2020 秋冬
软件需求工程
软件工程管理

G22 系统编码和实现计划

组长：张溢弛 3180103772

组员：张　琦 3180103162

聂俊哲 3180103501

康大凯 3180105501

潘凯航 3180103812

李　楠 3180103845

目 录

系统编码和实现计划

一、项目简介

该项目开发的软件为一个高校教学平台。当今是计算机网络技术全面深入运用的时代，现代信息技术在切实推进教育创新，深化教学改革方面起着非常重要的作用。搭建线上高校教学平台，能够有效推进信息化教学服务，促进教学资源共享，减轻老师的教学压力；同时，也使学生能够根据自己对任务的理解掌握情况，针对薄弱环节更自主、自由的探索交流解决方法。这改变了传统教学模式中教师的作用和师生间的关系，从而根本的改变了教学结构和教育本质。

高校教学平台无论是在线上教学，还是在辅助线下教学方面都发挥着重要的作用。教学网站方便了师生之间的信息传递与资源分享，有效解决了信息化教学的诸多瓶颈。优质的教学资源能够不受时空约束传递给需要的学生，有利于激发学生的学习兴趣，提高课堂教学效率，提升教学效果。教学平台还极大方便了师生沟通交流和教学管理，是现代教学过程中有效的教学辅助工具。

二、设计计划

2.1 模块划分

2.1.1 用户信息管理模块

用户：登录、查看个人信息、修改个人信息、修改密码、找回密码、注销

教师：编辑教师主页

2.1.2 课程信息模块

管理员：添加课程、删除课程、生成教学班、导入课程相关人员信息（助教、学生、教师）

学生：查看课程信息

教师：查看课程信息、修改课程信息、添加课程章节

游客：查看课程信息

助教：查看课程信息

2.1.3 考试及成绩评定模块

管理员：导入考试成绩、修改考试成绩

学生：查看成绩、申请复核成绩

教师：查看全班成绩、编辑总成绩构成方法

助教：查看成绩、修改作业成绩、处理学生对作业/考试成绩的申诉

2.1.4 通知模块

管理员：公布网站更新信息介绍、删除网站更新信息介绍、更新友情链接

学生：查看网站通知、查看课程通知

教师：公布课程通知、删除课程通知、修改课程通知、查看网站通知、查看课程通知

2.1.5 讨论版/答疑区模块

管理员：删除讨论、删除留言、查看举报内容、处理\回复举报内容

助教、教师：发布讨论、编辑讨论、删除留言、评论留言、点赞留言、反对留言、举报留言、举报讨论、查看举报通知、置顶优秀留言

学生：发布讨论、编辑讨论、删除留言、评论留言、点赞留言、反对留言、举报留言、举报讨论、查看举报通知

2.1.6 作业/考试模块

管理员：后台题库的增加/删除/修改/查询、后台题库答案的维护

教师：添加小组作业、添加个人作业、修改作业、删除作业、查看作业、查看学生作业完成情况、导出/下载提交的作业、批改作业、点评作业、处理学生作业成绩申诉、查看学生作业成绩情况

助教：查看作业、查看学生作业完成情况、导出/下载提交的作业、批改作业、点评作业、处理学生作业成绩申诉、查看学生作业成绩情况

学生：完成组队、查看作业、下载作业、提交个人作业、组长提交小组作业、修改已上传的作业、下载已上传的作业、作业成绩申诉

2.2 数据库设计

2.2.1 用户信息表

SQL 数据库设计

```
1. create table User (
2.     user_id varchar(12) not null,
3.     user_name varchar(20) not null,
4.     user_password varchar(32) not null,
5.     email varchar(64) not null,
6.     phone_number varchar(32)    default null,
7.     birth date default null,
8.     gender int(1) not null,
9.     user_kind int(1) not null, --  用户类型
10.      introduction varchar(512) not null,
11.      department varchar(64) default null,
12.      major varchar(64) default null,
13.      primary key(user_id)
14. );
```

Django 数据库设计

```
1. class User(models.Model):
2.     user_id = models.CharField(primary_key=True, max_length=12, verbose_name=u'学工号')
```

```
3.      user_password = models.CharField(max_length=20, verbose_name=u'登录密码',
default='123456')
4.      email = models.CharField(max_length=64, verbose_name=u'用户邮箱', default='无')
5.      introduction = models.CharField(max_length=512, verbose_name=u'个人介绍',
default='无')
6.      is_student = models.BooleanField(verbose_name=u'是否为学生', default=False)
7.      is_teacher = models.BooleanField(verbose_name=u'是否为教师', default=False)
8.      is_AT = models.BooleanField(verbose_name=u'是否为助教', default=False)
9.      department = models.CharField(max_length=64, verbose_name=u'院系', default='无')
10.     major_class = models.CharField(max_length=64, verbose_name=u'专业班级（为学
生时有效）', default='无')
11.     last_login_time = models.DateTimeField(verbose_name=u'最后登录时间')
12.     homework_not_corrected = models.IntegerField(null=True, verbose_name=u'教师/
助教未批改作业数')
13.     create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
14.     modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
15.
16.     class Meta:
17.         managed = False        # false 为 False 的时候，不会对数据库表进行创建、
删除等操作 可以用于现有表、数据库视图等
18.         db_table = 'user_sre'
19.         verbose_name_plural = '用户表'
```

2.2.2 课程信息表

SQL 数据库设计

```
1. create table Course (
2.      course_id varchar(12) not null,
```

```
3.      course_name varchar(64) not null,
4.      course_credit decimal(10,1) not null,
5.      course_description varchar(1024) not null, --  介绍
6.      department varchar(64) not null,
7.      primary key(course_id)
8. );
```

Django 数据库设计

```
1. class Course(models.Model):
2.      course_id = models.CharField(primary_key=True, max_length=12, verbose_name=u'课程编号')
3.      course_name = models.CharField(max_length=64, verbose_name=u'课程名称')
4.      credit = models.FloatField(verbose_name=u'学分')
5.      department = models.CharField(max_length=64, verbose_name=u'开课学院')
6.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  #  第一次保存时记录时间
7.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
8.
9.      class Meta:
10.         managed = False        # false  为 False 的时候，不会对数据库表进行创建、删除等操作  可以用于现有表、数据库视图等
11.         db_table = 'course_sre'
12.         verbose_name_plural = '课程表'
```

2.2.3 教学班级

SQL 数据库设计

```
1. create table Section (
```

```
2.      section_id varchar(16) not null,
3.      course_id varchar(12) not null,
4.      start_time date not null,
5.      end_time date not null,
6.      primary key(section_id),
7.      foreign key(course_id) references course(course_id)
8. );
```

Django 数据库设计

```
1. class Section(models.Model):
2.      section_id = models.CharField(primary_key=True, max_length=16, verbose_name=u'教学班编号')
3.      course_id = models.ForeignKey(Course, on_delete=models.DO_NOTHING, verbose_name=u'课程编号')
4.      startTime = models.DateField(verbose_name=u'开始时间')
5.      endTime = models.DateField(null=True, verbose_name=u'结束时间')
6.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  #  第一次保存时记录时间
7.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
8.
9.      class Meta:
10.             managed = False         # false  为 False 的时候，不会对数据库表进行创建、删除等操作  可以用于现有表、数据库视图等
11.             db_table = 'section_sre'
12.             verbose_name_plural = '教学班表'
```

2.2.4 教学班级的班级-教师映射表

SQL 数据库设计

```
1. create table Section2Teacher (
2.     section_user_id varchar(12) primary key identity(0, 1),
3.     section_id varchar(16) not null,
4.     user_id varchar(12) not null,
5.     primary key(section_user_id),
6.     foreign key(user_id) references User(user_id),
7.     foreign key(section_id) references Section(section_id)
8. );
```

Django 数据库设计

```
1. class Section2Teacher(models.Model):
2.     section_user_id = models.AutoField(primary_key=True, verbose_name=u'课程教师关系编号')
3.     section_id = models.ForeignKey(Section, on_delete=models.DO_NOTHING, verbose_name=u'教学班编号')
4.     user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING, verbose_name=u'用户学工号')
5.     create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
6.     modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
7.
8.     class Meta:
9.         managed = False        # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
10.        db_table = 'section2teacher_sre'
```

```
11.             verbose_name_plural = '教学班教师表'
```

2.2.5 教学班级的班级-助教映射表

SQL 数据库设计

```
1. create table Section2TA (
2.       section_user_id varchar(12) primary key identity(0, 1),
3.       section_id varchar(16) not null,
4.       user_id varchar(12) not null,
5.       primary key(section_user_id),
6.       foreign key(user_id) references User(user_id),
7.       foreign key(section_id) references Section(section_id)
8. );
```

Django 数据库设计

```
1. class Section2AT(models.Model):
2.       section_user_id = models.AutoField(primary_key=True, verbose_name=u'课程助教关系编号')
3.       section_id = models.ForeignKey(Section, on_delete=models.DO_NOTHING, verbose_name=u'教学班编号')
4.       user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING, verbose_name=u'用户学工号')
5.       create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
6.       modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
7.
8.       class Meta:
9.             managed = False        # false 为 False 的时候，不会对数据库表进行创建、
```

删除等操作 可以用于现有表、数据库视图等

```
10.             db_table = 'section2at_sre'
11.             verbose_name_plural = '教学班助教表'
```

2.2.6 教学班级的班级-学生映射表

SQL 数据库设计

```
1. create table Section2Student (
2.     section_user_id varchar(12) primary key identity(0, 1),
3.     section_id varchar(16) not null,
4.     user_id varchar(12) not null,
5.     primary key(section_user_id),
6.     foreign key(user_id) references User(user_id),
7.     foreign key(section_id) references Section(section_id),
8. );
```

Django 数据库设计

```
1. class Section2Student(models.Model):
2.     section_user_id = models.AutoField(primary_key=True, verbose_name=u'课程学生关系编号')
3.     section_id = models.ForeignKey(Section, on_delete=models.DO_NOTHING, verbose_name=u'教学班编号')
4.     user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING, verbose_name=u'用户学工号')
5.     create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
6.     modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
7.
8.     class Meta:
```

```
9.          managed = False        # false  为 False 的时候，不会对数据库表进行创建、删除等操作  可以用于现有表、数据库视图等
10.         db_table = 'section2student_sre'
11.         verbose_name_plural = '教学班学生表'
```

2.2.7 班级小组表

SQL 数据库设计

```
1. create table Group (
2.      id int primary key identity(1, 1),
3.      section_id varchar(16) not null,
4.      group_number int not null,
5.      user_id varchar(12) not null,
6.      primary key(id),
7.      foreign key(section_id) references Section(Secion_id),
8.      foreign key(user_id) references User(user_id)
9. );
```

Django 数据库设计

```
1. class Group(models.Model):
2.      id = models.AutoField(primary_key=True, verbose_name=u'编号')
3.      section_id = models.ForeignKey(Section, on_delete=models.DO_NOTHING, verbose_name=u'教学班编号')
4.      group_number = models.IntegerField(verbose_name=u'小组编号')
5.      user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING, verbose_name=u'用户学工号')
6.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
```

```
7.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
8.
9.      class Meta:
10.         managed = False  # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
11.         db_table = 'group_sre'
12.         verbose_name_plural = '小组表'
```

2.2.8 课程资料表

SQL 数据库设计

```
1. create table SectionMaterial (
2.      section_id varchar(16) not null,
3.      name varchar(128) not null,
4.      upload_time DATE not null,
5.      foreign key(class_id) references Section(section_id)
6. );
```

Django 数据库设计

```
1. class SectionMaterial(models.Model):
2.      material_id = models.AutoField(primary_key=True, verbose_name=u'编号')
3.      section_id = models.ForeignKey(Section, on_delete=models.DO_NOTHING, verbose_name=u'教学班编号')
4.      name = models.CharField(max_length=128, verbose_name=u'资料名')
5.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
6.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
7.
```

```
8.      class Meta:
9.          managed = False  # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
10.         db_table = 'material_sre'
11.         verbose_name_plural = '课程资料表'
```

2.2.9 作业信息表

SQL 数据库设计

```
1. create table Assignment (
2.      assignment_id int primary key identity(1, 1),
3.      section_id varchar(16) not null,
4.      assignment_name varchar(128) not null,
5.      grade_proportion float(1) not null,
6.      start_time date not null,
7.      end_time date not null,
8.      is_show_grade bit default 0,
9.      show_grade_time time,
10.       teacher_grading_percentage float(1) default 100,
11.       peer_grading_percentage float(1) default 0,
12.       introduction varchar(1024) not null,
13.       is_test bit default 0,
14.       time_limit int not null,
15.       submit_limit int default 1,
16.       primary key(assignment_id),
17.       foreign key(section_id) references Section(section_id)
18. );
```

Django 数据库设计

```
1. class Assignment(models.Model):
2.     assignment_id = models.AutoField(primary_key=True, verbose=u'编号')
3.     section_id = models.ForeignKey(Section, on_delete=models.DO_NOTHING,
verbose_name=u'教学班编号')
4.     assignment_name = models.CharField(max_length=128, verbose_name=u'作业名')
5.     grade_proportion = models.FloatField(verbose_name=u'占成绩比例')
6.     start_time = models.DateTimeField(verbose_name=u'开放时间')
7.     end_time = models.DateTimeField(verbose_name=u'结束时间')
8.     is_show_grade = models.BooleanField(verbose_name=u'是否公布成绩')
9.     show_grade_time = models.DateTimeField(null=True, verbose_name=u'公布成绩时
间')
10.     is_group_work = models.BooleanField(verbose_name=u'是否为小组作业')
11.     teacher_grading_percentage = models.FloatField(verbose_name=u'主观题教师评定
占比')
12.     peer_grading_percentage = models.FloatField(verbose_name=u'主观题互评占比')
13.     introduction = models.CharField(max_length=1024, verbose_name=u'作业介绍')
14.     # material_exist = models.BooleanField(verbose_name=u'是否有附件')
15.     is_test = models.BooleanField(verbose_name=u'是否为测试')
16.     time_limit = models.IntegerField(null=True, verbose_name=u'答题时间限制(单位
为 min)')
17.     submit_limit = models.IntegerField(null=True, verbose_name=u'提交次数限制')
18.     create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
19.     modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
20.
21.     class Meta:
22.         managed = False  # false 为 False 的时候，不会对数据库表进行创建、删
除等操作 可以用于现有表、数据库视图等
```

```
23.             db_table = 'assignment_sre'
24.             verbose_name_plural = '课程作业表'
```

2.2.10 作业题信息表

SQL 数据库设计

```
1. -- 作业题信息表
2. create table Question (
3.     question_id int primary key identity(1, 1),
4.     assignment_id int not null,
5.     question_number int not null,
6.     question_type int not null,
7.     question_point int not null,
8.     right_answer varchar(128) not null,
9.     material_exist bit default 0,
10.      primary key(question_id),
11.      foreign key (assignment_id) references Assignment(assignment_id)
12. );
```

Django 数据库设计

```
1. class Question(models.Model):
2.     question_id = models.AutoField(primary_key=True, verbose_name=u'问题编号')
3.     assignment_id = models.ForeignKey(Assignment, on_delete=models.DO_NOTHING,
verbose_name=u'作业编号')
4.     question_number = models.IntegerField(verbose_name=u'题号')
5.     question_type = models.IntegerField(verbose_name=u'题型')
6.     question_point = models.IntegerField(verbose_name=u'题目分')
7.     right_answer = models.CharField(null=True, max_length=128, verbose_name=u'正确
```

答案')

```
material_exist = models.BooleanField(verbose_name=u'是否有附件')
create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')

class Meta:
    managed = False  # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
    db_table = 'question_sre'
    verbose_name_plural = '课程作业表'
```

2.2.11 选项信息表

SQL 数据库设计

```
create table OptionInfo (
    option_id int primary key identity(1, 1),
    question_id int not null,
    option_position int not null,
    option_content varchar(512) not null,
    primary key(option_id),
    foreign key(question_id) references Question(question_id),
);
```

Django 数据库设计

```
class OptionInfo(models.Model):
    option_id = models.AutoField(primary_key=True, verbose_name=u'选项编号')
    question_id = models.ForeignKey(Question, on_delete=models.DO_NOTHING,
```

```
pverbose_name=u'问题编号')
    option_position = models.CharField(verbose_name=u'第几项')
    option_content = models.CharField(max_length=512, verbose_name=u'选项内容')
    create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
    modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')

    class Meta:
        managed = False  # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
        db_table = 'option_sre'
        verbose_name_plural = '题目选项表'
```

2.2.12 问题附件表

SQL 数据库设计

```
create table QuestionMaterial (
    material_id int primary key identity(1, 1),
    question_id int not null,
    name varchar(128) not null,
    primary key(material_id),
    foreign key(question_id) references Question(question_id)
);
```

Django 数据库设计

```
class QuestionMaterial(models.Model):
    material_id = models.AutoField(primary_key=True, verbose_name=u'作业附件编号')
    question_id = models.ForeignKey(Question, on_delete=models.DO_NOTHING,
```

verbose_name=u'题目编号')

```
4.      name = models.CharField(max_length=128, verbose_name=u'资料名')
5.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时间')  # 第一次保存时记录时间
6.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
7.
8.      class Meta:
9.          managed = False  # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
10.             db_table = 'question_material_sre'
11.             verbose_name_plural = '问题附件表'
```

2.2.13 学生作业情况表

SQL 数据库设计

```
1. create table Assignment2Student (
2.      id int primary key identity(1, 1),
3.      assignment_id int not null,
4.      user_id varchar(12) not null,
5.      status int default 0,
6.      submit_number int default 0,
7.      final_score float(1) default 0,
8.      foreign key(assignment_id) references Assignment(assignment_id),
9.      foreign key(user_id) references User(user_id),
10. );
```

Django 数据库设计

```
1. class Assignment2Student(models.Model):
2.      id = models.AutoField(primary_key=True, verbose_name=u'编号')
```

```
3.      assignment_id = models.ForeignKey(Assignment, on_delete=models.DO_NOTHING,
verbose_name=u'作业编号')
4.      user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING,
verbose_name=u'用户学工号')
5.      status = models.IntegerField(default=0, verbose_name=u'当前状态')
6.      submit_number = models.IntegerField(verbose_name=u'已提交次数')
7.      final_score = models.FloatField(null=True, verbose_name=u'最终得分')
8.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
9.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
10.
11.      class Meta:
12.          managed = False  # false 为 False 的时候，不会对数据库表进行创建、删
除等操作 可以用于现有表、数据库视图等
13.          db_table = 'assignment_student_sre'
14.          verbose_name_plural = '学生作业表'
```

2.2.14 学生每道题的情况表

SQL 数据库设计

```
1. create table Question2Student (
2.      id int primary key identity(1, 1),
3.      question_id int not null,
4.      user_id varchar(12),
5.      status int default 0,
6.      answer varchar(128),
7.      score float(1) default 0,
8.      teacher_score float(1) default 0,
9.      peer_score float(1) default 0,
10.      primary key(id),
```

```
11.         foreign key(question_id) references Question(question_id),
12.         foreign key(user_id) references User(user_id)
13. );
```

Django 数据库设计

```
1. class Question2Student(models.Model):
2.         id = models.AutoField(primary_key=True, verbose_name=u'编号')
3.         question_id = models.ForeignKey(Question, on_delete=models.DO_NOTHING,
verbose_name=u'问题编号')
4.         user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING,
verbose_name=u'用户学工号')
5.         status = models.IntegerField(default=0, verbose_name=u'当前状态')
6.         answer = models.CharField(max_length=128, verbose_name=u'答案')
7.         score = models.FloatField(null=True, verbose_name=u'最终得分')
8.         teacher_score = models.FloatField(null=True, verbose_name=u'教师评分')
9.         peer_score = models.FloatField(null=True, verbose_name=u'互评评分')
10.         create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')   #  第一次保存时记录时间
11.         modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
12.
13.         class Meta:
14.             managed = False    # false  为 False 的时候，不会对数据库表进行创建、删
除等操作  可以用于现有表、数据库视图等
15.             db_table = 'question_student_sre'
16.             verbose_name_plural = '学生问题表'
```

2.2.15 申诉表

SQL 数据库设计

```
1. create table Complaint (
2.     complaint_id int primary key identity(1, 1),
3.     assignment_id int not null,
4.     user_id varchar(12),
5.     is_read bit default 0,
6.     foreign key(user_id) references User(user_id),
7.     foreign key(assignment_id) references Assignment(assignment_id),
8. );
```

Django 数据库设计

```
1. class Complaint(models.Model):
2.     complaint_id = models.AutoField(primary_key=True, verbose_name=u'申诉编号')
3.     assignment_id = models.ForeignKey(Assignment, on_delete=models.DO_NOTHING,
verbose_name='作业编号')
4.     user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING,
verbose_name=u'用户学工号')
5.     is_read = models.BooleanField(default=False, verbose_name=u'是否已读')
6.     create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
7.     modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
8.
9.     class Meta:
10.         managed = False  # false 为 False 的时候，不会对数据库表进行创建、删
除等操作 可以用于现有表、数据库视图等
11.         db_table = 'complaint_sre'
12.         verbose_name_plural = '学生成绩申诉表'
```

2.2.16 公告

SQL 数据库设计

```
1. create table Notice (
2.     notice_id int primary key identity(1, 1),
3.     course_id varchar(12) not null,
4.     section_id varchar(12) not null,
5.     title varchar(50) not null,
6.     content varchar(1024) not null,
7.     publish_time date not null,
8.     primary key(notice_id),
9.     foreign key(course_id) references Course(course_id),
10.      foreign key(section_id) references Section(section_id),
11. );
```

Django 数据库设计

```
1. class Notice(models.Model):
2.     notice_id = models.AutoField(primary_key=True, verbose_name=u'通知编号')
3.     course_id = models.ForeignKey(Course, on_delete=models.DO_NOTHING,
verbose_name=u'课程编号')
4.     title = models.CharField(max_length=50, verbose_name=u'通知标题')
5.     content = models.CharField(max_length=512, verbose_name=u'通知内容')
6.     publish_time = models.DateTimeField(verbose_name='发布时间')
7.     create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
8.     modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
9.
```

```
10.         class Meta:
11.             managed = False    # false 为 False 的时候，不会对数据库表进行创建、删除等操作 可以用于现有表、数据库视图等
12.             db_table = 'notice_sre'
13.             verbose_name_plural = '课程通知表'
```

2.2.17 讨论表

SQL 数据库设计

```
1. create table Discussion (
2.         discussion_id int primary key identity(1, 1),
3.         course_id varchar(12) not null,
4.         section_id varchar(12) not null,
5.         user_id varchar(12) not null,
6.         title varchar(50) not null,
7.         content varchar(1024) not null,
8.         publish_time date not null,
9.         last_reply_time time not null,
10.          num_reply int default 0,
11.          primary key(discussion_id),
12.          foreign key(course_id) references Course(course_id),
13.          foreign key(section_id) references Section(section_id),
14.          foreign key(user_id) references User(user_id)
15. );
```

Django 数据库设计

```
1. class Discussion(models.Model):
2.         discussion_id = models.AutoField(primary_key=True, verbose_name=u'讨论编号')
```

```
3.      course_id = models.ForeignKey(Course, on_delete=models.DO_NOTHING,
verbose_name=u'课程编号')
4.      user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING,
verbose_name=u'发帖人学工号')
5.      title = models.CharField(max_length=50, verbose_name=u'讨论标题')
6.      publish_time = models.DateTimeField(verbose_name='发布时间')
7.      last_reply_time = models.DateTimeField(verbose_name='最后回复时间')
8.      num_reply = models.IntegerField(verbose_name='贴子总数')
9.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
10.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
11.
12.      class Meta:
13.          managed = False   # false 为 False 的时候，不会对数据库表进行创建、删
除等操作 可以用于现有表、数据库视图等
14.          db_table = 'discussion_sre'
15.          verbose_name_plural = '讨论表'
```

2.2.18 回复

SQL 数据库设计

```
1. create table Discussion (
2.      discussion_id int primary key identity(1, 1),
3.      course_id varchar(12) not null,
4.      section_id varchar(12) not null,
5.      user_id varchar(12) not null,
6.      title varchar(50) not null,
7.      content varchar(1024) not null,
8.      publish_time date not null,
```

```
9.      last_reply_time time not null,
10.      num_reply int default 0,
11.      primary key(discussion_id),
12.      foreign key(course_id) references Course(course_id),
13.      foreign key(section_id) references Section(section_id),
14.      foreign key(user_id) references User(user_id)
15. );
```

Django 数据库设计

```
1. class Reply(models.Model):
2.      reply_id = models.AutoField(primary_key=True, verbose_name=u'讨论编号')
3.      discussion_id = models.ForeignKey(Discussion, on_delete=models.DO_NOTHING,
verbose_name=u'讨论编号')
4.      user_id = models.ForeignKey(User, on_delete=models.DO_NOTHING,
verbose_name=u'回帖人学工号')
5.      content = models.CharField(max_length=1024, verbose_name=u'回复内容')
6.      reply_time = models.DateTimeField(verbose_name='回复时间')
7.      floor = models.IntegerField(verbose_name='回复楼层')
8.      create_time = models.DateTimeField(auto_now_add=True, verbose_name=u'创建时
间')  # 第一次保存时记录时间
9.      modify_time = models.DateTimeField(auto_now=True, verbose_name=u'修改时间')
10.
11.      class Meta:
12.          managed = False  # false 为 False 的时候，不会对数据库表进行创建、删
除等操作 可以用于现有表、数据库视图等
13.          db_table = 'reply_sre'
14.          verbose_name_plural = '讨论回复表'
```

2.3 数据库 E-R 图

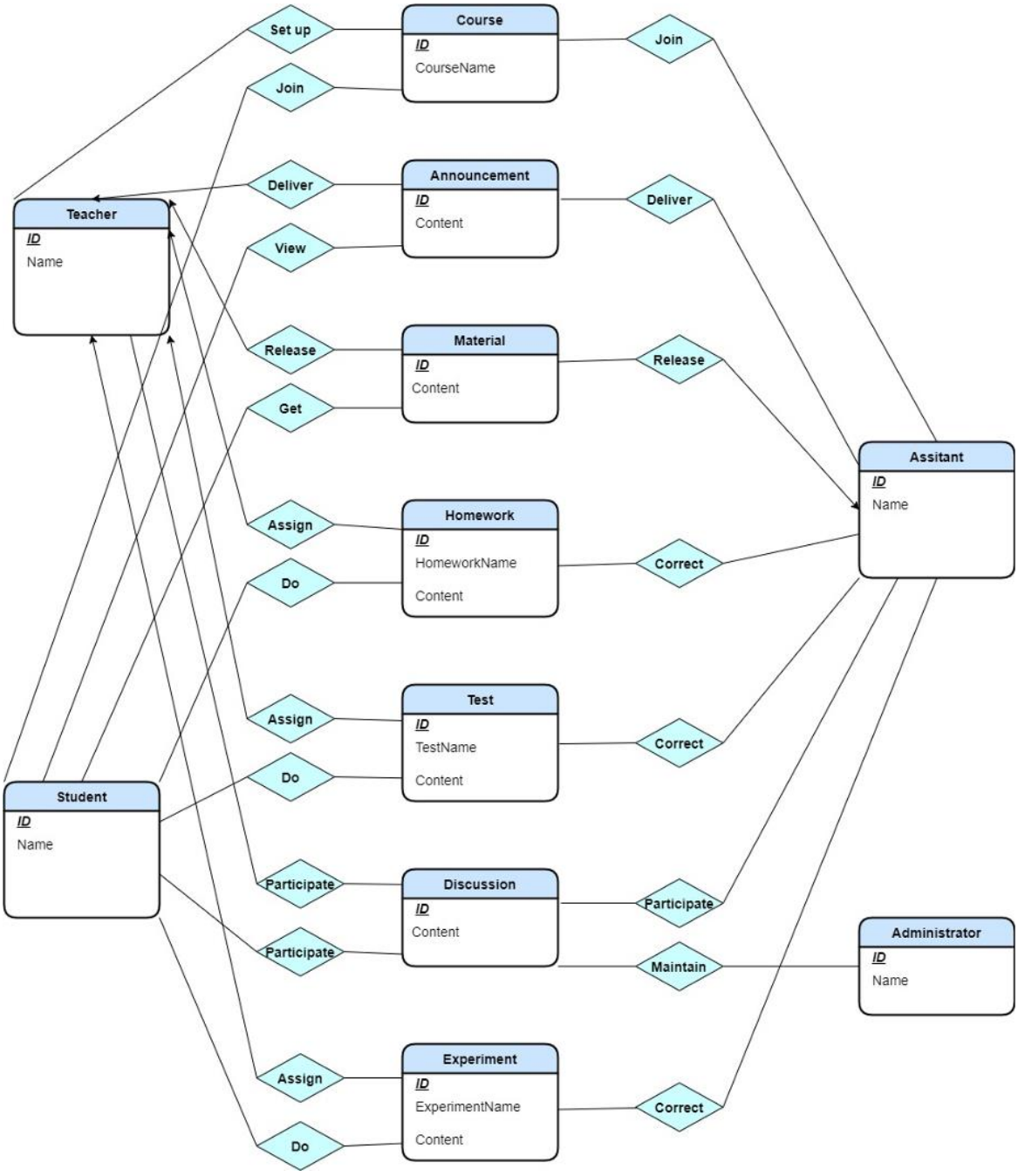

图 1 数据库 ER 图

三、系统编码计划

3.1 团队角色

角色	职责	负责人
项目经理	在约定时间和预算范围内以较高质量领导项目小组全体成员完成全部项目工作任务，并让客户满意。	张溢弛
产品经理	负责市场调查并根据用户需求，选择何种技术、商业模式等，根据产品的生命周期，协调研发、营销、运营等，确定和组织实施相应的产品策略。	张琦
测试经理	建立系统框架；数据库设计；概要设计；参加技术评审。	聂俊哲
设计总监	组织编写测试计划和测试方案；组织系统测试；参加技术评审。	潘凯航
质量经理	带领软件质量监督组成员制定质量保证计划，对监督组反映的质量问题进行汇总与产品经理、项目经理进行交流，当新的问题出现时最终由质量经理决定处理方式。	康大凯
UI 设计	对 Web 应用程序的 UI 进行设计。	李楠
开发人员	负责整个项目的编码工作以及	全体成员

	单元测试。同时进行系统集成，及时解决集成后测试出现的问题。	
测试人员	编写测试方案和测试用例，进行系统测试，并及时向开发人员反馈。	全体成员
软件质量监督人员	实时对项目经理与质量经理提供的项目进度与项目实际开发时的差异提出疑议，指出其中的原因，并分析改进方法。	全体成员

表格 1 团队角色表

3.2 团队分工

	职责	负责人
前端	根据用户需求，按照模块划分，配合后端实现，设计前端页面，设计良好的用户界面交互模式。	全体成员
后端	设计数据库，配合前端搭建，实现后端功能。	全体成员

表格 2 团队分工表

3.3 日程安排

项目阶段	持续时间	主要工作	可预见产出
项目启动	2020.9.15-2020.10.15	进行项目可行性分析，制定项目计划。	完成《项目可行性分析报告》、《项目章程》、《项目计划》。
需求分析	2020.9.30-2020.10.15	确定系统运行的客观环境，确定系统功能及性能，建立系统逻辑模型。	完成《前景与范围》、《质量保证计划》、《需求工程计划》、《软件需求规格说明书》。
系统设计	2020.10.16-2020.10.31	进行系统设计。	完成《系统设计计划》、《系统编码实现计划》、《软件概要设计说明书》、《测试计划》。
第一次需求访谈	2020.10.31	进行需求访谈。	完善需求分析。
系统培训	2020.11.1-2020.11.14	进行系统培训。	全体组员进行项目的预学习。
工程实现	2020.11.2-2020.11.30	进行项目的工程实现。	实现工程并将网站部署到云端，完成《用户手册》、《工程部署计划》、《培训计划》。
第二次需求访谈	2020.11.28	进行第二次需求访谈。	进一步完善需求分析。
需求维护	2020.12.1-2020.12.25	进行第三次需求访谈，根据需求变更进行控制。	完成《需求变更控制会规程》，《需求变更控制文档》，更新《软件需求规格说明书》。
系统迭代开	2020.12.26-2021.1.4	根据变更的需求进行系	更新系统的源代码，完善系

发		统的进一步开发，并完善系统	统功能，更新《测试计划》和《用户手册》
系统测试	2021.1.4-2020.1.7	进行系统测试，项目总结。	完成《测试报告》、《系统维护计划》、《项目总结报告》。
系统发布	2020.1.8	发布完善的系统	完成本学期所有报告和代码，准备最终答辩

表格 3 日程安排表