

PaperPass专业版检测报告

简明打印版

比对结果（相似度）：

总体：9 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：8 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

期刊库：5 %（期刊库相似度是指论文与学术期刊库的比对结果）

学位库：7 %（学位库相似度是指论文与学位论文库的比对结果）

会议库：2 %（会议库相似度是指论文与会议论文库的比对结果）

互联网：0 %（互联网相似度是指论文与互联网资源的比对结果）

编号：591F0C8D110F2S743

版本：专业版

标题：校园微博社交网络系统设计与实现

作者：慧锐

长度：13955 字符(不计空格)

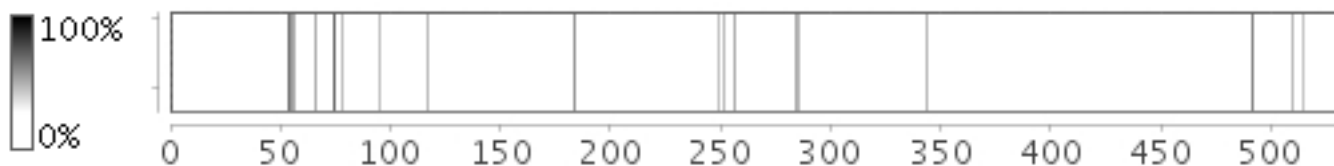
句子数：531句

时间：2017-5-19 23:17:33

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

1. 相似度：1 % 篇名：《基于Android手机平台的多微博整合登录系统的的设计和实现》

来源：学位论文 电子科技大学 2013 作者：姚宣屿

2. 相似度：1 % 篇名：《移动互联网:不只颠覆那么简单》

来源：学术期刊 《科学24小时》 2015年7期 作者：陈永东

互联网相似资源列表：

1. 相似度：1 % 标题：《慎用“微服务”架构 - 沙袋 - 博客园》

<http://www.cnblogs.com/gavntery/p/microservice-should-be-used-with-caution.html>

全文简明报告：

摘要

新浪微博无疑是一颗新星，从零到亿万用户量级的快速发展，平台架构几经扩展。 { 48 % : 而微博这种新的信息消费模式，也越来越让人接受。 } 一款专门的校园微博社交网络系统也就更加迫切需要。

本系统基于后端微服务和前端MVC框架实现了前后端解耦的、高可扩展的、高负载的、业务拆分的社交网络微博系统。前端用 AngularJs 框架处理页面渲染、页面数据显示逻辑、路由跳转逻辑，其更适用于构建 CRUD 应用的特性也克服了浏览器端 HTML 在构建 WEB 应用诸多方面的不足，让后端应用程序只需要关心于数据的处理和传输。后端用时下流行的微服务模式实现，其系统高稳定性、单一服务的简单易上手性契合了系统快速开发迭代的需求。微博应用场景下的大量消息消费和大用户量对系统数据库也产生很大的负荷， { 43 % : 因此使用数据库读写分离的技术让数据库的负荷分化，不让数据库成为整个系统的瓶颈。 }

构建校园微博社交网络系统时，本文模拟了系统的应用场景、学生使用场景，评估了时下互联网久经考验的技术，根据系统的特点由简到繁，自顶向下的设计了系统架构。用简单消息队列数据结构解决了微服务与微服务之间相互调用消息错乱的问题，创新性的用单页面应用模式开发了微博前端系统，让前端页面加载更加流畅、平滑。同时前后端解耦技术也可以让开发人员的协同工作更加容易、高效。

关键词： 微博； 社交网络； 校园； 微服务

Abstract

Sina , micro-blog is undoubtedly a new star , from zero to hundreds of millions of users of the rapid development of the scale , the platform architecture has expanded. And micro-blog this new mode of information consumption is becoming more and more acceptable. A special campus micro-blog social networking system is even more urgent need.

The system is based on the back- end micro services and front- end MVC framework to achieve the front and rear decoupling , highly scalable , high load , business split social networking micro- blog system. Use the AngularJs framework to deal with the rendering of the page , the page data display logic , logical characteristics of routing jump front , it is more suitable for the construction of CRUD application can overcome the browser of HTML in the construction of many aspects of WEB application , let the back- end application only need to concern in data processing and transmission. The back end is implemented with the current popular micro service model , which has high stability , simple service and easy chirality. It meets the requirement of rapid development and iteration of the system. A large number of news consumption and large amount of users of micro- blog application scenarios have a great load on the database system , so use the database to read and write separation technology to load differentiation database , that database has become the bottleneck of the whole system.

The construction of campus micro- blog social network system , this paper simulated the system application scenarios , students use scene , assessment of the Internet nowadays proven technology , according to the characteristics of the system by the simple , top- down design system architecture. With a simple message queue data structure to solve the micro and micro service service between the call message disorder problems , innovative use of single page applications developed micro- blog front- end system , make the front page more smooth , smooth loading. At the same time , the front end decoupling technology can make the collaborative work of developers easier and more efficient.

Keywords : Weibo; Campus; The Social Network; Microservice

1 绪论

1.1 引言

二十一世纪是属于互联网的世纪，从原始网络的出现到全球覆盖万维网，网络已经改变了人们的沟通方法，拓展了我们的社会结构。 { 47 % : 互联网也成为了人们日常中的一部分，丰富的网络应用让网民可以获得想要的信息。 } 从第一份电子邮件到网络即时通讯，越来越多的技术与创新刷新这人与人之间信息交流的模式。网络上小众个人Blog的兴起，到更加大众化的微博[]出现。通信技术的高速发展让用户能够通过多种方式随时随地使用WEB应用。相关WEB应用的技术也大跃前进。

{ 76 % : 根据2017年中国互联网络信息中心发布的第三十九次《中国互联网络发展状况统计报告》[]， } { 57 % : 截止2016年12月中国网民规模达7.31亿，互联网普及率为53.2%，其中微博应用的网民用户规模达到2.7143亿， } { 54 % : 使用率为37.1%，同时中国手机网民规模达6.95亿占比由2015年的90.1%提升至95.1%， } { 53 % : 手机微博的用户规模达到2.4086亿，使用率为34.6%，需要特别注意的是， } 从2011年开始互联网络信息中心把微博作为大类统计时微博用户规模只有6311万到如今的27143万。微博的高速发展也从另一个方面反应出微博是当下的潮流。

本文力图对比现有成熟微博应用新浪微博[]结合实际使用情况开发一种专为校园用户使用的校园微博社交网络系统， { 50 % : 给校园学生群体提供一个安全规范的微博交流平台。 } 结合校园网现状，用学生登陆校园网账号和学号双绑定模式限制用户群体。实现平台限制性。

国内微博产品虽然体量大但环境驳杂，而学生群体作为中国最大的群体之一，也是最应该保护的群体之一，应该健康安全的使用微博。同时学校应该更加关注学生的思想，让学生和老师们能够在一个平台下畅所欲言，愉快的交流，让学生们能够在一个良好的环境下学习、交流、交友和发微博。学校也可以与时俱进的用新兴模式参与到学生的日常生活中来。

{ 69 % : 最早的微博系统Twitter[]是由杰克·多西在2006年3月创办并在当年7月启动的。 } 而现在国内微博应用百花争鸣，各大厂商都有自己的系统架构实现。通过国内最大微博应用的架构师杨卫东的报告《构建高性能的微博系统——再谈新浪微博架构》[]， { 42 % : 新浪微博从第一版的LAMP架构到如今可扩展的微博系统，模块化，平台化分布式的微博系统， } 承载着海量的数据。

1.2 研究内容

{ 41 % : 本文研究了用微服务搭建微博类应用的方法。 } 微服务[]是一种软件架构风格，由Peter Rodgers博士于2005年度云端运算博览会提出的微Web服务(Micro-Web-Service)开始， { 100 % : 它是以专注于单一责任与功能的小型功能区块(Small Building Blocks)为基础，利用模组化的方式组合出复杂的大型应用程序， } 各功能区块使用与语言无关的API集相互通讯。

通过微服务的风格研究搭建系统后端服务，以模块化的方式研发服务，拆分业务。

{ 52 % : 前端尝试单页面应用三层架构模式 }

{ 43 % : 云端服务器生产部署技术, 让系统应用于实践。 }

1.3 论文结构

本论文研究基于微服务的具有高稳定性、单一服务简单易上手性的微博WEB应用。首先研究了本系统的相关工具与技术, 通过对实际使用情况的场景模拟细化系统设计, 用相关工具与技术实现系统各个微服务模块的功能。

第一章: 介绍了本论文结构, 微博系统的发展状况, 开发本系统过程中应用技术的研究内容以及论文内容安排等。

第二章: 介绍了基于第一章本系统所使用的主要技术和相关工具。前后端开发的解耦技术、微服务技术的应用、数据库方面的技术以及用Nginx搭配Tomcat实现反向代理[]和负载均衡[]来部署应用。

第三章: { 42 % : 模拟实际本系统的使用场景下系统功能和非功能需求, 完成本系统的需求分析。 } 用专业的用例图, 流程图描述具体需求。

第四章: { 42 % : 依托第三章的需求设计了本系统的总体架构设计, 页面展示设计, 数据库设计以及微服务功能设计。 }

第五章: { 46 % : 详细描述了本系统各个模块的具体实现、以及实现过程中出现的问题和解决方法。 }

第六章: { 50 % : 按照软件工程的测试流程, 对系统进行测试。 }

2 系统相关工具与技术研究

2.1 解耦技术

{ 40 % : 自从MVC开发模式的兴起, 在企业级应用方面前后端便仅仅的耦合在了一起。 } 这就造成后端服务器开发人员必须也懂得前端知识, 因为在 MVC模式下所有视图都是通过后端程序动态生成, 全栈工程师也随之出现, 但一个人精力有限不可能样样精通这就导致了 WEB页面难看, 操作困难等问题。为符合软件开发的模块化、具体化、流程化解耦必然是业界开发必经之路。

相比较于以往前端工程的发展 AngularJs[]创新性的用前端单页面形式, 仿照后端 MVC模式把前端工程抽象成应用和控制器模式, 通过[ng-view]标签对各个模块控制器进行显示处理, 极大地规范化了前端开发工作, 用内置% http函数解决了与后端数据传输问题, 实现了解耦。

{ 42 % : 本系统中多处使用到了JSON[]数据格式用来处理客户端与服务端、服务端与服务端等的数据传输。 } 其简单易懂, 易处理、易解析、易传输的特性非常适合于WEB服务间数据传输。

2.2 微服务的实验应用

微服务是时下最流行的开发模式。

Spring Boot[]是一系列Spring[]生态体系技术的整合，通过业界最佳开发工程师的多年实践为JAVA应用开发提供了合理快速开发技术。其快速搭建微服务的能力也是本文中服务端的搭建所使用的。

前后端解耦技术[7]的实现必然也造成了WEB应用安全性方面的挑战，有矛便有盾，随之出现了多种安全认证方式，oAuth2.0[]便是其中一种。因为后端采用开放API的形式开发，所以采用必要的安全认证方式来保证系统安全性。

Http[]协议从1.0到1.1最大的变化便是完善了 Http协议不支持状态转移的特性，随着1.1版本的全面普及，RestFul[]架构也随之兴起，对新 Http协议的特性也能充分利用。

2.3 读写分离

{ 43 % : 技术总是服务于解决问题，随着 WEB应用的使用，随着出现了数据库超压问题， } { 43 % : 考虑到本系统的应用场景，为了解决数据库瓶颈问题，采取数据库主从库设计， } { 48 % : 通过数据库读写操作的分离，从原理上分压数据库负荷。 }

2.4 Nginx搭配Tomcat

对一个WEB应用来说，网页的访问速度直接影响着使用者对WEB应用的第一映像，有研究表明任何应用的初始加载时间在两秒以下是使用者的最低容忍时间，{ 42 % : 如果时间超过两秒使用者就会失去耐心而离开。 } 为了能够给学生一个良好的用户体验加快WEB程序访问速度本系统采用配置反向代理服务器，通过对使用者请求资源的缓存加快响应速度。

Nginx[]本是一种PHP程序的服务器软件，但因其性能效率极高常常被用作为代理服务器软件。{ 59 % : 本系统实施反向代理服务器便采用Nginx。 }

2.5 Feed流

为了能有效实施给学生推送持续更新的微博消息、好友私信等功能，结合本系统初始版本需求，采用了Feed流[]的设计方案。在一个WEB微博与社交系统中最基本的功能就是用户关注人所发微博以及好友私信内容的实时更新。

2.6 本章小结

本章节对本系统用到的前后端解耦[]、微服务技术、服务器部署技术、Feed流设计进行了简要的叙述。

3 需求分析

{ 40 % : 准确的需求分析对软件开发的巨大作用已经在大量实践中得到证实。 } 本系统在需求阶段为了能够准确描述需求分析对系统应用场景进行了模拟实验。

3.1 需求综述

本文试图对微博系统、社交网络的研究，基于校园环境特性实现一个校园微博社交网络系统，初始版本在功能上要具有基本的学生登录注册，发微博和评论，好友私信功能，在性能上要实现准确的消息推送，较高的并发访问，页面加载流畅。考虑到系统用户隐私性，数据量较大以及数据的安全性在系统管理功能上采用服务器本地程序自动化管理实现。

3.2 用户说明

本系统面向广大校园师生，为了营造一个师生和谐互动，轻松便捷的微博社交系统，系统师生用户使用权限以及内容发布权限上平等对待，师生可以友好交流加好友沟通学习问题。学生与学生、老师与老师、学生与老师都可以友好互动，学生可以关注自己喜欢的老师，向老师提出学术问题。老师可以关注自己的学生留意学生的学习生活动态。

3.3 业务分析

1、本系统作为一个内容发布与消费的平台，业务流程相对简单。

{ 51 % : 2、模拟学生登录微博平台使用流程， }

- (1) 学生输入系统登录地址；
- (2) 已有账号登录，无则注册；
- (3) 登录成功会自动刷新出最新关注人微博Feed流；
- (4) 平行功能操作：
 - 手动刷新微博；
 - 评论微博；
 - 发微博；
 - 搜索新用户；
 - 查看我的微博；
 - 收发好友私信；
 - 不良信息举报；
- (5) 退出系统；

3、系统基础流程图如图3.1：

图3.1 系统基础流程图

3.4 功能结构

有如下几个模块：

师生注册： 师生必须通过学号或者工号注册账号；

师生登录： 师生登录后可以使用系统其他功能；

微博收发评论： { 52 %： 登陆后可以查看好友最近微博，评论微博； }

添加好友： 通过搜索其他同学的学号或老师的工号来添加好友关注；

好友私信： 可以对所有互为好友的好友发送私信；

不良信息举报： 在使用本系统过程中发现的不良信息进行举报处理；

{ 54 %： 系统角色只有注册用户和未注册用户，总用例图如图3.2： }

图3.2 总用例图

本系统主要面向已注册用户，整个系统功能也是围绕已注册用户来展开，用例图如图3.3：

图3.3 注册用户用例图

用户用例图如图3.4：

图3.4 用户用例图

表3.1和表3.2描述了用户用例。

表3.1 用户注册用例描述

用例名称用户注册

{ 67 %： 用例描述未注册用户进行注册 }

用例对象未注册用户

前置条件无

后置条件成功： { 43 %： 提示用户成功注册，数据库中用户表更新，跳转登陆页面 }

失败： 提示出错信息

基本事件流1点击注册

2用户根据要求输入各项信息

3点击注册

备选事件流注册失败提示出错信息并刷新注册页面

表3.2 用户登录用例描述

用例名称用户登录

用例概述用户进行登录

用例对象已经成功注册的用户

前置条件用户已注册

后置条件成功： 跳转至微博首页

失败： { 58 %： 提示出错信息（用户名、密码、验证码错误等） }

基本事件流1用户访问首页

2用户根据要求在相应位置输入信息

3点击登录

{ 53 %： 备选事件流登录失败提示出错信息并刷新登录页面 }

3.5 版本规划

{ 42 %： 本文中系统初始为0.0.1版本，实现系统基础微博收发，好友管理等功能。 } 在经过整体测试流程验收成功后定为1.0.0版本，此后可在此版本基础上进行继续开发。

3.6 功能需求分析

{ 43 %： 本文中对功能的优先级定了等级，分为1、2、3级，其中1级优先级最高。 }

1、微博浏览发送功能

此功能是学生用户在使用本系统，发送微博消息，查看他人微博消息的功能。 优先级为1级，是系统核心功

能。在发送微博上内容数据长度上设置了140字的限制长度。界面要求直观易用对数据限制有正确提示。使用本功能的前置条件是用户成功登陆。

2、微博评论功能

{ 55 % : 此功能是用户对他人微博发起评论的功能。 } 优先级为1级。评论内容数据长度同样设置了140字的限制。页面要求同发微博功能。前置条件为用户成功登陆。

3、好友功能

此功能是对本系统社交网络要求的基本功能。优先级为1级。要求界面能够准确美观显示好友列表。前置条件为用户成功登陆。

4、不良信息举报功能

此功能是对系统中不良微博举报处理。优先级为2级。要求能正确引导用户正确举报，对举报做基本分类。前置条件为用户成功登陆。

3.7 系统使用软硬件环境说明

本WEB系统支持Chrome、Firefox、Edge各版本浏览器，不支持IE系列浏览器，并且不对其他浏览器做预期支持，可根据实际情况而定。因采用Bootstrap[]框架渲染WEB页面，本系统同时支持跨端访问，在安卓、IOS、PC端都可以正常使用。浏览器限制参考对PC端限制。

4系统设计

{ 48 % : 本章依照需求分析对系统的架构、功能、数据库进行了具体设计。 }

4.1 架构设计

本系统在设计之初考虑使用SSH开发框架构建经典的巨石型MVC应用，即将所有的功能模块放在一个WEB容器中去运行。如图4.1：

图4.1 巨石应用架构图

依照如今类似微博社交系统的发展现状，这种经典的架构并不能支撑起微博类系统的诸多特性，以及需要面对的诸多挑战，例如海量的数据处理、即时的业务反应、高并发访问、用户对访问速度的苛刻要求等。考虑到一个系统在业务和数据大规模上升时候通常的解决思路便是加服务器、数据库分库分表、功能分解这三个方向。结合本文实际情况选择了功能分解的方法，将不同功能拆分为单独的服务，并且在多个服务API的管理上使用API gateway的方式集中管理具体的服务API，{ 42 % : 让WEB前端开发人员不用关心与具体服务的变更维护等工作。 } { 43 % : 结合一些常见的服务器和数据库技术设计出如图4.2的架构模型。 }

图4.2 架构图

{ 42 % : 用户通过浏览器访问本系统, 用户请求会先被反向代理服务器处理, 如果用户请求的数据在反向代理服务器中存在缓存, } { 46 % : 反向代理服务器会直接返回请求的数据, 请求到达负载均衡层后根据负载均衡算法分配处理服务器, } 从而返回响应结果。

4.2 功能设计

从需求设计出发, 本系统中对于用户类别没有做显示区分, 无论是学生用户还是老师用户, 在系统角色上只有已注册用户和未注册用户之分。 { 41 % : 已注册用户可以设置个人信息、发布校园微博、删除本人微博、浏览他人微博、评论微博、搜索好友、添加好友、不良信息举报等功能。 } { 68 % : 主要分为好友模块、微博模块、用户模块这三大模块。 } 结构图如图4.3 :

图4.3 功能结构图

4.3 数据库设计

{ 48 % : 在实际的项目中, 数据库的具体实施在整个系统中起着至关重要的作用。 } 并且在项目运行期间出故障最多的模块便是数据库方面, 因此本文在数据库设计方面结合实际经验对数据库的设计拟定几条规则 :

编码设置必须为UTF8MB4字符集。 说明 : UTF8MB4字符集是UTF8字符集的超集, 本系统作为微博社交类系统, 并且在中文使用环境下需要对汉字和emoji表情提供完美支持。

不使用存储过程、视图、触发器。 说明 : 实际情况实际对待, 考虑到本系统是一个具有 “并发量大, 数据量大” 特性的互联网项目, 因此此类项目的重点在于系统的吞吐量和性能方面, 对数据库性能负荷比较大的操作禁止使用。 把系统的复杂逻辑计算放到服务层去处理。

不使用外键。 说明 : 完整性约束可以在服务端用应用程序控制。

4.3.1 概念设计

本系统在角色上没有做区分, 本着畅所欲言, 和谐交流的精神没有设计管理员角色, 并且管理员角色的实际使用效率和效果并不突出。 但在系统中总会出现不健康不和谐的内容。 因此在程序和人工举报上做了设置。 { 45 % : 用户在成功登陆系统后可以发布微博、浏览微博、评论微博、添加好友。 } 对结构进行抽象后画出系统E-R图如图4.4 :

图4.4 E-R图

实体如下 :

{ 52 % : 用户实体包括账号、密码、昵称、用户状态、学号、头像路径、性别、学校。 }

{ 45 % : 微博实体包括内容、时间、所属用户、微博状态。 }

{ 44 % : 评论实体包括内容、时间、所属微博、所属用户。 }

好友实体包括用户ID、好友ID、好友头像。

4.3.2 表设计

所有关系表如表4.1：

表4.1 信息表

{ 58 % : 序号逻辑表名物理表名说明 }

1用户表c_user存储用户信息

2微博表c_weibo存储微博信息

3评论表c_reply存储评论信息

4好友表c_friend存储好友关系信息

用户表

表4.2 用户表

物理字段类型规则说明

IdUUID主键用户ID，128位UUID具有全局唯一性

schoolcodeVarchar（60）非空用户名，使用学号作为用户名，唯一

PasswordVarchar（60）非空用户密码，存储经过MD5加密的密码

NicknameVarchar（60）非空用户网名

imageUrlVarchar（1024）可空保存用户头像的路径

SexVarchar（4）可空用户性别

SchoolVarchar（20）可空用户学校名

StatusInt可空标示用户状态，0： 正常 1： 锁定

微博表

表4.3 微博表

物理字段类型规则说明

IdBigInteger主键微博Id，每条微博的唯一标示

TextText 非空微博内容，JSON格式数据，如{“data”：“内容”}

UidUUID非空微博发布人ID

CreatedateDatetime非空微博发布时间

StatusInt可空微博状态标示，0： 正常 1： 用户删除 2： 系统删除

评论表

表4.4 评论表

物理字段类型规则说明

IdBigInteger主键评论Id，每条评论的唯一标示

TextText 非空评论内容，JSON格式数据，如{“data”：“内容”}

UidUUID非空评论人ID

WidBigInteger非空所属微博ID

CreatedateDatetime非空评论发布时间

StatusInt可空评论状态标示，0： 正常 1： 用户删除 2： 系统删除

好友表

表4.5 好友表

物理字段类型规则说明

IdBigInteger主键好友信息Id，唯一字段，自增。

UidUUID非空用户ID。

FUidUUID非空好友ID

FimgVarchar非空好友头像路径

4.4 类设计

1、用户模块的主要类的设计如图5.5：

图5.5 用户服务类图

{ 55 % : MyUser类是用户的实体类，封装了用户的基本信息和方法。 }

MyUserRepository类是数据访问层对用户信息访问的接口类。

LoginService和MyUserService分别实现具体的数据处理流程。

LoginResource和MyUserResource分别实现了服务的RestFul接口。

2、微博模块类结构如图5.6：

图5.6 微博类图

WeiboResource类中实现微博操作的所有接口。 { 62 % : 服务只关心于数据的正确性和有效性。 }

WeiboService和ReplyService类分别实现具体的业务方法。

4.5项目结构

因为使用前后端分离技术，本项目分为前端工程和后端工程，前端使用WebStorm[]工具搭建工程，用npm管理项目依赖。 前端结构如图4.5：

图4.5 前端工程结构

{ 47 % : Config目录下为Angular框架的相关配置与项目数据处理逻辑JS文件。 } face下为表情文件夹，assets下为项目自定义CSS文件。

后端工程如图4.6：

图4.6 后端工程结构图

项目为Maven[]工程，使用IntelliJ IDEA[]工具，整个项目的依赖和Maven配置都在pom.xml文件。 项目结构是用Jhipster生成。 主要项目代码在java目录下，项目相关的Liquibase配置、Spring boot配置、国际化配置、日志配置都在resources目录下。

5 系统实现

本章结合之前的工作逐步的实现系统各个方面的功能

5.1 前端工程与CSS渲染

本系统的前端工程页面使用 Bootstrap框架对页面进行渲染，使用 AngularJs框架搭建前端工程，整个工程使用单页面形式，即只有一个 HTML页面，其中各个模块的页面分别为[body]中的[div]。通过AngularJS框架的 ngRoute库提供的路由支持，[ng-view]标签对各个页面进行控制。代码如下：

```
[body]

[ng-view]/[ng-view]

[script src= " node_modules/angular/angular.js " ][/script]

[script src= " node_modules/jquery/dist/jquery.js " ][/script]

[script src= " node_modules/angular-animate/angular-animate.js " ][/script]

[script src= " node_modules/angular-aria/angular-aria.js " ][/script]

[script src= " node_modules/angular-route/angular-route.js " ][/script]

[script src= " node_modules/angular-material/angular-material.js " ][/script]

[script src= " node_modules/bootstrap/dist/js/bootstrap.min.js " ][/script]

[script src= " assets/js/jquery.backstretch.min.js " ][/script]

[! --ng模块配置--]

[script src= " config/app.js " ][/script]

[script src= " config/config.js " ][/script]

[! --自定义JS--]

[script language= " javascript " src= " config/CustomerIndex.js " ][/script]

[script language= " javascript " src= " config/trim.js " ][/script]

[/body]
```

首页效果如图5.1：

图5.1 首页效果图

而注册页面通过首页弹出模态框实现，效果如图5.2：

图5.2 注册效果图

登陆后进入微博页，显示用户关注人的最新动态，在微博页用户可以看到个人信息，还可以搜索好友，添加好友，进入个人信息设置页，进入好友页，还可以举报不良信息。效果如图5.3：

图5.3 微博页效果图

点击个人头像会进入用户微博页，显示用户所有微博，效果如图5.4：

图5.4 个人微博页效果图

5.2 微服务实现

前端作为独立工程，已经搭好了页面渲染和数据展示框架，但没有具体的数据，所以前端工程只关心于获取正确的数据然后展示数据，而正确的数据就来源于后端的微服务系统。通过对系统的业务拆分，业务模块化，通过多种微服务的模式组合整体系统的架构快速迭代式的完成系统开发。

5.2.1 用户服务

顾名思义用户服务便是针对系统用户的业务程序而组成的服务。 { 41 % : 整个结构分为三层，domain实体层、Service服务层、Resource资源控制层。 } Domain实体层服务定义数据实体，无论是DAO、还是DTO都可通过扩展domain层类来实现。Service层用来编写具体业务逻辑。而Resource层用来编写对资源的具体控制以及开放接口。LoginResource开放了用户的登陆的API接口并且只对用户登陆的URL做出响应，同时通过编写Filter类对URL访问进行控制过滤。LoginFilter核心代码如下：

```
//获取访问的url

String accessUrl = httpRequest.getRequestURI();

logger.info( " access url is {} " , accessUrl);

//如果是访问登录的url，就继续访问

if (this.loginUrl.equals(accessUrl)) {

logger.info( " =====这是登陆url，不过滤，继续登陆 " );

filterChain.doFilter(servletRequest , servletResponse);

return;

//如果是注册url，就继续注册
```

```
if (this.registerUrl.equals(accessUrl)){

logger.info( " =====这是注册URL , 不过滤 , 继续注册. " );

filterChain.doFilter(servletRequest , servletResponse);

return;

//如果都不是判断是否已经登录

if (! StringUtils.hasText(userIdStr)) {

logger.info( " not find userId in header params " );

servletResponse.setContentType( " application/json; charset=utf-8 " );

servletResponse.getWriter().write(JsonUtil.toJson(new

return;

if (session.getAttribute(Constants.SESSION_STORE_USERINFO_KEY) == null) {

logger.info( " not find userInfo in session , so return back! " );

servletResponse.setContentType( " application/json; charset=utf-8 " );

servletResponse.getWriter().write(JsonUtil.toJson(new

return;
```

5.2.2 微博服务

微博服务在结构上同用户服务但在具体细节上有区别。 因为两个服务的拆分，对用户唯一性以及安全认证造成了困难，通过用OAuth2.0的认证机制对用户身份进行认证。 { 58 % : 对其微博数据进行有效访问。 }

这部分分别有微博和评论实体类，它们定义了微博和评论的数据结构。 通过各自不同的Service服务实现具体的业务逻辑。 { 40 % : 在WeiboResource类里面对这些业务资源进行控制，并传给客户端浏览器。 } WeiboService类定义了微博实体的发布，修改，删除，查询的操作方法，以供 WeiboResource类完成具体的业务逻辑， { 47 % : 例如用户刷新首页查看最新微博，用户发布微博，用户查看个人微博。 } Reply类定义了评论实体的操作方法。 供WeiboResource类使用，例如用户的发评论操作，查看评论的操作。

5.2.3 好友服务

对一个社交系统来说好友私信是必不可少的功能，本系统在构建好友私信实时通信功能时使用了GoEasy[]云服务，方便快捷的构建好友在线私信聊天功能。 实现代码如下：

```
[script type= " text/javascript " src= " http(s): //cdn.goeasy.io/goeasy.js " ][/script]
```

```
[script type= " text/javascript " ]
```

```
var goEasy = new GoEasy({
```

```
  appkey: ' BC-14eff6e3cb5e4af693c419d44f3f95b7 '
```

```
});
```

```
[/script]
```

6 系统测试

6.1 测试环境

测试环境1：

PC： dell inspiron 13笔记本

屏幕尺寸： 1366 × 768

操作系统： windows 10 专业版

CPU： i5-3337U

内存容量： 8G

浏览器： chrome、Firefox

测试环境2：

IOS手机： iphone6

屏幕尺寸： 4.7英寸

操作系统： IOS10.1

RAM： 1G

ROM： 16G

浏览器： safari

6.2 测试内容

主要对系统跨平台兼容性，功能的完成性，以及功能是否能够达到需求标准进行黑盒测试。

6.2.1 测试用例和结果

功能测试： { 55 %：对系统的基本功能进行逻辑功能测试和兼容性测试 }

{ 61 %：逻辑功能测试采用等价类划分法。 } 测试用例如表6.1：

表6.1 用例表

{ 53 %：测试功能测试内容预期结果测试结果是否通过 }

登陆输入正确账号密码成功，进入微博首页登陆成功，进入微博页是

{ 58 %：登陆输入错误账号密码弹出提示信息： } 账号或密码错误弹出提示信息： 账号或密码错误是

{ 49 %：登陆账户或密码为空点击登录弹出提示框： } 不能为空弹出提示框： 不能为空是

{ 46 %：注册点击注册按钮弹出注册模态框弹出注册模态框是 }

注册输入合法注册信息退出模态框状态，可以登陆退出模态框状态，可以登陆是

{ 45 %：注册输入不合法注册信息提示注册信息不合法注册失败提示相应信息是 }

发布微博输入合法字长微博内容发布成功，关注用户可以查看到微博关注用户能看到此条微博是

发布微博输入不合法字长微博提示发布不成功提示发布不成功是

发布微博不输入内容发布微博提示发布不成功发布不成功是

发表评论输入任意评论内容在微博下可以查看到评论内容可以查看到评论内容是

{ 46 %：搜索好友输入搜索内容显示相应搜索结果显示正常是 }

{ 41 %：添加好友添加搜索到的好友添加成功，可以在好友页面看到添加的好友添加好友存在是 }

好友私信给好友发私信相应用户能收到消息好友收到了消息是

{ 41 %：个人信息修改进去个人信息页，修改信息修改成功，显示修改后结果显示修改后结果是 }

性能测试： { 50 %：对系统响应时间以及服务器压力符合需求 }

系统响应时间测试结果如图6.1：

图6.1 响应时间测试结果图

模拟系统登录压力测试结果如图6.2：

图6.2 压力测试结果图

6.3 测试结论

本系统是一个具有基本微博收发，好友私信功能的微博社交网络系统。对系统的用户登录，用户管理，微博功能，评论功能等测试后基本确认了系统的有效性和完整性，初步完成了设计效果。

总结

对本系统的开发，深刻的体会到一个中大型WEB系统各项工作的具体流程和需要注意的细节。对一个系统构建过程中需要的技术选择以及技术运用有了深刻的理解。当下流行的微服务Spring Boot、前端AngularJS框架等通过实践检验的技术也更加熟练。随着现如今网络和终端的发展一个系统要同时满足安卓端、IOS端、windows phone端、PC端才能给本系统使用者提供更好的使用体验。而H5和Http1.1的出现非常完美的解决了跨端的问题，一个产品或者系统不用再通过多个端的专门团队去分开实现相应端的实现。大大的节省了开发成本。也使系统更加简单化规范化。