一级项目论文报告

目录

1 绪论 （e\*）

本次一级项目选择的题目是“利用python开发一个校内论坛”。

论坛是供人们作讨论的地方。讨论题材有很多，例如：娱乐、新闻、教育、旅游、休闲等等。有些论坛设有多项讨论题材，包罗万有，有的则只专注讨论某题材。在论坛中，很多用户还会跟他人分享资源， 例如：音乐、短片、图片等。有些公司、机构又或是学校，都会设有论坛，供该会的成员作讨论之用。[[1]](#footnote-1)

1.1 课题研究的背景和意义

在汕头大学，随着校内学生对信息获取需求的增加，如，学校公布信息的获取，课程表信息的获取，拼车信息的获取等，一个个对应的信息提供方出现，如，汕大课程表，微信拼车群，微信e出行拼车公众号，微信二手物品群等。

但是，这里面有这么几个奇怪的现象，汕大课程表的日流量很多，app里面有一个公共讨论的区域，很可惜，虽然流量多，但是这个使用这个功能的人很少；金凤BT作为一个BT资源提供的网站，其中一个主要的区域就是公共发言区，可是，每一页的讨论里面都会有一两条的其他消息：二手物品出手，失物招领，电脑问题求助等。

为解决汕大学生信息获取、发布的困难，一个校内论坛就显得很有必要。

1.2 主要研究内容 （需求、主要功能、主要技术）

为满足需求，论坛的主要功能有：发帖回帖；账号验证登陆；个人信息管理；信息查询（用户、历史帖子）；管理员系统；站内消息系统。

将上述功能按照板块来进行分类：

* 主页

1. 导航栏：搜索框、用户主页入口、首页入口、收到的消息、注销
2. 分区： 板块分类、分区

* 个人中心

1. 头像显示、名字、发帖历史
2. 访客访问页面：展示信息
3. 个人设置修改内容：头像、昵称（数据库中唯一）、编辑帖子（删除修改…）、其他个人信息、修改密码、回复历史

* 登陆注册

1. 登陆：用户名（昵称）、密码、忘记密码
2. 注册：用户民（昵称）、密码、确认密码、邮箱（正则表达式-校内邮箱）

* 单个帖子页面（post detail）

1. 标题、帖子正文、发帖人、发帖时间

* 板块列表（post list）

1. 帖子列表展示

* 发帖页面

1. 标题、帖子正文、发帖区域

为了方便论坛的维护，在众多的架构中，采用传统的MVC架构——View、Controller、Model

在View部分，主要实现网页的界面，采用的技术有：html5、JavaScript、CSS等（注：前端同学提供）

在Control部分，主要实现业务逻辑，实现的语言采用Python

在Model部分，主要实现数据的维护和更新，采用的技术有：MySQL数据库

由于采用的是Python语言来进行开发，从架构的角度来说，采用Python的Django框架较为合适，一方面是因为这个框架功能齐全，另一方面是因为，这个框架符合MVC的开发架构。

在数据库方面选择传统的MySql 数据库管理器。

在代码版本管理方面选择git来进行管理，并把代码托管到github上。

2 相关技术   
2.1 git （e\*）

Git是一种代码管理系统，属于分布式代码管理系统，由linux之父于2005年创建，目的是用于管理linux社区庞大的代码库，在此之后git变得流行。Git流行的原因要 得益于自身的分布式设计。在git之前，传统的代码版本管理系统都是集中式管理系统，这种系统的原理好比图书馆，所有的源代码都在一个中央服务器上面，如果使用代码就要从中央服务器上面下载。而分布式代码管理系统的原理，就好比每个人都是一个图书馆。源代码也是放在一个中央服务器，如果要使用代码，就把整个版本库下载到本地。

对比两种系统，分布式系统的好处是：1.安全性比集中式管理系统高。因为每个人都有自己的版本库，即使出现中央服务服务器不能工作或者中央服务器代码丢失的情况，也不会影响项目的开发；2.对网络依赖性小。集中式管理系统必须要通过联网才能进行工作，而且网络的速度直接影响项目开发，分布式系统对网络依赖性很小，因为每个开发者都有一个版本库。

由于这次项目在代码管理方面没有其他的要求，又，鉴于以上两个原因，我们选择分布式作为项目的代码管理系统，分布式代码管理系统有很多，我们选择其中最流行的git，并把代码托管到远程仓库GitHub上。  
2.2 Python （福龙）  
2.3 Django （福龙）  
2.5 MySql （张盛）

**介绍**

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

MySQL的优缺点

1. **MySQL的优点：**1． 它使用的核心线程是完全多线程，支持多处理器。   
   2． 有多种列类型：1、2、3、4、和8字节长度自有符号／无符号整数、FLOAT、DOUBLE、CHAR、VARCHAR、TEXT、BLOB、DATE、TIME、DATETIME、 TIMESTAMP、YEAR、和ENUM类型。   
   3． 它通过一个高度优化的类库实现SQL函数库并像他们能达到的一样快速，通常在查询初始化后不该有任何内存分配。没有内存漏洞。   
   4． 全面支持SQL的GROUP BY和ORDER BY子句，支持聚合函数(COUNT()、COUNT(DISTINCT)、AVG()、STD()、SUM()、MAX()和MIN())。你可以在同一查询中混来自不同数据库的表。   
   5． 支持ANSI SQL的LEFT 0UTER JOIN和ODBC。   
   6． 所有列都有缺省值。你可以用INSERT插入一个表列的子集，那些没用明确给定值的列设置为他们的决省值。   
   7． MySQL可以工作在不同的平台上。支持C、C＋＋、Java、Perl、PHP、Python和TCL API。
2. **MySQL的缺点：**1、 MySQL最大的缺点是其安全系统，主要是复杂而非标准，另外只有到调用mysqladmin来重读用户权限时才发生改变。   
   2、 MySQL的另一个主要的缺陷之一是缺乏标准的RI（Referential Integrity-RI）机制；Rl限制的缺乏(在给定字段域上的一种固定的范围限制)可以通过大量的数据类型来补偿。   
   3、 MySQL没有一种存储过程(Stored Procedure)语言，这是对习惯于企业级数据库的程序员的最大限制。   
   4、 MySQL不支持热备份。   
   5、 MySQL的价格随平台和安装方式变化。Linux的MySQL如果由用户自己或系统管理员而不是第三方安装则是免费的，第三方案则必须付许可费。Unix或Linux 自行安装 免费 、Unix或Linux 第三方安装 200美元，   
   SQL Server的优点众多，但是Microsoft SQL Server和其他数据库产品相比也存在着以下劣势：  
   1开放性。只能运行在微软的windows平台，没有丝毫的开放性可言。  
   2可伸缩性，并行性。并行实施和共存模型并不成熟，很难处理日益增多的用户数和数据卷，伸缩性有限。  
   3性能稳定性。SQLServer当用户连接多时性能会变的很差，并且不够稳定。  
   4使用风险。SQLServer完全重写的代码，经历了长期的测试，不断延迟，许多功能需时间来证明。并不十分兼容早期产品。使用需要冒一定风险。  
   5客户端支持及应用模式。只支持C/S模式。

安装

**（1）Python-MySQL**

要想使python可以操作mysql 就需要MySQL-python驱动，它是python 操作mysql必不可少的模块。

下载地址：<https://pypi.python.org/pypi/MySQL-python/>

下载MySQL-python-1.2.5.zip文件之后直接解压。进入MySQL-python-1.2.5目录:

>>python setup.py install

参考：<http://www.cnblogs.com/fnng/p/3565912.html>

**（2）Windows安装MYSQL查询软件**

Navicat for Mysql

安装与使用参考：<http://www.cnblogs.com/sshoub/p/4321640.html>

django中的操作方法

在django中使用了ORM（对象关系映射）的方法，用于实现面向对象编程语言里不同类型系统的数据之间的转换，换言之，就是用面向对象的方式去操作数据库的创建表以及增删改查等操作。

1 ORM使得我们的通用数据库交互变得简单易行，而且完全不用考虑复杂的SQL语句。快速开发，由此而来。

2 可以避免一些新手程序猿写sql语句带来的性能问题。

对数据进行增删改查

**查**

models.UserInfo.objects.all()

models.UserInfo.objects.all().values('user')    #只取user列

models.UserInfo.objects.all().values\_list('id','user')    #取出id和user列，并生成一个列表

models.UserInfo.objects.get(id=1)

models.UserInfo.objects.get(user='yangmv')

**增**

models.UserInfo.objects.create(user='yangmv',pwd='123456')

obj = models.UserInfo(user='yangmv',pwd='123456')

obj.save()

dic = {'user':'yangmv','pwd':'123456'}

models.UserInfo.objects.create(\*\*dic)

**删**

models.UserInfo.objects.filter(user='yangmv').delete()

**改**

models.UserInfo.objects.filter(user='yangmv').update(pwd='520')

或者

obj = models.UserInfo.objects.get(user='yangmv')

obj.pwd = '520'

obj.save()

2.6 ‘css框架’（俊宇、春红）

2.7…其他工具

**3 网站的设计与实现**

**3.2 概要设计（e\*）**

3.2.1 模块划分与分工（e\*）

3.2.2 MODEL的表结构（张盛）

创建表结构

文章表

-标题

-发布板块 [外键关联板块]

-上传文件 [图片]

-文章内容

-文章作者 [外键用户表]

-发布日期

-优先级

-浏览数

返回文章标题和作者

评论表

-评论的文章

-评论的用户 [外键用户表]

-评论的内容

-评论时间

-多级评论

返回评论者

点赞表

-被点赞文章

-点赞用户

-点赞时间

分区表（板块）

-板块名称

-板块管理员

用户表（继承django自带的用户组）

-名字（昵称）

-属组

-头像

-个性签名

返回用户名

用户组表

-用户组名

返回组名  
3.2.3 模块依赖关系（张盛）  
评论部分：django提供的comments模块

用户注册登录部分：django自带的auth模块

admin后台：django自带的admin模块

------------------↑↑↑↑↑↑↑期中检查部分↑↑↑↑↑↑↑↑-------------------

3.3 详细设计

--各个模块的详细介绍

4 模块功能测试

参考文献

1. https://zh.wikipedia.org/wiki/网络论坛 [↑](#footnote-ref-1)