

**Southwest University of Science and Technology**

本科毕业设计（论文）



题目：基于微信公众号的教学评分系统设计与实现

学生姓名: 李俊冬

学生学号: 20130525

专 业: 计算机科学与技术专业

指导教师: 杨雷

学院(部): 计算机科学与技术专业

教务处制表

**微信服务号教学评分系统的设计与实现**

摘要

微信服务号教学评分系统是为了解决传统教学中纸质评分的低效问题而开发的一套系统，评分系统依托于微信服务号为用户提供线上评分以及用户之间沟通的功能，该系统服务器主要包括用户线上创建课程，助教管理，学生成绩管理，课程关注、教师导出学生成绩等功能，为了方便用户找到系统入口，系统使用了微信服务号的 API，整个系统依托在微信服务号之上，用户可以随时访问，用完即走。系统采用的 MongoDB3.4作为数据库管理系统，采用 Node.js 7.10 作为应用服务器，采用了Koa.js 2.0 作为服务器框架，采用 Ajax 技术 实现异步传输，JSON 作为信息传递格式。论文主要围绕着微信服务号教学评分系统的设计与实现的过程进行阐述，内容包含了需求分析，概要设计，详细设计，并且对整套系统进行移动端的详细测试，测试结果表明实现内 容满足设计需求。

**关键词：**Node.js，服务号开发，教学评分

**DESIGN AND IMPLEMEMENTATION OF WECHAT MARK SYSTEM**

**ABSTRACT**

The system is designed to solve the inefficiency of paper grading in traditional teaching. The scoring system relies on the WeChat service number to provide users with online scoring and communication between users. The system server including the user to create online courses, teaching management, student scores management, curriculum attention, teachers, student performance and other functions, in order to facilitate the user to find the system entrance, the system uses a WeChat service number API, the entire system depends on the WeChat service number , Users can visit at any time, run out that go.The system uses MongoDB 3.4 as a database management system, using Node.js 7.10 as an application server, using Koa.js 2.0 as a server framework, using Ajax technology to achieve asynchronous transmission, use JSON as the information delivery format.The paper mainly elaborates on the design and implementation process of the teaching system of the WeChat service number, which includes the requirements analysis, the outline design, the detailed design, and the detailed testing of the whole system, and the test results show that the content meets the design requirements.

**Key words**：Node.js, Wechat develop, Education score

**目录**

[第一章：绪论 5](#_Toc483290737)

[1.1概述 5](#_Toc483290738)

[1.2选题背景及目标 5](#_Toc483290739)

[1.3相关技术 6](#_Toc483290740)

[1.3.1 Node.js 6](#_Toc483290741)

[1.3.2 Koa.js 7](#_Toc483290742)

[1.3.3 MongoDB 7](#_Toc483290743)

[1.3.4 EJS 模板引擎 7](#_Toc483290744)

[1.3.5微信公众平台介绍 7](#_Toc483290745)

[1.4系统开发环境介绍 9](#_Toc483290746)

[第二章：功能与需求 10](#_Toc483290747)

[2.1.需求概述 10](#_Toc483290748)

[2.2.功能性需求 10](#_Toc483290749)

[2.3系统模块划分 11](#_Toc483290750)

[2.4系统非功能性需求 11](#_Toc483290751)

[第三章：概要设计 12](#_Toc483290752)

[3.1.系统架构设计 12](#_Toc483290753)

[3.2页面设计 13](#_Toc483290754)

[3.3逻辑设计 13](#_Toc483290755)

[3.4持久化设计 13](#_Toc483290756)

[3.5数据库设计 13](#_Toc483290757)

[第四章：详细设计与实现 15](#_Toc483290758)

[4.1用户管理模块： 16](#_Toc483290759)

[4.1.1注册功能： 16](#_Toc483290760)

[4.1.2 用户注册功能实现 19](#_Toc483290761)

[4.2 课程管理模块 21](#_Toc483290762)

[4.2.1 用户创建课程模块 21](#_Toc483290763)

[4.2.3 教师管理助教 24](#_Toc483290764)

[4.2.3教师发布课程信息： 27](#_Toc483290765)

[4.2.4 助教评分 29](#_Toc483290766)

[4.2.5用户管理关注课程 32](#_Toc483290767)

[4.3 文件模块 33](#_Toc483290768)

[4.4搜索模块 34](#_Toc483290769)

[4.4.1搜索用户功能 34](#_Toc483290770)

[4.4.2搜索课程功能 34](#_Toc483290771)

[4.5用户交互友好 35](#_Toc483290772)

[第五章 系统运行与测试 35](#_Toc483290773)

[5.1系统测试环境 35](#_Toc483290774)

[5.1.1系统测试硬件环境 35](#_Toc483290775)

[5.1.2系统测试软件环境 36](#_Toc483290776)

[5.1.3 Node.js 服务器的配置和启动 36](#_Toc483290777)

[5.2 服务器系统测试 36](#_Toc483290778)

[5.2.1 测试需求概述 36](#_Toc483290779)

[5.2.2测试用例以及结果 37](#_Toc483290780)

[5.2.3测试总结 39](#_Toc483290781)

[结论 39](#_Toc483290782)

[致谢 40](#_Toc483290783)

# 第一章：绪论

## 1.1概述

Mark 评分系统是基于微信服务号的教师授课评分的辅助工具，教师可以通过系统上传学生名单，并在系统上管理助教，助教可以通过系统对上课学生进行评分，学生可以通过系统关注课程动态，及时获取课程消息。期末，教师可以导出学生成绩 Excel。通过这些功能给传统授课过程提高效率。

Mark 评分的开发是为了提高传统教学效率，提高助教的工作效率，减轻教师的工作负担，使得教师拥有更多的精力专注于教学。学生可以通过关注课程来跟进课程进度，及时获取教师发布的消息以及明确课后作业。以此来提高教师和学生之间的沟通效率。

## 1.2选题背景及目标

学习的意义在于应用，我们一直试着使用技术、使用网络提高我们的效率，随着技术个快速革新以及新平台的不断诞生，提高效率的门槛越来越低，这也是大家喜闻乐见的。

在 Web 智能实验室学习期间，我还担任着 Web 交互设计实验课的助教，主要工作是根据学生在课上对作业的完成程度进行打分，每节课的学生人数都维持在 150 人左右，人数很多，与我一起参加助教工作的还有另外六位同学，每堂课的评分时间，每位助教拿着打印的学生名单，在名单上挨个找名字进行评分，与检查和打分相比，过程中时间开销最大的过程就是在纸质表格中找到对应学生的位置，拉低了整个过程的效率。期末汇总成绩的时候需要将七八份纸质成绩单手动汇总到电脑中，过程中还可能出现名单丢失以及录入成绩出错等情况。目前校内的课程评分都是以这种传统的方式进行，如果将评分方式电子化，这将很大程度上节约任职教师与助教的时间，提高教学效率。这也是我选择做评分系统的原因。

李克强总理在政府工作报告中提出的 互联网+ 理念，是创新2.0下中国互联网发展新形态、新业态，是知识社会创新2.0推动下的互联网形态演进，而互联网+教育则应该是互联网下传统教育方式演进的未来。线上评分系统可以将所有的数据存储在云端，保证了数据的安全性与稳定性，进一步推进无纸化教育。

基于微信的评分系统的应用场景是线下，一个线下的低频的工具需要一个用户容易获取的入口，让用户精准、快速的获取到，而微信正是这样的平台。截止 2017年第一季度，微信的月活跃人数达到9.38 亿，并且 94% 的用户每天使用WeChat，61%的微信用户每天打开微信超过10次，36%的用户每天打开微信超过30次，32% 的用户每天使用微信超过2小时，微信的用户粘性更强。而大学生群体在这些海量用户群中占比极高，可以说当下的大学生人人都有微信，人人都用微信。

截止目前，微信应用于院校信息传播已经有了很多案例，学校利用微信公众平台推送关于本校校园文化、院系特色、专业研究方向等信息，而且推送最新资讯，如新生报到流程、师 院要闻等。不仅增加学生学习交流的机会。也扩大了学校在学生、家长、社区乃至全国的关注度，更可以节约成本，实现学校品牌的传播 不仅如此，微信还大范围地应用到了图书馆管理系统，学生不但可以通过微信订阅号查看图书馆发布的相关资讯，比如相关活动的安排、知识文化讲座或者书籍推荐等，也可以通过已有导航条选择所需要的服务，如论文检索、馆际互动、文献传 递、资讯互动、图书推荐等，同时还可以直接在导航条中选 择查看相关的公告，直接在手机端进行图书浏览。通过微信平台，图书馆工作人员将最新的图书馆资讯送到学生身边，而学生通过平台与管理员互动咨询，获得所需的最新 图书、 期刊信息，从而使双方传递信息时变得更加便捷和有效。这些系统已经大范围地应用在了我们的生活中，并且给我们的生活带来了极大的方便。这也是我开发这套系统的初衷。在评分系统的设计过程中，将微信服务号作为一个快速入口，不需要独立的 APP 支撑，评分系统的应用场景，用户的使用频率低，但是使用时又需要及时获取，所以使用服务号的方式既可以保证用户可以及时获取，又能保证用户可以用完即走。

## 1.3相关技术

### 1.3.1 Node.js

Node.js是一个开放源代码、跨平台的JavaScript语言运行环境。Node.js由Node.js基金会持有和维护。Node.js采用Google开发的V8运行代码，使用事件驱动、非阻塞和异步输入输出模型等技术来提高性能，可优化应用程序的传输量和规模。这些技术通常用于数据密集的事实应用程序。Node.js大部分基本模块都用JavaScript语言编写。在Node.js出现之前，JavaScript通常作为客户端程序设计语言使用，以JavaScript写出的程序常在用户的浏览器上运行。Node.js的出现使JavaScript也能用于服务器端编程。Node.js含有一系列内置模块，使得程序可以脱离Apache HTTP Server或IIS，作为独立服务器运行。

### 1.3.2 Koa.js

Koa是由Express之后的团队设计的下一代node.js框架，旨在为Web应用程序和API提供一个更小，更具表现力和更强大的基础。 通过利用生成器，减少异步回调，提高代码的可读性和可维护性，同时改进了错误处理。 Koa不会在内核中捆绑任何中间件，并提供了一套优雅的方法，使写入服务器变得快速。

### 1.3.3 MongoDB

MongoDB是可以应用于各种规模的企业、各个行业以及各类应用程序的开源数据库。作为一个适用于敏捷开发的数据库，MongoDB的数据模式可以随着应用程序的发展而灵活地更新。与此同时，它也为开发人员 提供了传统数据库的功能：二级索引，完整的查询系统以及严格一致性等等。 MongoDB能够使企业更加具有敏捷性和可扩展性，各种规模的企业都可以通过使用MongoDB来创建新的应用，提高与客户之间的工作效率，加快产品上市时间，以及降低企业成本。MongoDB是专为可扩展性，高性能和高可用性而设计的数据库。它可以从单服务器部署扩展到大型、复杂的多数据中心架构。利用内存计算的优势，MongoDB能够提供高性能的数据读写操作。 MongoDB的本地复制和自动故障转移功能使您的应用程序具有企业级的可靠性和操作灵活性。

作为 NoSql 领域的领头羊，MongoDB 相较于传统数据库有以下优势：读写效率高-由于文档模型把相关数据集中在一块，在普通机械盘上读数据的时候不用花太多时间去定位磁头，因此在IO性能上有先天独厚的优势；可扩展能力强-关系型数据库很难做分布式的原因就是多节点海量数据关联有巨大的性能问题。如果不考虑关联，数据分区分库，水平扩展就比较简单；动态模式-文档模型支持可变的数据模式，不要求每个文档都具有完全相同的结构。对很多异构数据场景支持非常好；模型自然-文档模型最接近于我们熟悉的对象模型。从内存到存储，无需经过ORM的双向转换，性能上和理解上都很自然易懂。

### 1.3.4 EJS 模板引擎

EJS是一个简单高效的前端模板语言，通过数据和模板，可以生成HTML标记文本，并且可以同时运行在客户端和服务器端，具有快速编译和渲染，简单的模板标签，自定义标记分隔符，支持文本包含，支持浏览器端和服务器端，模板静态缓存，支持express视图系统等优势。

### 1.3.5微信公众平台介绍

微信公众平台是腾讯微信软件中的一个重要的应用。公众号可以有企业、政府机构或组织及个人进行开发和运营。向订阅的用户提供群发或自动回复商品信息的服务，主要包括图片、文字、声音、视频几类。用户自主选择感兴趣的公众号进行订阅。

#### 1.3.5.1微信公众平台账号的分类

腾讯微信公众平台有三种不同类型的账号：订阅号、服务号和企业号。每一种类型都有两种模式：开发者模式和设计者模式。其中开发者模式针对的是具备一定编程能力的人群。设计者模式则不需要用到编程知识，当然使用的功能和效果也会相对来说比较单一。

三种类型的公众平台具体的区别如下：

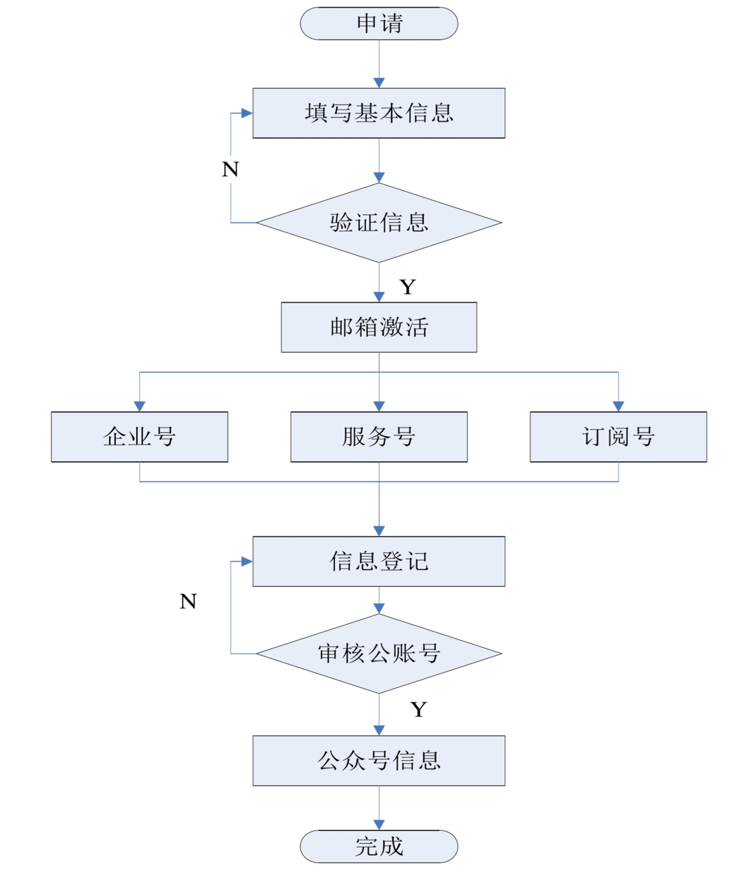
1、服务号：为企业和组织提供更强大的业务服务，如以往在线客服的工作，先如今很用户借助服务号可以自主完成。公众号服务平台帮助企业更加高效和节约的进行客户管理工作。

2、订阅号：订阅号每天可以主动推送一条消息，更加适合官方媒体和自媒体使用。订阅号是媒体和用户互动途径，也是一种消息传播的新方式。

3、企业号：主要应用于企业内部，管理企业内部各部门和的员工之间的关系。帮助企业快速构建和内部员工的信息通道。

#### 1.3.5.2 微信公众账号申请流程

本系统根据功能需求，选择了服务号进行本系统的开发。填写了一系列资料：提交申请，验证，激活，信息登记，再通过一系列的审核，如果对公账户、营业执照等等、法人身份，实名认证支付为期一年的认证费用等等。最终完成了公众号的申请。具体申请流程如图1-1 所示：



**图1-1 账号申请流程**

#### 1.3.5.3 微信公众平台 API 介绍

1. 获取 Access\_token

Access\_token的获取在微信公众平台接口开发中，Access\_token 占据了一个很重要的地位，它相当于进入各种接口的钥匙，拿到该个钥匙才拥有调用其他各种特殊接口的权限。Access\_token是公众号的全局唯一票据，公众号调用各接口时都需使用 Access\_token。Access\_token有效期是２小时，重复获取将导致上次获取的Access\_token失败。公众号使用 APPID和 APPSecret 调用如下表1-1的接口来获取Access\_token。而APPID和APPSecret 可在开放模式下获得。

**表1-1 接口调用请求说明**

|  |  |
| --- | --- |
| 请求方式 | GET |
| 请求地址 | https://api.weixin.qq.com/cgi-bin/token?grant\_type=client\_credential&appid=APPID&secret=APPSECRET |
| 参数说明 | APPID 和 APPSecret |

1. 自定义菜单接口

微信公众号推出自定义菜单功能，使其内容更丰富、界面更美观。目前自定义菜单里的一级菜单只能支持１－３个，每个二级菜单只能支持１－５个。一级菜单最多包含４个汉字，而二级菜单最多包含７个汉字，溢出部分Ｗ省略号代替。自定义菜单创建后，微信客户端只需重新关注服务号便可看到生成的新菜单。本服务号的开发主要用到的是 click 和 view 该两种类型的按钮如下：

（a）click：点击推事件

用户点击址 click 类型按钮后，便触发了微信服务器的推送事件，它将消息类型为 event的结构和对应的 key值都推送给开发者，开发者就可Ｗ通过自定义的 key 值与用户进行交互；

（b）view：跳转 URL

用户点击 view类型按钮后，便触发了微信服务器的跳转事件，客户端将直接打开按钮对应的网页 URL,结合对应的接口获得用户的基本信息。

服务号的自定义菜单结构如下：

＂button＂：［｛

＂type＂：＂view＂，

＂name＂：＂Mark 评分＂，

＂url＂：＂http://mark.lijundong.com＂

｝］

各参数含义如表1-2所示

**表1-2 自定义菜单参数**

|  |  |  |
| --- | --- | --- |
| 参数 | 是否必须 | 描述 |
| Button | 是 | 一级菜单组 |
| Type | 是 | 菜单的响应动作类型 |
| Name | 是 | 菜单命并且字数不超过256 个 |

## 1.4系统开发环境介绍

开发平台：MacOS 10.12.4，CentOS6，Chrome 59

开发工具：SublimeText3， iTerm2

软件平台：Node.js 7.9，Koa.js 2，EJS1.0，MongoDB 3.4

# 第二章：功能与需求

## 2.1.需求概述

Mark 评分系统提供一个平台，教师用户可以在平台上创建新课程，并在课程˙中更新学生信息，发布课程信息，管理课程助教，期末导出学生成绩表单，助教可以通过平台给课上学生进行评分，学生可以通过系统关注课程，获取到课程的信息并及时获取到教师发布的消息

## 2.2.功能性需求

登录注册功能：Mark 评分提供用户注册、登录、登出功能，对于平台而言，所有的用户都是普通角色，教师、助教、学生的角色是在特定的课程中相对而言的。用户仅需要提供邮箱以及设置密码就可以注册，注册成功，

账户信息管理功能：用户登录之后可以在个人中心设置个人信息包括用户头像、账户昵称、用户性别等资料。

微信账户授权登录：评分系统基于微信服务号，为了获取良好的用户体验，需要提供系统支持微信授权登录，用户通过关注公众号，进入评分系统，授权微信登录系统，即可进入系统。

用户创建课程：在平台上注册的用户，都可以创建课程，创建新的课程，创建成功后可以维护课程信息，设置课程的名称、课程教师、课程信息等信息。

导入学生名单 Excel：传统打分方式是助教拿着纸质学生名单，手动统计的方式，Mark 打分平台提供上传学生名单Excel 到服务器的功能，纸质表格电子化，简化助教和教师的工作流程，提高了教学效率。

导出学生成绩 Excel：传统教学，期末汇总成绩时需要将多张纸质成绩表手动汇总，效率低，Mark 打分可以自动汇总学生成绩，导出 Excel，很大程度地简化了过程，提高效率。

教师管理助教：用户创建课程成功后，可以在课程详情页通过搜索用户进行添加助教操作，并可以在课程详情页面移除已添加用户的主角权限。

教师发布课程信息：课程详情页提供入口，供教师用户发布课程消息，传统的方式一种是通过学习委员与学生进行信息传达，但是实验课这种类型，生源散乱，不适合这种方式，另一种是建立 IM 沟通群，但是这又涉及到群管理问题。通过微信评分系统发布课程信息，学生在需要的时候获取，教师可以做到高效的一对多信息发布。

助教用户评分：拥有的课程助教权限的用户在课程评分页面输入学生学号，课程号，课程得分学生进行评分。

用户管理关注课程：为了方便课程学生获取最新课程，用户可以通过搜索课程名称获取到课程列表，选择需要关注的用户进行关注课程，用户可以在已关注课程的详情页面取消关注课程。

搜索用户功能：评分系统提供用户搜索功能，可以通过模糊搜索方式搜索到用户，用户只需要提供用户名称的部分即可搜索到用户，例如搜索“王”，可以搜索到“王朝”，“王月”，“吴王”等用户。

搜索课程功能：用户可以通过搜索课程名称获取到课程列表，课程搜索支持模糊搜索，用户只需要提供课程名的某部分即可搜索到课程，例如搜索“计算机”，可以搜索到“计算机网络”，“计算机操作系统”等课程。

## 2.3系统模块划分

路由模块：服务器所有请求经过路由分发，进入相应的模块进行处理，包括页面渲染，数据交互等操作。

用户管理模块：通过用户管理模块，用户可以在Mark 评分平台上进行注册、更新用户信息，登录、登出等操作。

课程管理： 通过课程管理模块，用户可以创建课程，更新课程信息，管理助教权限，以及发布课程消息。

评分管理：前端模块里，教师和助教可以通过评分管理模块对学生进行评分。后台中，评分模块对学生的成绩进行汇总。

搜索模块：通过搜索模块，用户可以获取到特定的课程入口，教师可以通过搜索获取到特定用户的入口。

文件管理模块：通过文件模块，实现学生名单 Excel 的上传，以及导出学生成绩单到 Excel 表单。

## 2.4系统非功能性需求

系统可靠性：为了确保用户在使用中，系统的正常运行，用户数据的正常存储，确保系统的稳定性是非常必要的。Mark 打分系统主要应用场景在线下，虽使用频率不高，但是也要求系统在关键时刻不能出错，Mark 打分使用了强健的 Node.js 服务，可以确保平台在关键时刻正常运行。数据库使用了非关系型数据库 MongoDB，MongoDB是专为可扩展性，高性能和高可用性而设计的数据库。它可以从单服务器部署扩展到大型、复杂的多数据中心架构。利用内存计算的优势，MongoDB能够提供高性能的数据读写操作。 MongoDB的本地复制和自动故障转移功能使应用程序具有企业级的可靠性和操作灵活性。

系统可扩展性：Mark 打分平台目前还是一个很简单的产品原型，其功能还有很多想象空间，所以在技术实现上必须保证系统的可扩展性，系统使用 Koa.js+MongoDB 架构，Koad.js 框架结构清晰，组织明确，通过 Router 模块控制所有功能入口，各个模块之间相互独立，高内聚，低耦合。数据库采用非关系型数据库 MongoDB，在拓展性上远高于关系型数据库，MongoDB 在数据库设计过程中提倡优先内嵌，各个集合之间耦合性低，极大的降低了数据库扩展的成本。也方便开发人员后续的开发。

系统可维护性：Mark 打分使用 Koa.js + MongoDB 架构实现，系统分层明确，清晰，如需要对系统进行维护，修改代码，修改系统代码，可以通过 Router 模块快速定位到对应的代码模块。各模块代码之间相互独立，更改方便，不会造成代码之间相互污染的情况出现。

系统安全性：评分系统涉及教学评分数据，系统必须保证用户数据的安全性，以及用户登录的权限管理，以保证课程教学的正常进行。Mark 打分采用 Koa.js + MongoDB 架构，MongoDB 的安全性已经被大范围应用证明，MongoDB本身是具有企业级安全性的，花旗银行、巴克莱、高盛等顶级金融机构大规模使用就是一个很好的例证。通过使用 Koa.js 对用户登录操作时的权限进行控制，保证了评分系统安全，并且对用户的密码全部进行 MD5 不可逆的加密，使用户的密码安全性得到了保障。

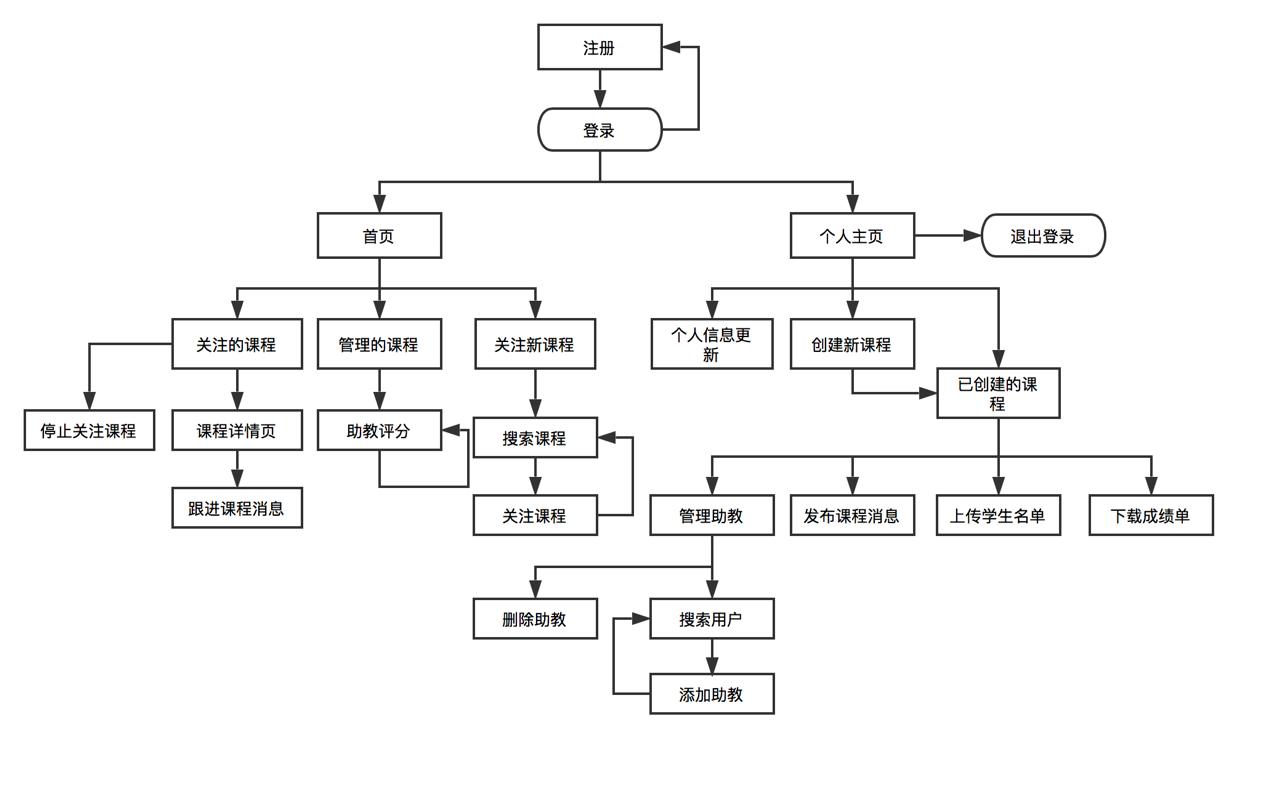
系统可交互性：Mark 打分系统设计与实现中，前端使用的是微信官方提供的 WEUI 框架，对微信生态原生友好，以及 jQuery-WEUI 框架，使用 Koa.js 服务端渲染，所有的数据交互全部采用 Ajax 请求，在保证了页面风格的情况下，交互数据请求响应迅速。Mark 评分系统整体风格沿用了微信的设计语言，用户使用时，快速熟悉易上手。移动端对用户交互有着更严苛的要求，因为系统基于微信服务号，所以需要保证系统与微信官方有统一的交互语言，保证用户的使用体验。系统在设计时做了大量的操作提示，用户操作时，可以及时准确的获取到操作的提示，例如完整填写表单、更正信息格式等信息。可以保证用户的使用体验。

# 第三章：概要设计

## 3.1.系统架构设计

Mark 评分采用了 Koa.js + MongoDB 架构，系统分层清晰，系统划分为控制层、服务层、DAO 层，路由层对服务器的所有请求进行分发，请求通过控制层分发到服务层，服务层通过调用 ADO层的方法处理请求，完成系统响应。

系统总体流程图如下：



**图3-1 系统流程示意图**

系统控制层通过 koa-router 接收服务器所有请求，并且调用服务层方法处理请求，并将处理后的数据返回。

系统服务层通过调用 DAO 层的方法，完成一个完整的服务。

## 3.2数据库设计

MongoDB 数据库设计先考虑内嵌， 直接按照对象模型来设计数据模型。如果对象模型数量不多，关系不是很复杂，可能直接一种对象对应一个集合就可以完成设计。内嵌是文档模型的特色，可以充分利用MongoDB的富文档功能来享受一些文档模型的性能和扩展性等特性。一般的一对一、一对多关系，比如说一个人多个地址多个电话等等都可以放在一个文档里用内嵌来完成。但是某些情况下，使用引用则难以避免。比如说， 一个明星的微博可能有几十万或者几百万的回复，这个时候如果把comments放到一个数组里，可能会超出16M的限制。这个时候可以考虑使用引用的方式，在主表里存储一个id值，指向另一个表中的 id 值。Mark 评分系统数据库设计共设计了三个集合，分别是用户集合、课程集合、评分集合，集合之间遵循优先内嵌的原则，保证了集合的高内聚低耦合。

用户集合包括了用户的姓名、电话、邮箱、头像、性别、创建的课程集合、管理的课程集合、关注的课程集合，创建的课程集合信息包括课程 ID、课程名称、课程教师名称、课程头像，管理的课程集合信息包括课程ID、课程名称、课程教师名称、课程头像，关注的课程集合信息包括课程ID、课程名称、课程任课教师，因为创建的课程集合与管理的课程集合直接嵌在了用户集合中，所以，Mark 评分系统没有设计角色字段，而是通过内嵌来实现权限的管理。

1. 用户集合

{

name: String,

phone: String,

email: String,

img: String,

gender: Number,

clesson: [{

lid: String,

lname: String,

lteacher: String,

limg: String

}],

mlesson: [{

lid: String,

lname: String,

lteacher: String,

limg: String

}],

following: [{

lid: String,

lname: String,

lteacher: String,

lmsg: Boolean //true: have message false: no message

}]

}

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 类型 |
| Name | 用户名 | String |
| phone | 用户电话信息 | String |
| email | 用户电子邮箱 | String |
| img | 用户头像 | String |
| gender | 用户性别 | Number 0/1 |
| clesson | 用户管理的课程 | Array |
| clesson.lid | 管理课程的 ID | String |
| Clesson.lname | 管理课程的名称 | String |
| Clesson.lteacher | 管理课程的教师名称 | String |
| Clesson.limg | 管理课程的头像 | String |
| followimg | 关注的课程 | String |
| Following.lid | 关注课程的 ID | String |
| Following.lname | 关注课程的名称 | String |
| Following.lteacher | 关注课程的教师名称 | String |
| Following.lmsg | 关注课程的信息 | Number |

课程集合中包含课程名称、课程信息、课程头像、课程时间集合、教师信息、教师用户ID、课程关注人数、课程发布信息集合、课程助教集合、以及课程 follow 用户集合，课程发布信息集合包括信息发布时间、信息详情，助教集合信息包括助教用户ID、助教用户名称、助教用户电话信息，课程的 follow 用户集合信息包括关注者用户 ID、关注者用户名称、关注者头像。

1. 课程集合

{

name: String,

info: String,

img: String,

time: [],

teacher: String,

teacherid: String,

follownum: Number,

msg: [{

mtime: Date,

mcontent: String,

receiver:[uid]

}],

assistant: [{

uid: String,

uname: String,

uphone: String

}],

follower: [{

uid: String,

uname: String,

uphoto: String

}]

}

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 类型 |
| Name | 课程名称 | String |
| Info | 课程信息 | String |
| Img | 课程头像 | String |
| Time | 课程时间 | Array |
| Teacher | 课程教师 | String |
| Teacherid | 课程教师 ID | String |
| Follownum | 课程关注人数 | Number |
| Msg | 课程消息 | Array |
| Msg.mtime | 课程消息发布时间 | Date |
| Msg.mcontent | 管理课程的头像 | String |
| Msg.receiver | 课程消息接受用户组 | Array |
| assistant | 助教数组 | Array |
| Assistant.uid | 关注课程的名称 | String |
| Assistant.uname | 关注课程的教师名称 | String |
| Assistant.uphone | 关注课程的信息 | String |
| follower | 关注用户数组 | Array |
| Follower.uid | 关注用户的ID | String |
| Follower.uname | 关注用户的名称 | String |
| Follower.uphoto | 关注用户的头像 | String |

分数集合最初设计嵌入课程集合中，因为分数涉及到频繁的查询修改，所以将其提出课程集合，设置为独立的集合，提高查询修改效率，分数集合包括所属的课程 ID、学生的班级信息、学生的姓名、学生的学号信息、学生的课程分数集合。

1. 分数集合

{

lid: String,

class: String,

name: String,

number: String,

mark: []

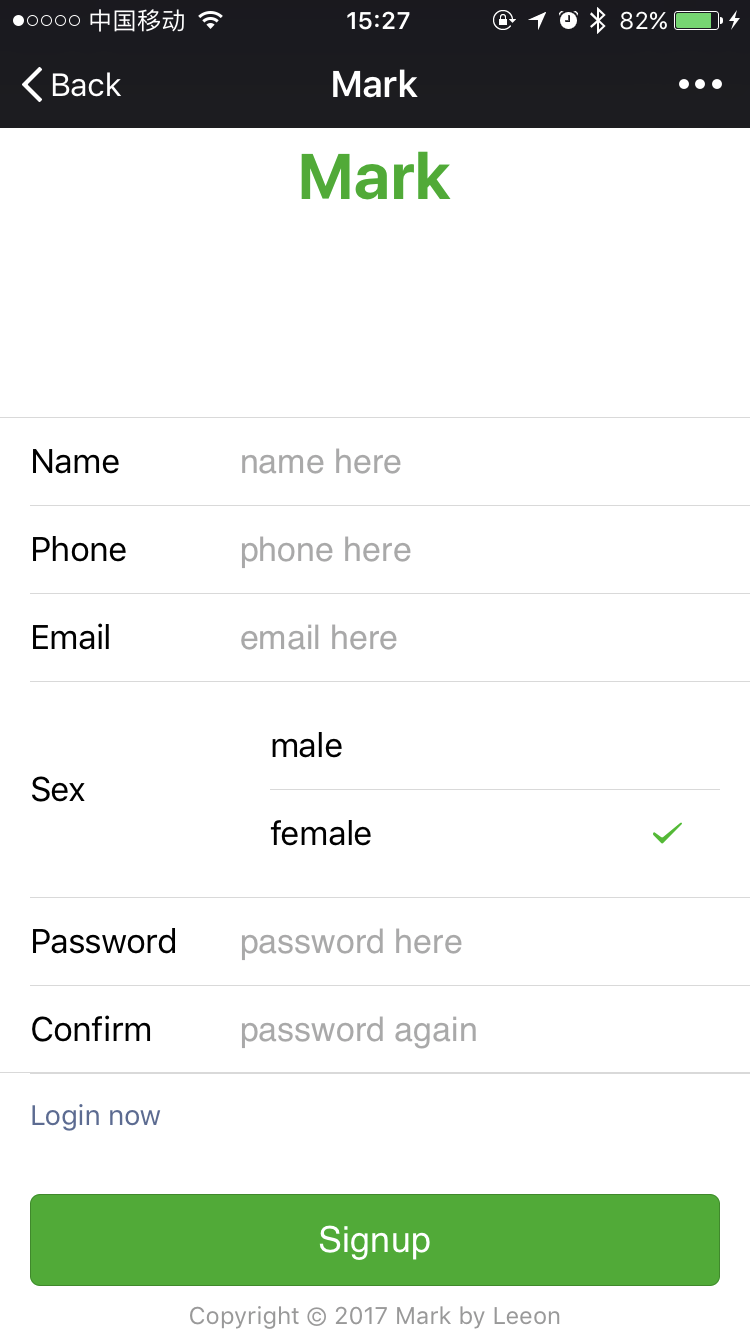
}

|  |  |  |
| --- | --- | --- |
| 字段 | 描述 | 类型 |
| Lid | 课程 ID | String |
| Class | 学生所在班级 | String |
| Name | 学生姓名 | String |
| Number | 学生学号 | String |
| Mark | 学生得分集合 | Array |

# 第四章：详细设计与实现

## 4.1用户管理模块：

### 4.1.1注册功能：



**图4-1 注册页面**

用户通过填写姓名、电话、邮箱、性别、账户密码等信息，完成注册，注册成功后，账户登录项是邮箱，账户密码即账户注册密码。信息未填写完整，点击注册后，页面会提示需要完成注册信息，注册请求不会提交，如果邮箱已经被注册了，注册信息提交之后，系统会识别邮箱已经注册，注册不会完成，并且页面提示该邮箱已经被注册了，请更换邮箱后重新提交注册申请，前端将用户填写的用户信息通过 Ajax 请求到后台，在提交之前，会对前端 Form 的内容进行校验，校验表单是否填写完整，以及填写项格式是否合法，例如电话以及邮箱的格式是否合法，如果用户填写了非法格式的邮箱或者电话，注册请求不会被提交，同时页面会提示需要填写正确格式的邮箱或者电话号码。

具体实现如下

$("#signupSub").click(function(e) {

if ($("#nameIn").val() && $("#phoneIn").val() && $("#emailIn").val() && $("#passIn").val() && $("#confirmIn").val()) {

if ($("#passIn").val() == $("#confirmIn").val()) {

var data = {

name: $("#nameIn").val(),

phone: $("#phoneIn").val(),

gender: ($("input#maleChoice").prop("checked") ? 1 : 0),

img: "/image/headpic-min.jpg",

email: $("#emailIn").val(),

password: $("#passIn").val()

};

$.ajax({

url: "signup",

type: "POST",

contentType: "application/json; charset=utf-8",

data: JSON.stringify(data)

}).done(function(res) {

if (res.status) {

$.toptip(res.msg, 'success');

window.location.href = "/login";

} else {

$.toptip(res.msg, 'error');

}

});

} else {

$.toptip("Password not match", 'error');

}

} else {

$.toptip("Complete info please", 'error');

}

});

请求经过 Koa 路由交给 user 模块处理，user 模块获取到前端提交的 json 数据后，通过 signup 方法创建 user 实例并写入 MongoDB。

具体实现如下：

业务层：

exports.signup = async function(ctx, next) {

const userData = ctx.request.body;

const user = await mUser.findOne({

email: userData.email

});

if (user) {

ctx.body = {

status: false,

msg: "Email already existed"

};

} else {

userData.password = bcrypt.hashSync(userData.password, saltRounds);

const newUser = await mUser.save(userData);

if (newUser) {

ctx.body = {

status: true,

msg: "Register success"

};

} else {

ctx.body = {

status: false,

msg: "Network error"

};

}

}

};

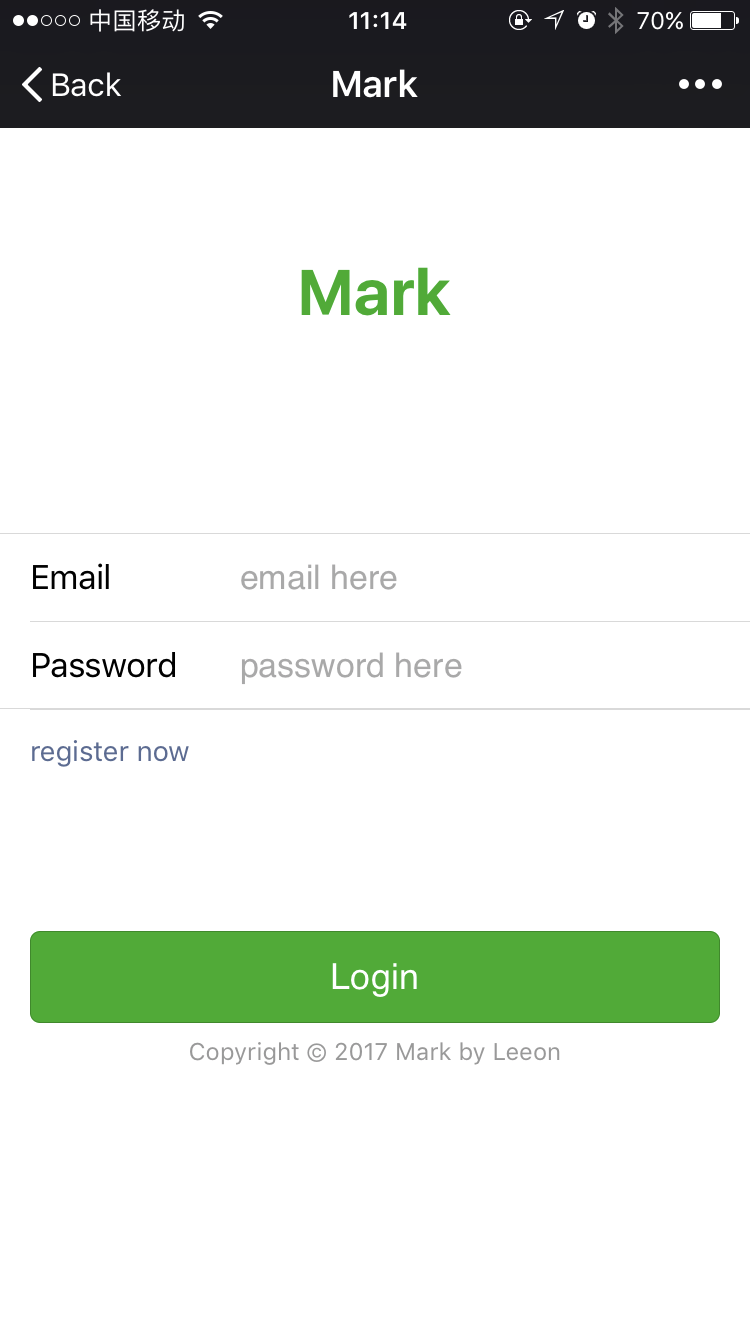
模型层：

userDAO.prototype.save = function(data) {

return new userModel(data).save();

};

### 4.1.2 用户注册功能实现



**图4-2 登录页面**

用户登录功能：

用户在登录界面，输入账户邮箱和账户密码，提交登录后，系统判断账户密码与账户邮箱吻合，页面提示登录成功，服务器生成用户 session，并跳转到主页面。在提交登陆请求之前，前端 javascript 会对提交信息进行校验，判断信息的完整以及格式的合法性，例如账户邮箱格式的合法性，如果信息没有填写完整就提交登陆申请，页面会提示需要完成填写所有信息之后才能进行提交。如果填写了非法格式的邮箱，登录请求将不会被提交，并且页面会提示输入正确格式的邮箱。输入的密码与注册的邮箱不匹配时，页面将提示密码错误，登录失败。

具体实现如下：

前端校验

$("#loginSub").click(function() {

if($("#emailIn").val() && $("#passIn").val()) {

var data = {

email: $("#emailIn").val(),

password: $("#passIn").val()

};

$.ajax({

url: "login",

type: "POST",

contentType: "application/json; charset=utf-8",

data: JSON.stringify(data)

}).done(function(res) {

if (res.status) {

$.toptip(res.msg, 'success');

window.location.href = "/";

} else {

$.toptip(res.msg, 'error');

}

})

} else {

$.toptip("Complete info please", 'error');

}

});

后端校验：

exports.login = async function(ctx, next) {

const \_user = ctx.request.body;

console.log(\_user);

const user = await mUser.findOne({

email: \_user.email

});

if (user) {

if (bcrypt.compareSync(\_user.password, user.password)) {

ctx.session.user = user;

ctx.body = {

status: true,

msg: "Login success"

};

} else {

console.log('password is not correct so failed');

ctx.body = {

status: false,

msg: "Password not correct"

};

}

} else {

ctx.body = {

status: false,

msg: "Account not exist"

};

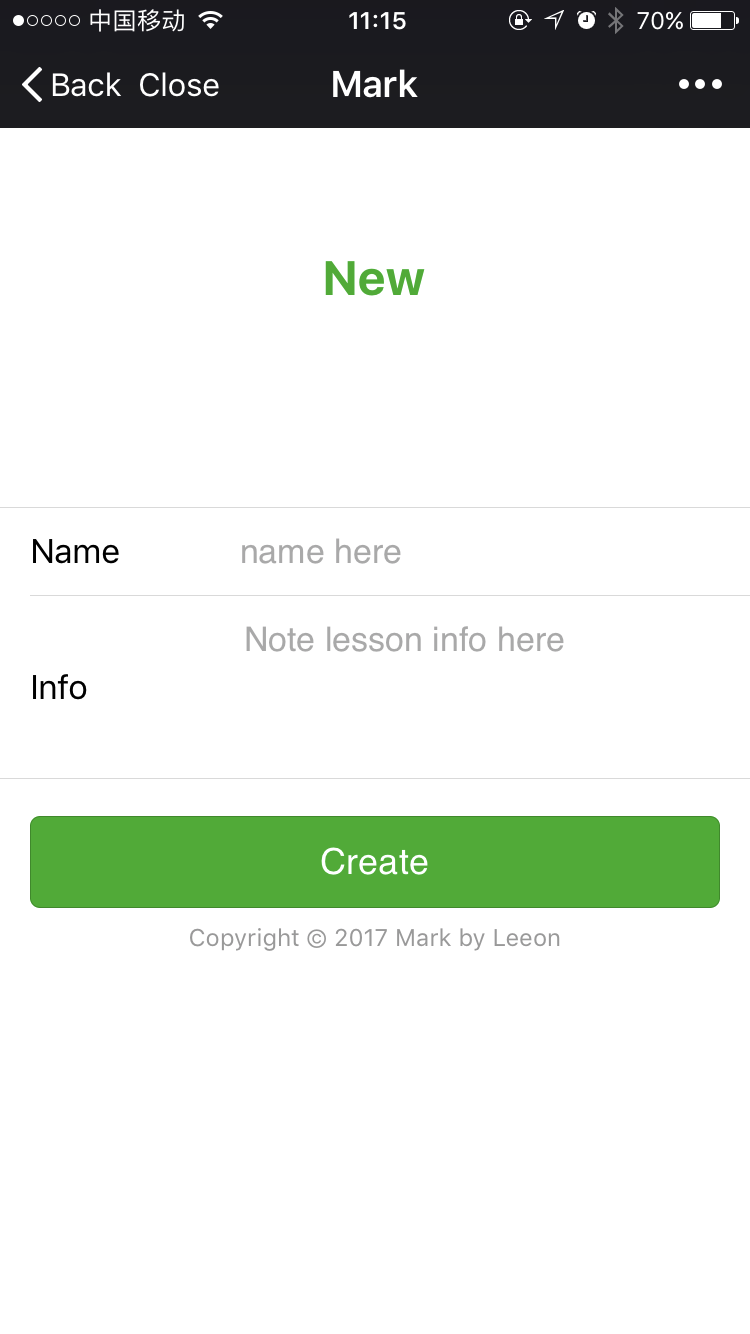
}

};

## 4.2 课程管理模块

### 4.2.1 用户创建课程模块

用户创建课程：



**图4-3 创建课程页面**

用户创建课程功能：用户登录成功后，可以从个人主页进入创建课程页面，用户只需要填写课程名称和课程信息即可以提交创建请求。提交创建之后，前端会对表单进行校验，信息没有填写完整就提交创建，请求不会被提交到后台，页面会提示填写完整信息后再进行提交。创建成功后，页面提示创建成功，并且跳转到课程详情页。

具体实现如下：

前端校验：

$("#createSub").click(function(e) {

if ($("#nameIn").val() && $("#infoIn").val()) {

var data = {

name: $("#nameIn").val(),

info: $("#infoIn").val()

};

$.ajax({

url: "create",

type: "POST",

contentType: "application/json; charset =utf-8",

data: JSON.stringify(data)

}).done(function(res) {

if (res.status) {

$.toptip("Create success", "success");

window.location.href = "detail/" + res.data.lid;

} else {

$.toptip("Network error", "error");

}

});

} else {

$.toptip("Complete info please", 'error');

}

});

后端实现：

exports.createLesson = async function(ctx, next) {

let lessonData = ctx.request.body;

let userSession = ctx.session.user;

lessonData.teacher = userSession.name;

lessonData.tid = userSession.\_id;

lessonData.follownum = 1;

lessonData.follower = [{

uid: userSession.\_id,

uname: userSession.name,

uphoto: userSession.img

}];

const newLesson = await mLesson.save(lessonData);

if (newLesson) {

let createLesson = {

lid: newLesson.\_id,

lname: newLesson.name,

lteacher: newLesson.teacher,

limg: newLesson.img

};

let followLesson = {

lid: newLesson.\_id,

lname: newLesson.name,

lteacher: newLesson.teacher,

lmsg: false

};

userSession.clesson.push(createLesson);

userSession.following.push(followLesson);

const updateResult = mUser.update(userSession);

if (updateResult) {

ctx.body = {

status: true,

msg: "Create success",

data: {

lid: newLesson.\_id

}

};

} else {

ctx.body = {

status: false,

msg: "Network error",

data: {}

};

}

} else {

ctx.body = {

status: false,

msg: "Network error",

data: {}

};

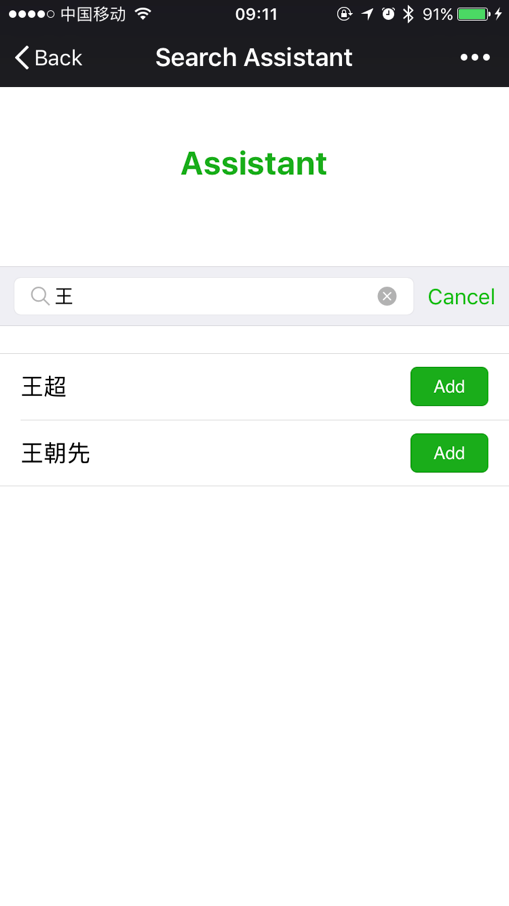
}

};

### 4.2.3 教师管理助教

课程中教师角色管理助教包括教师给用户添加助教权限以及教师删除用户助教权限。

#### 4.2.3.1教师添加助教：



**图4-4 搜索用户页面**

教师通过搜索用户，Mark 打分系统支持模糊搜索，搜索到用户后，系统展示一个添加按钮，点选添加按钮即可授权用户管理员权限，添加成功后，可以继续选择添加，具体实现如下：

exports.addAssistant = async function(ctx, next) {

var data = ctx.request.body;

const assistant = await mUser.findOne({

\_id: data.uid

});

const lesson = await mLesson.findOne({

\_id: data.lid

});

const flag0 = await mUser.addFollowing({

\_id: data.uid

}, {

lid: lesson.\_id,

lname: lesson.name,

lteacher: lesson.teacher,

lmsg: false

});

const flag1 = await mUser.addMLesson({

\_id: data.uid

}, {

lid: lesson.\_id,

lname: lesson.name,

lteacher: lesson.teacher,

limg: lesson.img

});

const flag2 = await mLesson.addAssistant({

\_id: data.lid

}, {

uid: assistant.\_id,

uname: assistant.name,

uphone: assistant.phone

});

const flag3 = await mLesson.addFollower({

\_id: data.lid

}, {

uid: assistant.\_id,

uname: assistant.name,

uphoto: assistant.img

});

const flag4 = await mLesson.addFollowNum({

\_id: data.lid

})

ctx.body = {

status: true,

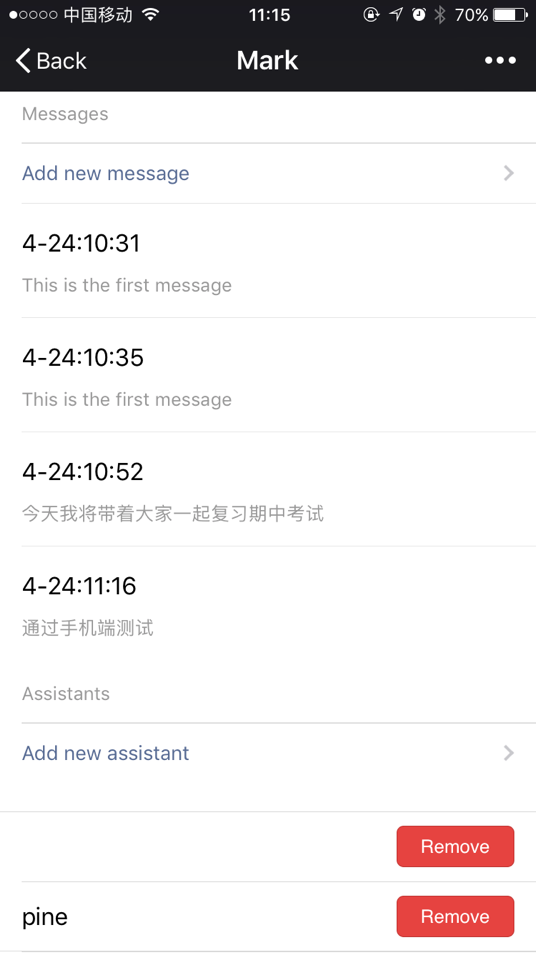
msg: "Add success",

data: {}

};

};

#### 4.2.3.2 教师删除助教



**图4-5 移除助教页面**

在课程详情页，会列出助教列表，并展现删除按钮，教师只需要点选删除按钮即可删除用户管理员权限，删除后，用户会在列表中移除，具体实现如下：

exports.delAssistant = async function(ctx, next) {

var data = ctx.request.body;

const flag0 = await mUser.delFollowing({

\_id: data.uid

}, {

lid: data.lid

});

const flag1 = await mUser.delMLesson({

\_id: data.uid

}, {

lid: data.lid

});

const flag2 = await mLesson.delAssistant({

\_id: data.lid

}, {

uid: data.uid

});

const flag3 = await mLesson.delFollower({

\_id: data.lid

}, {

uid: data.uid

});

const flag4 = await mLesson.delFollowNum({

\_id: data.lid

});

ctx.body = {

status: true,

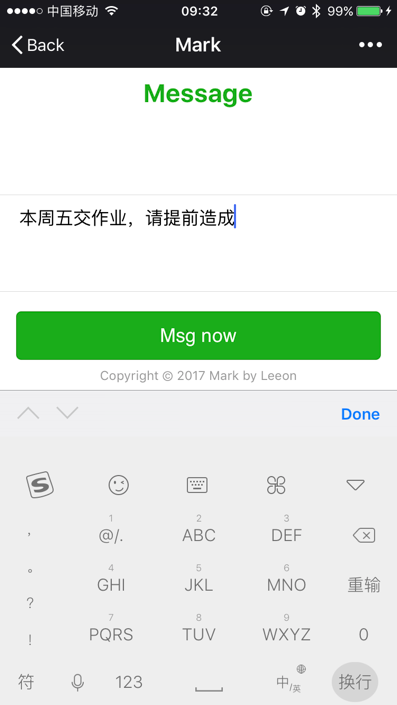
msg: "Remove success",

data: {}

};

};

### 4.2.3教师发布课程信息：



**图4-6 发布课程消息页面**

从课程详情页可以进入发布课程消息页面，输入课程消息后，点选发布按钮即可发布课程消息，Mark 打分系统在存储课程消息时会记录发布时间，发布人等消息，发布成功后，发布页面跳转到课程详情页，课程详情页会按时间先后展示课程消息，具体实现如下：



**图4-7 课程消息页面**

exports.createMsg = async function(ctx, next) {

let data = ctx.request.body;

let userSession = ctx.session.user;

let msg = {

mtime: new Date(),

mcontent: data.msg,

receiver: []

};

let query = {

\_id: data.lid

};

const newLesson = await mLesson.addMsg(query, msg);

if (newLesson) {

ctx.body = {

status: true,

msg: "Msg success",

data: {}

};

} else {

ctx.body = {

status: false,

msg: "Network error",

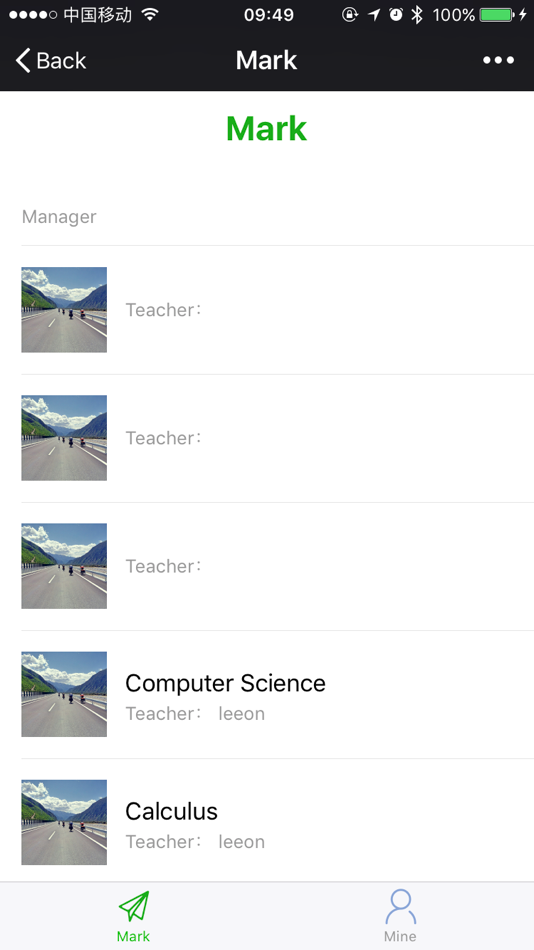
data: {}

};

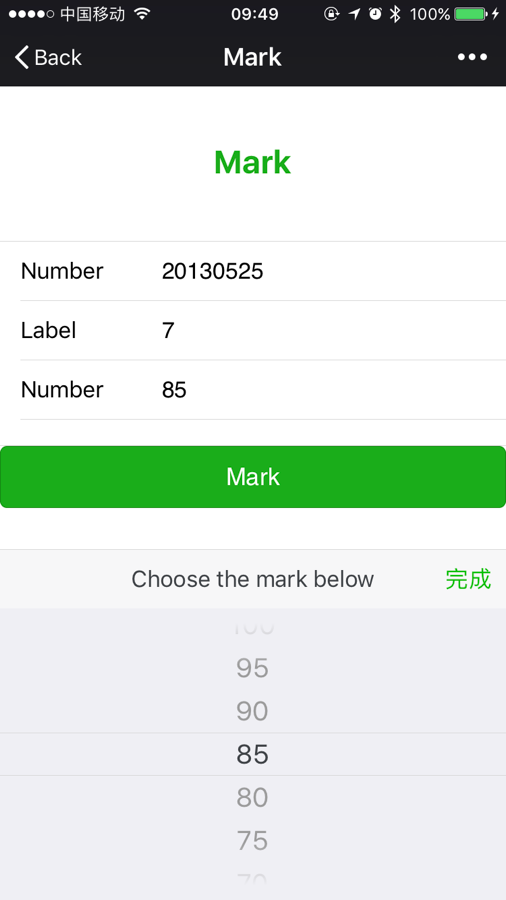
}

};

### 4.2.4 助教评分



**图4-8 管理课程列表**



**图4-9 助教评分页面**

助教评分是 Mark 评分的核心功能，用户在被教师添加成助教角色后，系统首页会出现管理课程列表，进入对应课程页面，页面展示的即是助教打分的页面，评分时需要添加学生学号、课程号，作业得分三个参数，为了提高助教评分效率，课程号和评分的填写摒弃了输入框填写的方式，全部以滚轮组件的方式填写，信息未填写完整，点选评分，请求将不会被提交，并提示需要填写完整的信息，提交完整的信息，并评分成功，页面会提示评分成功，表单清空，可以继续评分，具体实现如下：

前端实现：

$("#subMark").click(function(e) {

if($("#mark").val() && $("#label").val() && $("#number").val()) {

$.toptip("Mark success", "success");

var data = {

number: $("#number").val(),

label: $("#label").val(),

score: $("#mark").val(),

lid: $("#content").attr("lid")

};

$.ajax({

url: "/mark",

type: "POST",

data: data,

content: "application/json; charset=utf-8"

}).done(function(res) {

if(res.status) {

$.toptip("Mark success", "success");

} else {

$.toptip("Network error", "error");

}

});

} else {

$.toptip("Complete info please", "error");

}

});

后端实现

exports.mark = async(ctx, next) => {

let data = ctx.request.body;

const stu = await mScore.findOne({

number: data.number,

lid: data.lid

});

console.log(stu);

var score = stu.mark;

console.log(score);

score[data.label - 1] = data.score;

const result = mScore.update({

number: data.number,

lid: data.lid

}, score);

if (result) {

ctx.body = {

status: true

}

} else {

ctx.body = {

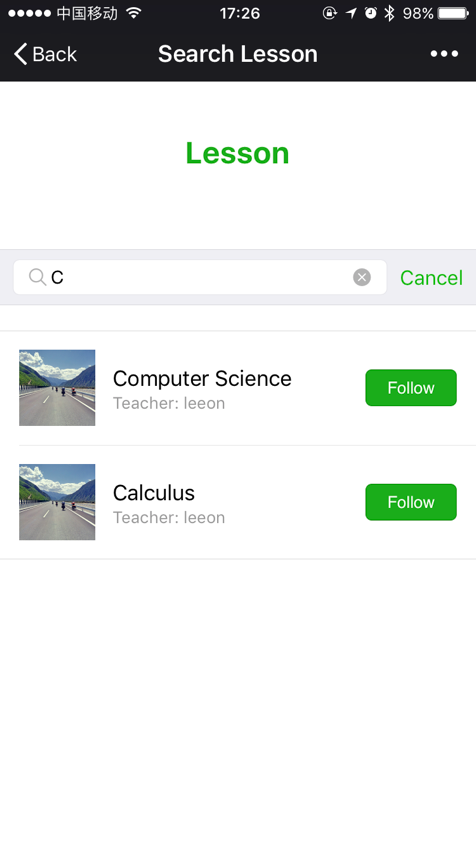
status: false

}

}

}

### 4.2.5用户管理关注课程



**图4-10 搜索课程页面**

用户登录 Mark 系统之后，从个人首页即可进入查询课程页面，搜索课程支持模糊搜索课程名，搜索的结果会展示相关课程列表，并且提供关注按钮，用户只需要点选关注按钮，即可关注该课程，页面会提示关注课程成功，已关注课程将会在搜索结果课程列表中移除，关注成功后可以继续搜索课程进行关注，用户已关注的课程将会出现在个人页面的已关注列表中。

具体实现如下：

exports.follow = async function(ctx, next) {

const lessonId = ctx.params.lid;

const user = ctx.session.user;

const lesson = await mLesson.findOne({

\_id: lessonId

});

const flag0 = await mUser.addFollowing({

\_id: user.\_id

}, {

lid: lesson.\_id,

lname: lesson.name,

lteacher: lesson.teacher,

lmsg: false

});

const flag1 = await mLesson.addFollower({

\_id: lessonId

}, {

uid: user.\_id,

uname: user.name,

uphoto: user.img

});

const flag2 = await mLesson.addFollowNum({

\_id: lessonId

})

ctx.body = {

status: true,

msg: "Follow success",

data: {}

};

};

## 4.3 文件模块

导入学生名单 Excel：

exports.uploadExcel = async function(ctx, next) {

console.log(ctx);

upload.single("list")

const list = nodeXlsx.parse('./public/excel/list.xlsx');

const data = list[0].data;

for (var i = 1; i < data.length; i++) {

let stuData = {

lid: "58fc1566575ae604393725cd",

class: data[i][2],

name: data[i][1],

number: data[i][0],

mark: []

};

let result = mScore.save(stuData);

console.log(data[i]);

console.log(result);

}

ctx.redirect('/mine');

};

## 4.4搜索模块

### 4.4.1搜索用户功能

教师用户在添加助教时，需要用户搜索功能，搜索支持模糊搜索，例如需要查询“李俊冬”这个用户，我只需要搜索 “李”或者“李俊”这类词条，“李俊冬”就会出现在搜索结果列表。

具体实现如下：

exports.fuzzyFind = async function(ctx, next) {

const result = await mUser.find({

name: new RegExp(ctx.params.name)

});

ctx.body = {

status: true,

msg: "search success",

data: result

};

};

### 4.4.2搜索课程功能

学生用户在添加助教时，需要课程搜索功能，课程搜索页面搜索框中输入需要搜索的课程名称，回车即可搜索到相关的课程，课程搜索支持模糊查询，例如需要查询“computer science”这门课，只需要填“computer”，“computer science”会出现在搜索结果列表中，具体实现如下：

exports.fuzzyFind = async function(ctx, next) {

const result = await mUser.find({

name: new RegExp(ctx.params.name)

});

ctx.body = {

status: true,

msg: "search success",

data: result

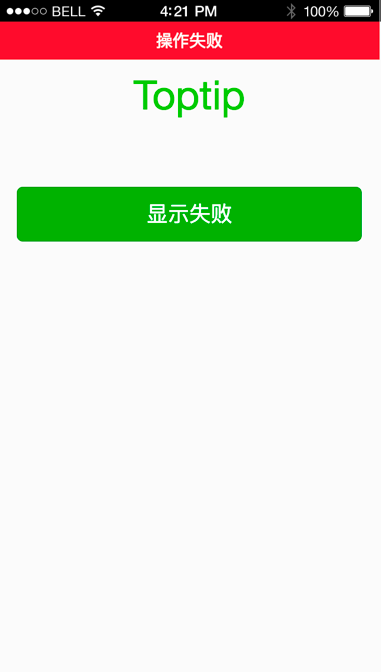
};

};

## 4.5用户交互友好

Mark 系统使用了微信官方提供的设计语言 WEUI 框架，对微信原生友好，用户界面简洁优雅，用户交互则使用了 WEUI 的衍生版本，jQuery-WEUI，在 一些操作时，做到用户交互友好。

交互提示：



**图4-11 页面交互提示**

# 第五章 系统运行与测试

## 5.1系统测试环境

### 5.1.1系统测试硬件环境

微信服务号教学评分系统的设计与实现需要服务器一台，开发机一台。

机器相关配置如下：

开发机配置: 2 GHz Intel Core i5，8 GB 1867 MHz LPDDR3，MacOS Sierra 10.12.5

服务器配置: 阿里云服务器 1G，1 核，CentOS 7

### 5.1.2系统测试软件环境

数据库: MongoDB 3.2.10

Node.js 版本： version 7.10

NPM 版本： version 4.2.0

NVM 版本： 0.13.1

测试平台： 微信 6.5.8 ，iOS 10.12

### 5.1.3 Node.js 服务器的配置和启动

在 CeontOS 中安装 NVM 成功后，通过 nvm install node 7 命令安装 Node.js version 7.10，安装成功后，进入项目目录，使用 npm install 命令安装项目所依赖的包，依赖包安装成功后，在项目根目录下，使用 node app.js 命令启动项目，即完成了项目的配置

## 5.2 服务器系统测试

### 5.2.1 测试需求概述

系统测试主要针对测试系统中关键功能的实现的正确性及稳定性，具体的测试需求如下：

1.基础功能测试：针对用户注册、用户登录、更新个人信息、搜索用户、搜索课程、上传表格、下载表格等功能进行测试

2.页面测试：针对注册页面、登录页面、课程详情页面、搜索页面、个人详情页面、更新个人信息页面、创建课程页面、搜索用户页面、助教打分页面是否正确工作。

3.教师角色功能测试：针对教师角色创建课程、更新课程信息、发布课程消息、更新学生名单、添加助教、删除助教、删除课程、下载学生成绩单等功能进行测试，测试功能运行的正确性以及稳定性。

4.助教角色功能测试：针对助教用户的评分功能进行测试。

5.常规用户功能测试：针对常规用户关注课程、取消关注课程的功能进行测试。

### 5.2.2测试用例以及结果

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 输入 | 预期结果 | 实际结果 |
| 注册1 | 邮箱：teacher@qq.com  密码：123qwe  确认密码：123qwe | 页面提示成功信息，页面跳转到登录页面 | 页面提示账户注册成功，页面跳转回登录页面 |
| 注册2 | 邮箱：teacher@qq.com  密码：123qwer  确认密码：123qwe | 页面提示错误信息，注册未完成 | 页面提示两次密码不相同，注册失败 |
| 注册3 | 邮箱：teacher@qq.com  密码：123qwe  确认密码：123qwe | 页面提示错误信息，注册未完成 | 页面提示该邮箱已被注册，注册失败 |
| 登录1 | 邮箱：teacher@qq.com  密码：123qwe | 页面提示成功信息，登录成功，页面跳转到系统首页 | 页面提示用户登录成功，页面跳转到系统首页 |
| 登录2 | 邮箱：teacher@qq.com  密码：123qwer | 页面提示错误信息，登录未完成 | 页面提示密码错误，登录失败 |
| 登录3 | 邮箱：teache@qq.com  密码：123qwe | 页面提示错误信息，登录未完成 | 页面提示该邮箱用户不存在，登录失败 |
| 创建课程 | 输入课程名称以及课程信息  课程名：计算机网络  课程信息：上课时间每周六下午两点半 | 页面提示成功信息，页面跳转到课程详情页 | 页面提示创建课程成功，页面跳转到课程详情页 |
| 发布课程消息 | 输入课程消息，点选发布课程消息  课程消息：本周的作业时完成书本第八章的课后习题 | 页面提示操作信息，页面跳转回课程详情页面 | 页面提示课程消息发布成功，页面跳转回课程详情页。 |
| 搜索用户 | 进入用户搜索页面，输入用户姓名，回车进行搜索  输入： 王 | 页面展示搜索结果 | 页面加载出搜索到的用户  显示搜索结果：  王朝  王伟 |
| 教师添加课程助教 | 在搜索结果页面，点击搜索结果用户的添加按钮 | 页面提示操作结果 | 页面提示添加成功，已添加的用户移除列表 |
| 教师删除课程助教 | 在课程详情页面，点击助教列表中，相应用户的移除按钮 | 页面提示操作结果 | 页面提示移除助教成功，相应用户从助教列表中移除。 |
| 教师上传学生名单 | 在课程详情页，选择规定格式的学生名单 Excel ，点击上传 | 页面提示操作结果 | 页面提示上传表格成功，学生名单导入成功 |
| 教师导出学生成绩 | 在课程详情页，点选导出学生成绩按钮 | 页面提示操作结果 | 页面提示保存学生成绩单，保存完成后，页面提示学生成绩导出成功 |
| 教师删除课程 | 在课程详情页，点选删除课程按钮 | 页面提示操作结果 | 页面提示删除课程成功，并跳转回课程列表页 |
| 助教评分1 | 助教进入评分页面，输入学生的学号，课程号，课程得分，点击评分按钮  学号：20130525  课程号：3  成绩： 90 | 页面提示操作结果 | 页面提示评分成功，并且清空输入框，可以继续评分 |
| 助教评分2 | 助教进入评分页面，输入学生的学号，课程号，课程得分，点击评分按钮  学号：201305257  课程号：3  成绩： 90 | 页面提示操作结果 | 页面提示该学生不存在，请核对信息后进行评分，评分失败 |
| 用户更新个人信息 | 用户进入个人主页，查看个人详细信息，更新个人信息，点击更新按钮 | 页面提示操作结果 | 页面提示个人信息更新成功，返回到个人详细信息页面 |
| 用户搜索课程 | 用户进入搜索课程页面，输入需要搜索的课程名称，回车进行搜索 | 页面展示搜索到的课程列表 | 搜索结果列表展示在搜索框下，并且展示了用户关注按钮 |
| 用户取消关注课程 | 用户进入关注的课程详情页，点选取消关注课程的按钮 | 页面提示操作结果，页面会进行调整 | 页面提示用户取消关注课程成功，并且页面跳转回关注的课程列表页面 |
| 用户注课程 | 用户在搜索课程页面，搜索到对应的课程，点选对应课程的关注按钮 | 页面提示操作结果 | 页面提示用户关注课程成功，已关注课程从搜索的课程列表结果页面中移除 |

### 5.2.3测试总结

经过反复多次多个账户的测试，上述测试用例在微信的 Android 与 iOS 端均正常稳定运行，期间出现了几处小问题，经过仔细的调试代码与测试都已经解决，并且在此之上还对某些地方的页面进行了调整。目前整套评分系统运行正常，但是某些特殊的问题没有测试出来，需要在后期的开发中继续完善。

# 结论

微信服务号教学评分系统的设计与实现，完成了Mark 评分系统的中的用户使用微信号登录、账号管理，用户管理个人信息，用户创建课程，管理课程信息，上传学生名单、教师管理助教、助教评分、用户搜索关注课程，教师导出成绩单等功能。

整个系统的设计与实现全部建立在 Koa.js + MongoDB 应用架构之上，代码结构清晰，功能分层实现，开发过程中设计的知识点较多，经过此次系统的开发工作，总结出该评分系统具有以下优点：

1.系统开发效率高：系统开发过程中，使用了大量的开源 Node.js 项目包，例如 Mongoose、koa-body、multer 等，使得开发效率得到很大的提高。

2.系统可扩展性高：Mark 打分系统使用了 Koa.js 2.0，通过 Router 管理所有功能的入口，各功能之间相互独立，降低了模块之间的耦合度，使得系统的扩展性得到很大的提高。

3.系统可维护性高：系统中使用的文件、函数以及参数、变量等都采用了与实际的功能相关的英文单词进行命名，并且在一些必要的位置标记有注释，方便今后进行维护。

4.系统安全性高：系统中使用了Koa.js + MongoDB 架构，对用户登录以及对资源的访问都进行了权限控制，保证了资源的安全性，在此之上，使用 Koa.js 在服务端进行渲染页面，所有页面的访问都需要先进行校验。进一步保证了资源的安全性。

5.系统面对用户友好：系统使用了微信官方提供的UI 组建 WEUI，因此，整个系统的设计符合微信官方的设计规范，用户在服务号使用系统时会获取与使用原生微信相近的使用体验，在此之上，使用了 WEUI 的衍生版本 jQuery-WEUI，使得系统面向用户操作友好，对于用户的操作，无论正确与否，系统都会及时的给予提示，提示的方式符合微信官方的设计规范。

Mark 评分还只是一个产品原型，系统功能可以想象的还有很大空间，因为时间与精力的缘故，还有一些设计的功能暂时还没有实现，需要在后续的开发中继续改进。

1.增加 PC 端支持：虽然评分系统设计之初是以移动端作为主要平台，但是对于文件操作等一类操作，移动端的使用体验距离 PC 端仍旧有很大差距，为了方便教师用户更方便地操作，后续会增加 PC 端支持。

2.课程信息可选择配置：目前的课程信息发布是默认所有人接收到，而且新消息发布，用户不会收到提醒，只能先进入课程详情页面，才能查看是否有新的课程消息发布，后续开发会改进这部分体验。

3.测试不完整，由于目前系统的功能较少，测试的用户比较少，在多用户使用时，可能会有新的问题出现，这些问题在后续的开发中都是需要改进的。

通过这段时间对 Mark 评分系统的开发，加深了自己对 Node.js 以及 Koa.js 框架的理解，对 Koa.js 2.0 一边进行学习，一边进行实践使自己的动手能力有了很大的提高，同时有了这次的开发经历，提高了自己对于项目开发的信心，对于今后新的 Node.js 项目会有更好的信心，有了这次的开发经验，对今后的职业道路也会大有裨益。

# 致谢

微信服务号教学评分系统的设计与实现以及学位论文的完成是在导师杨雷老师的精心指导下完成的。杨老师严谨的治学态度，严肃的科学态度，深深的激励着自己要好好努力。从开始选择课题到系统的完成，杨老师在整个过程中都给了我很多指导，使我学到了很多知识。自进入杨老师的实验室以来，不仅在学业上得到了杨老师的精心指导，同时在思想上和生活上也得到了很大的帮助，在此谨向辛勤教学的杨老 师致以诚挚的谢意和崇高的敬意。同时还要感谢实验室的学姐学长，在他们的帮助下，当遇到问题时才一步一步的 解决，在进入实验室后正是在他们的指导帮助中渐渐的学会了基本的开发知识，还要 感谢在实验室中的学弟学妹们，他们在实验中的学习的认真态度和积极向上的生活态 度时刻的在促使我积极向上。最后还要感谢同我一起完成毕业设计的同学们和父母，有他们的鼓励我才完成了 系统的设计与实现。

# 参考文献

[1] 白浩, 郝晶晶. 微信公众平台在高校教育领域中的应用研究[J]. 中国教育信息化:

基础教育, 2013 (2): 78-81.

[2]王萍. 微信移动学习的支持功能与设计原则分析[J]. 远程教育杂志, 2013, 31(6): 34-41.

[3]黄浩波, 何卫华, 叶青. 微信及其在图书馆信息服务中的应用[J]. 图书馆学刊, 2013 (1): 62-64.

[4]孔云, 廖寅, 资芸, 等. 基于微信公众账号的图书馆移动信息服务研究[J]. 情报杂志, 2013, 32(9): 167-170.

[5]蒋琦琦. 微信服务在图书馆应用的探索与实践[J]. 数字技术与应用, 2013 (7): 201-202.

[6] (美) Cantelon M, Harter M, Holowaychuk T J, et al. Node. js in Action[M]. Manning, 2014.

[7](美)Dayley B. Node. js, MongoDB, and AngularJS Web Development[M]. Addison-Wesley Professional, 2014.

[8] (美)Teixeira P. Professional Node. js: Building Javascript based scalable software[M]. John Wiley & Sons, 2012.

[9] (美)Rauch G. Smashing Node. js: JavaScript Everywhere[M]. John Wiley & Sons, 2012.