【XXX 系统】

——XXX 子系统

设计说明书

组长：

组员：

日期：

版本：

目录

1. 文档介绍

1.1. 编写目的

本文档描述软件产品功能设计说明书（SRS）的目的是：

1） 定义软件总体要求，作为用户和软件开发人员之间相互了解的基础；

2） 提供性能要求、初步设计和用户影响的信息，作为软件人员进行软件结构设计和编码的基础；

3） 作为软件总体测试的依据。

1.2. 文档范围

XXX 系统功能设备说明书主要是对该系统中功能及功能操作方式进行基本的描述，XXX，XXX……

1.3. 读者对象

编写详细设计人员及程序开发人员、XXXX、YYYY……

1.4. 术语与缩写解释

（对文档编写中可能遇到的专业词汇或陌生词汇进行说明）【以下斜体红色文字不应该出现在最终的文档里】

缩写、术语及符号	解释
SOA 架构	面向服务的体系结构。
元数据 Metadata	描述数据的内容、质量、状况和其他有关特征的数据。
数据中心 Data Center	以各类数据为核心，依托成熟的存储、数据库、GIS、网络等技术，按照统一标准，建立的具有信息管理、分析、查询、统计及服务的一体化数据管理体系。
数据管理 Data Management	利用数据库、数据仓库、元数据和网络等技术，建立分布式、集中式或集中加分布式数据管理系统，开展数据接收、组织存储、运行维护、更新、共享交换等工作，实现对数据资源的有效组织和应用。

1.5. 参考资料

（包含文档编写过程中参考的材料以及版本）

序号	文档名称	文档编号	版本	发布日期
	……			

2. 项目介绍

2.1. 项目说明

介绍产品的名称、任务提出者、开发者、用户群

项目名称：XXX 系统。

任务提出者：XXX。

开发者：XXX。

用户群：XXX

2.2. 项目背景

XXX

2.3. 需求概述

2.3.1 功能需求
2.3.2 性能需求
2.3.3 安全需求

2.4. 条件与限制

2.3.1 约束条件，主要是服务器运行环境限制，开发过程中代码规范，工具版本等限制

3. 总体设计

3.1. 基本设计概念和流程处理

本子系统是一个横跨前后端的大模块，主要负责教室添加、自动排课、手动排课、课 表查询等子模块。

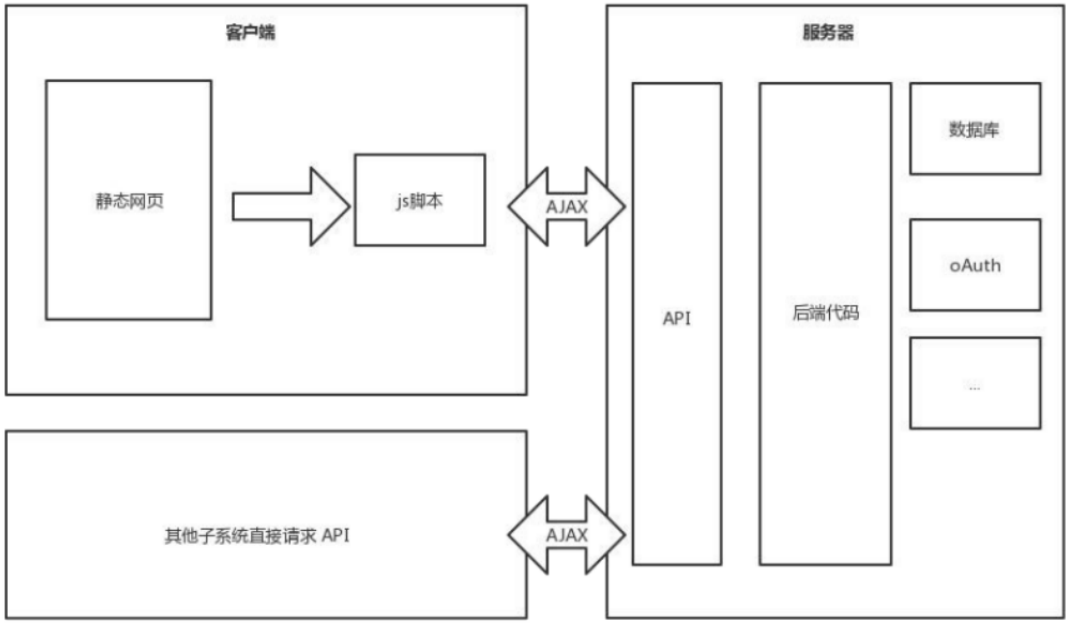
前端：Bootstrap 框架

后端：基于 Python 的 Django 框架

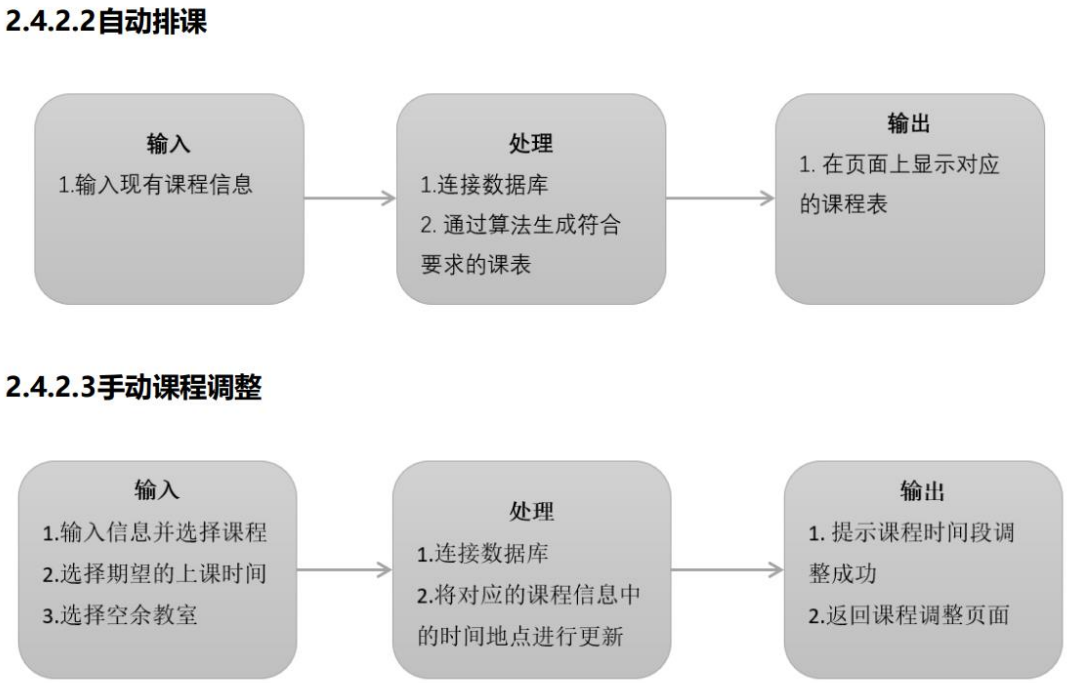
数据库：数据库采用 MySQL，通过 promise 来实现异步查询

客户端：考虑到本系统所使用的框架较新，推荐使用 Chrome、Firefox、Safari 或 Microsoft Edge 浏览器。不推荐使用任何一个版本的 IE 浏览器，如需使用，版本须在 11 及以 上。

处理流程图如下：

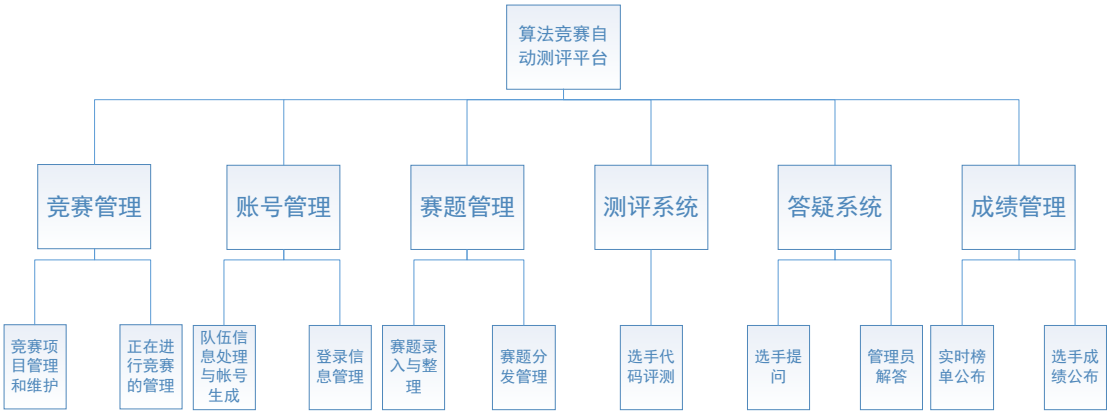


3.2. 功能 IPO 图



3.3. 系统结构

（参考需求分析中的用例分析、数据流图、活动图和状态图，绘制系统结构图。）
注意添加文字说明和绘制过程，

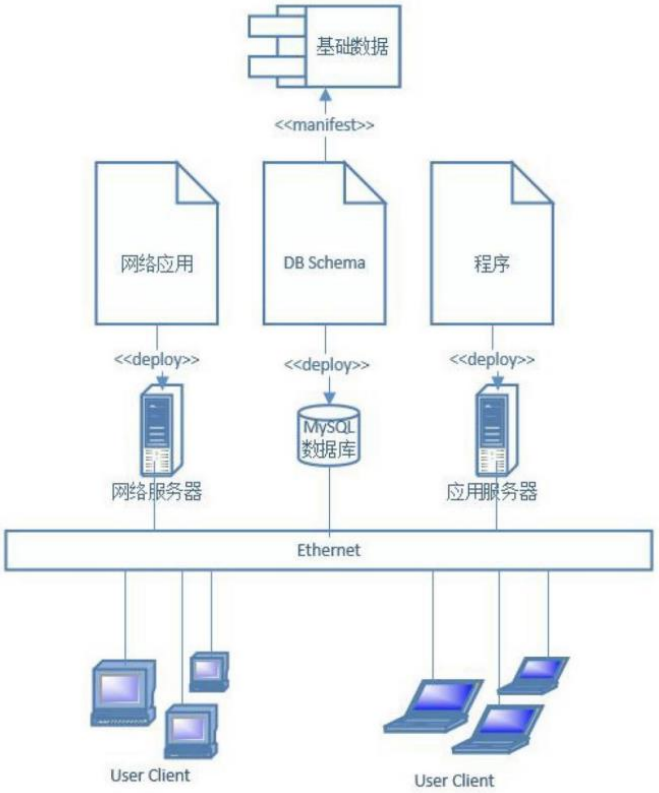


3.4. 技术介绍

（这里对项目采用的技术、语言、框架做系统介绍）

3.5. 部署图

（以服务器、网关、用户为单位描述整个系统的结构和部署过程）



3.6. 类图

（类图是对 CRC 卡的进一步细化）

3.7. 接口设计

3.7.1 内部接口

（描述子模块内部各个功能的关联和提供的操作接口（interface）。推荐以类+方法的形式提供。）

3.7.2 外部接口

（描述子模块间互相调用的接口，推荐以 RPC 协议形式提供，例如 restful。以及系统和外部物理设备交互的接口）

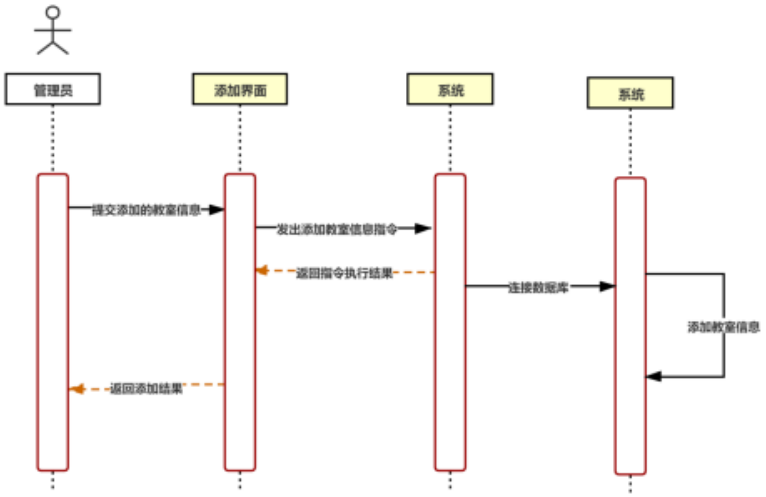
a. 硬件接口

b. 软件接口

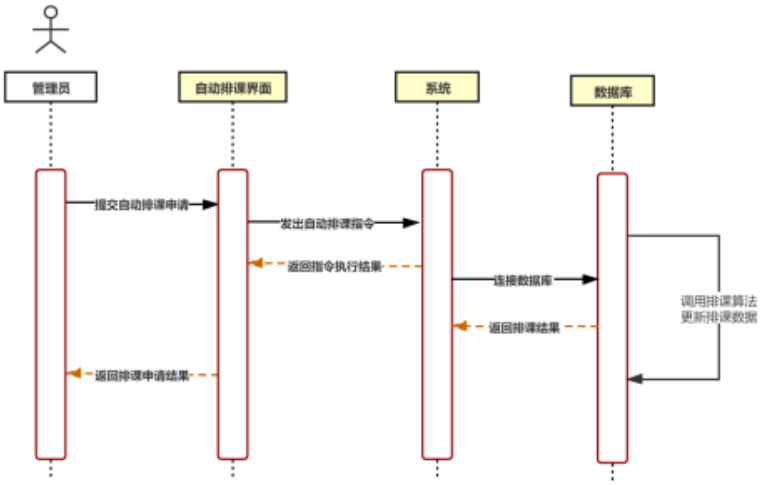
4. 详细设计

4.1. 顺序图

3.6.1 添加教室信息



3.6.2 自动排课



3.6.3 手动课程调整

4.2. 执行概念

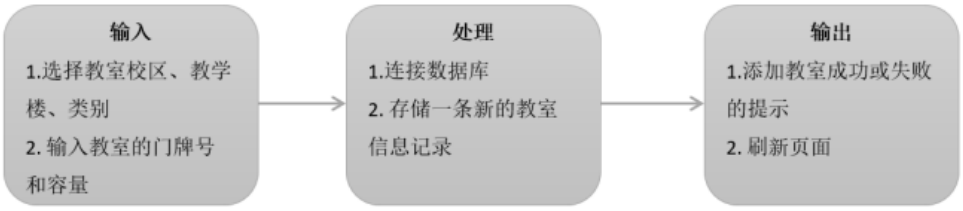
(对每个功能执行概念详细解释，下面仅提供内容例子)

4.1 添加教室信息

4.1.1 模块概述

在这个模块中，管理员可以向系统添加新的教室。添加教室时，管理员应当提供该教室的所在校区、所在教学楼、门牌号、类别、容量等信息，且门牌号必须为与同一教学楼已有教室不重复的值，否则系统将提示管理员重新输入。

4.1.2 IPO图



4.1.3 功能

添加一个新的可用教室到系统中。

4.1.4 输入项

名称	标识	类型和格式	输入方式
校区	new_room_campus_id	int	外部选择->脚本转换
教学楼	new_room_building_id	int	外部选择->脚本转换
门牌号	new_room_number	int	外部输入
容量	new_room_capacity	int	外部输入
类别	new_room_type	char	外部选择->脚本转换

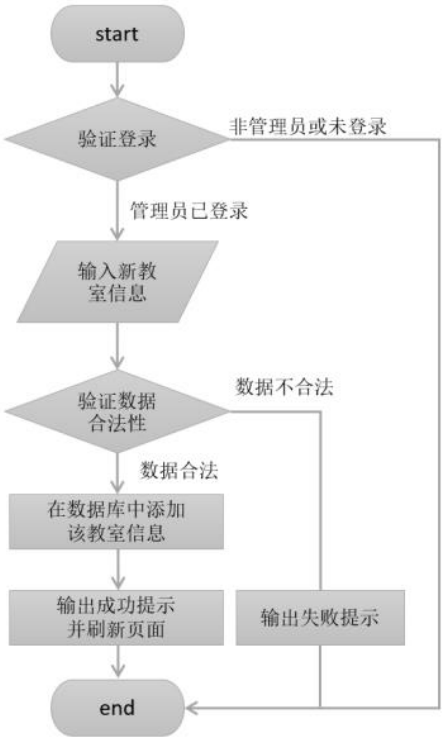
4.1.5 输出项

名称	标识	类型和格式	输出方式
添加结果	room_insert_result	boolean	由脚本输出

4.1.6 设计方法（算法）

```
if 管理员账号已登录
    连接数据库
    管理员选择校区、教学楼、教室类别，输入门牌号和教室容量
    检测输入信息是否合法（如信息是否完整、门牌号是否重复等）
    if 信息验证合法
        添加该教室记录到数据库中
        返回成功信息，并刷新页面
    else
        返回失败信息（信息不合法）
else 返回失败信息（跳转到登录界面）
```

4.1.7 流程图



25

4.1.8 测试计划

输入数据	预期结果
未登录，输入信息	返回错误信息，跳转到登录界面
输入信息不合法	返回错误信息，输入信息不合法
已登录，输入信息合法	返回成功信息，教室添加成功

4.2 自动排课

4.2.1 模块概述

在本模块中，管理员可以通过已经添加的课程自动生成课表。如果已经生成过课表，并且此时有新的课程信息添加，系统提示管理员是否需要重新生成课表。如果无法生成课表，则向管理员发送排课失败提示。

5. 用户界面

5.3 手动课程管理界面

以管理员身份登陆后可用。点击左侧边栏相应选项进入，分为处理排课申请、手动调课申请和申请通道管理三个子模块。在处理排课申请页面，点击列表中任意申请进入审核，选择处理结果并填写处理意见后点击提交确认。

6. 数据库设计

6.1 概念结构设计

（以竞赛管理系统为例，对于系统数据结构进行分析，绘制 ER 图）

6.1.1 选手和队伍信息

队伍为强实体，选手为弱实体，如图 7：

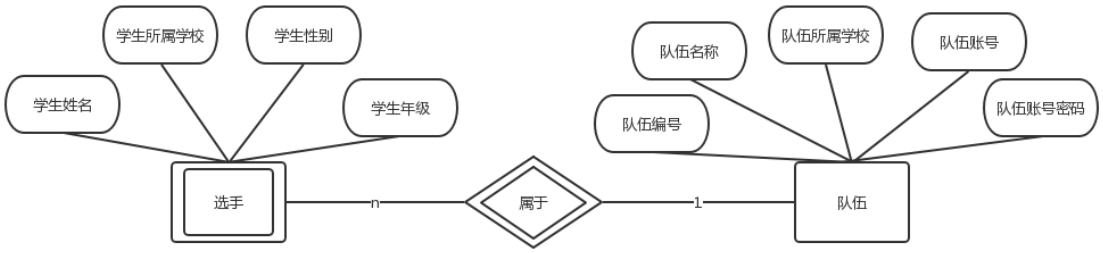


图 7：队伍和选手信息 E-R 图

6.1.2 竞赛信息

竞赛为强实体，如图 8：

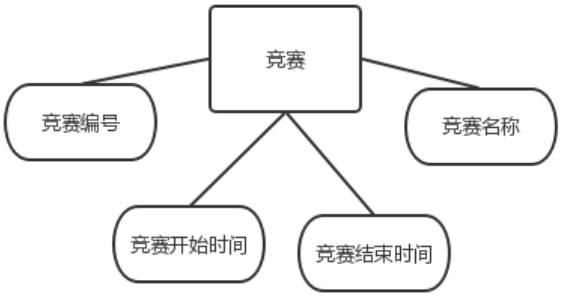


图 8：竞赛信息 E-R 图

6.1.3 题目信息

题目信息如图 9：

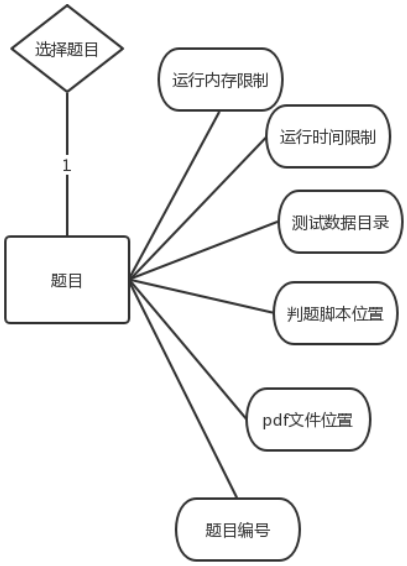


图 9：题目信息 E-R 图

6.1.4 提交记录

题目的提交记录也作为一个实体，如图 10：

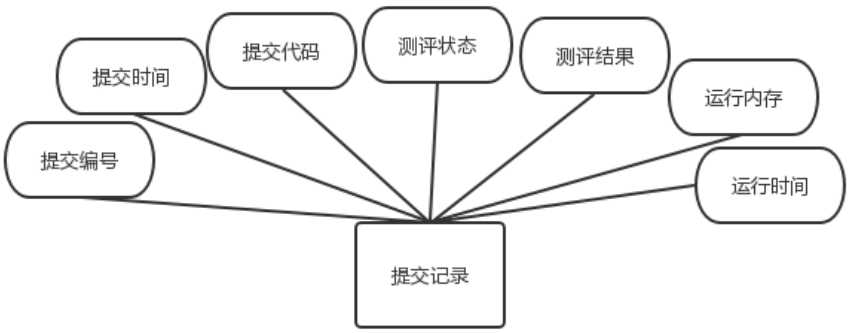


图 10：提交记录 E-R 图

6.1.5 管理员信息

管理员的账户和信息与操作系统用户关联，但是我们在数据库里依然要备份信息。具体的 E-R 图如图 11：

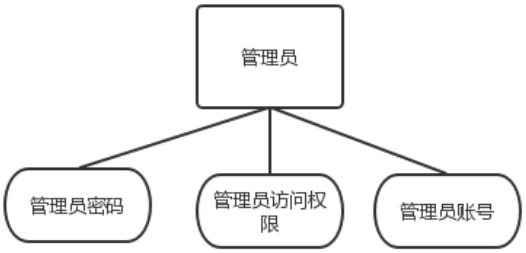


图 11：管理员信息 E-R 图

6.1.6 提问与答疑信息

比赛期间选手可以向管理员提出问题，管理员可以选择做出回答，如图 12：

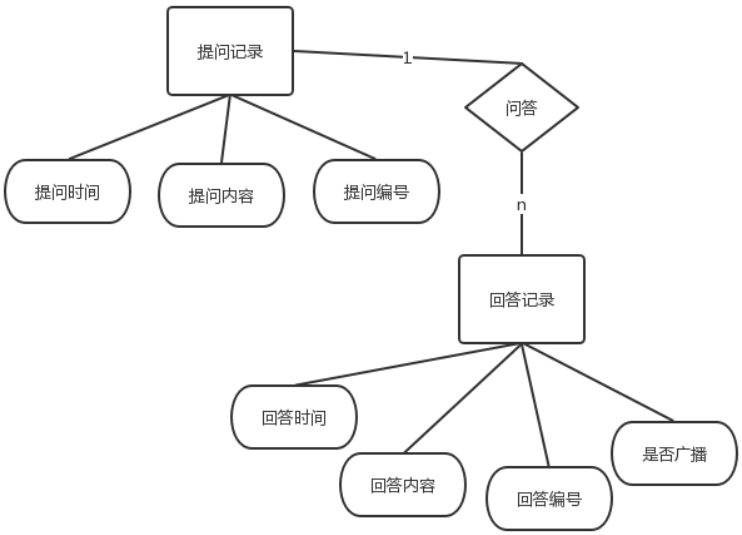


图 12：问答信息 E-R 图

6.1.7 总 E-R 图

总 E-R 图如图 13：

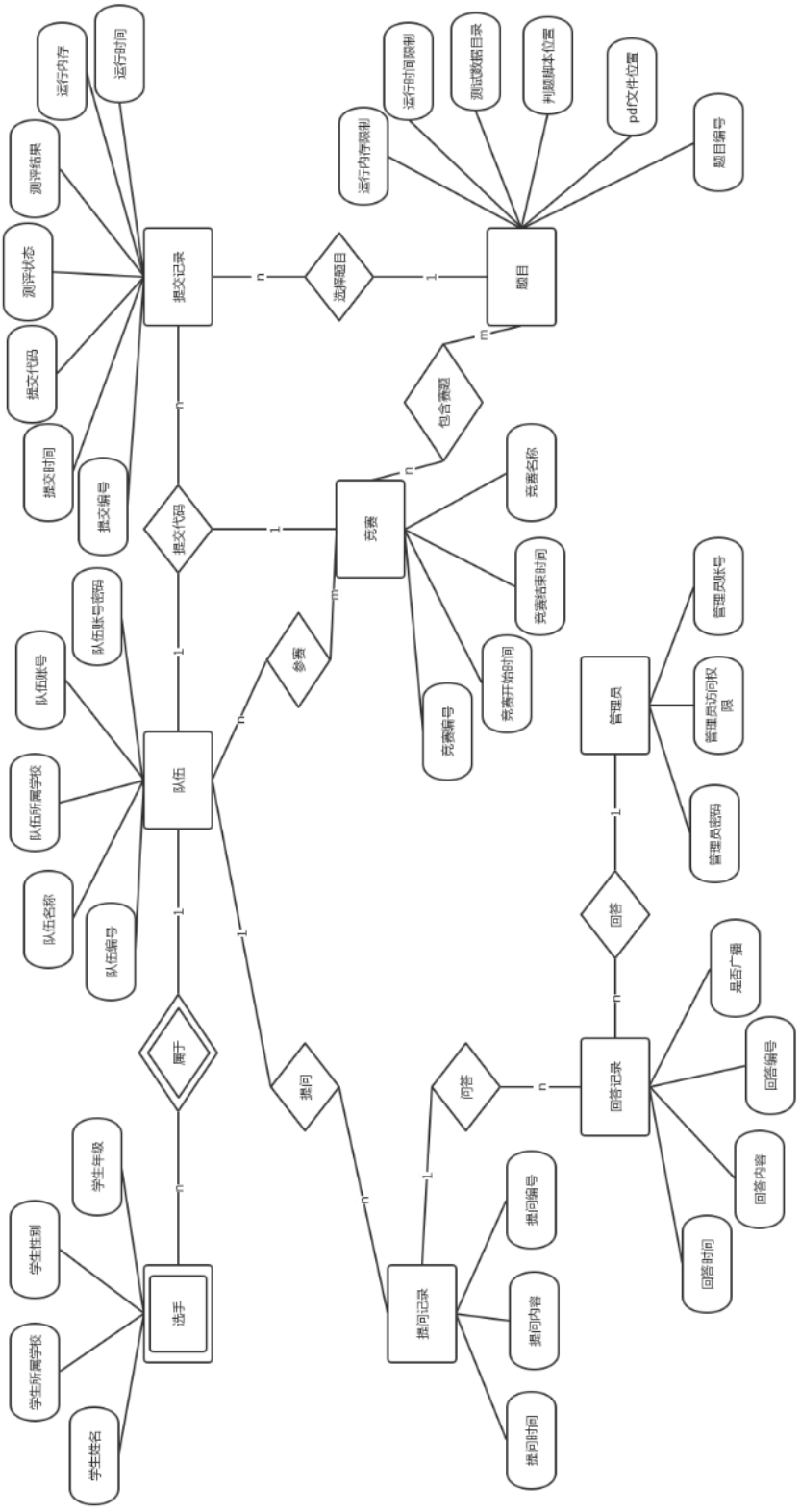


图 13：总 E-R 图

6.2 逻辑结构设计

参考：

department (*id*, campus_id, name)

专业

major (*id*, department_id, name)

课程

course (*id*, name, description, credit, capacity, durat

专业与课程的对应

major_has_course (*major_id*, *course_id*, course_type)

教学班

class (*id*, teacher_id, course_id, year, term)

教师用户

teacher (*id*, user_id, avatar)

管理员用户

6.2 物理结构设计

本系统的数据库表如下：

表 4：队伍信息表

编号	属性名	字段名称	数据类型	长度	备注
1	队伍编号	TID	Char	8	主键
2	队伍名称	TName	Char	256	不空
3	队伍所属学校	TSchool	Char	20	不空
4	队伍账号	TUsername	Char	20	
5	队伍密码	TPassword	Char	20	

表 5：选手信息表

编号	属性名	字段名称	数据类型	长度	备注
1	选手编号	SID	Char	8	主键
2	学生姓名	DName	Char	8	不空
3	学生性别	Sex	Char	2	不空
4	学生年级	SGrade	Char	8	不空
5	队伍编号	TID	Char	8	外键

表 6：竞赛信息表

编号	属性名	字段名称	数据类型	长度	备注
1	竞赛编号	CID	Char	8	主键
2	竞赛开始时间	CST	Datetime	8	不空
3	竞赛结束时间	CET	Datetime	8	不空
4	竞赛名字	CName	Char	156	不空

表 7：题目信息表

编号	属性名	字段名称	数据类型	长度	备注
1	题目编号	PID	Char	8	主键
2	PDF 文件位置	PFile	Char	256	不空
3	判题脚本位置	PScript	Char	256	不空

4	测试数据目录	PData	Char	256	不空
5	运行时间限制	PTimeLimit	Int	4	不空
6	运行内存限制	PMemoryLimit	Int	4	不空

表 8：提交记录信息表

编号	属性名	字段名称	数据类型	长度	备注
1	提交编号	SID	Char	8	主键
2	提交时间	STime	Int	4	不空
3	提交代码	SCode	Char	256	不空
4	测评状态	SState	Char	32	不空
5	测评结果	SResult	Char	32	
6	运行内存	SMemoryUsed	Int	4	
7	运行时间	STimeUsed	Int	4	
8	队伍编号	TID	Char	8	外键
9	竞赛编号	CID	Char	8	外键
10	题目编号	PID	Char	8	外键

表 9：提问记录表

编号	属性名	字段名称	数据类型	长度	备注
1	提问编号	QID	Char	8	主键
2	提问内容	QData	Char	1024	不空
3	提问时间	QTime	Datetime	8	不空
4	回答编号	AID	Char	8	

表 10：回答记录表

编号	属性名	字段名称	数据类型	长度	备注
1	回答编号	AID	Char	8	主键
2	回答内容	AData	Char	1024	不空
3	回答时间	ATime	Datetime	8	不空
4	是否广播	BroadCast	Bool	1	不空
5	管理员账号	Admin	Char	8	外键

表 11：管理员信息表

编号	属性名	字段名称	数据类型	长度	备注
1	管理员账号	Admin	Char	8	主键
2	管理员密码	AdminPassWord	Char	32	不空
3	管理员访问权限	AdminPri	Char	8	不空

表 12：竞赛包含赛题表

编号	属性名	字段名称	数据类型	长度	备注
1	竞赛编号	CID	Char	8	
2	题目编号	PID	Char	8	

表 13：队伍参赛表

编号	属性名	字段名称	数据类型	长度	备注
1	队伍编号	TID	Char	8	
2	竞赛编号	CID	Char	8	

7. 运行设计

这一段主要以用户视角（买家、卖家、管理员等），在使用本系统过程中需要做的操作和预期结果做文字性描述。

8. 系统出错设计

8.1 出错信息

系统输出信息的形式	含义	处理办法或相应对策
数据库连接失败	① 数据库配置出错	① 修改数据库配置 ② 限制并非访

57

8.2 补救措施

8.2.1 系统恢复

系统崩溃后，根据系统运行日志恢复系统和数据，并重新启动系统。

8.2.2 定时备份

（1）周期性地备份数据库中的数据，将其存储在更稳定的介质上，并及时进行校对、更新。

（2）将工程代码通过 GitHub 进行完善的版本管理。

8.2.3 人工操作

当出现紧急情况时，数据库管理员人工地对数据库中数据进行修改，并做相应的记录。

8.3 系统维护设计

由于实验条件有限，我们并不能提供专门的服务器运行系统，故将利用配置较高的PC作为服务器，保证服务器以及客户端间网络畅通即可。

（1）用户在该系统执行操作时应该留下痕迹，以方便检查系统是否被恶意篡改。同时系统管理员定时查看系统日志，统计非法攻击来源和次数，并针对相应攻击加强安全防范措施。

（2）系统管理员的登录不仅需要账户密码，还需要识别 IP，禁止非白名单 IP 的任何访问。

（3）系统维护人员及时更新技术漏洞，通过各种手段防止各种对系统的攻击，增强代码的可靠性。

（4）定期维护数据库，涉及到检查数据库表、检查日志文件等，确保数据库内数据的正确性。

（5）在可能出错的地方使用try-except语句捕获异常，并输出相应的出错信息和可能的处理方法提示。