



**本科毕业论文**

|  |
| --- |
| **基于微信平台的教务管理及服务系统** |
|  |

**邹毅锋**

**201330810228**

|  |  |
| --- | --- |
| 指导教师 | **刘昌余 讲师** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学院名称 |  | **数学与信息学院** | 专业名称 |  | **信息管理与信息系统** |
| 论文提交日期 |  | 2017年4月19日 | 论文答辩日期 |  | 2017年5月5日 |

摘 要

随着互联网的迅猛发展，人们获取信息的方式也越来越多样化。微信作为巨大的移动端流量入口，人们可以通过微信平台获取各中类型的信息，使得生活更加便利，同时也能够使得自己成为信息生产者，向特定人群推送他们所关心的信息，提升信息的转换和利用率。

基于微信的教务系统是一个web应用程序，针对我国中小兴趣教育机构的实际信息推广与教务需求的现实需求问题而设计、研发，具有一定的现实意义。系统在pc端可以通过对教育机构中的学生、教师、班级、课程等教务对象进行管理，并且在教务宣传方面，pc端通过发布校园动态、新闻，可利用微信平台进行传播，实现教务宣传，并且微信平台提供如留言问答，翻译等功能，进一步提升教务宣传功能。

该web应用后台开发主要使用Spring、Spring MVC和MyBatis等web应用框架，前端主要利用了vue、element-ui等框架，前、后端通信技术主要利用JSON技术。整个web应用设计合理，界面简洁而又不简单。

关键词：WEB 信息传播 教务管理 微信平台

**Educational Administration and Service System Based on WeChat**

Zou Yifeng

(College of Mathematics and Informatics, South China Agricultural University,

Guangzhou 510642, China)

**Abstract:** With the rapid development of the Internet, people get information more and more diversified. WeChat as a huge mobile side of the flow of the entrance, people can get through the Wechat platform for medium-sized information, making life more convenient, but also can make themselves as information producers, to specific groups of people to push their concerns, enhance information conversion and Utilization rate.

The WeChat-based teaching system is a web application, which is of practical significance for the practical needs of small and medium-sized educational institutions in China. The system can manage the students, teachers, classes, courses and other educational objects in the educational institution, and in the educational propaganda, the computer side through the release of campus dynamic, news, can use WeChat platform for communication, to achieve teaching and publicity, And WeChat platform to provide such as message Q & A, translation and other functions, to further enhance the educational function.

The Web application backend development mainly uses Spring, Spring MVC and MyBatis and other web application framework, the front-end development mainly uses the vue, element-ui and other framework, front-end and back-end communication technology mainly uses JSON technology. The entire web application design is reasonable, the interface is concise and not simple.

**Key words**:WEB Information dissemination Educational administration Wechat

目 录

华南农业大学本科生毕业论文成绩评定表

1 绪论

1.1 Spring

在互联网服务不断推陈出新，互联网移动化，社会网络化的现今，人们对于信息的接受途径变得越来越多样化。智能手机用户的爆炸式增长以及移动互联网的快速成长，极大地改变了行业竞争格局。传统互联网企业不断加强和完善移动互联网的布局，打造移动互联网产业链新模式（付亮，2013）。信息技术对教育发展具有革命性影响，用互联网思维推进校园宣传、管理，推动教育管理现代化（2016，孙志明）。

随着“互联网+”时代的到来，已经存在多年的传统儿童兴趣教育机构不得不做出新一轮的布局调整，抓住机遇，实现产业升级。传统模式下的儿童兴趣教育机构，其服务的方向主要有两个方面，一是对于孩子的兴趣的引导教育，二是针对现在年轻父母的工作时间长，无暇照顾孩子的情况，机构对儿童的拖带照看。立足于此，儿童兴趣机构的升级方向其一便是增加家校互动频率，机构主动向家长传递更多的孩子在机构学习成长的信息。

互联网时代儿童兴趣教育机构的困境与挑战，一是招生宣传手段落后，现阶段主要是依靠传单的方式宣传，通过数据调查显示，90%的儿童教育机构负责人反应，发传单做营销是现在最差的宣传方法。二是管理手段落后，不少机构还是依靠纸张记录的方式，去安排机构里的日常事务，时常会发生重要的数据丢失和翻查困难。通过数据调查反应，80%的儿童教育机构擅长的是给孩子带来高质量的课程，其若弱项是网络营销推广和教务管理，目前大部分教育机构未找到较好的方式结合互联网进行运营和管理。（李贤，2015）。

1.2 研究目的和意义

本文的研究目的是基于Web开发技术，结合微信平台，设计并开发实现一套适用于国内中、小儿童兴趣教育机构的教务系统，区别于常见的院校教务系统，该系统实现了针对机构中的学生、教师、班级、课程和教务新闻等的管理，并结合微信端实现了教务宣传、招生等功能，加大网络影响力。本文的研究意义是设计一套能解决当前儿童兴趣教育机构管理手段落后，宣传手段单一，家校互动方式匮乏等发展困境。随着“互联网+”的概念的深入人心，以及网络已经成为人们生活极大的组成部分，利用互联网技术手段，帮助机构实现教务管理升级，充分利用资源，使专注于提升教学质量。利用微信平台，传播机构动态，使得机构自身在网络上被更多的人发现，了解，也可以进一步提高机构与家长、社会之间的互动频率。

1.3 研究内容和技术方案

该项目主要有Java后台应用程序和Web前端组件两大部分组成。Web前端页面通过HTTP通信协议，访问已经提前通过花生壳域名映射工具对应的公网访问地址，再将请求映射到内网的Tomcat服务器端口，后台应用接收页面参数，访问并操作数据库，将数据以一定格式返回至浏览器端，将操作结果通过界面呈现给用户。

Java后台应用程序的软件架构设计，主要采用分层模式，由Controller、Service、DAO层组成，使得系统模块依照高内聚，低耦合的设计思想。Controller层主要利用拦截器，拦截客户端发起的HTTP请求，并生成具体的DTO类对象或实体类对象进行数据转换,并调用Service层的方法。Service层主要利用Controller中传递的对象，实现相关具体的业务逻辑和事务控制，并调用DAO层的方法，向上层的Controller层提供接口，并且使用DAO层提供的接口。DAO层主要负责数据库的增删查改，为上层的Service层提供接口。

Web前端开发的软件架构设计，主要采用组件化开发方式。组件化方式开发，将大块的业务界面，拆分为若干部分组件，其中包括对数据逻辑层的业务梳理，形成了不同层级的能力封装。组件功能可以单独开发测试，然后逐级拼装成更加复杂的组件。并且利用Webpack构建工具，实现静态资源的优化处理，如组件的动态加载，静态资源模块化。

微信公众平台是微信公众账号所有者（政府、媒体、企业、组织或个人）进行品牌推广，减小运营成本，提高影响力，与用户交流互动及提供服务的平台，公众账号通过消息，事件，菜单等交互方式为用户提供服务。其登陆方式既可通过Web端，也可通过客户端。

本系统的实现具体方案大致如下：技术调研，需求设计，系统架构设计，编码实现，系统调试、测试。

程序开发环境如下：

1. Java后台开发环境的搭建

在PC端搭建Java后台开发环境，其中包括JDK下载并且安装，配置CLASSPATH等环境变量，Tomcat服务器下载配置。

1. Web前端开发环境的搭建

在PC 端搭建nodejs开发环境，并安装npm包管理工具，配置Webpack构建工具，并启动Webpack静态服务器以及代理服务。

1. 微信端开发环境的申请和搭建

使用微信公众平台之前需要想注册账号，在注册过程中要进行信息登记，要选择相应的运营主体是组织还是个人。根据运营主体的不同，要求的提供不同的资质资料及证明。但运营主题为组织时，可以选择公众账号是订阅号还是服务号（郑永芹，2015）。

1. 开发相关工具的安装

在PC端安装Eclipse和Navicat for mysql数据库可视化操作软件。安装花生壳域名映射工具，将内网域名映射到公网，使得本地服务外网可访问。

2 相关技术

2.1 Spring

随着 Internet 的迅速发展，开发基于 Bowser/Server 的服务端的程序已经成为 Web 服务应用程序开发的重点。而传统的以 EJB为核心的 J2EE 架构对于大多数中小型企业应用来讲，显得过于复杂，而且其复杂的结构在开发过程中会降低开发效率， 增加开发难度。因此一个具有低侵入性，能够让具体的 J2EE 平台技术处于“松耦合”状态的轻量级的 Web 框架是迫切需要。Spring 框架的产生正式是满足了这种需要（陈鑫，2007）。

Spring框架提供了构造Web应用程序的全能MVC模块（Ben Alex etal，2012）。Spring MVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行制定。Spring MVC实现了 MVC的核心概念，它为控制器和处理程序提供了大量与此模式相关的功能。并且当向MVC添加反转控制时，它使应用程序高度解耦，提供了通过简单的配置更改即可动态更改组件的灵活性。

2.2 Spring MVC

Spring MVC采用是MVC的分层结构。MVC模式是软件工程中的一种软件架构模式，3个基本部分为Model、View、Controller。MVC模式会使应用程序产生分离（输入逻辑，业务逻辑和UI逻辑），同时提供这些元素之间的松耦合（Craig Walls，2011）。

（1）模型（Model）根据前端视图发送过来的数据，处理业务逻辑，然后返回处理结果。模型是整个模型的核心，封装了应用程序的数据，其中一般会组成的POJO类（Bruce Eckel，2003）。

（2）视图（View）是模型的外在表现，是和用户交互的界面，是应用系统和外界的接口，一方面接受用户输入的信息，一方面返回模型层处理的数据信息。视图是负责呈现模型数据。一般它生成的HTML输出，客户端的浏览器是能够解释。

（3）控制器（Controller）是模型和视图的联系纽带。接受用户发过来的请求，然后和模型，视图进行匹配，调用模型的方法，然后再根据返回模型处理后的结果，结合视图的页面进行渲染并显示给用户。负责处理用户的请求，并建立适当的模型，并把它传递给视图渲染。

2.3 MyBatis

MyBatis是一个Java持久化框架，它通过XML描述符或注解把对象与存储过程或SQL语句关联起来与其他的对象关系映射框架不同，MyBatis并没有将Java对象与数据库表关联起来，而是将Java方法与SQL语句关联。MyBatis允许用户充分利用数据库的各种功能，例如存储过程、视图、各种复杂的查询以及某数据库的专有特性。如果要对遗留数据库、不规范的数据库进行操作，或者要完全控制SQL的执行，MyBatis是一个不错的选择。与JDBC相比，MyBatis简化了相关代码：SQL语句在一行代码中就能执行。MyBatis提供了一个映射引擎，声明式的把SQL语句执行结果与对象树映射起来。通过使用一种内建的类XML表达式语言，或者使用Apache Velocity集成的插件，SQL语句可以被动态的生成。MyBatis与Spring Framework和Google Guice集成，这使开发者免于依赖性问题。MyBatis支持声明式数据缓存（declarative data caching）。当一条SQL语句被标记为“可缓存”后，首次执行它时从数据库获取的所有数据会被存储在一段高速缓存中，今后执行这条语句时就会从高速缓存中读取结果，而不是再次命中数据库。MyBatis提供了基于 Java HashMap 的默认缓存实现，以及用于与OSCache、Ehcache、Hazelcast和Memcached连接的默认连接器。MyBatis还提供API供其他缓存实现使用。

2.4 Vue

Vue是一款开源JavaScript库，用来协助单一页面应用程序运行的。它的目标是通过MVC模式（MVC）功能增强基于浏览器的应用，使开发和测试变得更加容易。

库读取包含附加自定义（标签属性）的HTML，遵从这些自定义属性中的指令，并将页面中的输入或输出与由JavaScript变量表示的模型绑定起来。这些JavaScript变量的值可以手工设置，或者从静态或动态JSON资源中获取。Vue是创建在这样的信念上的：即声明式编程应该用于构建用户界面以及编写软件构建，而命令式编程非常适合来表示业务逻辑。框架采用并扩展了传统HTML，通过双向的数据绑定来适应动态内容，双向的数据绑定允许模型和视图之间的自动同步。因此，Vue使得对DOM的操作不再重要并提升了可测试性。

Vue遵循软件工程的MVC模式，并鼓励展现，数据，和逻辑组件之间的松耦合。Vue为客户端的Web应用带来了传统服务端的服务，例如独立于视图的控制。因此，后端减少了许多负担，产生了更轻的Web应用。

2.5 MySQL

MySQL是一个小型关系型数据库管理系统，开发者为瑞典MySQL AB公司。在2008年1月16号被Sun公司收购。而2009年，SUN又被Oracle收购。对于Mysql的前途，没有任何人抱乐观的态度。目前MySQL被广泛地应用在Internet上的中小型网站中。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。

3 系统需求分析

3.1 总体用例分析

涉众和用例图如图1所示，涉众有注册用户（教师、学生）、访客和管理员，而访客为普通浏览网页的用户，并无账号登陆，注册用户定义为有账户的用户，可登录后台系统，管理员则是有账户的用户，并拥有系统的最高权限。图中描述了实际情况中访客时不能登录，注册用户可以查看个人相关教务信息，并修改个人信息，管理则能够使用系所有功能。

表1 需求统计表

|  |  |
| --- | --- |
| 参与 | 描述 |
| 访客 | 浏览校园新闻、师资力量等信息 |
|  | 参与问答留言 |
| 注册用户 | 查看、修改个人信息 |
| 查看个人所在班级的课程表 | |
| 管理员 | 新增、修改、删除班级，并为班级指派学生、教师，安排课程  为已有的班级发布班级动态  发布、删除新闻 |
|  | 新增、修改、删除学生，为学生指派班级 |
|  | 新增、修改、删除老师，为老师指派班级 |
|  | 回复、删除留言 |

表1 用例统计表

|  |  |
| --- | --- |
| 用例名称 | 描述 |
| 登陆系统 | 用户在登录系统时，输入账号，密码，用户类型 |
| 创建班级 | 管理员新增班级，为班级指派学生，任课老师，安排课程等 |
| 新增学生 | 新增学生对象，并可以为其安排班级， |
| 推送动态 | 管理员新增一条机构动态时候，微信端会自动推送一条消息 |
| 留言回复 | 当访客在pc端访问留下问题，管理员可以在后台对问题进行回复 |

3.2 系统主要用例

详述用例（fully dressed use case）可以结构化地、更加深入地展示了系统应用的各种场景（张海藩，2003）。以下将采用活动图的方式系统的主要用例流程进行详细描述。

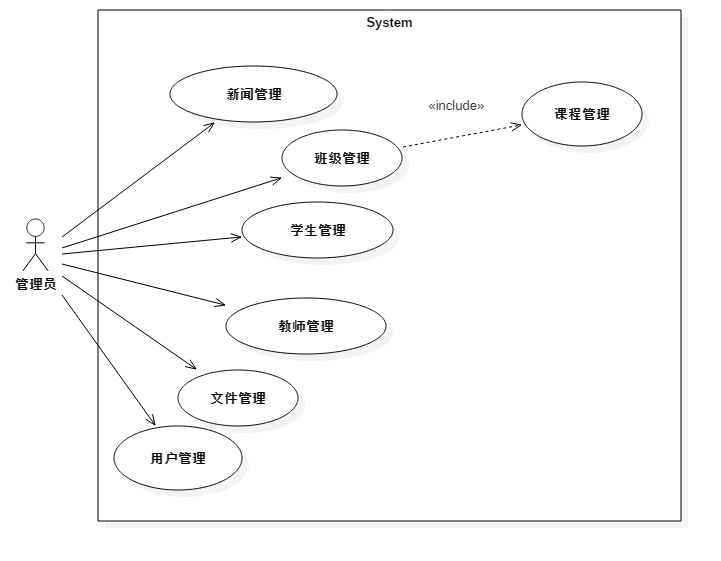


图 1 管理员用例图

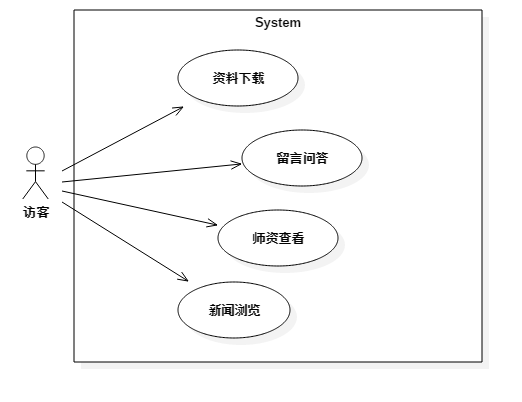
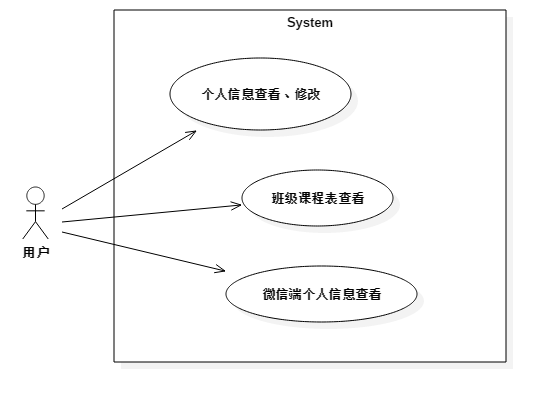


图 2 访客用例图



3.2.1 登录系统用例

登陆用例活动图如图3所示，用户登录的时候，如果填入的用户名，密码或者类型不正确的时候，后台无匹配到对应的数据时，则会页面上提示错误。如果验证正确，则会根据用户对应的类型，分配不同的权限，使得所呈现的页面不同。

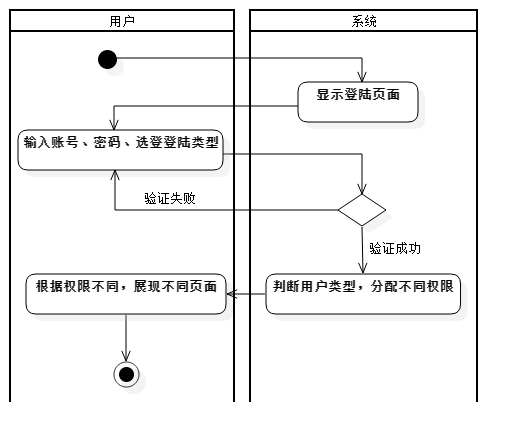


图 3登录用例活动图

3.2.2 创建班级用例

创建班级用例活动图如图所示，管理员登陆后台成功，选择班级管理选项后即可创建班级。管理员依次填写班级名称，班级介绍，是否当天开班选项等基本信息后，选择保存，若选择当天开班，这页面跳转至课程管理排课页面，若没有选择当天开班，则提示班级创建成功，但班级并无课程存在。

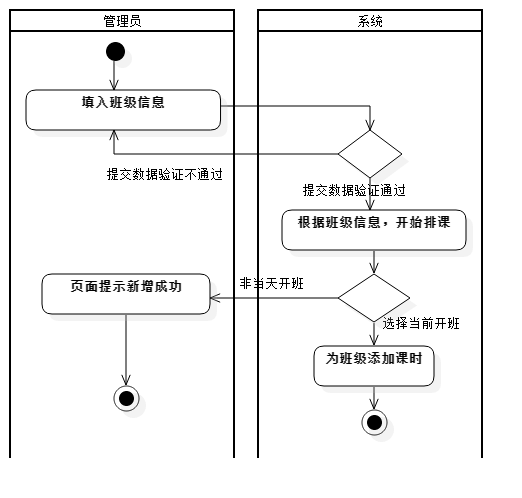


图 4 创建班级活动用例图

3.2.3 班级排课用例

管理登陆登陆后台成功后，选择为班级管理选项。点击课程管理选项时候，即可以为班级建立课程。依次填写课程名称，课时数量，选择每个星期要上课日期和课程简介，系统会自动根据课时数，为班级日后每周的特定日期生成课时。当点击保存，提示保存成功后，页面会跳转到当前的星期的课程表，页面也会展示最新的课程。若提交数据中，存在与后台数据的冲突的课程，则提示保存不成功。

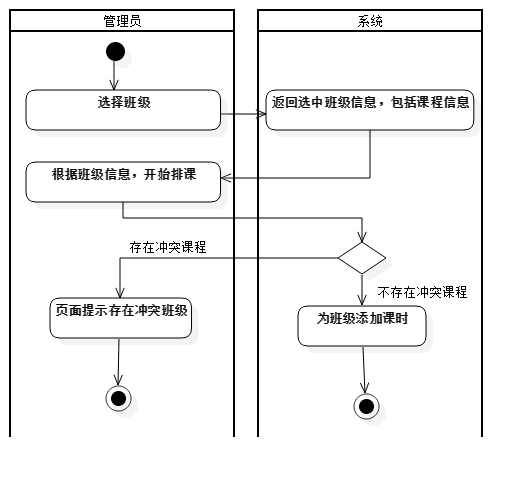


图 5 班级排课活动用例图

3.2.4 推送动态用例

推送动态用例图如图所示，当管理员在后台发布一条新闻，选择新闻类型，填写作者，新闻内容时候，点击保存后，所关注微信公众号的用户，其微信账号中将会接收到一条关于动态详情的模板消息，并从模板的链接中，阅读新闻详情。

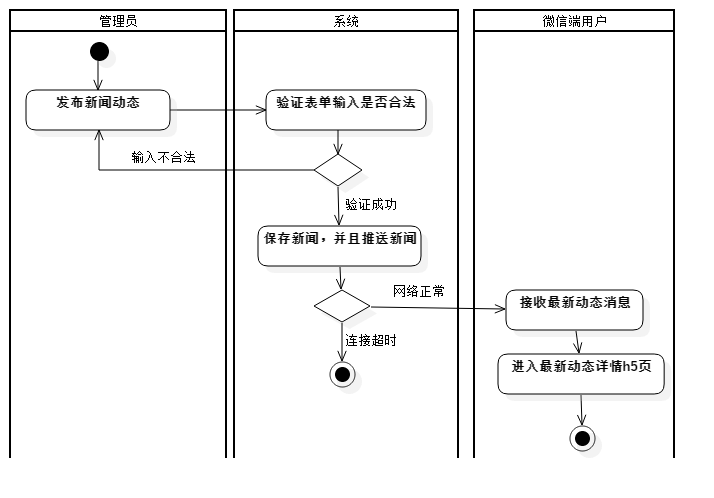


图 6 推送动态活动用例图

3.2.5 留言回复用例

留言回复用例活动图如图所示，用户通过在前端展示页面上输入所关心的问题，管理员通过登陆后台，进入留言管理选项，可以查看全部、已回复、未回复的消息，并回复和删除用户问题。

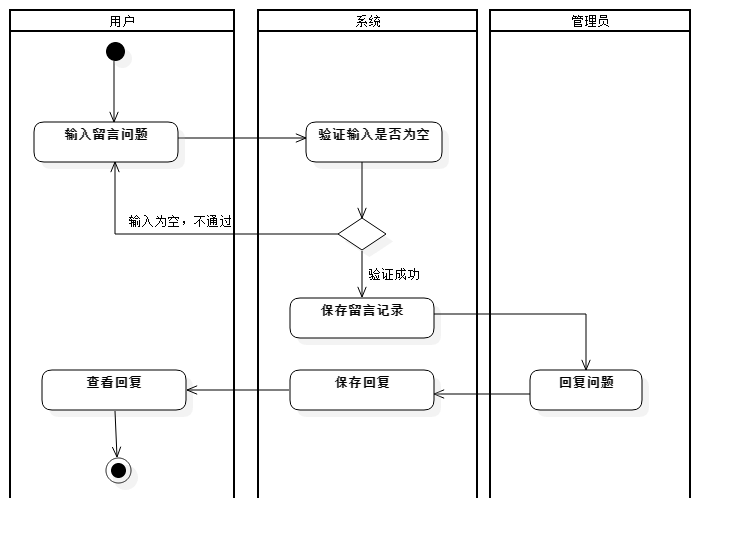


图 7 留言回复活动用例图

4 系统概要设计

4.1 系统体系结构

本系统整体采用3层B/S架构，分为客户浏览器终端、微信服务器、数据服务器端，通信基于HTTP协议，数据格式为XML、JSON。微信客户终端（手机、平板等）发起请求至微信服务器，微信服务器将请求转发给服务器端，服务器讲数据处理完毕后返回给微信服务器，微信服务器则返回给客户终端。

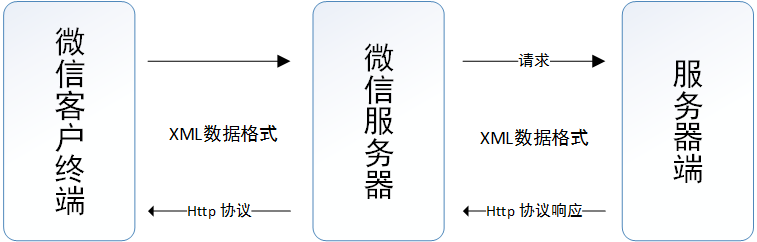


图 8 微信客户端与服务器通信

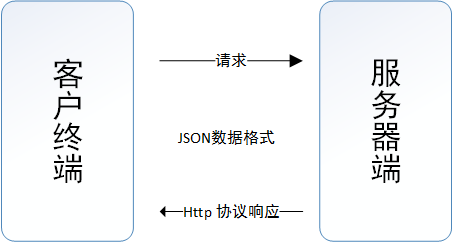


图 9 浏览器客户端与服务器端通信

1. 服务器端

位于系统架构的最底层，提供数据的解析、处理、封装和传输，本系统使用的是本地Tomcat服务器。服务器根据微信服务器推送过来的客户终端消息类型和内容进行解析，同时对其进行处理和回应；将回应消息按照微信服务器相应的格式进行封装，发送个服务器。

1. 微信服务端

负责实现对特别群体的消息的全方位够沟通，互动，就相当于一个转发服务器。

1. 客户终端

客户终端直接与用户互动，并负责为用户展现结构，带来良好的使用体验。主要采用三层架构：

表现层：与用户良好交互的UI设计。

逻辑层：实现系统的各种逻辑功能，为表现层提供数据访问等功能服务。

数据层：为逻辑层提供数据处理功能。

4.2 系统模块划分

本系统前期调研现主流管理系统和教务系统，挖掘贴切实际的产品需求，并设计出合理、人性化的UI界面。本系统主要分为两大模块，前端展示模块和后台管理模块。前端展示模块有可以分为微信端和PC端两大模块。每个模块都具有不用的功能和其合理的设计初衷，从良好的用户体验和贴近真实的用户场景考虑出发。系统功能结构图如图8所示，展示了本系统所具有的模块以及其具体的详细功能。

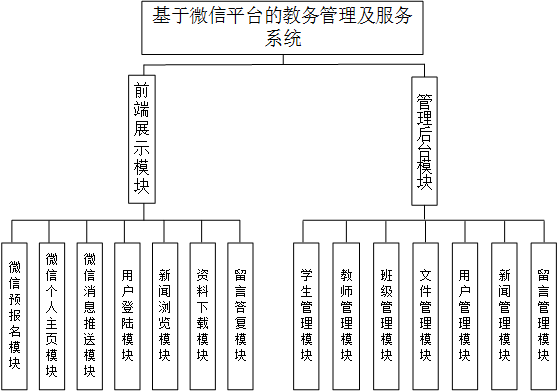


图 10 系统功能结构图

4.2.1 系统前端展示模块

该系统的前端PC端展示页主要功能包括轮播图、用户登陆模块，班级动态列表，交流成果列表，资料下载模块，留言问答模块等。前端展示模块总体结构图大致如图9所示。

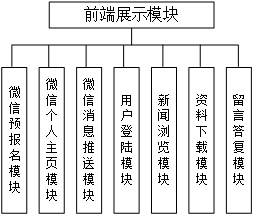


图 11 前端展示模块

1. 用户登陆模块：用户选择自己的身份类型，并输入自己的账号，密码，实现用户登陆
2. 班级、交流成果动态：选取机构最新发布的不同类型的动态新闻，点击“详情”能够查看动态详情内容。
3. 留言问答模块：访客可以将自己心中感兴趣的问题留言，向系统管理员反馈

该系统的前端微信移动端展示页主要功能包括模板消息、自动回复、最新动态、翻译小工具、学生主页、教师主页、招生预报名

1. 模板消息：当管理员在后台发布一条新闻动态，公众号的粉丝都会接受到一条关于这动态的模板消息。
2. 自动回复：根据用户在输入框输入并发送的值，动态返回数据
3. 最新动态：用户在公众号下方点击“新动态”按钮，返回最新发布的一条教务新闻。
4. 翻译小助手：用户在输入框输入期待翻译的字、词，后台调用百度翻译API，返回译文。
5. 学生、教师主页：根据用户微信账号的openId与后台的学生、教师账号绑定，返回教师、学生的个人信息和课程信息。
6. 招生预报名：通过预报名入口，用户填写相关信息，完成预报名。

4.2.2 系统后台管理模块

该系统的后台管理模块主要功能包括新闻管理、学生管理、教师管理、班级管理、文件管理、用户管理等功能。后台管理模块总体结构图大致如图10所示。

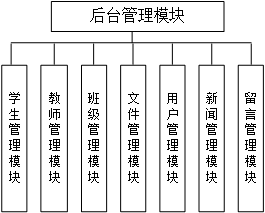


图 12 后台管理模块

1. 新闻管理：管理员可以通过选择不同的新闻类型，对新闻进行发布，删除，修改。每次发布一条新闻，关注公众号的粉丝都会收到一条消息推送，推送的详情内容即是新闻详情。
2. 学生、教师管理：管理员可以通过对学生、教师进行添加、删除、修改。新增学生、老师时，可以同时为其报班级，
3. 班级管理：管理员可以通过对班级进行添加、删除、修改、发布新闻、指派学生或教师、安排课程。安排课程后，即可以生成课程表。
4. 文件管理：管理员可以通过选择不同的文件类型，然后从本地选择文件上传。上传完毕后，前端展示页根据类别，分别展示不同类型的文件。
5. 用户管理：管理可以添加删除学生、教师用户，和添加管理员用户。学生、教师类型用户在创建时会自动生成对应角色的用户。此处可创建的角色是管理员用户。
6. 留言管理：管理员可以对前端展示模块的留言问题进行回答。成功恢复留言后，前端展示模块即可以查看到管理员最新的回复。

4.3 接口设计

该系统的数据持久层中，将通用的增加、删除、查询和修改的公共操作提取出来，封装成一个高度通用的BaseDao接口，只要继承该接口，即可调用接口中的方法，大大减少重复代码。表列出了接口的组成。

表18 BaseDaoI接口

|  |  |  |  |
| --- | --- | --- | --- |
| 方法与参数 | 返回值 | 功能描述 | |
| selectAll() | List<T> | 返回所有对象 | |
| selectSelective(T t) | List<T> | 返回所有符合条件的对象 | |
| selectSelective(T t , int pageNum, int PageSize) | List<T> | 分页获取符合条件的对象 |
| selectByPage(int pageNum, int pageSize) | List<T> | 分页获取数据 | |
| selectByPrimaryKey(int id) | T | 通过id得到实体类 | |
| update(T t) | int | 更新实体类 | |
| insert(T t) | int | 插入一条数据 | |
| delete(int id) | int | 通过带参hql得到一个对象 | |
| deleteSelective(T t) | Int | 根据对象删除 | |

该系统的前端请求接口中，都会将数据放在一个对象的data字段中，而后天接收请求的时候，通过封装一个JsonRequest对象，其具有一个Object类型的data成员变量用于接收页面参数，再利用com.fasterxml.jackson.databind.ObjectMapper类中实例方法convertValue([Object](eclipse-javadoc:%E2%98%82=wechat-education-system/C:%5C/Users%5C/kyd%5C/.m2%5C/repository%5C/com%5C/fasterxml%5C/jackson%5C/core%5C/jackson-databind%5C/2.5.2%5C/jackson-databind-2.5.2.jar%3Ccom.fasterxml.jackson.databind(ObjectMapper.class%E2%98%83ObjectMapper~convertValue~Ljava.lang.Object;~Ljava.lang.Class%5C%3CTT;%3E;%E2%98%82java.lang.Object) fromValue, [Class](eclipse-javadoc:%E2%98%82=wechat-education-system/C:%5C/Users%5C/kyd%5C/.m2%5C/repository%5C/com%5C/fasterxml%5C/jackson%5C/core%5C/jackson-databind%5C/2.5.2%5C/jackson-databind-2.5.2.jar%3Ccom.fasterxml.jackson.databind(ObjectMapper.class%E2%98%83ObjectMapper~convertValue~Ljava.lang.Object;~Ljava.lang.Class%5C%3CTT;%3E;%E2%98%82java.lang.Class)<[Admin](eclipse-javadoc:%E2%98%82=wechat-education-system/C:%5C/Users%5C/kyd%5C/.m2%5C/repository%5C/com%5C/fasterxml%5C/jackson%5C/core%5C/jackson-databind%5C/2.5.2%5C/jackson-databind-2.5.2.jar%3Ccom.fasterxml.jackson.databind(ObjectMapper.class%E2%98%83ObjectMapper~convertValue~Ljava.lang.Object;~Ljava.lang.Class%5C%3CTT;%3E;%E2%98%82com.laboratory.po.Admin)> toValueType),将data对应的对象转为具体的实体类。

5 系统详细设计

5.1 数据库设计

关系数据库设计是对数据进行组织化和结构化的过程，核心问题是关系模型的设计。在设计和操作维护数据库时，关键的步骤就是要确保数据正确地分布到数据库的表中。 使用正确的数据结构，不仅便于对数据库进行相应的存取操作，而且可以极大地简化应用程序的其他内容。数据应该尽可能少地冗余，这意味着重复数据应该减少到最少。如果存在过多的冗余数据，这就意味着要占用了更多的物理空间，同时也对数据的维护和一致性检查带来了问题。

5.1.1 E-R图设计

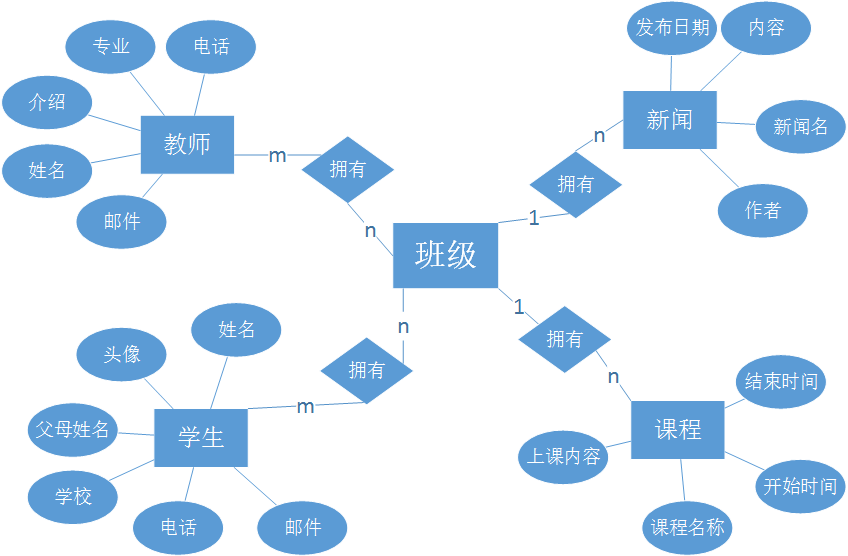


图 13 班级管理模块ER图

班级管理模块ER图如图所示，班级实体与其他实体，如学生、教师、课程都要有相关联的关系。在一家教育机构中，一个班级会存在多个学生，一个学生也可以选择都多个班级。同理老师与班级关系如此。所以具体到表的设计上，建立一张中间表，将多对多的关系关联起来。班级与课程的关系是一对多的关系。一个班级可能存在多个课程。同理，一个班级会发布多条新闻，班级与新闻的关系是一对多的关系。

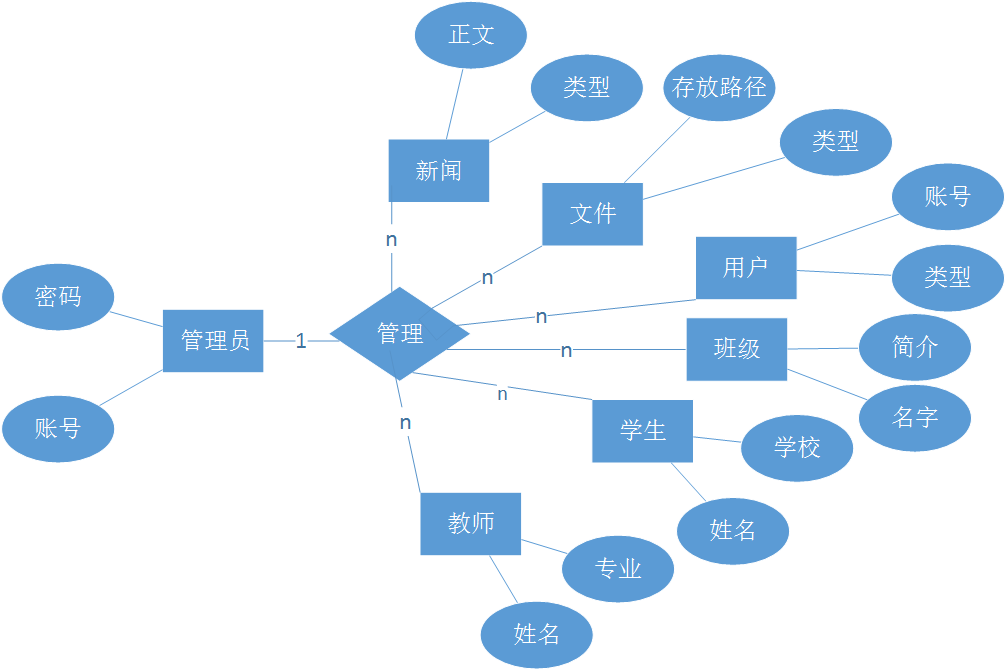


图 14 后台管理模块ER图

后台管理模块主要涉及的对象主要包括管理员，新闻，文件，用户，班级，学生，教师。管理员对这些对象都具有增删查改的权限，并且管理员对象与这些对象的关系是一对多的关系。

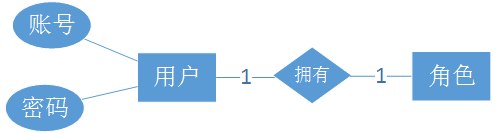


图 15 系统用户角色图

该教务系统的角色分为两类，一是管理员，二是用户（教师、学生）。不同的角色具有对系统不同等级的使用权限。

5.1.2 数据库表设计

数据库表的关系如图示。根据实体关系的分析，总共设计出10张表。

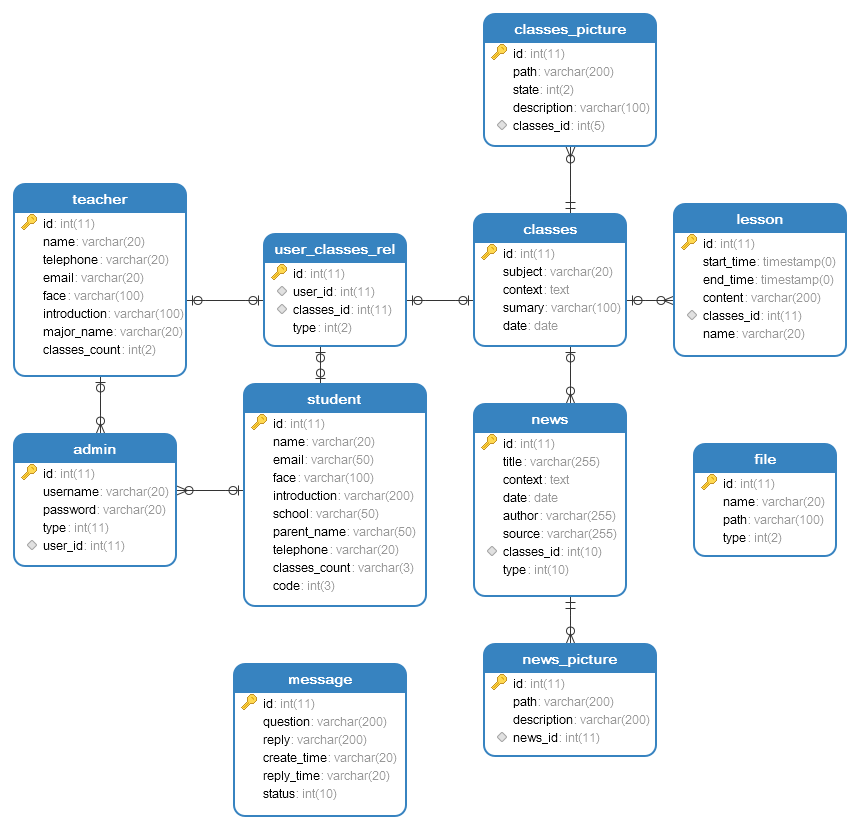


图 16 数据库关系表

1. admin表存储系统的用户信息，包括学生，教师和管理员，每个学生或老师都有一个后台登陆账号。一对一的关系关联student表或teacher表，关联字段为userId，外键为id.
2. Classes表为班级表.，因为学生、教师与班级是多对多的关系，因此产生一张中间表user\_classes\_id，记录学生、老师和班级的关系，关联字段为user\_id，其中type =1时为学生，type=2时为教师，外键为id；
3. file表为文件表，存储管理上传的文件信息，path为存储路径，type为文件类型。
4. lesson为课程表，与班级为多对一关系，关联字段为classes\_id，外键为classes表的id字段
5. news表为新闻表，用于记录所发布的新闻，type为不同类型的新闻，其中一种类型是班级活动，所以新闻与班级的关系是多对一，其关联字段是classes\_id，外键是classes表的id字段。
6. news\_picture表是新闻图片表，用于记录发布新闻中所上传的图片。其与新闻是一对多的关系，关联字段是news\_id，外键是news表的id字段。
7. student表、teacher表分别是学生表和教师表。用于记录学生、教师实体的信息。

5.2 业务流程设计

5.2.1 用户登陆

事先已经在管理后台注册成功的用户（学生，老师，管理员），在进入管理后台的时候，需要进行登陆操作。用户登陆顺序图如图所示，类图如图所示。用户在首页输入了用户名、密码并选择登陆身份后。页面脚本会进行校验，验证输入的字段是否符合符合规范。如果不符合，系统会提示错误信息，并不允许提交请求。校验通过后，LoginController会拦截提交请求，并将提交参数中的data字段转换为Admin实体类对象，并通过AdminService的findByPrinciple的方法，查找是否有对应的符合账号、密码、身份类型的用户如果存在，则将查找出来的Admin对象，封装在JsonResponse对象里，向客户端返回Json对象。若无查找出对应的Admin对象，则返回空对象。与此同时，后台会在session对象中添加user属性，并将查找出的Admin对象存入session，后台通过配置拦截器，如果请求类似“/admin/\*.action”，即是设计后台数据的操作，该请求都会通过被拦截，通过判断session是否过期或session中是否存在对应的Admin对象。若无，则会返回“登陆错误”提示，浏览器端便会跳转至首页，提示重新登陆账号。

客户端在接受到响应的结果后，对结果进行一定的逻辑判断，操作。若返回的对象中包含id属性，则会成功登陆管理后台，再根据不同用户身份类型，分配不同的权限，呈现不同的页面功能。

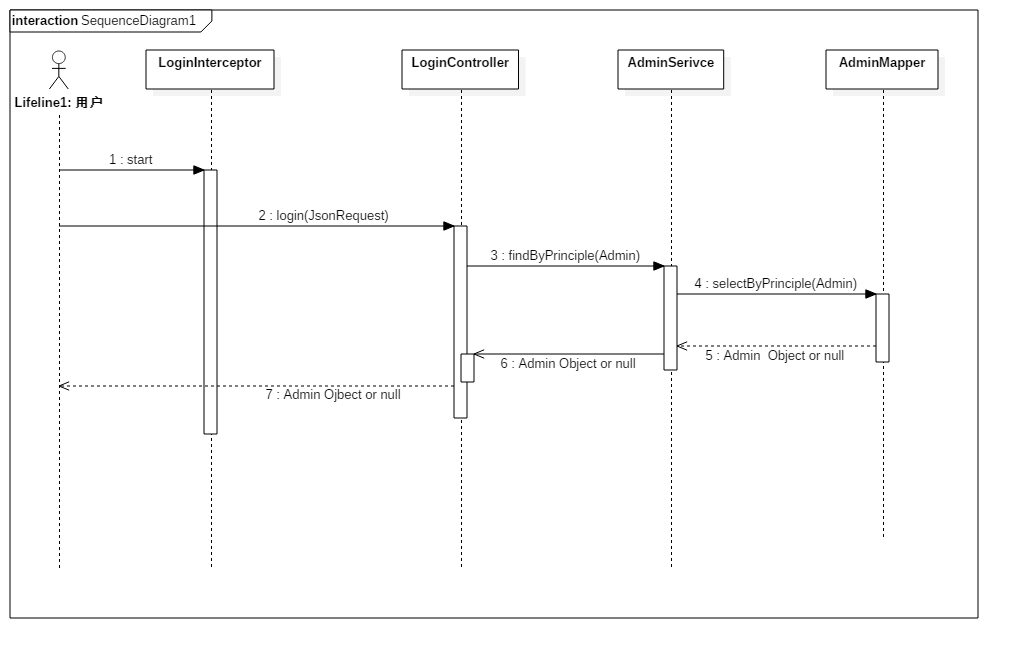


图 17 用户登陆顺序图

5.2.2 微信端用户登陆

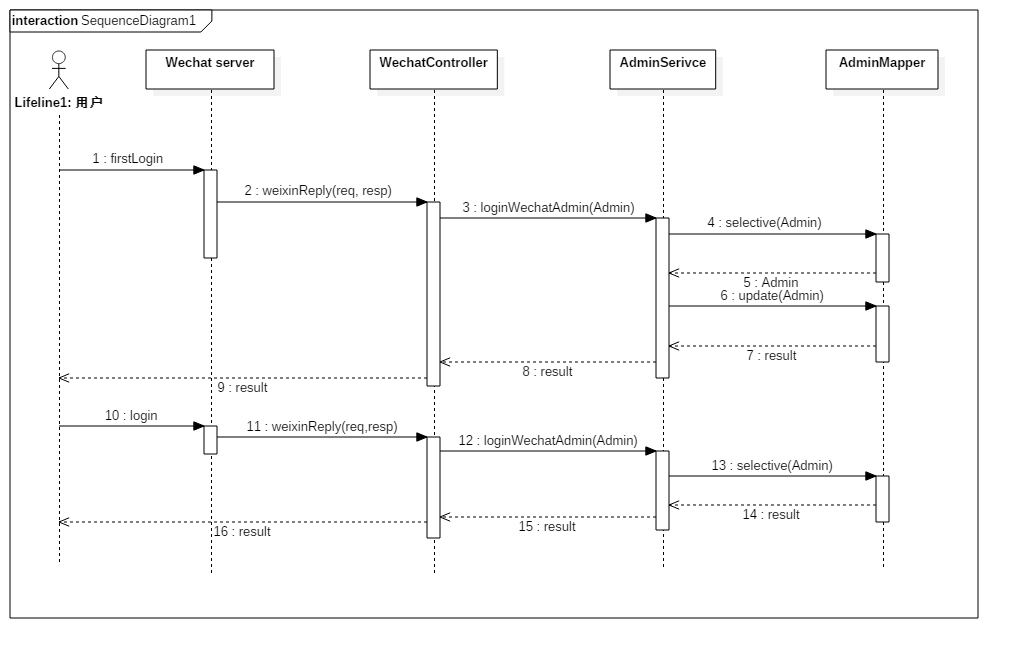
微信端用户登陆，当用户在微信订阅号中的点击底部按钮“个人主页”时，用户的操作会使得微信服务器通过事件推送的形式通知到开发者在开发者中心处设置的服务器地址，后台服务器判断出事件类型为“VIEW”类型，并自动跳转至服务器上的“#/personalIndex”路由页面，微信服务器并会把ToUserName、FromUserName、CreatTime、MsgType、Event、EventKey 参数传回至本地服务器上。用户首次访问个人 的时候，需要输入对应的账号密码，WechatController接收到账号名、密码和身份类型，通过调用AdminService的findByPrinciple方法，找出是否存在对应的用户。如存在，则同时将用户的openId入库对应的用户的openId字段。通过存在的Admin对象的userId和type的成员变量，找出微信访问的对应的学生或老师对象，并将老师或学生对象的个人信息和所在班级的课程表返回至个人信息主页。

图 18 微信端用户登陆顺图

5.2.3 新闻浏览

对访客访问主页时，页面脚本发起ajax请求，后台通过 newsController拦截 请求，并将请求参数转换为 实体类对象，在调用 Service中的 方法，根据时间降序排列和不同的时间类型，查找最新发布的 条新闻，通过封装成JosnResponse对象，返回至客户端。

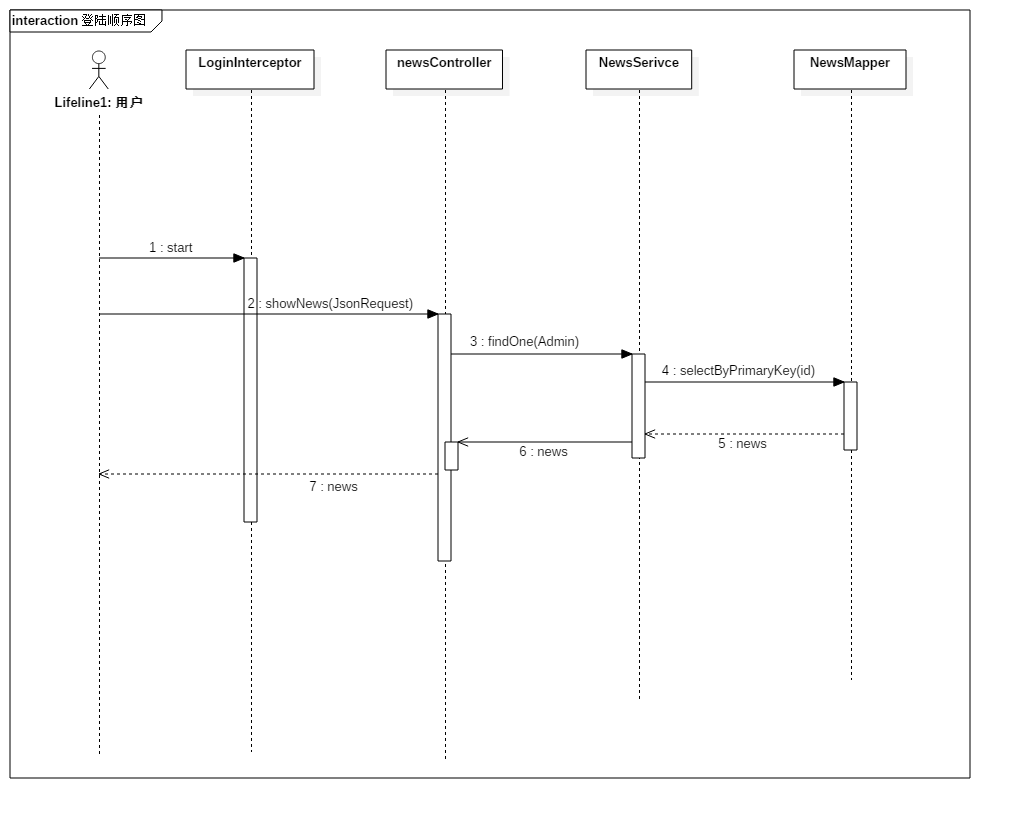


图 19 新闻浏览顺序图

5.2.4 微信端消息推送

当管理员在后台发布一条新闻动态时，页面脚本将数据通过ajax 发起请求，newsController接受请求，将数据转换为news对象，调用newsService的saveNews方法，保存成功后，调用wechatMessageUtil方法的send\_template\_message方法，群发模板消息。

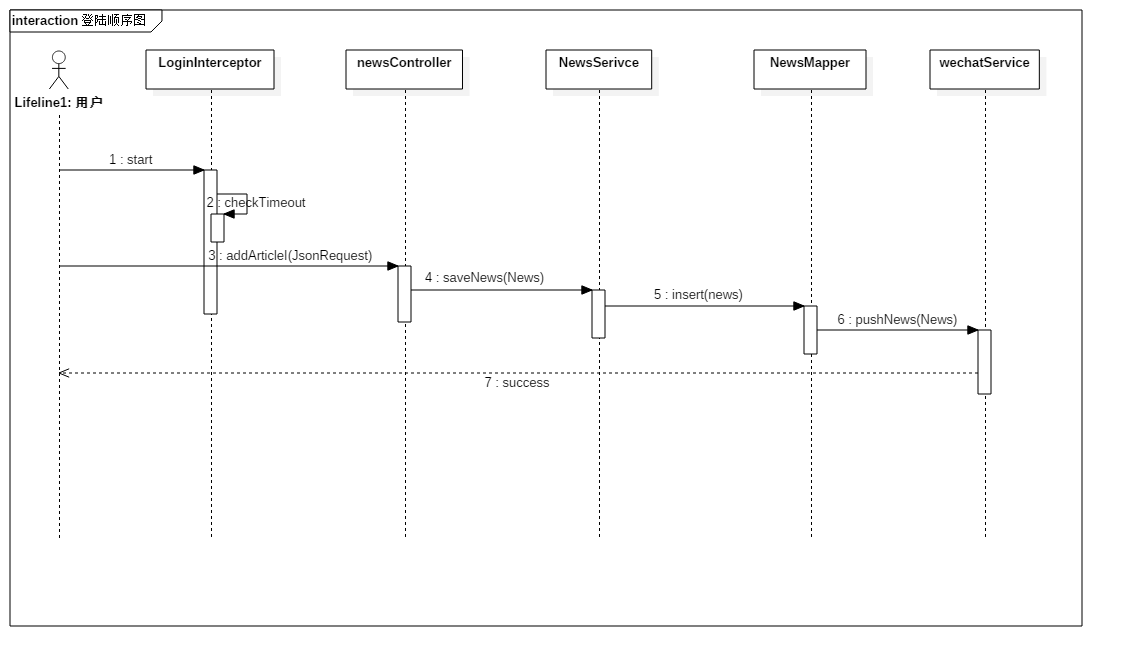


图 20 微信端消息推送顺序图

5.2.5 访客留言

访客在浏览首页的时候，在联系我们的板块，输入所关心的问题，点击提问按钮。通过ajax发起请求“message\_add.action”，对应的messageController拦截到请求，通过JsonRequest接收参数，并通过getData方法，将对象转换message对象，在调用 Service中的 方法，向数据库中插入一条记录。

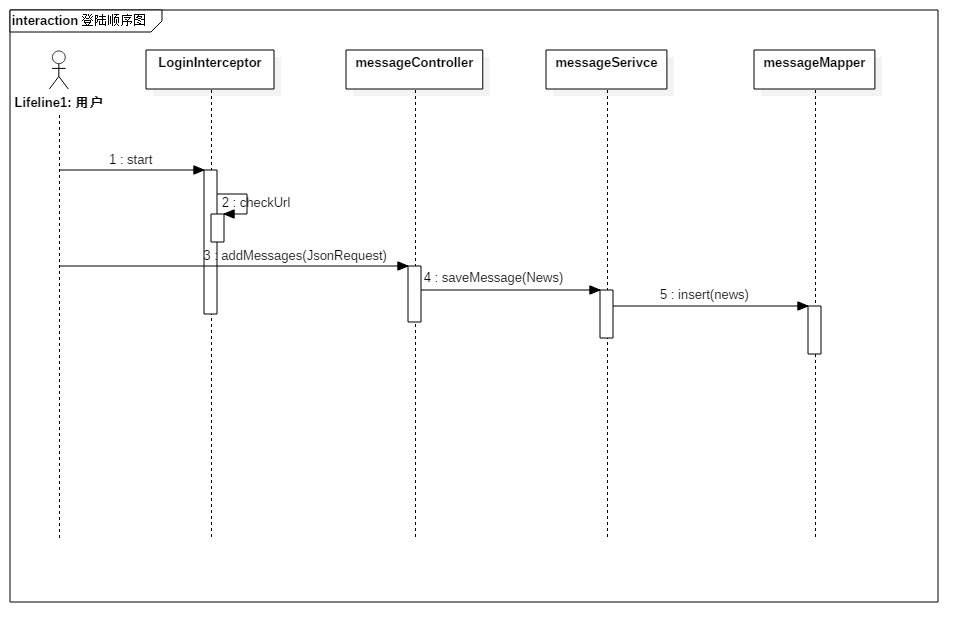


图 21 访客留言顺序图

5.2.6 新闻管理

管理员在登陆后进入新闻管理页面，前端页面发送请求“/admin/news\_list.action”，newsController接收请求，newsController中用JsonRequest对象接收实体类对象，PageInfo对象接收分页数据，并将参数转换为对应的news实体对象。调用NewsService的方法findByPage方法，返回对应的新闻分页数据。

查询新闻，管理员通过在后台页面上输入任一标题（title）或作者 (author) 字段，前端发送请求“/admin/news\_select.action”到newsController，newsController接收请求参数，转换为对应的news对象和pageInfo参数，并调用newsService的findNewsByCondition方法，传入news参数和pageInfo参数，newsService调用newsMapper中selectSelective方法，并分页地返回对应选取的数据。

编辑新闻，管理员选择要编辑的新闻，前端发起请求“/news.action”,newsController接受参数并将参数数据转换为news对象，newsService调用findOne方法，newsService完成逻辑校验后，调用newsMapper方法的selectByPrimaryKey，查找对应新闻的详情数据。管理员编辑完成后，前端再发起请求“/admin/news\_add.action”,newsController中完成逻辑校验，判断是否存在news的id，再调用newsService中的updateNews方法，完成更新新闻，并向前端返回成功更新提示。

删除新闻，管理员选择删除新闻，前端发起请求“/admin/news\_delete.acion”, newsController接受参数并将参数数据转换为news对象，newsService调用deleteNews方法，newsService完成逻辑校验后，调用newsMapper方法的delete方法完，完成删除新闻，并向前端返回成功删除提示。

发布新闻，管理员选择发布新闻，当管理员选择上传图片，上传存储使用的是七牛云存储，选择上传一张图片的时候，前端发起请求“/admin/news\_pic\_upload.action”,newsController利用MultipartFile文件格式接受图片文件，并上传至本地Tomcat服务其上的“/resource/news”目录，在调用UploadUtil的uploadPicture方法，读取本地服务器上的图片，调用七牛提供的API，创建QiniuImageUpload对象，调用uploadImage方法，上传至七牛服务器上，并返回存储路径。当管理员提交时，前端发起请求“/news/news\_add.action”, 前端再发起请求“/admin/news\_add.action”,newsController中完成逻辑校验，判断是否存在news的id，再调用newsService中的saveNews方法，完成新增新闻，并向前端返回成功新增提示。

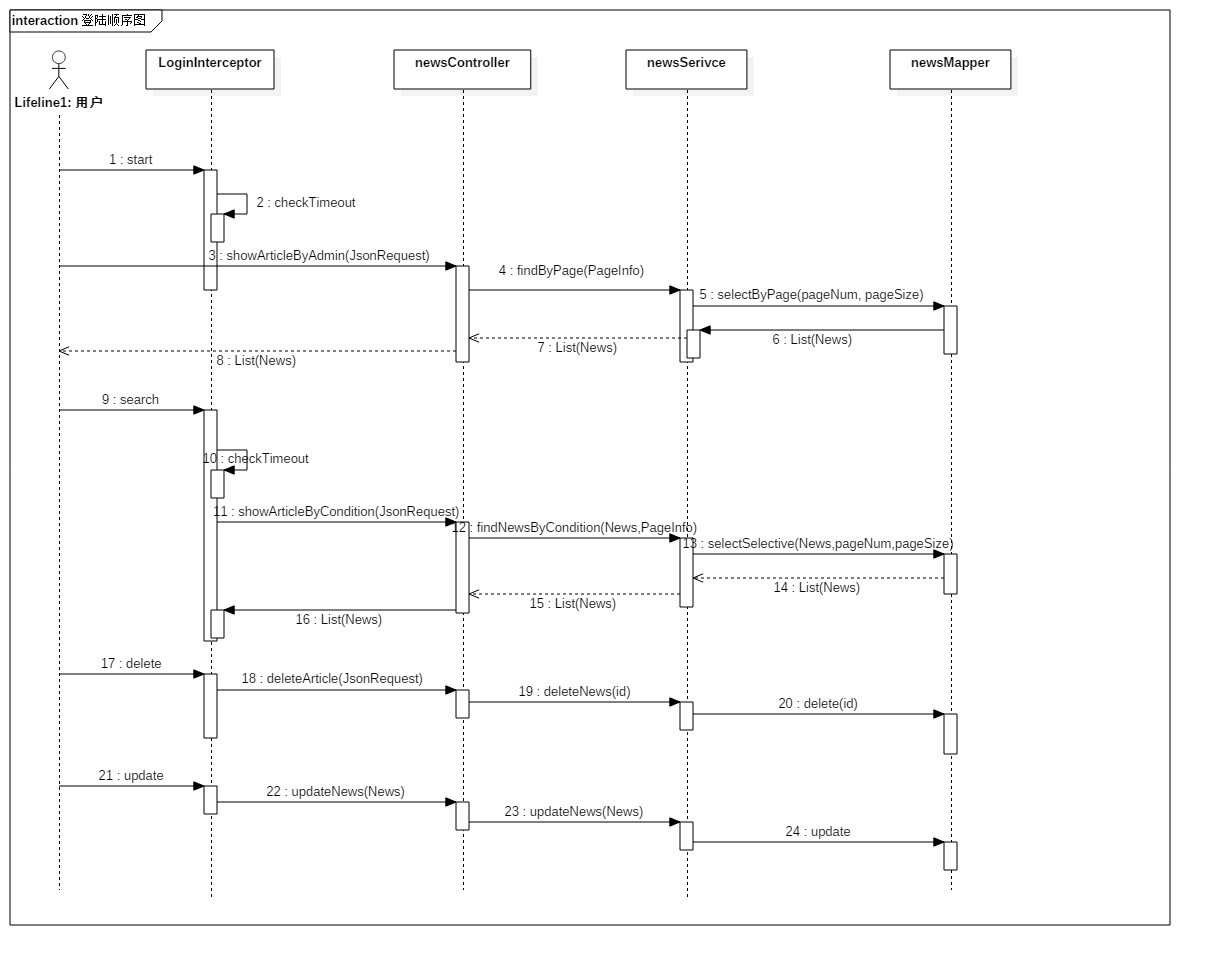


图 22 新闻管理顺序图

5.2.7 学生管理

管理员在登陆后进入学生管理页面，前端页面发送请求“/admin/student\_list.action”，studentController接收请求，studentController中用JsonRequest对象接收,并转换为student实体类对象，PageInfo对象接收分页数据。调用studentService的方法findByPage方法，返回对应的学生分页数据。

查询学生，管理员通过在后台页面上输入任一姓名（name）或就读学校 (school) 字段，前端发送请求“/admin/student\_select.action”到studentsController，studentController接收请求参数，转换为对应的student对象和pageInfo参数，并调用studentService的findStudentByCondition方法，传入student参数和pageInfo参数，studentService调用studentMapper中selectSelective方法，并分页地返回对应选取的数据。

编辑学生，管理员选择要编辑的学生，前端发起请求“/student.action”,studentController接受参数并将参数数据转换为student对象，studentService调用findOne方法，studentService完成逻辑校验后，调用studentMapper方法的selectByPrimaryKey，查找对应新闻的详情数据。管理员编辑完成后，前端再发起请求“/admin/student\_add.action”, studentController中完成逻辑校验，判断是否存在student的id，再调用studentService中的updateStudent方法，完成更新新闻，并向前端返回成功更新提示。

删除学生，管理员选择删除学生，前端发起请求“/admin/student\_delete.acion”, studentController接受参数并将参数数据转换为student对象，studentService调用deleteStudent方法，studentService完成逻辑校验后，调用studentMapper方法的delete方法完，完成删除学生，并向前端返回成功删除提示。

新增学生，管理员选择新增学生，当管理员选择上传学生头像，上传存储使用的是七牛云存储，选择上传一张图片的时候，前端发起请求“/admin/member\_pic\_upload.action”,studentController利用MultipartFile文件格式接受图片文件，并上传至本地Tomcat服务其上的“/resource/face”目录，在调用UploadUtil的uploadPicture方法，读取本地服务器上的图片，调用七牛提供的API，创建QiniuImageUpload对象，调用uploadImage方法，上传至七牛服务器上，并返回存储路径。当管理员提交时，前端发起请求“/admin/student\_add.action”, studentController中完成逻辑校验，调用studentService中的addStudent方法，同时也调用adminService的insertService方法，为该学生分配用户账号。然后返回相应数据，客户端根据响应提示新增成功。

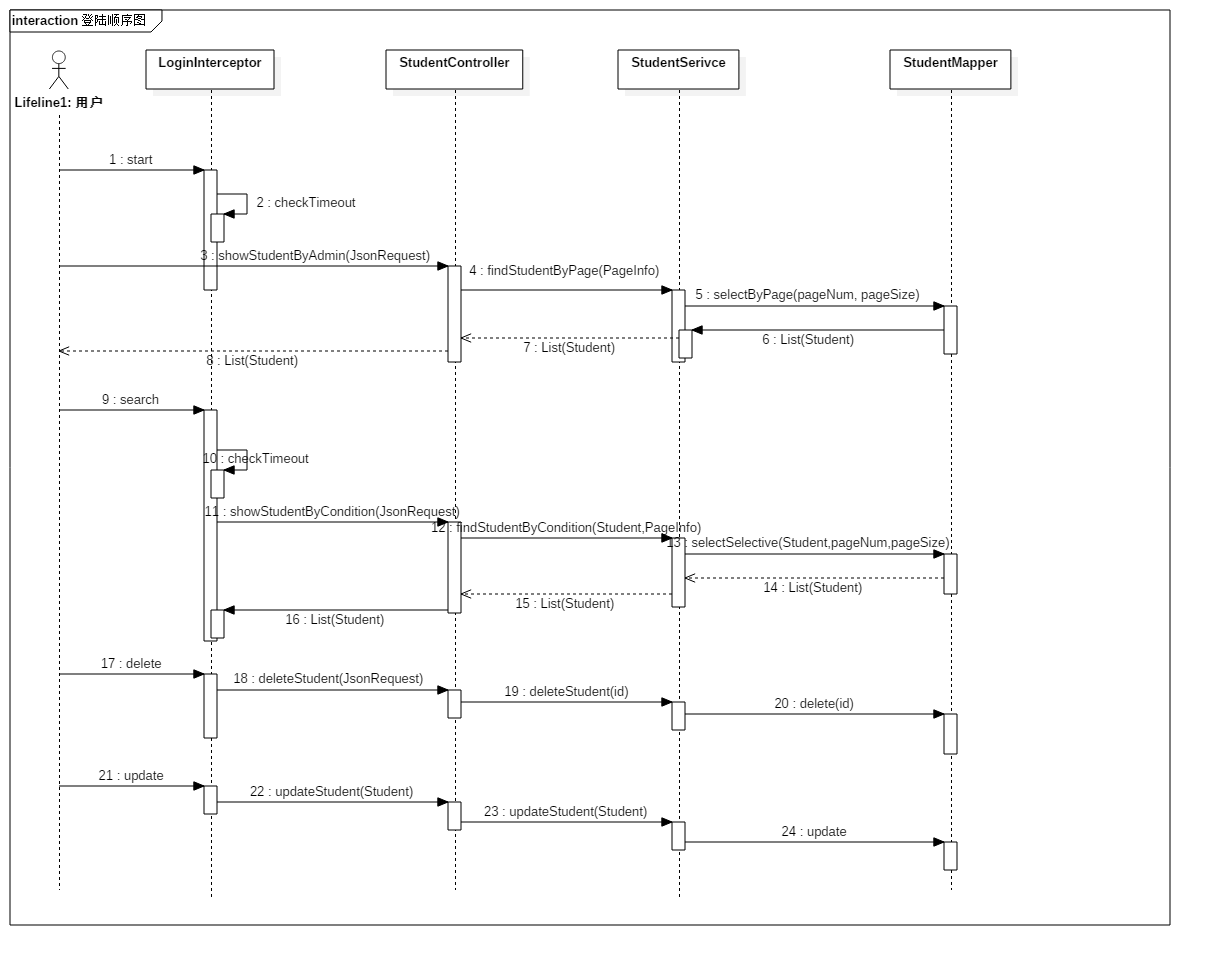


图 23 学生管理顺序图

5.2.8 班级管理

课程管理：

当管理员选择班级管理选项卡时，选择课程管理分选项卡，前端提交请求参数到“/admin/add\_lesson.action”,LessonController中将请求参数转换为ClassDto对象接收参数，将课程信息转换为List<Lesson> 对象存储，再调用LessonService中的addLesson方法为班级存储每一节课时。

当管理员选择课程表的分选项卡时，前端提交请求参数（该天的所在星期的星期日和星期六的日期字符串）到“/admin/lesson\_list.action”,表示请求该班级当前星期的所有课程信息，LessonController接收请求参数，调用LessonService的findLessonByDateRange的方法，逻辑验证完毕后，调用LessonMapper中的findLessonByDateRange方法，并将课时信息数据返回到前端，前端根据课时的详细信息，利用课时信息，绘制出该班当天的课程表。

成员管理：

当管理员选择班级管理选项卡时，选择成员分选项卡，前端提交请求参数到“/admin/classes\_student.action”和“/admin/classes\_teacher.action”,分别获取该班的教师和学生列表，将参数信息转换为UserClassesRel对象，调用userClassesRelService的selectSelective方法，向客户端返回学生，教师列表数据。

当管理员选择增加学生时候，前端发起请求“/admin/student\_list.action”,获取student表的所有学生数据，StudentController接收请求，利用 PageInfo对象接收请求参数，StudentService调用findByPage的方法，返回学生分页列表数据。但勾选学生并提交时，前端发起请求“/admin/classes\_add\_student.action”,ClassesController利用ClassesDto接收参数学生列表，并转化为List<Student>，调用userClassesRelService的add方法，建立学生和班级关系。

当管理员选择删除学生时候，前端发起“/admin/classes\_delete\_student.action”请求，ClassesController接收请求，将请求参数转换为userClassesRel 对象，UserClassesRel调用delete方法，删除选择学生与班级建立的关联关系，并返回响应结果，浏览器根据响应执行相应提示。

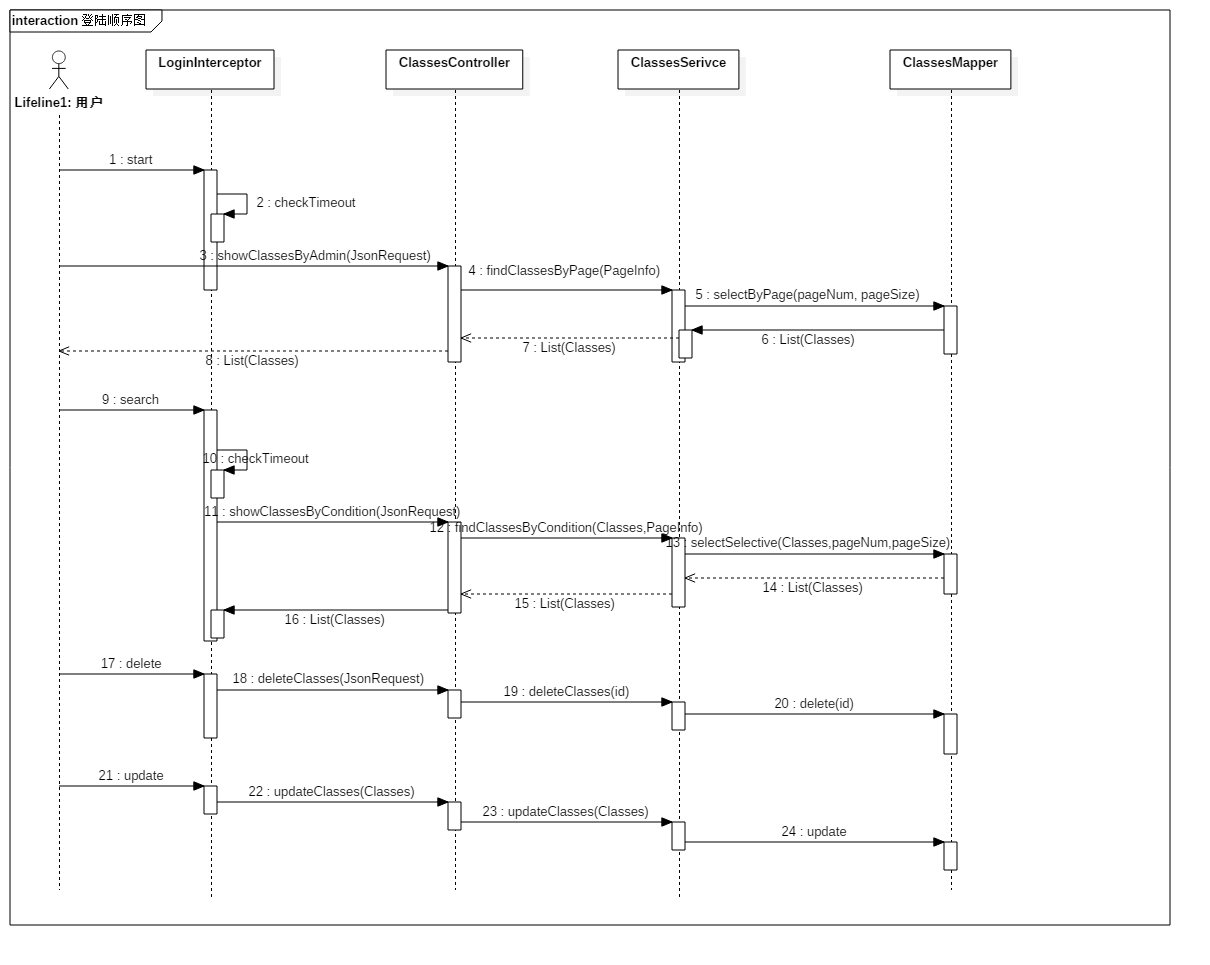


图 24 班级管理顺序图

6 系统实现

6.1 系统前端模块

6.1.1 新闻浏览

如图所示，当访客通过首页新闻列表，阅读到分类的新闻模块，和其属于的新闻标题。当点击“详情”，页面即跳转到详情页。

图 25 系统首页



图 26 新闻详情

新闻浏览模块关键代码如下：

表 3 新闻浏览模块查询

|  |  |
| --- | --- |
| Input:  JSON格式封装的查询新闻参数.  Output:  查询到的新闻对象，以JSON格式返回 | |
| 1: | @RequestMapping("/news") |
| 2: | public JsonResponse showNews(@RequestBody JsonRequest jsonRequest){ |
| 3: | News news = objectMapper.convertValue(jsonRequest.getData(), News.class); |
| 4: | News newArticle = newsService.findOne(news.getId()); |
| 5:  6  7  8 | news.setContext(news.getContext());  Map<String, Object> ret = new HashMap<String, Object>();  ret.put("news", newArticle);  return JsonResponse.newOk(ret); |
| 9: | }. |

6.1.2 资料下载

如图所示，访客可以用过进入资料下载页面，点击已经分类的资框，便触发自动下载。

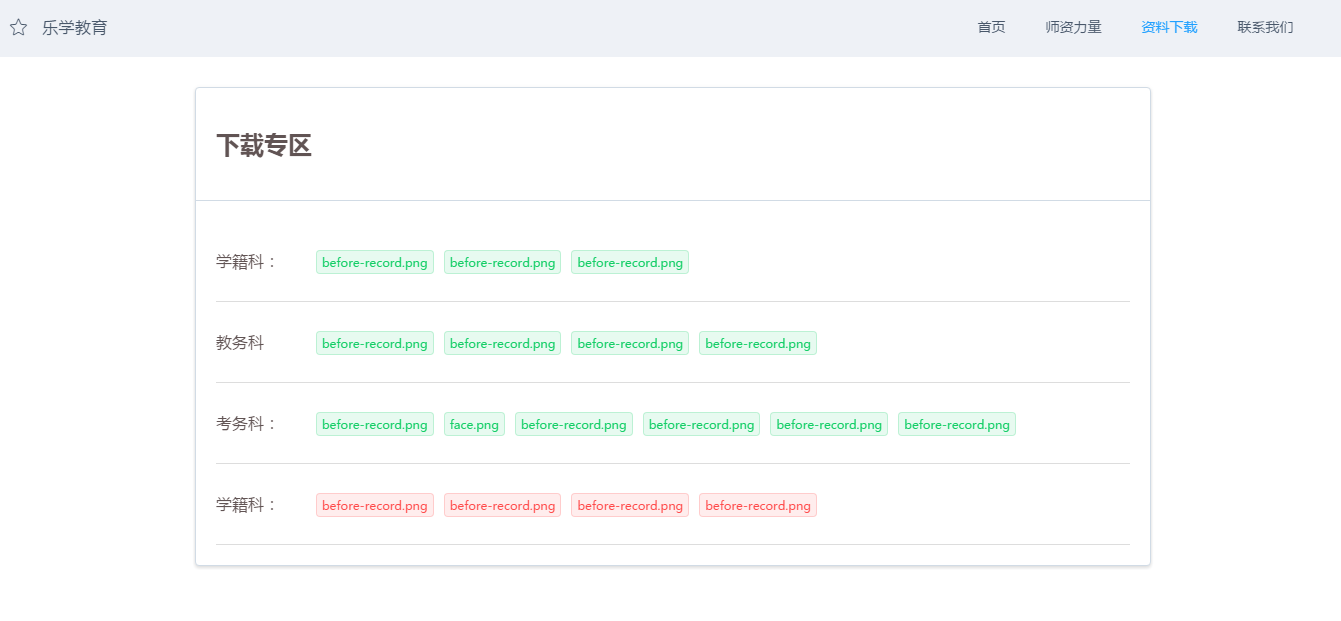


图 27 下载专区页

资料下载模块关键代码如下

表 3 资料下载模块查询

|  |  |
| --- | --- |
| Input:  JSON格式封装的查询文件参数.  Output:  查询到的文件对象列表，以JSON格式返回 | |
| 1: | @RequestMapping("/files\_list") |
| 2: | public JsonResponse showNews(@RequestBody JsonRequest jsonRequest){ |
| 3: | Files files = objectMapper.convertValue(jsonRequest.getData(), Files.class); |
| 4: | List<Files> filesList = fileService.findAllByType(files); |
| 5 | return JsonResponse.newOk(filesList); |
| 9: | }. |

6.1.3 留言板

访客在前端留言问答区域，输入想问的问题。管理员在后台查看到有未访问记录，回答留言，即能在留言板上查看。



图 28 留言问答页

留言问答主要代码如下：

表 3 留言问答模块查询

|  |  |
| --- | --- |
| Input:  JSON格式封装的查询文件参数.  Output:  查询到的文件对象列表，以JSON格式返回 | |
| 1: | @RequestMapping("/message\_add.action") |
| 2: | public JsonResponse addArticle(@RequestBody JsonRequest jsonRequest){ |
| 3: | Messagemessage = objectMapper.convertValue(jsonRequest.getData(), Message.class); |
| 4: | Date date = new Date(); |
| 5  6  7  8  9  10  11  12  13 | message.setCreateTime(DateUtil.formatToString("yyyy-MM-dd", date));  Map<String, String> ret = new HashMap<String, String>();  int i = messageService.saveMessage(message);  if( i>0 ){  ret.put("state", "2000");  }else{  ret.put("state", "5000");  }  return JsonResponse.newOk(ret); |
| 14 | }. |

6.1.4 微信端个人主页

学生或老师在微信登陆后，输入个人账号密码，查看个人信息，和课程表

a

图 29 微信端个人信息页

6.2 系统后台模块

6.2.1 教师管理

对老师进行增加，删除，查询，修改，指派班级

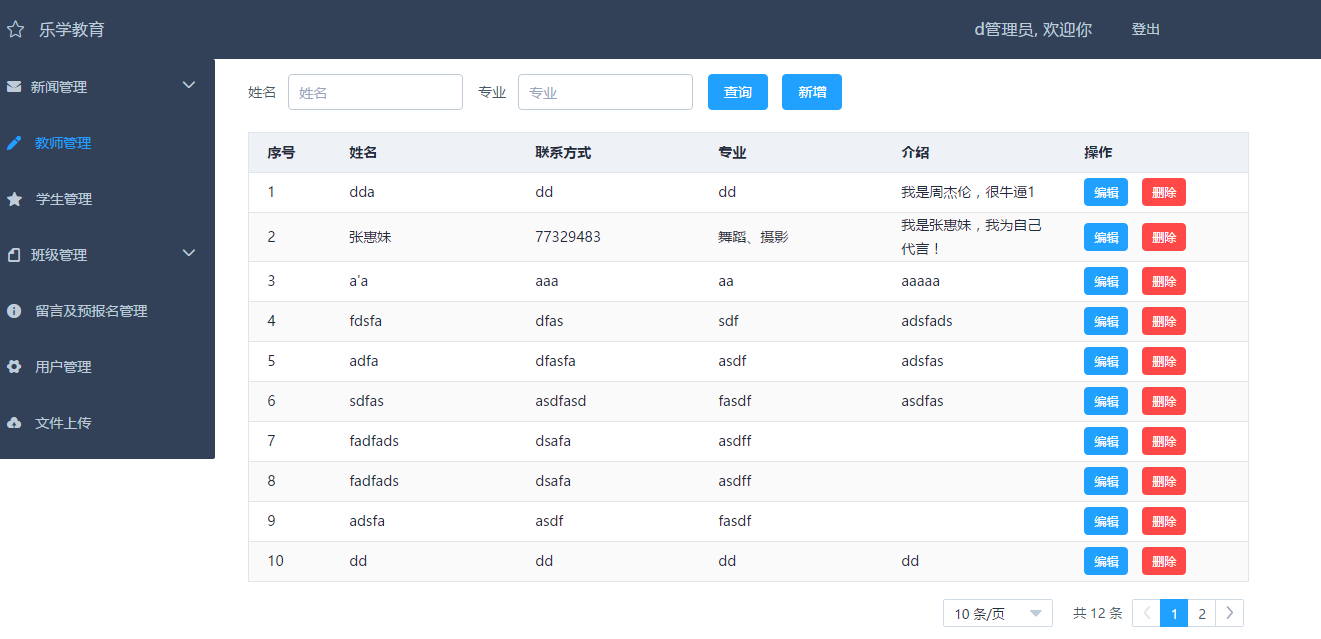


图 30 教师管理模块列表页



图 31 教师管理模块编辑页

6.2.2 班级管理

对班级进行增加，删除，查询，改变，指派学生，教师，安排课程，发布新闻



图 32班级管理列表页

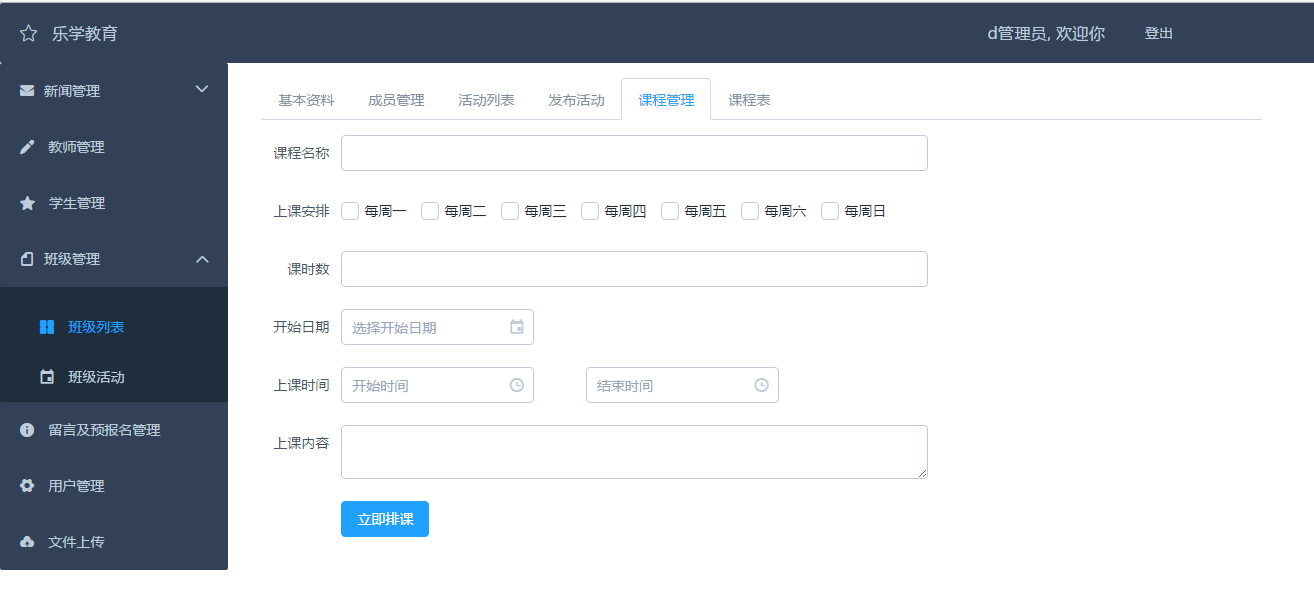


图 33班级管理排课页



图 34班级管理课程表页

6.2.3 新闻管理

对新闻进行增加，删除，查询，改变

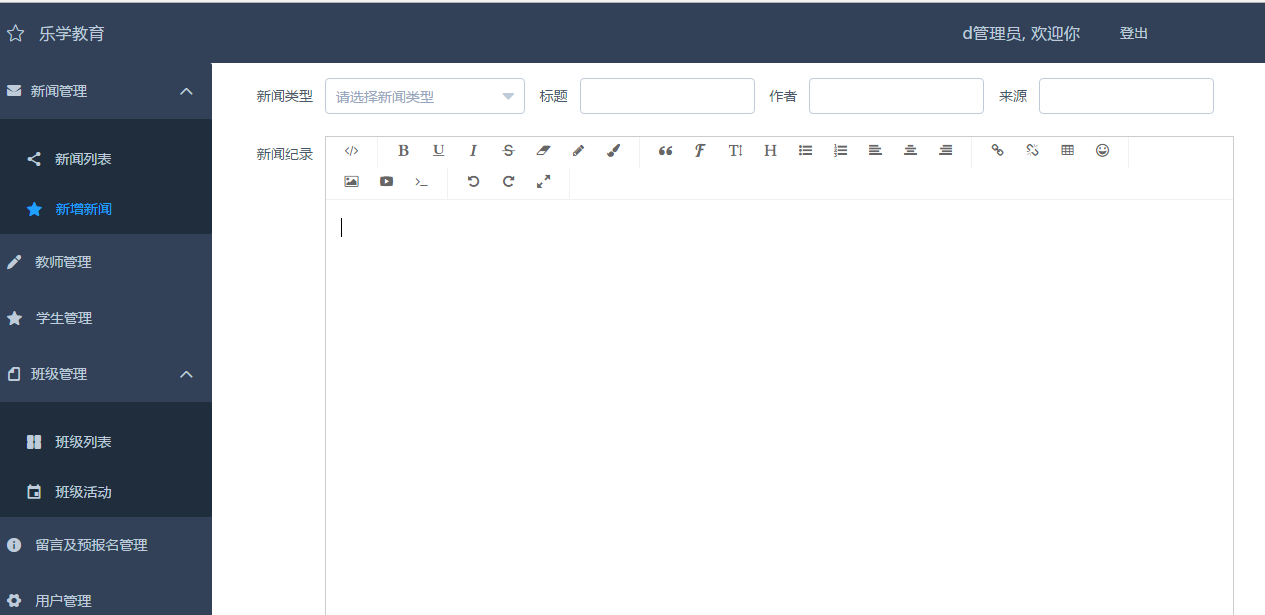


图 35新闻管理编辑页

新闻管理编辑主要代码如下：

表 3 新闻管理模块查询

|  |  |
| --- | --- |
| Input:  JSON格式封装的编辑新闻数据  Output:  保存新闻成功，以JSON格式返回数据成功提示 | |
| 1: | @RequestMapping("/admin/news\_add") |
| 2: | public JsonResponse addArticle(@RequestBody JsonRequest jsonRequest){ |
| 3:  4  5  6 | News \_new = objectMapper.convertValue(jsonRequest.getData(), News.class);  Date date = new Date();  \_new.setDate(DateUtil.formatToString("yyyy-MM-dd", date));  Map<String, String> ret = new HashMap<String, String>();, Message.class); |
| 4:  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | if(\_new.getId() == null) {  int i = newsService.saveNews(\_new);  System.out.println(\_new.getId());    if(i>0){  ret.put("message", "添加文章成功");  ret.put("newsId", \_new.getId().toString());  }else{  ret.put("message", "添加文章失败");  }  }else{  int i = newsService.updateNews(\_new);  if(i>0){  ret.put("message", "更新文章成功");  ret.put("newsId", \_new.getId().toString());  }else{  ret.put("message", "更新文章失败");  }  }  return JsonResponse.newOk(ret);  } |
|  |  |

6.2.4 留言管理

对留言进行增加，删除，查询，改变

7 结论与讨论

本文主主要在对我国儿童兴趣教育机构进行深度调研需求的情况下，以助力于儿童兴趣教育机构借助“互联网+”的思维，实习产业升级。本系统主要采用 web技术 + 微信平台，保证了机构的入门低门槛，系统接入方便，无障碍。同时采用前后端分离技术，前后端仅仅在接口耦合，保证了开发效率和稳定性，同时也带来了良好的用户体验。

系统的实现过程中运用了面向对象的思想，使用UML实现对业务进行建模，并用工程化的方法进行软件设计和开发。

在本系统中，运用了开源前段ui组件element-ui和框架Vue，不仅带来美化了界面，更加保证过了系统的优越的性能。后端主要采用主流的SSM框架，采用分层设计的思想，保证系统的稳定性和可维护性。

系统基本实现教务管理和教务宣传服务等功能，但还是有如下不足：

（1）系统前端表单校验不够完善，用户可以在比较重要的表单字段的空值输入

容易导致数据不完整或出错。

（2）后台的关联关系比较多，但这方面的处理还是不够完善，比较容易产生脏数据。

参 考 文 献

陈鑫. Spring框架的分析研究[J]. 电脑知识与技术:学术交流，2007，3(13)：108-110.

微信公众平台开发者文档[EB/OL]. https://mp.weixin.qq.com/wiki/home/index.html

微信公众平台[EB/OL]. <https://mp.weixin.qq.com>.

郑永芹.基于微信公众号平台的教务系统开发[D]. 华南理工大学，2015.

孙志明.以“互联网+”思维促进教育教学的变革与创新 [J]. 北京教育高教，2016.

李贤.“互联网+”时代儿童兴趣教育机构如何抓住机遇[J]. 教育教学论坛，2015(11):68-69

付亮. 2012-2013年移动互联网发展趋势综述[D]. 北京：光芒网络集团，2013.

Craig Walls. Spring in Action[M]. Greenwich：Manning Publication Co，2011.

致 谢

本系统从系统设计、编码实现到论文的完成，除了自己所投入的精力，也离不开刘昌余老师的悉心指导。刘昌余老师为人亲和，有渊博的学识和严谨的科研精神。在完成毕业设计的这段时间里，他给予了许多合理的建议和学习指导，在学习和处事上给予很大的启发。

除此之外，我还要感谢我的同学们。在大学本科学习阶段，大家互帮互助，时常相互分享学习心得和学习经验，共同营造维护良好的学习氛围。

最后，我要感谢我的家人，是他们的默默付出，支持和鼓励我去做自己所喜欢的事情，让我可以轻装上阵，放飞理想。

概要设计少

标题后的空格

概要设计