

# 深 圳 大 学

## 实 验 报 告

课程名称: 数据库系统

实验序号: 实验 4

实验名称: 数据库设计

学 号: 2022155026

姓 名: 龚驰

实验完成日期: 2024 年 12 月 14 日

## 一、实验目的：

- 1、掌握数据库设计方法；
- 2、了解概念模型、逻辑数据模型和物理数据模型之间的关系和不同；
- 3、掌握使用高级语言访问、操作数据库，加深对前后台数据交互的理解。

## 二、实验要求：

1. 确定选题，并进行需求分析，用高级语言实现一个小型数据库应用系统；
2. 完成一个小型系统的数据库设计，绘制 E-R 图；
3. 将 E-R 图转成逻辑数据模型和物理数据模型，导出 sql 脚本，创建数据库，E-R 图至少包括 8 个实体和 7 个联系；设计至少 1 个视图、1 个索引（非主键、非外键索引）、1 个触发器、1 个存储过程或存储函数（视图、索引、触发器、存储过程或存储函数可以直接在 DBMS 中创建，不一定要在 pdm 图的时候创建）；
4. 需要考虑关系完整性约束：主键约束、外键约束、空值约束；
5. 数据库使用 MySQL；
6. 编程语言不限；
7. 考察重点是数据库设计、前后台数据交互。图形界面是必须的，以简单明了为上。

## 三、实验设备：

计算机、数据库管理系统如 php，mysql 等软件。

## 四、实验内容

1、按要求完成数据库实验指导书，完成一个小型系统的设计与实现，除了指导书的明确要求，其他可以按照自己的思路拓展功能。（数据库设计 30 分，CDM 绘制及转换 20 分、可执行对象 20 分，前端 20 分，文档 10 分）

我的选题为**校园活动征招平台**，任务要求如下：

校园活动征招平台	<ol style="list-style-type: none"><li>1、用户注册（普通用户、管理员）；</li><li>2、管理员新增、修改、删除活动信息（时间、地点、人数、要求……）；</li><li>3、普通用户浏览活动信息和申请参加活动；</li><li>4、管理员可查询活动的申请人清单，并对申请人进行审核（通过或拒绝）；</li><li>5、普通用户能看到自己提交的申请是待审核、通过还是拒绝状态；</li><li>6、过期的活动系统自动设置为关闭状态。</li></ol>
----------	--

（1）首先是在 PowerDesigner 上面设计数据库的 E-R 图，并完成 Cdm、Ldm、Pdm 的绘制。

1) 概念数据模型（CDM）

实体：

User、User\_type、Common\_User、Managed\_User、Activity、Application、Application\_result

关系：

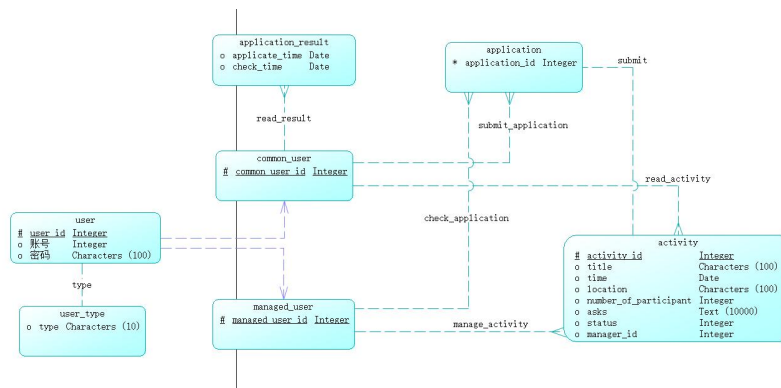
user 与 user\_type 通过 type 关联。

common\_user 与 application 通过 common\_user\_id 关联。

managed\_user 与 activity 通过 managed\_user\_id 关联。

application 与 activity 通过 activity\_id 关联。

application 与 application\_result 通过 application\_id 关联。



Cdm 图

## 2) 逻辑数据模型 (LDM)

实体:

user 包含属性 user\_id, password

user\_type 包含属性 user\_id, type

common\_user 包含属性 common\_user\_id

managed\_user 包含属性 managed\_user\_id

activity 包含属性 activity\_id, common\_user\_id, managed\_user\_id, title, time, location, number\_of\_participant, asks, status, manager\_id

application 包含属性 common\_user\_id, managed\_user\_id, activity\_id, application\_id

application\_result 包含属性 common\_user\_id, apply\_time, check\_time

关系:

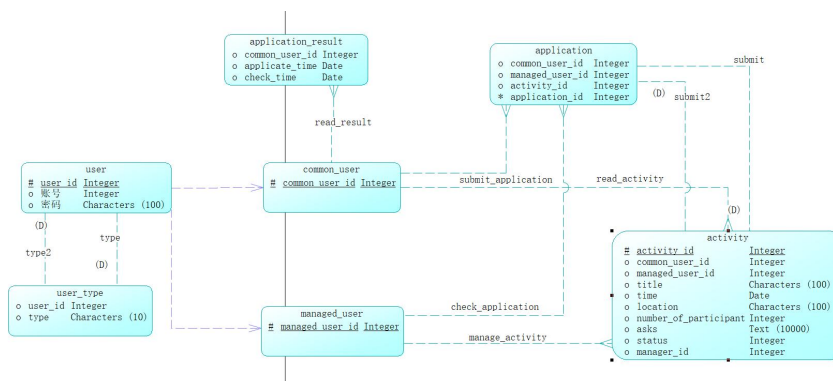
user 与 user\_type 通过 user\_id 关联。

common\_user 与 application 通过 common\_user\_id 关联。

managed\_user 与 activity 通过 managed\_user\_id 关联。

application 与 activity 通过 activity\_id 关联。

application 与 application\_result 通过 common\_user\_id 关联。



Ldm 图

### 3) 物理数据模型 (PDM)

实体:

user 包含属性 user\_id, password, 其中 user\_id 是主键。

user\_type 包含属性 user\_id, type, 其中 user\_id 是主键。

common\_user 包含属性 common\_user\_id, 其中 common\_user\_id 是主键。

managed\_user 包含属性 managed\_user\_id, 其中 managed\_user\_id 是主键。

activity 包含属性 activity\_id, managed\_user\_id, title, time, location, number\_of\_participant, asks, status, manager\_id, 其中 activity\_id 是主键。

application 包含属性 application\_id, common\_user\_id, managed\_user\_id, activity\_id, 其中 application\_id 是主键。

application\_result 包含属性 common\_user\_id, apply\_time, check\_time。

关系:

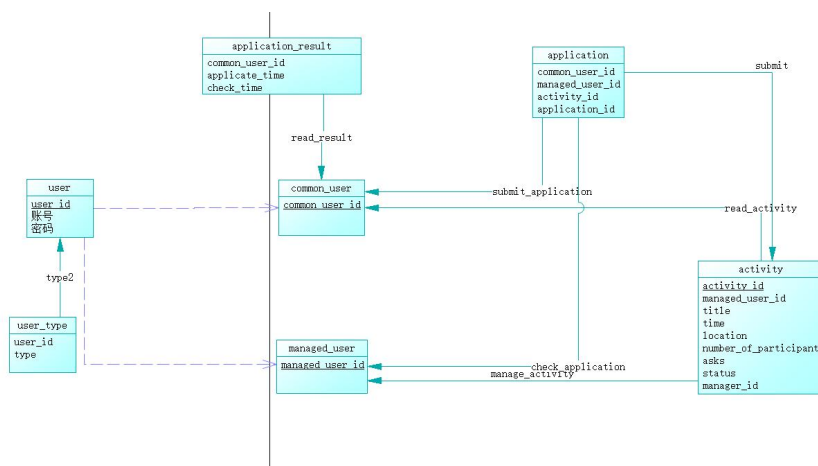
user 与 user\_type 通过 user\_id 关联。

common\_user 与 application 通过 common\_user\_id 关联。

managed\_user 与 activity 通过 managed\_user\_id 关联。

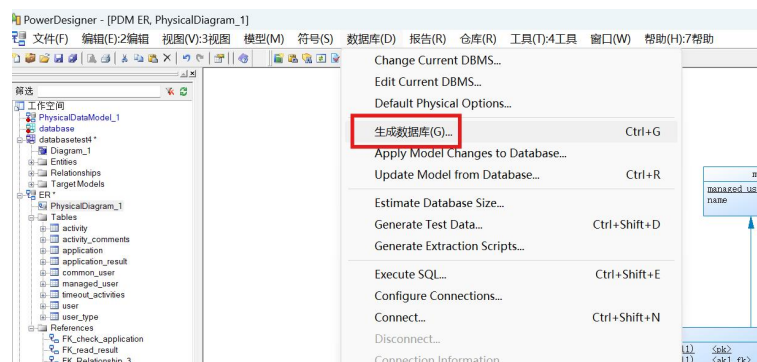
application 与 activity 通过 activity\_id 关联。

application 与 application\_result 通过 common\_user\_id 关联。



Pdm 图

(2) 绘制完 E-R 图并检查无误后，导出为 SQL 文件。然后将 SQL 文件导入进 MySQL 中创建数据库。



最终在 MySQL 中创建的数据表如下：

表	操作	行数	类型	排序规则	大小	多余
<input type="checkbox"/> activity	★ 浏览 结构 搜索 插入 清空 删除	4	InnoDB	latin1_swedish_ci	48 KB	-
<input type="checkbox"/> activity_comments	★ 浏览 结构 搜索 插入 清空 删除	3	InnoDB	latin1_swedish_ci	48 KB	-
<input type="checkbox"/> application	★ 浏览 结构 搜索 插入 清空 删除	2	InnoDB	latin1_swedish_ci	64 KB	-
<input type="checkbox"/> application_result	★ 浏览 结构 搜索 插入 清空 删除	2	InnoDB	latin1_swedish_ci	48 KB	-
<input type="checkbox"/> common_user	★ 浏览 结构 搜索 插入 清空 删除	2	InnoDB	latin1_swedish_ci	16 KB	-
<input type="checkbox"/> managed_user	★ 浏览 结构 搜索 插入 清空 删除	1	InnoDB	latin1_swedish_ci	16 KB	-
<input type="checkbox"/> timeout_activities	★ 浏览 结构 搜索 插入 删除 视图	~0	视图	---	-	-
<input type="checkbox"/> user	★ 浏览 结构 搜索 插入 清空 删除	3	InnoDB	latin1_swedish_ci	16 KB	-
<input type="checkbox"/> user_type	★ 浏览 结构 搜索 插入 清空 删除	3	InnoDB	latin1_swedish_ci	16 KB	-
9 张表	总计	~20	InnoDB	latin1_swedish_ci	272 KB	0 字节

其中，创建了一个视图 timeout\_activity，用于查看过期或管理员手动关闭的活动。

#	名字	类型	排序规则	属性	空	默认	注释	额外
<input type="checkbox"/> 1	activity_id	int(11)		否	0			
<input type="checkbox"/> 2	title	char(100)	latin1_swedish_ci	是	NULL			
<input type="checkbox"/> 3	time	date		是	NULL			
<input type="checkbox"/> 4	location	char(100)	latin1_swedish_ci	是	NULL			
<input type="checkbox"/> 5	number_of_participant	int(11)		是	NULL			
<input type="checkbox"/> 6	asks	text	latin1_swedish_ci	是	NULL			

在 activity 表中创建了一个基于时间字段的索引，用于用户在报名活动时，可以选择相应时间进行查看。

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
<input type="checkbox"/> 1	activity_id	int(11)		否	无	AUTO_INCREMENT			修改 删除 更多
<input type="checkbox"/> 2	managed_user_id	int(11)		是	NULL				修改 删除 更多
<input type="checkbox"/> 3	title	char(100)	latin1_swedish_ci	是	NULL				修改 删除 更多
<input type="checkbox"/> 4	time	date		是	NULL				修改 删除 更多
<input type="checkbox"/> 5	location	char(100)	latin1_swedish_ci	是	NULL				修改 删除 更多
<input type="checkbox"/> 6	number_of_participant	int(11)		是	NULL				修改 删除 更多
<input type="checkbox"/> 7	asks	text	latin1_swedish_ci	是	NULL				修改 删除 更多
<input type="checkbox"/> 8	status	int(11)		是	NULL				修改 删除 更多

操作	键名	类型	唯一	紧凑	字段	基数	排序规则	空	注释
编辑 删除	PRIMARY	BTREE	是	否	activity_id	4	A	否	
编辑 删除	FK Relationship 3	BTREE	否	否	managed_user_id	1	A	是	
编辑 删除	idx activity time	BTREE	否	否	time	4	A	是	

创建了一个触发器，用于检测活动时间是否已经过期，过期则自动将活动状态改为关闭。然后创建了一个时间，每隔一分钟调用一次触发器，检测活动状态。如下图所示：

时间 事件

编辑触发器

详细

触发器名称

set\_status\_to\_closed\_if\_past

表

activity

时机

BEFORE

事件

INSERT

定义

1 BEGIN  
2 IF NEW.time < CURDATE() THEN  
3 SET NEW.status = 2;  
4 END IF;  
5 END

用户

root@localhost

编辑事件

详细

事件名称

CheckAndUpdateExpiredActivities

状态

ENABLED

事件类型

RECURRING

运行周期

1 MINUTE

起始时间

2024-12-13 14:33:27

终止时间

定义

1 BEGIN  
2 UPDATE activity  
3 SET status = 2  
4 WHERE time < CURDATE();  
5 END

过期后禁用事件而不删除

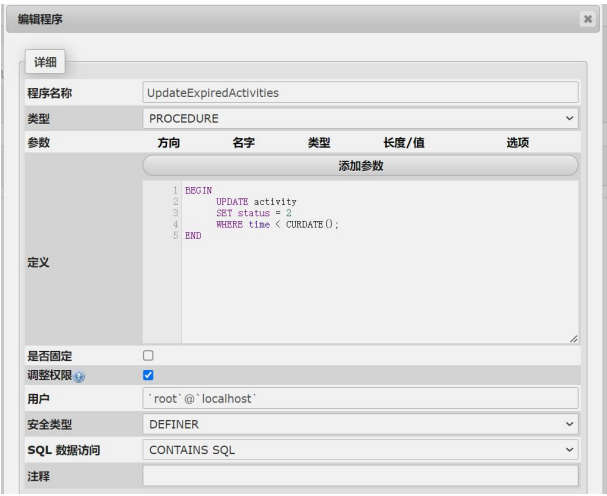
☐

用户

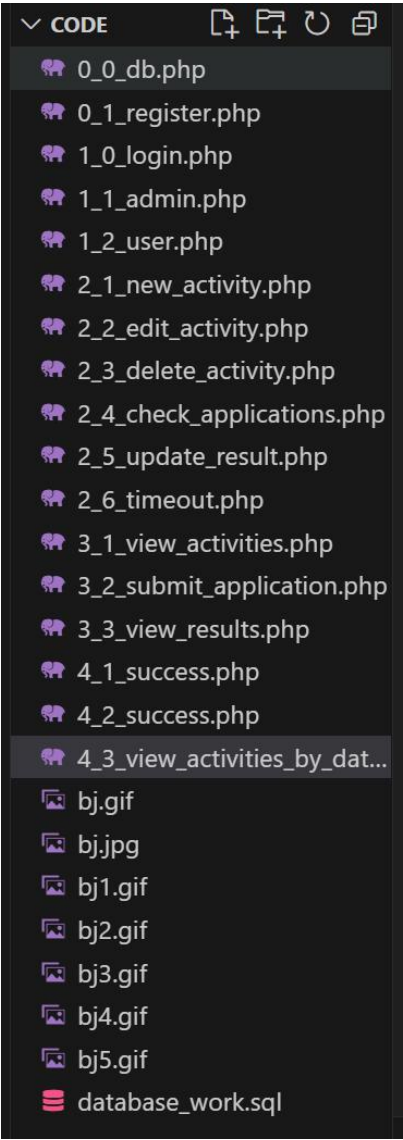
root@localhost

注释

创建了一个存储过程，当管理员在修改活动信息，将活动日期改成早于当前日期时，自动将活动状态更改为关闭。



(3) 前端设计,所撰写的代码文件如图，每个文件都对应一个页面。





2、附上相关关键问题的解决方案，以及运行结果，运行屏幕截图。

1) 在打开该校园活动征召系统时，首先看到的是登录界面，如果用户尚未注册，可以点击下方的注册新用户按钮，进行账户注册。



2) 在注册界面，用户在填写账号密码以及姓名之后，点击注册按钮便完成了账户的注册，接下来可以在该平台进行登录。



3) 如果是管理员用户进行登录，登录成功后会跳转到管理员界面，该界面包含六个功能，分别是新增活动、修改活动、删除活动、过期活动、审核申请、退出登录。



a) 新增活动，管理员可以在该页面进行新增活动的操作，并为活动设置相关信息。

新增活动

返回管理员界面

填写活动信息

管理员 ID:

活动标题:

活动时间:

年 / 月 / 日

活动地点:

参与人数:

活动问题:

活动状态:

活动开始

提交

b) 修改活动，管理员可以在该界面进行修改活动的操作，包括活动的各种信息。

修改活动

返回管理员界面

选择活动进行修改

STUDY

SLEEP

PLAY

TS STUDY

修改活动: STUDY

活动标题:

STUDY

活动时间:

2024/12/15

活动地点:

classroom

参与人数:

10

活动要求:

Let's study!

活动状态:

活动开始

提交修改

c) 删除活动，管理员可以在该界面进行删除活动的操作，将活动状态设置为关闭。

删除活动

返回管理员界面

选择活动进行删除

STUDY

SLEEP

PLAY

时间: 2024-12-15

地点: classroom

参与人数: 10

要求: Let's study!

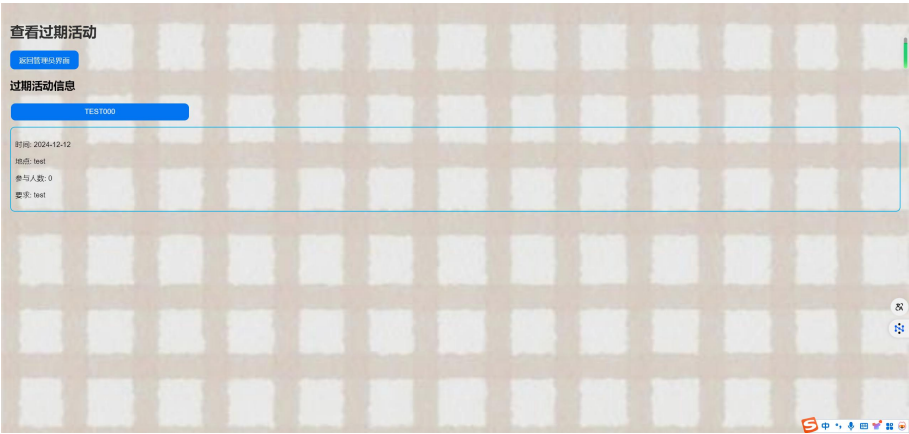
确认删除

localhost:3000 显示  
确定要删除这个活动吗?

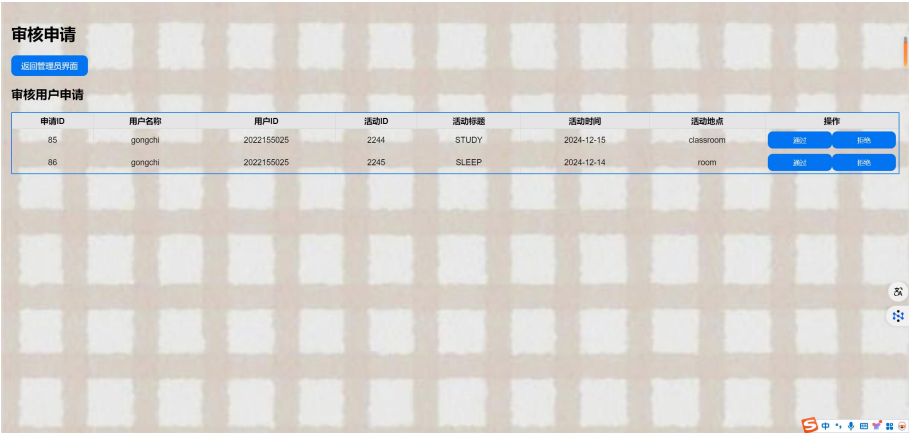
确定 取消



d) 查看过期活动，管理员可以在该界面查看过期活动。



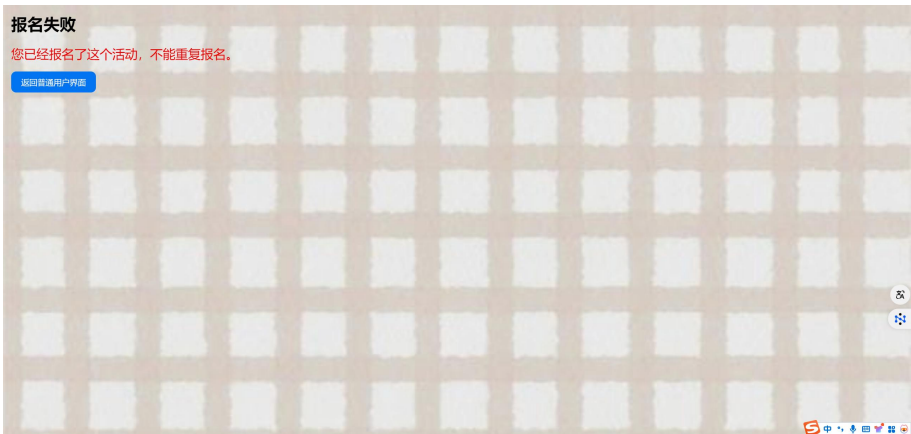
e) 审核申请，管理员可以在该界面进行审核用户申请的操作，通过申请或者拒绝申请。



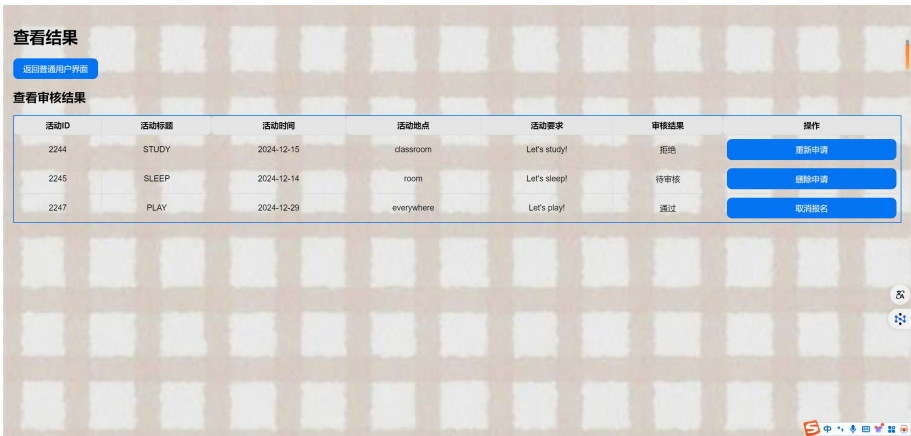
a) 查看活动，用户可以在该页面查看活动信息，并选择活动进行报名。



如果用户以及报名了该活动，这提示用户不能重复报名。



b) 查看结果，用户可以在该页面查看报名结果，如果报名失败，可以选择重新报名；如果报名成功，可以选择取消报名；如果报名未审核，可以选择重新报名；



3、（可选）录一小段视频，3 分钟左右，对自己的亮点进行解说展示。

## 五. 问题分析（碰到什么问题，如何解决）

在本次实验中，遇到了以下几个问题：

①数据库设计复杂性：设计一个包含多个实体和关系的数据库需要深入理解数据库范式和 E-R 模型。通过使用 PowerDesigner 工具来绘制 E-R 图，并将其转换为逻辑数据模型和物理数据模型，解决了这一问题。

②前端与后端的数据交互：在开发过程中，我需要确保前端界面能够正确地与后端数据库进行数据交互。通过使用 PHP 语言和 MySQL 数据库，实现了数据的增删改查功能，并确保了数据的一致性和安全性。

③界面设计和用户体验：为了使系统更加用户友好，需要设计直观且易于使用的界面。通过 HTML 和 CSS 进行了前端设计，并确保了界面的响应性和美观性。

④性能优化：在处理大量数据时，查询性能会受到影响。通过创建索引和优化查询语句，提高了数据库的查询效率。

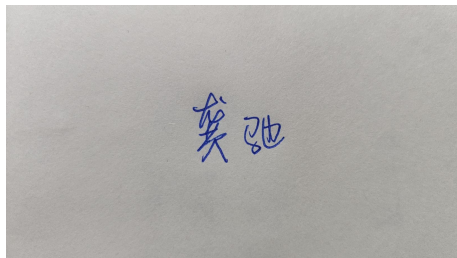
## 六. 实验心得

通过这次实验，我体会到了数据库设计、前端与后端数据交互以及系统实现的复杂性与挑战性。实验过程中，我不仅将理论知识与实践相结合，还学会了如何在遇到问题时进行有效分析和解决。这次实验经历不仅增强了我的技术能力，也培养了我的创新思维和解决问题的能力，为我未来的学术研究和职业发展奠定了坚实的基础。

## 七. 诚信承诺

本组成员郑重承诺在项目实施的过程中不发生任何不诚信现象，一切不诚信所导致的后果均由本组成员承担。

签名（手签，数字上传）： 龚驰



指导教师批阅意见：

成绩评定：

指导教师签字：  
年 月 日

备注：