



**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](#_Toc515567942)

[1.1. 项目背景 1](#_Toc515567943)

[1.2. 项目用途 1](#_Toc515567944)

[1.3. 项目开发 1](#_Toc515567945)

[1.3.1. 项目开发环境 1](#_Toc515567946)

[1.3.2. 项目所需配置 2](#_Toc515567947)

[1.3.3. JDBC的配置 2](#_Toc515567948)

[1.3.4. 项目使用方法 3](#_Toc515567949)

[1.3.5. 项目开发周期 3](#_Toc515567950)

[2. 设计思路 4](#_Toc515567951)

[2.1. 项目架构 4](#_Toc515567952)

[2.2. 需求分析 5](#_Toc515567953)

[2.3. 数据库设计文档 7](#_Toc515567954)

[2.3.1. 数据库命名规则 7](#_Toc515567955)

[2.3.2. 逻辑设计 7](#_Toc515567956)

[2.3.3. 物理逻辑设计 9](#_Toc515567957)

[3. 详细功能介绍 11](#_Toc515567958)

[3.1. 登陆注册功能模块 11](#_Toc515567959)

[3.2. 17](#_Toc515567960)

[3.3. 17](#_Toc515567961)

[3.4. 17](#_Toc515567962)

[3.5. 17](#_Toc515567963)

[4. 项目亮点 17](#_Toc515567964)

[4.1. 登陆与注册表单 17](#_Toc515567965)

[4.2. 使用JSON格式传输数据 17](#_Toc515567966)

[4.3. 用JS实现输入验证 18](#_Toc515567967)

[4.4. 响应式布局 19](#_Toc515567968)

[5. 项目难点 21](#_Toc515567969)

[5.1. 使用jQuery实现Ajax 21](#_Toc515567970)

[6. 体会总结 23](#_Toc515567971)

[6.1. 技术上的收获 23](#_Toc515567972)

[6.2. 为什么选择前端？ 24](#_Toc515567973)

[6.3. 对于未来的发展 24](#_Toc515567974)

[6.4. 遇到问题时我是怎么样去解决的 25](#_Toc515567975)

[6.5. 衷心的感谢 25](#_Toc515567976)

# 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

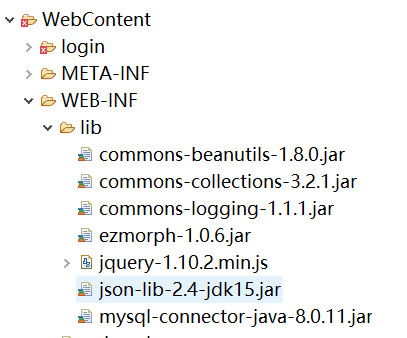
JDBC（Java DataBase Connectivity,java数据库连接）是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用Java语言编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程。

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

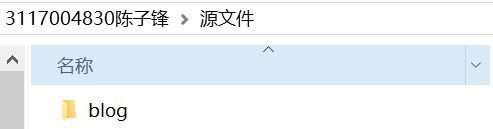
本项目数据库的配置信息：

* 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

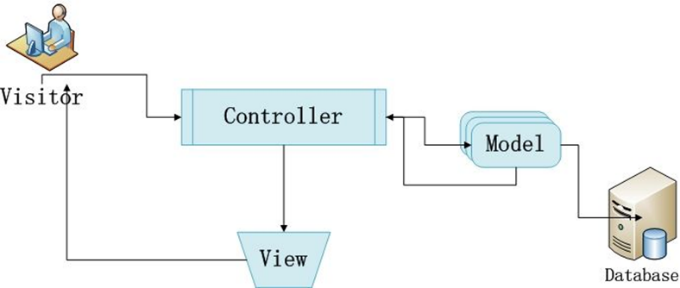
## 项目架构

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。

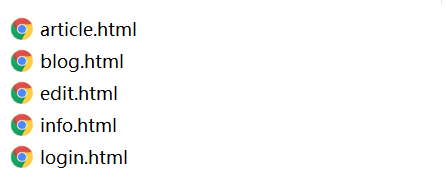
关于RESTful，REST是英文representational state transfer(表象性状态转变)或者表述性状态转移;Rest是web服务的一种架构风格;使用HTTP,URI,XML,JSON,HTML等广泛流行的标准和协议;轻量级,跨平台,跨语言的架构设计;它是一种设计风格,不是一种标准,是一种思想，它有以下特点：

* 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：

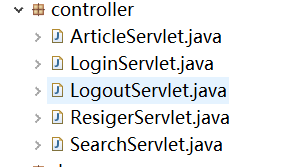


* View层：采用前端三剑客（HTML+CSS+JavaScript）编写，根据需求，共设计了五个页面，分别是登陆注册页面、项目主页、用户详细信息页面、编辑文章页面和文章的详细页面。

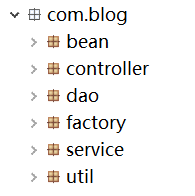


* Controller层：使用Java的Servlet技术，基本上每一个功能都具有一个以上的Servlet,该层负责操作Model层，处理转发等业务。

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

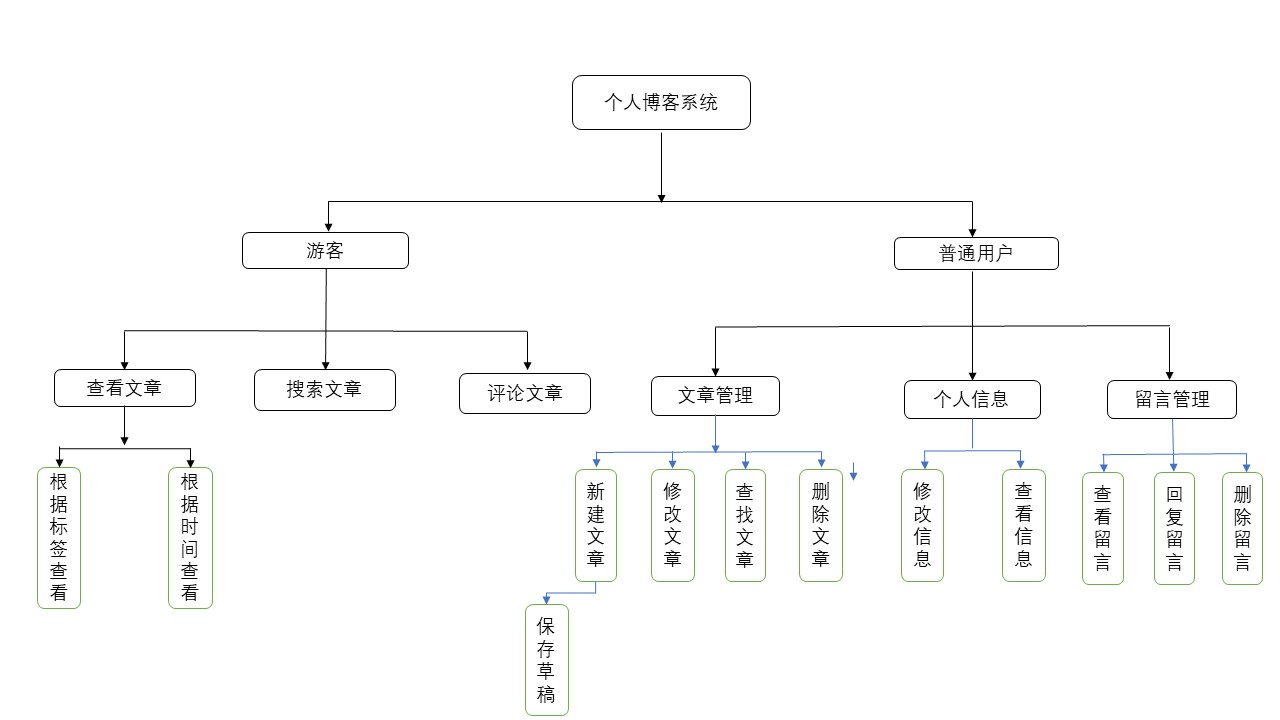
* 用户登陆
* 游客登陆
* 注册新用户

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

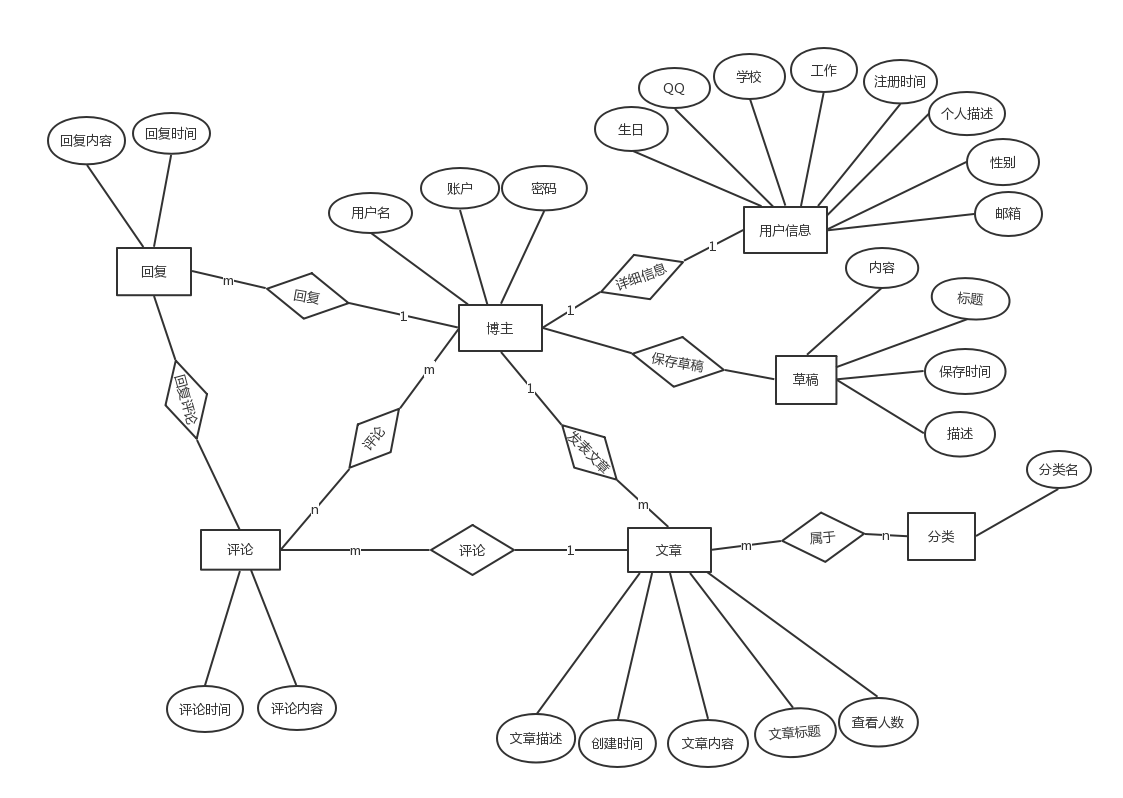
字段名：表名+下划线+相关属性的英文名

选择有意义的英文名来进行命名

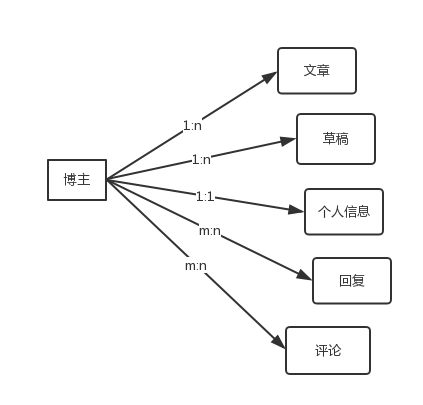
### 逻辑设计

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。

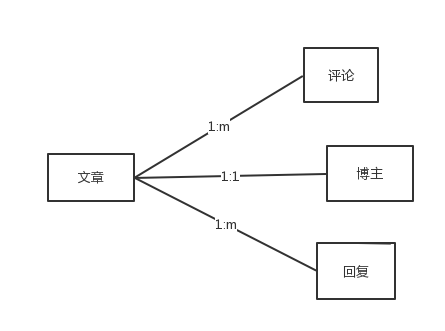
* 数据库E-R图：



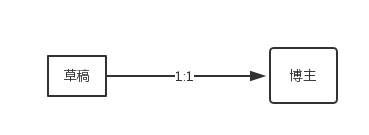
* 博主可以编辑个人信息，可以发布文章，可以发布评论，可以回复评论



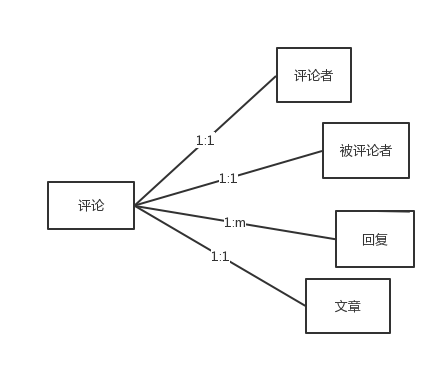
* 一篇文章有一个博主，多条评论，多条回复



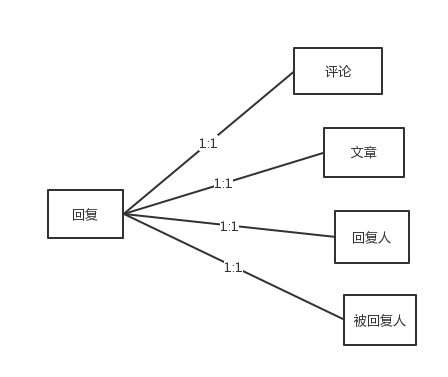
* 一篇草稿有一个博主



* 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

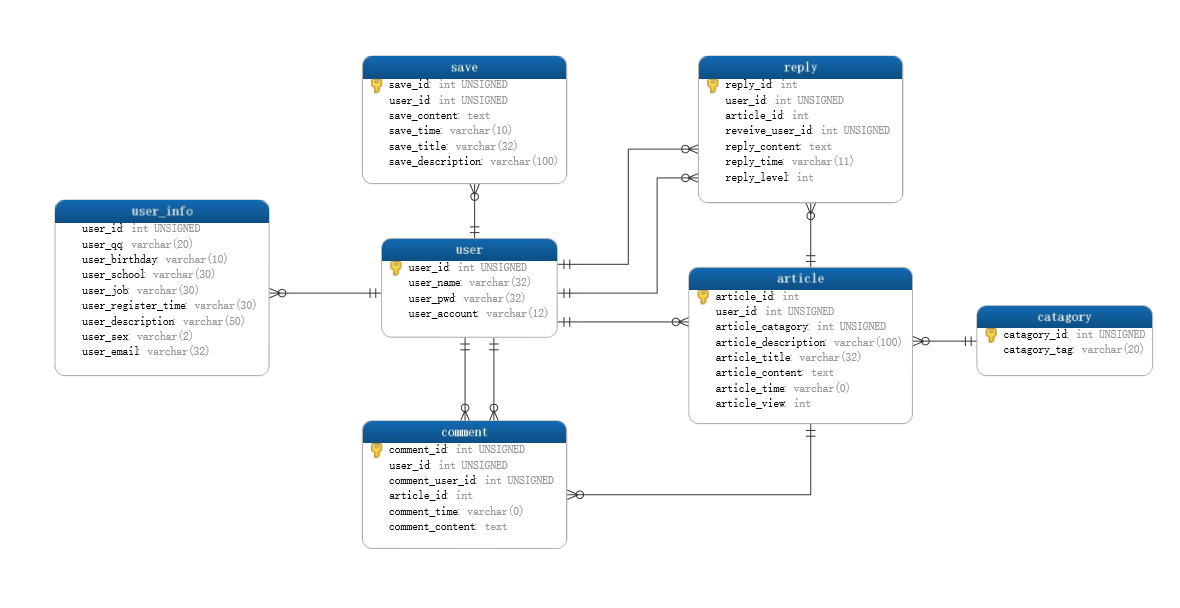
表的设计遵循数据库三大范式

* 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：

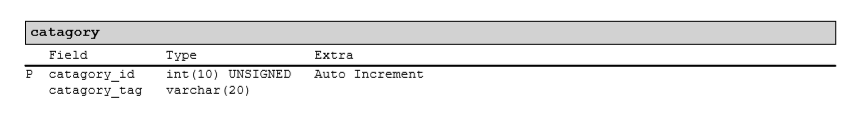
|  |  |
| --- | --- |
| 表名称 | 表功能 |
| user | 用于存储用户的必须数据，如密码、账号、用户名等 |
| user\_info | 用于存储用户的详细资料，如邮箱、个人描述、性别等 |
| article | 用于存储文章的详细信息，如标题、内容、创建时间等 |
| comment | 用于存储评论的详细信息，如评论内容、评论时间等 |
| reply | 用于存储回复的详细信息，如回复内容、回复时间等 |
| save | 用于存储草稿的详细信息，如草稿的内容、草稿的 |
| catagory | 用于存储文章分类的信息，如标签名等 |

表模型图：

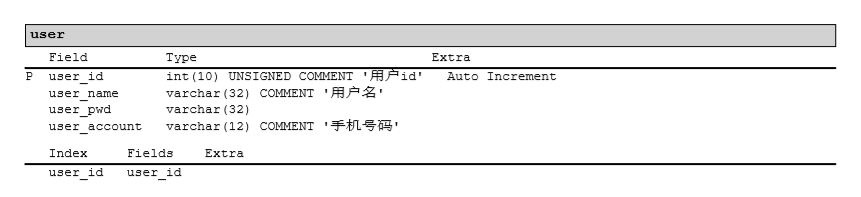


详细表：

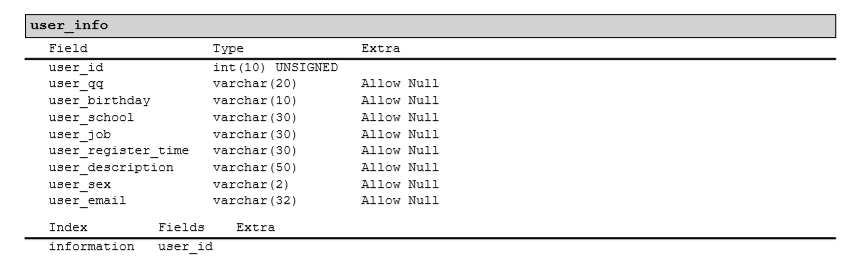
* catagroy表



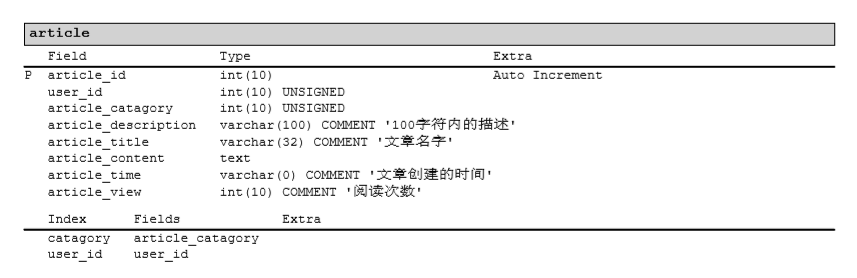
* user表



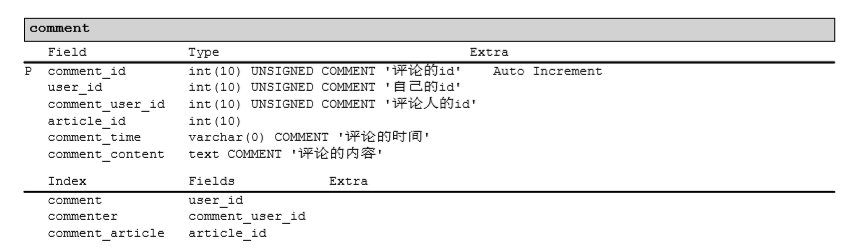
* user\_info表



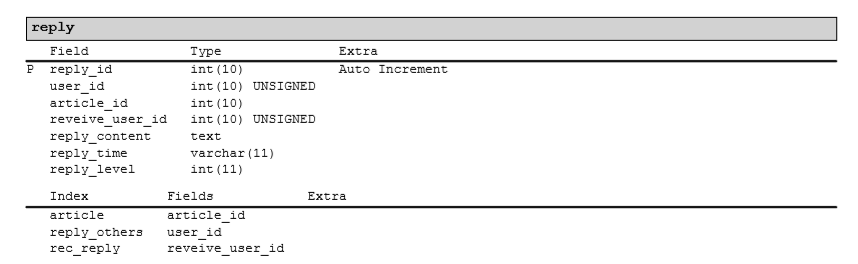
* article表



* comment表



* reply表

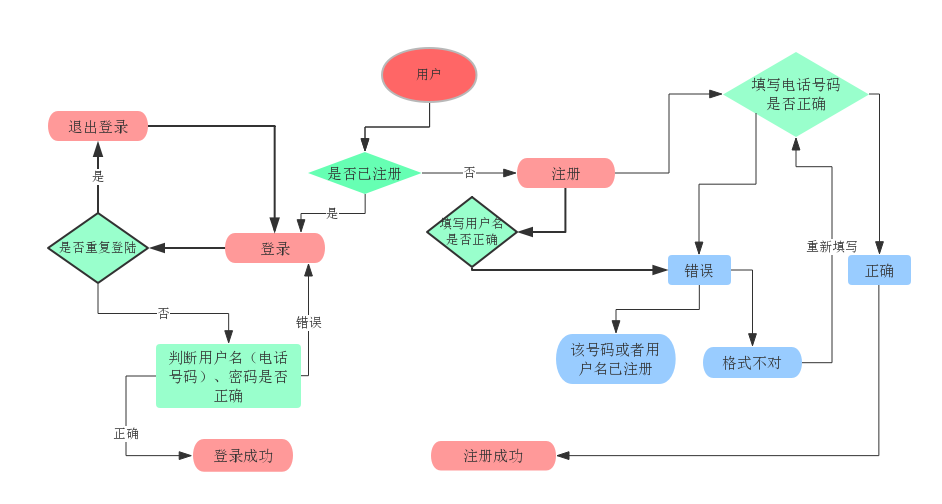


# 详细功能介绍

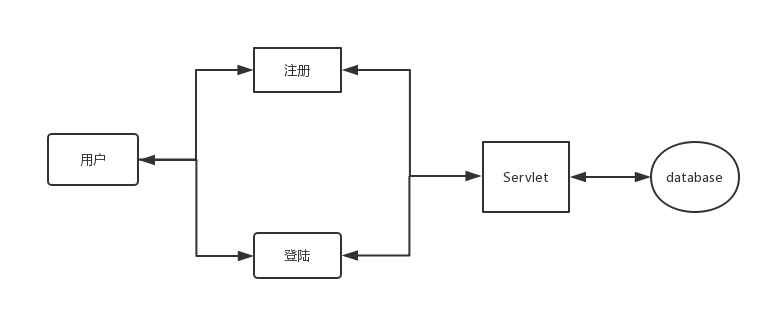
## 登陆注册功能模块

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：

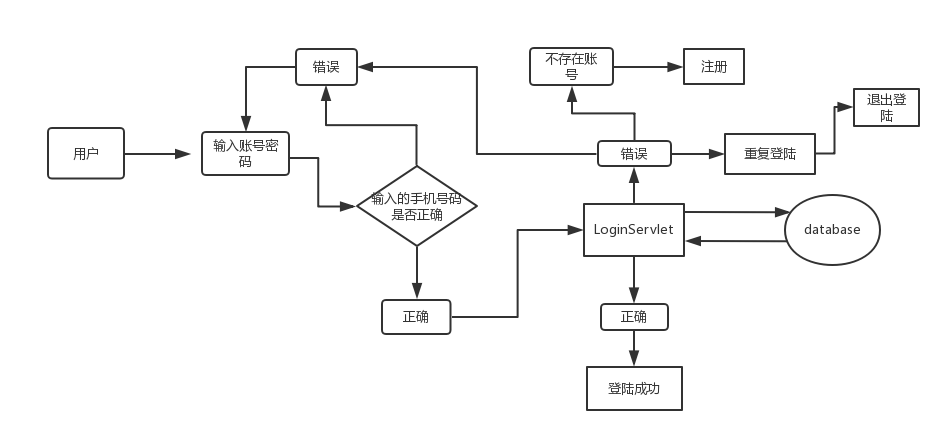


* 登陆功能：

登陆页面：



登陆功能流程图：

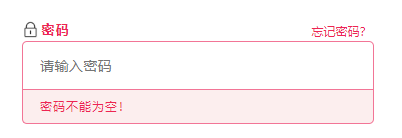


1. 登陆功能流程图讲解：

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。

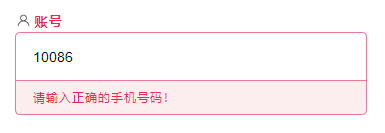
1. 用户登陆时需要输入手机号码和密码，但是当密码和手机号码都为空的时候，用户是不可以点击登陆按钮，此时会提示用户输入值不能为空。

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。

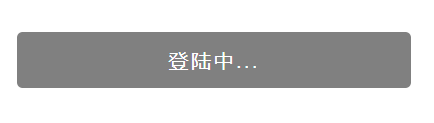
图为不可以登陆时登陆按钮的状态



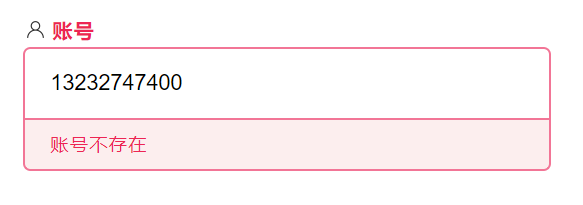
图为可以登陆时登陆按钮的状态：



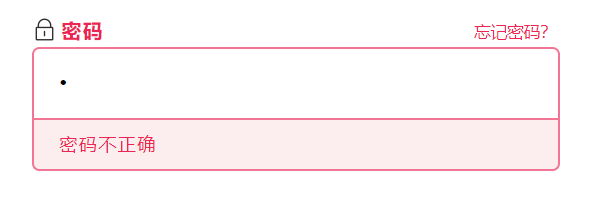
1. 当用户点击登陆后，登陆按钮会再次切换为不可以点击状态，并且“登陆”字样会变为“登陆中…”这样做的好处有两个，一个是可以让用户知道现在是处于什么一个状态，在网络状态不好的状况下，可以让客户更加耐心等待，不会不知所措而重复刷新页面或者关掉页面，二是可以防止用户重复提交表单，导致程序出错。这里主要运用了JQ中的AJax函数中的beforesend、success和complete函数。

图为等待后台响应时登陆按钮的状态：  


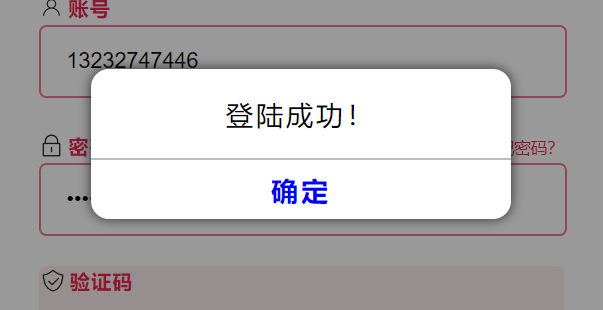
图为账号不存在的提示：

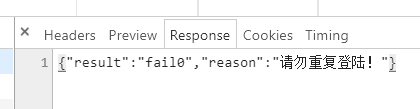


图为密码错误的提示：



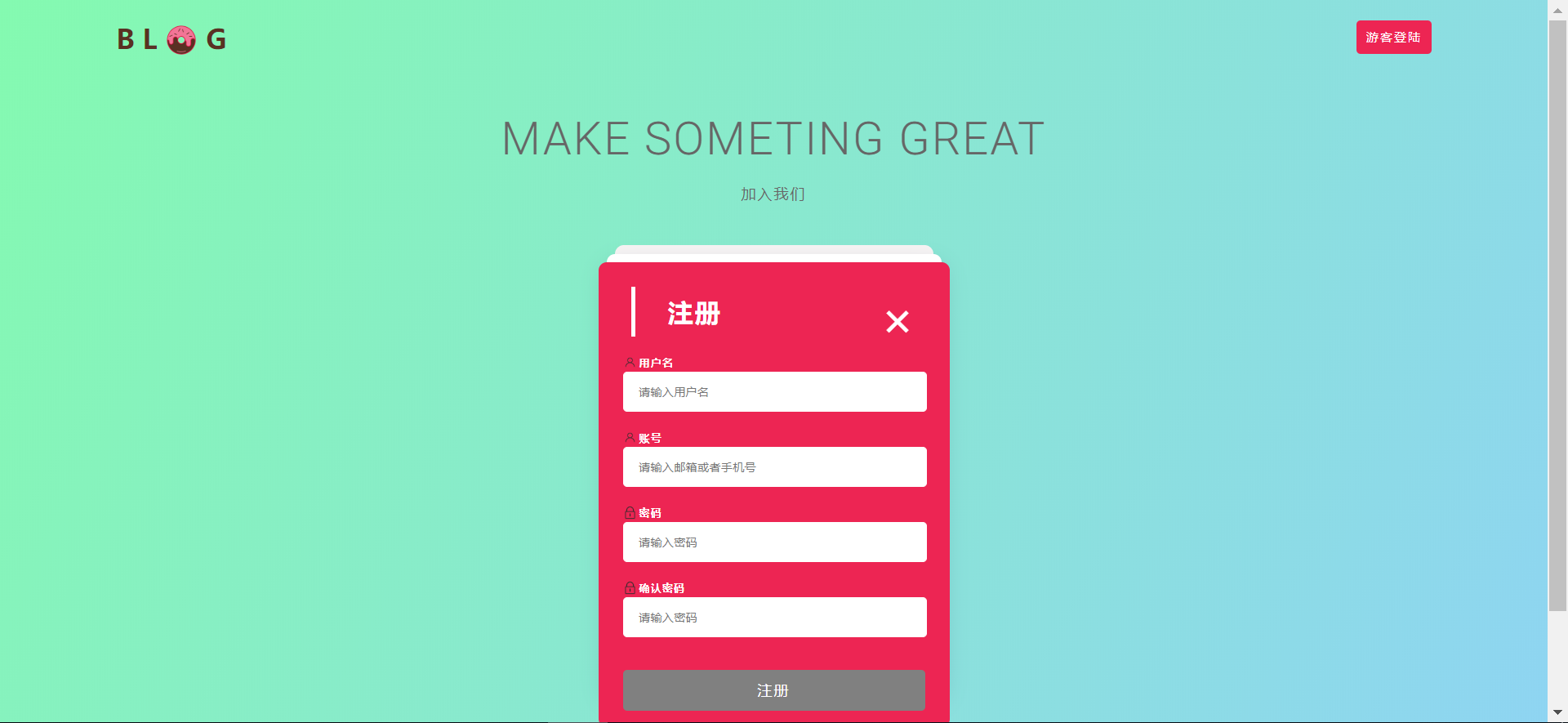
图为登陆成功的提示：

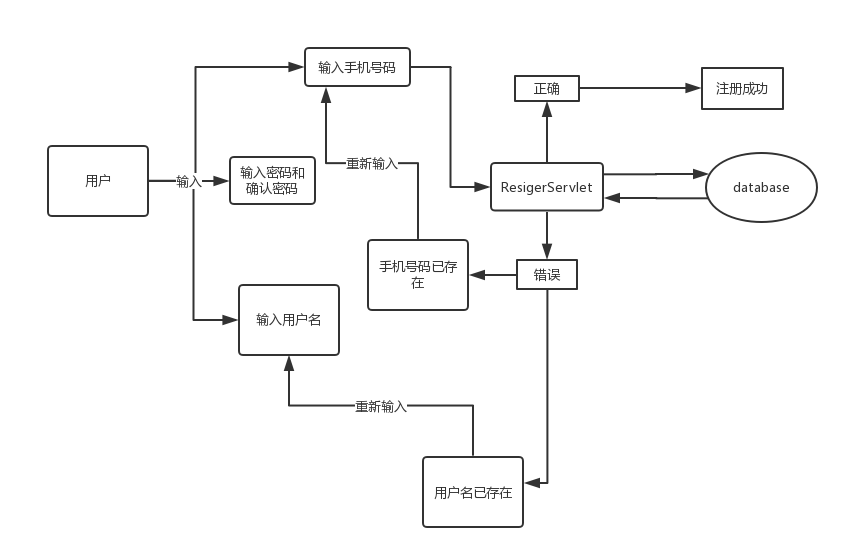


图为重复登陆时后台返回的Json数据：  


* 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：  

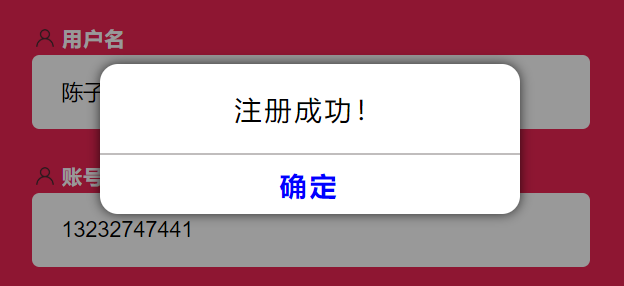

图为输入格式错误时的提示：



图为两次输入密码不匹配时的提示：  

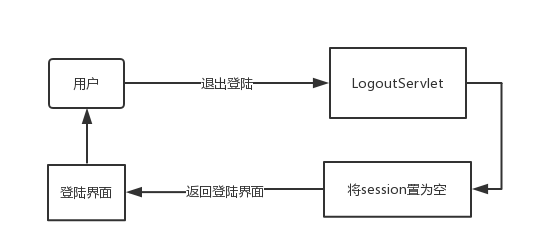

图为用户名不可用时的提示：  


图为账户不可能时的提示：  


图为注册成功时的提示：  


* 退出登陆功能：

退出登陆功能流程图:



1. 退出登陆功能讲解：

当用户点击退出登陆按钮时，前端会发送退出登陆请求，Servlet会将session保存的信息清空，前端切换至未登录状态，完成退出操作。

## 简单实现博客流

## 发布博客

## 修改个人信息

## 

# 项目亮点

### 登陆与注册表单

* 本项目中，登陆与注册表单都在同一个页面，仅需点击一下切换按钮，就可以自由切换登陆或者注册状态，这样做的好处是：用户不用花费多余的流量或者额外的时间去加载多一个注册或者是登陆的页面，尤其是在网络状态不好的情况下，会带来更好的用户体验。

实现原理：先将其中的一个表单隐藏起来，这里用到了CSS中的display属性设置为none，当用户点击切换按钮时，将隐藏的表单CSS中的属性display改为block，就成功切换至另一个表单。

* 登陆注册模块使用AJax技术，相比于同步提交表单，AJax的异步发送请求，可以在不重复刷新页面的同时，验证用户的输入，免去了多次加载同一页面造成的时间或者流量的浪费。同时，即使用户输入错误，也不会清空用户所输入的数据，而是可以让用户根据返回的错误提示修改自己输入的数据。想象一下，当你在填写一份问卷时，辛辛苦苦把前20个空填好了，然后因为某个地方错误了，就把你前面输入的数据全部清空了！这样的用户体验是非常差的。所以，本项目采用了异步提交表单的方式。

### 使用JSON格式传输数据

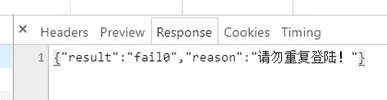
* JSON是什么

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

* JSON有什么好处
  + 与XML相比，JSON更加的简洁，我们可以一眼就看出其中的内容，方便检查排错
  + JSON更加轻量级，不管是编写，传输，还是解析都更加高效
  + JSON在传输过程中采用了压缩技术，更加的节省宽带
  + 众多的语言支持，如javascript，python，C，C++等主流语言都支持

简而言之，JSON是一种前后端传输数据的格式，它有着结构清晰，轻量化的特点，有便于开发人员操作数据、维护项目等特点。而且，在Java中，普通对象可以与JSON对象互转，普通数组也可以与JSON数组互转，非常简单自由，所以，本项目使用JSON作为前后端传输数据的格式。

图为进行登陆操作后台传回JSON格式的数据：



从图中，我们可以清楚地知道我们登陆操作的结果和原因

* result: fail0 代表错误类型是0
* reason：请勿重复登陆！代表错误原因是用户重复登陆了

这么清晰的数据格式，就算是不懂开发的人也知道是什么原因，因此，后台与前端的交接会变得简单许多，前端开发人员只需要根据传回来的数据就可以清楚地进行开发。

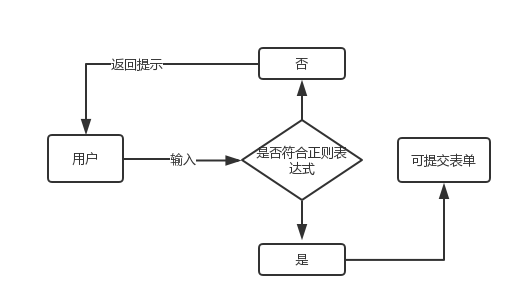
### 用JS实现输入验证

* 什么是正则表达式：

正则表达式是对字符串（包括普通字符（例如，a 到 z 之间的字母）和特殊字符（称为“元字符”））操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。正则表达式是一种文本模式，模式描述在搜索文本时要匹配的一个或多个字符串。

简而言之，正则表达式其中一个功能就是用来获得自己想要的数据的格式，过滤错误数据格式，除此之外，正则表达式还有很多种用途。一个做得好的前端项目，一定要有友好的提示，这样会使用户体验而本项目中，利用正则表达式实现了用户的输入验证。本项目中，主要用到了JavaScript中正则表达式的test()函数，如果用户输入的数据匹配正则表达式，会返回布尔值true，否则会返回false。然后根据test()函数的返回值制定相关的提示，提示用户修改自己输入的数据。

JS输入验证流程图：



例：下面是一条验证手机号码是否正确的正则表达式：

* + var patrn\_tel = /^1[0-9]{10}$/;

如果用户输入手机号码错误，那么输入框的下方会出现相应的提示，秉着没有提示就是最好的提示的原则，当用户输入正确后，错误提示会消失。

图为输入手机号码格式错误时的提示：  


图为输入正确时，错误提示会消失：



### 响应式布局

* 什么是响应式布局

根据维基百科及其参考文献，响应布局能够适应不同的设备。描述响应式布局最著名的一句话就是“Content is like water”，翻译成中文便是“如果将屏幕看作容器，那么内容就像水一样”。也就是说，同一个网页能根据不同设备屏幕大小不同，自动调整布局，呈现出几乎无区别的功能和完整的页面。

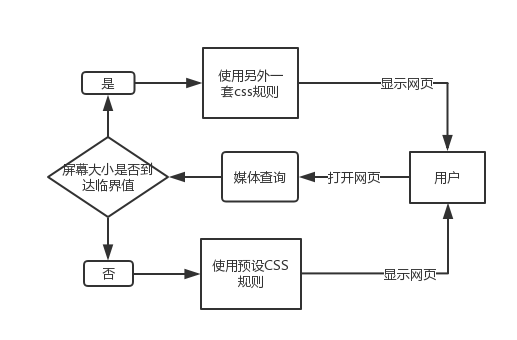
* 响应式布局的基本原理是：
  + 采用 CSS 的 media query 技术
  + 流体布局（ fluid grids ）
  + 自适应的图片/视频等资源素
  + 字体大小的设置，不能用固定像素值

在响应式布局出现之前，人们为了适应屏幕大小不同的设备，会额外开发多个页面，这样做不仅浪费了大量的时间、金钱、资源，而且增大了维护难度。而响应式布局，可以让一个网页适应多种设备，无论是手机、平板电脑、或者是pc都可以呈现出良好的效果，而且。响应式布局在网页上并不会自动为用户隐藏图片或简化菜单，而是将原本网页上所有的内容原封不动地进行展示，但是其他设备上的排版则会自动进行优化从而方便用户阅读、浏览以及使用该网页。

在现今这个时代里，科技日新月异，手机行业飞速发展，智能手机进入人们的生活，几乎达到了人手一台智能手机的普及速度，而手机行业的兴起带来了很明显的碎片化阅读的现象。用户对网页加载的速度要求越来越高，移动端有这么一个特点：不是所有的时候手机的网速都会很理想，所以一个网页没有针对移动端进行优化的话，漫长的等待时间会让用户失去耐心，比如打开一条链接需要10秒，而阅读链接里的内容也只花了20秒，那么在这个等待的过程中我是会失去耐心的，最后会导致流失用户。再者，假如没有针对移动端进行优化，进行适应移动端的布局，用户体验也是极差的，就好比在十分之一大小的电脑屏幕上使用同一个页面，用户就必须用手指不停放大缩小、上下左右腾挪，多费劲啊，我肯定会把它给关掉。用户的体验永远是第一位的，所以，必须要用适应性布局。

本项目中，本着这个原则，在某些页面实现了适应性布局的雏形，主要用到了CSS的媒体查询（media query）技术，还有CSS的flex布局技术等等，原理是CSS可以嗅探到设备屏幕大小的变化，当达到了设定值时就会执行新的CSS规则，比如改变一个容器的高度、宽度改变整体字体大小、改变图片大小等等。

适应性布局流程图：



图为Chrome浏览器下模拟Iphone X显示效果：



# 项目难点

### 使用jQuery实现Ajax

* jQuery是什么

jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库（或JavaScript框架）。jQuery设计的宗旨是“write Less，Do More”，即倡导写更少的代码，做更多的事情。它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。

jQuery的核心特性可以总结为：具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的css选择器，并且可对CSS选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。

* AJAX是什么

AJAX不是一种新的编程语言，而是一种用于创建更好更快以及交互性更强的Web应用程序的技术。使用 JavaScript 向服务器提出请求并处理响应而不阻塞用户！核心对象XMLHTTPRequest。通过这个对象，您的 JavaScript 可在不重载页面的情况与 Web 服务器交换数据，即在不需要刷新页面的情况下，就可以产生局部刷新的效果。AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息，而不是整个页面。

* AJAX 工作原理:

Ajax 是一种独立于 Web 服务器软件的浏览器技术。Ajax 基于下列 Web 标准：

JavaScript、XML、HTML与 CSS 在 Ajax 中使用的 Web 标准已被良好定义，并被所有的主流浏览器支持。Ajax 应用程序独立于浏览器和平台。Web 应用程序较桌面应用程序有诸多优势；它们能够涉及广大的用户，它们更易安装及维护，也更易开发。

不过，因特网应用程序并不像传统的桌面应用程序那样完善且友好。通过 Ajax，因特网应用程序可以变得更完善，更友好。

常用方法：

* + $.get(URL,callback);
  + $.post(URL,callback);
  + $.ajax();

Ajax技术在现代网页中几乎是随处可见，一个体验良好的网页，Ajax技术的应用几乎是必不可少的，本项目中也大量使用了Ajax技术。而jQuery中有一个非常强大的API，就是Ajax，它省掉了大量的冗杂代码。只需要简单写下几行代码，就可以实现强大的Ajax功能。JQ中Ajax接口有多种方法。本项目主要使用JQ的.ajax方法，这个方法非常强大，可以设置多种属性，还能根据请求的状态调用不同的函数。

来看一段.ajax代码的核心代码：



其中，声明了请求的方法为POST，希望得到返回的数据类型为json类型，全局属性设置为否，而提交的数据的形式为表单形式。async:”true”表示是异步提交请求，不使用缓存，而最后三行代码分别对应发送请求成功前、成功得到响应、和完成一次请求三个状态调用三个不同的函数。比如在登陆功能模块中，这三个函数我大致分为：禁用提交按钮防止重复提交表单，成功函数：根据后台返回的数据进一步对用户进行提示，完成函数：登陆成功，跳转至主页。

再来说说它为什么难，哪里难，我是怎么样去解决的。

1. 难在要理解HTTP协议，要了解POST、GET等请求的含义及其工作的流程，难在如何联合调试后台与前端，后台怎么样拿到前端传过来的数据，前端怎么样使用后台传来的数据，这里我用了Postman这个软件来帮助我去调试。
2. 难在JQ的这个Ajax api 在网上的教程是参差不齐的，也可以说是没有完整的教程，而又因为JQ高度的自由性，导致很多人都写出了一些不是很规范的代码，这个时候就会出现莫名其妙的BUG。再者，有一些属性的值需要仔细斟酌，就比如是否使用缓存这个属性，就需要搭配上全局属性，不然造成污染，导致整个网页的请求都不使用缓存，最后可能会出现一些莫名其妙的bug。我是怎么样去解决的呢？一方面，我通过在网上寻找资料，发掘一些好文章，看看别人是怎么样去解决同一个问题的，另一方面，我不停调试自己的代码，就比如属性值的问题，我就一个一个属性进行调试，最终得到自己想要的效果，前后端跑通了。

# 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。

### 技术上的收获

我把参加工作室的这段日子分为三个阶段：第一阶段：准备工作室的笔试面试，第二阶段：参加QG工作室的训练营，第三阶段最后12天的考核期，在这整个过程中，我在技术上可以说是收获颇丰。

首先，第一阶段其实上学期就已经开始的了。上学期我参加了QG工作室的学习交流分享会，要加入QG工作室，就必须要提前准备。自己确定了要走前端的方向，于是在寒假的时候就开始学习一些关于前端的知识，还有复习上学期学过的C语言，直到开学笔试前这段日子都是在做着这些准备。

在第一阶段技术上的收获不是很多，到了第二阶段，我顺利进入了QG工作室的训练营。在训练营的那段日子里，学习方向分为大组和小组，大组学习数据结构，小组学习网路组方向的知识。在大组培训里，我提前学习了大二才会学到的数据结构的部分知识，比如链表、排序、栈、队列等等，而且对于C语言也有更深入的理解，也更加熟练。比如使用多文件编程，用C语言写出能交互的程序等等。小组培训里，前端方面，当然是更深入学习、探索，我学到了如何用HTML做出一个良好的布局；我学到了如何用CSS去美化我的页面；我学到了如何用JS让我的页面具有交互性等等。而且我不仅学到了前端方面的知识，还学到了后台的方面的知识，我学到了HTTP协议、POST请求与GET请求的区别；我学到了数据库设计的三大范式；我学到了如何搭建一个本地服务器；经过这么多些日子的学习，我发现前端和后台其实是不可以分割的，两者都要相互了解对方的工作原理，要学会读懂对方的代码。

除了学会了写代码，我还学会了如何使用Git和Github去管理我的源码，我还学会了如何用优美的MD语言写出条理清晰、结构完整的笔记、周记。

第三阶段，为期12天的考核期，在这12天里，我从零开始学习后台相关的知识，学习怎么设计和搭建一个数据库，学习用Java编写后台，学习如何真正系统地开发一个项目，第三阶段更多是把前两个阶段学到的东西运用起来，并且不断学习，添加新的东西进去。

关于收获不仅仅只有技术上的收获，还收获了其他很多东西，只是技术上的收获比较直观，有些收获是体现在个人品质中的，是可以直接影响到一个人的气质特点、做事方面的，总而言之，在这段日子里，我收获了很多很多，非常感谢。

### 为什么选择前端？

当初是抱着试一试的心态去了解前端，在深入了解学习后，确认过眼神，我遇到对的前端。对于前端的作用定位，我自己认为，前端是用户与一个程序直接接触的部分，是一条桥梁，前端不仅仅局限于网页，前端可以是一个安卓程序的可操作部分，前端也可以是C语言的一个简单的黑窗口。前端是一个程序的展示部分，一个程序如果前端做得特别烂，那么就算是它的后台多么稳固或者算法写得多牛逼都永远是一个失败的程序，因为用户的体验才是第一位。前端其实也存在于生活中，平时我在生活中会去观察发现身边好看的东西，也喜欢去设计一些东西，比如班服、班旗等，还有编写出一篇好的推文，这些其实都和前端有一些相同的特点，比如追求美、追求稳定完整的结构等。我还认为，一个好的前端同时也是一个好的设计师，前端与设计师的区别是前端用的是代码进行设计，我不敢说自己是一名设计师，但至少我可以肯定我是一个有审美的人。所以，对于我而言，选择前端有自身兴趣的原因，都说兴趣是最好的老师，做一件自己喜欢的事情，总是可以投入更多的时间，保持热情，自然不会感到枯燥。因此，我有什么理由不选前端呢？

### 对于未来的发展

对于未来的发展，结合当下的情况，不论最终工作室的结果如何，我选择继续走前端这条路。我会把前端作为我专业的方向，我见过有些师兄到了大三大四才开始选择自己要走什么方向，那个时候已经太晚了，假如我在四年里精通一个方向，那么我肯定会在竞争者中更有优势。我会保持对前端的热情，继续学习前端方面的知识，我现在学习到的知识，还是比较表层的，比如我对网页是怎么渲染的、JS的工作原理等都是比较模糊的，还有很多深奥、底层的东西东西值得我去探索。如果我有能力，我可能还会在开源社区搭建一个属于我自己的博客，或者写一个真正能运行在互联网上属于我自己的博客系统。未来存在着很多不确定的因素，但是我相信只要我的心够坚定，未来有我的一席之地。

### 遇到问题时我是怎么样去解决的

在学习、编程的过程中，遇到问题是不可避免。针对不同类型的问题我有不同的解决方法，假如我遇到的是一个关于还没学到的知识的问题，而且我清楚地知道它是和哪部分的知识是有关的，我就会先放下这个问题，去学习这一部分的知识，我会看书、看视频，看博客文章，等我完全理解了这部分的知识点时再去解决这个问题。另外一类是已经学会了的知识点出现的问题，这些问题就是bug，那么我平时是怎么样debug的呢？假如是语法型的错误，有一些编译器会提示你哪一行代码错了，那么这个时候只需要去修正这一行代码。如果不是语法性的错误，通常我会用打断点，注释代码一块一块代码来进行测试，再者，我会利用搜索引擎，搜索同类型的问题，还可以看别人的技术博客。最后如果还是无法解决那么我会问同学或者是师兄，和他们一起讨论怎么解决这个问题。

### 衷心的感谢

不知不觉，QG工作室的训练营已经快要到达尾声了。这些日子以来，自己过得并不轻松，训练营的作业，课程的作业，还有其他一些事情，经常会比较忙，周末我都选择在图书馆度过，有时候真的会感觉分身乏术。考核期的压力更大。但是无论有多苦，自己都没有想过要放弃，而是继续坚持，坚持自己的选择。我本可选择走更轻松的路，但是我知道，QG工作室不仅仅是把你培养成一个好的程序员，而是要把你培养一个全方面发展的人，一个更优秀的人。因此，要变得更优秀，就必须比别人多吃一些苦。我知道自己可能不是一个天赋出众的人，但人贵有自知之明，我懂得笨鸟先飞的道理，要追赶别人，我就需要花比别人更多的时间、更多的努力。这一路上，开心的是，我认识了很多好友，我们一起学习，一起讨论编程上的问题，一起为加入工作室而努力。我也认识了很多优秀的师兄师姐，在他们的身上我能够学到优秀的品质，很感谢他们给我们上课，让我们学习到了新的知识。有时候能够得到师兄师姐们的肯定那种感觉是很棒的，虽然一路走来绝对不轻松，但是看看自己写下一行又一行的代码，看看自己在这段时间学习到的知识，看看自己的作品，就会觉得这一切都是值得的。

最后衷心感谢QG工作室给了我一个成为一个更优秀的人的机会，最后无论结果如何，对于我来讲，都已经是最好的结果。