

**公众号系统介绍及其中间件分析**

**课程报告**

课程名称：中间件实现技术

指导老师：胡鹤

学 生：马晓臣

学 号：2019104253

时 间：2020年10月31日

[1.系统总体介绍 3](#_Toc55068838)

[2.所做工作 3](#_Toc55068839)

[3.如何运行 3](#_Toc55068840)

[4.系统功能实现 4](#_Toc55068841)

[5.中间件——session 7](#_Toc55068842)

[6.中间件——sqlalchemy-ORM&dBSession 9](#_Toc55068843)

# ****1.系统总体介绍****

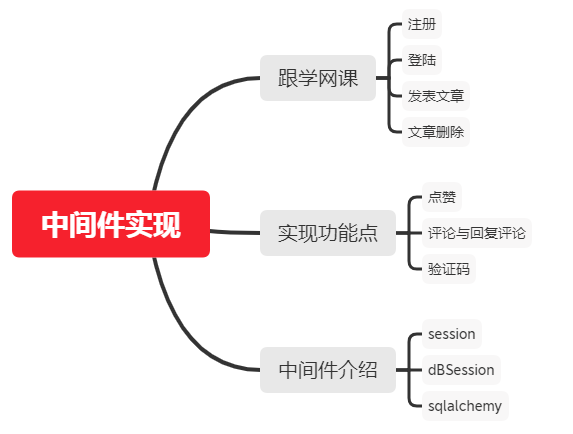
系统基于Flask框架，目前实现的功能主要包括注册、登陆、发表文章、文章删除、点赞、评论与回复评论、验证码。

模块：公众号模块和普通用户模块，用户模块只能点赞、评论、浏览，公众号模块可以有全部功能。

# ****2.所做工作****

因为自己跨专业，基础太弱，不太会开发，需要一些系统性学习。所以主要展开了以下的工作：跟着网课敲代码，实现一些基本的功能；自己尝试实现一些功能点；根据课上所学，研究该项目用到了什么中间件，展开介绍，如下图所示：

**注：**“跟学网课”的部分是网课现成的；“实现功能点”这部分是自己写的几个函数；“中间件介绍”是结合课程所讲进行的分析，也是该报告的重点。



# ****3.如何运行****

先安装相关的包，具体在requirements.txt中，使用Python3.6版本运行。本地运行，打开系统后，单击此处：

C:\Users\ADMINI~1\AppData\Local\Temp\1591023472(1).png

可以在http://127.0.0.1:5000/查看系统。

# ****4.系统功能实现****

**4.1注册**，注册分为普通用户和公众号注册，代码如下：

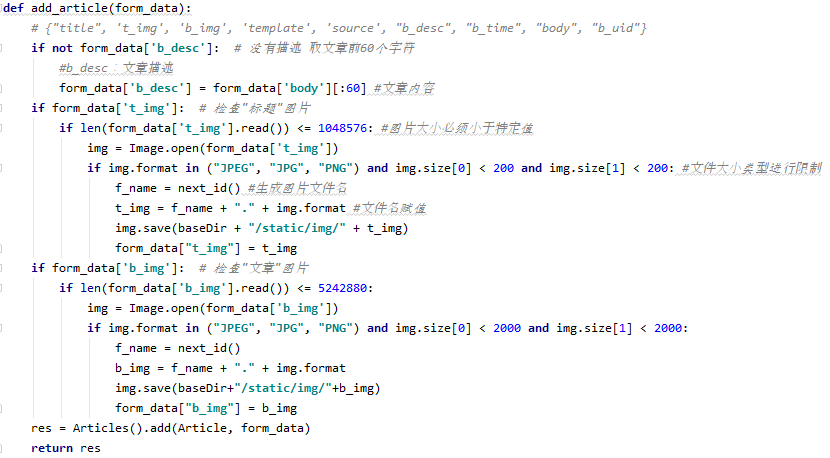




基本的注释已经写在代码中，再提一句的是头像图片不能大于1MB，一个邮箱只能注册一个账号。

**4.2登陆**，对应代码在index.py中247-259行，比较简单，不去赘述。

**4.3发表文章**，特殊说明的是，如果没有摘要，会自动提取文章前60个字符作为摘要。



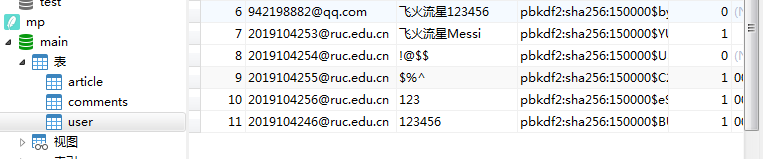
**4.4文章删除**，这里要先核实是不是作者本人，再就是likequery.is\_del 是进行逻辑删除，避免很多麻烦。



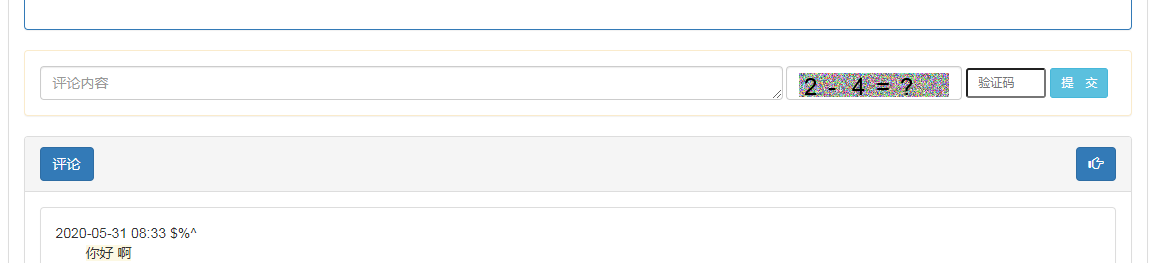
**4.5点赞**，对应代码在index.py中102-115行，比较简单，不去赘述。

**4.6评论与回复评论**，评论要核实是否登陆以及验证码是否正确，回复评论是公众号用户特有的功能，所以要核实是不是作者。代码在index.py中137-184行，比较长，就不截图了。

**4.7数据库**，数据存储在本地的Mysql数据库中，截图如下：



**4.8验证码与评论**，代码在index.py中344-393行。需要填写验证码才能进行评论，确保评论是人进行的操作。把评论和验证码结合了起来，但是这样每一次评论会很麻烦，同学建议直接在登陆的时候填写验证码就可以。



验证码实现主要包括以下步骤：选择颜色，数学运算，创建Draw对象，填充元素，输出文字。

# ****5.中间件——session****

**5.1 中间件定义：**

中间件是一种独立的系统软件服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。

中间件是介于应用系统和[系统软件](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E8%BD%AF%E4%BB%B6/215962" \t "_blank)之间的一类软件，它使用系统软件所提供的基础服务（功能），衔接网络上应用系统的各个部分或不同的应用，能够达到资源共享、功能共享的目的。

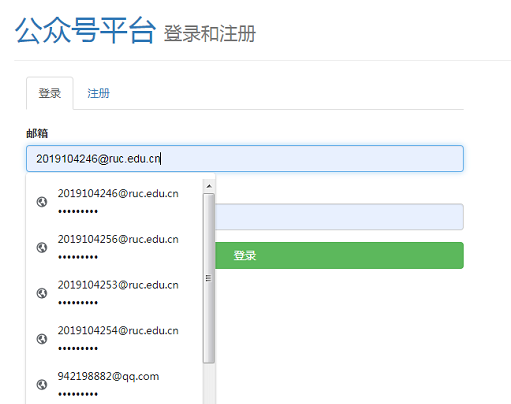
中间件=平台+通信。

**5.2 Flask-session：**

session 可以看做是在不同的请求之间保存数据的方法，因为 HTTP 是无状态的协议，但是在业务应用上我们希望知道不同请求是否是同一个人发起的。

**5.3 原理：**

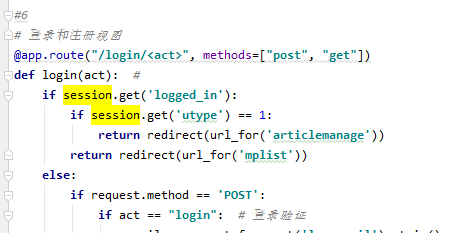
当客户端进行第一次请求时，客户端的HTTP request（cookie为空）到服务端，服务端创建session，视图函数根据form表单填写session，请求结束时，session内容填写入response的cookie中并返回给客户端，客户端的cookie中便保存了用户的数据。



当同一客户端再次请求时， 客户端的HTTP request中cookie已经携带数据，视图函数根据cookie中值做相应操作（如已经携带用户名和密码就可以直接登陆）。

**5.4 代码实现：**

该系统通过用户访问 POST /login 来实现登陆，如下图所示：



**5.5 存储与删除：**

用户登陆的信息在本地的存储如下：



session中的值可以通过以下三种方式删除：

-`session.pop(key)`。  
 -`del session[key]`。  
 -`session.clear()`：删除session中所有的值。

# ****6.中间件——sqlalchemy-ORM&dBSession****

**6.1定义：**

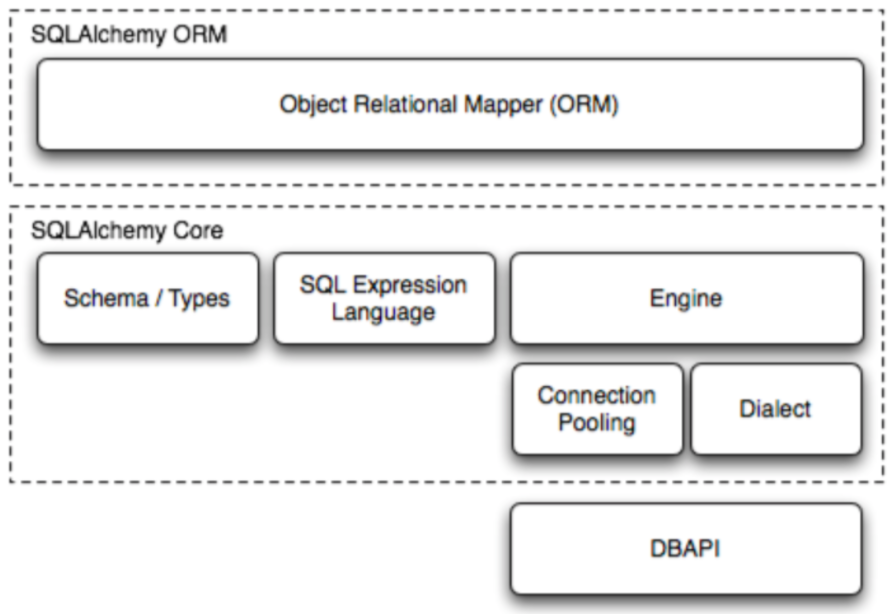
SQLAlchemy 是 Python 中一个通过 ORM 操作数据库的框架。他的作用就是能将数据库里面的实体类数据表映射成python里面的实体类，与Django的ORM框架各有优缺点。

SQLAlchemy对象关系映射器提供了一种方法，用于将用户定义的Python类与数据库表相关联，并将这些类（对象）的实例与其对应表中的行相关联。它包括一个透明地同步对象及其相关行之间状态的所有变化的系统，称为工作单元，以及根据用户定义的类及其定义的彼此之间的关系表达数据库查询的系统。

ORM：一般指对象关系映射。

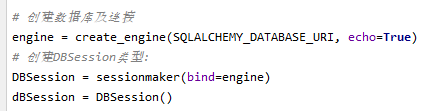
**6.2架构：**

架构图如下图所示，其中Schema/Types 定义了类到表之间的映射框架（规则），SQL Expression Language 封装好 SQL语句，Engine是操作者，Connection Pooling 是连接池，Dialect 根据用户的配置，调用不同的数据库 API(Oracle, postgresql, Mysql) 并执行对应的 SQL语句。

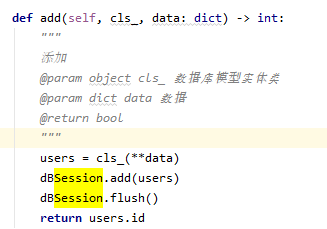


**6.3代码实现：**

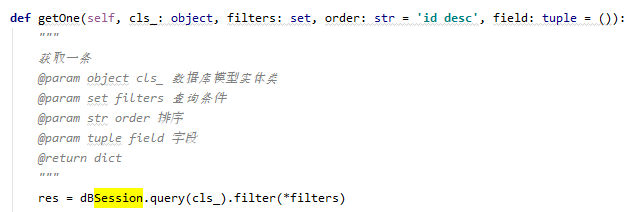
安装，略。连接数据库，初始化，如下图：



添加记录操作：



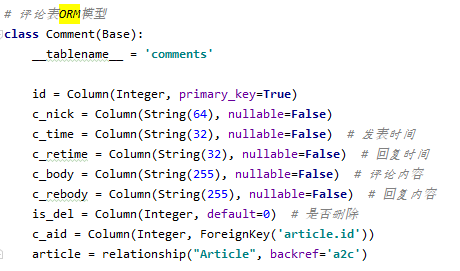
查询操作：



删除、更新操作类似，在此不赘述。

创建表，主要创建的表有三张：用户表，文章表，评论表。以评论表为例，建表需要用到的属性，以及效果如下图所示：

创建表：



数据库中存储评论的内容：

