基于WEB的“工程项目训练”管理系统—教师端子系统

摘要

软件工程课程设计无疑对学生的项目实践方面有着重大意义的影响。从软件工程课程设计I——III学生都使用网站进行签到汇报当天的工作情况。在课程设计验收时使用传统的纸笔进行组内和组间互评。对于学生来说这些不是有工作量的问题。但对于老师来讲，每天签到汇报当天工作的学生情况不能一目了然，需要进行核对才能得出结果。关于组内和组间互评，老师对于收集到所有纸条进行反复查看、计算、审核后统计得分，最后统计出结果。

因此，基于WEB的“工程项目训练”管理系统—教师端子系统，是针对于校内软件工程课程设计阶段的一个管理学生信息，签到情况，成绩统计等多个模块于一体的教师端管理系统。

在软件工程课程设计阶段，该系统作为老师对于学生最终成绩考核的主要工具。从开始创建班级和小组，学生加入后，学生每天的工作汇报，对自己的评价，是否签到以及最终成绩统计都会跟随学生的表现来动态改变， 这一系统的推出无疑为老师解决了最繁重的工作，可以让老师在一天繁忙的教学工作之后享受自己的生活。

系统应用了B/S架构，使用js操作LeanCloud远程数据库。vue框架数据驱动改变视图，echarts进行数据可视化。

关键词：课程设计，考勤管理，B/S,javascript,LeanCloud

**Abstract**

Software engineering course design will undoubtedly have a significant impact on students' project practice. From the software engineering curriculum design I-III students use the website to sign the work situation on the day of the report. In the course of design and acceptance, traditional paper and pens are used for intra-group and inter-group evaluations. For students, this is not a matter of . However, for teachers, the situation of students who work on the day of daily reporting is not clear at a glance, Need to check to get results. Regarding the mutual evaluation within and between groups, the teacher collects all the paper strips for reviewing, calculating and reviewing, and then statistically scores. Finally, the results are counted.

Therefore, based on WEB's "engineering project training" management system - the teacher terminal system, is a teacher-side management system for the school software engineering curriculum design phase of a student information management, attendance status, performance statistics and other modules in one.

During the software engineering course design stage, this system serves as the main tool for the teacher's assessment of students' final grades. From the beginning of the creation of classes and groups, after the students join, the student’s daily work report, his own evaluation, whether the check-in and final score statistics will be dynamically changed in accordance with the performance of the students. The introduction of this system undoubtedly solved the most arduous task for the teacher. allows teachers to enjoy their life after a busy day of teaching work.

The system uses the B/S architecture and uses JS to operate the LeanCloud remote database. Vue framework data driven change view, Using echarts for data visualization.

KeyWords: Course Design, Attendence Management, B/S, javascript, LeanCloud.

# 概述

## 课题背景

学校每年的软件工程课程设计对学生来说是增加项目实践经验的最好的途径，这样可以使那些没有项目经验的同学在找工作时简历上也可以有项目经验可写，同样也是自己实践能力的一个加强。但是，对于老师来说，组织一次课程设计到最终考核，所有的工作量是远远超出负荷的。

考核包括学生签到，平时成绩，文档撰写，组件互评以及组内互评五个模块的最终结果。这些模块都需要老师去依次设置模块占比，手动计算每个学生每个模块的成绩，最终成绩相加，得出考核成绩。

所以，基于WEB的“工程项目训练”管理系统—教师端子系统，旨在简化老师在课程设计阶段对学生考核的过度关注，将一切流程尽可能做到自动化，老师仅参与客观的评分考核，其余成绩都是根据学生平时表现以及同学间的评价进行自动化统计，展现给的就是老师一个最终结果。

## **1.2 国内外现状研究**

网络的普及，软件的应运而生，大大小小的机构都需要软件来进行业务的支撑，对于学校，软件更常用的就是各种信息管理。我国高校学生的考核内容繁多，要求高，管理细，标准多，因此用传统的档案管理学生信息以及考核情况已经完全不适应现在的场景。

因此，为了去适应高校信息管理的建设与发展，高校信息的管理系统应紧密结合于现代的网络技术，用来达到改进教学的管理、改善教学质量和提高教学效率目的，因此，高校管理的信息化已经成为了信息发展的大势所趋[1]。高校学生的考勤管理在各个高校的教学管理的工作中尤为重要，并且伴随着招生规模的日益扩大，再加上我国高校学生的考核内容繁多，要求高，管理细，标准多，因此用传统的档案管理学生信息以及考核情况已经完全不适应现在的场景[2]。

所以说，一个合理的过程管理模式，使得课程可以顺利开展，所以只是生搬硬套考核的管理模式，是完全不可能体现出考核的最初目的和最终结果。因此，想要更有效去解决关于课程考核的有关问题，必须去结合很多方面的考核标准和各个标准的占比，最终才会将考核的灵活性和公平性，规范化更好的体现出来，这样的考核才会更有存在意义[3]。并且，随着网络快速发展，与数据库技术的日益成熟，因此，建立一个自动化，人工智能的学生考勤以及信息的管理系统，很明显是各高校的管理部门燃眉之急。文章参考了目前国内高校学生管理的发展情况及管理模式，借鉴了目前的国内比较先进的信息管理系统，对本院学生课程设计过程的考勤管理和成绩统计进行了全面设计，开发了一个"基于Web的"运行平台，实现了方便的Web系统的考勤统计[4]。

在整个的软件课程设计的过程中，流程基本是完全基于人工统计考勤记录和成绩，手工方式进行各项考勤，不仅信息输入量大，容易出现错误，而且需教师每天固定时间去统计学生考勤，最终还是以传统的手工统计最终的结果再录入到教务系统中，整个过程下来耗费了老师大量的时间和精力，并且重复的工作使得最终的效率低[5]。针对于这一情况设计了一个基于WEB的“工程项目训练”管理系统—教师端子系统。

对体系的物理架构来讲，传统的C/S模式若是要去进级体系，必需进级全部的客户端；但是，在B/S模式下进行的系统升级与客户端完全不会影响，仅需升级服务器的程序，这样，简化客户端，减轻系统的维护，升级成本以及工作量，同时也降低用户总体的拥有成本[6]。

而基于WEB的"项目工程训练"管理系统---教师端子系统，采用的B/S架构，既是Broswer/Service 架构，以Internet为中心，使用该教师端子系统的用户经过Broswer来实现，并且有关于系统的大部分的逻辑都主要在服务器端进行实现。简言之，用户可以在任何的地方连网并且利用WEB浏览器来进行实时的考勤[7]。 所以，论文将信息管理系统之基于WEB技术的B/S架构，进行深入分析，最终设计了一个基于B/S结构的基于WEB的“工程项目训练”管理系统—教师端子系统。重点在于研究基于B/S结构的学生信息管理以及考勤管理的系统总体结构以及实现的方法[8]，推出了基于WEB的管理系统—教师端子系统，从而实现学生课程设计时间段学生信息，以及考勤信息管理和成绩审核的工作流程的系统化、规范化和自动化，基本实现了考核的动态话管理，管理效率大大提高了[9]。

## 1.3 课题研究目标

要实现软件工程课程设计考核的自动化管理，基于WEB的“工程项目训练”管理系统—教师端子系统能很好的满足这一需求。

基于WEB的"项目工程训练"管理系统---教师端子系统是一个教师端的学生信息以及成绩核定的管理系统，一位老师可以经管一到多个班级。在学生信息模块下，老师可以将学生移到其所在班级的某个小组，或者将其移到所管理班级下的某个小组。也可以将该同学从自己经管的这些班级中移除掉。

学生动态模块下，展示的是每个学生对自己当天工作情况的总结，以及对自己的表现评级。

考勤信息是老师们最关心的问题了，该模块可以直观反映出该学生是否签到。老师可以随时查阅某个特定日期自己管理班级的同学们签到情形和情况分布。作为老师，该模块学生是否签到的状态可以被修改掉。并且修改后的考勤状况会被直接的反映在该学生的总成绩中。

接下来的成绩统计，是反映学生考核情况，也是该系统体系中尤为重要的模块。作为最终成绩展示模块，该页面的计算量无疑是最大的，有一个比较优秀的算法是此系统最成功的部分。该模块展示的是老师无法修改的学生考勤成绩，组间互评成绩，组内互评成绩，以及老师可以自定义的学生平时成绩和文档打分。当然每个模块的占比也是不一样的，需要老师根据自己的标准来定义每个模块的比重，最终统计出结果。

在系统的信息管理模块中，它主要包括了班级管理，小组管理与用户的密码改变这三个模块。

在班级管理这个子模块中，教师可以添加一个新的班级，然后再去小组的管理页面中为刚刚创建的新班级添加一个到多个小组；同时，教师对自己管理的班级也可以进行删除掉，并且同时删除了其下的所有小组和学生。

在小组的管理子模块中，教师通过下拉框选择班级，然后为该班级添加一个到多个小组；同样的，教师可以删除某个选定班级下的某个小组并且同时会将小组内的所有学生删除。

密码修改根据预留手机号进行密码修改。

在将要登录系统的教师没有账号的情况下，可以选择立即注册一个新账号，再信息管理中成立班级和小组，再通知学生让其加入对应的班级小组。若登录该系统的教师忘记了登录的密码，可以验证用户名和预留手机号找回密码。

## 1.4本文的章节安排

该章节着重介绍了项目的研究背景，必要性和系统使用场景和系统适合的使用人群。

在接下来的第二章节中将会挨次的叙述一下系统需求，其中主要包含了该系统的功能性需求，非功能性需求。

体系的产品需求将利用流程图展现一个完全的流程。功能性需求采取用例图进行特定的模块功能展示。非功能需求针对该系统的界面，接口和性能，可靠性和安全性方面开展论述。

第三章主要介绍系统的逻辑架构和物理架构，界面设计以及对数据库的设计做出了详尽的叙述。

第四章对于系统做了全面的分析，着重展示了几个具有代表性和复杂性的系统的运行效果图，并针对该效果图对功能模块进行说明。软件的开发环境以及搭建步骤的简要描述，相关软件技术的介绍，应用核心技术的详细解释。

最后一章是对所做工作的一个回顾总结，列出系统的优点和独特性，但还要总结出系统当前的不足，并对未来系统将要进行进一步的改善或者进一步完善给出详尽的计划，并给出可行性的研究方案。

## 1.5 本章小结

在该章节的最先介绍了此项目的开发背景，必要性和意义以及影响的人群。

接下来分析了国内外高校对于学生考勤系统普遍存在的问题，并且都迫切需要一个可以自动化，多方面考核又不失准确性和公正性的学生考勤系统。

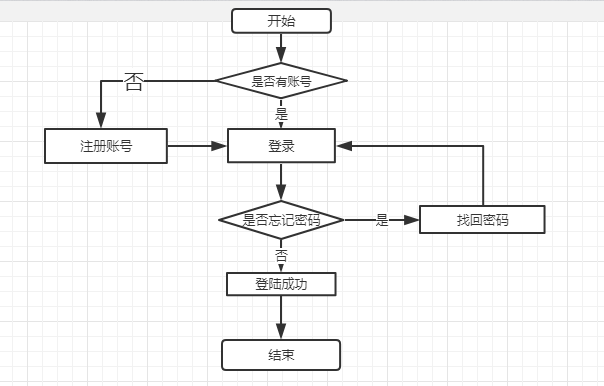
再介绍了该 基于WEB的“工程项目训练”管理系统—教师端子系统，旨在简化老师在课程设计阶段对学生考核的过度关注，将一切流程尽可能做到自动化，老师仅参与客观的评分考核，其余成绩都是根据学生平时表现以及同学间的评价进行自动化统计。最后，讲述了系统的使用流程和各模块的设计展现。

最后对于该篇论文的整体内容安排做了简要的概述，希望读者可以更清楚论文的结构和章节安排的理由。

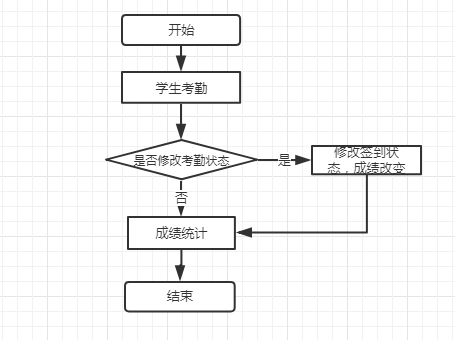
# 需求分析

## 2.1 产品需求

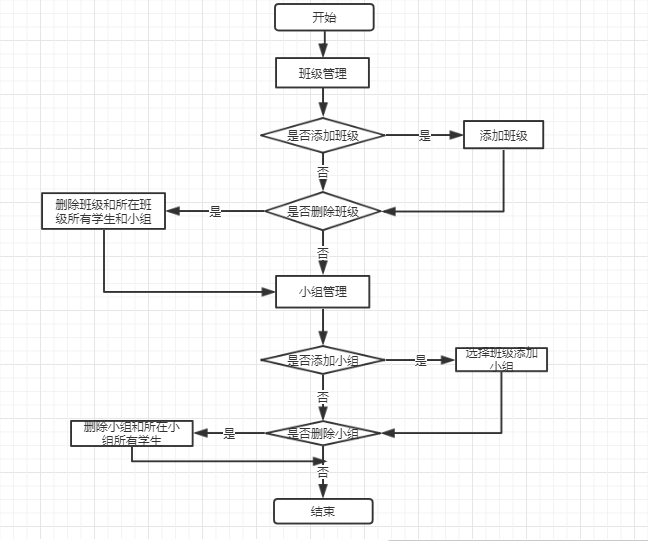
### 2. 1. 1 登录模块



### 2.2.2 学生考勤模块

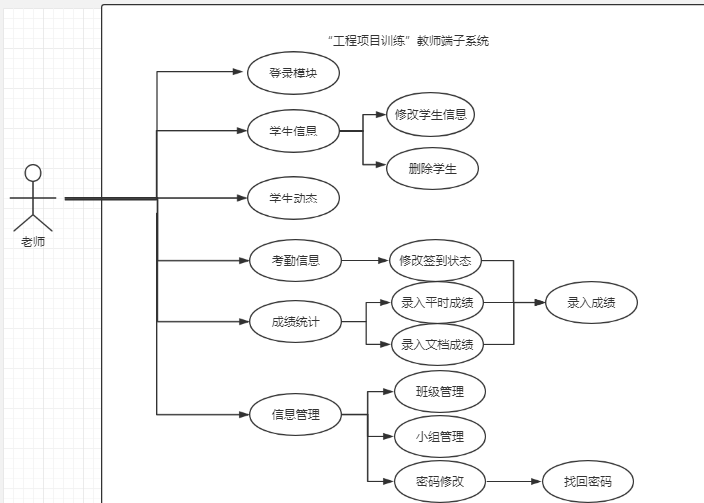


### 2. 2. 3 班级和小组管理模块



## 2.2 系统的功能需求

从上一章可以了解到项目的使用场景和使用人群，使用者为老师，那么各个业务模块的使用是必须了解的。



登录模块：首先在没有账号就要新建立一个账号。有账号但是忘记了密码就要正确填写用户名和预留手机号的验证新建立密码。登录完成后，若要新增班级，点击信息管理进入班级管理子模块的页面，建立班级后在小组管理子模块中通过下拉框选择班级，为对应班级添加小组。学生在新建立账号时就可以选择对应班级的对应小组加入。

信息管理模块：在教师一切的信息都建立完全后，登入该系统，直接会进入学生信息模块页面，展现加入该教师管理班级的所有同学的信息。（学号，姓名，班级，小组，电话）其中班级和小组是可以编辑的。老师可以将学生移到其所在班级的某个小组，或者将其移到所管理班级下的某个小组。也可以将该同学删除，从该教师管理的班级中。

学生动态模块：展示每个学生对自己当天工作情况的总结，以及对自己的表现评级。

考勤信息模块：是老师们最关心的问题了，该模块的表格展示可以直观反映出某学生某天是否签到。但是对于当天所有学生的签到情况却没有一个准确的了解。所以在页面右侧添加了一个toggle按钮，鼠标悬浮出现当天所有学生的签到情况和班级学生的签到情况的数据直观可视化的展示。作为教师，有权修改某个同学的签到的情况。并且该修改将会反映在总成绩中。

考核的统计模块：即成绩统计，对于学生来说是最重要的衡量标准，在系统中也尤为重要。作为最终成绩展示模块，该页面的计算量无疑是最大的，有一个比较优秀的算法是此系统最成功的部分。该模块展示的是老师无法修改的学生考勤成绩，组间互评成绩，组内互评成绩，以及老师可以自定义的学生平时成绩和文档打分。当然每个模块的占比也是不一样的，需要老师根据自己的标准来定义每个模块的比重，最终自动化的统计出结果。同样的，结果只是一堆数字，不能直观反映出学生成绩分布，在本页面的右边一样有一个toggle标志，鼠标mouseover可以呈现全部同学的成绩情况的饼图分布（忽略班级），柱状图展现的是以班级为单位的班级学生成绩散布情况。

## 2.3 系统非功能需求

### 2. 3. 1界面需求

A) 界面的原则要求，如易操作、简练、雅观、一致性等。整个系统的界面风格定义统一。

B) 色彩搭配：左边主导航采取沉稳的深蓝绿色，右边统一采取白色的背景。

C) 显示风格：信息展示统一采用Element-ui的table组件，分页使用Element-ui的pagimation组件且每页基本展示10条数据。少数的元素的结构分布利用bulma. 数据可视化采用Echarts提供数据展示。

### 2.3.2 接口采用

A) 使用云端数据库LeanCloud, 利用js与其给用户提供的SDK进行数据的云端同步[14]。

### 2.3.3性能需求

A) 作为集考勤与学生信息管理于一体的教师端系统，在修改学生信息或者成绩后都应响应式的全局改变，让老师所看到的都是最新数据展示。一次至多要求1000条数据。

### 2.3.4 安全性需求

A) 使用LeanCloud作为后台和数据库，对来访人员可以进行权限限定，非老师是不能登入到系统；

B) 修改密码时要求输入预留手机号，防止恶意修改；

C) 用户登录后其密码不能通过用户登录字段直接取得，避免歹意进犯；

### 2.3.5可靠性需求

A) 老师对于自己可编辑模块的任意操作都会及时反映到全局，保证全局的数据都是最新。

### 2.3.6适应性需求

A) 该系统基于vue框架，使用element-ui和bulma对页面进行布局，echarts数据可视化。

B) 对于IE8及以上的ie浏览器以及其他四种主流浏览器内核都支持良好。

C) 布局的响应式使该系统可以优秀的表现在所有的web端。

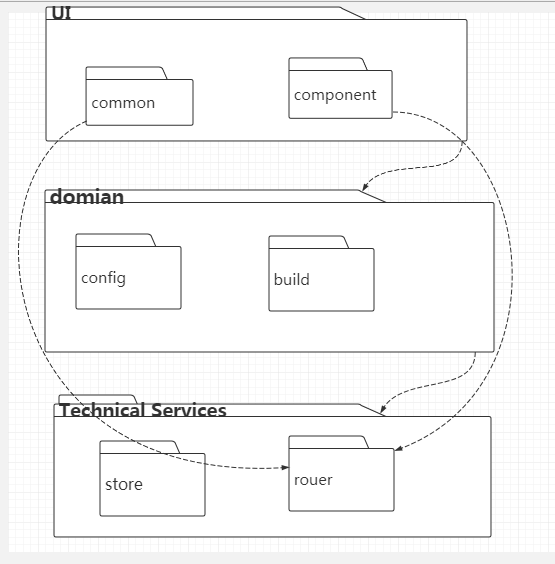
**2.4 本章小结**

本章对于系统的功能性需求和非功能性的需求进行了阐述，利用uml系统用例图对体系的各模块的功能和用法进行具体的展现。站在使用者的角度，使其对于系统的各个模块的使用有更熟练的掌握。

# 3. 系统设计

## 3.1 体系结构设计

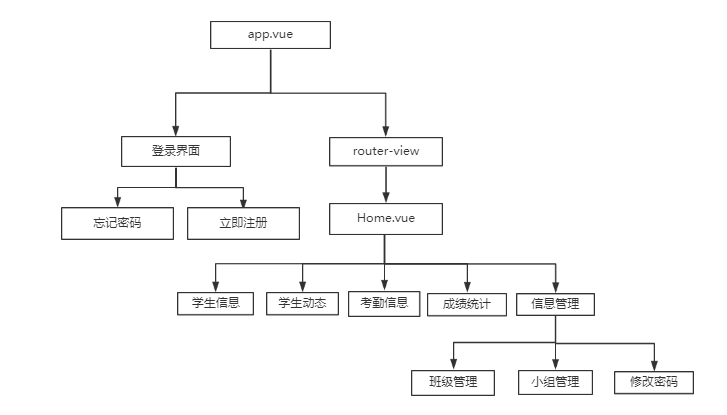
### 3.1.1 软件逻辑架构



### 3.1.2 软件物理架构

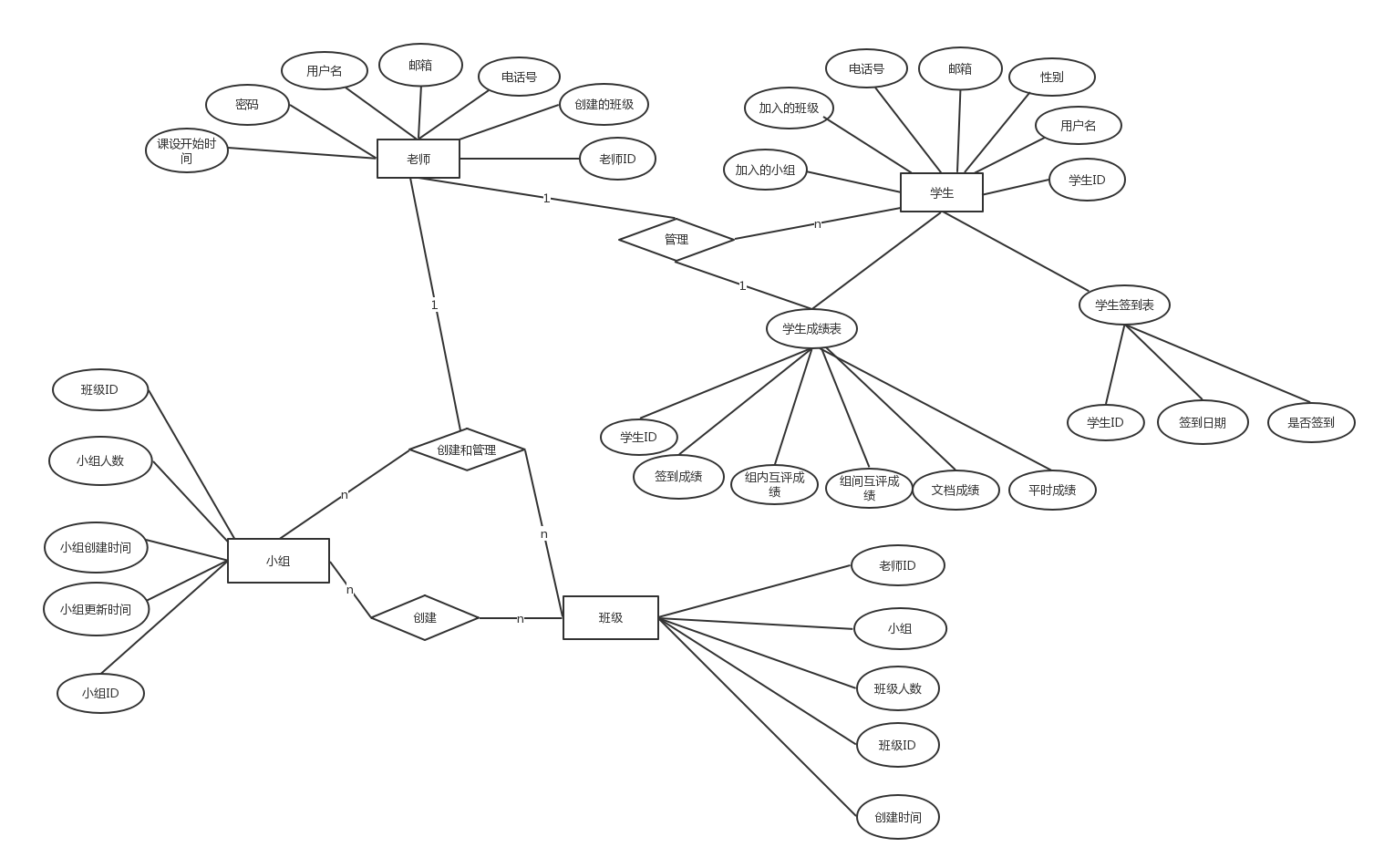
## 373981192202715625

## 3.2 界面设计



## 3.3 数据库设计

### 3.3.1 关系模型设计



|  |  |
| --- | --- |
| 数据库表 | 描述 |
| 用户表(\_User) | 表示用户是老师还是学生，存储基本信息 |
| 班级表(GradeTable) | 通过老师ID创建，表明此班级属于哪个老师的管辖 |
| 小组表(Team) | 通过班级ID创建，表示小组属于哪个班级 |
| 签到表(isSigninList) | 通过学生的ID创建，表示每个学生每天的签到情况 |
| 成绩表(ScoreList) | 通过学生的ID创建，表示每个学生每个模块的得分 |
| 组内互评表(InGroupComment) | 通过学生ID创建，表示该学生对其他人的所有评分 |
| 组间互评表(betweenGroupComment) | 通过小组ID创建，表示该组队其他组的所有评分 |
| 模块占比表(setStabdards) | 根据老师ID创建，且只能有一个，设置老师对于各个模块占比情况。 |
| 自评表(Comment) | 学生对于自己当天学习情况的汇报，以及对自己工作完成度的打分。 |

### 3.3.2 数据库表设计

### \_User(用户表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 表示用户唯一的ID |  |
| username | String | 否 | 否 | 用户名称 |  |
| password | String | 否 | 否 | 用户能否正常登录系统 |  |
| IsTeacher | Boolean | 是 | 否 | 不同权限登录不同系统 | false |
| grade | String | 否 | 否 | 学生所属班级 |  |
| Teamname | String | 否 | 否 | 学生加入的小组 |  |

### isSigninList(签到表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 表示签到唯一的ID |  |
| UserID | String | 否 | 否 | 学生ID |  |
| isSignIn | Boolean | 否 | 否 | 是否签到 | false |
| date | String | 否 | 否 | 日期 |  |

### scoreList(成绩表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 表示学生成绩唯一的ID |  |
| UserID | String | 否 | 否 | 学生ID |  |
| attendScore | Number | 否 | 否 | 学生签到总成绩 | 0 |
| usuallyScore | Number | 否 | 否 | 学生平时表现成绩 | 0 |
| documentScore | Number | 否 | 否 | 学生文档成绩 | 0 |
| inGroupScore | Number | 否 | 否 | 学生组内互评成绩 | 0 |
| betweenGroupScore | Number | 否 | 否 | 学生组内互评成绩 | 0 |

### GradeTable(班级表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 表示用户唯一的ID |  |
| TeacherID | String | 否 | 否 | 老师ID，表明班级所属老师管理 |  |
| createGrade | String | 否 | 否 | 班级名称 |  |
| Date | String | 否 | 否 | 该班级课设开始时间 |  |

### Team(小组表)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 |  | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id |  | String | 否 | 是 | 表示小组唯一的ID |  |
| GradeID |  | String | 否 | 否 | 班级ID |  |
| teamname |  | String | 否 | 否 | 小组名称 |  |
| createdAt |  | Date | 否 | 否 | 创建时间 |  |
| teamNumber |  | Number | 否 | 否 | 小组人数 | 0 |

**setStandards(模块占比表)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 唯一ID |  |
| TeacherID | String | 否 | 否 | 该老师的标准 |  |
| setAttend | Number | 否 | 否 | 考勤模块占比 | 20 |
| SetUsually | Number | 否 | 否 | 平时成绩占比 | 20 |
| setDocument | Number | 否 | 否 | 文档占比 | 20 |
| setInGroup | Number | 否 | 否 | 组内互评占比 | 20 |
| setBetweenGroup | NUmber | 否 | 否 | 组间互评占比 | 20 |

**inGroupComment(组内互评表)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 唯一ID |  |
| UserID | String | 否 | 否 | 发起评论人ID |  |
| TargetUserID | String | 否 | 否 | 被评价人ID |  |
| rating | Number | 否 | 否 | 对被评价人的打分 |  |
| Date | String | 否 | 否 | 评价日期 |  |

**betweenGroupComment(组间互评表)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 表示唯一的ID |  |
| UserID | String | 否 | 否 | 发起评价的人 |  |
| targetGroupID | String | 否 | 否 | 被评价小组ID |  |
| rating | Number | 否 | 否 | 对被评价小组评分 |  |
| Date | String | 否 | 否 | 发起评价的日期 |  |

**Comment(自评表)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 类型 | 是否可空 | 是否为主键 | 描述 | 默认值 |
| Id | String | 否 | 是 | 评价条目ID |  |
| UserID | String | 否 | 否 | 学生ID |  |
| Comment | String | 否 | 否 | 对自己今天表现的评价内容 |  |
| rating | Number | 否 | 否 | 对自己今天表现的评分 |  |
| date | String | 否 | 否 | 发表评论时间 |  |

## 3.4 业务模块设计

### 3. 4. 1 学生信息管理

* 用例名称：学生信息管理
* 范围：业务用例
* 级别：老师级别
* 主要角色：老师
* 涉众：老师、学生
* 前置条件：老师已成功登陆系统，已创建班级和小组，且有学生加入
* 后置条件：修改或删除学生，更新班级人数和小组人数，学生端信息相应 发生改变
* 主成功场景一： 1. 老师找到学生，编辑该学生的信息，比如移动到另一 个自己管理的班级

2. 移动成功，对应的班级和小组人数相应的增减

3. 所有有关学生信息展示部分都会发生相应改变

* 主成功场景二： 1. 老师找到学生，删除该学生

2. 删除完成，更新数据库，对应的班级和小组人数响应 式的改变

3. 所有有关学生信息展示部分都将会没有该学生

* 扩展（或替代流程）

A. 老师对此学生操作失败

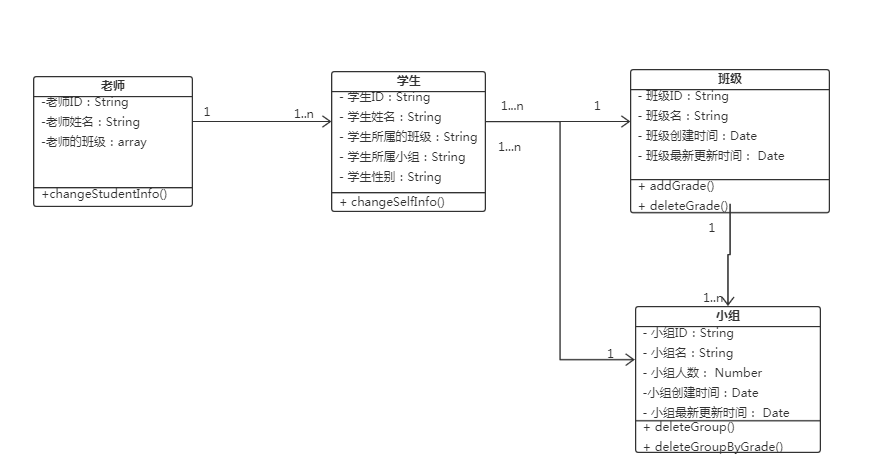
1. 老师通知学生让其退出老师管理的班级，或者重新加入班级

2. 老师等待系统稳定后重复主成功场景一或者主成功场景二

* 特殊需求：

批量操作学生信息

**3.4.1.1 软件单元构成**



### **3.4.1.2 业务流程处理**

### 未命名文件

### 3.4.2 考勤信息管理

* 用例名称：考勤信息管理
* 范围：业务用例
* 级别：老师级别
* 主要角色：老师
* 涉众：老师、学生
* 前置条件：老师已成功登陆系统，已创建班级和小组，且有学生加入
* 后置条件：修改学生的签到情况，签到状态发生改变，对应的签到总成绩 也会改变，从而会影响到总成绩。
* 主成功场景： 1. 老师找到学生，修改学生的签到情况。

2. 修改成功，签到状态发生改变，对应的签到总成绩 也会改变，从而会影响到总成绩发生改变。

3. 学生端的签到信息展示也会发生相应改变

* 扩展（或替代流程）

A. 老师对此学生操作失败

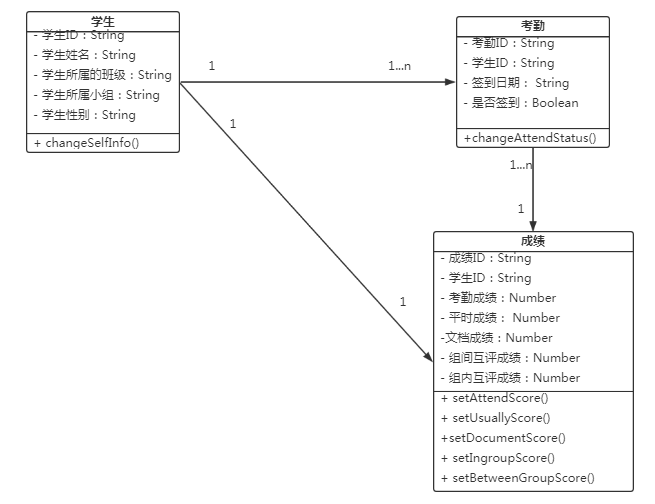
1. 老师等待系统稳定后重复主成功场景。

* 特殊需求：

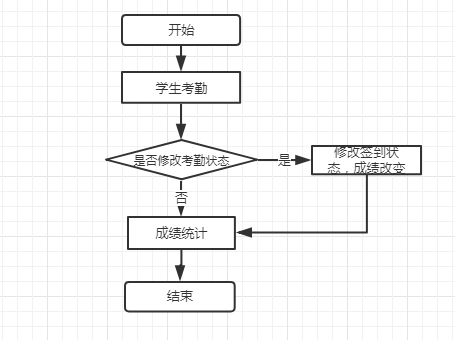
1. 批量操作学生签到情况

2. 展现某个学生的总体签到情况

**3.4.2.1 软件单元构成**



**3.4.2.2 业务处理流程**



### 3.4.3 班级管理

* 用例名称：班级管理
* 范围：业务用例
* 级别：老师级别
* 主要角色：老师
* 涉众：老师、学生
* 前置条件: 老师已成功登陆系统，已创建班级和小组，且有学生加入
* 后置条件: 教师建立了班级，在小组管理页面通过下拉框选择班级，对 该班级建立小组。学生端在注册的时候也可以检测到该班级 和其小组，并成功加入。老师删除班级后，其下的所有小组 和加入的同学全部被一并删除。
* 主成功场景一: 1. 老师创建班级成功。

2. 在创建小组页面会出现创建的班级，并可以选择该班 级，进行小组的创建

3. 创建班级和小组成功，学生端在注册的时候也可以 找到该班级和其小组，并成功加入

* 主成功场景二: 1. 老师删除班级成功。

2. 在小组管理页面没有该班级。

3. 该班级下的所有小组和学生都被一并删除。

* 扩展（或替代流程）

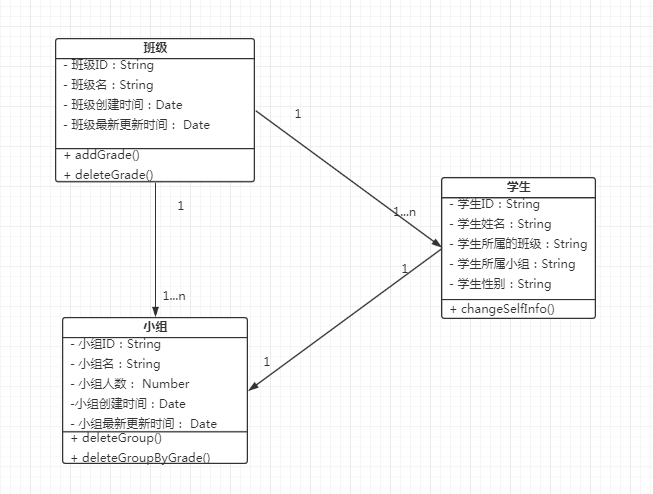
A. 老师对班级的添加和删除失败

1. 老师等待系统稳定后重复主成功场景一或者主成功场景二。

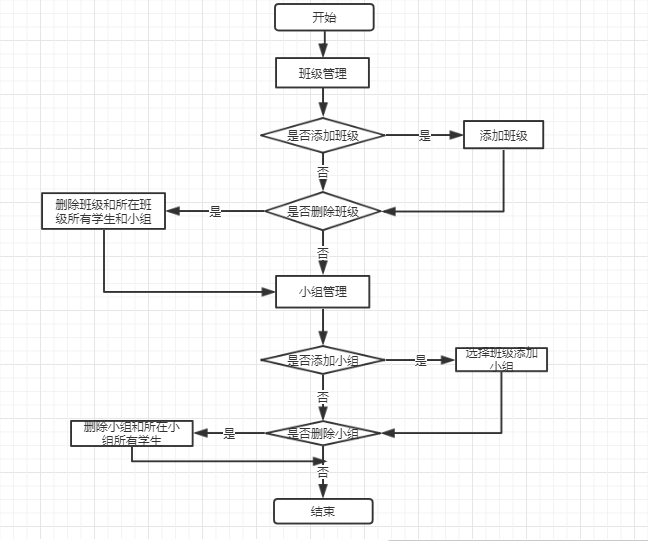
* 特殊需求：

1. 批量操作班级

**3.4.3.1 软件单元构成**



**3.4.3.2 业务处理流程**



# 系统实现与运行效果

## 4.1 开发环境

一个基于WEB的“工程项目训练”管理系统—教师端子系统，采用了现在比较先进的一些技术和主流的VUE框架，这些技术和框架都有较为成熟的技术支持和解决问题的社区，所以遇到问题之后还是比较容易可以找到解决方法。

本小结首要对软件的开发的环境（包含了：环境搭建，数据库和使用的框架，代码托管）等相关技术进行简略介绍。

### 4.1.1环境搭建

基于web的“工程项目训练”管理系统—教师端子系统，采用vue-cli脚手架搭建环境，webpack进行项目打包整合。

Vue是一个比较新型的框架，主要利用的是ECMAScript2015，因而像IE8及其更低的ie版本不支持vue的缘由就是这些浏览器对ECMAScript2015没有很好的内核支撑。因此通俗的来讲，就是vue支持所有可以对ECMAScript2015运行良好的浏览器。

利用vue的最普遍的简单方式就是在项目中引入下载的vue. js文件或引入vue的cdn.

但是为了项目以后整合的更容易，vue在构建比较大的项目时，推荐使用npm进行全局安装。项目环境搭建流程如下所示：

1. npm install vue 全局安装Vue

2. npm install --global vue-cli

3. Vue init webpack ‘项目名称’ 就可以生成一个基于webpack模板的 项目

4. cd 到你创建的项目根目录

5. npm install 安装项目所需的全部依赖

### 6. npm run dev 你的项目就在本机上可以跑起来了，一个采取vue-cli脚手架的webpack模板的项目就完成了。项目的环境搭建和初始化已经完成，所以接下来可以基于该项目结构就可以进行开发了。

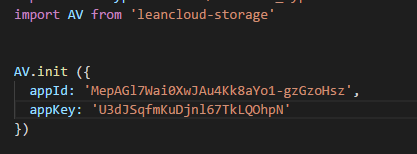
### 4.1.2 数据库使用

基于web的“工程项目训练”管理系统—教师端子系统，没有后台接口和数据库的支持，所以各种尝试之下，选择了云端数据库LeanCloud作为后台和数据库，使用javascript来对数据库进行常用的增删改查操作。

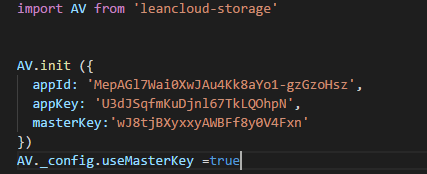
在使用LeanCloud之前，得先拥有LeanCloud账号，在项目中引用时需要进行以下初始化。

1. 利用npm, 来在全局安装LeanCloud. npm install leancloud-storage.

2. 在项目中需要用到的地方使永如下方法引入（一般该文件是作为控制层，直接对数据库行进操作）



在LeanCloud新建数据库的时候或自动生成一个\_User表，在该表中的用户可以使用该表自带的AV. User. signIn, 或者AV. User. SignUp和AV. User. LogOut方法针对于用户登录、注册和注销。所以说非常方便。但是，如果管理员和学生在同一个\_User表中，但是管理员还要对其他权限低于他的人进行信息操作，在网上搜，都是在\_User表中，操作非当前登录用户的信息是不允许修改的，所以我们需要更高的Master权限



这样做就可以对\_User表中非当前登录用户的信息进行操作了。

### 4.1.3 框架和组件的使用

除了vue这个基于数据驱动的MVVM模式的框架，在项目中还用到了两个从未使用过的框架：Element-UI和bulma. 使用ECharts实现一个可视化结构分为三个步骤：第一步：设置配置参数不同于D3．js的数据驱动方式，ECharts的所有可视化结构的配置选项都通过option进行设置[15] 。前者是饿了么基于vue开发的一个组件框架，所以对于我使用vue进行项目开发是十分适合的。而为什么要使用后者？ 因为在我第一次听说它，并花了一两天时间看完它的文档之后真的被它深深地吸引了，正如它官网所说Bulma? Bulma is an is an open source CSS framework based on Flexbox and built with Sass. It's 100% responsive, fully modular, and available for free[16]. 只要将它用在合适的地方完全的no style, no javascript, 即不用书写一行css代码，不用书写一行javascript.

说到一个css框架，好多人可能会想到bootstrap，所以这来简单的对bulma和bootstrap进行比较：

1. bulma有一个更现代化的样式组件，no javascript。而常用的Bootstrap是一个基于jquery前端的css样式库。

2. Bulma体积更小

3. Bulma的类名更具备语义化，如. title，. subtitle，. button, . input等等。Bootstrap却有冗杂的样式名。

4. 结合Font Awesome4 和Font Awesome5, 因此有更丰富的图标库。

5. Bulma有100+中css useful helper.

6. bootstrap有一些bulma没有元素和布局：比如列表组，格子布局或页面标题。

在了解了这两种样式框架之后就开始在项目中引用：

首先使用npm安装：

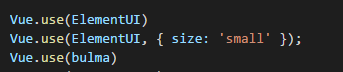
npm install element-ui

npm install bulma

main. js中引入：



在项目中使用，同样在main. js中添加：



这样就可以在项目任何地方使用了。

数据可视化的插件Echarts下载和引入与element-ui和bulma差不多，第一步采用npm下载：npm install echarts, 第二步在main. js中import进来，最后就是引用了。和前面两种框架的使用不一样，这就是为什么前两种是框架，echarts是插件了。

Echarts的引入其实就像是注册了一个全局方法。

import myEcharts from ‘Echarts’

Vue. prototype. $你起的名字 = myEcharts

在项目的任何地方引用的时候就这样使用：

this. $你起的名字. init()

### 4.1.4 代码管理工具的使用

在很早之前接触过github进行代码托管，但也就是只用了一两次的那种，所以github上的仓库还是空的，在经过了去公司的实习之后才真正的使用github托管代码，确实是一个很棒的代码托管工具。

1. 在自己的电脑上安装git.

2. 注册一个github账号，并建立一个新的仓库

3. 在本地的代码将要推送到github上，使用git init进行初始化仓库

4. git add . 来save所有文件的变更，并使用git commit --m ‘ 提交说明 ’ 来提交代码和git push推送到堆栈。

5. git pull从远程仓库拉取最新的代码，可以利用git checkout -b 新的分支名来建立新的分支。并在该分支上进行代码开发。

6. git log 查看你最近所有的commit, 后面跟的是一串序列号，当你要回退到某个版本的时候使用git reset -- soft commit后的序列号，回退到指定版本上，并保存该版本之后代码的全部变动。

git reset --hard commit后的序列号 ，回退到指定版本上，但不会保存该版本以后的代码变动（谨慎操作）。

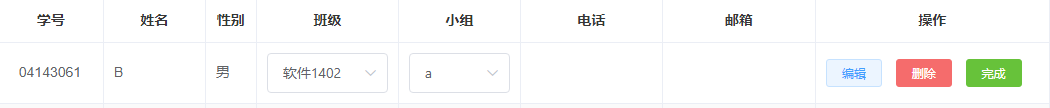
## 4.2 关键技术

### 4. 2. 1 Element-UI中table组件单元格的可编辑和Vue添加响应式的值

### **4. 2. 1. 1 Element-UI中table组件单元格的可编辑**

在使用element-ui的小伙伴们发现，element的table功能众多，就是没有单元格可编辑的这个方法或者属性，这确实让人头大，自己写的样式又特别丑，所以在权衡之下还是选择了element-ui的table组件。但是有要因为在我的项目中需要用到好多单元格可编辑的功能，这个问题是必须要克服的。所以在网上看了好久，发现了一个比较可行的办法，在单元格中嵌套一个input输入框，给每一行添加一个editFlag属性，默认为false, 首先让该输入框不显示。当点击该行或者满足某一条件的时候触发方法，使该行对应的editFlag为true, 则可以对单元格进行编辑。

理论点击编辑出现的效果是这样的：



### **4. 2. 1. 2 Vue添加响应式的值**

因此我使用this. result[i]. editFlag = false的方法为每一行添加了一个是否可编辑的标志，当点击该行的编辑按钮时，应该让input输入框显示出来，此时就可以进行编辑单元格了，但是并没有按照预期出现结果。视图并没有更新。

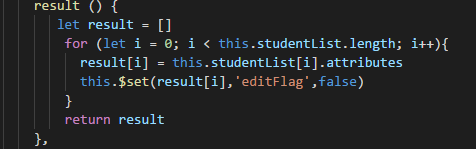
在频频查寻阅读官方文档和资料后发现，官方文档中已说得很清晰：vue的object属性响应式只建立在vue的data中object所拥有的属性，使用平常方法给对象添加新的属性都是不能实现object新增属性的响应式。是得不到效果的, 那么，怎样才可以给Vue对象添加响应式的值？

Vue. set( vue对象，要添加的属性，添加的属性值)

或者

this. $set( vue对象，要添加的属性，添加的属性值)

所以当我循环遍历数组，并的时候动态的给每一条数据这样加了一个属性：



如上方式对象添加的新属性就会拥有响应式。一次解决了Element-UI中table组件单元格的可编辑和Vue添加响应式的值两个issue, 同时对vue的object添加新属性的数据响应式有了更进一步的认识。

**4. 2. 2 vue-router保持状态，不用重复请求数据**

在基于WEB的“工程项目训练”管理系统—教师端子系统的开发时，使用vue-router进行页面的跳转，在router中配置好路由就可以实现所谓的单页面应用。

该基于WEB“ 项目工程训练” 管理系统--教师端子系统，分为六个大模块：登录模块，学生的信息模块，学生的动态模块，考勤的统计模块，成绩的统计模块，和信息的管理模块。对于信息的管理模块包括了三个小的子模块（班级的管理，小组的管理，修改密码）。对于需要实时获取到最新的学生数据，所以我在每个组件的beforeCreate钩子函数上面加了请求数据的方法，所以当页面每次进入的时候都会进行数据请求，如果只是单纯请求数据还好，尤其在成绩统计页面设涉及大量的计算，若每次点进来都要等页面加载一会儿，这样人不累觉得电脑都累。

所以怎样让页面第一次加载的时候请求数据，其余只要数据不变化就不用去请求，就跟离线浏览一样。

在网上搜，有人说计算页面加载的次数，如果是第一次请求数据加载，在本地存储数据，下一次加载的时候如果判断不是第一次加载，将数据与本地比较，相等的话取缓存，数据不相等请求。听起来好像是一个好办法，但是我觉得vue-router既然能出来，那么他们也能考虑到这个问题，应该还有更简单的方法。

果真在找到在router-view标签外包裹一层< keep-alive> 就完全可以解决这个问题。试了一下，在第一次加载的时候稍微慢一点，其次每次点击就真的像是在离线浏览一样，速度超快。当我改变了学生的签到情况之后，点击到成绩统计页面，意外发生了，并没有修改学生对应的考勤成绩。由于在没有加< keep-alive> , 页面每次加载都能触发beforeCreate中的methods去得到最新的数据，当包裹了< keep-alive> 后，页面只在第一次加载时执行beforeCreate中的methods. 那么怎样在数据变化之后点击相应页面可以得到最新的值？

在你变化了某些数据以后，及时去更新其他依赖它的变化的页面所需要的数据。也就是说，在签到页面修改了学生签到成绩之后，在加载一次统计学生签到成绩的函数，保证更新vuex中的数据是最新的。成绩统计页面的签到成绩就是在vuex的state中获取的。也就是说其实每次beforeCreate中的函数就是加载最新的云端数据到本地。我是在数据发生了变化的时候立即加载依赖于它的远端数据到本地。如此一来大大削减了请求数据库的次数，节省了带宽，加快了页面渲染呈现速度。

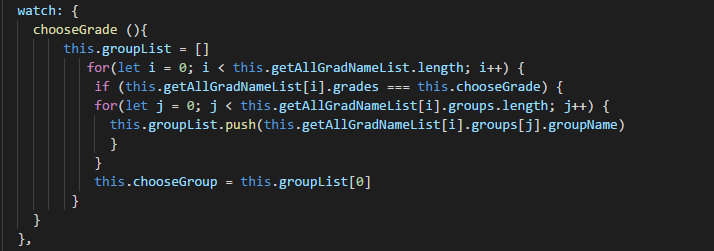
## **4. 2. 3 watch巧用之实现类似省市联动效果**

说到watch其实很多初学者都不太常用，因为官方文档也建议使用computed,所以在这之前我是几乎没有使用过watch的，因为computed已经完全满足了我的业务需求。

其实将watch与computed相比，两者区别是：computed是依靠其他数据的变化来触发自身的改变。Watch是本身的转变来触发其他数据的转变。（仅仅是我的理解）。

所以这次第一次使用watch实现的一个班级小组的选择是觉得非常有用的。

使用场景：在学生信息页面，老师点击编辑学生的信息时，会出现一个下拉框，先选择班级，选择之后小组下拉框中自动加载出来该班级所对应的小组，这样就实现了一个类似省市联动的效果。这样加入的学生就会被唯一的对应到班级的小组中。



ChooseGrade是选择的班级，当选择班级变化时就会触发该函数，立即加载grade对应的group.

点击软件1403加载出其所属的所有小组。



## 4.3 运行效果

### 4. 3. 1 登录模块



当输入非正确的用户名或非正确的password时会呈现出用户名或密码不正确的提示框。

遗忘了系统的登录的password, 可以填入正确的用户名和预留手机号重新设置新密码。点击立即注册，填写完善响应的信息会注册为一名教师。对加入其建立的班级的同学进行管理。在有了正确username和password后，登入系统，进入学生的信息页面。

### 4. 3. 2 学生信息模块



对于学生的信息模块，前置条件是老师已经建立了自己的班级和小组，已经有同学成功加入了。教师可以改变学生所属的班级和小组。若某同学属于教师管理的班级，该教师可以对该同学所在的班级和小组进行调剂；若该同学不属于该教师管辖的班级，教师可以删除该同学。让其在下次登录时重新加入班级和小组。

在这里我将B的班级和小组调整了，变化请看学生动态中B的班级和小组，此时数据库中B的信息也发生了改变。

### 4. 3. 3 学生动态模块

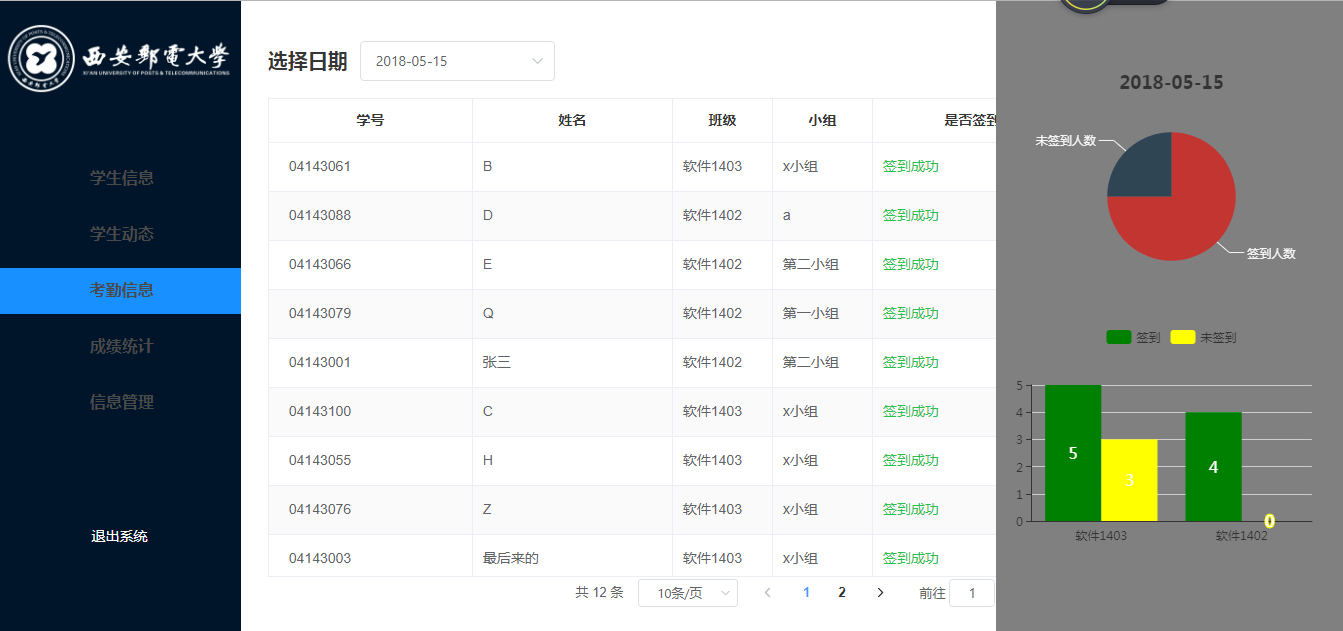


该模块主要作为展示的页面，位教师呈现每天同学们对自己工作进度的评价和完成度，和对本身今天表现的打分。

### 4. 3. 3 考勤信息模块

# 

展现每天的签到情况，选择日期可以查看具体某一天的学生签到情况，第二行展示的是学生的签到状态是可以选择的。细心地你可能注意到了右侧的灰色按钮：



当鼠标悬浮到灰色按钮上，出现使用echarts进行的数据可视化，饼图表示的是所有学生的签到情况，柱状图表示的是按班级展示的签到情况。

### 4. 3. 4 成绩统计模块



右边最上一排是老师设置各个模块的比重，点击编辑老师只能编辑学生的平时成绩和文档。组间互评和组内互评成绩是学生间相互打分给出的，教师无权改变分数。总分值按照各个模块的占比算出来的，不同区间成绩显示不同颜色。接下来，重新设置各个模块的占比，和B同学的平时成绩和文档成绩之后：



再次设置了各个模块的占比之后，学生的考核成绩总分就会被重新计算。同样右边还有一个灰色的按钮，鼠标悬浮后：



首先饼图根据整体情况展现出学生成绩等级占比情况：总分85分以上优秀，70-85中等，60-70良好，60以下不及格。

柱状图根据班级展现出学生成绩分布情况。

### 4. 3. 5 信息管理模块

**4. 3. 5. 1 班级管理模块**



该模块展现出此时有多少个班级，每个班级几个小组，和班级多少人，输入班级名点击创建，如果班级表中没有该班级则会创建成功，有的话提示信息，不能创建成功。

删除班级会删除该班级下的所有小组以及加入该班级的所有学生。

**4. 3. 5. 2 小组管理模块**



在刚才的班级管理页面，我创建了一个软件不存在小组的班级，这时候班级滚利页面中，小组个数为0，班级人数为0. 在小组管理页面中，选择班级时出现了刚创建的班级，选择之后，为其再创建小组，学生注册的时候就可以加入到该班级的该小组了。



当我给“软件不存在小组”班级添加了一个名为“不存在小组1”的时候，就会按如下展示，此时回到班级管理就会出现“软件不存在小组”小组个数为1, 班级人数为0.

# 总结及展望

## 5.1本文工作总结

在这次基于WEB“ 项目工程管理” 教师端子系统的开发过程中，采用了许多自己从未涉及到的框架和工具，vue, webpack, element-ui, bulma, echarts, github, LeanCloud等。

所以对自己来说是一个挑战，并且仍在通过自身学习和努力将这个系统继续完善。

从一次性接触到这么多新东西来说，没有觉得有多大困难，反而觉得是对能力的一种提升，对实践能力的增强，对学习能力的提高。

在使用以上的工具和框架之后，发现自己的学习能力变强了，对新东西的学习速度加快了，更有了学习的技巧。不再是拘泥于旧知识，而是适应框架的发展，对于新的知识不再是逃避和害怕，而是满怀求知欲的去学习，去拥抱新知识。

基于WEB的“工程项目训练”管理系统—教师端子系统的开发过程中，遇到了许多问题，享受了自己动手去解决问题的喜悦，也充满了对别人解决问题思路的敬佩之情，使自己的思维在这次基于WEB的“项目工程管理”教师端子系统的开发过程中也提高了不少和改变了许多。

基于WEB的“项目工程管理”教师端子系统旨在对软件课程设计过程的自动化，规范化以及公平公正化，最大程度的减少老师投放在课程设计成绩统计阶段的精力和时间，让老师可以在一天繁忙的教学工作之后享受自己的生活。

### 5. 1. 1 优势

界面方面：router-view( 右边区域) 采用白色底色。主导航采用深沉的灰蓝色，子导航active样式采用文字浅蓝绿色和文字底部加浅蓝绿色border的样式。表格采用了element-ui的table组件进行信息化展示和element-ui的pagination组件进行分页并且每页展示十条数据，可以选择每页展示数据的条数。Echarts对于数据的展示所有学生情况采用饼图，按班级分情况采用柱状图展示。

数据可靠性方面：该体系彻底实现了对同学的信息的动态化管理，成绩的实时性，在系统内的任一模块改变任一的数据，都会在依赖它的页面实时的展示出来，不会出现展现数据陈腐，或者数据拉取不实时的情况，在系统出现任何意外的情况下，全部可以通过再次登录该系统来处理。数据永远不会无缘无故消失或者展示错误。

使用难度方面：界面简洁，文字大小适中，各模块功能明确，条理分明，经过一次系统使用就可以完全掌握。总结：难度低，易操纵，需求明确。

### 5. 1. 2 不足

以一个可上线系统的标准来衡量，该基于WEB的“工程项目训练”管理系统—教师端子系统仍有许多不足：

A) 多语言需求。

B) 基本功能虽已经实现，但过于单调，内容可稍加丰富。

C) 界面没有专业美工设计的那种美感，布局稍欠合理。

D) 此系统应该不仅仅局限应用于软件工程的课程设计，应该满足于类似与软件工程课程设计的所有需求。

E) 多主题需求

F) 现在的后台是免费的，但是每次请求的数据量是有限制的，找一个好用的类似于LeanCloud的后台。

## 5.2未来工作展望

基于WEB的“工程项目训练”管理系统—教师端子系统旨在对软件课程设计过程的自动化，规范化以及公平公正化，最大程度的减少老师投放在课程设计成绩统计阶段的精力和时间，让老师可以在一天繁忙的教学工作之后享受自己的生活。

### 5. 2. 1 非功能需求

A) 可以请美工师做一下布局的重新设计，以及颜色搭配。

B) 目前的系统仅仅适用于软件工程课程设计的过程管理，以及类似于软件工程的课程设计过程管理的情况，却缺少一定的延展性。可以在成绩统计界面的成绩模块老师可以自定义有多少个模块，而不仅限于考勤、平时成绩、文档、组件互评和组内互评五个标准。

C) 现在的后台是免费的，但是每次请求的数据量是有限制的，找一个好用的类似于LeanCloud的后台。

### 5. 2. 2 功能需求

A) 老师作为使用者，应该有自己的擅长的语言，所以应该让系统为多语言。加一个语言选择，可以选择中文简体，中文繁体，English, 以及其他类型的语言。可以试利用vue的多语言来实现这一功能需求。

B) 页面过于简陋，可略加一些其余小功能。如：按同学的姓名或者学号search学生条目，按日期search学生的动态，根据班级展示学生的信息等类似于分类展示查询的小功能需求。

C) 做两套主题。将原本的css文件拆分为三个文件（公共样式css, 主题1, 主题2, 在页面加载前判断当前的localstorage中是否有该主题，有的话默认加载该主题，没有的话设置默认主题，创建localstorage的主题Item. 在换主题时，按照点击主题的值按需加载对应的css文件，改换主题。

致谢

参考文献：

[1] 叶露阳. 基于Web的学生管理信息系统的分析和设计[D]. 厦门大学，2014.

[2] 陈可赢. 考勤系统的发展背景. 厦门科技报. 2012, 11(3): 23~24.

[3] 王鲁云，张明君，王颖，于海霞，许少娟，李美花，刁立强. 实验课教学过程管理模式的改革与实践[B]. 大连理工大学城市学院. 2013

[4] 徐伟志. 基于WEB的学生考勤管理系统的设计. 嘉应学院医学院. 2017

[5] 赵海峰，石明翔，孟庆轩，杜娟，张春利，张丽华. 基于B/S结构的考勤系统设计. 北京现代职业技术学院. 2015

[6] 雷音，陈涛. 基于web的高校学生成绩管理系统的设计与实现[A]. 赣南医学院网络中心. 2008.

[7] 龙洋思. 浅析B/S架构的考勤管理系统的应用及存在问题. 成都水文地质工程地质队. 2016

[8] 李重，张震. 基于Web的网络考勤管理系统[J]. 武警工程学院. 2009(02).

[9] 李平. 安岳中学基于RFID学生门禁考勤管理系统的设计与实现[D]. 电子科技大学. 2013.

[10] https: // blog. csdn. net/ zjw0742/ article/ details/ 74012148 element-ui 表格实现单元格可编辑的方法.2017.6

[11] 杨小福 http: // www. jb51. net/ article/ 120079. htm 谈谈对vue响应式数据更新的误解. 2017. 8

[12] RoamIn. http: // www. jb51. net/ article/ 122570. htm 深入理解vue-router之keep-alive. 2017. 8

[13] 胡晓的简书 https: // www. jianshu. com/ p/ 48f1c97babfe Vue. js项目更换皮肤，同时支持多套皮肤. 2018

[14] 陈刚. 《基于MBaaS的iOS端网络管理系统设计与实现》[D]. 西安电子科技大学. 2015

[15] 杨志豪. 《应用协议支撑系统AIS的可视化支持》. 东南大学. 2016

[16] 《Card | Bulma: a modern CSS framework based on Flexbox》