



**QG工作室项目报告**

学 院 计算机学院

专 业 软件工程

班 级 2017级 2 班

组 别 网络组

姓 名 陈子锋

学 号 3117004830

2018 年 5月

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc515492362)

[1.1. 项目背景 1](#_Toc515492363)

[1.2. 项目用途 1](#_Toc515492364)

[1.3. 项目开发 1](#_Toc515492365)

[1.3.1. 项目开发环境 1](#_Toc515492366)

[1.3.2. 项目所需配置 2](#_Toc515492367)

[1.3.3. JDBC的配置 2](#_Toc515492368)

[1.3.4. 项目使用方法 3](#_Toc515492369)

[1.3.5. 项目开发周期 3](#_Toc515492370)

[2. 设计思路 4](#_Toc515492371)

[2.1. 项目架构 4](#_Toc515492372)

[2.2. 需求分析 5](#_Toc515492373)

[2.3. 数据库设计文档 7](#_Toc515492374)

[2.3.1. 数据库命名规则 7](#_Toc515492375)

[2.3.2. 逻辑设计 7](#_Toc515492376)

[2.3.3. 物理逻辑设计 9](#_Toc515492377)

[3. 详细功能介绍 11](#_Toc515492378)

[3.1. 登陆注册功能模块 11](#_Toc515492379)

[3.2. 16](#_Toc515492380)

[3.3. 16](#_Toc515492381)

[3.4. 16](#_Toc515492382)

[3.5. 16](#_Toc515492383)

[4. 项目亮点 16](#_Toc515492384)

[4.1. 登陆与注册使用同一页面 16](#_Toc515492385)

[4.2. 使用Json格式传输数据 16](#_Toc515492386)

[4.3. 用JS实现输入验证 16](#_Toc515492387)

[4.4. 响应式布局 16](#_Toc515492388)

[5. 项目难点 17](#_Toc515492389)

[5.1. 使用jQuery实现AJax 17](#_Toc515492390)

[6. 体会总结 17](#_Toc515492391)

[6.1. 遇到问题时的解决思路 17](#_Toc515492392)

[6.2 收获与成长 17](#_Toc515492393)

# 项目简介

## 项目背景

本项目是QG工作室2018年网络组招新考核大设计，项目名称为“个人博客系统”。本文档详细说明了项目的详细功能、项目需求分析、功能使用方法，亮点难点等等。

## 项目用途

本项目用于连接互联网上的多个用户，进行多人互动，达到信息交流互动等功能。用户可通过本项目创建属于自己的个人博客，成为博主，博主可在个人博客中发表自己的博客，删除自己的博客，评论他人的博客，回复他人的评论，与他人分享，交流等等。游客或者来访者可以通过本项目浏览博主分享的信息，并可以评论博主，博主可以回复评论者，进而实现多人交互。

个人博客系统是针对希望个性化使用博客的用户的需求而设计，是可以完成个人博客用户登入、发表、浏览、修改文章以及图片视频、留言、评论甚至个性化设计博客网站页面、排版布局等功能的系统。

## 项目开发

### 项目开发环境

本项目是基于Java、HTML、CSS、JavaScript等语言编写的Java web项目，存储数据采用了MySQL数据库，可以用主流编译器进行开发，包括windows下的文本编辑器，服务器使用了Tomcat搭建pc本地服务器。

详细开发环境

* 数据库：MySQL 8.0
* 编译器：Eclipse jee Oxygen, Visual Studio Code等
* 服务器：Tomcat 8.0.5.1
* 浏览器：Chrome，火狐浏览器、IE浏览器等。

备注：建议使用Chrome浏览器，兼容性：Chrome > 火狐 > IE

### 项目所需配置

本项目运用了MySQL，Json等技术，所以，想要完美运行本项目，请下载以下Jar包（源程序文件中附带）。

JDBC，运行数据库驱动

* mysql-connector-java-8.0.1.1.jar

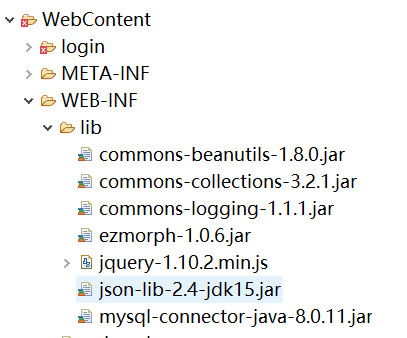
JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程。

JSON-lib及所需依赖包

* json-lib-2.4-jdk15.jar
* commons-beanutils-1.8.0.jar
* commons-collections-3.2.1.jar
* commons-logging-1.1.1.jar
* commons-lang-2.6.jar
* ezmorph-1.0.6.jar

JSON-lib可以将Java对象转成json格式的字符串，也可以将Java对象转换成xml格式的文档，同样可以将json字符串转换成Java对象或是将xml字符串转换成Java对象。

附：添加jar包方法：将下载好的jar包拖入到你的项目中的WEB-CONTENT下的WEB-INF文件夹下的lib文件夹中。



### JDBC的配置

本项目采用了较新版本的JDBC，所以，在连接数据库时，可能会出现一些未知错误，所以请跟随以下建议。

1．高版本的MySQL需要指明是否进行SSL连接。

* 在mysql连接字符串url中加入ssl=true或者false即可，如下所示。

例：url =jdbc:mysql://127.0.0.1:3306/blog?useSSL=false

2．设置时区

* 在mysql连接字符串url中加入serverTimezone=xxx，如下所示。

例:url=jdbc:mysql://127.0.0.1:3306/blog?serverTimezone=Hongkong

附：

本项目JDBC的URL（供参考）：

url=jdbc:mysql://127.0.0.1:3306/blog?useSSL=false&serverTimezone=Hongkong

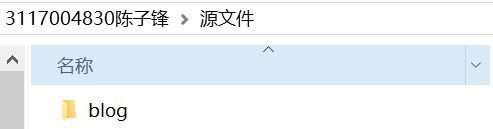
本项目数据库的配置信息：

* 数据库名：blog
* 用户名：root
* 密码：123

### 项目使用方法

* 打开eclipse，将源文件文件夹中的blog文件夹拖到项目树里、

源文件文件夹



Eclipse中的项目树



* 运行源文件文件夹里的数据库文件，方法多种不详细介绍



* 运行Tomcat服务器，运行本项目

### 项目开发周期

2018年5月12日9:00至2018年5月24日9：00，共12天。

# 设计思路

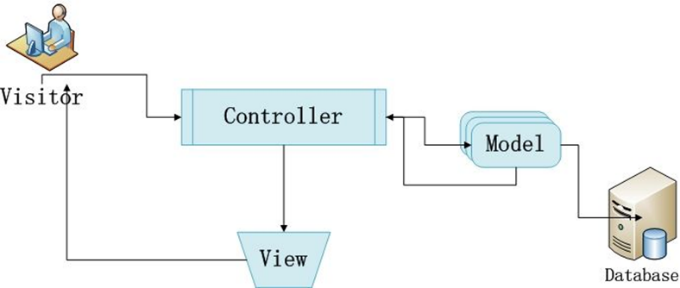
## 项目架构

本项目采用浏览器/服务器（B/S）架构，本项目是基于MVC（Model View Controller 模型-视图-控制器）开发模式下的RESTful风格的java web项目，MVC模式，是一种前后端分离的开发模式，Model层，是数据层，存放着用户的数据，数据的获取、存储、数据状态变化都是Model层的任务。View层，也就是视图层，是软件与用户直接接触交互的部分，是项目的前端部分，本项目的View层使用前端三剑客来编写(HTML+CSS+JavaScript)。Controller层，是项目的后台部分，使用java来编写，主要运用了java的Servlet技术，有接受前端传来的数据、处理前端的请求、转发请求、操作Model层等功能，是连接View层与Model层的桥梁。

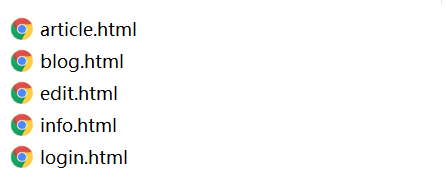
关于RESTful，REST是英文representational state transfer(表象性状态转变)或者表述性状态转移;Rest是web服务的一种架构风格;使用HTTP,URI,XML,JSON,HTML等广泛流行的标准和协议;轻量级,跨平台,跨语言的架构设计;它是一种设计风格,不是一种标准,是一种思想，它有以下特点：

* 网络上的所有事物都被抽象为资源
* 每个资源都有一个唯一的资源标识符
* 同一个资源具有多种表现形式(xml,json等)
* 对资源的各种操作不会改变资源标识符
* 所有的操作都是无状态的
* 符合REST原则的架构方式即可称为RESTful

MVC结构业务逻辑：

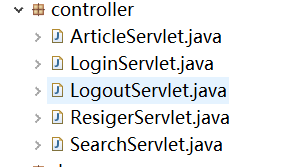


* View层：采用前端三剑客（HTML+CSS+JavaScript）编写，根据需求，共设计了五个页面，分别是登陆注册页面、项目主页、用户详细信息页面、编辑文章页面和文章的详细页面。

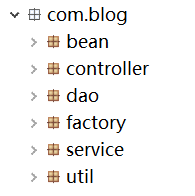


* Controller层：使用Java的Servlet技术，基本上每一个功能都具有一个以上的Servlet,该层负责操作Model层，处理转发等业务。

图为项目中的一些Servlet



* Model层：Model层负责数据的传输，本项目对Model层分离较为细，有存在实体类的JavaBean层，连接数据库的Util工具类，Dao层，DAOImpl层，处理逻辑的service层，factory层等。



## 需求分析

基于个人博客的特点，本系统实现个人博客的主要功能，包括博客系统的登录/退出/注册，博主发表文章、修改文章、删除文章等，游客发表评论、分页浏览文章等。博客系统主要区分了两类用户，分别为博主和普通游客。网站登录仅限于博主，博主可以登陆写下自己的文章，登陆即拥有网站管理的权限，未登录时均以游客身份访问。游客主要的功能是分页、分类、分标签阅读博主文章，发表评论和留言。博主既可以评论文章、又可以回复留言、删除留言等，以上是个人博客的系统需求分析的基本实现目标，由于个人博客的个性化特点，还应实现修改博主个人信息、更换博主背景图等个性化功能。

1. 按模块分析
2. 文章模块
   * + - 增加文章
       - 删除文章
       - 查看文章
       - 搜索文章
       - 更新文章
       - 保存草稿
3. 评论模块

* 评论文章
* 回复评论
* 删除评论

1. 个人信息模块

* 查看个人信息
* 添加个人信息
* 修改个人信息

1. 登陆注册模块

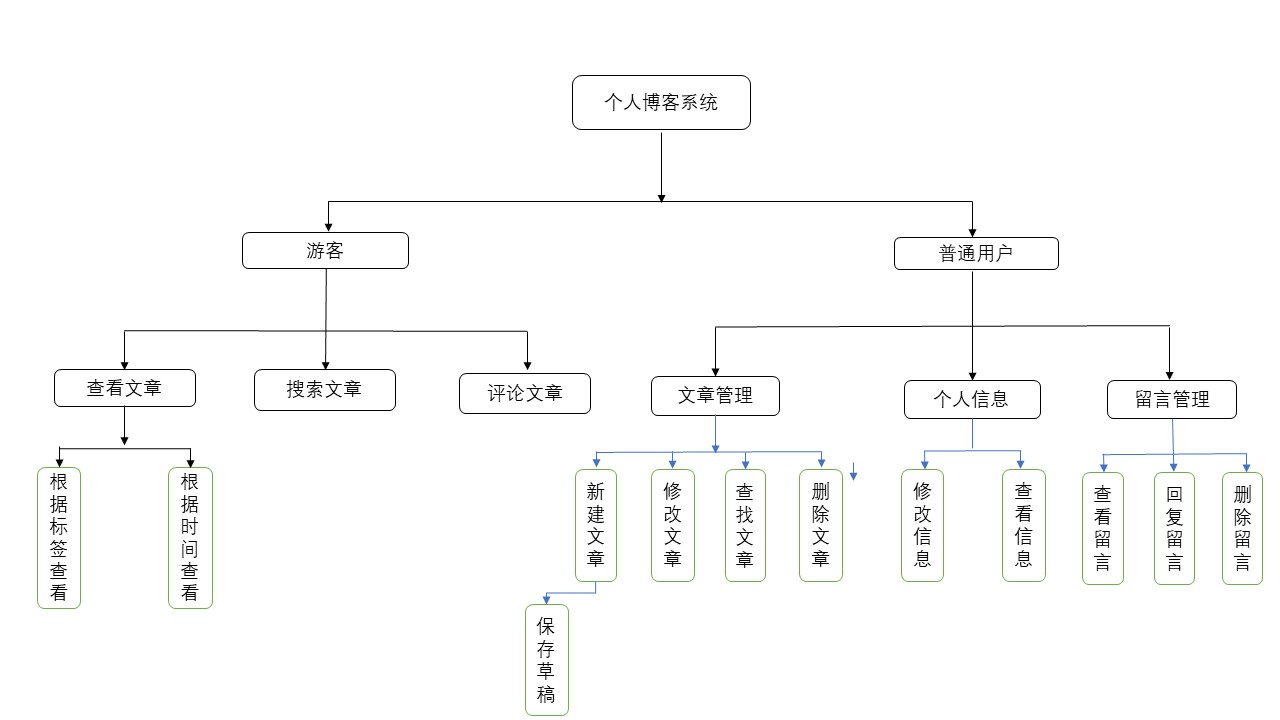
* 用户登陆
* 游客登陆
* 注册新用户

1. 按角色分析有普通用户和游客两种用户。
2. 普通用户

* 拥有用户模块所有功能
* 拥有评论模块所有功能
* 拥有个人信息模块所有功能

1. 游客

* 查看文章
* 搜索文章
* 评论文章
* 查看普通用户信息

需求分析图：****

## 数据库设计文档

### 数据库命名规则

数据库名称：blog

表名：英文（表的用途）+下划线+英文

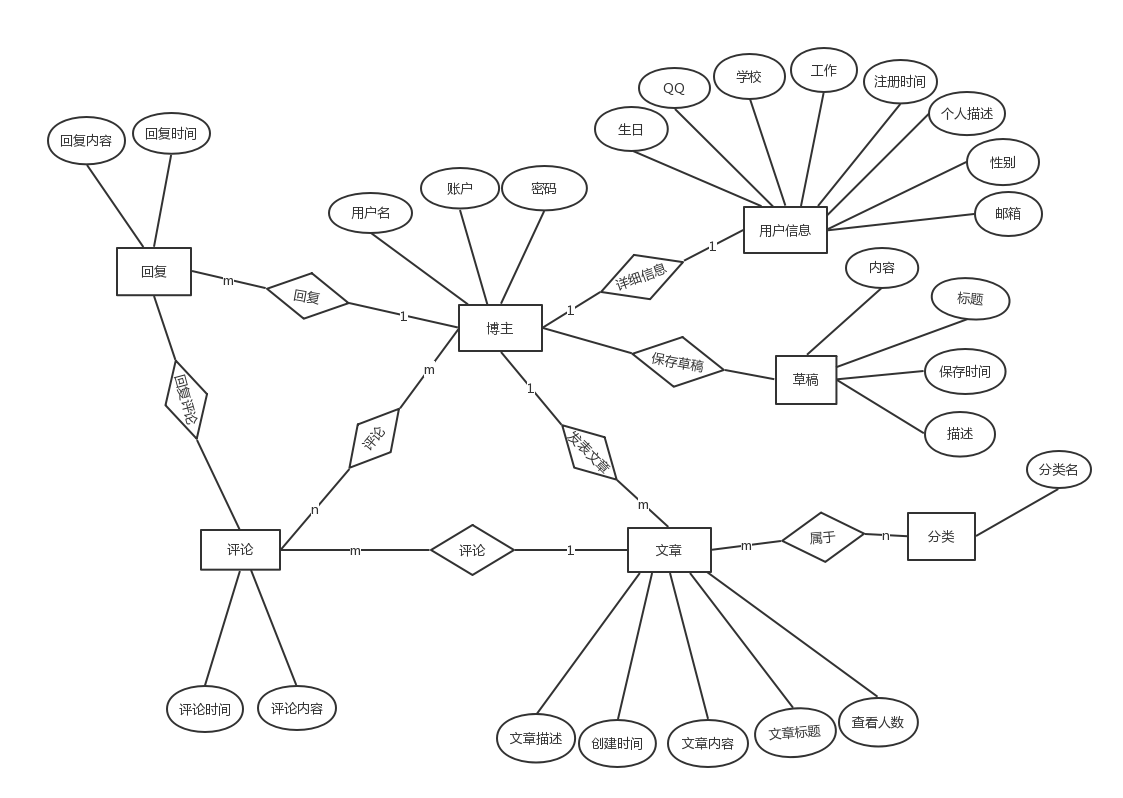
字段名：表名+下划线+相关属性的英文名

选择有意义的英文名来进行命名

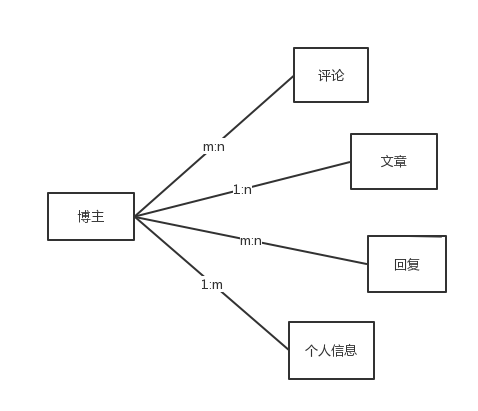
### 逻辑设计

本项目所要建立的是一个博客系统，根据需求文档，建立E-R 图如下所示，方形框为实体，椭形框为属性，菱形框为实体间的联系，连线上的数字为实体间的详细联系，如1对多，多对多等。一般地，实体对应表，实体的属性对应表中的字段，实体间的联系作为表的约束，逻辑设计中的大部分可以转化为物理设计中的表，但是他们不一定是一一对应的，也有增减。

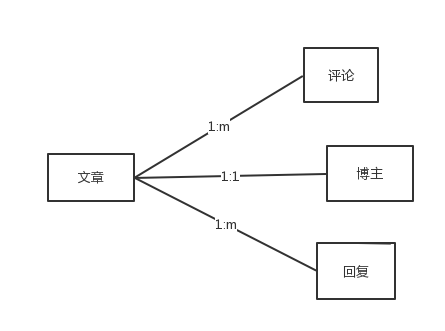
* 数据库E-R图：



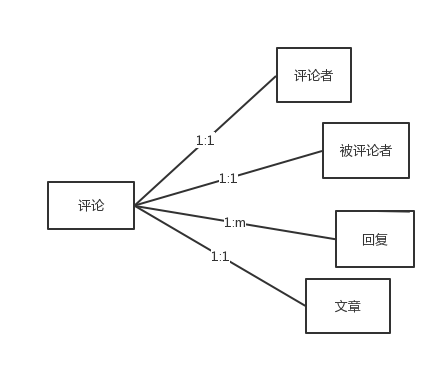
* 博主可以编辑个人信息，可以发布文章，可以发布评论，可以回复评论



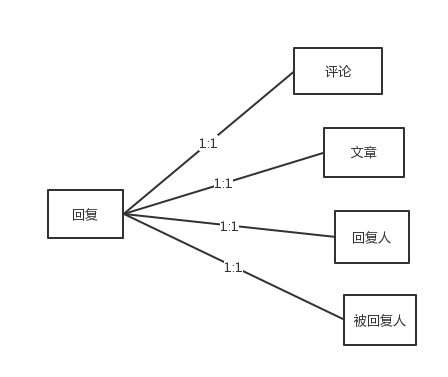
* 一篇文章有一个作者，多条评论，多条回复



* 一个评论对应一篇文章，一条评论可以有多条回复，一条评论一个评论者，一条评论有一个被评论者



* 一条回复有一个回复者，一个被回复者，一条回复对应一篇文章，一条回复对应一条评论



### 物理逻辑设计

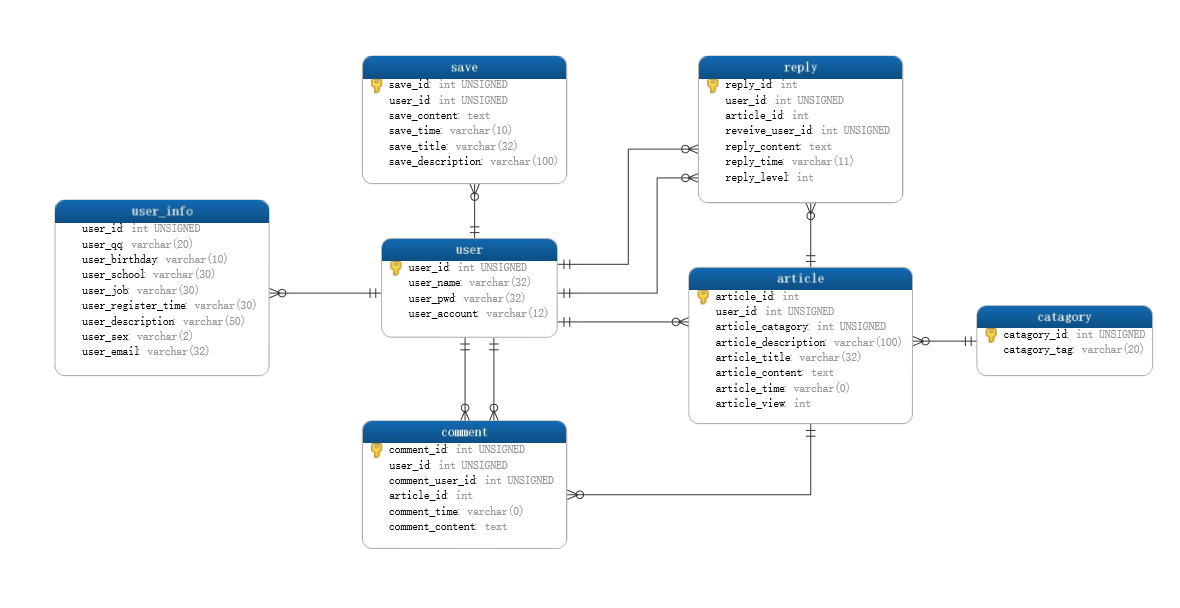
表的设计遵循数据库三大范式

* 第一范式（1NF）：数据表中的每一列（每个字段）必须是不可拆分的最小单元，也就是确保每一列的原子性
* 第二范式（2NF）：满足第一范式后，要求表中的所有列，都必须依赖于主键，而不能有任何一列与主键没有关系，也就是说一个表只描述一件事情
* 第三范式（3NF）:在第二范式基础上，任何非主属性不依赖于其它非主属性

表汇总：

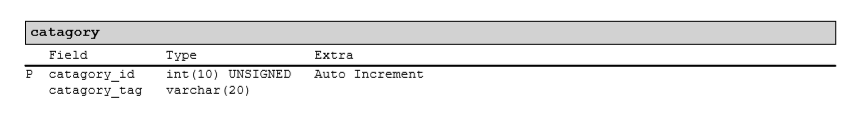
|  |  |
| --- | --- |
| 表名称 | 表功能 |
| user | 用于存储用户的必须数据，如密码、账号、用户名等 |
| user\_info | 用于存储用户的详细资料，如邮箱、个人描述、性别等 |
| article | 用于存储文章的详细信息，如标题、内容、创建时间等 |
| comment | 用于存储评论的详细信息，如评论内容、评论时间等 |
| reply | 用于存储回复的详细信息，如回复内容、回复时间等 |
| save | 用于存储草稿的详细信息，如草稿的内容、草稿的 |
| catagory | 用于存储文章分类的信息，如标签名等 |

表模型图：

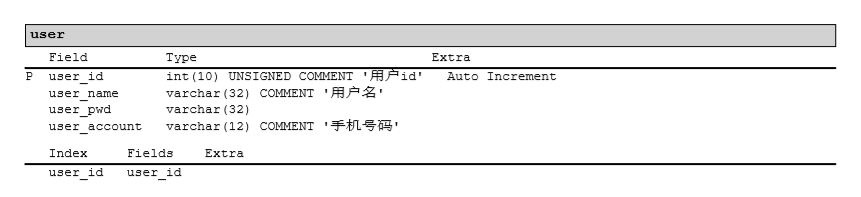


详细表：

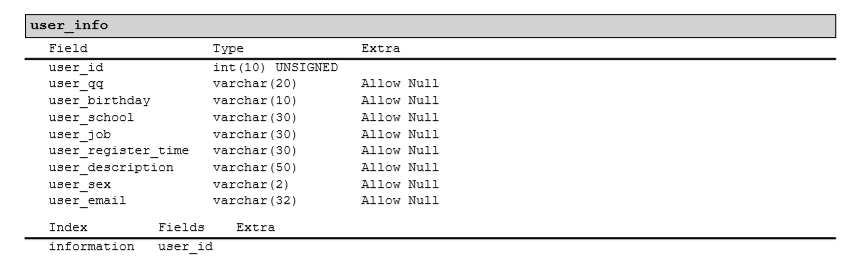
* catagroy表



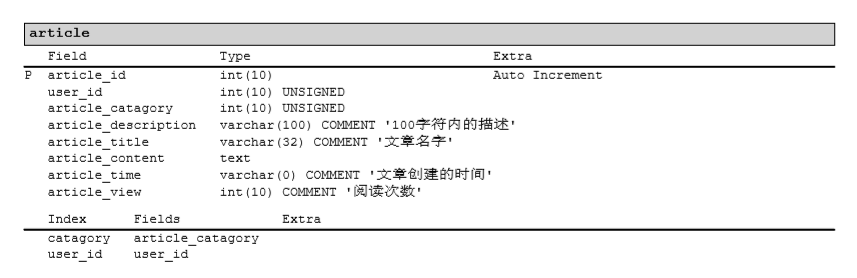
* user表



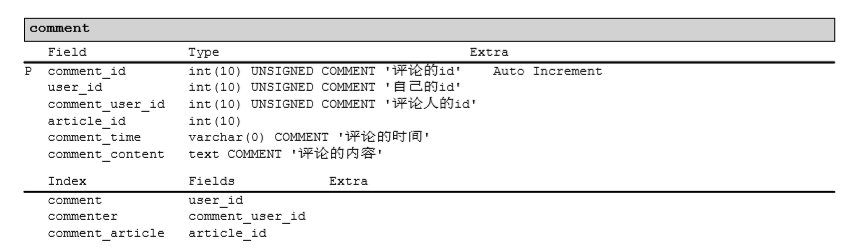
* user\_info表



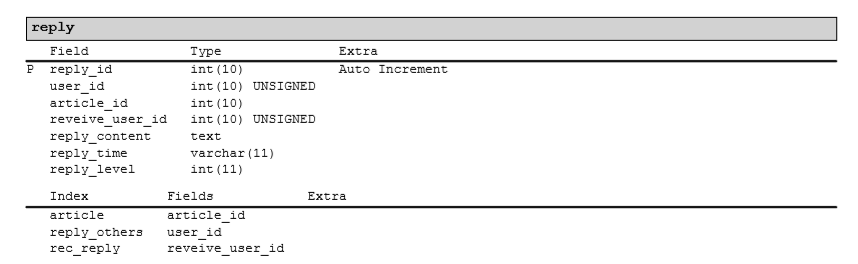
* article表



* comment表



* reply表

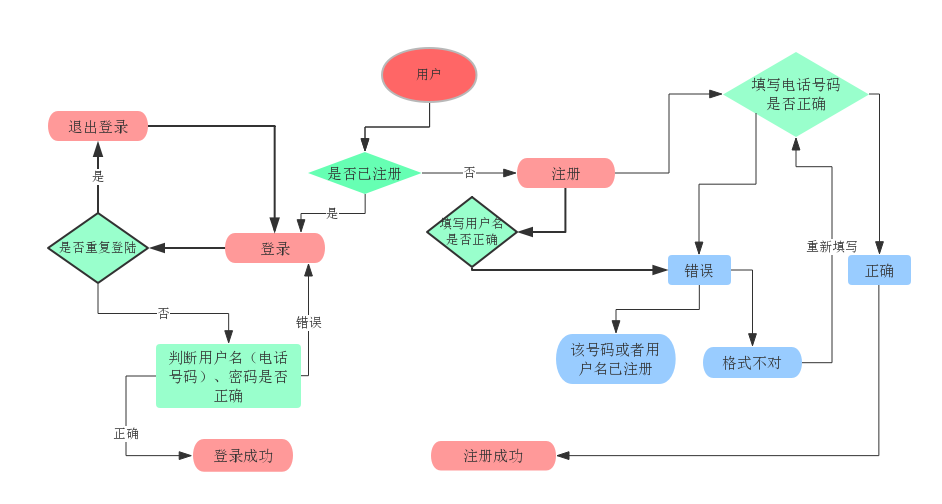


# 详细功能介绍

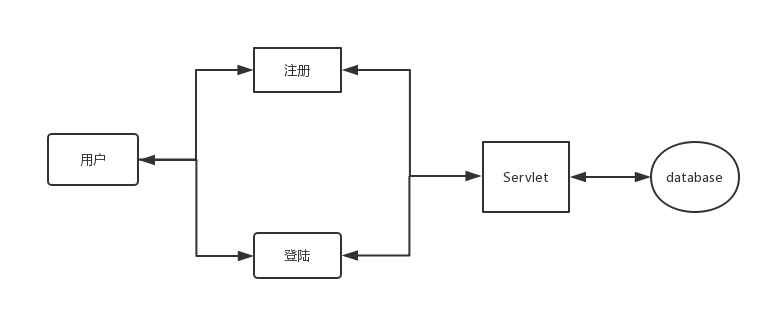
## 登陆注册功能模块

登陆注册功能模块包括：登陆功能、注册功能、和检测登陆状态功能。

图为登陆注册功能模块流程图：



图为程序MVC前后端交互图：

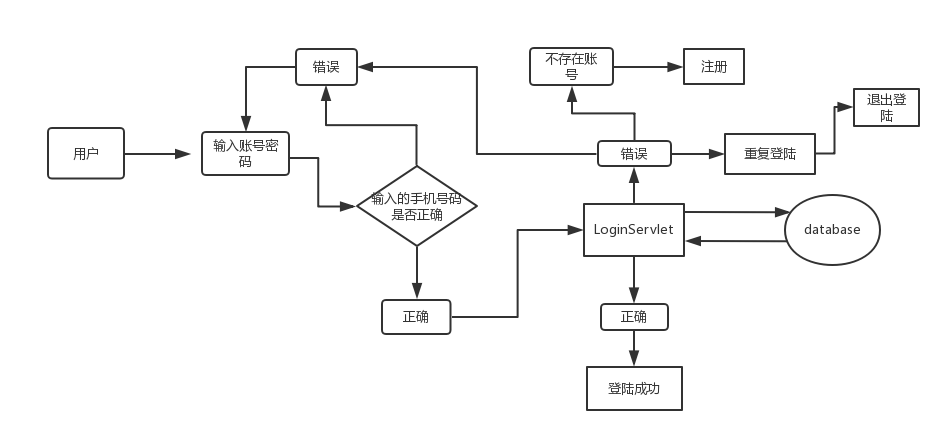


* 登陆功能：

登陆页面：



登陆功能流程图：

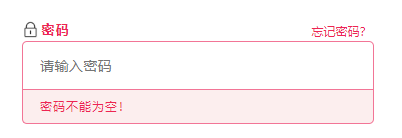


1. 登陆功能流程图讲解：

当用户输入账号和密码无误后，点击登陆按钮，项目会发送AJax请求，将用户输入的数据传到后台，LoginServlet接收到数据后，首先会在数据库中检索该手机号码，试图找到用这个手机号码注册的用户，如果找到了就会返回这个一个UserBean对象，其中对象的数据包括该用户的用户名、密码、用户唯一的标识user\_id，接下来，会用返回的用户的密码与用户输入的密码进行匹配，如果匹配成功，再去检查session中是否已经存有user\_id，如果不为空，证明已经是重复登陆，返回错误及错误原因到前端，如果为空，则根据该UserBean对象设置session的属性，返回成功给前端。如果找不到这个手机号码注册的用户，则返回错误及错误原因到前端，如果存在这个用户而密码匹配失败，则返回错误和错误原因到前端。前端根据返回的数据，分别显示不同的提示。登陆成功后，会自动跳转至项目主页。

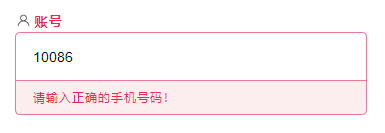
1. 用户登陆时需要输入手机号码和密码，但是当密码和手机号码都为空的时候，用户是不可以点击登陆按钮，此时会提示用户输入值不能为空。

图为密码为空时的提示：



1. 当用户输入手机号码时还会对用户的手机号码进行验证，如果输入的手机号码格式不正确，会出现错误提示。

图为手机号码格式不正确的提示：



1. 当用户输入正确后，登陆按钮会变为红色，并且变为可点击状态。

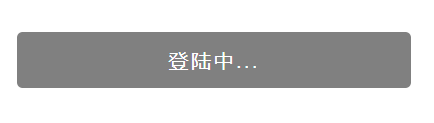
图为不可以登陆时登陆按钮的状态



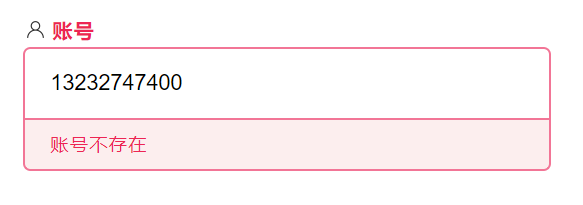
图为可以登陆时登陆按钮的状态：



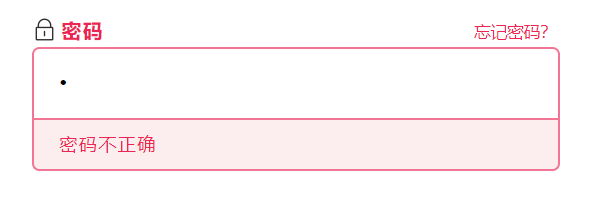
1. 当用户点击登陆后，登陆按钮会再次切换为不可以点击状态，并且“登陆”字样会变为“登陆中…”这样做的好处有两个，一个是可以让用户知道现在是处于什么一个状态，在网络状态不好的状况下，可以让客户更加耐心等待，不会不知所措而重复刷新页面或者关掉页面，二是可以防止用户重复提交表单，导致程序出错。这里主要运用了JQ中的AJax函数中的beforesend、success和complete函数。

图为等待后台响应时登陆按钮的状态：  


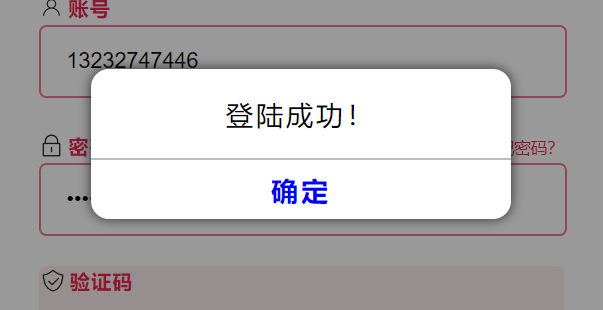
图为账号不存在的提示：

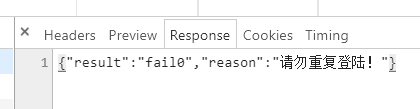


图为密码错误的提示：



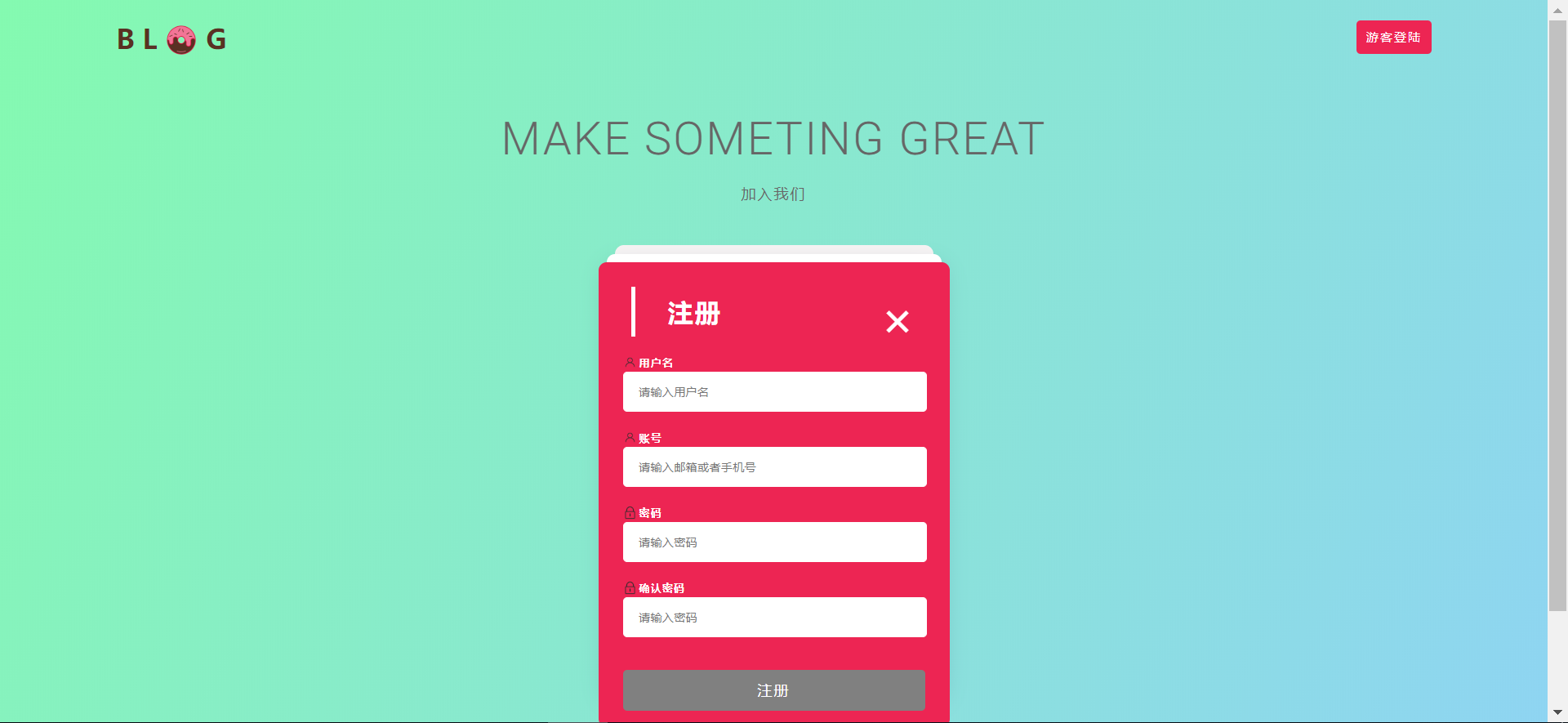
图为登陆成功的提示：

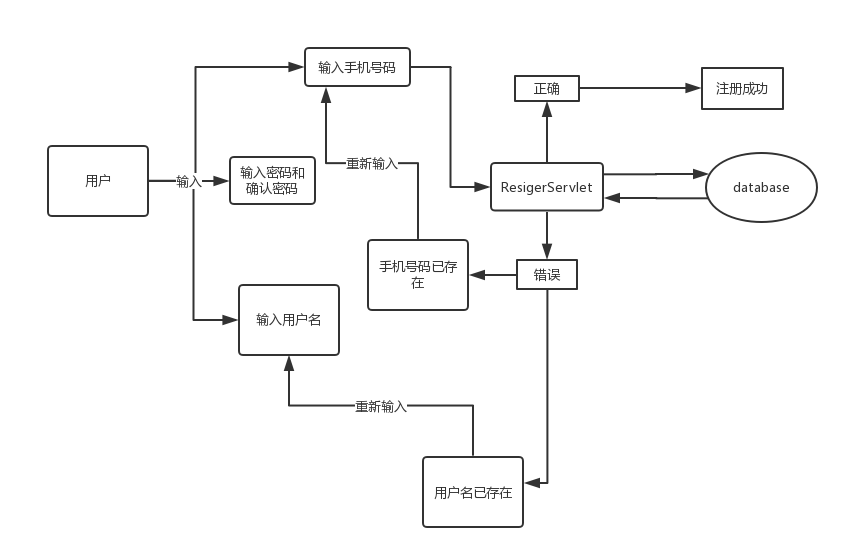


图为重复登陆时后台返回的Json数据：  


* 注册功能：

注册页面：



注册功能流程图：  


1. 注册功能流程讲解：

当用户填写完表单并没有报错提示的时候，点击注册按钮，此时会发送一个AJax请求到ResigerServlet，首先，ResigerServlet会根据用户输入的用户名在数据库进行检索，如果找到了就证明这个用户名已经被注册了，返回错误到前端。若用户名可用，则根据用户提交的手机号码再在数据库进行搜索，如果找到了证明这个手机号码已经被注册了，返回从错误到前端。如果手机号码可用，则把用户输入的信息写入数据库中，注册成功，返回成功到前端。

1. 注册功能和登陆功能一样，同样具有人性化的提示，当用户输入格式错误时，无法正常提交表单，当输入格式正确时，提示会消失。而且，输入的密码必须符合数字+字母+标点符号的组合格式，不能输入中文或者是空格等非法字符，而且长度要在8-20个字符之间。确认密码所输入的值必须要和上面输入的密码相匹配。

图为输入为空的提示：  

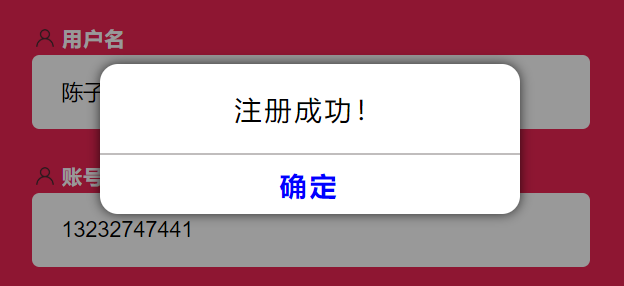

图为输入格式错误时的提示：



图为两次输入密码不匹配时的提示：  


图为用户名不可用时的提示：  


图为账户不可能时的提示：  


图为注册成功时的提示：  


## 

## 

## 

## 

# 项目亮点

### 登陆与注册使用同一页面

### 使用Json格式传输数据

### 用JS实现输入验证

### 响应式布局

# 项目难点

### 使用jQuery实现AJax

# 体会总结

不知不觉，QG训练营已经来到了尾声，在这里我将写下我在QG训练营期间发生的点点滴滴，总结我在这段时间内学到的东西，仔细挖掘自身的进步，认真反思自己的不足之处，

### 收获的知识

### 遇到问题时的解决思路

### 6.2 收获与成长