



山东大学
SHANDONG UNIVERSITY

课 程 设 计

社会实践活动管理系统

学 院：计算机科学与技术学院
专 业：人工智能
小组成员：李 阳 201900170249
 倪诗宇 201900180065
 齐育斌 201900130115
指导教师：甘 甜

2020-2021 学年
第一学期

目录

1	系统开发平台	1
1.1	开发平台简介	1
1.2	开发语言	1
1.3	开发工具	1
1.4	数据库	1
1.5	操作系统	1
2	数据库规划	2
2.1	任务简述	2
2.2	任务目标	2
2.2.1	用户注册登录	2
2.2.2	添加社会实践活动	2
2.2.3	成员加入	2
2.2.4	添加材料	2
2.2.5	管理员操作	3
3	需求分析	4
3.1	数据需求	4
3.2	事务需求	4
3.2.1	数据录入	4
3.2.2	数据更新	5
3.2.3	数据查看	5
3.3	系统需求说明	5
3.3.1	初始数据库大小	5
3.3.2	网络与共享要求	5
3.3.3	备份与恢复	5
3.3.4	用户界面	5
3.3.5	法律问题	5
4	数据库逻辑设计	6
4.1	ER 图	6
4.2	数据字典	6
4.2.1	从数据字典中抽取出来的系统实体描述	6
4.2.2	从数据字典中抽取出来的联系的描述	8
5	数据库物理设计	9
5.1	索引	9
5.2	视图	9
5.3	安全机制	9
5.3.1	系统安全	9
5.3.2	数据安全	9
5.3.3	其他	9

6	应用程序设计	10
6.1	功能模块	10
6.1.1	用户注册登录	10
6.1.2	项目添加加入	11
6.1.3	材料管理	12
6.1.4	后台管理	13
6.2	事务设计	14
6.2.1	用户相关	14
6.2.2	项目相关	14
6.2.3	材料相关	14
6.2.4	分数相关	15
7	测试与运行	16
7.0.1	认证系统测试	16
7.0.2	项目测试	16
7.0.3	材料测试	17
7.0.4	后台管理测试	17
8	总结	18
8.1	系统优点	18
8.2	系统不足	18
8.3	经验与收获	18

1 系统开发平台

1.1 开发平台简介



本项目采用 B/S 架构，使用 Python 语言作为主要开发语言，数据库选用 MySQL，使用 Django 框架进行开发

1.2 开发语言



Python + Html + CSS + JavaScript

1.3 开发工具

PyCharm 2020.2

1.4 数据库

MySQL - 8.0.22

1.5 操作系统



Windows 10 家庭中文版 x64

2 数据库规划

2.1 任务简述

在社会日益发展的今天，对大学生各方面的素质都有较高的要求，大学生不能再单纯的只从书本上获取知识，更要去社会中通过实践获取知识，运用知识，才能成为更全面的人才，为社会建设贡献出自己的力量。社会实践活动管理系统正是提供了这样一个平台，学生可以在这里申请社会实践活动，运用自己的知识，丰富自己的社会经验，获得更好的发展。

社会实践活动管理系统包括用户注册登录，申请社会实践活动，完成进度管理，项目评分管理等几大模块。社会实践活动管理系统提供两种界面，一是用户界面，用户可以执行登录进入系统，申请新的社会实践活动项目，更新已有社会实践活动项目的个人进度，查看其他个人或者队伍的社会实践活动项目等操作；二是管理员界面，管理员可以修改用户信息，对社会实践活动项目的申请进行通过，对已完成的社会实践活动项目进行评分。

2.2 任务目标

2.2.1 用户注册登录

- 新用户注册

用户输入用户名、两次密码、姓名、邮箱地址进行注册，需要检查是否已存在，检查两次输入密码是否相同且是否足够安全。将用户名、哈希加密后的密码、邮件地址、姓名、专业存入数据库。

- 用户登录

用户输入用户名以及密码进行登录，密码通过相同运算与数据库中的密码进行比对，如果不匹配提醒用户密码错误、重新输入，

- 重置密码

用户两次输入新的密码，检查合法后更新数据库。

2.2.2 添加社会实践活动

用户可在用户个人页面中点击“添加项目”来申请进行一个新的社会实践活动项目，项目申请需要提供项目名、项目描述、参与者的数目、项目开始时间。一个人在同一时间时只能作为一个项目的申请人。默认申请人为队长。

2.2.3 成员加入

项目申请通过后，已注册用户可自行选择要加入的项目，直到到达人数上限。

2.2.4 添加材料

项目的参与者可以在添加材料页面选择文件进行上传，并指定该材料所属的项目。

2.2.5 管理员操作

- **移除成员**

管理员可在后台界面查看某项目的所有成员，点击成员右方的移除即可将该成员从成员列表中移除。

- **添加项目**

管理员可以添加项目，并指定队长。

- **项目评分**

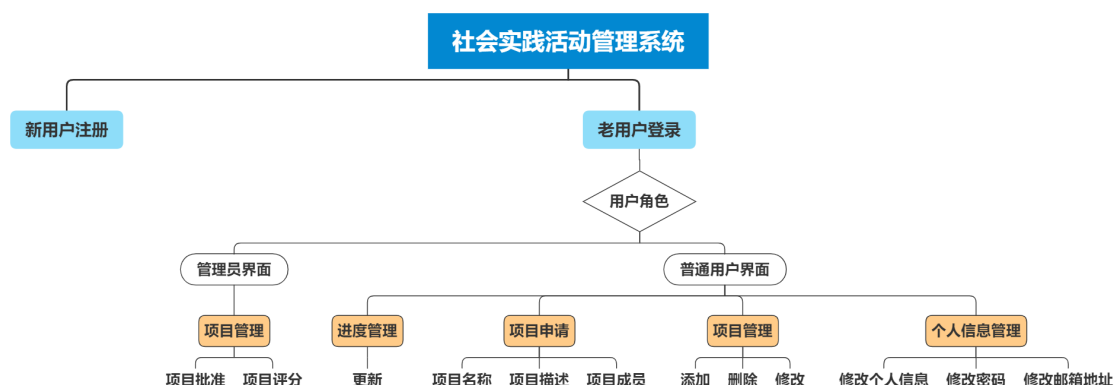
管理员可以为项目进行评分，并在项目信息中展示出来。

- **查看所有用户**

管理员可以在后台查看所有用户的信息。

3 需求分析

本项目的所有功能如下图所示：



3.1 数据需求

1. 用户信息：
用户 ID，密码，邮箱地址，用户组（普通用户、管理员），姓名。
2. 项目信息：
项目 ID，项目名，项目描述，分类，队长，项目开始时间，总进度，总分。
3. 材料信息：
材料 ID，项目 ID，材料名称，材料描述。
4. 评分信息：
评分 ID，项目 ID，用户 ID（限管理员），分数。

3.2 事务需求

3.2.1 数据录入

- 普通用户：
 1. 用户注册
用户 ID，密码，邮箱地址，用户组（普通用户、管理员）。
 2. 用户填写个人信息
用户 ID，姓名，专业。
 3. 用户添加项目
项目 ID，项目名，项目描述，分类，项目开始时间。
 4. 用户提交材料
材料 ID，项目 ID，材料名称，材料描述。
- 管理员：添加项目分数
评分 ID，项目 ID，分数。

3.2.2 数据更新

- 普通用户：

1. 用户个人信息的更新

用户 ID，姓名。

2. 用户邮箱更新

用户 ID，邮箱地址。

3. 用户密码更新

用户 ID，密码。

4. 项目更新

项目 ID，项目名，项目描述，分类，项目开始时间。

- 管理员：项目添加、成员删除。

3.2.3 数据查看

- 普通用户：

1. 查看个人的用户信息，材料信息，评分信息。

2. 查看个人和他人的项目信息。

- 管理员：

1. 查看所有项目信息。

2. 查看所有用户。

3.3 系统需求说明

3.3.1 初始数据库大小

三个初始用户，角色分别为两个普通用户、一个管理员。

3.3.2 网络与共享要求

支持用户同时在线，并能够承受用户频繁的更新完成状态。

3.3.3 备份与恢复

对用户数据进行备份，保证用户数据不丢失。

3.3.4 用户界面

使用 Web 端展示用户界面，并考虑以后将其扩展到移动端上。

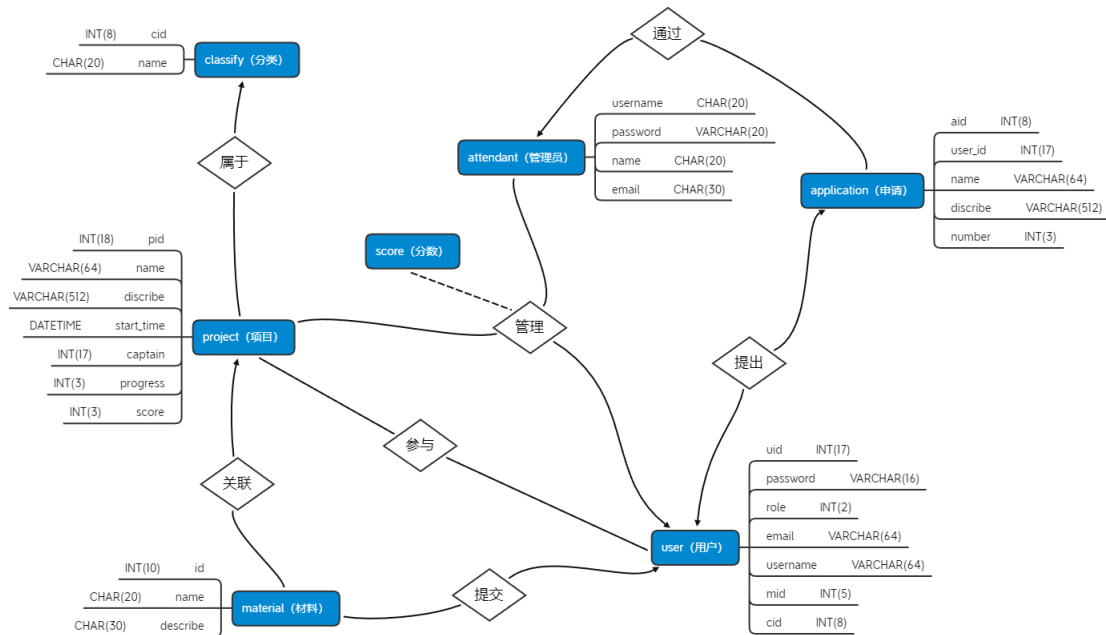
3.3.5 法律问题

遵守相应的法律法规，尊重用户隐私，采用必要的安全措施保证用户的隐私数据不外泄，管理员无权接触到用户的明文密码。

4 数据库逻辑设计

4.1 ER 图

本系统的 ER 图如下：



该 ER 图包含用户、项目、申请、材料、管理员、分类、专业七个实体和若干联系。

4.2 数据字典

4.2.1 从数据字典中抽取出来的系统实体描述

用户实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
uid	INT(17)	YES	YES	YES	YES			0
password	VARCHAR(16)		YES					
role	INT(2)		YES	YES				
email	VARCHAR(64)		YES					
username	VARCHAR(64)		YES					
class	INT(4)				YES			NULL
mid	INT(8)		YES					0

专业实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
mid	INT(5)	YES	YES	YES				
name	VARCHAR(64)		YES					

申请实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
aid	INT(8)	YES	YES	YES	YES		YES	
uid	INT(17)		YES		YES			
name	VARCHAR(64)		YES					
discribe	VARCHAR(512)							NULL
number	INT(3)		YES					1

项目实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
pid	INT(18)	YES	YES	YES	YES		YES	
name	VARCHAR(64)		YES					
discribe	VARCHAR(512)							NULL
captain	INT(17)	YES	YES	YES	YES			0
start_time	DATETIME		YES					
progress	INT(3)		YES	YES	YES	YES		0
score	INT(3)			YES	YES			NULL
cid	INT(8)		YES					0

材料实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
id	INT(10)	YES	YES	YES	YES		YES	
name	CHAR(20)		YES		YES			0
describe	CHAR(30)							

管理员实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
uid	INT(17)	YES	YES	YES	YES			
name	VARCHAR(64)		YES					

分类实体表

Column Name	Datatype	PK	NN	UQ	UN	ZF	AI	Default
cid	INT(8)	YES	YES	YES	YES			0
name	VARCHAR(20)		YES					

4.2.2 从数据字典中抽取出来的联系的描述

Entity	Multiplicity	Relation	Multiplicity	Entity
user	1	提出	1	apply
user	n	属于	1	major
user	n	参与	n	project
attendant	n	管理	n	project
attendant	n	管理	n	user
user	1	提交	n	material
project	1	关联	n	material

P.S. 这里只有多对多的关系需要额外建一张表，其余的皆可以使用外键表示。

5 数据库物理设计

5.1 索引

数据库中的索引：

表 5.1-1 数据库索引表

Table	Primary Key	Foreign Key
user	uid	mid
major	cid	
apply	aid	uid
project	pid	cid
material	id	
classify	cid	
attendant		

5.2 视图

该系统在程序逻辑中根据用户的角色判断其是否能进行某种操作，仅对特定权限的用户开放对应的功能，故未在数据库层面使用视图加以控制。

5.3 安全机制

5.3.1 系统安全

1. 本系统提供了充足的异常处理机制，能够捕获由各种错误引发的异常（如输入数据类型与库要求不一致、查询过程中出现的错误、操作不规范等）。
2. 在用户注册、登录等提交各种表单时，均对数据的合法性进行了判断，对于其中不合法的数据提醒用户重新输入。

5.3.2 数据安全

1. 只有用户登录才能对自己的信息数据进行修改操作，对自己领导的社会实践进行操作，对自己的进度进行更改。管理员也需要登录，才能进行查看、通过和拒绝申请的操作，进行评分操作。
2. 申请实体集元组的删除、项目实体集的总进度和总分等都是通过事务实现的，很好地保护了关系型数据库的强一致性，体现了数据安全。

5.3.3 其他

我们尽力打造了一个结构合理，冗余较少的关系模式，但是仍可能存在一些影响查询速度的设计以及一些冗余，导致性能未能达到最好。

6 应用程序设计

6.1 功能模块

该系统有用户注册登录，个人信息管理，项目管理、完成进度管理等几大功能模块，功能模块具体实现如下：

6.1.1 用户注册登录

1. 用户进入注册界面，输入用户名、邮箱、密码以及密码的确认，系统检查数据格式是否正确，两次密码是否匹配，用户名是否存在，无误后将用户 id 自增 1，用户名，邮箱，默认普通用户角色，加密后的密码，邮箱，昵称、当前注册时间存入数据库，提醒用户注册成功。

社会实践活动管理系统
Welcome to Social Practice Activity Management System !

请注册

用户名: student
用于登陆, 可以包括大小写字母和下划线。

昵称: student_1

密码:

重复密码:

电子邮件: 123@mail.com

注册

2. 用户进入登录界面，输入用户名和密码，系统检查数据格式是否正确，根据用户名查询用户，校验加密后的密码是否匹配，匹配则用户登录成功，成功信息加入 session 中。

社会实践活动管理系统
Welcome to Social Practice Activity Management System !

请登陆

用户名: student

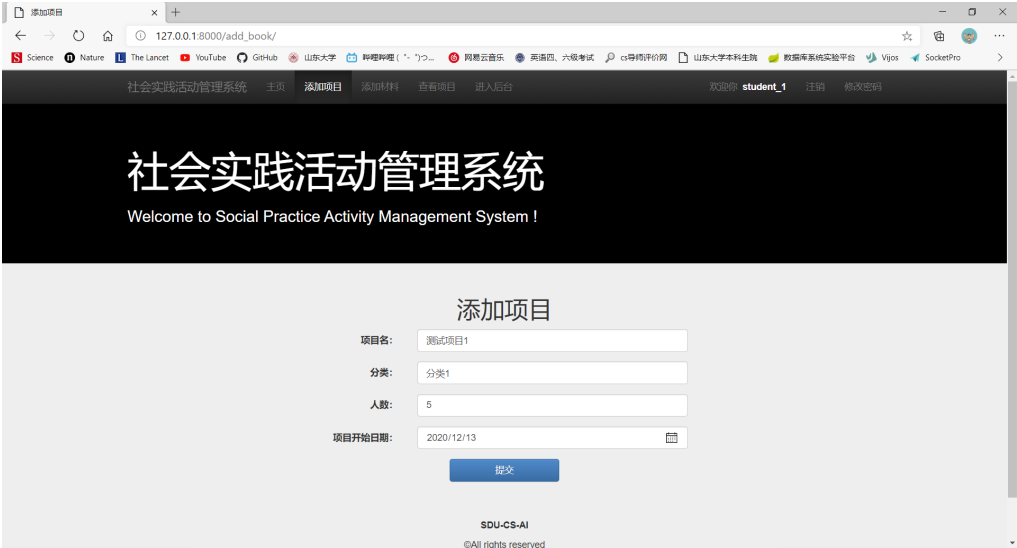
密码:

登陆

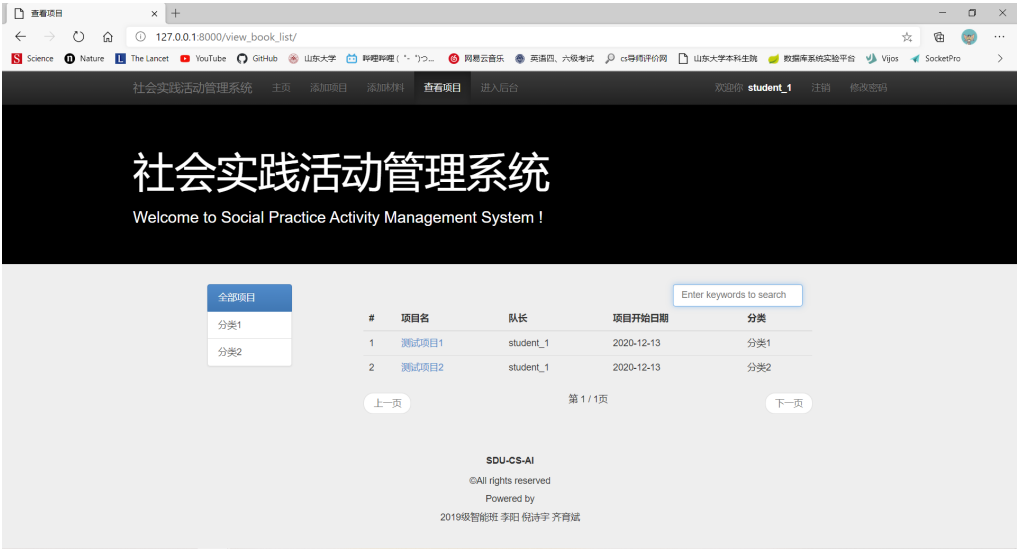
SDU-CS-AI
©All rights reserved
Powered by
2019级智能班 李阳 倪诗宇 齐育斌

6.1.2 项目添加加入

1. 用户进入添加项目界面，输入项目名、分类、人数和项目开始日期，系统检查数据格式是否正确，项目是否已存在，无误后将项目 id 自增 1，项目名称、分类、人数、项目开始日期存入数据库，提醒用户添加项目成功。



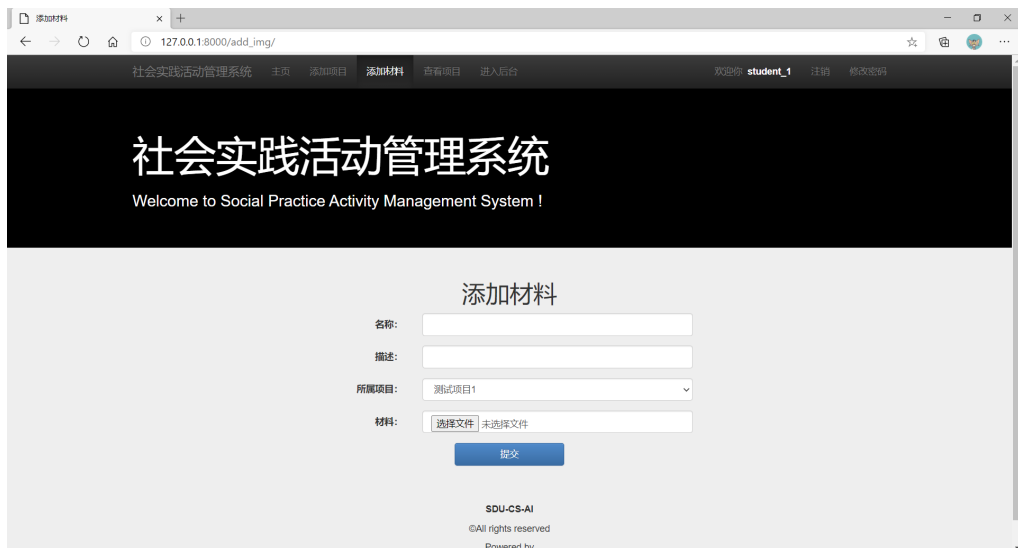
2. 用户进入查看项目界面，在左侧导航栏选择相关分类，找到需要加入的项目名称，核对队长信息，无误后即可单击项目名称进入项目信息界面，再次确实项目无误后，点击下方“点击此处加入项目”即可加入，完成后提醒用户加入成功。





6.1.3 材料管理

- 用户进入添加项目页面，输入名称、描述，并在下拉菜单中选择材料所属的项目，选择文件后即可提交，系统检查输入数据格式是否正确，无误后将材料 id 自增 1，材料名称、描述文件存入数据库，提醒用户添加材料成功。



- 用户每增加一项材料，即可更新相应的 25% 进度，直到进度达到 100% 后不再增加。

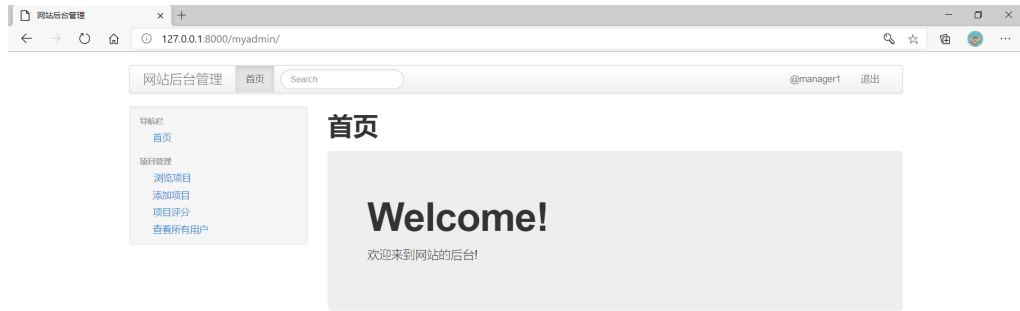


6.1.4 后台管理

1. 管理员进入登录界面，输入用户名和密码，系统检查数据格式是否正确，根据用户名查询用户，校验加密后的密码是否匹配，校验用户身份是否正确，检验成功则管理员登录成功，成功信息加入 session 中。



2. 管理员进入后台管理界面，可在后台界面查看某项目的所有成员，点击成员右方的移除即可将该成员从成员列表中移除。可以添加项目，并指定队长。可以为项目进行评分，并在项目信息中展示出来。可以在后台查看所有用户的信息。



6.2 事务设计

6.2.1 用户相关

- 用户注册

从程序逻辑中获得用户名、邮箱、密码、当前时间等信息，在数据库中搜索是否存在相同用户名、密码，不存在即把这条记录存入数据库，返回成功结果，否则返回失败结果，并通知用户是因何引起。

- 用户登录

从程序逻辑中获得用户名、密码等信息，在数据库中搜索是否有该用户名，输入密码是否与该用户名对应的用户的密码匹配，若匹配则返回成功结果，否则返回失败结果。成功后更新该用户的最近登录时间。

- 密码修改

从程序逻辑中获得用户旧密码与新密码，在数据库中查询该密码是否与该用户的密码匹配，若匹配则更新该用户的密码。

6.2.2 项目相关

- 创建项目

从程序逻辑中获得项目名、项目描述等信息，在数据库中插入一条新的项目记录，其中项目的 ID 是自增 1 的，创建者即为当前用户。

- 加入项目

从程序逻辑中获得用户 ID，项目名等信息，在数据库中更新对应的记录。

6.2.3 材料相关

- 上传材料

从程序逻辑中获得项目 ID，在数据库中插入一条新的材料信息，其中材料的 ID 是自增 1 的，创建者是当前用户，同样关联着之前的项目 ID。对应的项目进度会根据材料提交的个数增加相应的数值，如果达到 100% 则项目进度不会再增加。

6.2.4 分数相关

- 分数更新

插入一条记录，包括当前的用户名，变化的分数，时间戳，在更新分数表时务必同时更新项目的分数属性，若在插入分数数据出现问题时应进行回滚，用户的分数属性复原，用户的分数永远等于分数表中该用户的所有分数增量的平均值，若不相等则数据库发生不相容问题。

7 测试与运行

7.0.1 认证系统测试

- 注册测试

输入邮箱地址、用户名、密码以及密码确认，通过测试可以确定：

1. 用户名包含特殊字符无法通过。
2. 用户名重复无法通过。
3. 用户名长度过长无法通过。
4. 密码长度过长无法通过。
5. 两次密码输入不相同无法通过。
6. 任意一项未填无法通过。
7. 数据库内容与输入相符。

- 登录测试

输入用户名、密码，通过测试可以确定：

1. 用户名包含特殊字符无法通过。
2. 用户名长度过长无法通过。
3. 密码长度过长无法通过。
4. 用户名不存在无法通过。
5. 密码不匹配无法通过。
6. 任意一项未填无法通过。

- 密码重置

可以点击重置密码链接，输入旧密码来重置密码，通过测试可以确定，在旧密码不正确时无法通过，旧密码输入正确后可重新设置密码，同样输入两次，如果不匹配则无法通过，可以发现数据库中对应密码发生变更。

7.0.2 项目测试

- 新建项目

可以输入项目名、项目描述、项目人数、项目开始日期，通过测试可以确定：

1. 除项目描述外，其他任意一项未填均无法通过。
2. 项目人数必须是正整数，否则无法通过。
3. 项目名过长无法通过。
4. 项目描述 Markdown 与 HTML 一致。
5. 可发现数据库内容新增一条记录，内容与输入相符。

- **加入项目**

可以在查看项目页面点击对应的项目名称，点击“点击此处加入项目”连接，通过测试可以发现对应数据库中新增一条记录，且对应成员 id 正确。

7.0.3 材料测试

- **添加材料**

可以输入材料名、材料描述，选择对应项目、文件，通过测试可以发现

1. 除材料描述外，其他任意一项未填均无法通过。
2. 材料名过长无法通过。
3. 材料描述 Markdown 与 HTML 一致。
4. 文件过大无法通过。
5. 可发现数据库中新增一条记录，内容与输入相符。

7.0.4 后台管理测试

- **管理员登录**

输入用户名、密码，通过测试可以确定：

1. 用户名包含特殊字符无法通过。
2. 用户名长度过长无法通过。
3. 密码长度过长无法通过。
4. 用户名不存在无法通过。
5. 密码不匹配无法通过。
6. 任意一项未填无法通过。
7. 用户身份不正确无法通过。

- **移除成员**

可以在后台界面点击浏览项目，在对应项目下查看该项目的所有成员，点击移除。通过测试可以发现，数据库中对对应记录删除，再次查看项目可以发现成员已不在列表中。

- **项目评分**

输入项目名称、分数，通过测试可以发现：

1. 项目名不存在无法通过。
2. 评分不是 0-100 的正数无法通过。
3. 输入的分数精度超过 2 位无法通过。
4. 可发现数据库中新增一条记录，内容与输入相符。
5. 再次查看项目，可发现对应分数信息已经更新。

8 总结

8.1 系统优点

1. 本社会实践活动管理系统想法新颖独特，抓住了日常校园生活的短板，有较强的实用性。
2. 能够在有限的时间内将系统的几大模块设计实现完全。
3. 使用 B/S 架构，无需下载安装客户端，跨越所有平台，使用方便。
4. 前端使用 bootstrap 框架，后端使用 Django 框架，均为时下最流行的开发框架，方便进一步开发。
5. 交互界面美观大方，大小屏幕适配较好，操作简单，易用性高。
6. 系统鲁棒性较高，能够对异常进行处理，能够在用户提交操作不规范的时候给出错误信息。
7. 能够采用可视化技术对数据进行直观的展现，便于用户整体把握。
8. 分别开发了用户界面和后台界面，针对不同用户组提供不同的功能，用户体验好。
9. 加入了文件上传等对技术能力要求较高的功能，也更契合该系统设计的初衷。

8.2 系统不足

1. 数据库的设计可能还存在一些问题，需要进一步的优化。
2. 后台管理员界面的设计还不够美观，在某些功能的设置上考虑欠妥。
3. 个别功能的实现有些许臃肿。

8.3 经验与收获

至此。数据库课程设计已经接近尾声，再去回忆这一个半月以来的投入，从最开始的选题，到之后对各种功能和 ER 图的设计，我们这三个人的 team 走了不少的弯路。我也是第一次在一个完全陌生、起点为零的领域作为一个领导者来统筹规划整个课设的细节，虽然最后呈现的成果可能有些许单薄，但是我们每个人都付出了非常大的心力，也尽力满足了我们最初的预期，这种成功的喜悦也是其他人难以体会到的。

回到这个课设本身，在 ER 图的设计上，甘甜老师给了我们很大的帮助。因为我的一时兴起，我们团队选择了 Django 作为后端开发的框架，而 Django 本身适用于大型 web 开发，也就加大了我们的难度，等到我再去接触 php 的时候已经为时已晚，好在最终的结果是美好的。作为一个 leader，我在分工方面也出现了一些小错误，也让团队的一位成员中途从后端转战前端，很庆幸他有非常强的学习能力，让我们的整体的进度不降反增。“纸上得来终觉浅，绝知此事要躬行”，在开始实践之前，我可能没有觉得之前的书本知识有多重要，直到真正开始做的时候才认识到我们设计数据库，是需要理论作为指导的，才不至于盲目的去实现，然后再修改，就像算法一样，需要预先用理论分析算法的复杂度，才不至于实际去实现运行才发现这个方案行不通，这在实践中确实是极为重要的。

通过这一个月多的课程设计，除了在原有 Python 的基础上学会了后端的开发以外，我最大的收获就是信息检索的能力，面对无数的 bug 和一连串的问题，无论是百度还是 csdn 博客还是官方文档，我都花费巨大的精力去检索有用的信息，我也相信这方面能力的提高对我日后做科研工作有相当大的帮助。感谢我的两个队友的巨大付出，我们三人每周都会有两天花费整个下午共同探讨如何解决问题，那种在一片叹息接近放弃之后突然有人灵机一动然后问题解决的快乐，让我至今都难以忘却，现在我们三人也是非常要好的朋友，以后也会有更多合作的机会，期待我们都能有更好的未来。

李阳 写于 2020 年 12 月 16 日

这次数据库的实验中，我担任的工作较为轻松，只在项目开始的阶段参加了关于 E-R 图的设计，但是设计的 E-R 图却被老师公开处刑，说写的不够完善，导致拖累了项目进度。我们在经历了老师的点拨之后，了解到原本的 E-R 图确实会导致一些预期的功能没法完整的实现，并且缺少很多应有的功能，于是我们又重新完善了 E-R 图，比如增加了材料实体集，使得提交材料的功能能够获得很好的实现，又比如将进度实体集去除，使之变为项目实体集的一个属性，避免了冗余，这些修改操作使得预期的功能能够实现，并且能够为用户提供更优质的服务。

在一开始队伍对我的要求是去学习后端，可惜创业未半而中道崩殂，我们队伍中的大佬单凭一己之力就构造出了后端，于是我的任务变成了学习前端，可惜创业未半又中道崩殂，学习到一半，队伍中的大佬又把前端写完了。于是我们的共同任务又变成了将前后端打通和将普通用户界面和管理员界面连接。在连接的过程中，前端和后端都有所学习的我终于感受到了参与到项目中的感觉，学前端的不太理解后端中的一些语言，学后端的不太理解前端中的一些语言，而我可以在团队中大概起到一个翻译官的作用，方便了两位大佬之间的沟通和交流。

在此次实验中，令我印象最深刻的就是评分功能实现的过程。原本的设想是点击评分后，先进入一个提交表单，在表单中输入各种信息后，再进行提交，将信息储存进数据库。但是在写完 url 和函数之后，发现并没有出现我们预想的效果，然后我们一直对表单页面进行 debug，但是无论怎么更改，都无法出现我们想要的效果，而是直接调用提交函数，因为信息缺失而报错。后来经过思考，我考虑是不是在连接的搭建上出了问题，点击评分后并没有进入表单网页，而是直接调用了函数。事实证明，确实如此，因为我们没有注意函数的命名重复问题，所以导致表单的链接和函数的连接的名字相同，而我们又没有写返回表单页面的函数，导致表单页面一直没有被访问。在构造了返回表单页面地址的函数以及对 url 进行重命名之后，问题得到了解决，评分功能得到了实现。

在此次数据库的大实验中，我的贡献可能并不多，大多数时间都是依靠大佬们的带领，但是我在过程中也学到了很多，提升了自己的能力，交出了一份较为满意的作业。

齐育斌 写于 2020 年 12 月 19 日

本次课设我主要负责前端相关内容，系统地学习了 HTML CSS JS 的相关教程，对 HTML, CSS 有了较为深入的了解，CSS 中关于定位，浮动的相关内容理解有待加强（虽然也没怎么用上，因为有库），大体了解了 JS 相关知识，能够编写基础的 JS 语句实现相关功能。

在项目的搭建过程中，对 django 后端框架有了一定了解，理解了 django 后端的工作机制，但是对于其中语句细节的实现仅限于能够看明白（在后端同学的讲解下）。

在前后端衔接的过程中，我看了很多路径分配的教程，能够熟练运用表单提交方式，对于异步编程方式有了基本的了解，但是对于其他前后端交互的方式了解较少（比如 ajax）。

在项目开发过程中，给我印象最深刻的一个道理就是应该边学边做而不是想着一股脑先把知识掌握了，再去做项目。在理解项目表单 action 中的 url 分配时，我清楚这个 url 是一个路由，告诉这个 url 来处理这个表单，但是一直不理解这个 url 到底该填个什么东西。后来搜了很多相关内容，还是没有理解 url 应该写什么。直到搜完的第二天我们小组一起讨论相关功能的后端开发时，我才知道 url 在 django 框架有一个专门的指定方式，编写对表单进行处理的相关功能，然后通过 url 路径分配将 html 页面中的内容与相关的函数进行关联，数据发送给后台通过这个函数处理，然后返回一定的值或者返回一个页面，这个问题一下子就解决了。而且在项目开发过程中，一度不知道怎么将自己从教程中学到的知识运用的开发中。所以说，应该边学边做，边做边学。

由于我们是三人一个小组来完成一个项目，每个人都负责一部分，确实减少了工作量。但是这样也有一个很明显的弊端，负责前端的同学很难理解后端的代码，后端的同学很难理解前端代码，所以沟通起来很费时间，往往是一起分享思路，但是具体解决方案只能由相关的同学去实现。

因为水平有限，可能项目只是基本实现了功能，仍然存在很多不足（我自己就有很多问题），希望在以后的学习或者比赛中能够捡起这一段知识和经历，然后真正掌握一个 web 项目的制作与完善。

倪诗宇 写于 2020 年 12 月 19 日