|  |  |
| --- | --- |
| 学院标志(改) 副本 | 惠州学院  HUIZHOU UNIVERSITY |

毕 业 论 文（设 计）

**中文题目：­­­­­­­­** 基于GitHub的实验教学系统的

的设计与实现

**英文题目：**Design and Implementation of an Experiment

Teaching System based on GitHub

**姓 名 \_梁研朊 \_\_**

**学 号\_\_1514080901114 \_**

**专业班级\_15计算机科学与技术1班 \_\_**

**指导教师\_\_\_\_曾少宁 讲师 \_\_\_**

**提交日期\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

教务处制

**学术诚信声明**

本人所呈交的毕业论文，是在指导教师的指导下独立完成。研究工作所取得的成果、数据、图片资料均真实可靠。除文中已注明引用的内容外，不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究做出重要贡献的个人和集体，均已在文中以明确的方式标明。本人完全意识到本声明的法律结果由本人承担。

本人签名： 日期：

基于GitHub的实验教学系统的设计与实现

作者姓名：梁研朊 专业班级：15计算机科学与技术1班 指导老师：曾少宁讲师

**摘 要**

基于GitHub的实验教学系统含有三类用户，分别是管理员、教师和学生。管理员用户对课程、实验、教师和学生信息进行管理，教师可以查看课程安排和发布课程实验，通过此系统统计学生的实验成绩。学生可以在系统上查看课程安排、实验安排和成绩。统计实验成绩的实现是通过GitHub的API，使用API对用户进行身份验证并将用户的相关库的数据抓取回来，再对抓取的数据进行贡献者，提交次数等的判断，生产统计情况表，教师自己设置评分标准生产成绩。

**关键词：**GitHub；实验教学；协同；教学管理系统

Design and Implementation of an Experiment Teaching System based on GitHub

Author: Yanruan Liang Specialty: computer science and technology Tutor: Shaoning Zeng

**Abstract**

Base on the Githut…Base on the Githut…

Base on the Githut…

Base on the Githut…

Base on the Githut…

**Keywords：** GitHub; Experiment Teaching; Teaching Management System

目录

[1 绪论 8](#_Toc1300477)

[1.1 研究的背景 8](#_Toc1300478)

[1.2 研究的意义 8](#_Toc1300479)

[1.3 国内外现状与发展 8](#_Toc1300480)

[2 关键技术介绍 8](#_Toc1300481)

[2.1 基于GitHub的API接口获取实验库的实验信息 9](#_Toc1300482)

[2.1.1 身份认证 9](#_Toc1300483)

[2.1.2 通过API请求数据 9](#_Toc1300484)

[2.2 Web技术 10](#_Toc1300485)

[2.3 前端框架 10](#_Toc1300486)

[2.3.1 jQuery 10](#_Toc1300487)

[2.3.2 Bootstrap 10](#_Toc1300488)

[2.4 数据库技术 11](#_Toc1300489)

[3 需求分析 11](#_Toc1300490)

[3.1 任务概述 11](#_Toc1300491)

[3.1.1 目标 12](#_Toc1300492)

[3.2 需求规定 12](#_Toc1300493)

[3.2.1 用户特点 12](#_Toc1300494)

[3.2.2 需求模型 12](#_Toc1300495)

[3.3 功能概述 13](#_Toc1300496)

[3.3.1 基础功能 13](#_Toc1300497)

[3.3.2 教师用户 13](#_Toc1300498)

[3.3.3 学生用户 14](#_Toc1300499)

[3.3.4 管理员用户 14](#_Toc1300500)

[4 系统设计 14](#_Toc1300501)

[4.1 系统总体结构设计 14](#_Toc1300502)

[4.1.1 管理员用户子系统 15](#_Toc1300503)

[4.1.2 教师用户子系统 15](#_Toc1300504)

[4.1.3 学生用户子系统 15](#_Toc1300505)

[5 系统功能模块的设计与实现 15](#_Toc1300506)

[5.1 管理员用户子系统 15](#_Toc1300507)

[5.1.1 个人中心模块 15](#_Toc1300508)

[5.1.2 前台管理模块 16](#_Toc1300509)

[5.1.3 教学任务管理模块 16](#_Toc1300510)

[5.1.4 实验任务管理模块 16](#_Toc1300511)

[5.2 教师用户子系统 16](#_Toc1300512)

[5.2.1 个人中心模块 16](#_Toc1300513)

[5.2.2 教学任务管理模块 16](#_Toc1300514)

[5.3 学生用户子系统 16](#_Toc1300515)

[5.3.1 个人中心模块 16](#_Toc1300516)

[5.3.2 实验任务模块 16](#_Toc1300517)

[5.4 数据库设计 16](#_Toc1300518)

[5.4.1 E-R模型图 16](#_Toc1300519)

[5.4.2 数据库表清单 18](#_Toc1300520)

[5.4.3 主要数据库表字段设计 19](#_Toc1300521)

[6 平台实现 20](#_Toc1300522)

[6.1 开发运行环境及开发工具 20](#_Toc1300523)

[6.1.1 开发运行环境 20](#_Toc1300524)

[6.1.2 开发工具 20](#_Toc1300525)

[6.2 框架运行界面 20](#_Toc1300526)

[7 系统测试 20](#_Toc1300527)

[7.1 软件测试的意义 20](#_Toc1300528)

[7.2 软件测试的方法 21](#_Toc1300529)

[7.3 系统验证 21](#_Toc1300530)

[7.3.1 测试用例设计 21](#_Toc1300531)

[7.3.2 功能测试用例 21](#_Toc1300532)

[7.3.3 测试结果分析 21](#_Toc1300533)

[8 结论与展望 21](#_Toc1300534)

[8.1 结论 21](#_Toc1300535)

[8.2 展望 21](#_Toc1300536)

[致 谢 22](#_Toc1300537)

[参考文献 23](#_Toc1300538)

[附录 24](#_Toc1300539)

# 1 绪论

* 1. **研究的背景**

近年来，计算机编程类课程实验主要还是以教师教学，学生独立完成某一例子作业的形式进行，而且作业实验报告的形式一般以文档为主，学生将实验代码写入文档。这种教学方式存在如文档并不利于教师阅读实验代码，汇总成绩依赖手工耗费大量时间等问题，且学生独立完成的形式无法锻炼学生的协同开发能力。此外，也出现了其他面向教育的课堂工具。GitHub是一个开源协同开发平台，其推出一个面向教育的课堂工具，与其他用于教育的开源技术类似[1-2]，且有研究表明，使用GitHub替代传统教学资源与教学方法，可以改进教学过程[3-4]。为解决传统计算机编程类课程实验的短板，迫切需要一个教学辅助工具，协助教师完成编程类实验的评测与成绩汇总，提高办事效率，节省不必要时间浪费。

* 1. **研究的意义**

使用GitHub布置编程类课程作业，是一种较为常见的辅助教学方法，只使用GitHub可以顺利的完成教学过程，但最终还是免不了手工的评测与汇总学生成绩，这种教学方式还有待改善。借助GitHub支持代码库（repository）、问题跟踪（issue）、项目分支（fork）、代码提交（commit）、代码合并请求（pull）、可标注标签等功能，设计一个基于 GitHub协同开发平台的实验教学辅助工具，可以协助教师完成教学课程的功能，解决传统编程实验教学过程中的学生提交实验代码的时间空间的限制与教师评测实验代码和汇总成绩依赖手工消耗不必要的时间等问题。

* 1. **实验教学系统国内外现状与发展**

国内比较多的是注重实验教学过程中的实验室管理等，同时也有对GitHub的研究

国外有教多关于基于GitHub教学工具的试用的案例，但也存在作业抄袭等问题

1. **关键技术介绍**
   1. **基于GitHub的API接口获取实验库的实验信息**
      1. **身份认证**

身份认证包含密码验证、Token认证和使用OAuth进行验证。基础验证是GitHub API最简单的认证方式，使用GitHub用户名和密码通过基础认证，此时客户端每小时可以制造5000个请求。其中用户名和密码的认证方式不是很安全，需要将登录的用户名和密码暴露在代码中，所以一般采用Token认证的方式比较多。OAuth 用令牌（token）可替代用户名和密码，是自己的用户名和密码不被共享。令牌有两大特色：一是可撤销访问：用户能够在任何时候撤销对第三方 app 的认证。二是有限访问：用户能够在授权一个第三方 app 之前检验特定的准入权限。若想使用自己的GitHub凭据代表很多用户发起请求，仍然有问题。鉴于此，如果你的网站或服务使用GitHub API 请求GitHub API 中的信息，应该考虑使用OAuth，并且使用用户共享的身份验证信息请求GitHub API。因此，将使用Token认证和OAuth验证来对用户身份进行验证。

* + 1. **通过API请求数据**

**通过向GitHub发送API的接口请求获取对应数据，而此系统中主要是先通过API获取贡献者（**contributors**）、**提交记录（commit）、合并记录请求（pull）、问题记录跟踪（issue）的数据，并根据传送回来的具体内容再进行更多内容的请求，如每一条合并记录对应的具体内容等。以下具体介绍主要的请求：

1. 请求贡献者的数据

https://api.github.com/repos/orgName/projName/contributors? client\_id=client\_id+&client\_secret=client\_secret&PER\_PAGE100&PAGE

1. **请求**提交记录（commit）数据

https://api.github.com/repos/orgName/projName/commits? client\_id=client\_id+&client\_secret=client\_secret&PER\_PAGE100&PAGE

1. **请求**合并记录请求（pull）数据

https://api.github.com/repos/orgName/projName/pulls? client\_id=client\_id+&client\_secret=client\_secret&PER\_PAGE100&PAGE

1. **请求**问题记录跟踪（issue）数据

https://api.github.com/repos/orgName/projName/issues? client\_id=client\_id+&client\_secret=client\_secret&PER\_PAGE100&PAGE

1. **请求具体的**提交（commit）数据

https://api.github.com/repos/orgName/projName/commits/shaNumber?client\_id=client\_id+&client\_secret=client\_secret

1. **请求具体的**合并（pull）数据

https://api.github.com/repos/orgName/projName/pulls/Number?client\_id=client\_id+&client\_secret=client\_secret

1. **请求具体的**问题（issue）数据

https://api.github.com/repos/orgName/projName/issues/Number?client\_id=client\_id+&client\_secret=client\_secret

* 1. **Web技术**

平台使用的Web前端技术是由HTML、CSS和JavaScript组成。网页主要也是由这三部分组成。这三种语言的入门比较简单，入门后就可以进行简单的网页设计。如果用房子来解释这三种语言对网页的作用，那么HTML就如一个框架与结构，决定网页的结构与内容；CSS就像房子的装潢，侧重对网页进行静态渲染，一个页面的界面是否美观友好，就由CSS来决定；JavaScript实现房子的具体功能，也就是可以控制网页的行为，对网站进行动态渲染，以此丰富网页的呈现效果，让用户获得更好的体验。

* 1. **前端框架**
     1. **jQuery**

jQuery是一个兼容多浏览器的轻量级的JS库，它可以帮助用户更方便地处理HTML、网页事件以及实现各种各样的动画特效，同时也方便客户端与服务器端之间进行AJAX交互，是当前使用率较高的JavaScript库之一[5]。

* + 1. **Bootstrap**

Bootstrap是基于HTML、CSS、JavaScript开发的简洁、直观、功能强大的前端开发框架，它提供了优雅的HTML和CSS规范，丰富的Web组件，使得Web开发更加快捷**。**

* 1. **数据库技术**

常见的数据库主要有三种，分别是SQL Server，Oracle以及MySQL。Oracle系统可移植性好、使用方便、功能强，适用于各类大、中、小、微机环境，基于Oracle收费且主要面向企业用户，对学生群体中使用SQL Server和MySQL更为普遍。相较与由于SQL Server的安装非常庞大而且复杂与使用起来没有MySQL方便，最终确定MySQL为平台的数据库。MySQL是一个多用户、多线程的开源关系型数据库管理系统，可以完全兼容各种操作系统，其使用的SQL语言为多种编程语言提供API，支持多种存储引擎。

1. **需求分析**
   1. **任务概述**

基于GitHub的实验教学系统的一个主要功能为辅助教师在GitHub完成教学过程的教学系统。当前编程类课程的实验的进行存在以下的不足：

1. 传统实验教学主要为文档形式，不方便教师的阅读；

2）传统实验教学学生提交实验代码等需要耗费较多的时间，如需要学习委员收集提交作业，有时间和空间的限制；

3）现在也有很多教师将作业布置在GitHub上，在GitHub上布置实验和完成实验虽然方便教师评测，但其还是无法避免需要教师花费时间手工进行实验成绩的汇总等；

为解决编程类课程实验过程中现在存在的耗费不必要的人力和时间问题，基于GitHub的实验教学系统应该具备以下基础的功能：

1. 管理员用户对网站前端内容、学生、教师、课程、实验信息进行编辑、管理；
2. 对课程的实验对应在GitHub上的实验库的数据进行获取；
3. 可设置实验成绩的评分标准，对学生的实验成绩进行汇总，并可导出实验成绩；
4. 教师可以在系统上发布实验要求；
5. 学生可以在系统上提交试验完成情况；
6. 提醒学生完成未过期的实验；
7. 提供友好的人机交互界面，让用户能够通过简易的操作获取想要的信息。

### 3.1.1目标

实验教学系统的目标是实验教学系统在设计和开发过程的总体依据及要求，经过对其总体分析之后，确定目标如下：

1. 可靠性：系统可以连续正确的处理业务，面对各种复杂情况的处理，出错率低；
2. 可理解性：设计友好的用户界面、简单易用的操作；
3. 时间效率性：系统反映速度快，网络延迟低；
4. 可维护性和适应性：系统要能适应业务需求随着时间的推进而不断变化的需要，易于修改、扩充和维；
5. 可用性：系统要充分考虑用户的日常使用需求，具备用户日常所需的各种功能。
   1. **需求规定**

### 3.2.1用户特点

系统由于是实验教学系统，应适用于一个基础的教育环境，如大学中的一个院系。因此系统的用户主要有三种类型，具体如下：

1. 教师：教师是教学的具体实施者，除了教学内容还有一系列的工作如布置实验作业，评测实验，汇总成绩等，对于编程类课程，其面对阅读量较大的实验评测，此外汇总成绩也需要手工进行，耗费大量时间；
2. 学生：学生主要就是接收实验任务和提交实验完成情况，其次学生可能会忘记提交作业；
3. 管理员：管理员需要对学生、教师、课程、实验信息进行编辑、管理，此外应将教学工作等的安排更新到系统的前端网页。

### 3.2.2 需求模型

**1）教师**

**个人信息管理**

**课程实验管理**

**2）学生**

**个人信息管理**

**实验任务管理**

**查看成绩**

**3）管理员**

**个人信息管理**

**前台内容管理**

**师生信息管理**

**课程教学管理**

**实验任务管理**

* 1. **功能概述**
     1. **基础功能**

1. 登录功能

登录用户可能出现认证失败、忘记密码和首次登录这三种情况。

1. 认证失败：用户账号或密码错误，即非法用户，不予以登录；
2. 忘记密码：通过向用户邮箱发送找回密码的链接，重设密码；
3. 首次登录：首次登录的用户未设置用户邮箱，将引导用户设置邮箱并重设密码，重新登录。
4. 个人信息设置功能

除设置邮箱、手机号、密码外，教师用户和管理员用户还可设置GitHub认证所需的GitHub用户名、Token、CLIENT\_ID和CLIENT\_SECRET。

1. 系统前端展示功能

前端内容为不用登录也可访问，且管理员可以在后台修改前端内容，也就是展示管理员在后台设置的前端内容，并提供系统后台的入口。

* + 1. **教师用户**

1. 查看课程信息：在系统上可以查看自己的任课班级及课程；
2. 发布实验任务：发布任务时在课程信息中点击“发布作业”，然后编辑作业标题和内容，设置作业最后期限，点击“发布”即可。
3. 统计GitHub实验信息：在使用系统统计实验信息前需先设置用户的GitHub信息，然后选择组织库名称、实验库名称、对应的班级和课程，设置评分标准，点击“统计”即可。统计时间的长短由要统计的实验库的大小决定，实验库的内容越多，统计时间越长。因此需要提示用户。
4. GitHub实验信息管理：有筛选功能，可根据班级、学期、课程、学号进行查找；有删除和导入数据的功能，此外还有添加功能，添加功能是由于学生没有按一定规则在GitHub上提交实验代码时，将会导致统计的实验信息中没有其学号，也就需要手工将其成绩信息添加上去。
5. 原始实验数据：查看GitHub原始实验数据，方便未录入成绩的同学手动添加实验成绩。
   * 1. **学生用户**
6. 查看课程：在系统上可以查看自己的课程及任课教师；
7. 实验任务：查看实验要求，对于未超过最后期限的作业，可以进行作业提交，已超过期限的作业不能提交；
8. 实验记录：查看提交过的实验记录，并可对未超过最后期限的实验进行再次提交；
9. 查看成绩：查看成绩。
   * 1. **管理员用户**
10. 页面风格管理：可以修改网站的Logo、背景图片、站名、建设单位、单位地址、背景颜色和文字颜色；
11. 一级标题管理：对网站前台的一级标题进行添加、删除、修改；
12. 二级标题管理：对网站前台的二级标题及对应的内容进行添加、删除、修改；
13. 课程安排：进行排课；
14. 教学管理：对教师、学生、课程、学期、班级信息进行添加、删除、修改；
15. GitHub实验管理：可以管理教师发布的实验任务、学生提交的实验记录、实验成绩原始，查看的GitHub实验数据，统计GitHub实验信息。
16. **系统设计**
    1. **系统总体结构设计**

根据基于GitHub的实验教学系统的系统需求和功能需求，系统主要包括三个子系统，分别为管理员用户子系统、教师用户子系统和学生用户子系统。每个子系统下可以进一步划分，管理员用户子系统含有个人中心模块、前台管理模块、教学任务管理模块和实验任务管理模块；教师用户子系统含有个人中心模块和教学任务管理模块；学生用户子系统含有个人中心模块和实验任务模块。

* + 1. **管理员用户子系统**

管理员用户子系统为系统的管理员操作，管理员对网站前台的内容、教学任务安排、所有课程实验任务和实验成绩进行管理，向下可分为四个模块，每个模块可继续按功能细分。个人中心模块含更改密码、GitHub用户信息设置、绑定邮箱功能；前台管理模块含页面风格管理、一级标题管理、二级标题及内容管理功能；教学任务管理模块含有教师信息管理、学生信息管理、班级信息管理、课程信息管理、学期信息管理、课程安排管理功能；实验任务管理模块含教师实验任务管理、学生实验记录管理、统计GitHub实验、实验成绩管理、GitHub实验原始数据表功能。管理员用户子系统的模块划分图如下图所示：



图

* + 1. **教师用户子系统**

教师用户子系统为系统的教师操作，教师主要在系统查看课程和发布实验任务并统计实验成绩。主要可以分为两个模块，为个人中心模块和教学任务管理模块。个人中心模块含有更改密码、GitHub用户信息设置和绑定邮箱的功能；教学任务管理模块含有查看课程、实验任务管理、统计GitHub实验、实验成绩管理和查看GitHub实验原始数据的功能。教师用户子系统的模块划分图如下图所示：



图

* + 1. **学生用户子系统**

学生用户子系统为系统的学生操作，学生主要在系统查看课程和完成实验任务并查看实验成绩。主要可以分为两个模块，为个人中心模块和实验任务模块。个人中心模块含有更改密码、更改基础个人信息和绑定邮箱的功能；实验任务模块含有查看课程、实验任务管理、查看成绩功能。学生用户子系统的模块划分图如下图所示：



图

1. **系统功能模块的设计与实现**
   1. **管理员用户子系统**

管理员用户子系统主要包括个人中心模块、前台管理模块、教学任务管理模块、实验任务管理模块和教师用户子系统模块。管理员的工作是对教学工作进行管理，所以会含有教师用户子系统和学生用户子系统的主要功能，方便进行管理。由于实验任务管理模块将会在教师用户子系统和学生用户子系统有更详细的介绍，以下对个人中心模块、前台管理模块、教学任务管理模块分别进行介绍。

* + 1. **个人中心模块**

**个人中心模块主要是对用户的个人信息进行管理。**

1. 功能说明

个人中心模块含更改密码、GitHub用户信息设置、绑定邮箱功能**。**管理员用户会含有初始的登录账号和密码，首次登录时会提示绑定邮箱，只有验证邮箱后才可以使用系统，绑定邮箱的目的是为了方便找回密码；而更改密码是为了定期的更改密码让账号更加安全；设置GitHub用户信息是为了使用“统计GitHub实验数据”的功能。

1. 设计实现

下面主要介绍绑定邮箱的流程图：



图 绑定邮箱的流程图

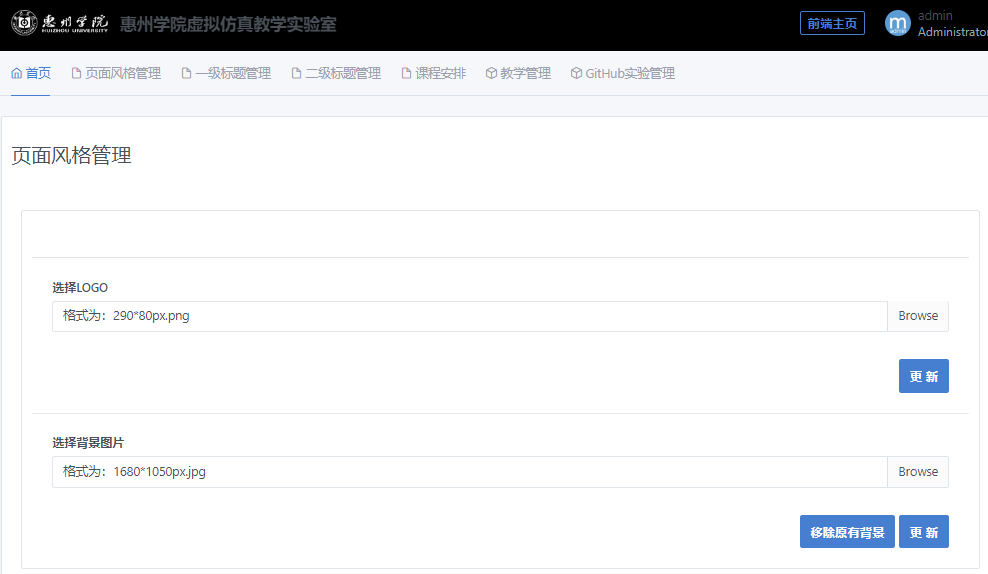
* + 1. **前台管理模块**

（1）功能说明

前台管理模块是在网站后台可以对网站前台的内容进行自定义编辑，方便对前台内容进行更新，含有页面风格管理、一级标题管理、二级标题及内容管理的功能。

页面风格管理可以对网站前台的logo、背景图片或颜色、文字颜色、网站名称及建设单位等进行自定义设置。其操作界面及操作逻辑简单明了，通过点击相关操作的按钮即可完成。如更改logo，操作为点击“选择Logo”栏目下的“更新”按钮，然后选择图片，上传即可。

（2）界面设计



* + 1. **教学任务管理模块**
    2. **实验任务管理模块**
  1. **教师用户子系统**
     1. **个人中心模块**
     2. **教学任务管理模块**
  2. **学生用户子系统**
     1. **个人中心模块**
     2. **实验任务模块**
  3. **数据库设计**
     1. **E-R模型图**

根据需求分析得出需求，并将其抽象为信息结构，得出数据库各主要实体的E-R图如下图所示：

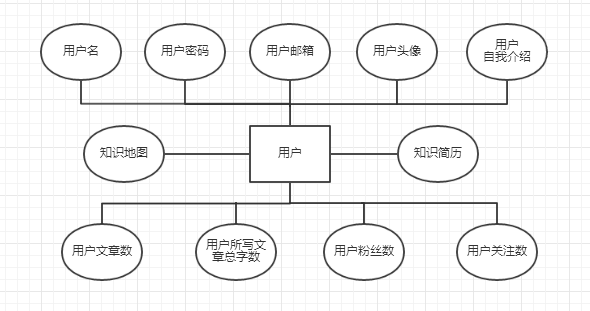


图4-20 用户实体及其属性

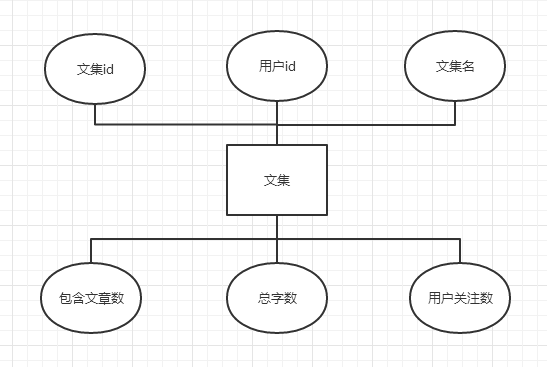


图4-21 文集实体及其属性

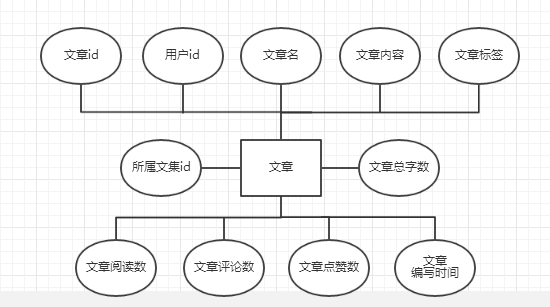


图4-22 文章实体及其属性

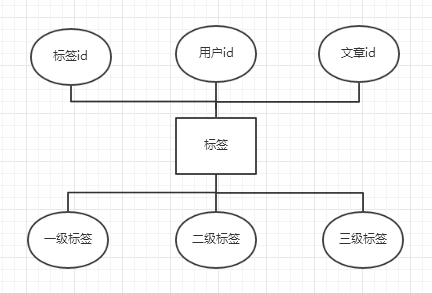


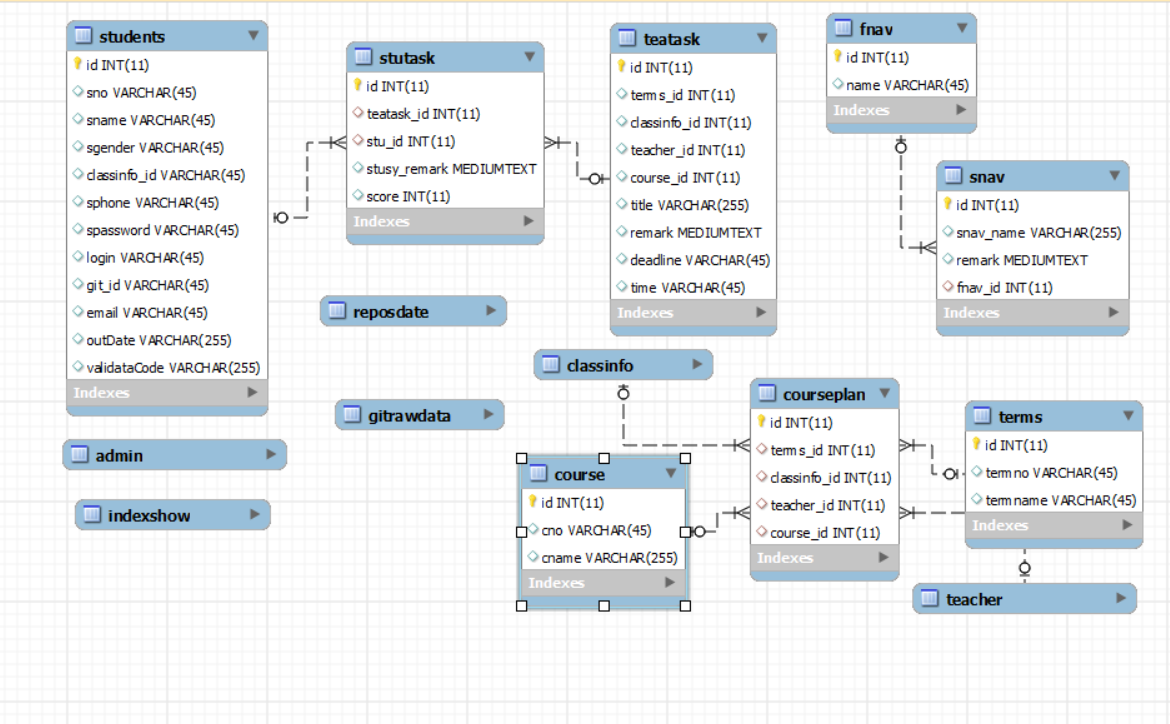
图4-23 标签实体及其属性

* + 1. **数据库表清单**

通过分析平台的功能需求，完成模块设计之后，得出所需的数据表如下所示：

|  |  |
| --- | --- |
| 表名 | 描述 |
| admin | 管理员信息表，存储管理员的基本信息和GitHub账号信息 |
| classinfo | 班级信息表，存储班级信息 |
| course | 课程信息表，存储课程信息 |
| Couseplan | 课程安排表，存储课程安排信息 |
| fnav | 一级标题信息表，存储前台页面中的一级标题的信息 |
| gitrawdata | GitHub原始数据信息表，GitHub实验中原始数据表的存储位置 |
| indexshow | 前台页面风格信息表，存储系统前台页面的风格信息 |
| reposdate | GitHub实验成绩表，存储GitHub实验相关信息与实验成绩 |
| snav | 二级标题信息表，存储前台网页中二级标题及其对应的内容信息 |
| students | 学生信息表，存储学生的基本信息 |
| stutask | 学生实验任务信息表，存储学生的实验任务信息 |
| teacher | 教师信息表，存储教师的基本信息 |
| Teatask | 教师发布实验任务信息表，存储教师发布的实验任务信息 |
| terms | 学期信息表，存储学期的基本信息 |

各个数据库表之间的关系如下图所示：



* + 1. **主要数据库表字段设计**

教师实体的属性有姓名、工号、性别、手机号码、住址、密码、GitHub用户、TOKEN、CLIENT\_ID、CLIENT\_SECRET、职称、邮箱、验证有效期和验证码。教师信息表的字段说明如下：

表 教师信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 是否主键 | 字段名称 | 数据类型 | 描述说明 |
| 是 | id | INT（11） | 自增长主键id |
| 否 | tName | VARCHAR(45) | 教师姓名 |
| 否 | tNo | VARCHAR(45) | 教师工号 |
| 否 | sex | VARCHAR(45) | 性别 |
| 否 | tPhone | VARCHAR(45) | 手机号码 |
| 否 | tAddr | VARCHAR(45) | 住址 |
| 否 | tPwd | VARCHAR(45) | 登录密码 |
| 否 | gitUsername | VARCHAR(100) | GitHub账号用户名 |
| 否 | TOKEN | VARCHAR(255) | 认证GitHub账号使用的Token |
| 否 | CLIENT\_ID | VARCHAR(255) | 认证GitHub账号使用的CLIENT\_ID |
| 否 | CLIENT\_SECRET | VARCHAR(255) | 认证GitHub账号使用的CLIENT\_SECRET |
| 否 | proftitle | VARCHAR(45) | 教师的职称 |
|  | Email | VARCHAR(255) | 邮箱，默认值为“0” |
|  | outDate | VARCHAR(255) | 认证邮箱时验证码的有效期 |
|  | validataCode | VARCHAR(255) | 验证码 |

课程安排表如下表X-x所示：

表 – 课程安排表

|  |  |  |  |
| --- | --- | --- | --- |
| 是否主键 | 字段名称 | 数据类型 | 描述说明 |
| 是 | id | INT （11） | 自增长主键id |
| 否 | terms\_id | INT（11） | 学期的标识符，存储学期编号 |
| 否 | classinfo\_id | INT （11） | 班级的标识符，存储班级编号 |
| 否 | teacher\_id | INT（11） | 教师的标识符，存储教师编号 |
| 否 | course\_id | INT（11） | 课程的标识符，存储课程编号 |

GitHub实验成绩表如下表x-3所示：

表 – GitHub实验成绩表

|  |  |  |  |
| --- | --- | --- | --- |
| 是否主键 | 字段名称 | 数据类型 | 描述说明 |
| 是 | id | INT （11） | 自增长主键id |
| 否 | Login | VARCHAR(45) | 存储学生GitHub账号的用户名 |
| 否 | Num | VARCHAR(45) | 存储学生的学号 |
| 否 | IssueNumber | INT（11） | 存储学生GitHub实验库中的Issue号 |
| 否 | IssueCount | INT（11） | 存储学生GitHub实验库中的Issue号的数目 |
| 否 | IssueLabels | VARCHAR(45) | 存储学生GitHub实验库中对Issue的标签内容 |
| 否 | Events | INT（11） | 存储学生GitHub实验库中的事件数 |
| 否 | FirstTime | VARCHAR(45) | 存储学生在GitHub实验库第一次提交实验代码的时间 |
| 否 | Pulls | INT（11） | 存储学生在GitHub实验库请求合并的次数 |
| 否 | PullDone | INT（11） | 存储Pull被标记为完成的实验的次数 |
| 否 | PullGood | INT（11） | 存储Pull被标记为好的实验的次数 |
| 否 | PullLate | INT（11） | 存储Pull被标记为迟交的实验的次数 |
| 否 | PullCopy | INT（11） | 存储Pull被标记为抄袭的实验的次数 |
| 否 | PullBad | INT（11） | 存储Pull被标记为完成的不好的实验的次数 |
| 否 | PullBug | INT（11） | 存储Pull被标记为有错误的实验的次数 |
| 否 | PullDemo | INT（11） | 存储Pull被标记为样例的实验的次数 |
| 否 | PullInteresting | INT（11） | 存储Pull被标记为有趣实验的次数 |
| 否 | Commits | INT（11） | 存储提交的次数 |
| 否 | Additions | INT（11） | 存储添加的次数 |
| 否 | Deletions | INT（11） | 存储删除的次数 |
| 否 | ChangedFiles | INT（11） | 存储更改文件的次数 |
| 否 | Comments | INT（11） | 存储评论的次数 |
| 否 | ReviewComments | INT（11） | 存储审核评论的次数 |
| 否 | org | VARCHAR(100) | 存储组织库的名称 |
| 否 | proj | VARCHAR(100) | 存储实验库的名称 |
| 否 | terms\_id | INT（11） | 存储改实验库对应的学期 |
| 否 | course\_id | INT（11） | 存储改实验库对应的课程 |
| 否 | score | INT（11） | 存储实验成绩 |
| 否 | tid | INT（11） | 存储改实验库对应的教师 |

教师发布实验任务信息表表如下表X-x所示：

表 – 课程安排表

|  |  |  |  |
| --- | --- | --- | --- |
| 是否主键 | 字段名称 | 数据类型 | 描述说明 |
| 是 | id | INT （11） | 自增长主键id |
| 否 | terms\_id | INT（11） | 学期的标识符，存储学期编号 |
| 否 | classinfo\_id | INT （11） | 班级的标识符，存储班级编号 |
| 否 | teacher\_id | INT（11） | 教师的标识符，存储教师编号 |
| 否 | course\_id | INT（11） | 课程的标识符，存储课程编号 |
| 否 | title | VARCHAR(255) | 存储实验任务的标题 |
| 否 | remark | MEDIUMTEXT | 存储实验任务的要求 |
| 否 | deadline | VARCHAR(45) | 存储实验任务完成的最后期限 |
| 否 | time | VARCHAR(45) | 存储实验任务的发布时间 |

1. **平台实现**
   1. **开发运行环境及开发工具**

### 6.1.1开发运行环境

### 6.1.2开发工具

* 1. **框架运行界面**

1. **系统测试**
   1. **软件测试的意义**
   2. **软件测试的方法**
   3. **系统验证**
      1. **测试用例设计**
      2. **功能测试用例**
      3. **测试结果分析**
2. **结论与展望**
   1. **结论**
   2. **展望**

# 致 谢

在论文完成之际，我要感谢我的导师老师 XXX 副教授，感谢他对我的孜孜不倦的教导，耐心和关怀，让我可以很顺利的完成毕业设计，他严谨的教学态度，过硬的理论知识和极强的实践能力都深深的感染着我，让我每时每刻都能拥有一颗积极向上的心，让我能对每件事都付出努力和激情，把事情做好，做完美，感谢导师，总是耐心的给我们开会，给我指导，给我鼓励，鼓励勇敢的提问，多点与同学交流，别因为面子之类的问题而导致了知识的不巩固，现在出来社会工作了，就更加体会到唐老师的好，知道当初是他的苦口婆心的教导，才使我的基础巩固，在社会工作中上手快，进步飞速，还有就是特别是在完成此论文和插件等过程中，给出了非常多的建议和意见和帮助，才使我能这么顺利的完成，在此，向 X 老师致以最崇高的谢意！

其次，感谢实验室所有成员，感谢这一年多的时间一起的奋斗，我的大学生活因为有你们而感到饱满和愉快，平时多得你们的帮助，知识得到的飞快的成长，生活得到更好的欢乐，感谢大学有你们，我得到的不仅仅是团结合作的精神，更多的是不会忘记的真挚友谊，感谢公司的王工，吕工，金工，在暑假时一起在实验室的奋斗与探讨交流，给予我实际中的开发技巧和经验都让我终生难忘；

最后感谢爸爸妈妈妹妹一直以来的体谅和支持，感谢所有教导过我的老师，感谢你们的耐心的教导，感谢你们的奉献，感谢大学四年一起学习的同学朋友，感谢师弟师妹们的一起学习和奋斗；

祝爸爸妈妈身体健康，祝 X 老师万事如意，祝老师们开开心心，祝同学们朋友们找到如意的工作，祝师弟师妹们天天向上，生活愉快！

# 参考文献

[1] Stracquadanio G, Yang K, Boeke J D, et al. BioPartsDB: A synthetic biology workflow web-application for education and research[J]. Bioinformatics, 2016, 32(22): 394.

[2] Izbicki M. Open Sourcing the Classroom(Abstract Only)[C]//ACM Technical Symposium on Computing Science Education. Memphis: ACM, 2016: 723-723.

[3] Murray J. Open education: A revolution of resources and community[C]// TCC Worldwide Online Conference. Honolulu: TCC， 2014: 65-72.

[4] Kamvar Z N, Lópezuribe M M, Coughlan S, et al. Developing educational resources for population genetics in R: An open and collaborative approach[J]. Molecular Ecology Resources, 2016, 17(1): 120-128.

[5] 袁江. jQuery开发从入门到精通[M]. 清华大学出版社, 2013.

# 附录