

（深圳）

实验报告

开课学期： 2021春季

课程名称： 数据库系统

实验名称： HITBlog博客系统设计与实现

实验性质： 设计型

实验学时： 2 地点： T2210

学生班级： 计算机7班

学生学号： 180110704

学生姓名： 段裕

评阅教师：

报告成绩：

实验与创新实践教育中心制

2021年1月

# 实验环境

*请填写用到的操作系统和主要软件版本。*

实验环境：

Windows 10

编程语言：

Python

解释器版本：

Python 3.7.10

主要第三方库依赖：

PyMySQL 1.0.2

Flask 1.1.2

jinja2 2.11.3

Werkzeug 1.0.1

编程工作站：

PyCharm 2019.3.3

网站教程参考：

1. HTML/CSS/Java Script

<https://www.w3school.com.cn/>

1. Flask 教程

<https://dormousehole.readthedocs.io/en/latest/>

1. 其它文章

<https://www.cnblogs.com/sxs161028/p/7278219.html>

<https://www.cnblogs.com/qq3245792286/p/6390504.html>

等

# 实验过程

## 系统功能

*请结合文字、表格、截图等方式，清晰描述系统的功能。亮点功能请用\*标志。*

一、文字描述

1、用户的【注册】与【登录】

2、发布【文字博客】，博文带标签

**\*3**、根据【标签】找到同标签的其它文章

4、可以在博文下进行【评论】

5、给文章【点赞】

**\*6**、给评论点赞

**\*7**、【收藏夹】功能，可以新建收藏夹，将文章放入收藏夹

**\*8**、登录后查可看【个人主页】，具体信息包含

8-1、【粉丝】列表

8-2、【关注】列表

8-3、【博客摘要】列表

8-4、收藏夹列表

8-5、【个人信息】（例如：手机、邮箱、性别，注册时间等）

**\*9**、登录后查看【他人主页】

9-1、【部分非敏感信息】

9-2、【关注该用户】

\*10、游客模式查看【用户主页】

10-1、【部分信息】

10-2、更多的操作需要登录

11、修改个人主页

仅可以修改【手机】、【邮箱】、【昵称】

其它的不可以改

12、博客主页是近期较热文章列表，无论是否登录都可以查看内容

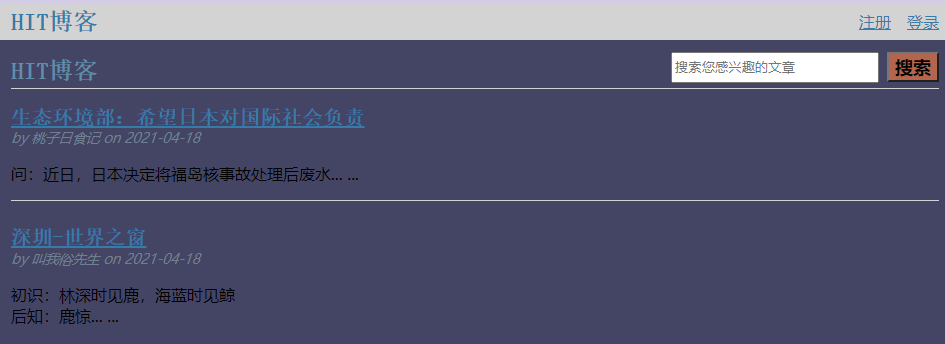
13、输入关键字根据标题进行检索

1. 截图说明

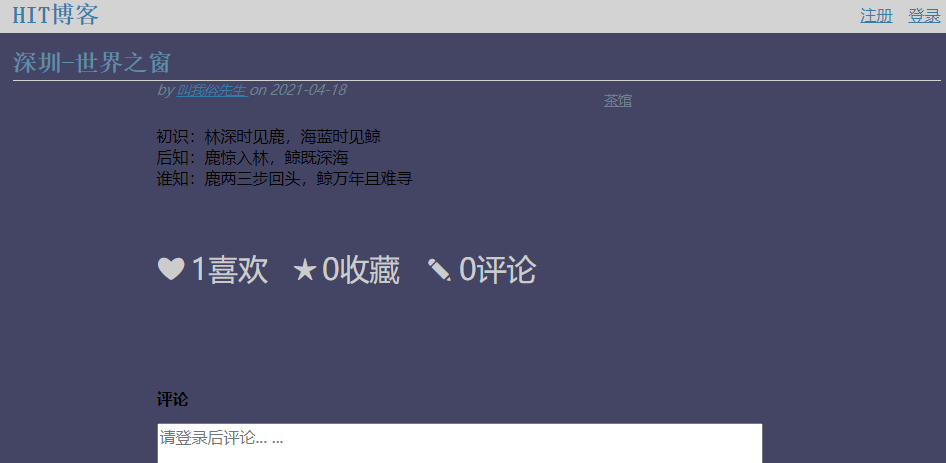
说明，为方便演示，提前注册了两个用户，他们各自发了一篇文章，同时也有部分评论

**A、未登录状态**

1、主页，核心是一个文章列表



1. 点击文章进入详情列表，可以查看内容，但评论喜欢等操作无法进行

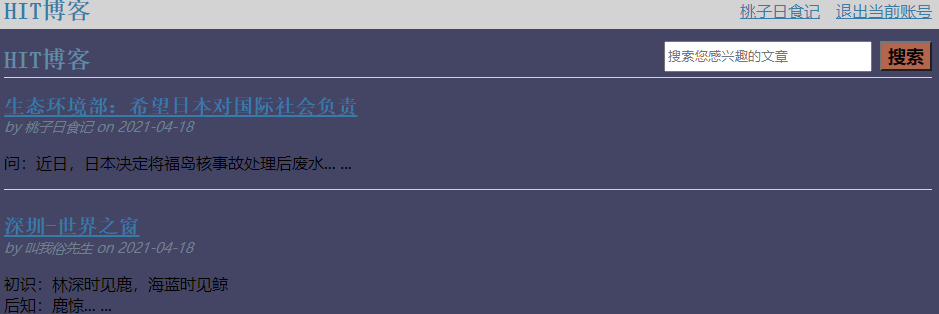


1. 点击文章作者名，可以进入TA的个人主页

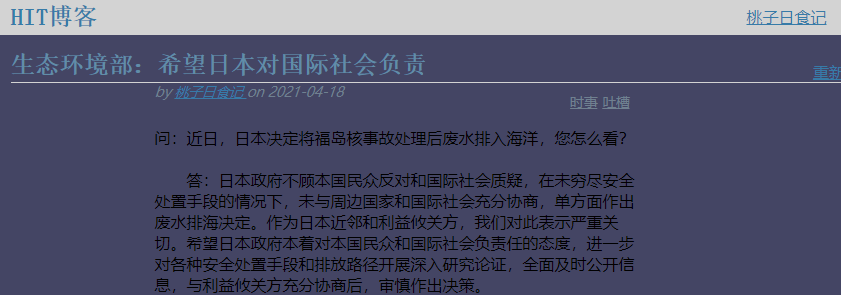


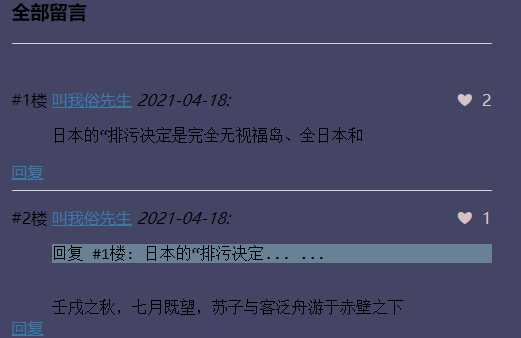
1. **登录账号桃子日食记**



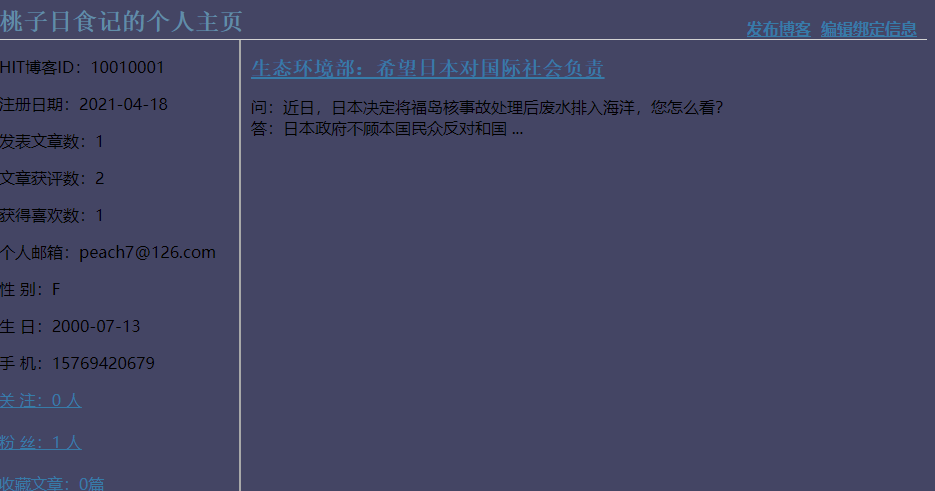


点击文章进入详情页，可以点赞、评论、收藏，如果是自己的文章，可以重新编辑或者删除，删除后文章以及相关的评论等都会被清空

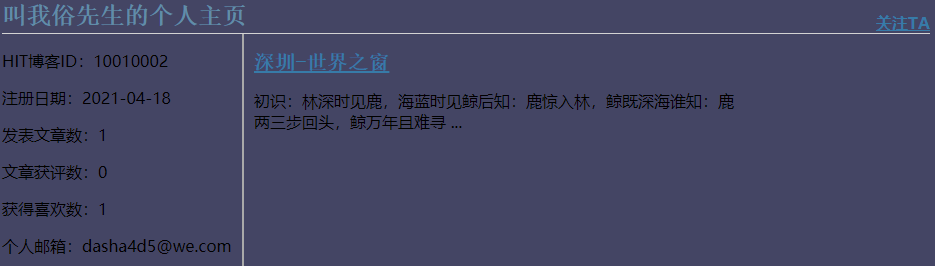




点击昵称进入个人主页可以发布博客、查看关注粉丝列表，查看收藏夹和新建收藏夹等操作。



点击左上角HIT博客回到主页，点击另一篇文章，作者为**叫我俗先生**，点击进入TA的个人主页：



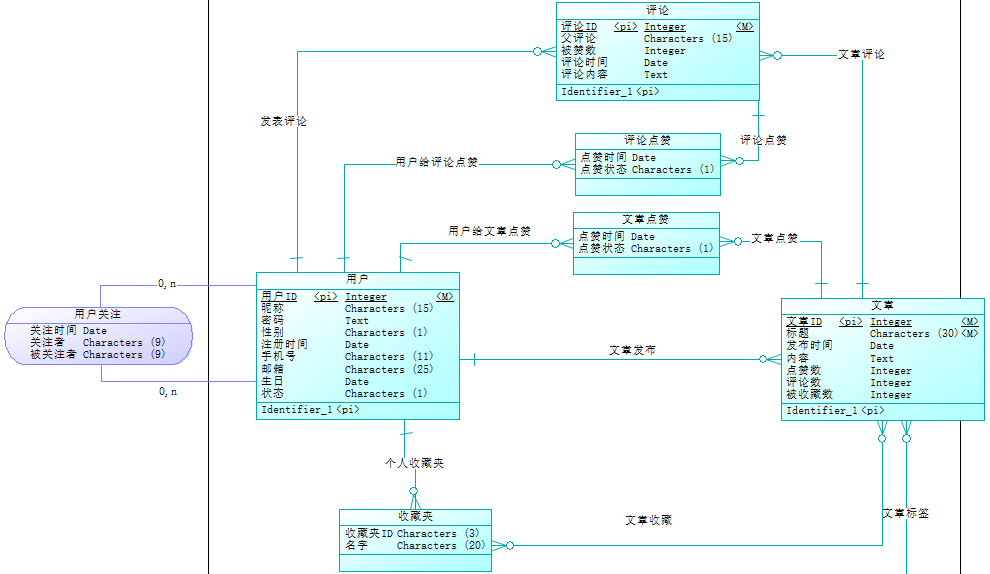
看到右上角变为关注的字样，意为你可以关注TA

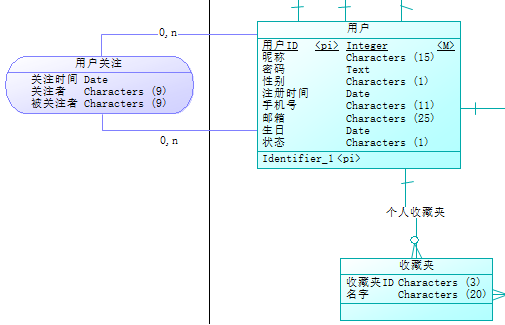
1. 搜索文章以及根据标签归类文章，以及文章收藏等等，这些放在视频中一一演示，不再在此列举

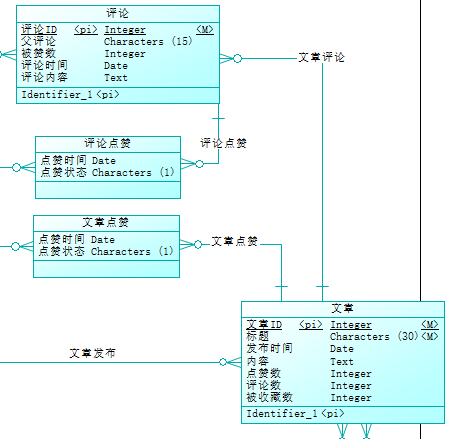
## 数据库设计

### ER图

*要求：截图务必清晰，如果图太大可截图一个总图，然后截几个部分图展示。如果看不清截图会影响成绩。*

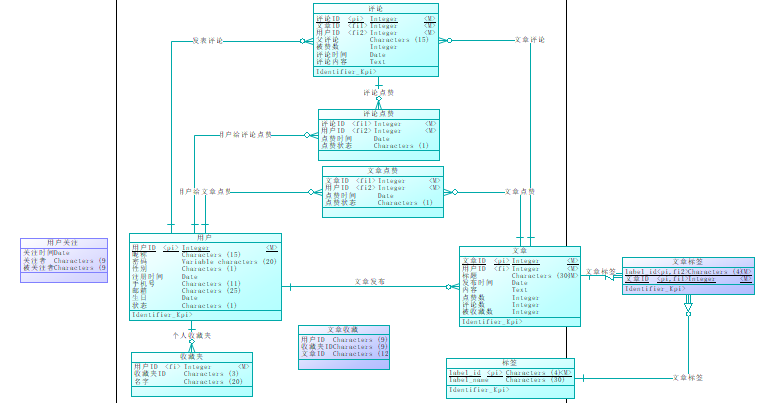


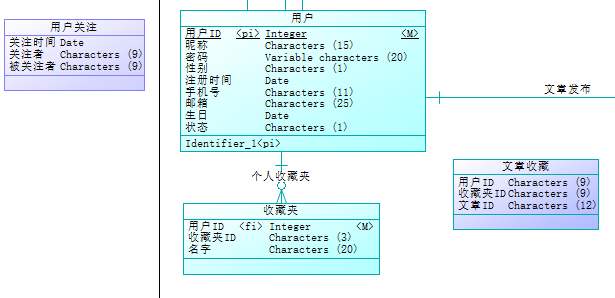


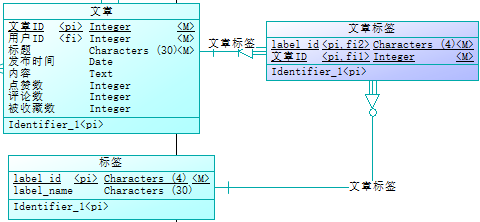


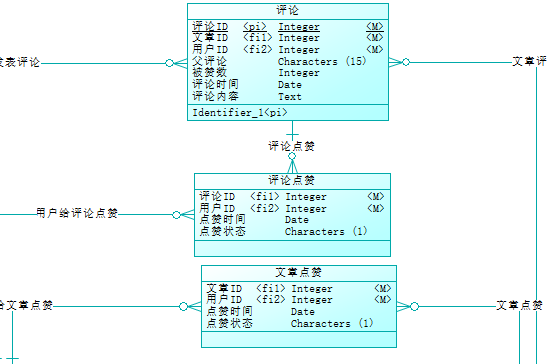
### LDM图

*要求：截图务必清晰，如果图太大可截图一个总图，然后截几个部分图展示。如果看不清截图会影响成绩。*



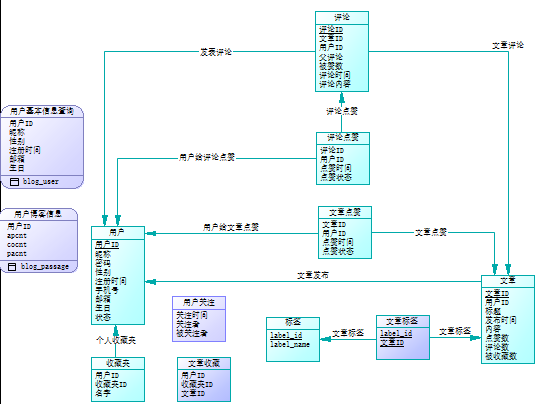






### PDM图

*要求：截图务必清晰，如果图太大可截图一个总图，然后截几个部分图展示。如果看不清截图会影响成绩。*

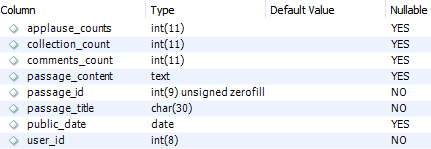


### 数据库表结构

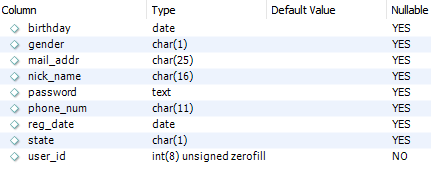
1. **表结构**

*选取几个比较有代表性的表结构截图，体现主键约束、外键约束、空值约束。*

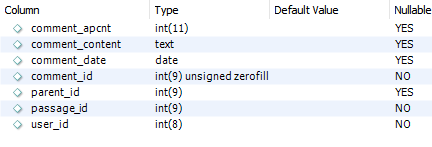
博客表



用户表



评论表



这三个表是最核心的表，博客最基本的功能就是用户注册登录，发布浏览博文，以及博文下的评论回复。E

**用户表**：存放用户的基本信息，包含手机、邮箱、昵称等等，每一个用户都有一个编号，同时作为表的主键

**博客表：**存放博客信息，存放博文信息，包含标题、内容与发布日期，另外还有一些统计变量，例如收藏数、评论数、喜欢数，这里涉及到三对触发器和至少三个表，每一篇博客都有唯一的ID作为主键，同时记录是谁发表的文章，即user-id作为外键，这两个字段均非空。

**评论表：**评论表的设计考虑的点更为复杂，在实现的时候也遇到重重阻碍。下面将一一说明。

一是表的设计，评论首先是有两个依赖，也就是谁评论了哪一篇文章，那么user\_id和passage\_id都将作为外键，同时每个评论也要有自己的编号comment\_id作为主键，另外评论还可以依附另一个评论，也即评论是可以被回复的。这里有两种思路，一个是另建一个回复表，一个是增加一个父评论ID字段，根据课上理论所讲，我倾向于后一种做法，这样感觉更好理解。另外如果评论不是回复，父评论ID置0.这样做的另一个优点是，查询一篇文章的所有评论只用查一张表，只要passage\_id匹配就行。

二是前端的设计，这里也有两种方式，可以在一个评论下追加各个回复，也可以将评论回复统一起来，按楼层展示，对于回复，需要展示父评论的楼层以及简略内容。这里我思前想后使用了第二种方式，这样统一管理的方式更方便。

三就是前后端数据的交互了，需要互相传值，传的数据也较复杂，这是一个较大的瓶颈，几乎使我放弃。

1. **索引**

*讲解你建的非主索引，说明为什么要建这个索引。*

1、用户名索引

通过用户名找到对应的记录，获取用户基本信息，一般而言，对外展示的是昵称而非一串ID号，通过昵称进行索引可以保证大多数使用情况下的高效。

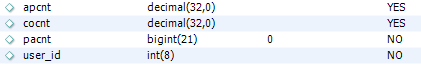
2、文章标题索引

理由与用户名索引类似，通过标题直接获取文章信息。

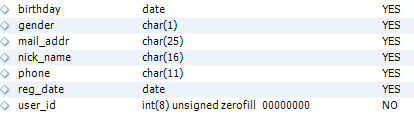
1. **视图**

*讲解你建的视图，说明为什么要建这个视图。*

统计用户博客信息，包含获赞、获评、发表文章数等，用于展示个人主页



统计用户基本信息，相当于user的一个子表



1. **触发器（选做）**

*讲解你为什么建触发器和如何实现触发器。*

1. trigger\_com\_appcnt

create trigger com\_appcnt

after insert on comment\_applause

for each row

begin

update comments

set comments.comment\_apcnt = comments.comment\_apcnt + 1

where comments.comment\_id = new.comment\_id;

End

评论点赞是单独的表，用户点赞时，会插入一条新记录（如果之前没有的话）

2、trigger\_upd\_comappcnt

create trigger upd\_comappcnt

after update on comment\_applause

for each row

begin

update comments

set comment\_apcnt = comment\_apcnt + (new.ap\_status - old.ap\_status)

where comments.comment\_id = new.comment\_id;

End

我之前赞了，可我又不想赞了，再然后又想赞了，然后又取消了...。用一个标志位标记点赞状态，0到1表示我想赞，1到0就是赞了后又收回赞。

3、trigger\_passage\_appcnt

4、trigger\_upd\_pasappcnt

类似与1和2，只不过现在是文章点赞

1. trigger\_update\_pascomcnt
2. trigger\_delcom

分别是插入和删除文章评论

1. trigger\_default\_collector

create trigger default\_collector

after insert on blog\_user

for each row

begin

insert into collector values(new.user\_id, '001', '默认收藏夹');

End

每新注册一个用户，添加一个默认收藏夹

1. trigger\_ins\_cp
2. trigger\_del\_cp

分别表示收藏文章和取消收藏时用户收藏文章总数的变化

1. **事务（选做）**

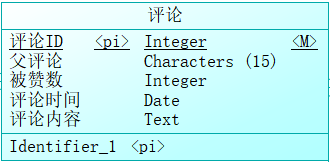
*讲解你为什么建事务和如何实现事务。*

### 分析

*选择较为有代表性的实体和联系进行分析，讲解如何从ER图到LDM图、再到PDM图，最后变成数据库表结构的转换过程（参考实验二PPT里“补充知识 三、ER图🡪LDM图🡪PDM图”中老师讲的例子）。*

这里就以**评论**这个实体作为示例进行分析，它涉及到的关系较多，是博客主体核心之一。

1. ER图



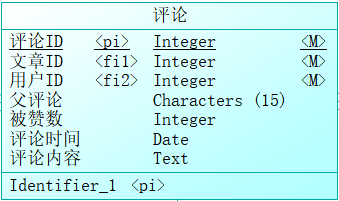
ER图反映一个实体所需要的属性，暂时不考虑与其它实体的联系，在构建ER图时需要着重思考的是：

我在实现一个什么样的东西？它有什么功能？为此它应该具有什么特性？

这一步是确定需求，例如对于一个评论，它首先需要一个主键ID，其次有一个评论内容，考虑到回复等情况，还需要一个父评论ID，此外评论的时间以及被点赞的数量等也可以添加

1. LDM图

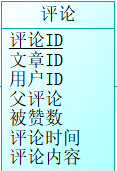
这个时候需要把第一步（需求分析）得到的各个实体间的联系考虑清楚，是一对多还是一对一还是多对多，实体是否部分参与联系，实体间是否有一些可实体化（带属性）的动作。例如，对于实验PPT中的用户与图书，动作可以有借书、还书等，这些动作都可以成为实体。在我考虑的HITBlog中，若以评论为中心进行举例，一个**用户**可以发表很多评论，也可以不发表评论，一条评论由且仅由一个用户所发表，同理，一篇**文章**下可以有很多评论，也可以没有评论，一条评论在且仅在一篇文章下（特指那条评论，不考虑内容），一个用户可以赞很多评论，也可以不赞，一条评论可以没有赞，也可以被很多用户赞...等等



这里，添加了两个Foreign Key

1. PDM图

实际上，经过前两步的打磨，我们在一定程度上把任务要求的实体以及实体关系都考虑好了，至少，在后期可能需要修改的只是表的某个字段的添加删除或修改，又或者添加视图、触发器等优化效率，而不是新增表或放弃表。这一步我们将之前的构想转为物理实体视图同时生成SQL语句。



需要说明的是，SQL语句可能不一定完全符合语法，可以将生成的SQL文件放到MySQL Workbench里实际执行一遍，然后手动插入删除更新观察表数据是否合乎预期，测试无误后就可以考虑具体实现细节。

# 收获和反思

*请填写本次实验的收获，记录实验过程中出现的值得反思的问题及你的思考。*

一分耕耘、一份收获，只有不断踩坑，我们才能不断进步，但同时也要懂得避坑，否则只会陷入困境。

具体说来，本次实验我认为较为成功，符合我的预期，有很多的收获。其一是一种建模的思想方法的加深理解，好的开头是成功的一半。假如没有实验二的铺垫，我可能不知道从何下手，只觉得哇好难，怕是顶不住哦，会造成焦虑并且浪费很多时间。我更加理解了，只有先弄明白做什么才能有下一步具体的实施方案细则。

其二是对于前端知识的接触以及与后台数据交互的接触。在这之前完全没接触过，做完实验后只能说我接触过了，了解从前端到后端实现的一种或几种方式。接下来我想谈一谈我的开发历程。

在做完实验二后，我立即开始了实验三的准备，首先当然是老师给的参考资料，因为这些资料一般更可靠，我选择了python + Flask，尝试创建虚拟环境，安装依赖，运行第一个Flask版的Hello World，后来渐渐了解到python的三个框架django flask tornado等，此外还有vue等，也了解到java在这一类开发上才是大头，同时flask是轻量级框架，django是更加完善的重量级框架。在这个阶段我找到的是flask中文官方教程，先跟着做一个示例，示例是一个很简单的博客应用，示例做完了，里面用到的和老师给的示例现在看来也差不多（只是代码数量多少的差别），示例里用到的是内置的sqlite数据库。紧接着我开始学习pymysql库，做出两个尝试，一个是仅使用python命令以及SQL语句就可以连接SQL Server并建立数据库以及表结构，一个是把示例的实现转为我博客的实现。这一阶段第一个工作量较大的阶段，我初步实现了用户登录，也就是blog\_user这一个实体是完全可控的。至于博客，因为还有评论点赞收藏等一系列的东西牵扯，暂时还没有对应的前端视图。

接下来的阶段可能就是真正头大的阶段了，因为Flask中文教程示例能学的都学到了，前端很多东西不知道怎么展现，除了form表单提交数据别无他法，这一阶段我在寻找HTML教程，先是考虑如何使点赞评论收藏样式看着差不多还要水平排布，等等，细节不多说。在寻找教程的同时我发现即使是flask开发也是遵循一定的模式的，例如MVC模式，数据的处理统统交给Model，这是封装好的class，视图函数用于处理响应请求等，而在我已有的系统中是直接执行SQL不存在Model这一说法，代码看起来不规范。我更加忧心了，我需要推倒重来吗，还是应该继续寻找突破口？因为此时我忽然发现，我没法灵活的向HTML发送数据，也没法在满足约束条件的情况下从前台拿到数据，很多东西也不知道怎么展示，这里就是评论回复的瓶颈，如何显示回复框？如何实现楼层分层？评论后如何立即显示在网页上，如何把一系列信息交给后台同时不影响前台运行？

我决定先抛开问题，认真找一些教程看看，终于，评论回复功能渐渐的也都实现了，这期间学习到了Java script 与HTML DOM的一些知识，那么到现在，我已经知道HTML元素操纵的魔法啦！这之后，各个功能迅速完善起来，当然只是相对开始做的时候来说，这其中也遇到许多坑，有一种突破的感觉。