

# 数据库系统上机报告

西安电子科技大学

计算机科学与技术学院

2203015 班

团队成员：党子清，田一涵，刘恩，包乃淦

2024 年 11 月



学号	姓名	任务分配内容	项目贡献比重
22009201170	党子清	数据库创建, 前后端交互	3.5
22009200561	包乃浈	前端设计	2
22009200652	刘恩	撰写报告	2
22009200239	田一涵	撰写报告	2.5

## 一、开发环境

服务端:

选用的 OS 型号: Windows

选用的 DBMS 型号: DM8

客户端:

OS 系统: 安卓/Windows/IOS

实现:

- 前端
- 后端

## 二、需求分析

### 1.数据库设计需求

- (1) 根据语义描述, 设计数据库概念结构, 绘制 ER 图。
- (2) 将 ER 图转化为关系模型, 明确各关系模式的主码、外码及完整性约束。

### 2.数据库创建需求

- (1) 创建后台数据库。
- (2) 创建基本表, 并建立必要的完整性约束。

### 3.角色与权限需求

- (1) 创建“管理员”和“普通用户”两种角色。
- (2) 管理员角色具有全部权限, 普通用户角色具有 insert、update、select 权限。
- (3) 为普通用户创建视图, 确保数据安全性。

### 4.功能模块需求

- (1) 管理员用户功能:

① 导入 Excel 表数据，包括赛题信息、评委信息、参赛团队信息、团队成员信息、指导教师信息、团队选题信息、作品成绩汇总表。

② 对以上表中的记录进行增加、修改和删除操作。

(2) 普通用户功能：

① 修改团队获奖等级。

② 查询各团队最终得分，并按照赛题编号升序、团队编号升序、最终平均分降序输出。

③ 查询获奖团队详情信息。

④ 生成电子证书信息表。

⑤ 根据团队号查询该团队是否获奖。

基于项目需求，本团队决定采用达梦数据库（DM Database）官方提供的 dmPython ODBC 接口，以实现高级编程语言与数据库之间的交互操作，并利用游标对象进行数据检索和管理。在系统架构设计方面，我们将利用 Flask 框架来构建我们的 Web 应用系统，以确保系统的灵活性和可扩展性。

### 数据库交互层设计：

本项目将采用达梦数据库的 dmPython 模块，该模块作为 ODBC 接口，为 Python 提供了与 DM 数据库的连接和操作能力。

通过配置 ODBC 数据源，建立与 DM 数据库的连接，并利用游标（Cursor）对象执行 SQL 语句，实现对数据库的查询、更新、插入和删除操作。

### 系统架构设计：

系统后端将基于 Flask 框架进行开发，Flask 是一个轻量级的 Web 应用框架，适用于构建中小型的 Web 应用系统。

利用 Flask 的路由系统，我们将定义多个端点，以支持管理员和普通用户的各项功能需求。

通过 Flask 的蓝图（Blueprints）功能，我们将实现模块化的代码组织，提高系统的可维护性和可扩展性。

系统将采用 RESTful API 设计原则，提供清晰的接口定义，便于前端调用和数据交换。

### 安全性与权限控制：

系统将实现基于角色的访问控制（RBAC），分别为管理员和普通用户分配不同的操作权限。

通过 Flask 的认证机制，确保只有授权用户才能访问特定的资源和执行相关操作。

为普通用户创建受限的数据库视图，以保护敏感数据不被未经授权访问。

### 功能模块实现：

系统将提供一系列功能模块，包括数据导入、记录管理、成绩查询、获奖信息检索和电子证书生成等。

各功能模块将通过 Flask 视图函数实现，并遵循 MVC（模型-视图-控制器）设计模式，以实现业务逻辑与表现层的分离。

通过上述设计，本团队将构建一个高效、安全、可扩展的竞赛组织管理系统。

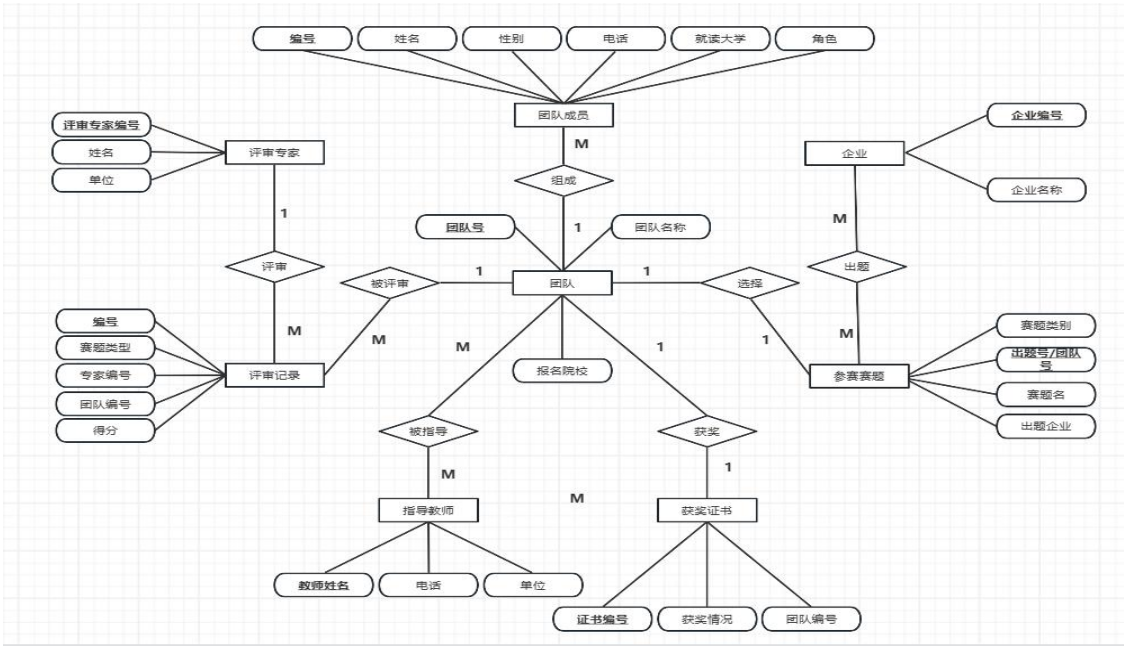
### 三、需求实现

#### 1、实体/模式设计

语义	实体/模式
团队	Team
成员	Member
指导老师	Advisor
A 题	AProblem
B 题	BProblem
专家	Expert
评审信息	ReviewInfo

#### 2、概念结构设计

ER 图模型：



#### 3、逻辑结构设计

团队成员表

编号(主键)
姓名
性别
就读大学
角色
团队编号（外键）

团队表

团队编号（主键）
团队名称

企业表

企业编号（主键）
企业名称

参赛赛题表

编号(主键)
赛题类型
出题与团队编号（外键，关联团队表）
赛题名称
出题企业（外键，关联企业表）

评审专家表

评审专家编号（主键）
姓名
单位

评审记录表

编号(主键)
赛题类型
专家编号(外键，关联评审专家表)
团队编号(外键，关联团队表)
得分

指导教师表

指导编号（主键）
教师姓名
电话
单位

## 8. 获奖证书表

证书编号(主键)
获奖情况
团队编号(外键, 关联团队表)

## 4、关系描述

### 1) 团队与团队成员

关系：一对多（一个团队有多个成员）

外键：团队成员表中的“团队编号”关联团队表。

### 2) 参赛赛题与团队

关系：多对一（一个赛题可以由多个团队选择）

外键：参赛赛题表中的“出题与团队编号”关联团队表。

### 3) 参赛赛题与企业

关系：多对一（一个赛题由一个企业提供）

外键：参赛赛题表中的“出题企业”关联企业表。

### 4) 评审记录与团队、评审专家

关系：

多对一（一个评审记录对应一个团队）

多对一（一个评审记录对应一个评审专家）

外键：评审记录表中的“团队编号”关联团队表，“专家编号”关联评审专家表。

### 5) 团队与指导教师

关系：多对多（一个团队可以有多个指导教师，一个教师可以指导多个团队）

解决方式：增加中间表，包含“团队编号”和“指导编号”作为联合主键。

### 6) 团队与获奖证书

关系：一对多（一个团队可以获得多个证书）

外键：获奖证书表中的“团队编号”关联团队表。

## 三、系统实现

### 1. 后端

本系统采用达梦数据库的 **dmPython 模块**，该模块作为 ODBC 接口，为 Python 提供了与 DM 数据库的连接和操作能力。

通过配置 ODBC 数据源，建立与 DM 数据库的连接，并利用游标（Cursor）对

象执行 SQL 语句，实现对数据库的查询、更新、插入和删除操作。

#### a. 管理员用户

1) 根据给定的以下 Excel 表，将数据导入系统。给定的 Excel 表有“赛题信息、评委信息、参赛团队信息、团队成员信息、指导教师信息、团队选题信息、作品成绩汇总表”。

代码：

```
@app.route('/admin/import_excel', methods=['POST'])
def import_excel():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'}), 400

    file = request.files['file']
    if file.filename == "":
        return jsonify({'error': 'No selected file'}), 400

    if file:
        df = pd.read_excel(file)
        return jsonify({'message': 'Data imported successfully'}), 200

    return jsonify({'error': 'Failed to import data'}), 500
```

2) 增加、修改和删除以上表中记录。

代码：

```
@app.route('/admin/operation', methods=['GET', 'POST'])
def operate():
    if request.method == 'POST':
        operation = request.form['operation']
        try:
            if cursor.execute(operation):
                return jsonify({'message': 'Operation completed successfully'}),
200
        except Exception as e:
            return jsonify({'error': 'Failed to complete operation'}), 500
```

#### b. 普通用户功能

1) 查看团队获奖等级。

代码：

```

@app.route('/user/team_award', methods=['GET', 'POST'])
def team_award():
    team_id = request.form["teamId"]
    if not team_id:
        return jsonify({'error': 'Team ID is required'}), 400

    cursor.execute(f'SELECT TeamID, Award, TeamName FROM Team WHERE TeamID = {team_id}')
    result = cursor.fetchone()

    if result:
        # return jsonify({'队伍号': result[0], '获奖情况': result[1]}), 200
        return render_template('team_award_output.html', team_id=result[0],
award=result[1], team_name=result[2])
    else:
        return jsonify({'error': 'No award found for the given Team ID'}), 404

```

2) 查询各团队最终得分。输出信息有“团队编号，团队名称，团队选题，团队最终平均分”，要求按照赛题编号升序，团队编号升序，最终平均分降序输出。

代码：

```

@app.route('/user/team_score', methods=['GET', 'POST'])
def team_score():
    cursor.execute("""
SELECT TeamID, TeamName, ProblemName, SCORE FROM Team
WHERE SCORE IS NOT NULL
ORDER BY score DESC;
""")
    data = cursor.fetchall()
    return render_template('team_score_output.html', data=data)

```

3) 查询获奖团队详情信息，输出”序号、赛题类别，所获奖项、团队编号，团队名称，赛题名称，所属院校，团队成员姓名，团队成员角色，指导教师姓名，指导教师院校“。

代码：

```

@app.route('/user/team_details', methods=['GET', 'POST'])
def team_details():
    team_id = request.form["teamId"]
    if not team_id:

```



```

        return jsonify({'error': 'Team ID is required'}), 400
    cursor.execute(f"""
        SELECT
            Team.TeamID AS 团队编号,
            Team.TeamName AS 团队名称,
            Team.SCORE as 得分,
            Team.Award AS 所获奖项,
            Team.Category AS 赛题类别,
            Team.ProblemName AS 赛题名称,
            Team.School AS 所属院校,
            LISTAGG(Member.Name, ';') WITHIN GROUP (ORDER BY
Member.Role) AS 团队成员姓名,
            Advisor.Name AS 指导教师姓名,
            Advisor.Department AS 指导教师院校
        FROM
            Team
        JOIN
            Member ON Team.TeamID = Member.TeamID
        JOIN
            Advisor ON Team.School = Advisor.Department
        WHERE
            team.TeamID = {team_id}
        GROUP BY
            Team.TeamID, Team.TeamName, Team.SCORE, Team.Award,
Team.Category, Team.ProblemName, Team.School, Advisor.Name,
Advisor.Department
        """)
    data = cursor.fetchone()
    team_memeber = data[7].split(';')
    team_memeber = ','.join(list(set(team_memeber)))
    if data:
        return render_template('team_details_output.html', data={'队伍号': data[0],
'队伍名': data[1], '得分': data[2],
'获奖情况': data[3], '赛题类别': data[4], '赛题名': data[5],
'所属学校': data[6], '团队成员': team_memeber, '指导老
师': data[8]})
    else:
        return jsonify({'error': 'No award found for the given Team ID'}), 404

```

4) 生成电子证书信息表。信息有“团队编号，团队名称，团队成员姓名，

团队成员院校，指导教师姓名，指导教师院校 “。

代码：

```
@app.route('/user/generate_certificate', methods=['GET', 'POST'])
def generate_certificate():
    team_id = request.form['teamId']
    if not team_id:
        return jsonify({'error': 'Team ID is required'}), 400
```

5) 根据给定的团队号，查询该团队是否获奖。

代码：

```
cursor.execute(f"""
    SELECT
        Team.TeamID AS 团队编号,
        Team.TeamName AS 团队名称,
        Team.SCORE as 得分,
        Team.Award AS 所获奖项,
        Team.Category AS 赛题类别,
        Team.ProblemName AS 赛题名称,
        Team.School AS 所属院校,
        LISTAGG(Member.Name, ';') WITHIN GROUP (ORDER BY
Member.Role) AS 团队成员姓名,
        Advisor.Name AS 指导教师姓名,
        Advisor.Department AS 指导教师院校
    FROM
        Team
    JOIN
        Member ON Team.TeamID = Member.TeamID
    JOIN
        Advisor ON Team.School = Advisor.Department
    WHERE
        team.TeamID = {team_id}
    GROUP BY
        Team.TeamID, Team.TeamName, Team.SCORE, Team.Award,
Team.Category, Team.ProblemName, Team.School, Advisor.Name,
Advisor.Department
""")
data = cursor.fetchone()
team_memeber = data[7].split(';')
team_memeber = ','.join(list(set(team_memeber)))
if data:
```

```

return render_template('generate_output.html', data={'队伍号': data[0], '队伍名': data[1], '得分': data[2],
                                                    '获奖情况': data[3], '赛题类别': data[4], '赛题名': data[5],
                                                    '所属学校': data[6], '团队成员': team_memeber, '指导教师': data[8]})
else:
    return jsonify({'error': 'No award found for the given Team ID'}), 404

```

## 数据库代码部分：

### 1. 创建表：

```

-- 创建团队表
CREATE TABLE Team (
    NUM INT NOT NULL,
    TeamID CHAR(7) NOT NULL,
    TeamName VARCHAR(50),
    School VARCHAR(50),
    Category VARCHAR(10),
    ProblemName VARCHAR(100),
    Score DECIMAL(5, 2),
    Award VARCHAR(10),
    CONSTRAINT PK_Team PRIMARY KEY (NUM)
);

-- 创建成员表
CREATE TABLE Member (
    NUM INT NOT NULL,
    Name VARCHAR(50) NOT NULL,
    TeamID CHAR(7) NOT NULL,
    Gender VARCHAR(10) NOT NULL,
    Phone VARCHAR(20) NOT NULL,
    University VARCHAR(50) NOT NULL,
    Role VARCHAR(10) NOT NULL CHECK (Role IN ('队长', '副队长', '队员')),
    CONSTRAINT PK_Member PRIMARY KEY (NUM)
);

```

### 2. 创建用户

```

CREATE TABLE IF NOT EXISTS managers (
    username VARCHAR(255) PRIMARY KEY,
    password VARCHAR(255)
);

-- 创建管理员用户
CREATE USER admin with password '123456';

-- 赋予管理员所有权限
GRANT ALL PRIVILEGES ON DATABASE TO admin;

-- 创建普通用户
CREATE USER user with password '123456';

-- 赋予普通用户基本的INSERT, UPDATE, SELECT权限
GRANT INSERT, UPDATE, SELECT ON TABLE Team TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE Member TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE Advisor TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE AProblem TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE BProblem TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE Expert TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE ReviewInfo TO user;
GRANT INSERT, UPDATE, SELECT ON TABLE AProblem TO user;

```

### 3. 插入数据：

```
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (1, '【A01】', '外贸企业碳管理SAAS系统开发', '【阿里巴巴】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (2, '【A02】', '基于百度飞桨的3D医疗数据智能解析平台', '【百度】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (3, '【A03】', '智能家居场景语音识别挑战赛', '【移动创新院】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (4, '【A04】', '基于机器学习实现涉诈网址自动分类识别', '【移动创新院】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (5, '【A05】', 'MasterGo插件设计开发: HTML to Mastergo', '【零微数广】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (6, '【A06】', '基于多场景应用的对象存储平台', '【深信服】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (7, '【A07】', '冰箱制冰模块创新', '【海尔创新设计中心】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (8, '【A08】', '空调技术创新', '【海尔创新设计中心】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (9, '【A09】', '智能代码溯源分析系统', '【东软集团】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (10, '【A10】', '智能家居远程控制系统', '【东软集团】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (11, '【A11】', '智能政策解读系统', '【浪潮卓数】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (12, '【A12】', '高性能产线识别平台', '【长安计算】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (13, '【A13】', '景区室内外游览任务调度与群组推荐平台', '【长安计算】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (14, '【A14】', '基于在线教学平台的数据挖掘与学习行为分析', '【超星集团】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (15, '【A15】', '虚拟化身——2D虚拟人语音驱动算法', '【万兴科技】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (16, '【A16】', '虚拟试衣算法', '【万兴科技】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (17, '【A17】', '安全的人脸识别认证系统', '【极维网络】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (18, '【A18】', '随手买-随车智能营销平台', '【融创软通】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (19, '【A19】', '基于云计算平台构建教学管理系统', '【华云数据】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (20, '【A20】', '面向汽车烘干设备智能协同APP设计与开发', '【雷浪云】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (21, '【A21】', '面向汽车排产的生产预测仿真系统的设计与开发', '【雷浪云】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (22, '【A22】', '基于容器海量数据的存储系统', '【信华信】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (23, '【A23】', '基于区块链的元宇宙数字作品知识产权保护服务平台', '【云象网络】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (24, '【A24】', '基于区块链的分布式电力交易身份及数据隐私保护', '【云象网络】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (25, '【A25】', '订单驱动的制造产业链完整性与风险评估方案设计与实现', '【华粮振宇】');
INSERT INTO AProblem (NUM, ProblemID, ProblemName, Company) VALUES (26, '【A26】', '基于工业互联网数据驱动的设备维护与故障预测', '【五维世航】');
```

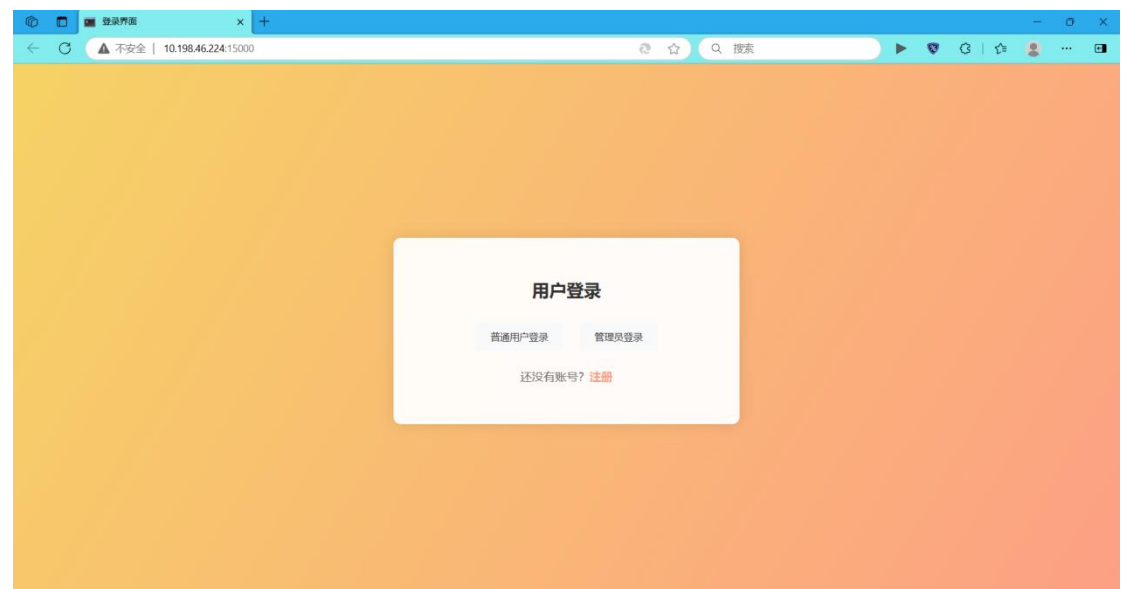
## 五. 界面展示

### 前端实现

采用 html 语言结合 css，实现前端的设计。

### 前端展示

登陆与注册界面：



用户登录

普通用户登录

管理员登录

用户名

密码

登录

还没有账号？[注册](#)

用户注册

用户名

密码

确认密码

注册

[返回首页](#)

管理员界面：

创赛结果查询

高效管理 · 精准统计 · 便捷服务

管理员操作

管理员使用sql语句直接对数据库操作

导入excel表格

导入各参赛团队的获奖等级及相关评定信息

查看团队获奖等级

查询各参赛团队的获奖等级及相关评定信息

查询各团队最终得分

查看团队评分详情及总分排名情况

查询获奖团队详情信息

获取获奖团队的详细资料及联系方式

生成电子证书信息表

一键生成团队成员电子版获奖证书

退出登录

© 逐梦华清团队信息管理系统

普通用户界面



管理员直接操作界面：



查看团队获奖等级：



查询队伍详情：



结果展示：



团队获奖详情

队伍号	2203104
队伍名	咖啡冲不冲
得分	92.67
获奖情况	一等奖
赛道类别	A
赛道名	【A34】医学文献智能识别与检索系统设计【慧逊科技】
所属学校	电子科技大学
团队成员	谭宇乔,赵秘,陈艺昌,张云博,刘松涛
指导老师	陈安龙

返回首页

生成电子证书：

电子证书信息表

队伍号	2203104
队伍名	咖啡冲不冲
得分	92.67
获奖情况	一等奖
赛道类别	A
赛道名	【A34】医学文献智能识别与检索系统设计【慧逊科技】
所属学校	电子科技大学
团队成员	谭宇乔,陈艺昌,刘松涛,赵秘,张云博
指导老师	陈安龙

返回首页