

Beijing University of Posts and Telecommunications
北京邮电大学

2020web 开发技术基础课程

简易版微博客系统的设计与实现

技术文档

2018211314 班 李志毅 2018211582

2018211314 班 陈硕 2018211584

2018211314 班 杜政龙 2018211578

2021 年 01 月 04 日

目录

一、项目简介	3
二、开发思路	3
三、开发环境	4
3.1 环境综述	4
3.2 环境介绍	4
3.2.1 Spring MVC	4
3.2.2 Spring Boot	4
3.2.3 H2 数据库	5
3.2.4 Bootstrap 4.0 框架	5
3.2.5 阿里云对象存储 OSS	6
四、总体设计	7
4.1 用户服务	7
4.1.1 用户注册	7
4.1.2 用户登录	7
4.1.3 用户退出	7
4.1.4 用户重置密码	8
4.1.5 用户账户保护	8
4.2 微博服务	8
4.2.1 微博列表展示	8
4.2.2 微博发布	8
4.2.3 微博评论	8
4.2.4 微博点赞	8
4.3 个人服务	9
4.3.1 个人主页	9
4.3.2 关注与取消关注	9
五、模块设计	10
5.1 页面与逻辑设计	10
5.1.1 需求分析	10
5.1.2 环境	10
5.1.3 概要设计	10
5.1.4 详细设计	12
5.2 接口设计	26
5.2.1 需求分析	26
5.2.2 详细设计	26
5.3 数据库设计	32
5.3.1 数据库环境	32
5.3.2 关系模式设计	32
5.3.3 图片的存储	32
六、问题总结	33
6.1 李志毅	33
6.2 陈硕	33
七、完结感想	34

一、项目简介

设计实现一个简易的类“微博”的单体式 Web 应用，其中需要实现的功能包括基础功能和拓展功能。

基础功能包括：

- 用户账号服务：用户注册、用户登录、用户退出
- 微博查看：微博列表，支持按发布时间、评论数、点赞数等排序方式，微博列表可分页，未注册用户可以查看微博内容及评论。
- 微博发布：注册用户可以发布微博，微博内容可包含图片
- 评论：注册用户可对微博发布评论，点赞
- 关注：支持用户可以关注其他用户

拓展功能包括：

- 登录密码输入错误 3 次后账号锁定 1 小时
- 支持通过注册邮箱重置密码
- 支持@用户功能
- 采用 AJAX 技术减少界面的整体刷新次数

二、开发思路

1. 便捷性

用户仅需要简单的登录和注册即可进入主页面，并且登录用户具有发布微博、点赞微博、发表评论、关注用户等权限，即使是非注册用户，也可以直接查看主页上的微博内容和微博下的评论内容

2. 简洁性

整个系统总计 6 页面，分别是用户服务的注册、登录、重置密码页面，以及整体的主页、个人主页、错误返回页面。

系统简洁明了，各项功能和页面的导览清晰可见，对于用户有较好的体验效果，对于游客也有良好的用户体验。

3. 灵活性

系统拥有排序和分页功能，对于主页微博列表，可以根据三种不同的参数进行排序分页，适用于不同场景的微博查看，同时对于用户点赞、关注和评论等功能有着较好的灵活选择性。

三、开发环境

3.1 环境综述

基于 Spring MVC+SpringBoot+H2 数据库+Bootstrap4.0 的开发,
开发环境为 IntelliJ IDEA 2020.2.3 工具
在 chrome 浏览器环境和 Microsoft Edge 环境下进行调试

3.2 环境介绍

3.2.1 Spring MVC

Spring Web MVC 是一种基于 Java 的实现了 Web MVC 设计模式的请求驱动类型的轻量级 Web 框架，即使用了 MVC 架构模式的思想，将 web 层进行职责解耦，基于请求驱动指的就是使用请求-响应模型，框架的目的就是帮助我们简化开发，Spring Web MVC 即可以用来简化我们日常的 Web 开发。

Spring Web MVC 也是服务到工作者模式的实现，但进行可优化。前端控制器是 DispatcherServlet；应用控制器其实拆为处理器映射器(Handler Mapping)进行处理器管理和视图解析器(View Resolver)进行视图管理；页面控制器/动作/处理器为 Controller 接口（仅包含 ModelAndView handleRequest(request, response) 方法）的实现（也可以是任何的 POJO 类）；支持本地化（Locale）解析、主题（Theme）解析及文件上传等；提供了非常灵活的数据验证、格式化和数据绑定机制；提供了强大的约定大于配置（惯例优先原则）的契约式编程支持。

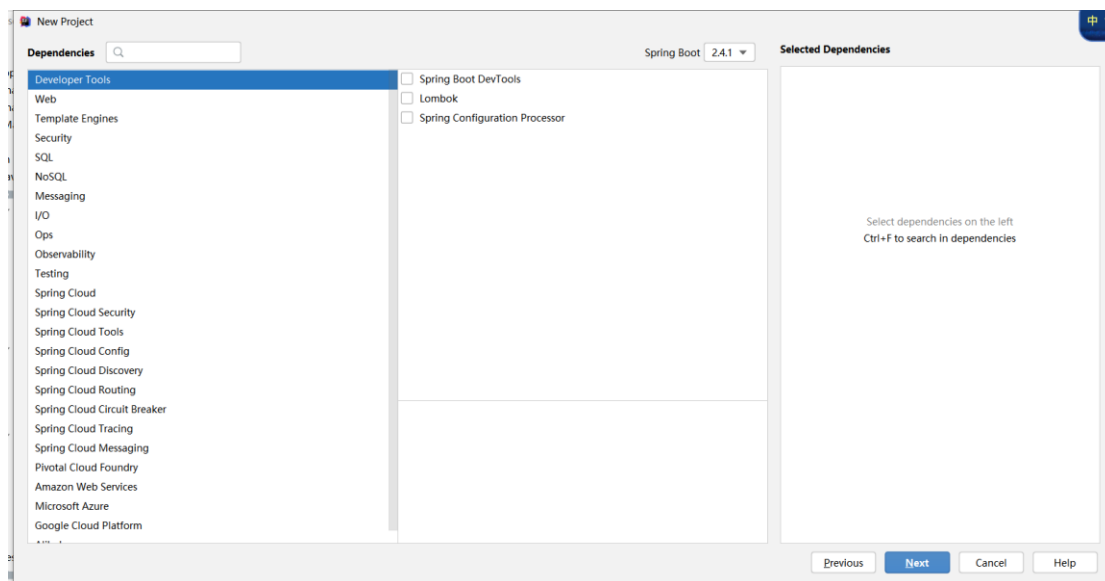
在 Spring 的 Web MVC 框架提供了模型 - 视图 - 控制器架构以及可用于开发灵活，松散耦合的 Web 应用程序准备的组件。

3.2.2 Spring Boot

SpringBoot 是由 Pivotal 团队在 2013 年开始研发、2014 年 4 月发布第一个版本的全新开源的轻量级框架。它基于 Spring4.0 设计，不仅继承了 Spring 框架原有的优秀特性，而且还通过简化配置来进一步简化了 Spring 应用的整个搭建和开发过程。另外 SpringBoot 通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

SpringBoot 应用系统开发模板的基本架构设计从前端到后台进行说明：前端常使用模板引擎，主要有 FreeMarker 和 Thymeleaf，它们都是用 Java 语言编写的，渲染模板并输出相应文本，使得界面的设计与应用的逻辑分离，同时前端开发还会使用到 Bootstrap、AngularJS、jQuery 等；在浏览器的数据传输格式上采用 Json，非 xml，同

时提供 RESTful API; Spring MVC 框架用于数据到达服务器后处理请求; 到数据访问层主要有 Hibernate、MyBatis、JPA 等持久层框

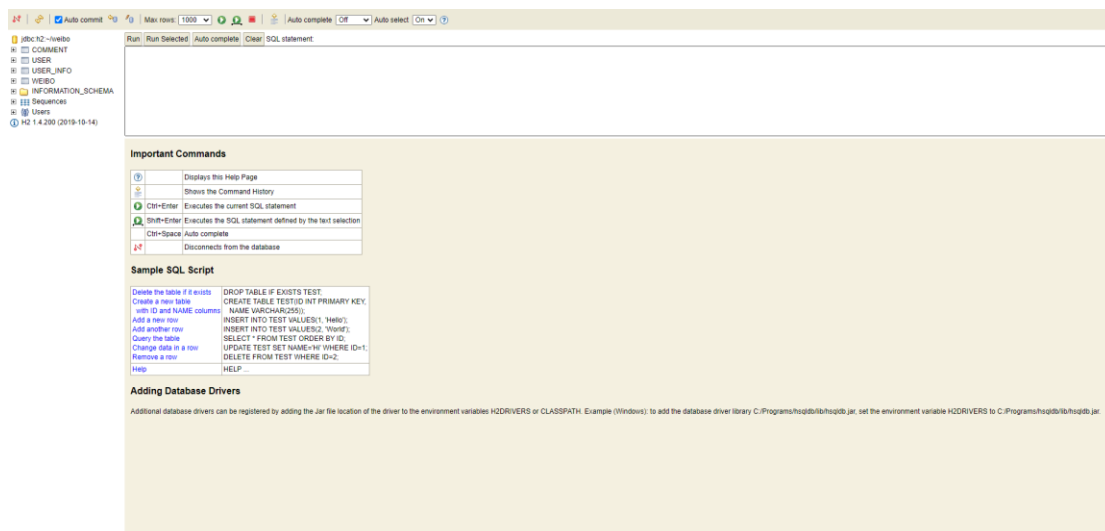


3.2.3 H2 数据库

H2 是一个 Java 编写的关系型数据库，它可以被嵌入 Java 应用程序中使用，或者作为一个单独的数据库服务器运行。

H2 数据库的前身是 HypersonicSQL，它的名字的含义是 Hypersonic2，但是它的代码是从头开始编写的，没有使用 HypersonicSQL 或者 HSQLDB 的代码。

H2 提供了一个十分方便的 web 控制台用于操作和管理数据库内容。H2 还提供兼容模式，可以兼容一些主流的数据库，因此采用 H2 作为开发期的数据库非常方便。



3.2.4 Bootstrap 4.0 框架

Bootstrap 是美国 Twitter 公司的设计师 Mark Otto 和 Jacob Thornton 合作基于

HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。Bootstrap 一经推出后颇受欢迎，一直是 GitHub 上的热门开源项目，包括 NASA 的 MSNBC（微软全国广播公司）的 Breaking News 都使用了该项目。国内一些移动开发者较为熟悉的框架，如 WeX5 前端开源框架等，也是基于 Bootstrap 源码进行性能优化而来。Bootstrap 是全球最受欢迎的前端组件库，用于开发响应式布局、移动设备优先的 WEB 项目。

Bootstrap 是一套用于 HTML、CSS 和 JS 开发的开源工具集。利用其提供的 Sass 变量和大量 mixin、响应式栅格系统、可扩展的预制组件、基于 jQuery 的强大的插件系统，能够快速开发出原型或者构建整个 app。

Bootstrap

Bootstrap 是全球最受欢迎的前端组件库，用于开发响应式布局、移动设备优先的 WEB 项目。

Bootstrap 是一套用于 HTML、CSS 和 JS 开发的开源工具集。利用我们提供的 Sass 变量和大量 mixin、响应式栅格系统、可扩展的预制组件、基于 jQuery 的强大的插件系统，能够快速为你的想法开发出原型或者构建整个 app。

[快速入门](#)[下载 Bootstrap](#)

当前版本: v4.2.1



3.2.5 阿里云对象存储 OSS

对象存储 OSS 是海量、安全、低成本、高可靠的云存储服务，提供 99.999999999%(12 个 9)的数据持久性。

OSS 作为阿里巴巴全集团数据存储的核心基础设施，多年支撑双十一业务高峰，历经高可用与高可靠的严苛考验。OSS 的多重冗余架构设计，为数据持久存储提供可靠保障。同时，OSS 基于高可用架构设计，消除单节故障，确保数据业务的持续性。

对于微博中的图片，我们使用 OSS 进行存储，通过 url 进行访问。



四、总体设计

4.1 用户服务

单个用户服务包含四个大块，用户注册、用户登录、用户退出以及用户通过邮箱重置密码服务

4.1.1 用户注册

其中用户注册需要提供以下信息

- 用户名：用户自拟，不得与已有用户名重复且长度需小于十位
- 邮箱：符合正规邮箱格式，不得重复注册
- 密码：用户自拟，符合密码规范

用户注册若不满足上述任意一个条件，都会弹窗警告，并允许用户再次提交，直到注册成功。注册用户可拥有发微博、发评论、关注和点赞等功能权限，而游客用户仅拥有对微博和评论进行查看的权限。

4.1.2 用户登录

注册成功的用户即可根据用户名和密码进行登录，用户名和密码需要匹配且符合规范、登录后即可跳转主页，对于未登录用户，也可直接访问主页但会缺少大部分权限。

4.1.3 用户退出

默认同一浏览器环境下仅允许一个用户登录，若已存在一个用户登录，在同一浏览器下再次登录操作将被禁止，因此主页右上角将提供退出按钮，用户点击后即可退出当前账户，登录另一账户。

4.1.4 用户重置密码

对于忘记密码的用户，可以通过邮箱和用户名进行重置密码，在登录页面和主页都提供重置密码的服务入口，用户提供用户名和邮箱之后，由系统进行判定是否一致并对邮箱进行验证，验证完毕后即可重置密码，重置密码需输入两次新密码，两次密码一致后重置才能生效。

4.1.5 用户账户保护

用户登录密码输入错误 3 次后账号锁定 1 小时，用于保护用户被恶意攻击

4.2 微博服务

4.2.1 微博列表展示

用户登录成功或游客直接访问主域名时，会跳转到主页，在主页上进行微博列表的展示，默认整个主页能容下三条微博(包含评论)，微博列表可以按照三种方式进行排序，分别是发布时间、微博点赞数量、微博评论数量，点击相应的 tabbar 可展示出不同排序下的微博列表，同时提供分页服务和分页操作，3 条以上的微博可以通过点击分页按钮进行上下页变动查看，分页操作提供首页和末页按钮，对于第一页微博和最后一页。

4.2.2 微博发布

发布微博按钮位于主页右侧方，对于已登录用户，提供发布微博的权限，发布微博可以包含图片，微博内容不得为空或超过 50 个字符，微博发布成功后会刷新主页，即可看到发布出的微博界面

4.2.3 微博评论

对于已登录用户，可以给主页上展示出的微博评论，每条微博下方提供评论区和评论书写栏，用户可以对相应的微博书写评论，点击发布后即可将评论发布而出并展示。

4.2.4 微博点赞

对于已登录用户，可以给主页上展示出的微博点赞，每个用户仅允许点击一次赞，再次点击后对应的赞将会取消并刷新页面，未登录用户不允许点赞。

4.3 个人服务

4.3.1 个人主页

每个已注册用户都拥有个人主页，个人主页中包含用户的用户名、头像、关注数、粉丝数、以及用户所发布的微博列表，主页有两种进入方式

- 一、通过点击微博列表主页左上角按钮“个人主页”，即可进入自己的主页中查看
- 二、通过点击每条用户发布的微博中对应的用户头像，即可进入他人的主页中查看

4.3.2 关注与取消关注

在用户个人主页中，提供了关注按钮（一颗小红心），用户点击小红心即可关注该主页对应的用户，用户不能关注自己，关注他人后，他人主页粉丝数刷新，且关注按钮小红心变为已关注的标志（对号√）

五、模块设计

本项目主要通过四部分进行实现，分别是，页面与逻辑设计模块，接口设计模块，数据库设计模块。

页面与逻辑设计模块主要通过简洁的界面与用户进行交互，通过界面的设计以及界面功能的实现，完成用户信息的收集以及用户信息的显示，同时完成微博列表的展示和用户对应功能入口的展示，包含前端和后端两个部分，分别代表前端的页面设计和后端的 Controller 设计。

接口设计模块辅助前端与后端进行交互，前端各类请求，接口对请求进行处理，并将信息存储至数据库中，或将数据库中的信息返回至前端。从而完成数据的保存与数据的获取。

数据库设计模块主要保存用户和群组的信息，后端通过接口访问数据库，数据库保存信息，并为由后端获取后给前端提供信息。

5.1 页面与逻辑设计

5.1.1 需求分析

当下软件的 GUI 界面设计成为了一个良好服务性程序的必要部分，一个好的 GUI 设计不仅可以为用户简洁的提供用户所需信息，以及所需功能，同时可以提供给用户良好的使用体验。因此，页面设计成为了本小程序主要需要完成的功能之一。

需要通过界面设计，使用户可以简洁快速的直达每一个功能模块，使用户可以快速找到想要的功能以及想要查看的内容，用户登录、注册、主页查看等。

5.1.2 环境

使用 Bootstrap4+jquery 作为前端的主要框架和辅助，对于前端样式和页面展示和用户不同设备下的体验都具有良好的支持。

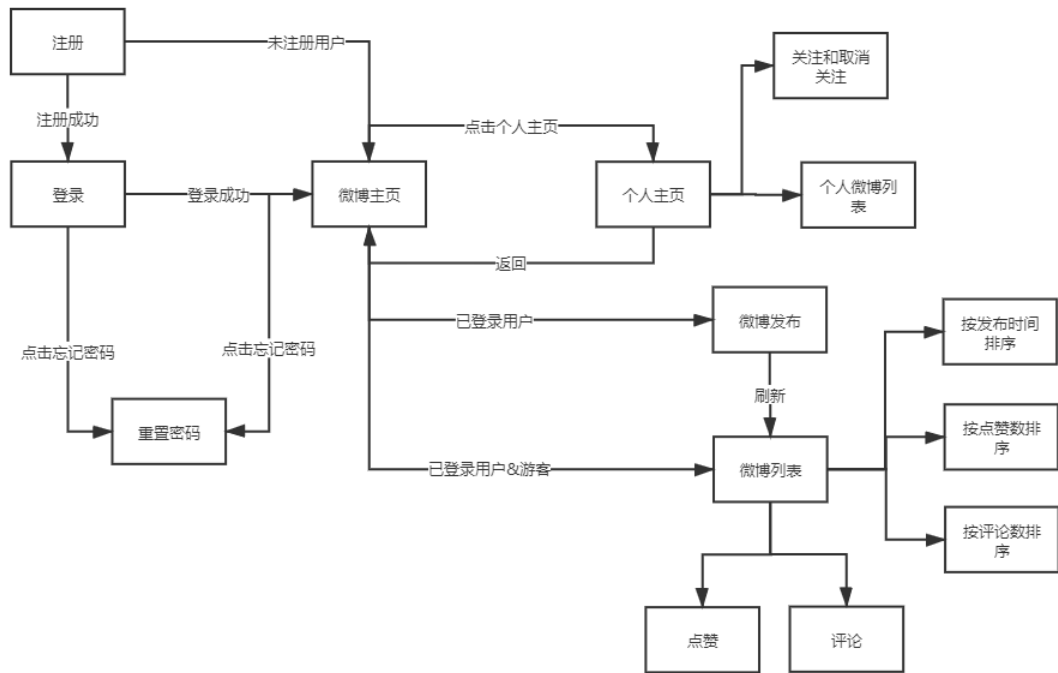
采用 Spring Boot 作为后端开发的框架，使用 H2 数据库和阿里云对象存储 OSS 进行数据的存储。

5.1.3 概要设计

每个页面都包含多项功能，其中主要功能包含显示信息功能，以及给用户相关操作功能，对于每一个页面的功能设计，可以提供给用户良好的使用体验，为用户清晰简洁的提供所需的信息。

下面通过页面调用关系图，展示各个页面的调用及跳转关系。

页面调用关系关系图：



用户打开系统后，首先进入主页，在主页可以查看当前系统中所有的微博列表。

右上角提供登录和注册按钮以及重置密码按钮，点击后会跳转到相应的页面进行操作

左上角 tabbar 中包含“我的主页”，点击后，可以对于用户账号进行基本的管理，以及主页的查看，关注和取消关注等功能。

在主页面上包含微博发布和微博列表两个主体展示，微博列表支持三种方式排序，微博发布可以发表图文内容后更新微博列表并展示，单个微博可以进行点赞和评论。

5.1.4 详细设计

5.1.4.1 注册页面

5.1.4.1.1 界面样式

用户注册

用户名(长度不超10位)

邮箱

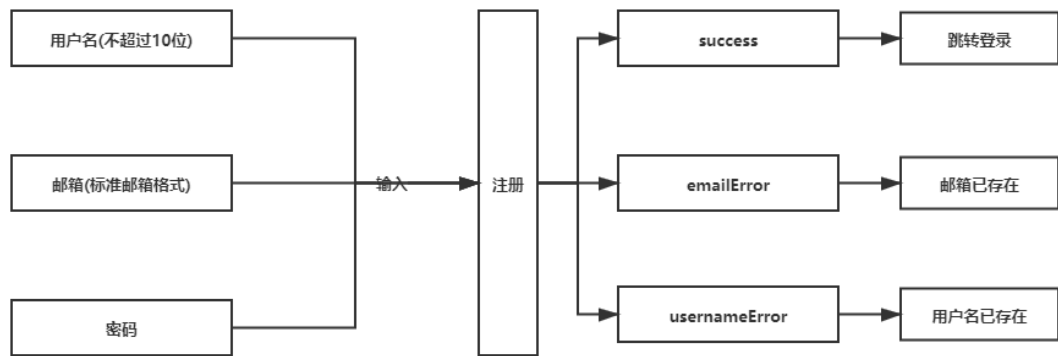
.

注册

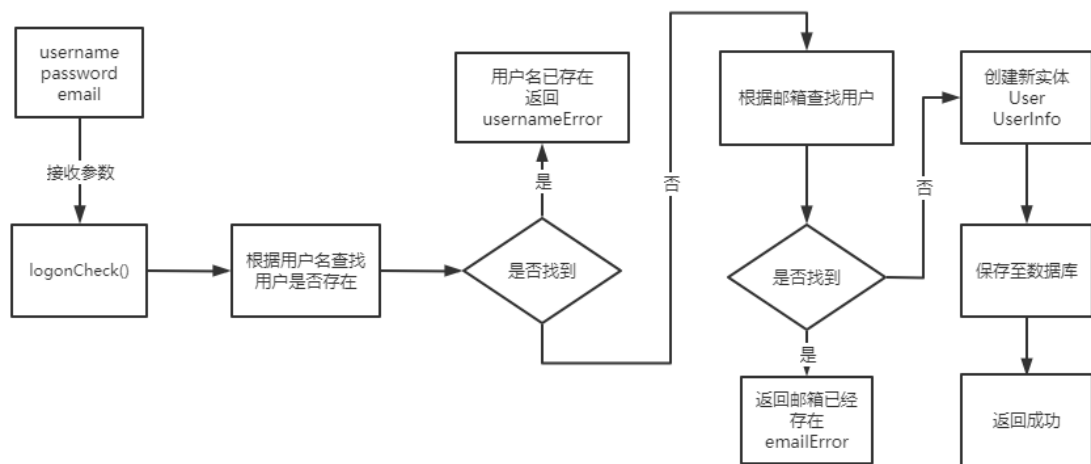
5.1.4.1.2 界面说明

页面包含标题和 title、以及三个 input 输入框，对应的类型分别为 text、email、password，其中用户名不得超过十位，邮箱为标准邮箱格式，点击注册按钮后将首先检测输入值是否合法，全部合法后向后端发送 ajax 请求注册，根据后端返回的信息进行下一步操作，后端返回三种 ResponseBody

- success：代表注册成功，跳转登录界面
- emailError：代表邮箱已被注册，邮箱已存在，需重新注册
- usernameError：代表用户名已存在，需重新注册



5.1.4.1.3 逻辑功能实现



函数 `logonCheck()` 接收参数，先通过用户名查找是否存在该用户，若存在，返回用户名已存在；若不存在，则根据邮箱查找用户是否存在，若存在，返回邮箱已存在错误，否则，注册成功。创建实体，并保存至数据库。

5.1.4.2 登录页面

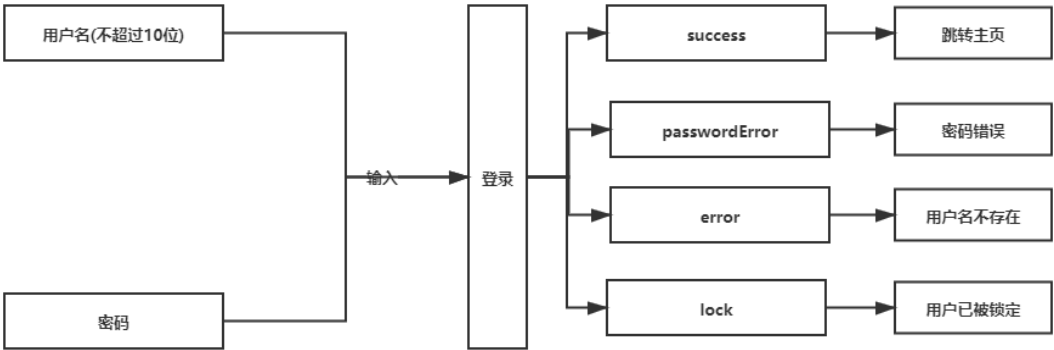
5.1.4.2.1 界面样式



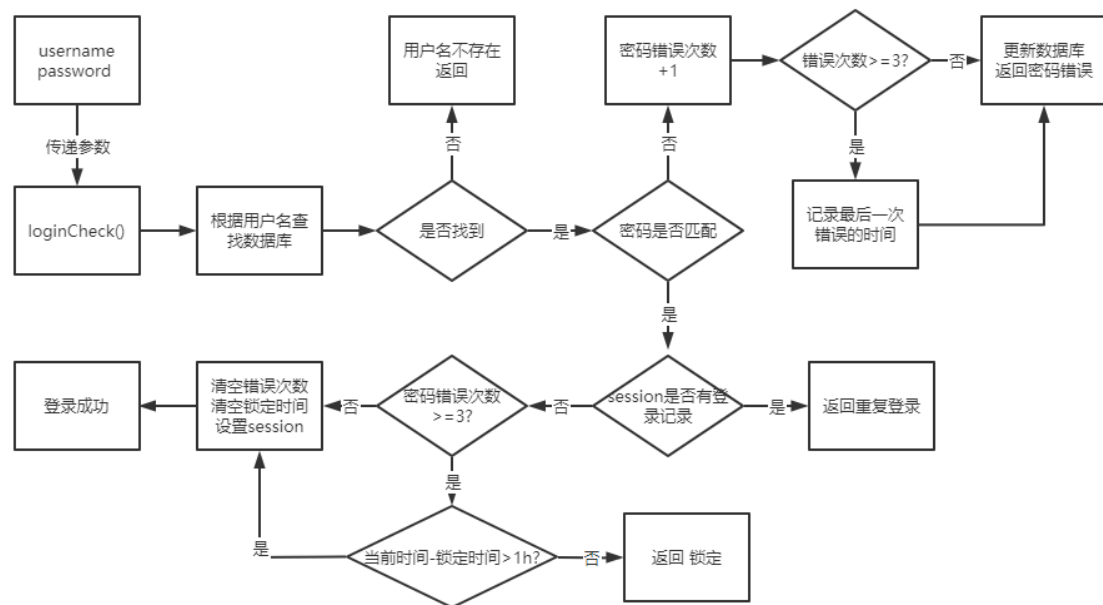
5.1.4.2.2 界面说明

页面包含标题和 title、以及两个 input 输入框，对应的类型为 text、password，用户输入用户名和密码后，点击登录按钮将首先检测输入值是否合法，全部合法后向后端发送 ajax 请求登录，根据后端返回的信息进行下一步操作，后端返回三种 ResponseBody

- success：代表登录成功，跳转主页界面
- passwordError：代表密码错误，用户名和密码不匹配
- error：代表用户名不存在，数据库中查无此人
- lock：代表当前用户已被锁定（输入三次错误密码）



5.1.4.2.3 逻辑功能实现



函数 `loginCheck()` 负责登录逻辑，本函数在一般的登录功能基础上实现了锁定功能，即在用户 3 次输错密码后，会将账号锁定 1 小时。同时，如果检测到当前浏览器已经登录过一个账号，则会拒绝第二个账号登录。

5.1.4.3 重置密码页面

5.1.4.3.1 界面样式

重置密码

您的用户名

您的邮箱地址

输入新密码

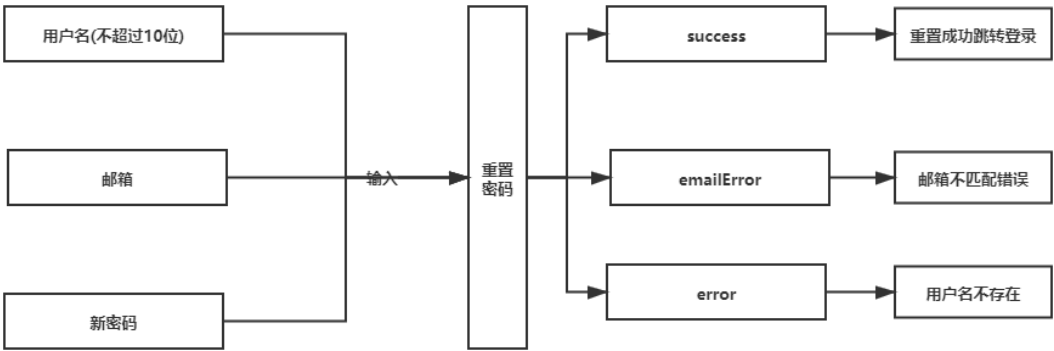
再次输入新密码

重置

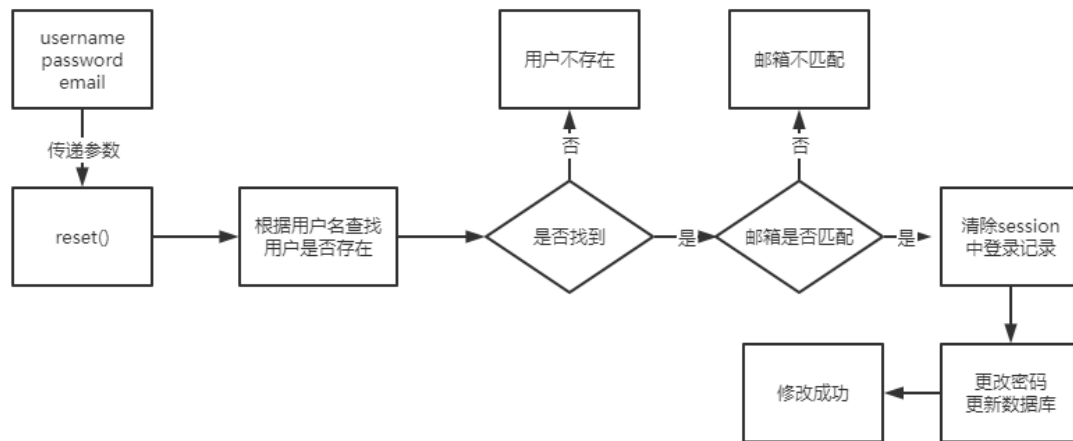
5.1.4.3.2 界面说明

页面包含标题和 title、以及四个 input 输入框，对应的类型为 text、email、password、password，分别代表用户名、邮箱地址和两次输入的新密码，用户输入完成后，点击重置密码按钮将首先检测输入值是否合法（两次密码是否匹配等），全部合法后向后端发送 ajax 请求重置密码，根据后端返回的信息进行下一步操作，后端返回三种 ResponseBody

- success：代表重置成功，跳转登录界面
- emailError：代表邮箱不匹配错误 error
- error：代表用户名不存在



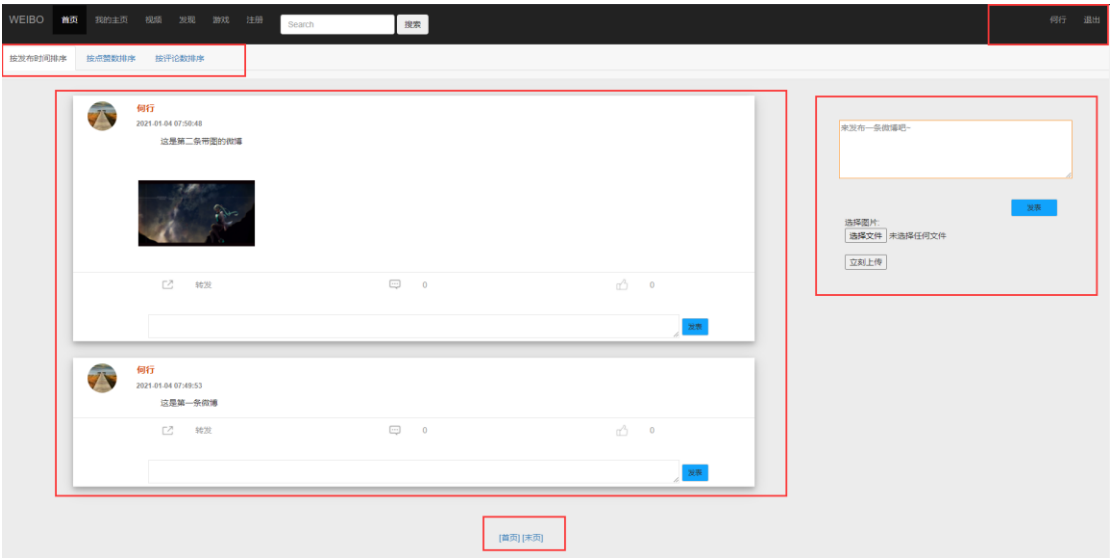
5.1.4.3.3 逻辑功能实现



函数 `reset()` 实现了重置密码功能，在修改成功时，清除 session 中的登录记录，让用户输入新密码进行重新登录。

5.1.4.4 微博列表主页

5.1.4.4.1 界面样式



5.1.4.4.2 界面说明

主页包含多个元素，其中主要功能的展示和入口均在微博列表主页中，主要分为如下几个部分

1.tabbar



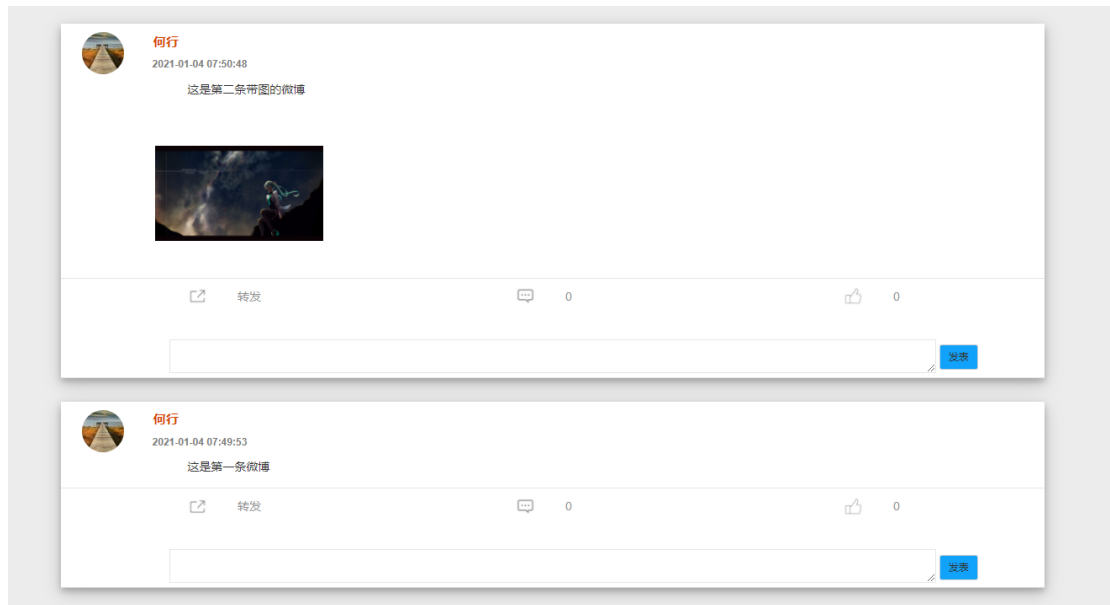
顶栏导航栏部分由几部分组成，左侧为相应功能导览，主要是其中的“首页”和“我的主页”导览，点击后会进入相应的页面，右侧包含用户名和退出按钮(若未登录用户名一项显示为登录)，点击后可跳转至相应页面。

2.排序方法

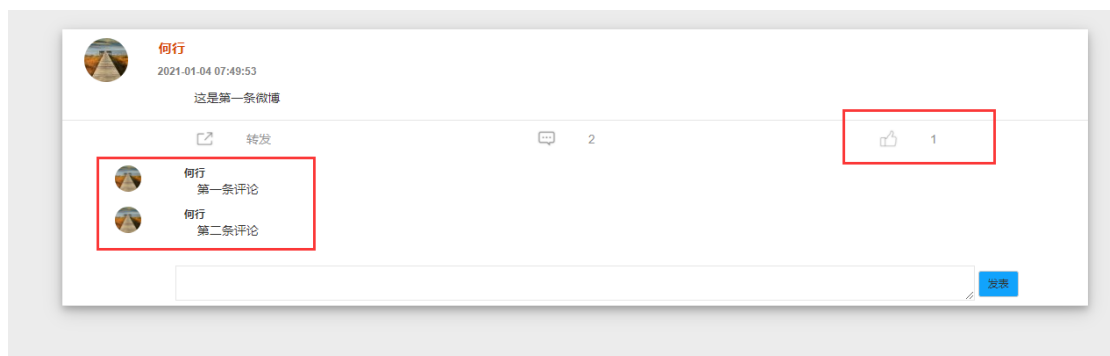


支持三种排序方式，分别为按发布时间排序、按点赞数排序、按评论数排序，点击后会更新微博列表中的微博单体顺序

3.微博列表



微博列表如上图, 微博内容可包含图片, 每条微博下可以进行评论和点赞(已登录用户) 点赞和评论效果为:



4.分页按钮



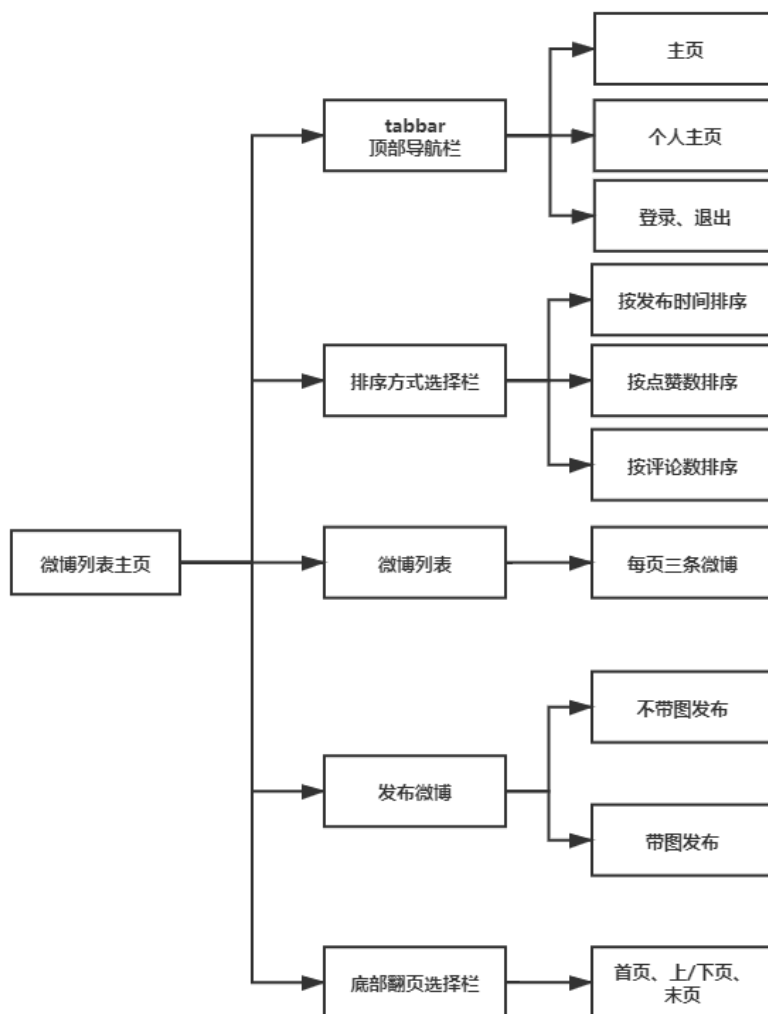
分页按钮位于最下方, 默认一页容纳三条微博, 点击后可进行翻页操作

5.发布微博



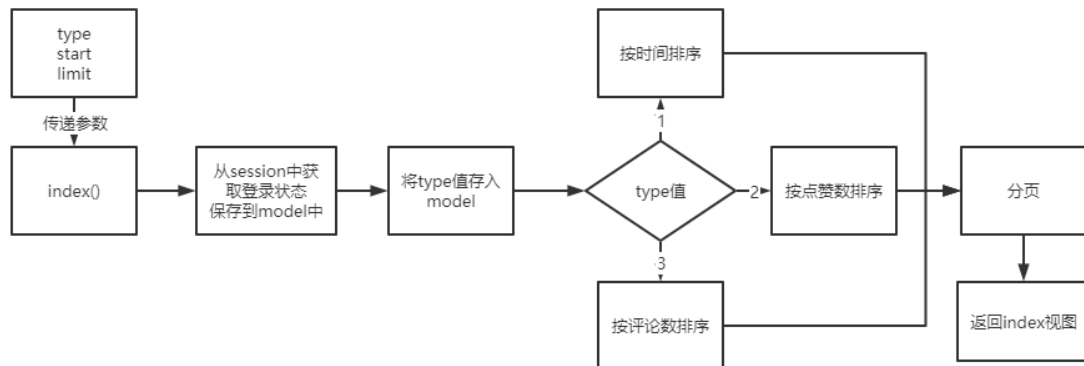
发布微博位于主页的右侧，已登录状态下发表按钮才会处于 able 状态，发布微博可以只文字发布，或带图发布，带图发布首先选择图片文件并点击上传后，再点击“发表”按钮即可，不允许发送空内容微博。

主要流程如下：



5.1.4.4.3 逻辑功能实现

1. 微博显示



index()函数有两个，分别使用 GET 方法和 POST 方法。

对于 GET 方法，前端需要判断当前排序的方式进行标签的显示，如下图：



所以，需要向 model 里传入一个 type 参数，让前端容易判断。

对于 POST 方法，因为点击另外一页不需要刷新整个页面，因此，我们使用了 thymleaf 的分段功能，在需要分段的地方设置

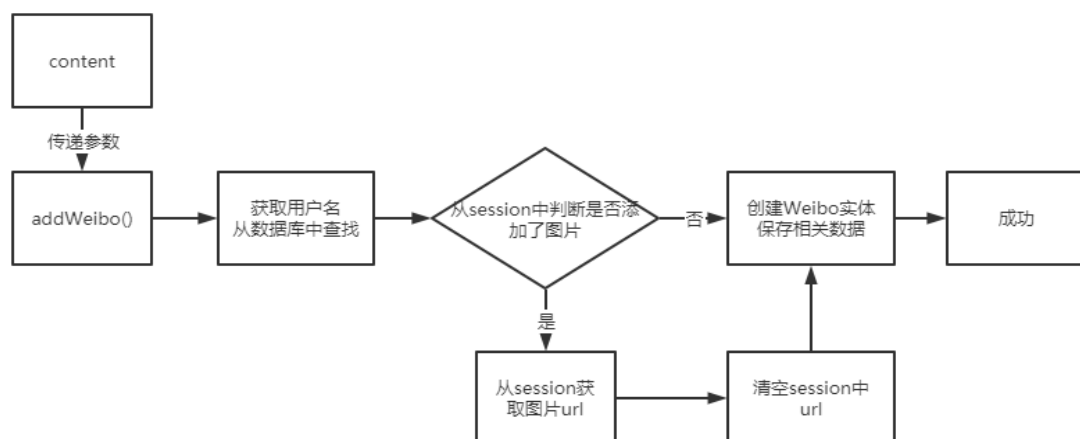
```
1. <div th:id="weiboList" th:fragment="weiboList">
```

后端返回为

```
1. return "index::weiboList";
```

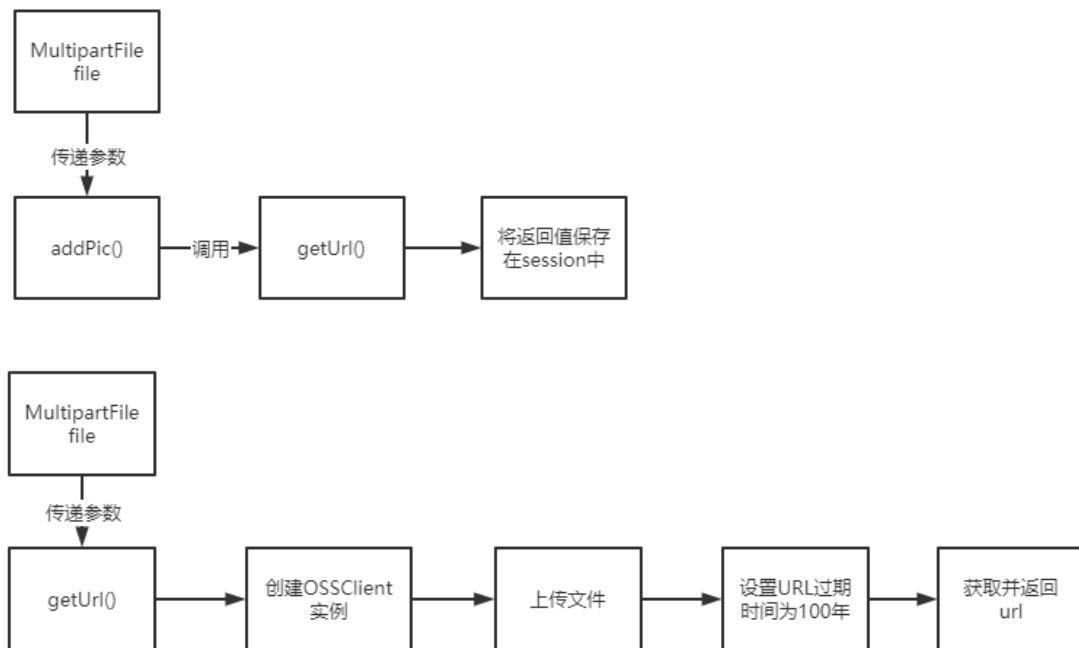
这样就可以只对对应部分进行重新渲染。

2. 添加微博



函数 addWeibo()用于添加微博。

3. 图片添加

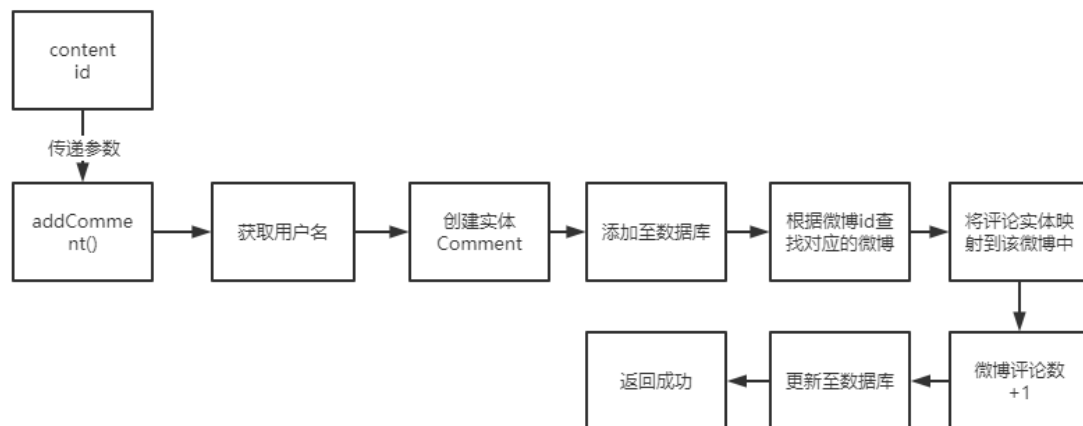


函数 addPic()用于保存图片，该函数又调用函数 getUrl()

getUrl()用于与 OSS 的交互，设置好自己的账户信息后，便可以将图片上传到 OSS 中，并获取其 url

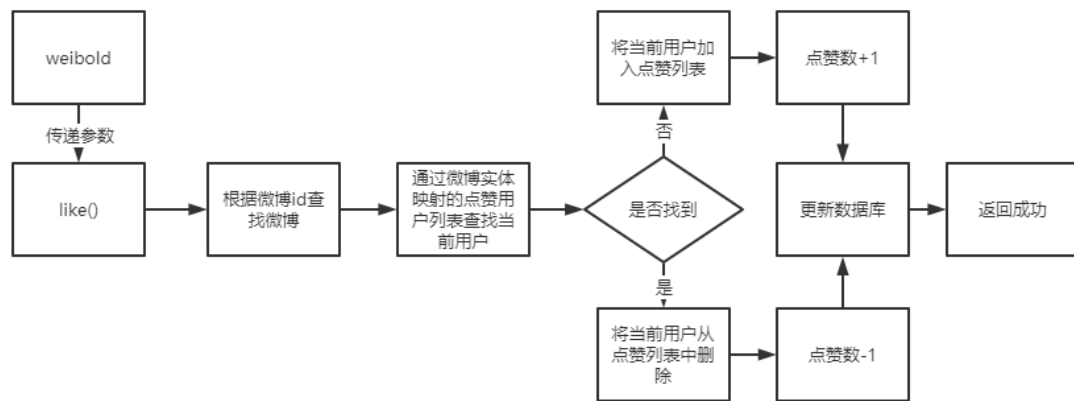
函数 addPic()将 getUrl()返回的 url 保存至 session 中，供发微博使用。

4. 发表评论



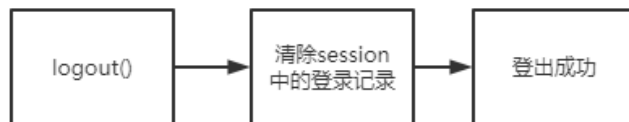
此处比较关键的一步是实体间的映射，我们需要将评论实体映射到对应微博实体中，以便于查找评论。

5. 点赞/取消赞



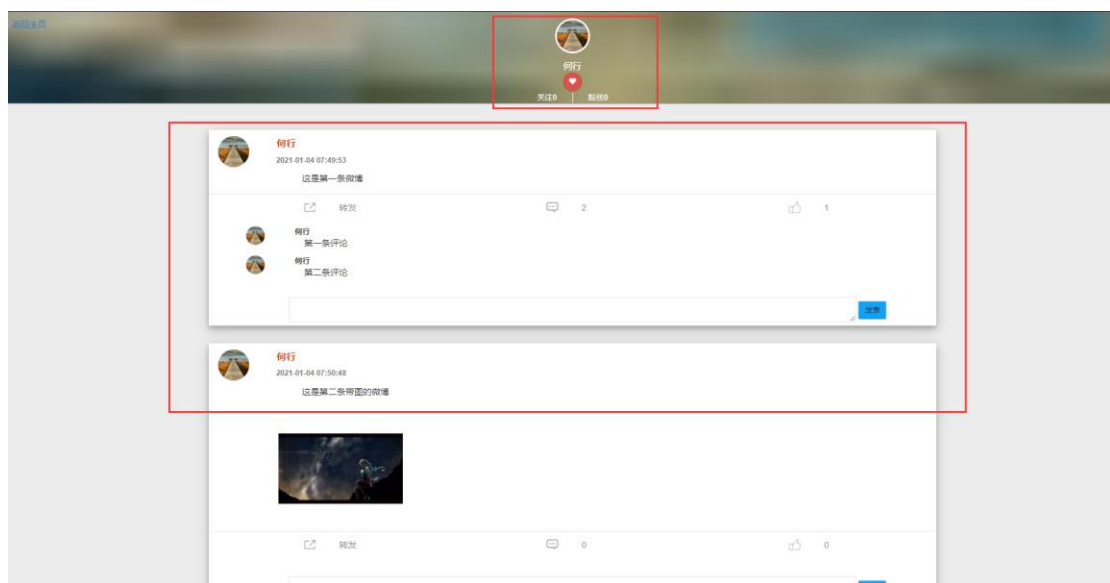
本函数实现了两个功能，点赞，当当前用户已经赞过，则再次点击会取消赞。

6. 登出



5.1.4.5 个人主页页面

5.1.4.5.1 界面样式

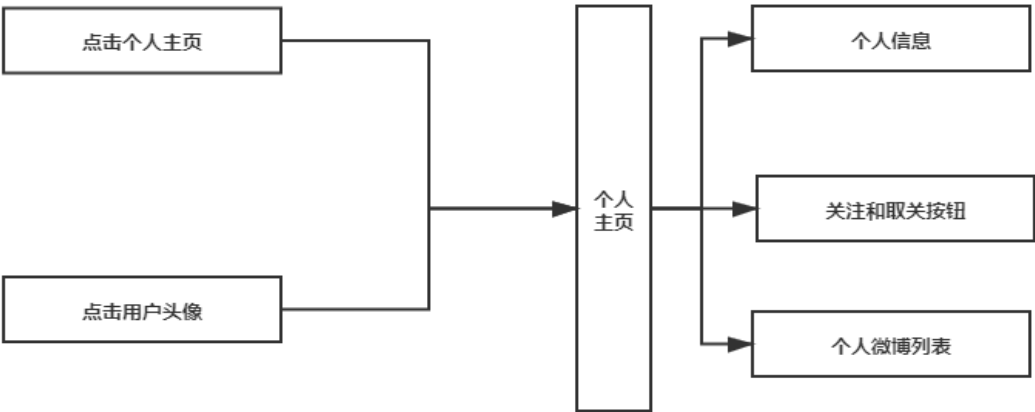


5.1.4.5.2 界面说明

个人主页页面包含当前用户的个人信息，以及其发布的所有微博列表，在每个用户的用户名下方都会有一个按钮，按钮若为小红心代表可以点击小红心来关注该用户，若已关注该用户，则按钮会变成对勾，入下图所示：

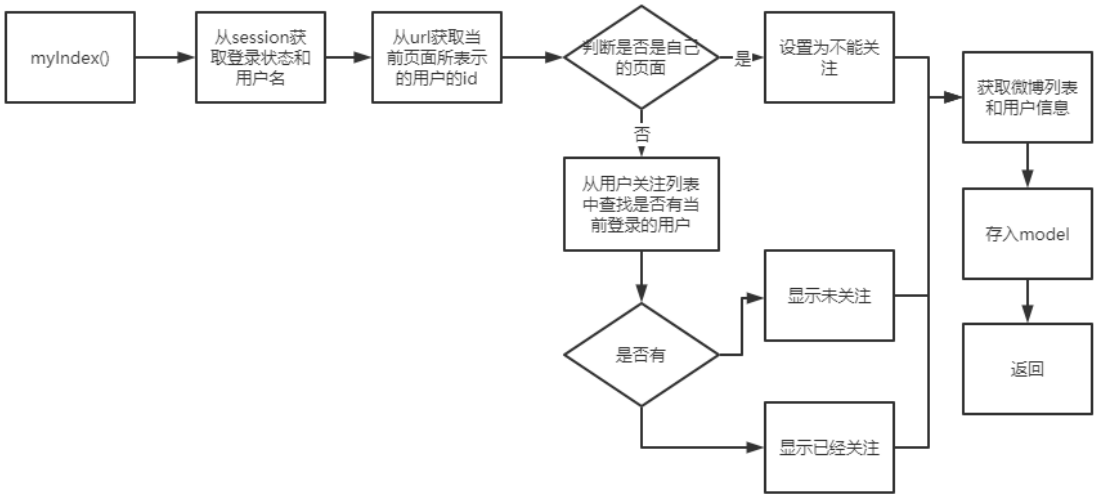


主要页面逻辑为：



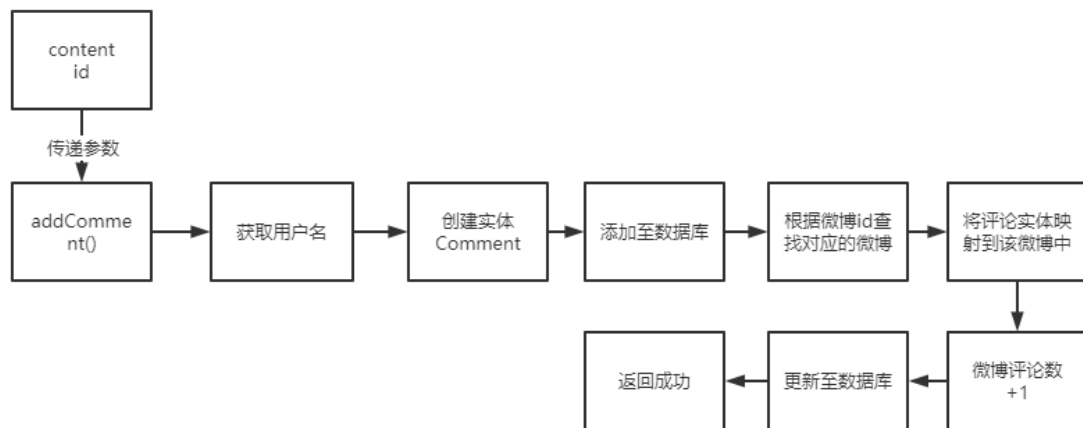
5.1.4.5.3 逻辑功能实现

1. 微博显示



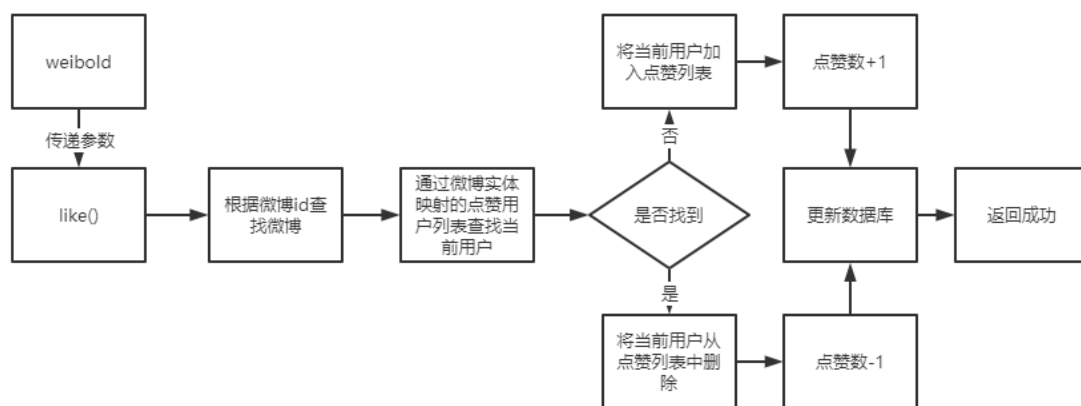
由于用户不能关注自己，因此需要判断所打开的页面是否是自己的主页。

2. 发表评论



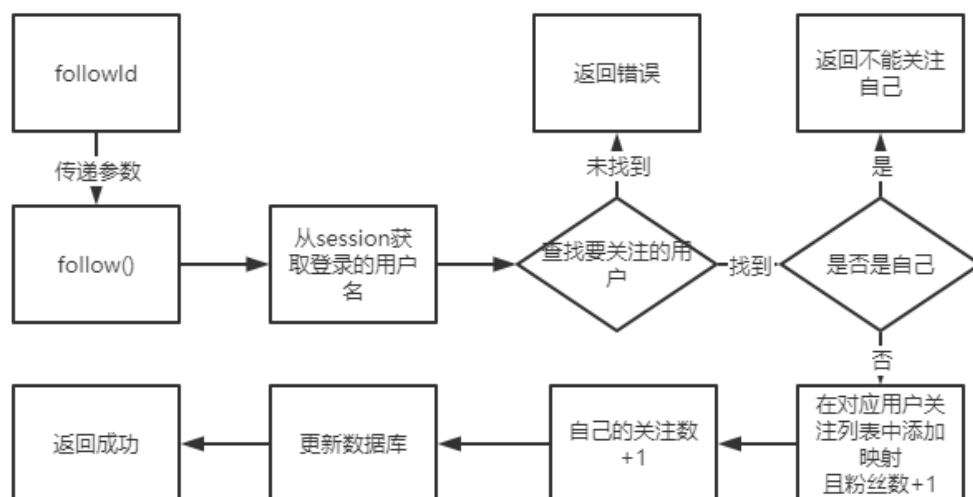
与微博列表主页面发表评论同理。

3. 点赞/取消赞



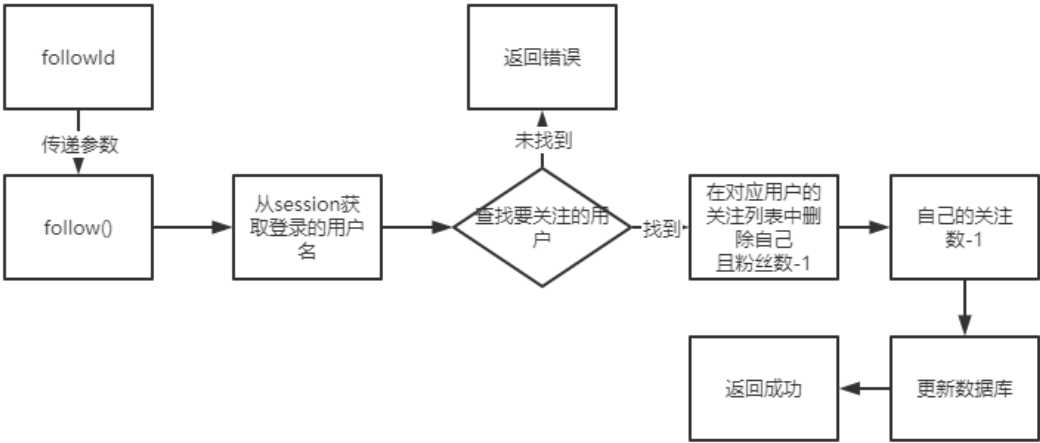
与微博列表主页面点赞同理。

4. 关注



该函数实现了关注功能，在关注之前先检查是否是自己，如果是，则不能关注。
当自己已经关注了该用户，则显示为取关按键。

5. 取关



与关注功能的原理类似。

5.2 接口设计

5.2.1 需求分析

需要前端信息通过接口与后端数据进行交互。前端将通过请求，以及 url 告知接口自己希望获得的数据，或者需要写入的数据。良好的接口可以时前端后端更易管理。

5.2.2 详细设计

5.2.2.1 用户信息核验

- 接口 URL: /loginCheck
- Request 方法: POST
- 接口功能: 用户信息核验并判断是否允许登录
- 请求参数:

参数名	必填	示例值	说明
username	是	admin	用户名
password	是	123456789	登录密码

- 返回 Response: (返回下列 String 之一)

参数名	类型	说明
success	String	登录成功
passwordError	String	密码错误

error	String	用户不存在
lock	String	账号被锁定
repetition	String	重复登录

5.2.2.1 注册表单提交

- 接口 URL: /logonCheck
- Request 方法: POST
- 接口功能: 用户注册
- 请求参数:

参数名	必填	示例值	说明
username	是	admin	用户名
password	是	123456789	登录密码
email	是	Xxx@163.com	邮箱

- 返回 Response: (返回下列 String 之一)

参数名	类型	说明
success	String	注册成功
emailError	String	邮箱已存在
usernameError	String	用户名已存在

5.2.2.3 主页微博列表（默认）

- 接口 URL: /index
- Request 方法: GET
- 接口功能: 微博默认按时间排序并分页
- 请求参数:

参数名	必填	默认值	说明
type	否	"1"	排序方法
start	否	"0"	
limit	否	"3"	

- Model 参数:

参数名	参数值	说明
isLogin	null "1"	null 未登录, 1 登录
username	null 用户名	
type	"1" "2" "3"	标注排序方式, 用于渲染对应视图
weibos	Page<Weibo>	可以进行分页显示

- 返回视图 index

5.2.2.4 微博排序显示

- 接口 URL: /index
- Request 方法: POST
- 接口功能: 将微博按一定方式进行排序
- 请求参数:

参数名	必填	默认值	说明
type	否	"1"	排序方法
start	否	"0"	
limit	否	"3"	

- Model 参数:

参数名	参数值	说明
isLogin	null "1"	null 未登录, 1 登录
username	null 用户名	
type	"1" "2" "3"	标注排序方式, 用于渲染对应视图
weibos	Page<Weibo>	可以进行分页显示

- 返回 thymleaf 的一个分段 index::weiboList

位置如下

```
<div class="container-fluid weibo row">
  <div th:id="weiboList" th:fragment="weiboList"...>
```

5.2.2.5 添加微博

- 接口 URL: /addWeibo
- Request 方法: POST
- 接口功能: 发微博, 在数据库中添加对应项
- 请求参数:

参数名	必填	示例值	说明
content	是		微博内容

- 返回 Response: "success"

5.2.2.6 添加图片

- 接口 URL: /addpic
- Request 方法: POST
- 接口功能: 添加图片
- 请求参数:

参数名	必填	示例值	说明
-----	----	-----	----

file	是	图片	MultipartFile
------	---	----	---------------

- 返回值：无

5.2.2.7 发表评论

- 接口 URL：/addComment
- Request 方法：POST
- 接口功能：发表评论，并将评论存入数据库
- 请求参数：

参数名	必填	示例值	说明
content	是		评论内容
Id	是	1	微博主键 id

- 返回 Response： " success"

5.2.2.8 点赞/取消赞

- 接口 URL：/like
- Request 方法：POST
- 接口功能：点赞，再点一次取消赞
- 请求参数：

参数名	必填	示例值	说明
weibold	是	1	微博主键 id

- 返回 Response： "success"

5.2.2.9 登出

- 接口 URL：/logout
- Request 方法：POST
- 接口功能：注销登录，清除对应 session 中的内容
- 请求参数：无
- 返回：无

5.2.2.10 显示个人主页

- 接口 URL：/my
- Request 方法：GET
- 接口功能：展示个人主页
- 请求参数：无
- Model 参数：

参数名	参数值	说明
isAttention	"0" "1"	0 未关注 1 已关注
weibos	List<Weibo>	微博列表
user	User	用户信息

- 返回视图 "my"

5.2.2.11 个人主页发评论

- 接口 URL: /my/addComment
- Request 方法: POST
- 接口功能: 发表评论, 并将评论存入数据库
- 请求参数:

参数名	必填	示例值	说明
content	是		评论内容
Id	是	1	微博主键 id

- 返回 Response: " success"

5.2.2.12 个人主页点赞/取消赞

- 接口 URL: /like
- Request 方法: POST
- 接口功能: 点赞, 再点一次取消赞
- 请求参数:

参数名	必填	示例值	说明
weibold	是	1	微博主键 id

- 返回 Response: "success"

5.2.2.13 关注

- 接口 URL: /my/follow
- Request 方法: POST
- 接口功能: 关注用户
- 请求参数:

参数名	必填	示例值	说明
followId	是	1	用户 id

- 返回 Response: (返回下面 String 之一)

参数名	类型	说明
success	String	成功

error	String	失败
self	String	不能关注自己

5.2.2.14 取消关注

- 接口 URL: /my/cancelFollow
- Request 方法: POST
- 接口功能: 取消关注
- 请求参数:

参数名	必填	示例值	说明
followId	是	1	用户 id

- 返回 Response: (返回下面 String 之一)

参数名	类型	说明
success	String	成功
error	String	失败

5.2.2.15 重置密码

- 接口 URL: /reset
- Request 方法: POST
- 接口功能: 重置密码
- 请求参数:

参数名	必填	示例值	说明
username	是	admin	用户名
password	是	123456789	登录密码
email	是	Xxx@163.com	邮箱

- 返回 Response: (返回下面 String 之一)

参数名	类型	说明
success	String	成功
error	String	失败
emailError	String	邮箱不匹配

5.3 数据库设计

5.3.1 数据库环境

使用 h2 数据库。

5.3.2 关系模式设计

5.3.2.1 用户

用户有两个实体，一个 UserInfo 用来存储简略信息，包括自身的主键 id，用户 id，用户名。

另一个为 User 实体，内容为主键 id，用户名，用户密码，邮箱，粉丝数，关注数，密码错误次数，锁定时间，还有一个一对多映射，由一个 User 对应多个 UserInfo，用于表示关注队列。

5.3.2.2 微博

微博实体 Weibo 包括一个主键 id，用户 id，微博内容，图片的 url，发布日期，点赞数，评论数。

两个一对多映射。

一个 Weibo 对应多个评论 Comment，用来表示该微博的评论。

一个 Weibo 对应多个用户 UserInfo，用来表示点赞的用户。

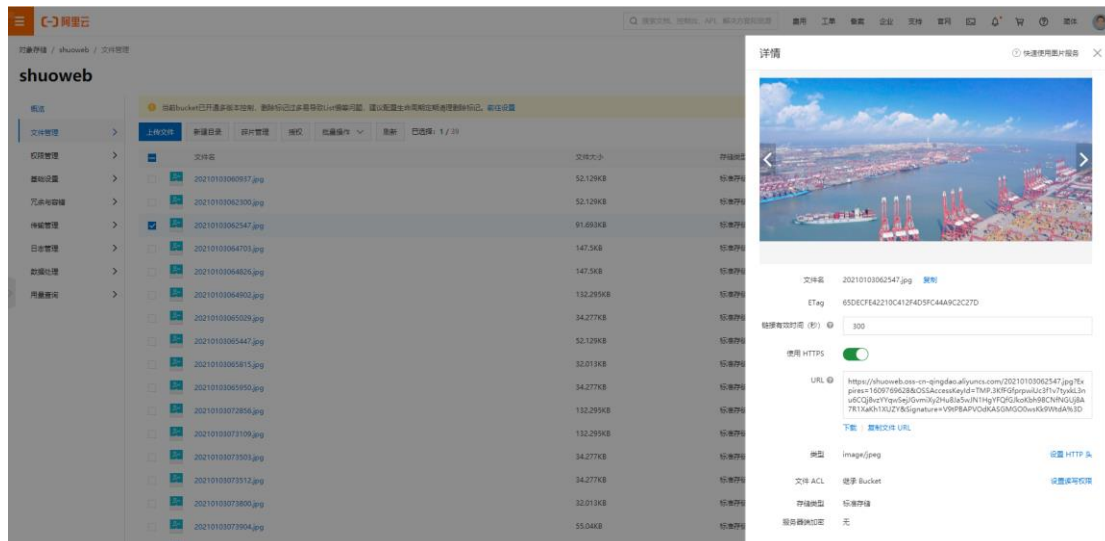
5.3.2.3 评论

评论实体 Comment 包括主键 id，对应微博 id，用户名，评论内容，评论日期。

5.3.3 图片的存储

由于 h2 数据库对图片的存储的效果不理想，因此我们选用阿里云的 OSS 来存储，实体中只需保存该图片的 url 便可以访问该图片。

Spring Boot 的依赖中也支持阿里巴巴的产品，因此实现起来也比较简单。



六、问题总结

6.1 李志毅

1. Bootstrap4.0 框架运用不熟练

第一次使用 Bootstrap，对于栅格系统和一些默认样式的运用不熟练，对于前端一些布局的设计不够熟练，导致设计和编写的页面对于自适应的支持度不高，页面只有在 PC 端样式和布局正常，在手机端等布局会发生变化。

2. ajax 运用不熟练

对于 ajax 请求的参数和携带内容，以及返回值或错误情况的处理有时候容易不知所措，尤其是 ajax 请求失败时，需要打开 chrome 调试网络并查看后端后才能大概知晓问题，有时候报错信息非常繁多，对于问题的筛查会比较麻烦。

3. jQuery 的使用问题

jQuery 的一些语法结构与原生 JavaScript 有一些差别，使用起来需要查询相应的语法后才能编写并调试。

4. Thymeleaf 的使用问题

对于一些复杂情况的 Thymeleaf 的使用不会，例如 Thymeleaf 元素中嵌套另一个元素，或如何在 jq 语句中获取 Model 的值，以及开发中发现若一些 Thymeleaf 元素设置不正确，会导致部分页面元素的样式失效，需要设置正确且语法正确的 Thymeleaf 格式。

6.2 陈硕

1. 对一些注解理解不够正确

对于某些注解只知道其作用，却不能正确使用。在后端代码编写的过程中，需要查阅资料才能知道其具体用法。

2. H2 数据库问题

对于映射关系使用不够熟练，在经过较长时间的调试后，才逐渐理解其原理，但仍有运用不恰当的问题。

3. 图片储存问题

一开始想将图片转化成字节流或者字符串，若是不对其进行拆分，其大小远大于 h2 String 的最大值。因此想到将 OSS 当做图床，数据库中只存储 url。

4. Thymeleaf 使用问题

页面部分刷新一开始不知道怎么实现，在查阅有关资料后，知道了 th:fragment 用法，后端只需要返回对应部分的视图即可。

七、完结感想

7.1 李志毅

在本次实验中通过一个小型微博系统的设计和开发，让我更加透彻地理解了 web 开发的概念，在本次大作业中基本采用前后端分离开发的模式，但虽然是这样，大家遇到问题时也会一起解决，并不能真正做到完全的前后端分离，大家会一起讨论出现的一个 bug，并找到是前端发送请求的问题？还是后端处理数据的问题？这都需要大家一起集思广益，共同探讨。

在前端的开发过程中，通过实际的开发对框架自适应、css3 样式、html5 标签、jQuery 语法、Thymeleaf 语法、ajax 请求、HTTP 请求等进行了深入理解。本次大作业中涉及到的很多未接触或接触不深的技术，通过实际的开发磨炼，让我对于这些技术从运用和操作更加娴熟，对相应的原理解也更加透彻。

在最后结合大作业需求、相关资料以及相应的文档和课程 ppt 指导，我们完成了最终的大作业，并出色的编写了文档，感谢我的两位队友和王老师，谢谢你们！

7.2 陈硕

通过本次大作业，我们实现了一个简单的微博系统的设计和开发，我也初步了解了一个 web 应用从设计到完成的整个过程。在这个过程中，通过和队友的讨论、调试，我们克服了一个又一个 bug，并不断加深对相关内容的认识和理解。我也明白了前后端分离开发模式的优势，并初步形成了前后端分离的开发思想。

在后端的开发过程中，我遇到了很多困难，但都在查阅相关文档后一一克服，同时还接触到了一些新的知识。通过本次大作业的锻炼，我的技术水平又进一步加深，对常用的一些技术的运用更加娴熟。

最后，感谢老师能在百忙中抽出时间对我们进行指导，并且在深夜为我们验收，老师您辛苦了！同时，也要感谢我的队友，大家都很棒！

7.3 杜政龙

学习 WEB 开发也有一段时间了，从 html 开始到 jdbc、servlet、jsp、div、css、javascript、

MVC 之类的各种 web 技术一点一点的不断学习。从一开始对 web 项目的没有兴趣和手足无措到现在开发中遇到问题时自己努力去攻克, 更多的时候是自己在一头猛扎下去对技术进行暴力“破解”, 少了一些思考, 却多了一些迷惑。通过这次大作业让我对这些内容进行了一遍系统性的梳理, 也让我更具备了软件工程的思想。