**学 号：201414600207**





**毕业设计说明书**

GRADUATEDESIGN

**设计题目：****“WE创”创新团队组建和创意分享平台**

**—创意分享子系统**

**学生姓名： 于志鹏**

**专业班级： 14计算机科学与技术2班**

**学 院： 信息工程学院**

**指导教师： 马月坤 教授**

**2018年6月1日**

# 摘 要

在国家提倡建设创新型国家，培养创新型人才的发展趋势下，高等院校的学生渴望实现自身的价值，创造价值，利用自己的知识与创新去做一些实际的事情意愿越来越强烈，他们希望多多参与项目实践，实现自己的创新想法，但是却苦于没有这样的锻炼的平台，或者有的学生有很好的创意的点子，但是却苦于不能找到志同道合的同伴来实现梦想，创造价值。

通过结合我校学生创意项目管理与人才实际需求，采用现在成熟的python web开发框架与其他相应技术实现的“we创”创意项目管理平台，满足了用户发表创意，相互交流，招揽人才，开创项目，参与项目的需求，解决了人才与项目需求之间的矛盾，为大学生创意实现与项目团队组建提供了有力的支持与保障。

整个系统的开发过程遵循软件工程过程及规范，具体完成了该系统的分析、设计、编码实现等环节的相关任务。

**关键词:** 项目管理；MySql；软件工程；Python;

# Abstract

In the trend of developing innovative countries and cultivating innovative talents, the students of colleges and universities are eager to realize their value, create value, and make use of their knowledge and innovation to do some practical things more and more strongly. They hope to take part in the practice of the item and realize their own innovative ideas. But there is no platform for such exercise, or some students have good ideas, but they are hard to find like-minded Companions to realize their dreams and create value.

By combining the creative project management and the actual needs of the students in our school, the "we creation" creative project management platform which is realized by the mature Python web development framework and other relevant technologies is adopted to meet the users' creativity, mutual exchange, recruit talents, create projects, meet the needs of the project, and solve the talents and items. The contradiction between the needs of the students provides a strong support and guarantee for the creative realization of the university students and the formation of the project team.

The development process of the whole system follows the software engineering process and specification, and specifically completes the related tasks of the system's analysis, design, coding implementation and other links..

**Keywords**: project management; MySql; software engineering; Python;

目 录

[摘 要 I](#_Toc515531662)

[Abstract II](#_Toc515531663)

[第1章 绪论 1](#_Toc515531664)

[1.1本课题研究的背景 1](#_Toc515531665)

[1.2本课题研究的意义 1](#_Toc515531666)

[1.3.本课题的来源 1](#_Toc515531667)

[1.4本文主要内容 2](#_Toc515531668)

[1.5 本章小结 2](#_Toc515531669)

[第2章实现技术 3](#_Toc515531670)

[2.1 Python 3](#_Toc515531671)

[2.2 Django 3](#_Toc515531672)

[2.4 AJAX 4](#_Toc515531673)

[2.5 JavaScript 与 jQuery 4](#_Toc515531674)

[第3章 人才招募平台的需求分析和设计 6](#_Toc515531675)

[3.1 系统需求分析 6](#_Toc515531676)

[3.2 系统概要设计 6](#_Toc515531677)

[3.3系统非功能性需求分析 7](#_Toc515531678)

[3.3.1 技术可行性 7](#_Toc515531679)

[3.3.2 经济可行性 8](#_Toc515531680)

[3.34 系统安全性 8](#_Toc515531681)

[3.4 本章小结 9](#_Toc515531682)

[第4章 数据库设计 9](#_Toc515531683)

[4.1数据库概念结构设计E-R图 9](#_Toc515531684)

[4.2数据库逻辑及物理结构设计 10](#_Toc515531685)

[4.3本章小结 19](#_Toc515531686)

[第5章 系统详细设计与实现 19](#_Toc515531687)

[5.1开发工具与环境 19](#_Toc515531688)

[5.1.1 MySQL 数据库 19](#_Toc515531689)

[5.1.2 Redis 20](#_Toc515531690)

[5.1.3 Sublime Text 3 20](#_Toc515531691)

[5.1.4 PyCharm 21](#_Toc515531692)

[5.1.5 Axure RP 21](#_Toc515531693)

[5.1.6 Git 21](#_Toc515531694)

[5.2 用户注册与登录模块 23](#_Toc515531695)

[5.2.1 用户注册 23](#_Toc515531696)

[5.2.2 用户登录 24](#_Toc515531697)

[5.3 系统详细设计 24](#_Toc515531698)

[5.3.1平台展示 25](#_Toc515531699)

[5.3.2 优秀项目、优秀创意展示 26](#_Toc515531700)

[5.4招募项目模块 27](#_Toc515531701)

[5.4.1招募项目展示 27](#_Toc515531702)

[5.4.2项目详细信息 27](#_Toc515531703)

[5.4.3评论与申请招募 29](#_Toc515531704)

[5.5.创意分享模块 30](#_Toc515531705)

[5.5.1 创意分享展示模块 30](#_Toc515531706)

[5.5.2 创意详情模块 32](#_Toc515531707)

[5.7 个人中心模块 33](#_Toc515531708)

[5.7.1 用户信息模块 33](#_Toc515531709)

[5.7.2个人项目模块 37](#_Toc515531710)

[5.7.3 创意管理模块 38](#_Toc515531711)

[5.7.4用户消息模块 41](#_Toc515531712)

[5.7.5用户关注模块 41](#_Toc515531713)

[第6章 结论与展望 42](#_Toc515531714)

[参考文献 44](#_Toc515531715)

[谢 辞 44](#_Toc515531716)

[注 释 45](#_Toc515531717)

[附 录 47](#_Toc515531718)

注 释 [12](#_Toc198646055)

附 录 [12](#_Toc198646055)

# 第1章 绪论

## 1.1本课题研究的背景

中国正走在由中国制造转向中国创造的路上，而中国青年正应该是这次浪潮上的弄潮儿，高等院校大学生知识理论丰富，思维活跃，正是产生创造的最佳时期。但是很多大学生苦于没有项目或者人才资源，有好的创意想法与创意项目没有办法与他人分享讨论，很多好的创意想法为此搁浅，还没有孵化就结束在了想法阶段，或者还有一些学生掌握了专业技能，想要进一步磨练自己的技术本领，但是没有好的项目去发挥磨砺自己的本领，无法将知识转换成生产力，不能更加深入了解自己获取的知识，造成人才资源的浪费。

为了让学生更好的发挥并实现他们的想法创意，让有技术的同学能偶学以致用，“WE创”创意项目管理系统通过信息化手段管理分享学生的创意，让更多的学生看到并参与讨论，进行思维的碰撞，讨论创意实现的可能性，将想法变成现实，由创意变成实际项目，并可以在该平台招募项目所需人才，为学生创新扫清障碍，搭建起一条高效的创新项目--人才沟通平台。

该平台还提供了项目发起与项目招募功能，项目发起人有好的项目，可在该平台发起项目与招募，注明招募要求，发出招募贴，实现团队组建。而且目前国家支持创新，提倡培养双创人才，在国内举办各大创新创业比赛，学生可以在本平台进行创意分享，项目招募，团队组建，申报项目并且获得参赛机会，开拓眼界，认识不同的人，发散思维，进行学科交叉，更好的完善项目，提升自己的水平，为学校争取荣誉。如果有好的创业想法，还可以在这里寻找创业伙伴，为创业积攒人才资源

当前我国大学生在校学习到科学知识文化后 很难有平台将所学的东西进行应用，转换成自己   而且我国提倡创新发展 创建创新型国家，很多青年学子相应国家号召，在生活中学习中创新，但是有时只有创新的想法却没有相应的人才资源，根据实际生成的创新想法却因为资源匮乏被扼杀在摇篮里，而有的学生在学校学到了很多的知识，而且也会自学一些好的技术，但是苦于没有施展的空间，很多只能纸上谈兵，不能让知识很快转换成自身能力，不能学以致用，加深对知识的理解，不能让自身能力得到使用，实现自己的价值。

## 1.2本课题研究的意义

在创新创业潮流的影响下，大学生创业成功的案例比比皆是。分析这些成功的案例，我们发现重要的不仅仅是好的创意点子，也需要的是优秀的团队、好的机遇和实现创业的平台。现在在校大学生，见识广泛，思维活跃，有时会产生的很好的创意或者项目想法，但是因为很难找到志同道合的合伙人，创意或者项目就会被搁置在脑海中无法实现。而且大学高等教育比较注重理论知识的培养，很多学生学了很多的理论知识却很难马上应用到实际中，有时候有很好的项目却很难找到相应其他学科的合伙人，也很难实现学科交叉技术创新。“WE创”通过对人们创意项目的整合与分享就可以很好的解决这些难题，人们可以在本平台模块上进行创意讨论，思想碰撞，公布项目，发布招募贴来招到匹配的人才，或者加入心中向往的项目，该平台实现了人才与人才的交流，人才与项目的双向互动交流，帮助用户找到志同道合的伙伴共同完成项目，也使得参与用户学有所用，为项目与人才搭建一个桥梁，达到双方互利共赢的效果。

## 1.3.本课题的来源

青年大学生有很多创业想法或者参与创业奉献自己的知识，锤炼自己知识的想法与需求，有时学校或者学院也有很好的资源想要与学生分享却苦于没有分享路径与宣传平台，造成了人才的闲置与资源的浪费。

为此我们调研分析了大学生的实际需求，我们的应用解决了人才需求与人才供应的信息供给不合理的情况，

为了让人才可以在实践中磨砺锋芒，让资源得到合理的应用，让有想法的同学实现自己的创意，有技术的同学在实践中得到锻炼，提高自身竞争力

为了让有能力的同学实现自身价值，提高自身知识实践的能力， 让有创新意识创新能力的同学有更好的资源去实现自己的创新创业想法，因此我们从大学生学习生活入手深入分析大学生实际需求，研制了该平台系统

## 1.3本文主要内容

本系统是针对当代大学生创意项目开发需求与人才需求进行分析，并结合大学生的行为习惯开发的”WE创”创意项目开发平台。该平台让有创新抱负者轻松找到自己所需的人才资源，让有才能技术者有地方有资源来施展自身本领，让项目资源与人才资源更好的结合在一起。

本文第一章主要讲述了本课题研究的背景、意义和内容以及本课题的来源 等;第二章介绍了系统所需的技术支持，包括 MySQL 数据库的介绍、基于 Python 下的 django 框架，Bootstrap 前端框架技术等;第三章主要是对系统的需求和功能进行分析，之后根据需求和功能分析对系统模块进行了划分;第四章主要介绍了系统的逻辑结构设计，包括数据库设计与体系结构设计等;第五章主要讲述了系统的详细设计，主要包含本子系统中创意项目与个人中心模块。

## 1.4 本章小结

本章节主要介绍了“WE创”人才招募系统研发的相关背景、课题的来源和意义，并对大学生学习生活与国家提倡创新创业背景进行了相关的了解，最后对本文涉及的内容进行了概括。

# 第2章 实现技术

## 2.1 Python

Python是一种面向对象、直译式的计算机程序设计语言，也是一种功能强大的通用型语言，已经具有近二十年的发展历史，成熟且稳定。它包含了一组完善而且容易理解的标准库，能够轻松完成很多常见的任务。Python的语法非常简捷和清晰，与其它计算机程序设计语言最大的不同在于，它采用缩进来定义语句块。Python支持命令式编程、面向对象程序设计、函数式编程、面向切面编程、泛型编程多种编程范式。与Scheme、Ruby、Perl、Tcl等动态语言一样，Python具备垃圾自动回收功能，能够自动管理内存使用。Python经常被当作脚本语言用于处理系统管理任务和Web编程，然而它也非常适合完成各种高阶任务。Python虚拟机本身几乎可以在所有的操作系统中运行。使用一些诸如py2exe、PyPy、PyInstaller之类的工具可以将Python源代码转换成可以脱离Python解释器执行的程序。Python是支持类和多层继承等的面向对象编程技术。

Python有多种数据类型，代码能打包成模块和包，方便管理和发布，支持异常处理，能有效捕获和处理程序中发生的错误。[7]强大的动态数据类型支持，不同数据类型相加会引发一个异常。Python支持如生成器和列表嵌套等高级编程功能。自动内存碎片管理，有效利用内存资源。强大的类库支持，使编写文件处理、正则表达式，网络连接等程序变得相当容易。Python的交互命令行模块能方便地进行小代码调试和学习。Python易于扩展，可以通过C或C++编写的模块进行功能扩展。Python解析器可作为一个编程接口嵌入一个应用程序中。 Python可运行在多种计算机平台和操作系统中，如各位unix，linux,windows，MacOS,OS/2等等。 Python是开源的，可自由免费使用和发布，并且可用于商业用途以获取利润。

## 2.2 Django

Django 是一个可以使 Web 开发工作愉快并且高效的 Web 开发框架，基于 MVC 构造，使用 Django 能够以最小的代价构建以及维护高质量的 Web 应用。 Django 的主要目的是简便、快速的开发数据库驱动的网站。它强调代码复用,多 个组件可以很方便的以“插件”形式服务于整个框架，Django 有许多功能强大 的第三方插件，你甚至可以很方便的开发出自己的工具包。这使得 Django 具有 很强的可扩展性。Django 包括以下几个优点:

(1) 完美的官方文档。

(2) 全套的解决方案，Django 提供全套的解决方案，包括:cache、feed、orm、geo、auth 等，全部由 Django 自己创建，Django 将开发网站常用的 工具都做好了，使得开发效率与后期问题修改都十分可观。

(3 )强大的 URL 路由配置，Django 让你可以设计出非常优雅的 URL。

(4) 自助管理后台，admin interface 是 Django 里比较吸引眼球的一项contrib，让你几乎不用写一行代码就拥有一个完整的后台管理界面。

2.3 MVC 设计模式

MVC 设计模式的提出者是 Trygve Reenskaug，全名 Model View Controller， 是模型、视图、控制器三个英文首字母组成的缩写。MVC 最早是专门为编程语 言 SmallTalk 发明的一种设计模式，目前已被广泛应用，并慢慢变为交互式和界 面系统的根基。该模型把界面、逻辑以及数据三者相互分离开来，能给出很多不 同的视图，视图发生改变时无需重新编写业务逻辑。MVC 这种设计模式，逐渐 被应用在很多不同类型的分布式体系的规划和分析里面，而且还能够确认系统当 中每个模块相互之间的联系，因此受到越来越多的开发者的欢迎。

这个模式把系统分为了模型(Medel)、控制器(controller)以及视图(view) 三个模块，强制性的将输出、处理、输出分开，相互之间工作各自独立。

(1)模型。其对业务规则以及企业数据进行表征，为这三者当中处理任务 最多的。控制器控制的数据和视图的相识信息都保存在模型组件里，模型用来处 理业务逻辑，所有的核心数据都封装在模型里，同时完成功能所需的计算和逻辑 关系。模型返回的信息具有中立性，同时还和 I/O 没有关系，所以一种模型能够 把数据传送给不同的视图。

(2)控制器。这部分组件主要是负责用户输入的接收，同时根据对应的视 图以及模型对客户的需求做出应答，最终完成系统和用户之间的交互。控制器的 目的在于限制变动的蔓延，从而确保用户界面和视图两者之间没有差错。控制器 并不输出任何东西也不做任何处理，它只是发送输入数据，然后将处理的结果输 出。

(3)视图。也就是用来和用户相互交流的界面，能够很好地把经过处理的 结果生动的向客户展示出来，先接收结果之后再展示，未发生实质性的处理过程， 仅仅是简单的呈现数据。

三个模块相互独立之后，能够让很多视图显示成一个模型，因此假如某个模 型中的数据发生变动，和它有关系的以及能够牵连到的视图全部有可能将这些变 化展示给用户。所以，不管何时只要数据发生变化，都将被快速地反馈到所有的 视图，然后对新得到的结果进行显示。

**2.4** Bootstrap

Bootstrap 是 Twitter 推出的一个开源的用于前端开发的工具包。是一个 CSS/HTML 框架。 Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。该框架是基于jQuery 框架开发的，它在 jQuery 框架的基础上进行了更为个性化和人性化的完善，形成一套自己独有的网站风格，并兼容大部分 jQuery 插件。

该框架中还包含了丰富的 Web 组件，根据这些组件，可以快速的搭建一个漂亮、功能完备的网站。其中包括以下组件:下拉菜单、按钮组、按钮下拉菜单、 导航、导航条、面包屑、分页、排版、缩略图、警告对话框、进度条、媒体对象等。而且该框架还自带了 13 个 jQuery 插件，这些插件为 Bootstrap 中的组件赋予了 “生命”。其中包括:模式对话框、标签页、滚动条、弹出框等，而且更方便的是Bootstrap还可以定制自己的框架代码，可以对 Bootstrap 中所有的 CSS 变量进行修改，依据自己的需求裁剪代码。

## 2.5 JavaScript 与 jQuery

JavaScript 一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语 言，内置支持类型。它的解释器被称为 JavaScript 引擎，为浏览器的一部分，广 泛用于客户端的脚本语言，最早是在 HTML(标准通用标记语言下的一个应用) 网页上使用，用来给 HTML 网页增加动态功能。

JQuery 是继 prototype 之后又一个优秀的 Javascript 库。它是轻量级的 js 库 ， 它兼容 CSS3，还兼容各种浏览器(IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)， jQuery 使用户能更方便地处理 HTML(标准通用标记语言下的一个应用)、events、 实现动画效果，并且方便地为网站提供 AJAX 交互。jQuery 能够使用户的 html 页面保持代码和 html 内容分离，也就是说，不用再在 html 里面插入一堆 js 来调 用命令了，只需要定义 id 即可。

## 2.6 AJAX

AJAX 即“Asynchronous Javascript And XML”(异步 JavaScript 和 XML)， 是指一种创建交互式网页应用的网页开发技术。通过在后台与服务器进行少量数 据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页 的情况下，对网页的某部分进行更新。

2.7 本章小结

本章为相关技术介绍，介绍了系统采用的核心技术与开发工具，核心技术包 括了对 Python 语言的介绍与其特点，然后介绍了依托于 Python 语言的 Django 框架，介绍了 Bootstrap 与它的特点，AJAX 技术，JS 和 jQuery，以及什么叫 MVC 设计模式。

# 第3章 人才招募平台的需求分析和设计

## 3.1 系统需求分析

当今社会国家提倡创新，建设创新型国家，一些学生有创新意识与创新能力，但是创新实现有时候需要学科交叉，创新实现有时候会需要一些人才的资源，但是高等教育院校的学科交叉的交流比较少，有真正需求的学生有时很难找到符合要求的人才来帮助自己实现目标，无形中增加了创新的难度，  当今大学生课余时间较多，学习的多为书本知识，但是古语说得好，纸上的来终觉浅，绝知此事要躬行，学到的科学文化知识要在实践中应用才能真正理解吸收为自己的知识养分，所以在我们在对大学生进行深入调研后发现的问题与他们的需求，我们研发了该系统来解决这种人才需求与人才供给的问题。

该系统的主要面向对象是在校大学生，针对当前学生与学生之间的一个人才的供应与需求交流平台。学生在该平台既可以作为需求者，也可以作为被供应者，但是主要用户分为三种，作为需求者的学生可以在该平台发出招募，有相应技能的同学可以根据相应的招募要求并结合个人情况进行分析，如果合适则加入项目团队中。该平台还提供了项目与创意的展示平台，人们有好的创意可以拿出来共享，与他人讨论，产生更出色的创意。并结合相应的技术支持团队，来完成创意项目的孵化。

## 3.2 系统概要设计

（1）首页功能展示功能

 ① 功能描述

该功能是面向进入平台浏览的同学，在用户进入平台之后能够快速了解该平台的作用，包括该平台可以做什么事情，比如进行项目的申报，创意的申报，项目团队组成的招募，以及应该如何进行团队的组建，能够让用户了解该平台使用的流程。

（2）招募项目查看功能

 ① 功能描述

该模块主要是用户进入后进行项目的查看与浏览，用户应该可以在右侧的标签处进行筛选，可以查看相应的项目详情，比如项目的名称，项目的简介，招募人数以及已经招募的人数，截止日期等详细信息，对相应的项目感兴趣的话，可以进行点赞关注等操作，或者点击相应的项目进入项目详情查看招募情况，如果感觉自己合适，可以填写招募申请表，并点击提交。如果对项目感兴趣，可以进行评论等的操作。

 ② 约束

约束：

1. 没有注册登录，不能进行点赞、关注等操作。
2. 没有登录不能进行评论或者回复他人的评论。
3. 没有登录不能进行招募表的填写，不能申请加入项目。

（3）创意查看功能

 ① 功能描述

 ② 约束

1. 点赞或者关注操作必须在登录之后才能操作。
2. 评论或者回复他人评论要求用户必须已经登录

（5）团队展示功能

1. 必须先进行登录操作并且作为项目负责人才可以填写项目申请表，进行申请团队帮助
2. 必须进行登录操作才可以进行评论操作

用户在该平台应该可以查看各种创意，可以根据创意标签进行筛选操作，应该在创意展示页面可以看到创意的一些基本信息，比如创意的名称，创意的简介，创意的发起人，该创意获得的赞数，关注数，了解该创意的热度，并且应该可以在这里直接对创意进行点赞或者关注操作。而且也会针对每一个创意都会有一个相应的页面，在该页面会有该创意的详细信息，在项目详情的右侧应该会有针对该创意的推荐创意，在详情的下方还可以对该创意进行评论操作，还可以针对他人的评论进行回复操作。

① 功能描述

帮助团队模块是针对实验室平台开放的功能，在这里应该可以展示该平台的长处功能，应该有相应的标签对应团队，让用户可以根据标签来筛选团队，在团队详细页面应该展示该平台可以对其他的项目组提供的帮助。在这里展示该团队参与过的项目，获得的赞数，以及其他人对该团队的评价，如果评价很高，便可以相信该团队，并可以向该团队申请帮助，填写申请表，提交申请帮助表。

② 约束

## 3.3系统非功能性需求分析

### 3.3.1 技术可行性

在项目开始之前对项目可能会要到的技术进行了相应的调研，因为该项目为人才招募系统，主要是实现用户发布项目有，分享创意，提交加入申请审批申请等操作  ，所以该项目使用web技术进行相应的业务实现，主要需要web开发相关技能，现在web开发技术成熟，web框架相当成熟，我们决定对比了相应的几种开发语言后决定使用Python进行项目的开发，因为该语言已经非常成熟，而且相应的web框架也很多，可供选择性大，在硬件方面，市面上的主流电脑就能符合本系统开发的最低要求，所以综合考虑软件的技术可行性与硬件的开发需求可行性，本项目的技术可行性达到了要求，可以实习本系统的开发以及投入使用工作

针对技术可行性分析需要考虑到目前的技术条件是否可以帮助顺利完成系 统开发的相关工作，同时需要考虑软硬件系统的配置是否符合要求等。本系统主 要依赖于 Python 技术，现如今它已经十分成熟且功能强大。硬件方面，市面上 的主流电脑配置或者更低配置的计算机都符合本系统的开发要求，因此在技术层 面上也是可行的。软件方面，目前市面上网站开发的软件较多且功能强大。所以 在技术可行性方面可以实现本系统的开发以及投入使用工作。

### 3.3.2 经济可行性

在项目开发前便对该项目涉及到的人力物力资源进行过调查，搜集了相关的资料，该项目为web开发项目，不需要大量的硬件设施，是个纯软件的项目，所以在设备方面几乎没有成本，而且作为计算机专业的学生，开发的电脑不需要再次购买，在一定程度上降低了开发的成本，开发使用的各种软件经过调研发现在市场上都有相应的免费的版本或者可以免费试用，这样减轻了很大的物力财力的开销，从经济可行性方面来看本系统是可 以实现的。

### 3.34 系统安全性

系统的安全性主要是指系统安全、数据安全和网络安全程序安全性这四个 方面。

系统安全是指操作系统安全、数据库安全和应用服务器安全。操作系统安全 是指及时下载操作系统补丁，关闭不必要的端口和安装杀毒软件等。数据库安全 是指数据库用户名和密码设置以及用户权限设置，数据库的密码设置一定要复 杂，以保证密码不容易被暴力破解;针对不同用户，设置不同的数据库资源访问 权限。服务器安全是指系统采用 web 服务器的安全性。

数据安全是指数据备份与恢复和数据加密。数据备份与恢复是指系统的数据 库一定要严格定期进行备份，以防止数据库系统所在硬盘突然被破坏或者遭受攻 击，导致数据库系统数据被破坏，是数据库不能正常运行，此时需要对数据库系 统进行数据恢复，而之前的备份正好派上用场。

网络安全是指数据在网络传输过程中的安全性。保密性要求高的数据要严格 采用加密技术，使用密文在网络中进行传输;在操作系统中安装防火墙，以抵御 恶意的网络攻击;进行网络安全监控，定期进行网络安全监测。

程序安全是指代码安全、资源访问控制和程序日志。代码安全是指代码中存 在的安全漏洞，这些漏洞很有可能被黑客利用破坏系统，所以代码开发测试阶段 一定要小心避免和发现漏洞。资源访问是指不同级别用户可以访问到不同权限资 源，一定要严格控制，以防破坏系统数据。程序日志是指系统运行过程中要记录 用户的系统操作日志，当系统数据或者系统运行出现问题时，可以根据日志记录 进行数据恢复和发现问题。

## 3.4 本章小结

本章主要介绍了大学生招募平台的 功能性需求分析与非功能性需求分析、功能性需求分析包括对系统模块与功能的 分析，其中功能分析包括平台首页、招募项目、创意灵感、个人中心。 非功能性需求分析主要包括对系统开发过程中经济可行性、技术可行性、系统安全性的分析。

# 第4章 数据库设计

## 4.1数据库概念结构设计E-R图

本系统主要有以下实体联系：

(1) 一对一联系(1 ∶1)

用户—创意是1:1的联系，一个用户发表一个创意

(2) 一对多联系(1 ∶N)

项目—用户是1：N的联系，一个项目对应多个学生。

标签--图片是1：N的联系，一个管理员管理多张图片。

(3)多对多联系(M ∶N)

用户—项目是N：M的联系，多个用户可以同时参与多个业务。

用户—评论是N：M的联系，多个用户可以同时拥有多个评论。

点赞记录—用户类别是N：M的联系，多个用户可以同时拥有多个点赞记录。

项目—标签类型是N：M的联系，多个项目可以同时拥有多个标签。

用户标签—项目标签是N：M的联系，多个用户标签可以同时对应多个项目标签。

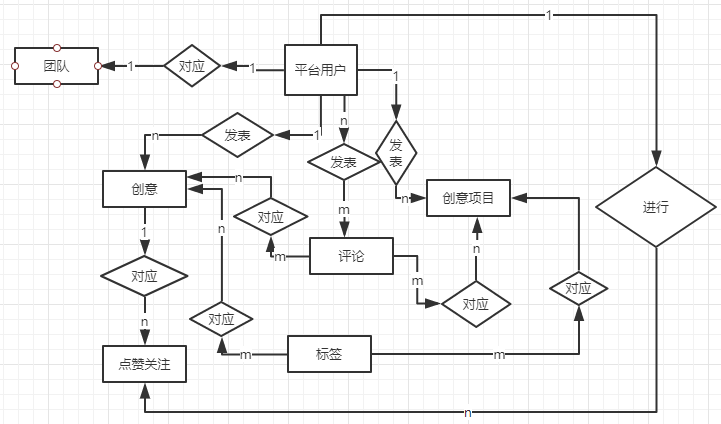


Figure 4 E-R图

E-R图也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。根据以上实体关系描述转换为E-R图，如Figure 4 E-R图所示。

## 4.2数据库逻辑及物理结构设计

(1)项目表

记录项目的详细信息，如项目id，项目名称等，以及项目的状态

表格 1项目表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| Id | 项目Id | char(3) | 主 |  |  |  |  |
| ProjectName | 项目名称 | Char(20) |  |  |  |  | 不可重 |
| StartTime | 开始时间 | dateTime |  |  |  |  |  |
| EndTime | 结束时间 | dateTime |  |  |  |  |  |
| Description | 描述 | Text | 2 |  |  | 项目描述 |  |
| Number | 人数 | Int(3) |  |  |  |  |  |
| State | 状态 | Int(1) |  |  |  | 0招募中  1正在进行  2 补招  3 已完成  4 烂尾项目 |  |
| Img | 照片 | char(32) |  |  |  |  |  |
| Summary | 总结 | Text |  |  |  |  |  |
| Progress | 进度描述 | Text |  |  |  |  |  |

数据模型如下

class Project(models.Model):

    Id = models.AutoField(primary\_key=True)

    ProjectName = models.CharField(null=False, max\_length=20, unique=True)

    Description = models.TextField(null=False)

    StartTime = models.DateTimeField(auto\_now\_add=True)

    EndTime = models.DateTimeField(null=False)

    Statue = models.PositiveIntegerField(default=0)

    Number = models.PositiveIntegerField(default=1)

    Img = models.ImageField(upload\_to='photos/%Y/%m/%d/project', null=True)

    Summary = models.TextField(null=True)

    Progress = models.TextField(null=True)

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

    def \_\_unicode\_\_(self):

        return self.ProjectName

(2)项目成员表

项目中存在的各个成员都在这个表里找到，这里标示了项目成员的身份，以及在项目中的评价

表格 2项目成员表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| **Id** | **项目成员Id** | **AutoField** | **主** |  |  |  |  |
| **user** | **用户Id** | **char(36)** | **外** |  | **User** |  |  |
| **project** | **项目Id** | **Autofield** | **外** |  | **Project** |  |  |
| **Identity** | **身份** | **Int(1)** |  |  |  | **0** **发起人**  **1** **指导教师**  **2** **参与者**  **3** **淘汰者**  **4** **主动退出** |  |
| **Evaluation** | **评价** | **Text** |  |  |  |  |  |

数据模型

class ProjectUser(models.Model):

    Id = models.AutoField(primary\_key=True)

    user = models.ForeignKey(User, related\_name='ProjectUser\_User\_set')

    project = models.ForeignKey(Project, related\_name='ProjectUser\_Project\_set', null=False)

    Identity = models.PositiveIntegerField(default=0)

    Evaluate = models.TextField()

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

    def \_\_unicode\_\_(self):

        return self.project

(3)项目标签表

表格 3项目标签

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| **Id** | **项目标签ID** | **Autofield** | **主** |  |  |  |  |
| **ProjectLabelName** | **项目标签名称** | **Char(10)** |  |  |  |  | **不可重** |
| **IsUse** | **是否可用** | **BoolField** |  |  |  |  | **True** **为可用（默认）** **，False为不可用** |

数据模型

class ProjectLabel(models.Model):

    Id = models.AutoField(primary\_key=True)

    ProjectLabelName = models.CharField(max\_length=20, null=False, unique=True)

    IsUse = models.BooleanField(default=True)

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

    def \_\_unicode\_\_(self):

        return self.ProjectLabelName

(4)项目-项目标签表

表格 4项目-项目标签表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** |  | **外键主表** | **备注** | **重复性** |
| **Id** | **项目项目标签Id** | **AutoField** | **主** |  |  |  |  |  |
| **ProjectId** | **项目** |  | **外** |  |  |  |  |  |
| **projectLabel** | **项目标签表** | **Char** | **外** |  |  |  |  |  |

数据模型

class Project2ProjectLabel(models.Model):

    Id = models.AutoField(primary\_key=True)

    projectLabel = models.ForeignKey(ProjectLabel, related\_name='Project2ProjectLabel\_ProjectLabel\_set', null=False)

    project = models.ForeignKey(Project, related\_name='Project2ProjectLabel\_Project\_set')

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

    def \_\_unicode\_\_(self):

        return self.projectLabel.\_\_unicode\_\_() + '  ' +self.project.\_\_unicode\_\_()

(5)招募帖

表格 5招募帖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| **Id** | **招募贴Id** | **AutoField** | **主** |  |  |  |  |
| **project** | **项目Id** | **AutoField** | **外** |  | **项目表** |  |  |
| **StartTime** | **开始时间** | **dateTime** |  |  |  |  |  |
| **EndTime** | **结束时间** | **dateTIme** |  |  |  |  |  |
| **Description** | **描述** | **Text** |  |  |  |  |  |
| **State** | **状态** | **Bit** |  |  |  | **招募帖是否关闭** |  |
| **Times** | **次数** | **Int(2)** |  |  |  |  |  |
| **PredictNumber** | **预计招募人数** | **Int(5)** |  |  |  |  |  |
| **RecruitedNumber** | **已招募人数** | **Int(5)** |  |  |  |  |  |

数据模型

class Recruit(models.Model):

    Id = models.AutoField(primary\_key=True)

    project = models.ForeignKey(Project, related\_name='Recruit\_Project\_set', null=False)

    StartTime = models.DateTimeField(auto\_now\_add=True)

    EndTime = models.DateTimeField(null=True)

    Describe = models.TextField(null=False,)

    State = models.PositiveIntegerField(default=0) ##

    Times = models.PositiveIntegerField(default=1)

    PredictNumber = models.PositiveIntegerField(default=1)

    RecruitedNumber = models.PositiveIntegerField(default=0)

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

（6）申请表

表格 6申请表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| **Id** | **申请表Id** | **AutoField** | **主** |  |  |  |  |
| **recruit** | **招募贴Id** | **AutoField** | **外** |  | **招募贴表** |  |  |
| **user** | **申请者** | **AutoField** | **外** |  | **用户表** |  |  |
| **Describe** | **描述** | **Text** |  |  |  |  |  |
| **State** | **状态** | **Bit** |  |  |  | **0** **未读**  **1已读**  **2** **拒绝**  **3接受** |  |
| **SendTime** | **发出时间** | **DateTime** |  |  |  |  |  |
|  | **反馈信息** | **Text** |  |  |  | **对申请者的反馈** |  |

数据模型

class Apply(models.Model):

    Id = models.AutoField(primary\_key=True)

    user = models.ForeignKey(User, related\_name='Apply\_User\_set', null=False)

    recruit = models.ForeignKey(Recruit, related\_name='Apply\_Recruit\_set', null=False)

    Describe = models.TextField(null=True)

    State = models.PositiveIntegerField(default=0)

    SendTime = models.DateTimeField(auto\_now\_add=True)

Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

(7)创意表

表格 7创意表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| **Id** | **创意表Id** | **AutoField** | **主** |  |  |  |  |
| **Name** | **创意名称** | **Char(30)** |  |  |  |  |  |
| **Description** | **描述** | **Text** |  |  |  |  |  |
| **user** | **用户Id** | **AutoField** | **外** |  | **User** |  |  |
| **Date** | **发布时间** | **Datetime** |  |  |  |  |  |
| **IsUse** | **是否可用** | **Bool** |  |  |  |  |  |
| **Img** | **照片** | **Char(32)** |  |  |  |  |  |
| **LastChange** | **最近更改时间** |  |  |  |  |  |  |

数据模型

class Creation(models.Model):

    Id = models.AutoField(primary\_key=True)

    Date = models.DateTimeField(auto\_now\_add=True)

    user = models.ForeignKey(User, related\_name='Creation\_User\_set')

    LastChange = models.DateTimeField(auto\_now=True)

    Describe = models.TextField(null=True)

    Name = models.CharField(max\_length=20, null=False)

    IsUse = models.BooleanField(default=True)

    Img = models.ImageField(upload\_to='photos/%Y/%m/%d/user')

Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

(8)创意-项目标签表

表格 8创意-项目标签表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| **Id** | **创意-项目标签Id** | **AutoField** | **主** |  |  |  |  |
| **creation** | **创意表** | **Char(30)** | **外** |  |  |  |  |
| **ProectLabel** | **描述** | **Text** |  |  |  |  |  |

数据模型

class Creation2ProjectLabel(models.Model):

    Id = models.AutoField(primary\_key=True)

    creation = models.ForeignKey(Creation,related\_name='Creation2ProjectLabel\_Creation\_set', null=False)

    projectLabel = models.ForeignKey(ProjectLabel, related\_name='Creation2ProjectLabel\_ProjectLabel\_set', null=False)

Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

(9)举报表

表格 9举报表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| Id | 分值表Id | **AutoField** | **主** |  |  |  |  |
| user | 举报人 |  | **外** |  |  |  |  |
| project | 项目表 |  | **外** |  |  |  |  |
| creation | 创意表 |  | **外** |  |  |  |  |
| comment | 评论表 |  | **外** |  |  |  |  |
| ReportDate | 举报日期 |  |  |  |  |  |  |
| DealTime | 处理日期 | **Date** |  |  |  |  |  |
| State | 处理状态 | **Int** |  |  |  |  |  |

数据模型

class Report(models.Model):

    Id = models.AutoField(primary\_key=True)

    user = models.ForeignKey(User, related\_name='Report\_User\_set', null=False)

    project = models.ForeignKey(Project, related\_name='Report\_Project\_set', null=True)

    creation = models.ForeignKey(Creation, related\_name='Report\_Creation\_set', null=True)

    comment = models.ForeignKey(Comment, related\_name='Report\_Comment\_set', null=True)

    Reason = models.TextField()

    ReportDate = models.DateTimeField(auto\_now\_add=True)

    DealTime = models.DateTimeField(null=True)

    State = models.PositiveIntegerField(default=0)

Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

(10)评论表

表格 10评论表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| Id | 评论表Id | **AutoField** | **主** |  |  |  |  |
| user | 评论者 |  | **外** | **可空** |  |  |  |
| creation | 创意表 |  | **外** | **可空** |  |  |  |
| project | 项目表 |  | **外** | **可空** |  |  |  |
| Date | 评论日期 | **Date** |  |  |  |  |  |
| Content | 评论内容 | **Text** |  |  |  |  |  |
| IsUse | 是否可用 | **Bool** |  |  |  |  |  |
| commited\_user | 被评论的用户 |  | **外** | **可空** |  |  |  |
| IsAdopt | 是否采纳 | **Bool** |  |  |  |  |  |
| commentedId | 被评论ID |  |  |  |  |  |  |

数据模型

class Comment(models.Model):

    Id = models.AutoField(primary\_key=True)

    user = models.ForeignKey(User, related\_name='Comment\_User\_set', null=False)

    creation = models.ForeignKey(Creation, related\_name='Comment\_Creation\_set', null=True, blank=True)

    project = models.ForeignKey(Project, related\_name='Comment\_Project\_set', null=True, blank=True)

    commited\_user = models.ForeignKey(User, related\_name='commited\_user\_set', null=True, blank=True)

    commentedId = models.UUIDField(null=True, blank=True)

    Date = models.DateTimeField(auto\_now\_add=True)

    Content = models.TextField()

    IsUse = models.BooleanField(default=True)

    IsAdopt = models.BooleanField(default=False)

    Isreply = models.BooleanField(default=False)

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

(11)关注表

表格 11关注表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| Id | 关注表Id | **AutoField** | **主** |  |  |  |  |
| user | 关注用户 |  | **外** |  |  |  |  |
| project | 项目表 |  | **外** |  |  |  |  |
| creation | 创意表 |  | **外** |  |  |  |  |
| Follower | 被关注用户 |  | **外** |  |  |  |  |

数据模型

class Follow(models.Model):

    Id = models.AutoField(primary\_key=True)

    user = models.ForeignKey(User, related\_name='Follow\_User\_set', null=False)

    project = models.ForeignKey(Project, related\_name='Follow\_Project\_set', null=True, blank=True)

    creation = models.ForeignKey(Creation, related\_name='Follow\_Creation\_set', null=True, blank=True)

    Follower = models.ForeignKey(User, related\_name='Follow\_Follower\_set', null=True, blank=True)

Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1())

(12)用户表

表格12用户表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **名称** | **类型** | **键** | **可空** | **外键主表** | **备注** | **重复性** |
| Id | 用户Id | char(36) | 主 |  |  |  |  |
| UserName | 用户名 | Char(10) |  |  |  |  |  |
| Account | 账号 | Char(20) |  |  |  | 教师则为工号  学生则为学号 | 不可重 |
| PassWord | 密码 | Char(20) |  |  |  |  |  |
| Identity | 身份 | bit |  |  |  | 1为教师  0为学生  2为团队用户 |  |
| Score | 分值 | Int(5) |  |  |  |  |  |
| Img | 照片 | Char(32) |  | 可空 |  |  |  |
| Sex | 性别 | Bit |  |  |  | 0为女  1为男 |  |
| Email | 邮箱 | Emailfield |  |  |  |  | 不可重 |
| Phone | 联系电话 | Char(10) |  | 可空 |  |  | 不可重 |
| RegistTime | 注册时间 | DateTtime |  |  |  |  |  |
| Institude | 学院 | Char(20) |  | 可空 |  |  |  |
| Major | 专业 | Char(20) |  | 可空 |  | 老师无专业 |  |
| School | 学校 | Char(20) |  | 可空 |  |  |  |
| Introduction | 介绍 | Text |  |  |  |  |  |

数据模型

class User(models.Model):

    Id = models.AutoField(primary\_key=True)

    UserName = models.CharField(max\_length=10, null=False, unique=True)

    Account = models.CharField(max\_length=20, null=True, unique=True)

    PassWord = models.CharField(max\_length=20, null=False)

    Identity = models.PositiveSmallIntegerField(default=0, null=False)

    Sex = models.PositiveSmallIntegerField(default=0, null=False)

    Email = models.EmailField(null=False, max\_length=32, unique=True)

    Score = models.PositiveIntegerField(default=0, null=False)

    RegistTime = models.DateTimeField(auto\_now\_add=True)

    Phone = models.CharField(null=True, blank=True, max\_length=25)

    Img = models.ImageField(upload\_to='photos/%Y/%m/%d/user', null=True, blank=True)

    Introduction = models.TextField(null=True)

    School = models.CharField(null=True, max\_length=20)

    Institude = models.CharField(null=True, max\_length=20)

    Major = models.CharField(null=True, max\_length=20)

    Uuid = models.UUIDField(null=True, blank=True, default=str(uuid.uuid1()))

## 4.3本章小结

本章主要介绍了系统的数据库设计，包括数据库概念结构设计、逻辑结构设 计以及数据库模型图，对系统中各实体属性以及实体之间的关系通过 E-R 图的形 式表现出来，在 E-R 图的基础上创建数据字典，建立完整的表结构

第5章 系统详细设计与实现

## 5.1开发工具与环境

### 5.1.1 MySQL 数据库

， MySQL 是一种关联数据库管理系统，它具有客户机/服务器体系结构的分布式数 据库管理系统，它是一个真正的多用户、多线程 SQL 数据库服务器。其速度的 增加与灵活性的提高主要体现在数据的保存方式上，与将所有数据放在一个大仓 库中的方式不同，关联数据库是将数据保存在不同的表中。MySQL 中 SQL 表示 “结构化查询语言”，在访问数据库这一行为中，SQL 是最常用标准化语言。MySQL它拥有优化的 SQL 查询 算法，有效的提高了查询的速度，它是一个可以处理拥有上千万条记录的大型数 据库。MySQL 具备以下几个特点:

(1)开源。MySQL 数据库不仅开源免费，同时它还能够对其源代码进行自 行修改与完善。

(2)该数据库能适应较低的网络带宽，使用者无需花销很多就能够安全稳 定的使用该数据库。

(3)MySQL 数据库为关系型数据库，在经过了高度的优化之后，与其他数 据库相比查询速度更快。

(4)该数据库具有较多的管理工具，数据库的恢复以及备份都能够很轻松 的完成。

(5)该数据库包含各种各样的数据引擎，能够给客户提供丰富的选择。 (6)MySQL 数据库操作简单，容易学习，能够快速上手。

### 5.1.2 Redis

Redis是一个使用ANSI C编写的开源、支持网络、基于内存、可选持久性的键值对存储数据库。

Redis 优势

性能极高 – Redis能读的速度是110000次/s,写的速度是81000次/s 。

丰富的数据类型 – Redis支持二进制案例的 Strings, Lists, Hashes, Sets 及 Ordered Sets 数据类型操作。

原子 – Redis的所有操作都是原子性的，意思就是要么成功执行要么失败完全不执行。单个操作是原子性的。多个操作也支持事务，即原子性，通过MULTI和EXEC指令包起来。

丰富的特性 – Redis还支持 publish/subscribe, 通知, key 过期等等特性。

### 5.1.3 Sublime Text 3

Sublime Text 是一个代码编辑器，也是 HTML 和散文先进的文本编辑器。Sublime Text 是由程序员 Jon Skinner 于 2008 年 1 月份所开发出来，它最初被设计为一个具有丰富扩展功能的 Vim。

Sublime Text 具有漂亮的用户界面和强大的功能，例如代码缩略图，Python 的插件，代码段等。还可自定义键绑定，菜单和工具栏。Sublime Text 的主要功 能包括:拼写检查，书签，完整的 Python API ， Goto 功能，即时项目切换， 多选择，多窗口等等。Sublime Text 是一个跨平台的编辑器，同时支持 Windows、 Linux、Mac OS X 等操作系统。其主要优点如下:

(1)主流前端开发编辑器

(2)体积较小，运行速度快

(3)文本功能强大 (4)支持编译功能且可在控制台看到输出

(5)内嵌 python 解释器支持插件开发以达到可扩展目的 (6)Package Control:ST 支持的大量插件可通过其进行管理

### 5.1.4 PyCharm

PyCharm是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。

• 代码分析与辅助功能，拥有补全代码、高亮语法和错误提示；

• 项目和代码导航：专门的项目视图，文件结构视图和和文件、类、方法和用例的快速跳转；

• 重构：包括重新命名，提取方法，引入变量，引入常量、pull，push等；

• 支持网络框架： Django, web2py 和 Flask；

• 集成Python 调试器；

• 集成单元测试，按行覆盖代码；

• Google App Engine下的Python开发；

• 集成版本控制系统：为Mercurial， Git， Subversion, Perforce 和 CVS提供统一的用户界面，拥有修改以及合并功能。

### 5.1.5 Axure RP

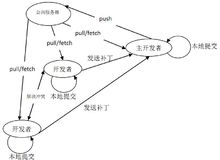
Axure RP 是一个专业的快速原型设计工具，让负责定义需求和规格、设计 功能和界面的人员能够快速创建应用软件或 Web 网站的线框图、流程图、原型 和规格说明文档。作为本课题的原型设计工具，它能快速、高效的创建原型，同 时支持多人协作设计和版本控制管理。

### 5.1.6 Git

git是一个[分布式版本控制](https://zh.wikipedia.org/wiki/%E5%88%86%E6%95%A3%E5%BC%8F%E7%89%88%E6%9C%AC%E6%8E%A7%E5%88%B6)软件，

分布式相比于集中式的最大区别在于开发者可以提交到本地，每个开发者通过克隆（git clone），在本地机器上拷贝一个完整的Git仓库。

下图是经典的git开发过程。

[[](https://baike.baidu.com/pic/GIT/12647237/0/a71ea8d3fd1f4134ca7667d8251f95cad0c85ed6?fr=lemma&ct=single)](https://baike.baidu.com/pic/GIT/12647237/0/a71ea8d3fd1f4134ca7667d8251f95cad0c85ed6?fr=lemma&ct=single)

Git的功能特性：

1、从服务器上克隆完整的Git仓库（包括代码和版本信息）到单机上。

2、在自己的机器上根据不同的开发目的，创建分支，修改代码。

3、在单机上自己创建的分支上提交代码。

4、在单机上合并分支。

5、把服务器上最新版的代码fetch下来，然后跟自己的主分支合并。

6、生成补丁（patch），把补丁发送给主开发者。

7、看主开发者的反馈，如果主开发者发现两个一般开发者之间有冲突（他们之间可以合作解决的冲突），就会要求他们先解决冲突，然后再由其中一个人提交。如果主开发者可以自己解决，或者没有冲突，就通过。

8、一般开发者之间解决冲突的方法，开发者之间可以使用pull 命令解决冲突，解决完冲突之后再向主开发者提交补丁。

从主开发者的角度（假设主开发者不用开发代码）看，git有以下功能：

1、查看邮件或者通过其它方式查看一般开发者的提交状态。

2、打上补丁，解决冲突（可以自己解决，也可以要求开发者之间解决以后再重新提交，如果是开源项目，还要决定哪些补丁有用，哪些不用）。

3、向公共服务器提交结果，然后通知所有开发人员。

优点：

适合[分布式开发](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E5%BC%80%E5%8F%91)，强调个体。

公共服务器压力和数据量都不会太大。

速度快、灵活。

任意两个开发者之间可以很容易的解决冲突。

## 5.2 用户注册与登录模块

### 5.2.1 用户注册

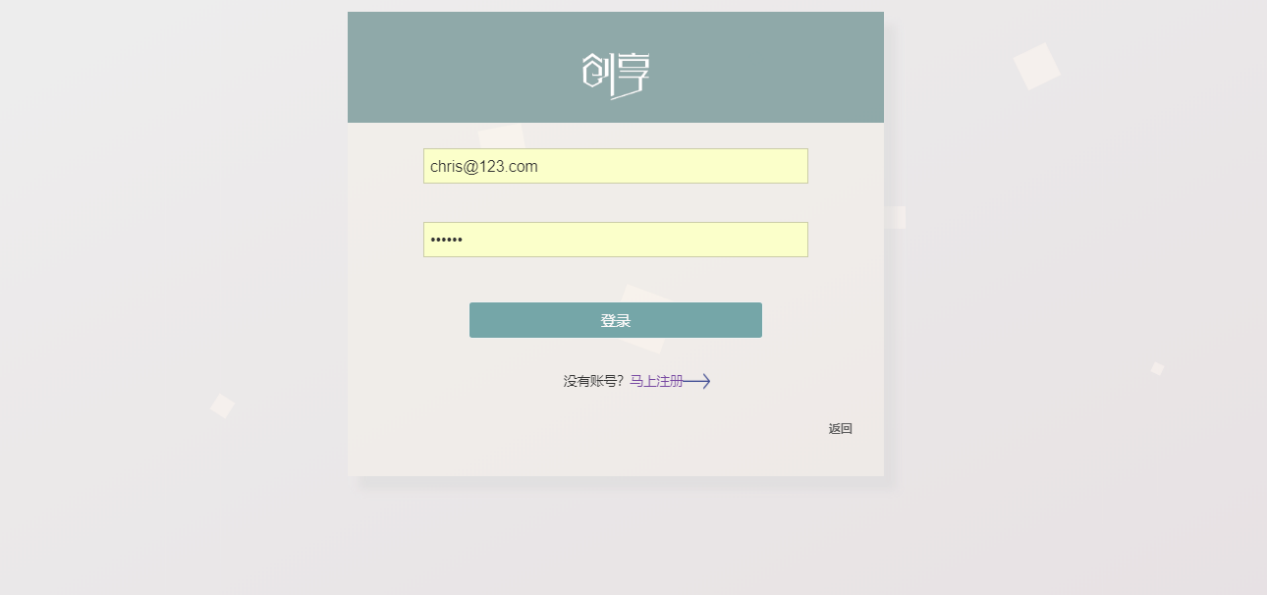
用户在使用该平台时需要注册成为该平台的用户，这样用户才可以享受在该平台进行各种点赞关注与评论的操作，

在这里输入用户名，用户的注册邮箱，密码， 确认密码，点击获取验证码，收取验证码并将验证码输入验证码框内即可完成注册操作。

用户注册时需要点击 我已阅读并同意《用户协议》，用户在使用该平台的服务的时候需要提前阅读该用户协议，因为该平台面向大众学生并且开放评论与创建项目创意的服务，所以需要用户在使用该平台时确保知道如何正确使用该平台。

而且用户注册使用的邮箱也是唯一的邮箱，如果使用相同邮箱多次注册用户，将会受到制止，不能注册成功。

### 5.2.2 用户登录

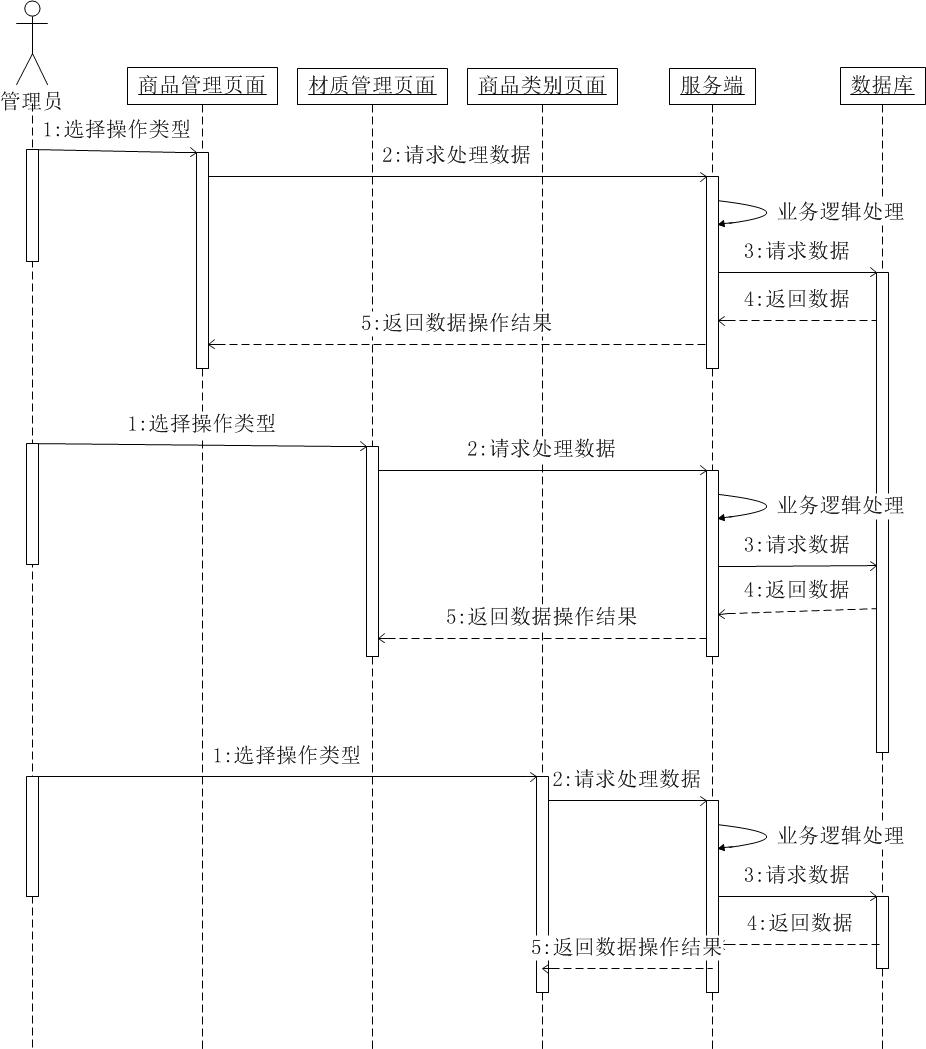


用户在注册成为该平台用户的后使用该系统内之前，通过输入自己注册的用户名与密码，点击登录按钮，即可进行登录该系统，如果用户没有用户名密码，可以点击马上注册进入到注册登录页面进行注册操作，如果用户只是想要观看浏览该平台，可以点击返回按钮返回主页面进行平台项目与创意的浏览。

## 5.3 系统详细设计

本系统属于面向客户定制的印章电子商务系统，将系统分为前台与后台两大 部分:前台面向客户实现客户的在线查询、定制、下单等操作;后台面向管理员， 实现对平台的管理维护，包括用户的管理、商品具体信息的维护、网站公告等， 为网站的正常运营提供有效的保障。本文涉及的子系统主要涉及五大模块:后台 商品管理模块、前台商品列表模块、后台客户信息模块、前台登录注册模块、

及前台已登录用户的个人中心模块。商品管理与客户信息子系统功能模块图如图 5.2.1 所示:



### 5.3.1平台展示

平台的首页主要用于展示平台的使用方法与使用流程，让大家最快的了解平台的使用方法



picture 1平台展示

### 5.3.2 优秀项目、优秀创意展示



picture 2项目、创意展示

这里可以看到推荐的项目与创意，不用费事就能最快看到最新最好的创意与项目

## 5.4招募项目模块

### 5.4.1招募项目展示

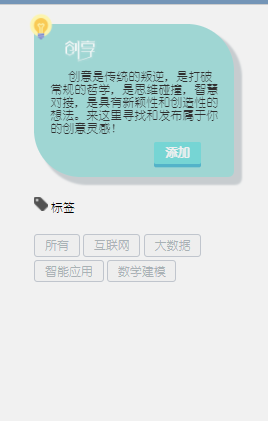
在项目招募模块中，用户可以查看浏览所有正在处于招募状态的项目，并获悉项目招募状态，截止日期，所需人员技能，了解项目基本情况。也可以通过项目标签筛选自己感兴趣的项目，在项目详细内容界面可以对项目进行点赞、关注操作，如果发现项目中包含不当内容，可及时进行举报，后台管理员审核后进行相应处理。在招募项目的详细内容页面用户可以查看招募贴内容，招募剩余时间以及招募进度，如果发现感兴趣的项目并且自身符合招募帖要求，可以立即申请加入，填入申请内容以及对该项目自己的想法看法，自己可以如何为整个项目做出贡献等，点击提交就可以等待项目发起人的回复。如果对项目感兴趣可以在评论区进行评论，或者对他人的评论进行回复，如果发现不当言论，可以进行相应的举报，后台管理员会进行相应的查明并尽快处理。



picture 3招募项目

### 5.4.2项目详细信息

用户如果想要查找自己感兴趣的项目，可以在右侧的标签处进行筛选，点击相应的标签，相应标签的项目便会展示出来，在该页面可以直接查看到相应的项目信息，比如项目起止时间，项目状态，招募状态等信息，如果对该项目感兴趣，可以进入该项目中查看相应的招募贴。



picture 4项目筛选

再对项目筛选后选定相应的项目，点击进行查看，可以看到项目的详细信息



picture 5项目详细信息

### 5.4.3评论与申请招募

如果用户对项目感兴趣，可以进入相应的项目中进行查看，查看项目的详细信息与招募帖的信息，能够对招募帖有充分的认识，如果对该招募帖感兴趣，可以点击申请加入按钮，填写相应的信息进行申请操作。如果对该项目感兴趣，也可以在下方的评论专区进行评论操作，或者回复他人的评论



picture 6项目详细信息

如果对项目感兴趣，或者对招募贴里的技能自己比较有信心，可以点击申请加入进入申请理由的界面，填写申请理由，点击提交便生成了一条申请记录，该申请会发送到项目发起人那里，被审批。如picture 7申请理由 所示



picture 7申请理由

## 5.5.创意分享模块

### 5.5.1 创意分享展示模块

在创意分享模块中，用户可以查看浏览所有的有趣的创意，可以通过创意标签进行筛选操作，筛选出自己感兴趣的创意，如果对创意感兴趣还可以直接进行点赞或者关注操作，如果想要详细了解创意信息，可以在创意详情界面可以对创意的详细信息进行查看，也可以对该创意进行点赞、关注操作，如果相对该创意进行评论，可以在评论区进行评论，还可以针对别人的评论进行回复，如果发现创意中包含不当内容，或者评论有不当的内容，可及时进行举报，后台管理员审核后进行相应处理。



picture 8创意展示

如果展示的创意已经浏览完，可以点击创意下方的加载更多来加载更多的创意，来完成创意浏览。



picture 9创意加载

### 5.5.2 创意详情模块

如果对相应的创意感兴趣，可以点击相应的创意，进入相应的创意界面，在详情界面可以查看详细的创意信息，还可以看到创意的标签，也可以在这里对创意进行点赞与关注，如果对该创意感兴趣，可以在下方对创意进行评论，以及对他人的评论进行回复。在右侧也针对相应的创意有推荐创意，可以点击进入查看其他的创意。



picture 10创意详细信息

进入详情页面可以查看相应的创意信息，以及多少人对该创意进行过点赞，该创意的热度



picture 11评论专区

在评论专区进行评论，也可以对他人的评论进行回复，感受评论的趣味，在右侧还有相应的推荐创意，在这里可以随时进行查看相应的标签的创意，享受在创意中穿梭的乐趣。

## 5.7 个人中心模块

### 5.7.1 用户信息模块

在用户登录系统后，在页面右上角有用户图像，用户可以通过点击用户头像进入用户个人中心，如果想要退出登录，可以点击头像右侧的退出按钮退出登录。

进入用户个人中心后，可以点击编辑资料按钮进入个人资料页面。

在 编辑资料界面-个人信息处 可以修改用户昵称，所在学院，所属学校，主修专业的信息。点击保存按钮进行个人信息的保存。

点击账号信息，可以对账号信息进行修改保存工作，可以修改手机号码，更改用户头像，点击选择文件按钮，选择用户头像图片，点击保存按钮即可保存相应修改的信息。

更改密码

用户可以在个人中心修改自己的密码。输入旧密码，输入新密码与确认密码，点击提交按钮，如果输入的旧密码与用户密码相同，则可以修改成功，用户可以通过新密码进入平台，如果用户输入的旧密码与用户当前密码不同，则密码修改失败

修改标签

用户标签用于给用户提供项目与创意推荐，用户可以选择自己感兴趣或者自己正在涉足的领域标签，这样可以在项目推荐与展示上优先展示这些项目与创意。标签分为两行，第一行为已选择标签，第二行为待选择标签，选择标签时，点击想要选取的标签，点击保存即可，删除已经删除的标签，双击需要删除的标签即可删除。最中点击保存即可保存用户设置。

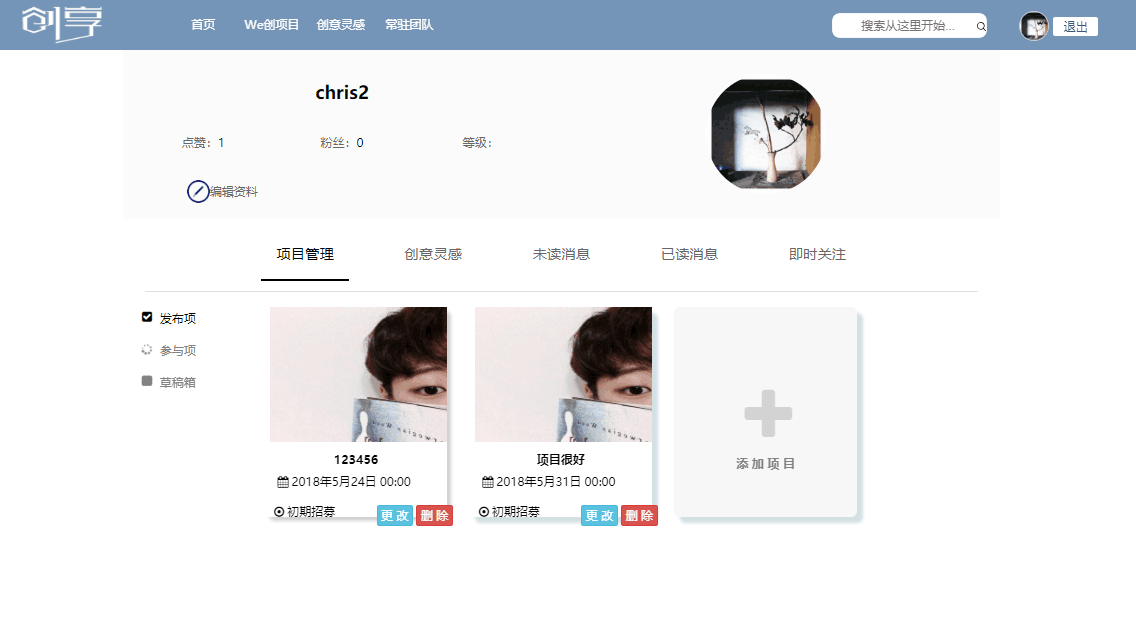
对个人信息修改完善之后可以点击返回主页按钮，即可返回个人信息主页。





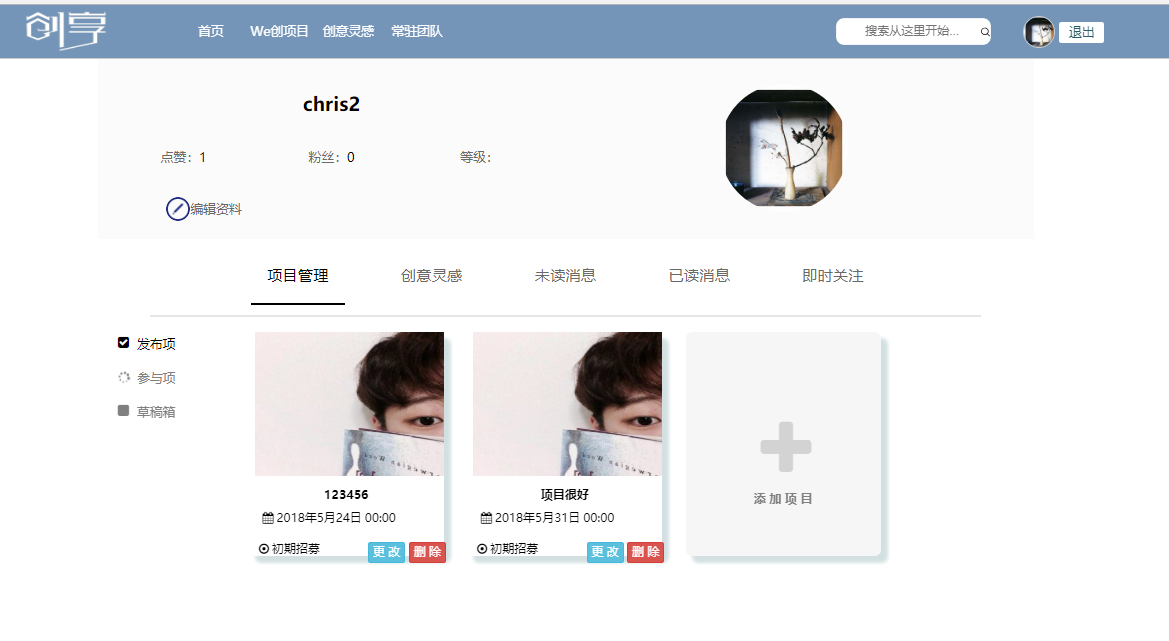




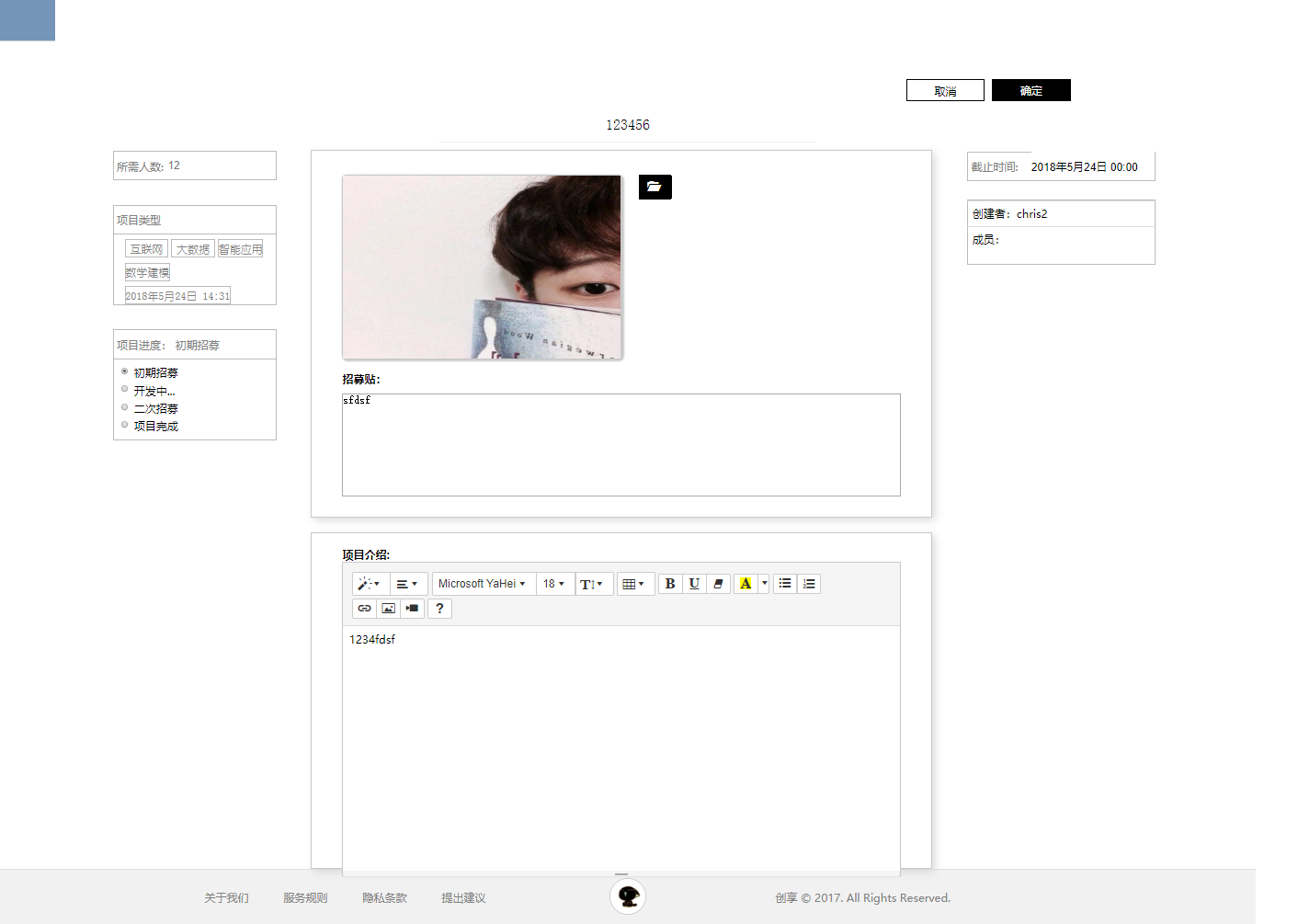
****

### 5.7.2个人项目模块

用户在个人中心处可以进行项目管理，包括项目的修改删除，项目的新增，筛选等操作。在个人中心页面点击项目管理页面，即可进入项目管理，首先展示的是用户自己创建的项目，展示信息包括项目名称，项目创建时间与项目状态，如果用户有好的项目，可以点击右侧的添加项目的按钮，即可进入项目配置页面进行项目的创建。



在不同项目中都包含更改与删除按钮，点击更改按钮，即可进入项目详情，如果需要进行相应的修改，

如图所示，输入项目名称，所需人数，选择项目类型，项目进度，输入项目介绍与招募贴内容等，点击确认按钮即可进行项目的发布。

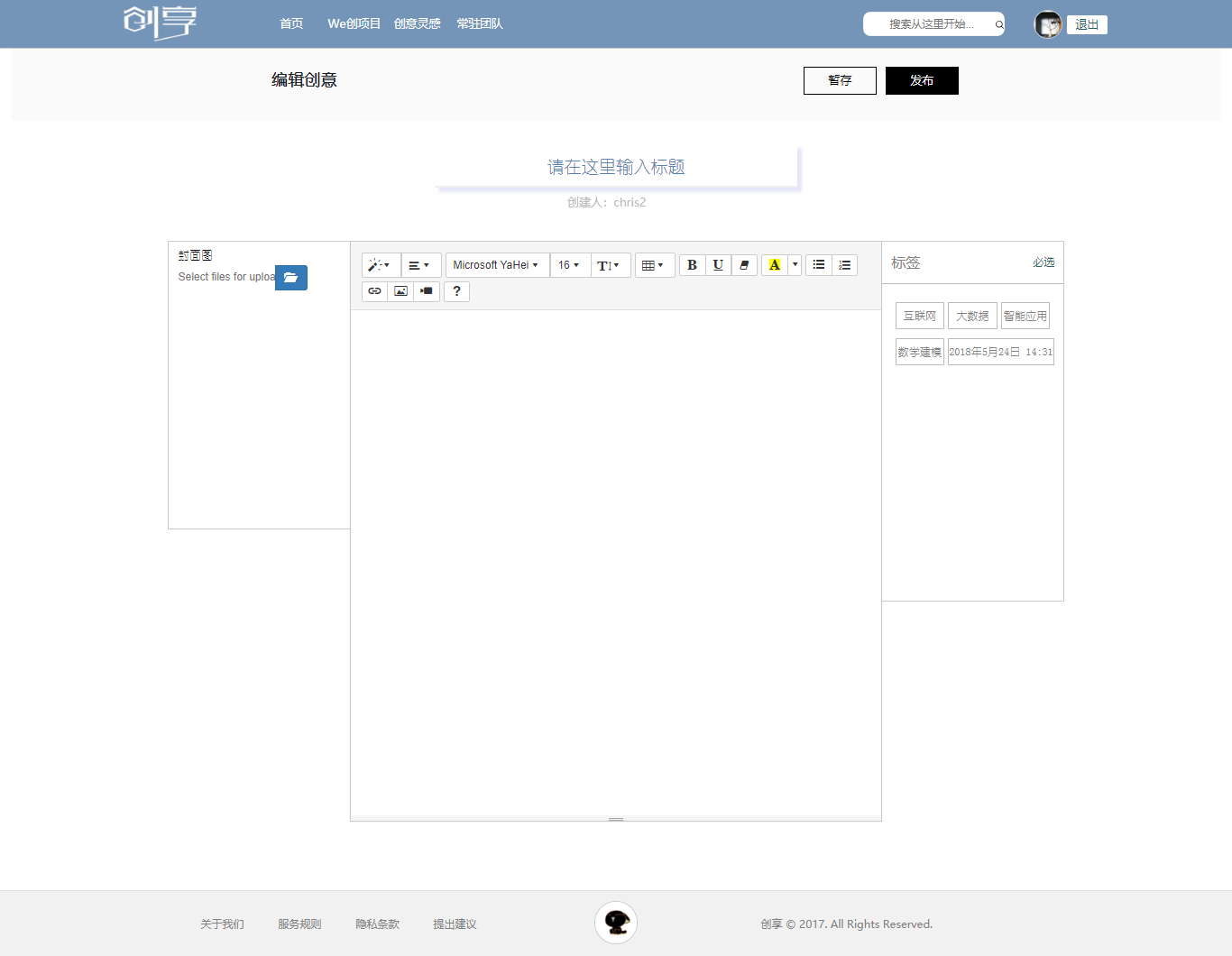
如果需要更新项目的状态或者对项目内容进行调整可以点击相应项目的更改按钮，进入项目详情，进行相应的修改，同样点击确认按钮即可实现项目更新保存操作。

### 5.7.3 创意管理模块

用户可以在个人中心的创意灵感模块进行创意灵感的管理，在这里用户可以新增，删除与查看自己的创意。点击创意灵感模块的添加按钮，即可进行创意编辑，在创意编辑页面根据指示输入创意名称，选择封面图与创意标签，输入创意内容点击发布即可完成创意的发布。此时在个人中心便会有对应创建的创意。

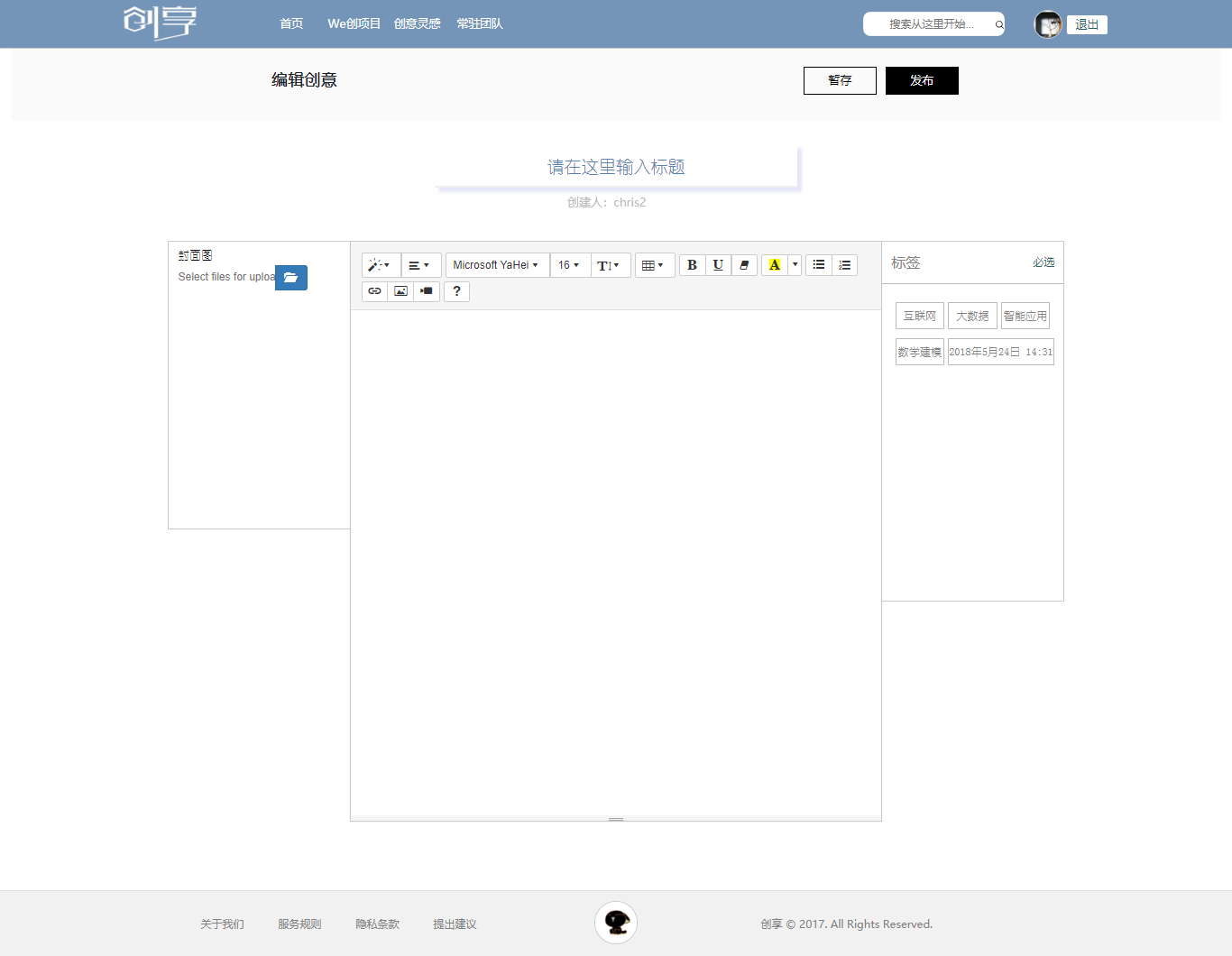
选择不同的创意，点击查看详情，即可查看对应创意详情。如果对创意不满意或者如果其他原因需要删除的话，可以点击删除按钮进行创意的删除。







添加创意详情



### 5.7.4用户消息模块



个人中心消息模块用于用户处理招募申请与查看招募申请情况的地方，在这里查看详情可以查看到自己提出的申请是否被同意，或者处理其他人发出的参与申请，在消息模块如果有人给你的项目点赞或者添加关注，在这里还对有相应的消息提示。

### 5.7.5用户关注模块

用户在浏览项目或者创意的时候，如果发现好的创意或者项目，可以进行关注操作，此时关注的创意或者项目会进入到用户的个人中心的即时关注中，在这里用户可以查看自己关注的项目或者创意，如果需要删除关注的项目或者创意，可以点击对应项目或者创意的垃圾桶标志进行删除。



第6章 结论与展望

本论文介绍的是基于web平台的大学生项目招募平台，该平台的实现基于Python语言以及相应成熟的Django框架，MySQL数据库技术，MVC框架等多种技术，在进行系统学习后将这些技术与知识融会贯通应用在项目开发中。该论文主要分析研究了以下的内容：

1.分析了当今社会对大学生的需求以及当学生的自身需求，国家对创新人才的需求，研究该平台研发的意义与慕白从根本保证该项目的实用性与可执行型

2 在认真研究以及进行技术对比 实现方式比较后，确定研发所需的技术，对确定会涉及到的关键技术进行了深入的学习与研究，团队最终决定使用Python+Djnago+MVC的开发方案是正确的，这种开发架构安全可靠稳定成熟，能够保证系统上线后正常稳定的运行，创造真正的价值

3 根据大学生这类的特殊群体上这一特点上，以及借鉴了一些项目招募网站的设计以及自己的设计使用流程，我们对该系统进行了详细的分析与阐述，对该系统的功能都有了深入的剖析与了解，这为我们以后的数据库设计以及系统设计都奠定了良好的基础

4 在确定了系统需求以及整体业务逻辑之后，开始进行数据库层面的设计与实现，针对不同的业务逻辑抽象出不同的数据库对象，确定各个数据库实体之间的关系，列出数据字典，通过powerdesigner更好的更直观的看出各个数据实体之间的关系，确定了表与表，数据字段与数据字段的联系。

5通过Axure进行项目原型的设计与搭建，实现一个原始的但是项目功能齐全的Demo,通过对该Demo的设计与反复论证保证业务流程的顺畅，并对应数据字典，进行相应的查漏补缺，校验了数据字典。

6 对完成的模块进行模块测试，模块测试通过后进行相应的联调测试，有时增加了新的feture还会进行相应的回归测试，保证原有功能可用，确保系统的适用性、安全性与稳定性及数据完整性

在完成该项目的过程中，我学到了很多在大学课本上学习不到的知识于经验，自己的眼界也得到了很大的拓展，通过设计参与开发该项目，我深刻的了解了Python这门语言的艺术以及Django框架构建项目的快速与简洁，也在前期的项目需求调研的阶段学到了很多的调研方法，自己感觉成长了很多。

该系统能够满足学生在课余时间利用自己的知识技能参与项目开发，以及满足了一些学生创业创心项目的人才需求的问题，解决了创新道路上的最后一个问题，为大众创新，万众创业提供了良好的人才储备平台，也让学生提钱加入到项目开发中，提高了眼界与实际操作能力，补充了大学实际操作的短板

web项目访问简单，只需要用户拥有电脑或者只能手机就能访问该服务，平台可持续发展能力强，未来可以根据实际需求进行一些功能的添加，而且web服务只要更新web服务即可，不用更新客户端的代码，这给该平台带来了很大的优势，平台之后可以考虑在向社会引资方面进行相应的调整，现在平台实习的功能还比较片面，不能将大学生创业所需要的资源全部囊括起来，还有很大的局限性，之后应该考虑将该平台打造成一个大学生创业资源的闭环平台，在这里大学生创新创业就能完全实现，在这里大学生可以找到他们需要的人才资源，项目资源，设备资源，资金资源，还可以在该平台实现项目的推广与转卖，在实现了以上诉求后，可以再在区域内推广，因为该平台可复制性很强，几乎没有推广费用，所需的就是提高产品知名度，让更多的大学生能够参与进来，提升产品的用户量，进而省内推广，然后在推广到全国的所有高校中去，  这样就能真正服务大学生，服务国家的创新型国家的建设，为社会发展做出更大的贡献

# 参考文献

[1] 虫师.Web接口开发与自动化测试:基于Python语言[M].北京:电子工业出版社,2017.

[2]萨默维尔.软件工程 [M].北京：机械工业出版社, 2011

[3]巴里. 深入浅出Python[M].北京：中国电力出版社, 2012.

[4]高云. jQuery技术内幕：深入解析jQuery架构设计与实现原理[M].北京：机械工业出版社, 2015.

[5]Lawson B, Lawson B. Introducing HTML5[M]. Berkeley：New Riders, 2010.

[6]陶国荣.jQuery权威指南[M].北京：机械工业出版社.2008.

[7]弗雷恩，王永强.响应式Web设计：HTML5和CSS3实战[M].北京：人民邮电出版.2013

[8]Stephen Radford.Learning Web Development with Bootstrap and AngularJS[M].Packt Publishing.2015.

[9]John Adams.Learning Kendo UI Web Development[M].Pack Publishing.2013.

[10]萨默费尔德，爱飞翔.Python编程实战:运用设计模式、并发和程序库创建高质量程序[M].北京：机械工业出版社.2014

[11]Holovaty A, Kaplan-Moss J. The Definitive Guide to Django[M]. Berkeley：Apress, 2009.

# 谢 辞

本文的研究工作是在我的导师马月坤教授的悉心指导和严格要求下完成的。马月坤老师在学习方法、工作方法和研究思路等方面给予了许多有益的启迪；同时，他对我的研究工作提出了宝贵的建议和意见，使我在研究工作中不断取得新的进展。马月坤老师深厚的专业知识、严谨的治学精神和求实创新的工作作风深深的影响着我。在此，谨向马月坤老师致以我最崇高的敬意和真挚的感谢！

感谢我的家人和朋友对我生活上的关心，学习和工作的支持，这些使得我能够安心的完成我的研究工作。

最后，对在我的学习和成长道路上给予帮助的所有老师和朋友们表示深深地感谢，对评阅该论文的所有专家表示最崇高的敬意和真挚的感谢！

# 注 释

# 附 录