|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | |
| 50da81cb39dbb6fdf8b8fff80b24ab18962b3708 | | |  | | | | ○ **A** **基础理论**  ● **B** **应用研究**  ○ **C** **调查报告**  ○ **D** **其他** | | | | | |
|  | | **岭南师范学院** | | | | | |  | | | | |
|  | **本科生毕业论文（设计）** | | | | | | | | | |  | |
|  | | | | | | | | | | | | |
|  | **校园发布助手的设计和开发** | | | | | | | | | | |  |
|  | | | | | | | | | | | | |
|  | | | | **二级学院** | **：** | **信息工程学院** | | | |  | | |
| **专 业** | **：** | **计算机科学与技术 （软件服务外包）** | | | |
| **年 级** | **：** | **2015级** | | | |
| **学 号** | **：** | **2015874136** | | | |
| **作者姓名** | **：** | **李田锋** | | | |
| **企业导师** | **：** | **软件工程师 职称** | | | |
| **学院导师 ： 教授 职称** | | | | | | | | | | | | |
|  | | | | **完成日期** | **：** | **2018年5 月12日** | | |  | | | |
|  | | | | | | | | | | | | |

**校园发布助手的设计与开发**

专业名称：计算机科学与技术

作者姓名：李田锋

企业导师：杨海波

学院导师：杨俊杰

**论文答辩小组**

**组长**： 吴涛

**成员**： 杨俊杰

刘强峰

**论文成绩：**

**目 录**

[1 前言 2](#_Toc6343756)

[1.1 系统开发目标及范围 2](#_Toc6343757)

[1.1.1 系统开发目标 2](#_Toc6343758)

[1.1.2 系统开发范围 3](#_Toc6343759)

[1.2 系统开发工具及应用介绍 3](#_Toc6343760)

[1.2.1 开发工具IntelliJ IDEA概述 3](#_Toc6343761)

[1.2.2 Java概述 3](#_Toc6343762)

[1.2.3 Spring Boot概述 4](#_Toc6343763)

[1.2.4 Ngnix概述 4](#_Toc6343764)

[1.2.5 Redission概述 5](#_Toc6343765)

[1.2.6 Redis概述 6](#_Toc6343766)

[1.2.7 SeaweedFs概述 6](#_Toc6343767)

[1.2.8 T-io概述 6](#_Toc6343768)

[1.2.9 Docker概述 7](#_Toc6343769)

[1.2.10 Node.js概述 8](#_Toc6343770)

[1.2.11 Nerv.js概述 8](#_Toc6343771)

[1.2.12 MySQL概述 8](#_Toc6343772)

[2　系统的需求分析 9](#_Toc6343773)

[2.1 高校互联应用的现状和存在的问题 9](#_Toc6343774)

[2.2 校园发布助手的研究意义 10](#_Toc6343775)

[3　系统的总体设计 10](#_Toc6343776)

[3.1 系统的功能简介 10](#_Toc6343777)

[3.1.1 校园模块 11](#_Toc6343778)

[3.1.2 机构模块 11](#_Toc6343779)

[3.1.3 通讯模块 12](#_Toc6343780)

[3.1.4 个人模块 12](#_Toc6343781)

[3.2 系统的结构分析 12](#_Toc6343782)

[3.2.1 系统用例分析 12](#_Toc6343783)

[3.2.2 系统功能结构分析 16](#_Toc6343784)

[3.3 系统部分需求及关键业务分析 17](#_Toc6343785)

[3.3.1 移动端登录 17](#_Toc6343786)

[3.3.2 校园资讯浏览 18](#_Toc6343787)

[3.3.3 校园资讯发布 19](#_Toc6343788)

[3.3.4一对一即时通讯 20](#_Toc6343789)

[3.3.5 即时通讯集群 21](#_Toc6343790)

[3.3.6 组织机构创建并添加成员 22](#_Toc6343791)

[3.4 系统的关系型数据库设计 23](#_Toc6343792)

[3.4.1 关系型数据库概念设计 23](#_Toc6343793)

[3.4.2 关系型数据库逻辑设计 24](#_Toc6343794)

[4　系统的具体实现 32](#_Toc6343795)

[4.1 即时通讯消息格式定义 32](#_Toc6343796)

[4.1.1 移动端登录消息格式 32](#_Toc6343797)

[4.1.2 移动端聊天消息格式 33](#_Toc6343798)

[4.1.3 移动端已读聊天消息格式 34](#_Toc6343799)

[4.1.4 移动端获取聊天消息格式 34](#_Toc6343800)

[4.2 部分需求及关键业务实现 35](#_Toc6343801)

[4.2.1 移动端登录 35](#_Toc6343802)

[4.2.2 校园资讯浏览 37](#_Toc6343803)

[4.2.3一对一即时通讯 39](#_Toc6343804)

[4.2.4 即时通讯集群 41](#_Toc6343805)

[5 问题总结与分析 42](#_Toc6343806)

[6 结论 45](#_Toc6343807)

**校园发布助手的设计与开发**

作者　李田锋　　指导导师 杨海波 杨俊杰

(岭南师范学院信息工程学院，湛江 524048)

**摘　要：**本系统旨在提高校内消息传达的质量和效率以及提高校内各大学生群体的沟通和办事效率，便于学生群体的交流。系统采用B/S结构，前端基于前端工程化的概念，运用Node.js、npm、Nerv等工具或框架辅助前端开发；后台则基于Java语言，运用Spring boot、T-io、Mybatis等框架，依赖于SPI机制实现服务发现。持久化及优化方面则采用了Mysql和Redis进行信息的存储。系统提供了校园活动浏览和发布、校园活动FAQ、消息收藏、机构信息管理、个人与个人，机构内标注消息阅读状态的即时通讯等功能，原生支持即时通讯服务器、文件服务器的集群部署。系统志在时校园消息更广泛的渗透到校园的各个领域，为校内各大师生提供一个统一可靠的信息交流平台。

**关键词：**校园；消息发布；即时通讯；集群；Java；Redis；

**Design and Development of a Residential Property Management System**

LI Tianfeng

School of Information Engineering, Lingnan Normal University, ZhanJiang, 524048 China

**Abstract:** The purpose of this system is to improve the quality and efficiency of intramural information transmission, as well as the communication and work efficiency of College students, so as to facilitate the exchange of students. The system adopts B/S structure, and the front-end is based on the concept of front-end engineering, using Nodejs, Npm, Nerv and other tools or frameworks to assist the front-end development; the back-end is based on Java language, using Spring boot, T-io, Mybatis and other frameworks, relying on SPI mechanism to achieve service discovery. Mysql and Redis are used for information storage in persistence and optimization. The system provides the functions of browsing and publishing campus activities, FAQ campus activities, message collection, organization information management, instant messaging from person to organization, with message marked with reading status during the exchange. It supports the cluster deployment of instant messaging servers and file servers. The system aims to provide a unified and reliable information exchange platform for teachers and students.

**Key words:** Campus; Message publishment; Instant messaging; Cluster deployment; Java; Redis

# 1 前言

随着计算机科学技术的飞速发展，校园数字化的浪潮势不可档。在信息化的时代下，各大高校抓紧校园信息化建设，不断丰富校园文化生活，推动教学质量、科研水平和管理效率的提高。特别是在迎来微信公众号、小程序的爆发式发展的前提下，校园级服务应用更加进入学生们的视野。开发校园发布助手类型软件在于方便师生的在校的日常生活，减少没有必要的精力付出。当前校园虽有相关程序满足师生对校园信息的查阅，但对于消息的传递处理几乎处于开始阶段，各大机构只能通过社交软件依次进行“口头”传达，对消息的精准投放没有一定的保障性，而本系统致力于提升校园级应用的服务和质量，从校园工作出发，基于微信公众号的设计角度，将企业级即时通讯软件钉钉设计理念带进校园，统一消息的管理和分发，降低消息生命周期成本，使消息可以低门槛在校园传播，同时提升校园信息化程度，在校园网络全面覆盖的前提下，满足校园工作对消息可达程度进行掌握的实时通讯需求，提高校园工作的效率和质量，系统化、结构化校园工作信息，用信息技术给校园工作和生活优化增速。开发本软件旨在建立面向校园、面向师生的统一机构门面，满足校园内消息通知的需求。

## 1.1 系统开发目标及范围

### 1.1.1 系统开发目标

建立健全的消息传递系统，将高校校园生活和工作分离，在校园消息传播过程中采用实名制的前提下，满足校园学生工作管理要求。在业务目标下，移动H5端和微信小程序端提供消息的、提问、搜索、浏览和收藏等功能，满足校园消息发布者和消息的目的人群对消息的掌控需求；搭建校园内各个组织的官方账号，允许组织内发布能够在一定范围内进行消息的发布，并享有组织内部享有满足工作状态的即时通讯功能，同时组织对外则是提供组织通讯录功能，满足有需要的师生能够及时找到需要联系到的该组织内的人员，必要时也可以提供给某机构提供意见反馈，以提升某组织的自身的管理质量。在技术目标下，在满足校园师生的日常使用情况下，实现前端和后台组件化要求，提高代码的灵活性、复用性和可维护性；同时为了适用于局域网部署，减少对外部网络的依赖；强调系统中各个组件的解耦程度，系统的主要服务能够分离部署；原生支持即时通讯服务的集群部署，实现即时通讯的高可用、高性能、低延迟的响应。

### 1.1.2 系统开发范围

本系统是为校园日常工作交流的而定制的，系统的参与者有学生、教师、校内机构，主要聚焦于学生群体，因此不涉足于校外机构参与到该系统的互动中；系统后端由Java语言编写，将SeaWeedfs作为系统的文件存储服务器，前端主要由JavaScript、HTML、CSS编写。利用Mysql和Redis数据库来进行数据的中转和存储。面向的客户端有H5移动端和小程序端。

## **1.2 系统开发工具及应用介绍**

### 1.2.1 开发工具IntelliJ IDEA概述

IDEA是一款以Java语言编写，由JetBrains公司开发而成的集成开发环境。IDEA作为一款常用的Java开发工具，它以精美的用户界面、多样化的开发插件而著称，并提供了人性化的多语言代码提示功能、精心定制的代码导航功能、优秀的代码重构等功能，相对于Java开发环境Eclipse，IDEA更能减少程序开发人员的重复性工作，提高编程效率和质量。

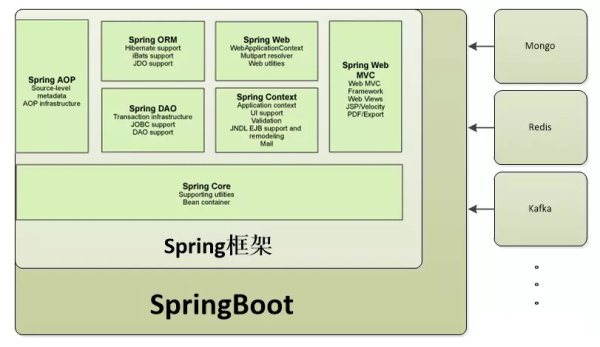
### 1.2.2 Java概述

Java是当今较为流行的面向对象的编程语言，凭借着健壮性、安全性、可移植性、平台无关性和简单性的特点[1]。Java在嵌入式、Android开发、桌面程序和Web应用程序中应用广泛，特别是在Web开发领域中，所提供的多种核心技术规范，如：JDBC、JNDI、RMI、JMS、JavaMail等[2]，经过长达数十年的发展，衍生出了许多强大的应用程序框架，诸如在IOC容器的Spring、Google Guice框架、数据持久层的Hibernate、MyBatis框架，网络应用程序的Netty、 Mina、T-io框架，网络安全的Spring Security、Shiro框架，模块化的OSGI框架，微服务的Dubbo、Spring Cloud框架，大数据应用领域Hadoop、HDFS框架，架设起来的企业级应用早已经具有不可撼动的地位，即使在不断的新技术、新概念、新语言诞生的今天，欣欣向荣的Java依然焕发强大的生命力，继续向微服务、分布式、模块化、函数式靠拢，较之与简单简洁，对开发者的友好的，在人工智能、科学计算领域有出色表现的Python依然不显劣势。

### 1.2.3 Spring Boot概述

Spring Boot是Java开发领域应用最为广泛的应用程序开发框架，它不依赖于任何具体的应用环境，因此Spring Boot可以被集成到多种开发环境中，有如Web开发环境。该Spring Boot框架是由Spring开发团队下的基于Java的用于简化应用程序的开发、测试和部署的集成框架，该框架遵循习惯优先于配置的原则[3]，在整合Spring框架原生态资源的基础上，以默认配置的方式，为框架使用人员准备了许多的默认的开箱即用的开发组件，相较于传统的Spring类型的项目，大大节省了项目用于框架功能集成和开发时间，这种优良的特性也为Spring Cloud的诞生打下了基础。此外，Spring Boot还提供了服务发现机制，在按照该机制的默认规则下，开发人员能够就简单地将项目中的互不影响的业务独立组装成一个组件，以在业务层面上提高代码地复用效率。

Spring Boot的基本架构如下图所示：



**图1 Spring Boot架构**

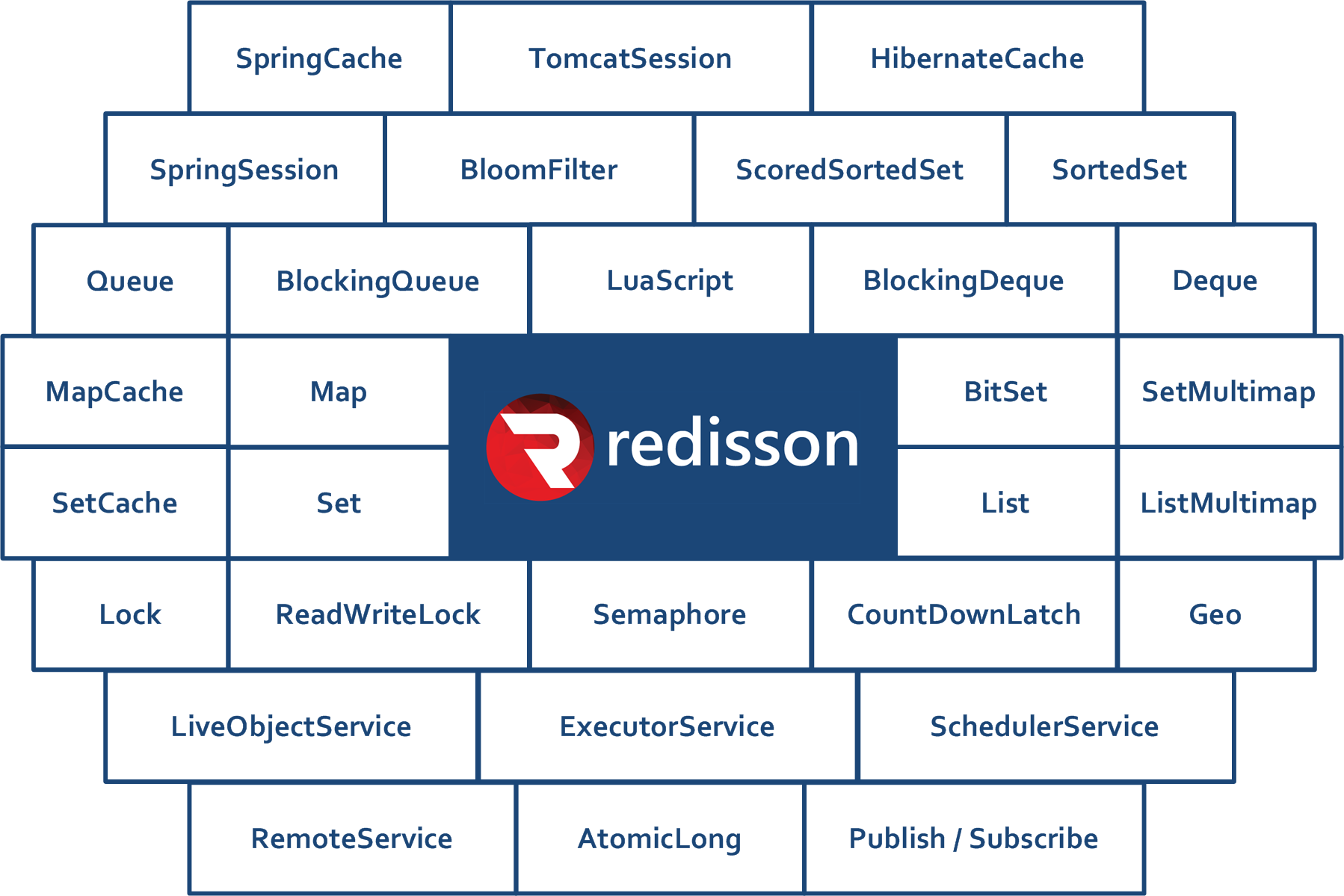
### 1.2.4 Ngnix概述

Ngnix是一款是由俄罗斯的程序设计师Igor Sysoev所开发高性能的 Web和 反向代理 服务器，也是一个 IMAP/POP3/SMTP 代理服务器[4] ，其基于IO模型epoll and kqueue，由C语言进行编写，也因此并发能力强、内存开销小、部署简单、易于配置，在一些高并发情况下，常用作为Apache服务器的替代品，用于HTTP的代理服务器。在本项目中，Ngnix则用来做简单的长连接的负载均衡和容错处理。

### 1.2.5 Redission概述

Redission是架设在Redis键值数据库基础的上的Java驻内存数据网络。该框架充分了Redis数据库的优势，提供简明易懂的API调用接口，赋予了程序分布式多机多线程多并发的执行能力，大大降低了设计和开发分布式系统的难度。同时依赖于非阻塞的高性能、异步事件驱动的网络编程框架Netty的Redission的底层，在原Redis的能力基础上，增加了非常的分布式数据结构和服务，有如：有界阻塞列队（Bounded Blocking Queue）、布隆过滤器（Bloom Filter）、分布式锁（Distributed Lock）等。

更多有关于Reddission的功能与特性如下图所示：



**图2 Redission功能与特性**

在本项目中利用Ressdission提供的服务，以使得即时通讯模块获得聊天数据的数据缓存的能力，满足即时通讯系统即时相应数据的要求，同时在被Reddision赋予更丰富的特性的发布订阅（Sub/Pub）模式的驱动下，为通讯模块提供基于增加数据处理节点的横向拓展通讯性能的能力。

### 1.2.6 Redis概述

伴随着Web2.0时代的兴起，传统的关系数据库的运用在数据规模大、数据种类繁多、高并发的场景中的表现略显乏力，在这种情况下,在CAP(一致性、可以用性、分区容错性理论指导下的NoSQL数据库产品的呼声和使用范围日渐高涨，在事物目前非关系数据库的种类有：键值数据库、列族数据库、文档数据库以及图数据库。Redis则作为键值数据库的代表，是由ANSI C语言编写，是一个高性能的提供原子性操作和的分布式键值数据库，相对与memcached数据库，Redis支持更多的数据类型，不仅如此，它还提供数据持久化操作[5]。在应用领域方面，Redis可以用来缓存热数据，提高系统的相应性能；可以用来作为队列，解决在高并发的情况的请求阻塞问题，典型的框架有ActiveMQ、RocketMQ；可以实现分布式锁，实现在高并发情况下事务的问题。

### 1.2.7 SeaweedFs概述

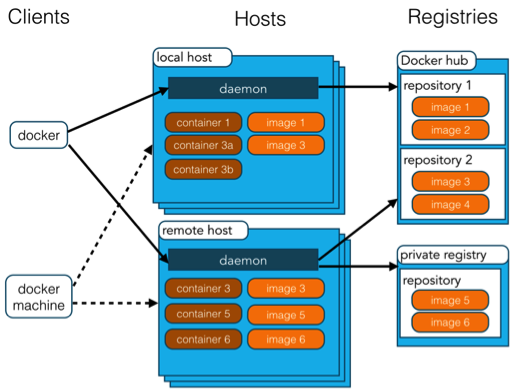
SeaweedFs是基于Go语言开发而成的分布式的适合于中小文件存储的文件存储系统。SeeweeFS是Facebook针对于海量中小文件存储，提高系统并发性能而提出的有关于图片存储服务器的论文Finding a needle in Haystack: Facebook’s photo storage[6]的一个开源实现。该文件存储系统的分布式部署主要是采用主从结构部署，将文件内容和文件的元信息的存储分离，其中元信息为40字节的大小，文件内容以二进制的方式进行存储，也仅消耗O（1）磁盘读取时间。在文件存储上主服务器只是管理部分的文件的元数据和文件卷轴信息，而将文件的元数据管理下行并存储到卷服务器当中，从而提高文件系统的并发性能。相对于FastFS，SeaweeFs提供针对于图片类型的文件访问提供了尺寸调整的API，也自带了Restful地址的访问风格和可供自定义的文件冗余复制策略，使得该文件存储系统的可用性和可维护性得到提高。

### 1.2.8 T-io概述

T-io是国产的Java网络编程框架，其前身是中兴的通讯模块EMF，该框架架设于运输层和应用层之间，利用自JDK7新一代的Aio（异步非阻塞）编程API开发而成，拥有更优越的网络通讯性能，在单机环境的测试中每秒发送500W条通讯聊天信息、支持数十万人同时在线。和Netty、Mina适用于同一类框架，但T-io更专注于提供网络通讯业务的解决方案，自带了半包和粘包的处理，为开发者准备了丰富的网络通讯API，如同用户ID到channel的绑定、channel到群组的绑定、SSL通讯能力、流量监控等[7]。有赖于良好的社区环境和友好的API支持，目前，在开源社区中，基于T-io开发而成的应用和框架良多，有HTTP服务器、TCP服务器、WebSocket服务器、即时通讯J-IM、游戏服务端贝密游戏等，本项目中的通讯服务器模块则基于T-io网络通讯框架开发而成。

### 1.2.9 Docker概述

在一个计算机层次结构中，有硬件层、操作系统层、函数层和应用层。抽象于硬件层和操作层之间的虚拟机，虽然性能的问题上还有待调整，但在云计算、大数据的发展浪潮中解决了虚拟化、多租户的问题。而抽象于操作系统层和函数层的容器化技术，不同于包含太多有关于操作系统的层面上的虚拟，因而轻量、系统开销低。容器化技术中的代表Docker是基于Go语言开发的一个开源应用容器引擎，其内部采用沙盒机制，不同的容器的运行相互隔离，每个容器里的仅仅是“操作系统”的内核，开发者可以将应用、以及依赖包打包到容器中，即可以在任何到Linux机器上运行。



**图3 Docker体系结构**

在上图中，Docker machine是运行了容器实例的机器，容器镜像与容器就如同类和对象的关系，一个镜像创建多个容器实例，Registies则为镜像的远程仓库，用户可以自行搭建私有的镜像容器仓库。运行时，Docker的守护进程将会检查目标容器镜像是否存在于本地机器中，当不存在时，Docker守护进程将从远程查找指定的容器镜像并下在到本地机器后在进行实例容器创建和运行的操作，用户可以在客户端可以对容器实例进行访问和控制。

Docker技术广泛用于Web应用的自动化打包和发布中并常常出现在SOA架构中。本项目将采用Docker容器化测试和部署相关的服务器[8]。

### 1.2.10 Node.js概述

自HTML,CSS,Javascript等被冠于前端技术[9]的代名词，前端系统结构不清晰，代码复用程度低，不易调试等问题一直被人诟病，但经过长期的发展，Node.js的出现为前端系统化奠定了温床。Node.js是一个开源跨平台的能够在浏览器之外运行Javascript的运行环境，采用事件驱动IO模型，基于Google V8引擎开发而成，拥有极好的性能，能够让开发使用Javascript语言编写命令行程序或者是动态网页的服务端。和Node.js一同发布的Javascript包管理工具npm，能够远程获取第三方Javascript开发工具包，从而提高Javascript的编程效率。在本项目中采用Node.js作为Javascript语言的运行环境，辅助组件化、工程化的前端的开发。

### 1.2.11 Nerv.js概述

Nerv.js是国产的由京东凹凸实验室开源的一个类React组件框架，使用虚拟Dom技术，拥有和React基本一致的生命周期函数，但相对于React的虚拟Dom算法进行改进，并精简大量的事件合成逻辑，从而大幅度的提高框架的性能[10]。

Nerv.js的基本特性：

1. 精简的框架体积，使其可以在低性能的移动端中依然有良好的加载速度
2. 采用起别名的形式，能够轻松地从React转移到Nerv的开发当中，并获取React中的良好的生态支持。
3. 结合中国国情，兼容IE8+。

### 1.2.12 MySQL概述

MySQL是一个采用双授权政策的关系数据库，自1995年MySQL诞生以来，作为一项开源产品，在随后便获得的广大的互联网公司青睐，成为了广受欢迎的数据库产品。在本项目中将采用MySQL5.7来进行持久化的数据存储。同时为了强调数据库的高可用性，可以使用国产数据库中间件MYCAT对数据库持久层进行抽象封装隔离，其支持数据库的读写分离、集群和软负载均衡，能够在项目代码不需要进行改动的情况下可以动态地调整的数据库集群的性能，提高系统的灵活性。

# 2　系统的需求分析

## 2.1 高校互联应用的现状和存在的问题

在国内，随着网络技术、电子信息技术的不断发展，早在本世纪初国内很多高校开始进行数字化校园建设且许多高校的基础网络早已建成，而现在高校的建设重点放在了具有个性化的校园应用软件中来。现今，围绕校园的应用早已遍地开华，百家争鸣，其主要分为以下10种[11]：

1. 校园社交类，一款在校园内学生与学生之间，学生和老师之间，老师和老师之间进行社交的应用。
2. 校园生活类，一款针对大学生的使用校园应用，它的应用领域主要有课程表、选课和评教等。
3. 校园求职类，一款针对于学生兼职或者实习的应用类。
4. 校园情侣类，一款针对于校园恋爱的应用。
5. 校园学习类，针对学生知识辅导的应用。
6. 校园商城类，针对于尚未有确定收入的学生定制的一款应用，专注于销售白菜价的应用。
7. 二手物品信息类，有如许多的寄宿于学生宿舍的学生在毕业之后，留下的大件物品将在二手售卖平台上进行转手。
8. 取快递类，针对于大学生开发而出的一款方便于大学生取快递的校园应用。
9. 日志类。
10. 动漫类。

校园中围绕着以上10种应用的领域的应用繁多，特别是在迎来微信公众号和微信小程序的爆发式增长后，有赖于微信的便利，校园的应用遍地开花，各大机构和社团为了方便工作的开展，也创建了相应的微信公众平台和开发微信小程序，方便了广大是师生了解和参与到校园的日常活动中来，对于营造校园的良好的生活和工作氛围起到了重要的作用。

在这几种常见的校园类型应用中，在方便于校园的信息交流的同时，也存在自身的不足和缺陷。

1. 并没有针对应用于学校各大学生社团机构日常工作交流和统一学生信息投放的平台的应用。工作负责人总需要在各大社交平台上投放校园信息，在信息层层传递的过程中带来人力资源消耗的同时，校内消息分布在各个角落，也得不到有效的投放和管理。
2. 没有相对较好的机制确认消息下发时目标人群的接收情况。对于比较紧急的消息下发时，任务负责人总需要确认在目标人群对信息的接收情况，因此在有需要的情况下，为保证信息有效传达，总需要去统计和确认具有哪些人没有接收到信息后，再通过其他方式通知对方。针对于此种刚需，放眼于市场，在市面上，有针对于企业之间工作交流的阿里出品的企业级应用：钉钉。钉钉（DingTalk）是阿里巴巴集团专为中国企业打造的免费沟通和协同的多端平台，适合于中国企业之前交流应用，而直接套用于校园当中而显得冗余而不足。
3. 跨专业需求得不到的有效的转达。信息技术的发展在传统行业的支持下才能焕发强劲的生命力，新工科的推进亦是如此，跨专业交流和合作将会成为常态，而校园内没有满足该需求的平台进行支撑，造成了需要在广大社交圈子中传播专业需求的状况。
4. 缺乏社团和机构意见反馈机制。

## 2.2 校园发布助手的研究意义

如何开发出使用于校园内交流的平台，提高校内信息传递的效率和学生机构办事效率，减少没有必要的人力开销，使得大学生能够将更多的精力放到更有意义的事情上，应成为高校认真探讨的问题。而校园发布助手能提供校内为广大师生统一的信息交流平台，将信息得到有效传递成本降到最低，在运用行业前沿的技术和经验，保证校园发布助手持续稳定高效运行的同时，校园发布助手出于便于办公的需求下，也将为学校各团队、学生机构、社团之内，以及普通学生和团队、社团和学生机构间的交流平台，使得校园中的每个人都能够被紧密地联系在一起，降低校内团体和个人，个人和个人之间交流的门槛，提高校园的信息化程度，从而达到提高校园生活质量的目标。

# 3　系统的总体设计

## 3.1 系统的功能简介

校园发布助手是一个专注于校园里是师生进行日常信息交流的系统，主要体现为H5端和微信小程序端。在该平台中，校园师生均采用实名制交流，它能够帮助用户及时了解校内的社团、机构、教师以及他们的最新资讯；能够充当通讯录，使得用户能够找到相关负责人尽快为你解决问题；能够帮助用户争取话语权，为校园社团和机构的建设建言献策；能够让用户及时了解消息的传达情况，使用户更好地在校园里工作和生活。该系统主要四个模块：校园模块、机构模块、通讯模块、个人模块。

### 3.1.1 校园模块

该模块提供校园日常资讯的浏览，方便校园师生了解校内的动态。校园信息的分类主要包括：要闻、招聘、活动和其他。涵括的功能点有：

1. 校园资讯浏览功能。
2. 校园资讯的提问和回答功能。用于阅读到该条咨询时感到困惑，可以直接在咨询发文处进行提问和解答，有助于消息的传播。
3. 校园资讯的搜索功能。
4. 校园通讯的提醒功能。

### 3.1.2 机构模块

该模块是校园中所有的学生机构和社团的聚集处，是系统中学生和机构、社团交流的通道，也是学生和机构中的人员管理和发布校园咨询的通道。包含的功能有：

1. 学生机构或社团浏览功能。其中，学生机构采用多级菜单的形式展示，学生机构提供多种搜索方式方便用户找到目标社团。如用户属于某学生机构或社团，则直接进行展示并提供机构或社团的入口。
2. 学生机构或社团管理功能。学生机构或社团的负责人能够修改社团或机构的简介、公告，调整社团或机构的成员、查阅反馈信息。
3. 即时通讯工作群新建功能。满足机构或社团为进一步落实学院工作，统一目标学生群体的通知工作，同时为满足便捷添加目标学生到群中，提供了批量搜索和添加成员的功能。
4. 校园资讯的收藏功能。
5. 学生机构或社团信息查阅功能。提供社团和机构专属平台，使师生有更多的机会了解机构或社团以及交流。
6. 学生机构或社团信息反馈功能。无论是否属于该学生机构或社团的学生，都能够为该社团或机构反馈信息，使得机构或社团能够得到更好的发展。

### 3.1.3 通讯模块

该模块满足校园工作的要求，为学生机构或社团提供即时通讯服务。其主要的功能有：

1. 即时通讯功能。用户能够与所属的学生机构或社团中的任一成员进行通讯，也能够与社团或机构群体进行通讯，并提供群中的图片和其他文件的搜索功能和其他常用的及时通讯的功能。
2. 文件传输功能。通讯双方传输的文件将半永久地存储到服务器当中。
3. 未读聊天信息提示功能。
4. 联系人、聊天记录、文件以及图片搜索功能。
5. 群管理功能。群管理者能够进行调整群成员、发布群公告以及解散工作群的操作。
6. 聊天信息、图片等文件收藏功能。

### 3.1.4 个人模块

个人模块提供个人信息的管理功能，用户能够有限地对自身的修改和发布。也能够自定义客户端参数，满足个性化定制的要求。

1. 个人收藏的信息管理功能。用户能够删除个人收藏的信息，点击收藏的信息能够查看该消息的详情。
2. 个人信息的查看和修改功能。提供QQ邮箱和手机号的绑定，便于校园资讯和即时通讯信息的即时推送和提供忘记密码是取回账号的凭证；提供用户自身的信息可见性的设置，一定程度下保障个人的信息安全；提供个人的留言设置，方便他人联系。
3. 客户端自定义功能。如是否记住登录用户和密码。
4. 登录和退出功能。

## 3.2 系统的结构分析

本项目通过了解校园不断增长的使用需求，整理和研发出一款满足学生日常工作和生活的校园助手。在校园的推动下，该校园助手可以扫除消息传播的障碍，为校内师生了解和参与到校园的工作和生活中，有效减轻校园工作的负担。

### 3.2.1 系统用例分析

1. 本系统的参与者有普通学生、教师、机构或社团成员、管理员和文件服服务器。



**图4 系统中的参与者**

在上图的参与者中，由于机构或社团成员拥有普通学生的行为，在表现上机构或社团成员拥有更多的个人，因此，机构或社团成员与普通学生的属于泛化关系；在本系统中，文件服务器将从业务本身玻剥离出来，使得文件能够进行横向拓展，并为系统以Restful风格提供文件上传、文件下载、文件删除等功能；Redis服务器为系统中的即时通讯业务提供缓存服务，优化即时通讯响应性能；由于系统主要面向移动端用户，管理员应有多种类型且担当重要角色，但在次系统不足以体现，因此不在论文阐述的范围内。

1. 学生是本系统中的重要参与者，也是本系统中的一个重要服务对象。



**图5 学生用例图**

在本系统中，共分为四大模块，有资讯模块、机构模块、通讯模块和个人模块。在资讯模块，学生用户能够浏览、搜索校园资讯和收藏校园资讯，在对校园资讯的提问尚未被回答时拥有删除提问的权限；在机构模块，学生用户能够搜索校园学生机构或社团及查看其的详细信息，能够对其进行信息的反馈，也可以取得该学生机构或社团的联系信息；在通讯模块，系统处理提供即时通讯机制之外，还能够浏览、搜索和删除聊天信息； 在个人模块，学生用户能够自定义客户端的行为，如是否删除持久化数据、用于下次免登录的保存用户信息和密码功能，还能够查看收藏的信息等。

1. 机构和社团成员作为校园资讯生产者，其用例图如下：



**图6 机构或社团成员用例图**

机构或和社团成员除了拥有学生享有的功能之外，还拥有属于该角色的功能。在不考虑机构或社团中成员的权限范围之内，该角色能够发布校园资讯、查阅反馈信息，能够在机构模块中创建工作群、管理社团或机构，能够在通讯模块中管理工作群。

1. 教师可以在系统中查询一定范围内的学生信息，以便于日常事务的交流。



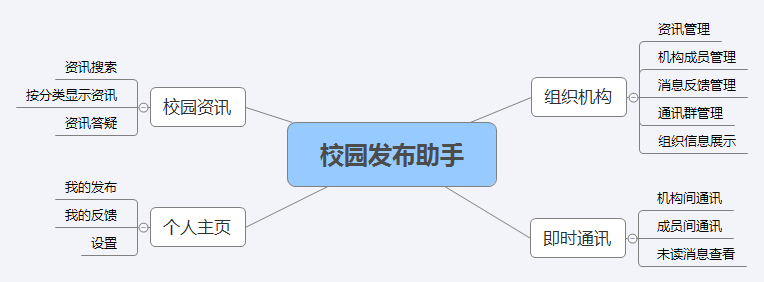
**图7 教师用例图**

教师角色除了继承了学生角色拥有的功能， 还拥有发布校园资讯、发布公告、查阅反馈信息、查询班级信息和管理工作群的功能。

### 3.2.2 系统功能结构分析

为满足不同用户的使用需求，本系统共分为子系统：校园发布助手H5端，校园发布助手小程序端，校园发布助手后台管理系统。

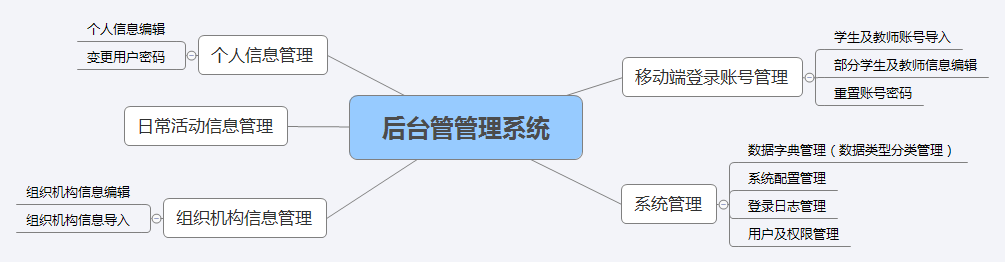
1. 校园发布助手H5端及小程序端持有相同的功能，面向校园普通用户，如校园各大社团机构成员、学生及老师，满足校内消息流通基本需要。其功能结构如下图：



**图7 校园发布助手移动端功能结构图**

其中，校园资讯子功能中，按照校园资讯分类进行显示，个人主页中能够查看个人发布的校园资讯信息及反馈信息；组织机构中按照社团机构类别进行展示；

1. 在校园发布后台管理系统中，面向的用户是管理员，功能结构如图所示。



**图8 校园发布助手后台管理系统功能结构图**

后台管理系统维护系统中的数据字典类型具体的数据字段分类、校园学生社团机构和班级信息的管理、登录系统中学生或教师基本信息；设置系统基本配置信息，如移动端的基础安全信息配置；检查发布在移动端中的校园资讯等。

## 3.3 系统部分需求及关键业务分析

为满足系统上的功能需求和非功能需求，达到在该系统的使用下，进一步优化校园内消息传播的效率，提供校园工作和生活的质量，有效减轻负担，需要针对部分需求针对实现上使用统一建模语言UML进行相应的整理和归纳，建立软件分析模型[12]。

### 3.3.1 移动端登录

在移动端需要开发即时通讯功能的前提下，为满足实时获取信息以及聊天成员的要求下，移动端的登录业务顺序图如下：



**图9 移动端登录顺序图**

在本系统的设计当中，为保障即时通讯的性能及系统拓展要求，提高程序代码的可复用性，实时性需求不高的系统服务将通过HTTP协议的方式进行提供。此次在移动端以WebSocket协议进行移动端的登录，在用户输入账号信息后，移动端程序将发送至WebSocket服务器中，当WebSocket服务器组装成一定格式的信息后，发起调用登录系统服务，服务提供者进行登录信息的验证后，将登录结果返回到WebSocket服务器中，而当登录信息正确时，WebSocket服务器将生成用于登录其他服务器的Token，请求相关的群组以群组成员信息返回至移动端程序中。

### 3.3.2 校园资讯浏览

校园资讯作为该系统中重要的模块，每天为广大校园师生提供及时的校园资讯，满足校园资讯无障碍流通的要求，在移动端进行校园资讯展示时，系统的业务流程如下图所示。



**图10 移动端校园资讯浏览活动图**

当需要访问HTTP协议的资源时，则把在WebSocket协议下完成登录返回Token存放在HTTP请求的头部中，为HTTP的认证和授权使用。每个请求也将会携带该Token作为验证的使用，而当验证不通过时，将不再返回请求的系统服务资源，故而下文不再赘述该过程。在本例中，用户在浏览点击调转到校园资讯页面时，移动端程序携带Token并发起校园资讯请求，在验证通过时，系统将会根据登录用户的账号信息和校园资讯发布时定义的校园资讯可见性进行校对，并根据资讯的创建时间和资讯的紧迫程度进行排序输出。

### 3.3.3 校园资讯发布

在本系统中校园资讯不能以个人形式发布，而是以所处的组织的身份进行消息的发布，由于在校学生担任多样，因此在组织结构模块出提供快捷的社团机构入口，校园资讯发布的活动图如下所示。



**图11 移动端校园资讯发布活动图**

当用户进入组织结构标签界面，选择所属的目标组织进入组织界面，进而填写相应的校园资讯信息，正指定完成相应的校园资讯范围后，提交到服务器当中进行验证操作，再服务器处理完毕时，将返回发布的状态结果返回到移动端中。

### 3.3.4一对一即时通讯

在用户在移动端登录后，标志着服务器间和客户端之间的双工通信已被打通，在WebSocket服务器进行存储和转发下，校园师生能够进行一对一、一对多的有反馈的及时通讯。一对一即时通讯顺序图如下图所叙，在该图中是站在用户1在线的角度进行描述，暂不涉及离线信息、持久化信息等WebSocket服务器当中的一系列的业务逻辑。



**图12 一对一即时通讯顺序图**

在图中，用户1选择目标用户2，点击发送消息，在WebSocket服务器接受到用户1的消息时，根据用户1提供的消息字段找到用户2的消息通道并向用户1返回发送成功响应消息，而当用户2在线时，WebSocket服务器开始向服务器发送用户1的消息，用户2完成用户1的消息阅读时，并开始向WebSocket服务器发送已读用户1消息的响应信息，WebSocket服务器在接收到用户2发送而来的消息后，将已读消息重新发送到用户1并发送响应消息给用户2，使得用户1和用户2能够得知已读消息已经发送成功且已被阅读。

### 3.3.5 即时通讯集群

随着用户数量的增长，单机服务器难以同时处理大量用户之间的即时通讯数据量，为保证即时通讯服务的即时通讯的及时响应性能，应当使用负载均衡策略，如DNS轮询、随机分配、针对IP哈希等策略、对用户进行分流。当目标即时通讯用户在同一机器时，将直接进行消息的存储和转发，当目标即时通讯用户不在同一个处理服务器时，将发布该消息到具有TOP/PUB策略的服务器上，由该服务器进行消息存储和转发的操作，在本项目中使用Redis非关系键值数据库承担消息的存储和分发操作，提供即时通讯系统的可用性和健壮性。



**图13 一对一即时通讯集群交互顺序图**

上图为在集群环境中通讯双方都同时在线时，而不在同一个WebSocket服务器上的一对一即时通讯集群交互顺序图，上图中不带有实心黑色箭头的为异步消息，即是发送完毕后不是等待消息的返回，而是执行其他任务，而当消息返回时才执行先前留下的任务。在图中，用户1将消息到所在的WebSocket服务器1上，当WebSocket服务器1找不到用户2所在的通讯通道时，将会把用户1发送的消息发布至Redis服务器上并返回发送成功响应信息，订阅到该频道的WebSocket服务器2将会收到用户1发送而来的消息，继而将该消息发送至用户2。用户2在接收到该消息并发送已读消息到WebSocket服务器2，当WebSocket服务器2检测到该消息的目标不存在于该服务器中时，将会把该已读消息发布至Redis数据库当中。同样订阅了该频道的WebSocket服务器将会接受到Redis数据库发送而来的已读消息，也将该已读消息发送至用户1，从而完成本次一对一通讯的过程。

### 3.3.6 组织机构创建并添加成员

创建组织机构和新增社团和机构成员的功能并不放在移动端当中，而由后台管理系统进行处理。创建组织机构和添加成员的活动图如下所示。



**图14 组织机构创建并添加成员活动图**

上图中，管理员进行登录后台管理系统后，点击创建组织机构，后台管理系统进行基本信息校验后，为组织结构添加信息，并新增一条聊天工作群，存放到聊天群表中，而当管理员在为该新建组织添加成员时，后台系统在添加成员的同时，也会在聊天群成员表中添加该成员的基本信息。

## 3.4 系统的关系型数据库设计

### 3.4.1 关系型数据库概念设计

本系统开发的总体目标为为广大高校师生提供信息交流平台。目的在于提高学生机构或社团产生的校园信息的传播的质量和效率；充当校园通讯录角色，免去寻找目的人而花费的时间；提供稳定有效的即时通讯服务，提高信息交流的质量。针对于以上的需求和目标，在数据库的设计阶段的任务的体现上，主要是对用户信息、日常资讯信息、聊天信息、组织机构信息、资源文件等信息进行操作和处理。因此在数据库概念设计阶段中，将会对实体数据的内在语义进行分析，以把显示现实世界中存在的实体转化为概念数据模型。如下图8所示。

**图15 实体关系图**

在上图中，一个角色拥有多个系统资源，一个系统资源能被多个角色拥有，因此为多对多关系；一个用户能够担任多种角色，而一个角色能够同时分配给多个用户，因此为多对多关系；一个用户能够在多个机构中任职，同时一个机构能够同时招收多个用户，应为多对多关系；一个机构能够发布多个校园资讯，而一个校园资讯仅能被一个机构发布，因此为一对多关系；一个用户能够加入多个聊天群，一个聊天群亦能够被多个用户加入，因此它们的关系为多对多关系；一个用户能够收藏多个校园资讯和聊天信息，而一个校园资讯和聊天信息也能够被多个用户收藏，因此它们的关系分别为多对多关系。

### 3.4.2 关系型数据库逻辑设计

本系统采用MySQL5.7作为系统数据持久化的关系型数据库，在建模软件EA中对数据进行关系建模后，可直接导出建库语句。

1. 用户信息表，用于记录用户的用户名、密码、邮箱、手机长号、短号等信息，为系统接入用户信息，是给用户提供校园资讯、即时通讯等服务的前提。

表1　用户信息表



1. 系统资源表，用于记录系统中的资源，如系统服务接口URL、客户端按钮或表单资源等。由于采用前后端完全分离的模式开发，前后端的资源需要一并记录。

表2　资源权限表



1. 系统角色表，用于记录系统中的角色，如普通学生角色、管理员角色、机构或社团成员角色等。角色作为权限的集合，基于RBAC（基于角色的权限访问控制）设计而成，能够灵活地管理和拓展系统资源。

表3　系统角色表



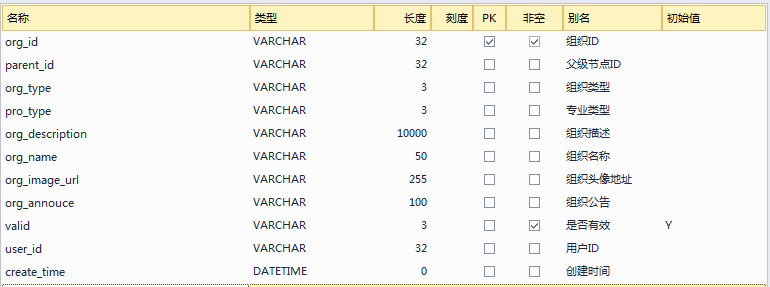
1. 系统角色权限表，用于记录角色的拥有的权限，由于角色和权限是多对多的关系，以此表来记录角色和权限之间的关系，使得角色和权限能够分离发展。

表4　系统角色权限表



1. 部门组织表，用于记录校园中部门组织的基本信息，包括班级、机构和社团，由于班级和机构或社团有一定的同性，班级则将被记录到该表中，此外部门组织普遍存在上下级关系，用parent\_id字段来表示部门组织的树形结构。

表5　部门组织表



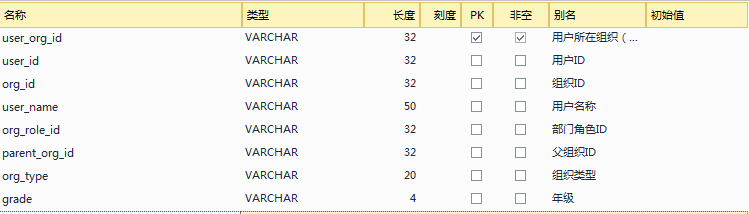
1. 部门角色表，用于记录部门包含的具体角色信息，对应于部门和系统角色的多对多关系，为具体部门中的角色保留泛化空间，如限制部门中的角色数量。

表6　部门角色表



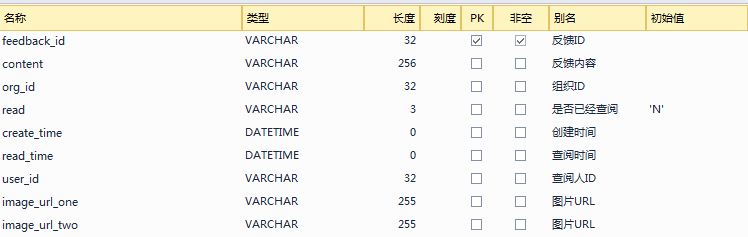
1. 用户部门表。用于描述用户在部门中担任的具体角色，系统中允许一个用户可以在一个组织当中担任多种角色，以满足在社团或机构在人员紧缺下需要多职位担当的需求。

表7　用户部门表



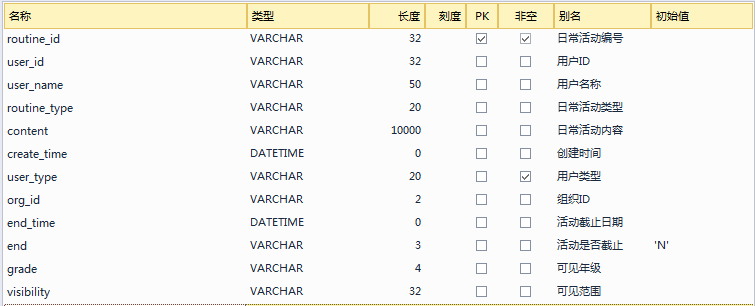
1. 信息反馈表。记录用户对机构的执行情况的反馈信息信息，与用户、组织分别被一对多的关系，由于系统中采用匿名信息反馈机制，因此本表中将不会记录反馈信息的用户ID。此外，对于用户反馈的信息，只能带有两张以内的图片，而在组织内人员审阅该反馈信息后，反馈信息将变为以审阅状态。

表8　信息反馈表



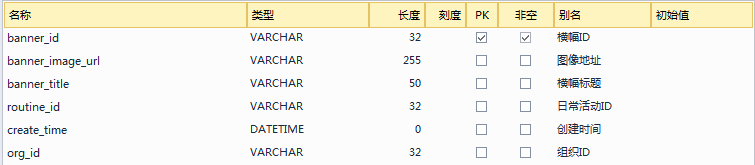
1. 校园资讯表。存储在系统中的发表的校园资讯信息，先校园资讯的信息的主要分类有：要闻、活动、招聘和其他。为了实现校园资讯的主动对用户不可见的功能，有如可见范围：所有人可见、本院师生、本机构和本部门，在系统中用可见范围和可见年级字段记录校园资讯的主动可见范围。

表9　校园资讯表



1. 资讯横幅表。用于记录学院发布校园资讯时带有的横幅信息，与学院的关系为一对一的关系。

表10　校园资讯表



1. 资讯评论表。存放用户对于校院资讯的质询和回答的信息，不存储评论的多层关系，便于消息发布者能够在清晰地针对有效地针对性地回答用户的提问，提高校园资讯传播的准确度。

表11　校园资讯表

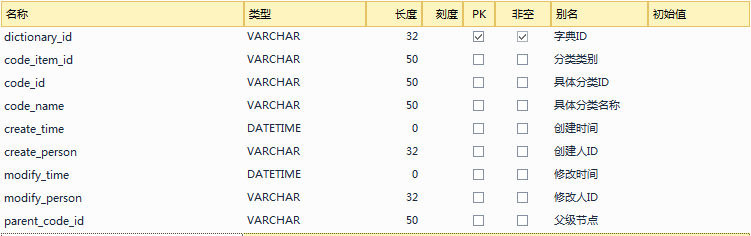


1. 用户收藏表。根据不同的收藏信息，收藏内容存储不同的信息，用户能够收藏的数据类型有聊天信息和校园资讯，一个用户能够收多条聊天和校园资讯信息。

表12　用户收藏表

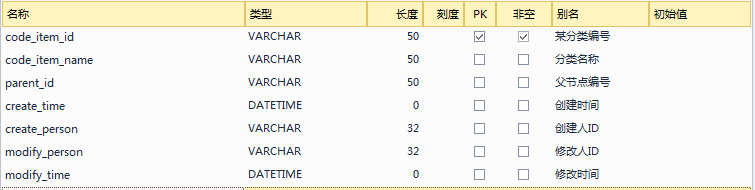


1. 字典类型表。记录某实体具体的分类信息。在实体类型查询时可通过分类类别和具体分类名称进行实体类型的确认。

表13　字典类型表

1. 数据字典表。用户记录系统中实体类型有限的类型信息，而不是针对某中实体而新建一张表来存储它的类型信息，从而达到节省表数量、提高开发效率的目的。

表14　数据字典表



1. 通讯群表。用于记录在系统中能够进行集体通讯的群体信息。在学生社团或机构新建时，系统将默认为其创建一个不可直接删除的通讯群号，且在学生机构或社团中提供创建工作群的入口。

表15　通讯群表



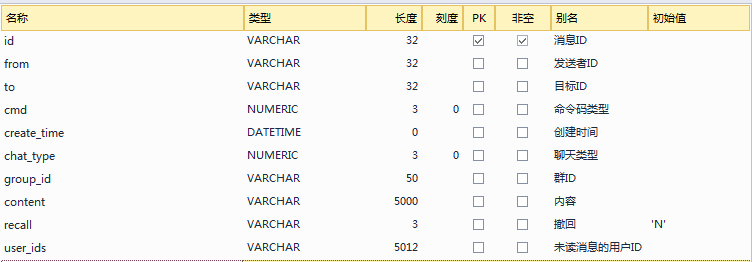
1. 群成员信息表。该表记录了工作群中的具体成员，用于在即时通讯系统中设置消息分发的目标群体。

表16　群成员信息表



1. 聊天信息表。用于记录在多方通讯过程中产生的信息，由命令码类型字段记录聊天信息的类型、聊天类型字段记录该消息是否是群发还是私聊信息类型消息。

表17　群成员信息表



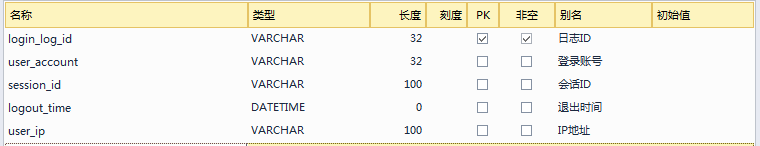
1. 文件信息存储表。存储在校园资讯和聊天信息过程中的产生的文件元信息，具体的文件内容将存储到文服务器当中，可通过业务ID和业务表名称查找相应的信息中带有的文件的相对地址列表。

表18　文件信息存储表



1. 登录日志表。记录用户在系统当中的登录情况，可用于在跨域身份验证解决重放攻击的问题。保证系统的稳定性和安全性。

表19　文件信息存储表



1. 系统配置表。记录系统中的配置信息。如用于安全通讯的公钥和私钥信息、是否持久化即时通讯过程中产生的聊天信息、版本号等。

表20　系统配置表



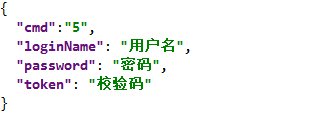
# 4　系统的具体实现

## 4.1 即时通讯消息格式定义

为保证即时通讯系统的灵活性和可拓展性，避免即时通讯代码的复用程度底，为后来的系统的维护带来便利性，在灵活使用工厂模式、责任链模式等优秀的后台系统设计模式的前提下，亦需要推进即时通讯消息格式的标准化和规范化。下文的消息格式均采用JSON消息格式，用消息的键代表某属性，属性冒号后代表其值，在不失良好的语义的同时能够减少网络传输带宽的同时适合转换成JavaBean。

### 4.1.1 移动端登录消息格式

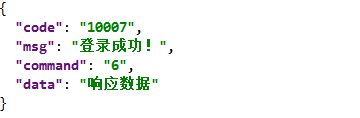
1. 请求时



**图15 移动端登陆请求消息格式**

图15为移动端登录系统消息格式，其中，“cmd”键代表映射到服务器中某一个具体的服务类，以下均为JavaBean实体中的一个具体字段。

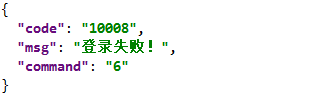
1. 成功返回时



**图16 移动端登陆登录成功消息格式**

图16中，登陆成功时，相关的群组信息将封装在data所处动字段值中。

1. 失败返回时



**图17 移动端登陆登录失败消息格式**

### 移动端聊天消息格式

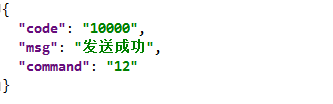
1. 请求时



**图18 移动端聊天消息格式**

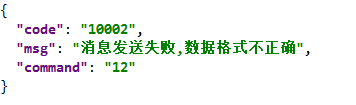
在图18中，“msgType”字段的现有类型有text、image、voice、file；“chatType”字段来表示此消息是否是群发消息；“groupId”字段在字段“chatType”为群发消息时有效。

1. 成功返回时



**图19 移动端聊天消息成功返回消息格式**

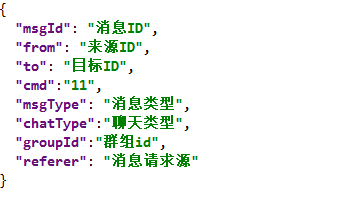
1. 格式错误时



**图20 移动端聊天消息格式错误消息格式**

### 移动端已读聊天消息格式

1. 请求时



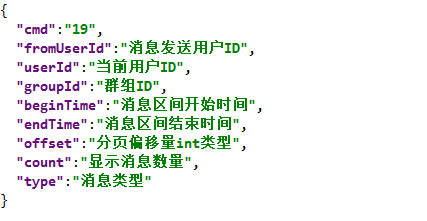
**图21 移动端聊天已读消息格式**

在图20中，“referer”字段包含源消息发送者信息，包括来源ID、目标ID等信息，供服务器及移动端进行消息的还原处理。

1. 成功返回时，同图19。
2. 返回失败时，同图20。

### 移动端获取聊天消息格式

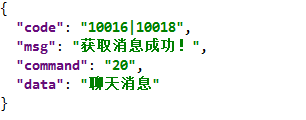
1. 请求时



**图22 移动端获取聊天消息格式**

在图21中，“offset”字段表示获取消息的偏移值，从0开始进行编号。“groupId”字段不为空时表示获取与群组有关的聊天信息，为空时表示以“fromUserd”字段获取与指定用户有关的聊天信息，“type”字段获取指定的类型聊天信息，如历史消息、用户离线信息。

1. 成功返回时



**图23 移动端获取聊天消息返回消息格式**

在图23中，“code”字段表示的值由于请求消息格式中的“type”字段有关。当“type”字段为指定获取历史消息时，“code”字段值为10016，而“type”字段为指定获取离线消息时，“code”字段值为10018**。**

## 部分需求及关键业务实现

### 4.2.1 移动端登录

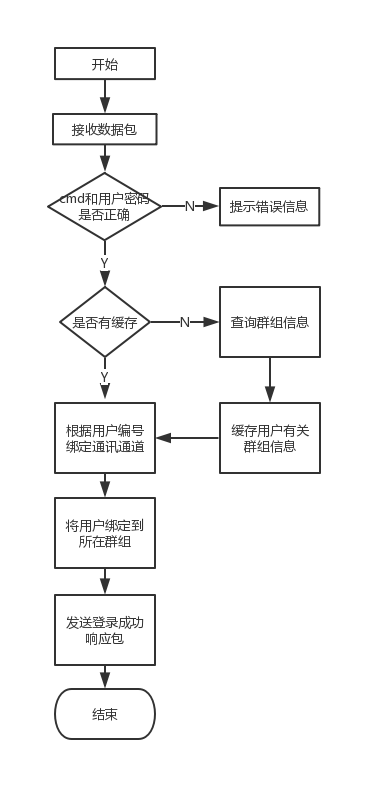
在进行登录操作时，需要有一个客户端和后台服务器握手的过程，在这个过程时，客户端和服务器进行密钥的交换，当握手操作完成后，客户端将账号和密码传输到后台服务器中。

代码1 移动端进行登录操作代码

|  |
| --- |
| // 登录:调用其他在socketTask的发送方法。 async login(payload) {  const socketTask = dispatch.socketTask;  await loginFunc(socketTask, payload);   }, // 登录之后调用的方法 async afterLogin(payload) {  // 改变用户登录状态  this.changeLoginStatus(payload);  if (payload.code == '10007') {  将用户信息发送至userState当中；  // 当登录成功时，更新相关state，进行页面调转  Taro.redirectTo({  url: '/pages/index/index'  });  将群组信息发送至group当中；  group信息刷新到messageAndChatGroup当中；  处理群组当中的用户信息；  保存token；  } } |

在代码1中，当用户产生登录事件，发起函数login的调用，函数loginFunc封装通过WebSocket协议发起登录的代码，当服务器返回结果时再发起对afterLogin的调用，在此函数中判断得知登录成功后进行回转数据的保存。

在移动端发送数据包到服务器时，服务将开始如图24所示的代码流程进行运行。



**图24 服务器响应登录请求代码流程图**

在整个代码流程当中，用户账号和密码验证通过时需要通过数据库查询与用户有关的群组信息和所在群组当中所有的用户信息，此时对这些信息进行缓存有利于提高通讯系统的响应性能；每个与服务器进行连接的用户总会使服务器生成一个与能够与用户进行通讯的通道实例，为了能够通过用户编号找得到该通道实例，需要通过Map对象进行两者之间的绑定，同时也需要将当前用户的通讯通道绑定到所有的所在群组当中，使的用户能够和群组当中的成员能够进行消息的交流。

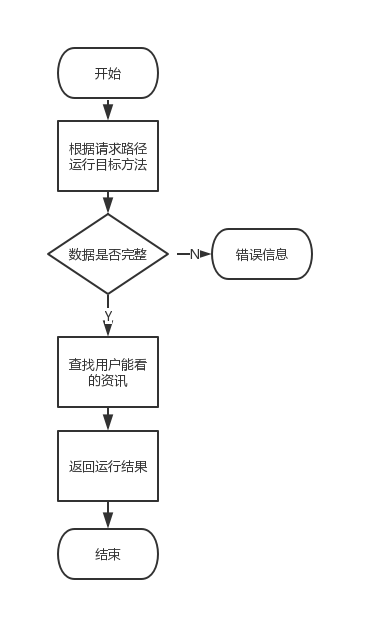
### 4.2.2 校园资讯浏览

移动端用户点击进入校园资讯界面时，将触发fetch函数的运行。

代码2 移动进行登录操作代码

|  |
| --- |
| const {dispatch} = this.props; // 触发函数fetch的运行  dispatch({  type: "routine/fetch",  payload: {} })  // 发起资源请求  async function queryRoutine(params) {  return request('/rest/appRoutineService/queryByPager', {  method: 'POST',  body: {  ...params  }  }); }  // 调用资源请求函数，保存资源  async fetch(payload) {  const response = await queryRoutine(payload);  this.saveEntitiesAndPagination(response);  this.deleteAndSaveCurrent(response); } |

如代码2所示，await和async关键字属于ES7当中进行异步操作语法糖，在await表达式处只有在请求处理完毕后，后续的代码才会被运行。在接收到服务器的响应后，将数据添加到本state结构树当中，供容器组件进行界面的渲染。

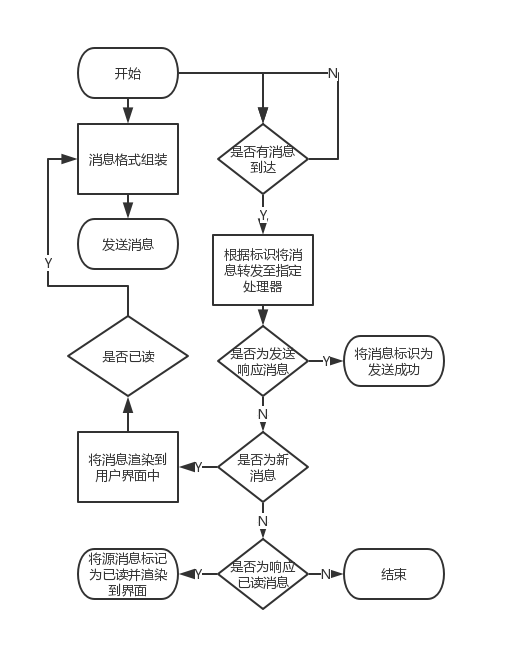


**图25 服务器查询资讯代码流程图**

对于验证数据的完整性的相关操作，在实现中交付与Aspect对象进行完成，当有服务方法标注了该横切面的切入点时，该Aspect开始根据实现定义好的规则进行数据的校验；查找用户能够查看的资讯时将绝大部分逻辑放进SQL代码中，从而充分利用SQL语言本身的查询优势。

### 4.2.3一对一即时通讯

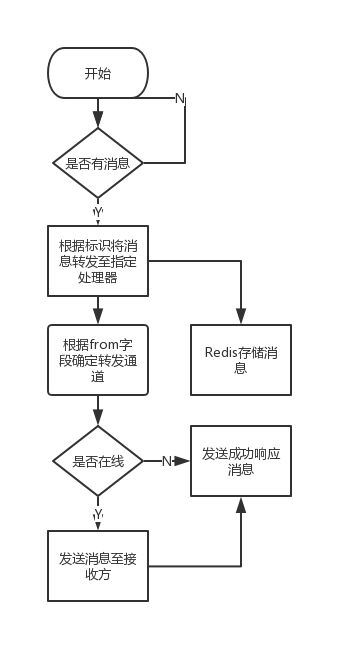
对于点对点即时通讯，在信息将被发送处理时，客户端的程序流程图如下所示：



**图26 客户端一对一即时通讯代码流程图**

上图暂不涉及到异常处理情况。当移动端在和服务器建立好连接好通道后，能够将移动端产生的具有一定格式的信息发送到服务器。当服务响应完成后，为WebSocket实例添加的onMessage事件处理函数将会被触发，作为发送方将接收到发送响应消息，然后让用户了解到该消息已发送成功。作为接收方将接受到该消息并认为该消息新消息，当用户已经阅读到该消息时，移动端程序将会把该消息的已读响应消息发送至服务器，以告知发送方接收方已经阅读了该消息，源发送方在接收到该已读响应消息时，将会把原发送消息标识为已读。

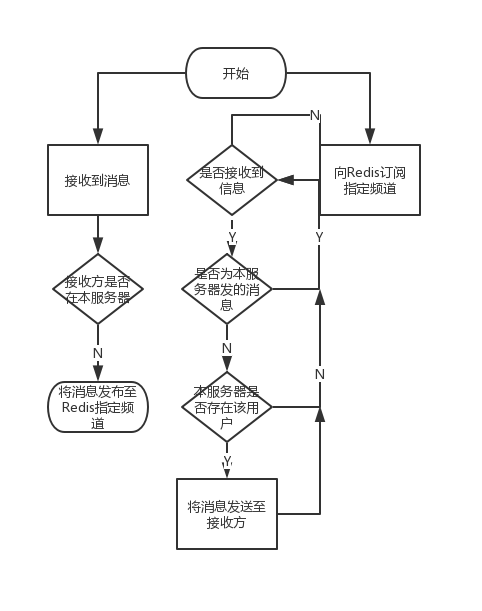
移动端将消息发送至服务器后，服务器做出如图27所示的代码流程进行响应。



**图27 服务端一对一即时通讯代码流程图**

接收到消息的服务器首先将判断消息格式的正误，接而将该消息转发到相应的处理类当中。而同时也回为该消息启用固定线程数量的线程池，使得在不妨碍服务器根据标识字段把消息发送到目标用户的移动端的情况下，对消息进行存储操作，为了让服务器能够对存储到Redis数据库的消息进行快速的查找和分页操作，通常根据需要对消息的发送方编号、接收方编号进行消息键名的动态生成。

### 4.2.4 即时通讯集群



**图28 服务器通讯集群处理代码流程图**

根据Redis连接配置，服务器连接到Redis数据库并向Redis数据库订阅指定频道信息，当Redis接收到某通讯服务器发来的消息时，开始向订阅了该频道的服务器进行无差别的消息通知，这时服务器在接收到该消息时需要判断该消息是否属于自身发布的消息，以防止不必要的性能的浪费。当该消息的接收方属于本服务器的用户时，服务器将会把该消息发送至接收方。对于本服务器接受到客户端发来消息里指定的消息接收方不存在于该服务器当中时，服务器将消息存储至Redis数据库的同时，也会将消息发布至指定的Redis订阅频道，把消息的转发任务交给集群中的其他服务器。

# 问题总结与分析

在项目进行调研和开发过程中，总会遇到大大小小的问题，现针对开发提出一些问题，并做简要处理方案解答。

1. 如果看待面向对象程序设计。

在OOP中，我们将客观存在的实体抽象为我们程序中的类或对象，把实体中的特征和行为分别抽象为类或对象中的属性和方法，这就是一种封装，但不仅仅是这样，它还被赋予了面向对象的高层次特性，不仅让子对象继承父对象的特性和行为，还让对象能够根据不同的环境因素下，作出不同的反应，从浅层的角度来，是提高了代码的复用性，从深一层来讲，它屏蔽了类或对象中的内部细节，使得我们能够从更高的角度进行设计和开发实现，比如说广为流传的设计模式，可以说是面向对象程序设计的产物。

1. 如何前端技术的选型。

自HTML、CSS、JavaScript等被冠以前端技术的代名词，前端系统结构不清晰，代码复用程度低，不易调试等问题一直被人诟病，但经过长期的发展，Node.js的出现为前端系统化奠定了温床，NPM的出现为统一依赖解决方案铺平了道路, webpack、Jest的出现使得前端工程化、项目化成为了可能，Vue、React、Angular的出现更是标志着前端向组件化、独立化的大前端方向大步迈进。因此在本项目中，亦采用Node.js、npm、Jest、React全家桶等优秀的前端开发工具或框架，以对移动前端和后台前端进行开发。

1. 如何在前端工程化开发过程中，进一步提高标签或代码可重用性。

在使用React编写前端功能界面时，特别是在编写后台管理系统时，对应于一张表的功能代码界面总会有类似的用于请求后台和展示界面的功能代码，而如果将这两种代码进行分割开来，各自进行组件化，用组件间传递的参数模糊不同数据库表中在数据库请求和界面展示之间的差异，从而使得前端代码能够得到很好的复用，除了需要后台业务逻辑系统中需要统一接口的参数传递标准之外，也需要对前端组件代码进行容器组件和展示组件的区分。

1. 如何解决程序中用户操作的权限问题。

在前后端完全分离的现在，几乎所有动界面元素是由前端代码自主渲染产生，而且前端的页面跳转已经完全不受后台的控制，只是提供数据接口的后台将无法统一和控制页面输出和具体业务场景权限，如果要对界面元素的操作进行控制，那么就需要对该界面元素请求的数据进行权限控制，也需要界面元素的可操作性进行限制，在实现方式上，数据库中的权限表应添加某一具体服务对应的API，当在前端登录成功时，获取有关的资源操作权限数据并将之作用于前端界面标签的渲染中。

1. 如何实现模块间代码间的解耦合，即是一个独立的功能模块不包含另外一个功能模块的任何代码，而又能像组装电脑一样，在使用时共同发挥作用。

要使得模块间代码相互独立，不受业务逻辑的影响，首先要保证第一点是项目工程之间的独立性，可以采用Maven、Gradle等项目工程管理工具进行工程项目的托管，在原本的项目中建立子项目，把属于子项目中的功能代码编写进子项目中，其次是需要规定一中代码加载机制，只要编写符合该机制的功能模块代码，主项目就能够通过既定的规则找到该功能模块代码，并且进行加载操作，从而该功能模块能够被运用到主项目中，进而发挥该有的模块能力。在Java语言当中，提供了一个SPI机制，是一种服务发现机制，该机制原本是被用于开发厂商之间针对同一个服务接口的不用实现，即是规定在类路径的META-INFO目录下定义接口的实现类的配置文件，在程序运行时进行配置文件的查找，然后根据配置文件加载响应的实现类。后来该机制被应用于框架和大型应用的开发工作中，在这种机制的帮助下，可将大系统解耦，从而使系统达到模块化，组件化的目的，从而提高系统的灵活性和可拓展性。在本后台工程项目当中将借助Spring Boot的SPI机制的实现，通过编写相应的starter类，来实现控制层的独立化。

1. 多组件分布式集群部署时，如何才能进行会话共享和保证屏蔽内部通讯负责逻辑，保障能够简单调用后台即时通讯API。

首先，即时通讯服务器是基于主内存数据网络框架Redission集群，在Redis数据库的发布订阅模式下，使得不同服务器之间的用户能够无障碍通信，而在即时通讯服务器与其他HTTP服务器的交流情况下，由于改动即时通讯服务器的难度高而使两者之前的会话共享变得尤其困难，为了解决这样的问题，就需要与平台无关的会话共享机制，使得用户在登录即时通讯服务器时也能够在HTTP服务器无障碍浏览，使得JWT机制成为了首选。在用户登录即时通讯服务器时，将部分有效可靠的信息写入Token当中，在用户将要去访问其他服务器时，携带该Token去访问即可。

1. 如何进行校园资讯可见范围的设置。

可见范围的设置与发送消息的所属机构有关，且层层递进。在班级发送时，消息的范围有：本班级，本学院，本校；社团机构发送的消息可见范围有：本部门，本机构，学院本校；此时对应于数据库的设计上，需要新增可见范围和年级筛选字段，并在用户部门表上添加父组织节点，对于具体的数据组织上，如果是本班级或者本部门范围，可见范围字段填写上本班级或者本部门的组织编号，查询时查看用户是否属于本部门或者本班级即可;如果时本机构、本学院时，查询时查看用户部分表上的父节点编号字段上数据是否匹配的上即可；如果是本校范围时，则无需认可的过滤查询条件综上所述，在查询校园资讯时，只需要将可见范围字段和可见组织编号分别与用户部门表上的匹配用户的本级部门编号和上一级部门编号进行或条件等值查询即可，如果有年级的过滤需求，则再外加年级的过滤条件。

1. 如何处理项目中已读和未读消息的消息显示

处理该类问题，必须要明确的，是未读消息到已读消息的转变流程。首先是发送方将信息发送至服务器，服务器将信息发送至指定的接收方，然后将发送结果发送至发送方，而当接受方已读时，则对未读用户编号列表中数据进行修改，再将信息发送至发送者，这时才将消息标志为已读状态。要处理该类问题，一定要明确，未读消息到已读消息的转变流程。首先是发送方将信息发送至服务器，服务器将信息发送至指定的接收方，然后将发送结果发送至发送方，而当接受方已读时，则将未读用户列表中数据修改，再将信息发送至发送者，这时才将消息标志为已读状态。在了解该流程后，就是在程序中如何实现的问题，这些过程的核心问题是消息的消息编号问题，考虑到消息编号在服务器生成将会大大增加交互的负责程度，因此消息编号将会在移动端进行生成，在保证消息编号的唯一性前提下，使得移动端在接收到该编号的聊天信息不再认为该条信息应该还需要重新在页面上进行渲染操作，从而将其标识为已发消息的一个未读消息的响应。

1. 如何实现树形结构数据的高效查询

对于此种树型结构数据采用编号制，用编号来表示数据的层级关系，可规定编号中每三位数字为表示单位，当编号只有三位数字时表示为根节点，此后每个属于该节点的子节点的编号将以本节点的编号和父节点编号的拼接组成，这样，在数据库查询中根据模糊匹配的特性，就能够快速查找出对应某节点的所有子节点信息。

# 结论

在本系统整个开发过程中本人主要完成的是四个模块：校园资讯模块、组织机构模块、即时通讯模块、个人信息模块。在校园模块中的功能主要有查看校园资讯、校园咨询的搜索和对校园资讯的提问和答疑；在组织模块中的功能主要有查阅目标的学生机构、学会社团等在校机构，在所属机构当中在用户能够进行调整社团机构成员、创建工作群、发布校园资讯等操作，而作为非所属该机构的用户能够查询该机构的基本信息，对该机构进行信息的反馈以及能和该机构成员进行交流等；在即时通讯模块中，用户能够既能同时和机构内成员进行有反馈的工作交流，也能够点对点地和目标用户信息日常交流。为了避免单机带来的信息处理瓶颈，通过Redis的发布订阅模式将即时通讯服务其并联起来，以集群的方式共同处理消息的存储和转发；在个人信息模块中，除了提供查看个人发布的校园资讯、个人针对某个组织提出的反馈意见之外，也提供了移动端的退出接口、移动端的基本配置。在通过本项目的设计和实现，使我对后台和前端开发知识有了更进一步的认识，也希望能够借助此项目给校园带来不一样的变化。

**参考文献**

1. 陈国君，陈磊. Java程序设计基础[M]. 清华大学出版社， 2015
2. 金焱，许建仁等. JavaEE企业级应用开发[M]. 东软电子出版社, 2012
3. 嘟嘟MD. Spring Boot干货系列：（一）优雅的入门篇[EB/OL].[2017-02-26]. <http://tengj.top/2017/02/26/springboot1/>
4. 乔鑫. "Nginx 新一代 web 服务器软件 [J]." 科技浪潮 1 (2009): 11-12.
5. 马豫星. (2015). Redis 数据库特性分析. 物联网技术, 3, 105-106.
6. Beaver D, Kumar S, Li H C, et al. Finding a Needle in Haystack: Facebook's Photo Storage[C]//OSDI. 2010, 10(2010): 1-8.
7. talent. t-io文档 [EB/OL].[2018-12-25]. <https://www.t-io.org/doc/tio/86>
8. Anderson C. Docker [software engineering][J]. IEEE Software, 2015, 32(3): 102-c3.
9. 林跃进，姜涛. Web前端开发技术与JavaScript框架编程[M].东软电子出版社， 2013
10. 京东凹凸实验室. Nerv 高性能前端框架[EB/OL].[2018-12-25].

http://ju.outofmemory.cn/entry/348222

1. 2016最新校园app排名[EB/OL]. [2015-11-18].

https://wenku.baidu.com/view/869ca3cba98271fe900ef93f.html

1. 李勇,杨晓军等. UML2软件建模入门与提高[M]. 清华大学出版社. 2015-1

**致 谢**

通过本次毕业设计的立项研究和开发，使本人具备了较好的调查与研究能力。本人的毕业设计从选题到基本顺利实现，离不开杨俊杰论文指导老师、东软工程师们以及同学们的关心、鼓励和支持，仅在此表示衷心的感谢！