | 否 | 否 | 是 | 年龄 |
| avatar | varchar | 500 | 否 | 否 | 是 | 头像地址 |
| email | varchar | 50 | 否 | 否 | 是 | 邮件地址 |
| power | int | 1 | 否 | 否 | 是 | 权限 |
| priority | int | 2 | 否 | 否 | 是 | 优先级 |
| description | varchar | 100 | 否 | 否 | 是 | 个人签名 |
| status | int | 1 | 否 | 否 | 否 | 账号状态，默认0为正常，1异常 |
| deleted | int | 1 | 否 | 否 | 否 | 逻辑删除默认0不删，1删除 |
| create\_time | datetime | 0 | 否 | 否 | 是 | 创建时间 |
| update\_time | datetime | 0 | 否 | 否 | 是 | 更新时间 |

管理员(教务处主任)表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 长度 | 主键 | 自增 | 可空 | 备注 |
| id | int | 11 | 是 | 是 | 否 | 讲师id |
| admin\_no | varchar | 36 | 否 | 否 | 否 | 讲师编号 |
| username | varchar | 50 | 否 | 否 | 是 | 昵称 |
| password | varchar | 36 | 否 | 否 | 否 | 密码 |
| realname | varchar | 36 | 否 | 否 | 否 | 真实姓名 |
| jobtitle | varchar | 20 | 否 | 否 | 是 | 职称 |
| remark | varchar | 255 | 否 | 否 | 是 | 备注 |
| license | varchar | 500 | 否 | 否 | 是 | 证件照地址 |
| teach | varchar | 10 | 否 | 否 | 是 | 教授科目 |
| address | varchar | 255 | 否 | 否 | 是 | 地址 |
| telephone | varchar | 50 | 否 | 否 | 是 | 联系电话 |
| age | int | 3 | 否 | 否 | 是 | 年龄 |
| avatar | varchar | 500 | 否 | 否 | 是 | 头像地址 |
| email | varchar | 50 | 否 | 否 | 是 | 邮件地址 |
| type | int | 1 | 否 | 否 | 否 | 管理员类型 |
| power | int | 1 | 否 | 否 | 是 | 权限 |
| priority | int | 2 | 否 | 否 | 是 | 优先级 |
| description | varchar | 100 | 否 | 否 | 是 | 个人签名 |
| status | int | 1 | 否 | 否 | 否 | 账号状态，默认0为正常，1异常 |
| deleted | int | 1 | 否 | 否 | 否 | 逻辑删除默认0不删，1删除 |
| create\_time | datetime | 0 | 否 | 否 | 是 | 创建时间 |
| update\_time | datetime | 0 | 否 | 否 | 是 | 更新时间 |

# 详细设计

# 系统实现

# 系统测试

# 结论

# 谢辞

参考文献