**广州大学华软软件学院**

**本科毕业设计任务书**

**设计题目 高校教师数据采集与汇总系统**

**的设计与开发**

**系 别 游戏系**

**专 业 网络与新媒体**

**班 级 15网络与新媒体2班**

**学 号 1540624214**

**学生姓名 谢昇**

**指导教师 袁冠远**

下发时间： 2018年 10 月 24 日

**毕业设计须知**

1、认真学习和执行广州大学华软软件学院学生毕业论文（设计）工作管理规程；

2、努力学习、勤于实践、勇于创新，保质保量地完成任务书规定的任务；

3、遵守纪律，保证出勤，因事、因病离岗，应事先向指导教师请假，否则作为缺席处理。凡随机抽查三次不到，总分降低10分。累计缺席时间达到全过程l／4者，取消答辩资格，成绩按不及格处理；

4、独立完成规定的工作任务，不弄虚作假，不抄袭和拷贝别人的工作内容。否则毕业设计成绩按不及格处理；

5、毕业设计必须符合《广州大学华软软件学院普通本科生毕业论文（设计）规范化要求》，否则不能取得参加答辩的资格；

6、实验时，爱护仪器设备，节约材料，严格遵守操作规程及实验室有关制度。

7、妥善保存《广州大学华软软件学院本科毕业设计任务书》。

8、定期打扫卫生，保持良好的学习和工作环境。

9、毕业设计成果、资料按规定要求装订好后交指导教师。凡涉及到国家机密、知识产权、技术专利、商业利益的成果，学生不得擅自带离学校。如需发表，必须在保守国家秘密的前提下，经指导教师推荐和院领导批准。

|  |  |  |
| --- | --- | --- |
| 课题名称 | 高校教师数据采集与汇总系统的设计与开发 | |
| 完成日期： 2019年 4月 10日 | | |
| 1. 题目来源及原始数据资料：   题目来源： 老师推荐  原始数据资料：   1. 收集汇总Excel表格资料的时候，总是会经历繁琐的复制粘贴和分类汇总，本系统意图节省这一流程，在同类型的工作中，节约人力，提高效率； 2. Node.js快捷的开发流程，Javascript方便的开发体验，使得Web应用在跨平台开发上有着绝对的优势，用户不必安装软件，打开浏览器即可使用； | | |
| 二、毕业设计要求：   1. 开发工具：Node.js; 2. 程序有良好的设计风格和代码规范 3. 了解与游戏程序相关的软件技术 4. 程序能正确地运行 5. 进行必要的调研和资料搜集、文献阅读； 6. 要有完整的开发文档 7. 有游戏玩法详细说明文档 8. 每周有开发进度报告   项目目标：  1、以工号为标识登录，有相应的账号管理功能  2、后台生成所需xlsx文件，前端下载对应文件  3、前端读取xlsx信息，检查纠错，编译后上传  4、后台审核用户上传的信息、管理、汇总  5、后台可以通过系别、部门等分类导出  6、后台查询当前上传人员进度，圆图展示，数量标识  7、增加附件上传模块，后台链接对应论文，重命名文件进一步管理  8、增加信息反馈渠道，后端可记录、查询、标记、备注、删除  9、登录功能升级，对接Mysise系统  10、局域网部署服务  11、Electron打包、重构测试 | | |
| 三、进度安排、应完成的工作量：   1. 2018年10月23日前，下达任务书。 2. 2018年10月30日前，上交任务书，完成毕业论文开题，开始游戏项目设计。 3. 2018年 12月 5日前，完成游戏项目第一版 4. 2019年1月20日前，完成游戏项目第二版，开始撰写毕业论文。 5. 2019年2月10日前，提交毕业论文的初稿。 6. 2019年3月10日前，提交毕业论文的完成稿。 7. 2019年3月15日前，提交毕业论文的打印稿。 8. 2019年3月24—3月30日，组织毕业论文（设计）答辩及成绩评定。 | | |
| 四、主要参考文献  [1]林朝炽.基于Elastic Stack的分布式数据采集与分析平台探索[J].网络安全技术与应用,2019(02):38+61.  [2]周永辉. 基于网络的短信收发系统的设计与实现[D].郑州大学,2017.  [3]杨诚. 基于Android的心理咨询服务系统的设计与实现[D].电子科技大学,2013.  [4]张志恒,张显库,杨光平,马超,冯永孝.基于Visual C++的Excel工作簿数据处理[J].软件导刊,2017,16(01):135-137.  [5]高小茜. 高校业务数据管理系统的研究与实现[D].西安科技大学,2016.  [6]罗文,刘星,兰全祥.基于Node.js的创新团队管理系统的设计与实现[J].赤峰学院学报(自然科学版),2018,34(10):83-85.  [7]陈健.广东高校图书馆数字电子期刊资源分布与利用研究[J].图书馆学刊,2017,39(02):39-44.  [8]胡佳静. 基于electron的待办事项管理app开发[D].华中科技大学,2018.  [9] 不止前端. electron入门心得[Z]. https://www.cnblogs.com/buzhiqianduan/p/7620099.html,2017  [10] 秋雨仲夏. 记录Python模拟登录爬取Mysise学生管理系统的信息[Z]. https://www.jianshu.com/p/3576b0d4c457, 2017.  [11]罗贤春,余波,姚明.信息价值研究的热点领域可视化分析[J].现代情报,2015,35(11):120-126.  [12]于文洋.针对办公自动化系统的信息安全问题的探讨[J].办公自动化,2018,23(24):44-45.  [13] muffinfish.Node.js：浅析高并发与分布式集群[Z].https://segmentfault.com/a/1190000015841624，2018. | | |
| 指导教师（签名）： | | 系（教研室）主任（签名）： |

摘要 21世纪以来，互联网技术发展迅速，web2.0时代的进程不断推进。用户生产内容的情况随处可见，打开手机微博、抖音、腾讯微视等软件每天每时每分每秒都在不断的出现新产品，对人们的生活、工作方式产生了极大的变化。尤其是在各大高校“象牙塔”内，随着“互联网 +”、“自动化办公”的发展，各项原本人工耗时耗力的任务，通过数字化、自动化处理，变得方便快捷。高校内，校园信息化的需求在不断增加，广州大学华软软件学院近期新增的数字设施“智慧食堂”就体现了这一点，各个高校都在不断增添信息化建设。而每当校内进行教师考核的时候，考核信息的采集收集，仍然采用人工整理的方式，占用了大量的工作时间和精力；高校教师数据采集与汇总系统的开发，便是为了解决这一窘境。

关键词 Node.js；分布式；Vue；Electron；MongoDB

**ABSTRACT** Since the 21st century, Internet technology has developed rapidly, and the process of web2.0 era has been advancing continuously. The production of content by users can be seen everywhere. New works are constantly emerging every minute of every day when users open their mobile phones' Weibo, Tik Tok, Tencent Weishi and other software, bringing about great changes to people's life and work style. Especially in the "ivory tower" of colleges and universities, with the development of "Internet +" and "Office Automation", various tasks that were originally time-consuming and labor-intensive have become convenient and quick through digitalization and automation. In colleges and universities, the demand for campus informatization is increasing, which is reflected in the "smart canteen" of digital facilities recently added by South China Institute of Software Engineering.GU. All colleges and universities are constantly adding informatization construction. However, when teachers are assessed in the school, the assessment information is still collected manually, which takes up a lot of work time and energy. The development of data collection and summarization system for college teachers is to solve this dilemma.

**KEY WORDS** Node.js; Distributed; Vue; Electron; MongoDB

目 录

[摘要 I](#_Toc5118883)

[ABSTRACT II](#_Toc5118884)

[目 录 III](#_Toc5118885)

[第1章 绪 论 1](#_Toc5118886)

[1.1 背景简介 1](#_Toc5118887)

[1.2 信息的价值 1](#_Toc5118888)

[1.3 信息自动化处理的意义 1](#_Toc5118889)

[1.3.1 自动化与信息安全 2](#_Toc5118890)

[1.3.2 准确高效处理事务 2](#_Toc5118891)

[第2章 需求分析 3](#_Toc5118892)

[2.1 开发背景 3](#_Toc5118893)

[2.2 用户需求 3](#_Toc5118894)

[2.2.1 高校教师 3](#_Toc5118895)

[2.2.2 信息收集人员 3](#_Toc5118896)

[2.3 业务分析 3](#_Toc5118897)

[2.3.1 高校教师 3](#_Toc5118898)

[2.3.2 信息收集人员 5](#_Toc5118899)

[第3章 技术栈介绍 6](#_Toc5118900)

[3.1 项目难点 6](#_Toc5118901)

[3.1.1 保留旧有的工作习惯 6](#_Toc5118902)

[3.1.2 方便数据筛选 6](#_Toc5118903)

[3.1.3 分布式 6](#_Toc5118904)

[3.2 前端Vue.js与ElementUI库 7](#_Toc5118905)

[3.2.1 自由搭配前端库 7](#_Toc5118906)

[3.2.2 组件化 7](#_Toc5118907)

[3.3 后端Node.js 7](#_Toc5118908)

[3.3.1 异步、事件驱动 7](#_Toc5118909)

[3.3.2 非阻塞I/O 8](#_Toc5118910)

[3.3.3 分布式处理 8](#_Toc5118911)

[3.4 桌面应用框架Electron 8](#_Toc5118912)

[3.4.1 跨平台的桌面应用框架 8](#_Toc5118913)

[3.4.2 兼容性说明 9](#_Toc5118914)

[3.4.3 调用系统的API 9](#_Toc5118915)

[3.4.4 IPC通讯模块 9](#_Toc5118916)

[第4章 概要设计 10](#_Toc5118917)

[4.1 总体介绍 10](#_Toc5118918)

[4.1.1 需求规定 10](#_Toc5118919)

[4.1.2 运行环境 11](#_Toc5118920)

[4.1.3 基本设计概念和处理流程 11](#_Toc5118921)

[4.2 接口设计 12](#_Toc5118922)

[4.2.1 高校教师接口 12](#_Toc5118923)

[4.2.2 信息收集员接口 13](#_Toc5118924)

[4.3 本章小结 14](#_Toc5118925)

[第5章 服务器设计 15](#_Toc5118926)

[5.1 服务器概要 15](#_Toc5118927)

[5.1.1 服务器目录设计 15](#_Toc5118928)

[5.1.2 登录拦截 16](#_Toc5118929)

[5.1.3 数据库连接 17](#_Toc5118930)

[5.1.4 日志输出 17](#_Toc5118931)

[5.2 登录 17](#_Toc5118932)

[5.2.1 功能描述 17](#_Toc5118933)

[5.2.2 Schema对象 18](#_Toc5118934)

[5.2.3 路由 18](#_Toc5118935)

[5.3 模板管理 18](#_Toc5118936)

[5.3.1 功能描述 18](#_Toc5118937)

[5.3.2 Schema对象 18](#_Toc5118938)

[5.3.3 路由 18](#_Toc5118939)

[5.4 数据管理 18](#_Toc5118940)

[5.4.1 功能描述 18](#_Toc5118941)

[5.4.2 Schema对象 19](#_Toc5118942)

[5.4.3 路由 19](#_Toc5118943)

[5.5 文件上传 19](#_Toc5118944)

[5.5.1 功能描述 19](#_Toc5118945)

[5.5.2 Schema对象 19](#_Toc5118946)

[5.5.3 路由 19](#_Toc5118947)

[第6章 数据库设计 20](#_Toc5118948)

[6.1 概念设计 20](#_Toc5118949)

[6.2 逻辑结构设计 21](#_Toc5118950)

[6.3 物理结构设计 21](#_Toc5118951)

[第7章 系统测试 22](#_Toc5118952)

[7.1 测试方案 22](#_Toc5118953)

[7.1.1 测试平台 22](#_Toc5118954)

[7.1.2 测试内容 22](#_Toc5118955)

[7.2 运行结果 22](#_Toc5118956)

[7.3 应用展示 23](#_Toc5118957)

[7.3.1 客户端首页 23](#_Toc5118958)

[7.3.2 模板下载列表 23](#_Toc5118959)

[7.3.3 数据上传 24](#_Toc5118960)

[7.3.4 文件上传 25](#_Toc5118961)

[7.3.5 已上传数据列表 25](#_Toc5118962)

[7.3.6 后台首页 26](#_Toc5118963)

[7.3.7 模板生成 27](#_Toc5118964)

[参考文献 28](#_Toc5118965)

[致 谢 29](#_Toc5118966)

1. 绪 论
   1. 背景简介

广州大学华软软件学院中，有着许许多多需要收集汇总信息的工作，教师考核、团员评议、新生报到等等，不胜枚举。学生或者老师每每录入信息，都需要反复经历下载、查看规范、认真核对、发送给处理人等步骤；信息在接收人手里，仍然要经历长时间的打开文件、复制粘贴、汇总排序。

为了简化教师考核的工作流程，本系统以不改变老师原本的工作习惯为宗旨，保留xls/xlsx填写资料的习惯，将原本发送到某一个人手里处理的工作，转换为计算机的工作，节省信息汇总以及分类的时间，直接得出结果。

信息汇总、筛选、分类、导出的工作，将借鉴Microsoft Access数据库软件，并预留出足够的自定义空间，结合实际使用体验，进行简化。

* 1. 信息的价值

在市场经济条件下，信息已经成为一种极其重要的商品。新闻同样是信息的一种，以小见大，信息的价值主要体现在——真实性、针对性和时效性三个方面。当然，不仅如此，信息的价值也是具有多维性的。不同的信息，对于不同的受众来说，具有不同的价值，价值的多寡，受到信息利用者的价值观影响。

高校教师数据采集与汇总系统，旨在统筹教师考核时收集的一切信息，分门别类，高效可视。其中涉及到的数据即信息，对于每个教师都十分重要，信息的精准与否，直接影响到教师考核的评价。

* 1. 信息自动化处理的意义

信息的自动化处理，利用计算机网络，收集大量同类信息，并提前筛查排列，利用图表，将结果展现到用户面前，能够极大地缩减工作时间，提高工作效率。

然而，随着“互联网 +”、“大数据 + 智能化”被大众所普遍地认可和接受，网络信息安全的问题也随之浮出水面，受到人们极大地关注。

* + 1. 自动化与信息安全

信息安全就是要保护信息系统中数据信息的安全性、完整性和机密性等重要特性。在高校教师数据采集与汇总系统中，一切数据采集的源头，由各位教师亲手发布，数据采集到服务器的过程中，全程通过计算机自主执行。在系统正常运行的情况下，保证信息采集的完整性和机密性，避免了人为操作导致的信息不匹配。保障了数据提供者（教师）的个人权益。

* + 1. 准确高效处理事务

在设计中，数据采集到服务器之后，将直接存储到数据库。数据存储的形式为json格式，相比原来的数据采集方式xlsx、xls表格，json能够被系统直接处理，省去了大量打开表格文件、复制、粘贴的时间，在导出为总表的时候更加方便快捷。

在信息收集完成之后，原本可能需要花费两到三小时的整理工作，通过系统，所需时间直接降到三十分钟内，极大地方便教师考核工作的开展。

1. 需求分析
   1. 开发背景

高校教师数据采集与汇总系统的诞生，需求主要在与信息安全、准确、高效地汇总而方便后续管理。节省了老师的精力，解放出更多的“生产力”。广州大学华软软件学院，每当需要进行教师考核的时候，一个个发出去的模板，回收时候的一张张xls表格，面对长长的一系列文件，需要汇总人员一个一个打开、复制，之后再粘贴到新表，工作繁复且容易出错。

假如在信息收集的源头，就读取Excel数据，进行初步处理，汇总到后台的时候，就是一张完整的信息列表，那不论是筛选、排序、分类还是到处，都将更加方便快捷。

系统适用范围：高校教师、学生（本系统的衍生学生版系统）。

* 1. 用户需求
     1. 高校教师

对于普通高校教师，本系统提供下列功能：

下载xlsx/xls；批量上传论文记录；上传证明文件（压缩包）；修改已上传记录；查看记录。

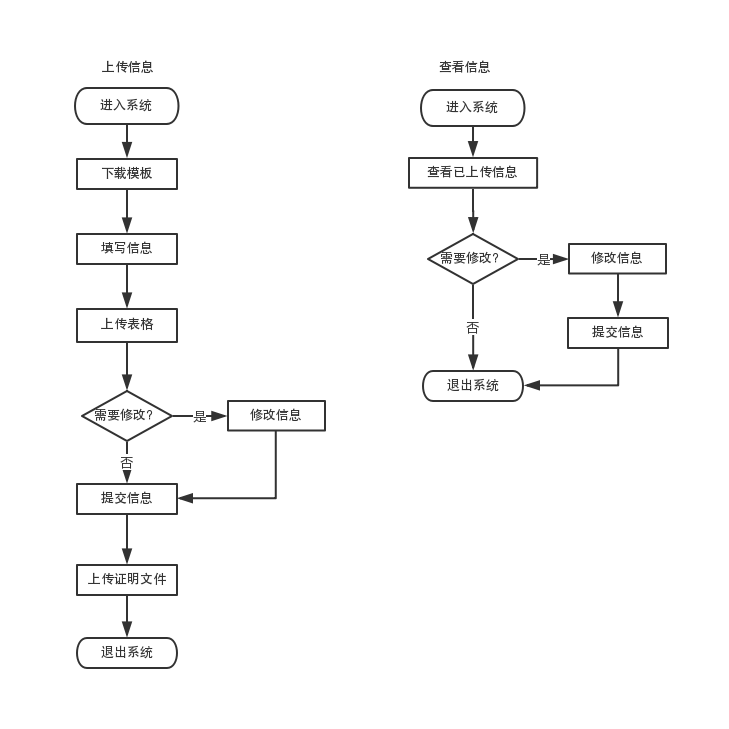
* + 1. 信息收集人员

对于单次工作的信息收集人员，本系统将提供下列功能：

发布信息模板；查看收集进度；查看已收集信息；筛选信息；导出xlsx。

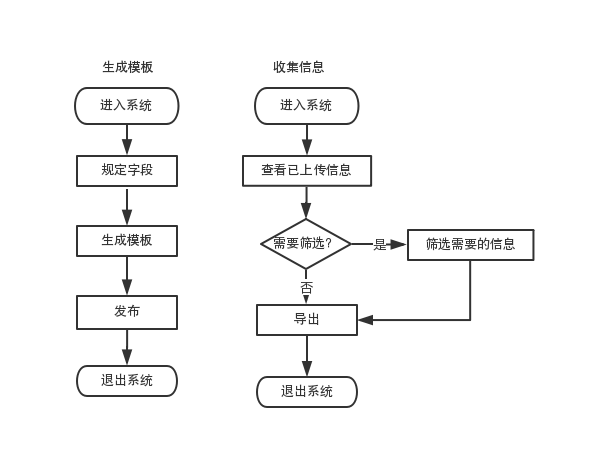
* 1. 业务分析
     1. 高校教师

高校教师进入本系统，下载信息收集模板，按照收集人员的规定，填写相关信息，上传xlsx/xls文件，有需要时修改上传信息；上传证明文件。如图2-1所示。

图2-1 高校教师业务流程图

* + 1. 信息收集人员

信息收集员进入本系统，发布信息收集模板，查看已收集信息，按需筛选，导出。如图2-2所示。

图2-2 信息收集员业务流程图

1. 技术栈介绍
   1. 项目难点
      1. 保留旧有的工作习惯

在过去，教师考核的信息收集，主要依赖于excel表格的填写和收集。此类信息主要以二维表的形式保存，而在系统中如何延续这种数据风格，可以考虑的数据库类型有：MySQL，SQL Server，Oracle，MongoDB。

为了实现高度自由的字段定义，本系统将采用MongoDB作为数据库。

* + 1. 方便数据筛选

即便在数据导出之后，可以通过excel强大的筛选功能，进行信息分类、排序、筛选；却也不如在导出之前，就将数据分门别类，等待导出。表格排序的工具，本系统将使用JQuery插件tablesorter，有了这个插件，用户即可在前端直接对杂乱的数据进行排序。

至于分类功能，本系统将提供部门分类：

1. 十系部:

财会系,电子系,管理系,国贸系,软件系,外语系,网络系,计算机系,数码系,游戏系

1. 基础部门:

思政部,基础部,教务处,教学保障处

1. 行政部门:

党群工作部,人事处,团委,学生处,招生办

1. 其他部门:

创新创业学院,对外交流,外包基地,图书馆,心理中心

* + 1. 分布式

构建一个承载大量访问量的网站，对于服务器的压力非常高，往往需要提高响应速度的情况下，可以考虑静态缓存或者数据缓存；然而后期都需要考虑分布式缓存、服务器、数据库等。

考虑到实际，在校园局域网内，可以忽略网速影响，那么压力就集中在服务器的单位时间访问量上面。为了解决这个问题，本系统了采用Node.js的Koa2框架的WebServer。

* 1. 前端Vue.js与ElementUI库
     1. 自由搭配前端库

Vue是一套用于构建用户界面的渐进式框架。组件化的开发模式，大大缩减了前端开发成本。Vue本身并没有足够的预设组件可用，但只要配合ElementUI库这样丰富的前端组件，前端的构建，就变得轻而易举了。

* + 1. 组件化

Vue实现了MVVM（Model-View-ViewModel）的双向绑定和数据监控，在组件化方面，非常优秀。通过ViewModel配合所对应的数据进行页面渲染，使得界面与数据双向绑定，即界面改变，数据跟着变，反之亦然。并且在频繁的数据变换中依然可以保证流畅的执行效率，对于需要大量且多次渲染同一个页面的情况，来说非常合适。

著名网站“哔哩哔哩弹幕网”便是使用Vue.js重构的，每时每刻超大量的数据更新，也依旧能够保持体验流畅。

* 1. 后端Node.js
     1. 异步、事件驱动

异步处理事件，Node对于并发的请求，采用异步处理，可以维持不间断地接收，并在后台继续处理事件（I/O）；等待事件结束，再进行回调（Callback）。相对于传统的Java程序以及PHP程序，Node.js支持更高的并发量，理论上持续有用户发送请求，都能够进行响应，但由于是单线程，事件的累加，依然会导致响应时间的不断延长。

* + 1. 非阻塞I/O

Node.js遇到一个请求事件I/O（Input/Output），会创建一个线程来处理，主线程会继续执行；因此在遇到一个需要从多个数据源请求数据的业务，Node的多个拉取动作，会并发执行，在多源头的情况下，能够节省大量时间。

相对于传统的PHP方案，多个请求的串联执行，一个数据拉取完成再拉取下一个，事件处理时间大大延长，影响用户体验。

* + 1. 分布式处理

由于Node.js是单线程方案，因此中间一个环节代码崩溃，就将导致应用退出，需要重启。在系统重启期间，将不能给用户提供服务。

解决分方案是采用pm2 和 nginx工具，检测服务器的状态，在服务器出错时，将用户请求转发到正常运行的服务器上面，并尝试重启出错服务。此为Node.js分布式构建方案的其中之一。

Node.js分布式的另外一个方案，便是使用cluster模块。它可以开启多个工作线程，利用多核服务器创建多个进程，充分利用每个核心，在使用这种方案的时候，需要处理好多个进程之间的通信问题。

本系统采用的是pm2+Nginx，根据服务器的核心数，生成相应的工作线程，同时利用Nginx来处理静态的资源以及反向代理、负载均衡。

* 1. 桌面应用框架Electron
     1. 跨平台的桌面应用框架

桌面软件的开发，随着互联网技术的发展，不再是局限在C++、JAVA等旧有开发语言。跨平台的应用开发，没有必要再通过不同开发语言去处理。如今，有了一个跨平台的桌面应用开发框架——Electron，将网页标签作为应用UI，用迷你的浏览器打开一个web应用。

为什么说这是一个跨平台的软件开发方案？因为HTML+CSS的Web应用的发展，人们只需要打开一个网址，就能使用软件；而现在，Electron，相当于提供了一个可以兼容不同平台的浏览器，那我们完成一个WebApp的时候，就能够得到一个兼容不同平台的应用。

* + 1. 兼容性说明

传统的前端开发人员，需要应对不同浏览器，不同系统；不同的环境下，对于网站的CSS样式和JavaScript脚本的支持不尽相同；导致用户体验受到影响。开发人员往往需要花费大量的精力处理兼容性问题。

而使用Electron进行开发，只需面对Chrome谷歌浏览器；这大大减少了开发人员的负担，他们可以将更多的精力集中在业务的具体表现上。

* + 1. 调用系统的API

Electron提供了大量调用操作系统功能的API，能够方便地调用剪贴板、系统对话框、HTML5 File原生API操作文件等等。使用Electron开发桌面应用和一个网站开发的过程相当接近，前端开发通过Electron，摇身一变成为了桌面应用开发；使得前端开发的业务场景开拓到了桌面应用上面。

* + 1. IPC通讯模块

IPC（Inter-Process Communication，进程间通信）是Electron中传递信息的重要模块。

每一个Electron桌面应用对应一个主程序，主程序对应一个主窗口，通过这个主窗口，Electron桌面应用实现与原生应用窗口进行交互。当主程序调用BrowserWindow 模块创建新窗口的时候，新窗口会产生对应的渲染进程；渲染进程与主进程的共同合作，就能实现整个应用的页面渲染和交互。

然而，主进程和渲染进程之间并不能直接进行通讯，需要通过IPC进行数据传递。IPC的功能类比Vue中的vuex，应用场景主要在进程之间的数据传递；而vuex是Vue的状态管理模块，一般定义需要全局统一的应用状态例如登录态、用户信息等。也有初学者用它来充当组件之间的信息桥梁。

1. 概要设计
   1. 总体介绍
      1. 需求规定

表4-1 高校教师功能需求表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **功能** | **输入** | **输出** | **备注** |
| 1 | 登录 | 用户姓名/工号  （cookie） | 用户ID | POST: XXX\_0000\_D  （拼音首字母\_工号\_default默认用户） |
| 2 | 查看模板 | model\_list | Array | GET: json格式 |
| 3 | 上传表格 | Json | Success/failed | POST: 前端处理成规定格式 |
| 4 | 上传附件 | File，msgId | Success/failed | POST |
| 5 | 查看已上传 | msg\_list | Array | GET: json格式 |
| 6 | 修改信息 | \_msgId，Object | Object | POST: 输出修改后的对象 |

表4-2 信息收集人员功能需求表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **功能** | **输入** | **输出** | **备注** |
| 1 | 登录 | 用户姓名/工号 | 用户ID | POST: XXX\_0000\_C  （拼音首字母\_工号\_collect数据采集用户） |
| 2 | 生成模板 | String | xlsx文件  下载URL | POST: 上传到服务器 |
| 3 | 发布模板信息 | Object | Success/failed | POST: 发布后可通过普通用户查看 |
| 4 | 查看已提交的  教师信息 | msg\_list | Array | GET: json格式 |
| 5 | 导出信息 | Json | File | 前端处理 |

* + 1. 运行环境

操作系统：Windows、Mac

软件：Chrome

* + 1. 基本设计概念和处理流程

高校教师数据采集与汇总系统由模板管理、数据管理、文件系统三大系统组成。其中模板管理系统主要负责数据收集模板的生成、发布、删除功能；数据管理系统主要负责统筹用户上传/修改数据、收集人员查询/筛选数据、导出数据等功能；文件系统主要负责记录数据模板、管理证明附件。

图4-1 系统结构

* 1. 接口设计
     1. 高校教师接口

系统为教师用户提供了完备的接口，有登录接口、模板查询接口、数据上传接口、附件上传接口、已上传数据查询接口、修改信息接口：

1. 登录接口：
   1. 方法：POST
   2. 参数：realName(String), workID(String)
   3. 返回：XXX\_0000\_D
2. 模板查询接口：
   1. 方法：GET
   2. 参数：\_id(XXX\_0000\_D)
   3. 返回：Array: [

{\_mId: ‘’, title: ‘’, version: ‘’, create\_time: ‘’},

…

]

1. 数据上传接口：
   1. 方法：POST
   2. 参数：Json: {

user: ‘XXX\_0000\_D’,

msg\_list: [

{\_msgId: ‘’, title:’’, lastTime:’’, custom\_1: ‘’, custom\_2: ‘’, …},

…

]

}

* 1. 返回：String(success/failed)

1. 附件上传接口：
   1. 方法：POST
   2. 参数：{

user: ‘XXX\_0000\_D’,

\_msgId: ‘’,

url: ‘upload/ XXX\_0000\_D/\_msgId/xxxx.xxx’

}

* 1. 返回：String(success/failed)

1. 已上传数据查询接口：
   1. 方法：GET
   2. 参数：\_id(XXX\_0000\_D), \_msgId
   3. 返回：Array: [

{\_msgId: ‘’, title:’’, lastTime:’’, custom\_1: ‘’, custom\_2: ‘’, …},

…

]

1. 修改信息接口：
   1. 方法：POST
   2. 参数：Json: {

user: ‘’,

\_msgId: ‘’,

msg\_object: {\_msgId: ‘’, title:’’, lastTime:’’, custom\_1: ‘’, custom\_2: ‘’, …}

}

* 1. 返回：Object:{\_msgId: ‘’, title:’’, lastTime:’’, custom\_1: ‘’, custom\_2: ‘’, …}
     1. 信息收集员接口

系统则为信息采集汇总的人员提供了登录接口、模板生成接口、发布模板信息接口、查看已上传数据列表接口：

1. 登录接口：
   1. 方法：POST
   2. 参数：realName(String), workID(String)
   3. 返回：XXX\_0000\_C
2. 模板生成接口：
   1. 方法：POST
   2. 参数：String(字段1, 字段2, 字段3, …)
   3. 返回：String(download/model/\_mId.xlsx)
3. 发布模板信息接口：
   1. 方法：POST
   2. 参数：Object: {\_mId:’’, Name:’’, Url:’’}
   3. 返回：String(success/failed)
4. 查看已上传数据列表接口：
   1. 方法：GET
   2. 参数：\_mId
   3. 返回：Array: [

{user:’’, \_msgId: ‘’, title:’’, lastTime:’’, custom\_1: ‘’, custom\_2: ‘’, …},

…

]

* 1. 本章小结

概要设计是软件系统开发的基础，在概要设计中需要展现软件总体结构，确定系统的组成部分。本章通过图4-1 系统结构图，清晰地展示了高效教师数据采集与汇总系统的三个子系统：模板管理系统、数据管理系统以及文件系统。通过表4-1 高校教师功能需求表和表4-2信息收集人员功能需求表，展示了系统的两类主要用户主体对于系统的功能需求。在接口设计中，分别列出了两类主体的各个结构设计，例如：有登录接口、模板查询接口、数据上传接口、附件上传接口、已上传数据查询接口、修改信息接口、模板生成接口、发布模板信息接口、查看已上传数据列表接口。

1. 服务器设计
   1. 服务器概要

高校教师数据采集与汇总系统使用了Node.js的Koa2框架，快速搭建一个稳定高效的Web服务器。

* + 1. 服务器目录设计

Node.js中Web服务器设计大同小异，本次采用的目录结构如图5-1所示。



图5-1 服务器目录

* + 1. 登录拦截

Koa提供了优秀的登陆拦截中间件，koa-passport，koa-session，能够进行用户登录验证，配合拦截器，可以阻止用户在登录前进行请求操作，代码如下。

//定义允许直接访问的url

const allowpage = ['/login','/api/login']

//拦截

function localFilter(ctx) {

let url = ctx.originalUrl

if (allowpage.indexOf(url) > -1) {

logger.info('当前地址可直接访问')

}else {if (ctx.isAuthenticated()) {

if(url==='/'){

ctx.redirect('/projectList')

}

console.log('login status validate success')

} else {

console.log('login status validate fail')

console.log(ctx.request.url)

ctx.redirect('/login')

}

}

}

//session拦截

app.use(async (ctx, next) => {

localFilter(ctx)

await next()

})

* + 1. 数据库连接

MongoDB数据库的链接建立较为便捷，在服务器中，使用了mongoose进行数据库连接，代码如下。

const mongoose = require("mongoose");

function connect(){

let db = mongoose.connect("mongodb://localhost:27017/blog");

db

.then(()=>{console.log("连接数据库成功")},)

.catch((err)=>{console.log(`连接数据库失败\n${err}`)});

};

* + 1. 日志输出

Koa2脚手架中，配备了koa-logger中间件，方便管理人员进行服务器监视，代码如下。

const logger = require('koa-logger')

// logger

app.use(async (ctx, next) => {

const start = new Date()

await next()

const ms = new Date() - start

console.log(`${ctx.method} ${ctx.url} - ${ms}ms`)

})

* 1. 登录
     1. 功能描述

登录是用户使用本系统的基本前提，在完成登录之前用户无法实现各种数据操作。

* + 1. Schema对象

{

\_id: {type: Number},

realName:{type: String}

}

* + 1. 路由

高校教师POST: /api/login/default

数据收集人员POST: /api/login/collcet

* 1. 模板管理
     1. 功能描述

用户需要使用模板时，需要在模板列表页面请求模板列表。而创建模板需要在后台界面，由数据收集员进行创建。

* + 1. Schema对象

{

\_id: {type: Number},

title: {type: String},

edit\_date: {type: String},

keys: {type: Array}

}

* + 1. 路由

查询列表GET: api/model/get\_list

生成模板POST: api/model/create

* 1. 数据管理
     1. 功能描述

用户进行数据上传，数据修改，数据读取。

* + 1. Schema对象

{

\_id: {type: Number},

title: {type: String},

edit\_date: {type: String},

\_mId: {type: Number},

data: {type: String}

}

* + 1. 路由

上传POST: api/data/update

查询GET: api/data/get\_list

修改POST: api/data/modify

* 1. 文件上传
     1. 功能描述

用户进行文件上传。

* + 1. Schema对象

{

\_id: {type: Number},

title: {type: String},

edit\_date: {type: String},

\_msgId: {type: Number},

url: {type: String}

}

* + 1. 路由

上传POST: api/file/upload

1. 数据库设计
   1. 概念设计

需求分析阶段一般需要将所确认需求转换为实体-关系模型，我将高校教师数据采集与汇总系统的应用需求，抽象成为实体与实体之间的关系，形成了如图6-1的“实体关系图”：

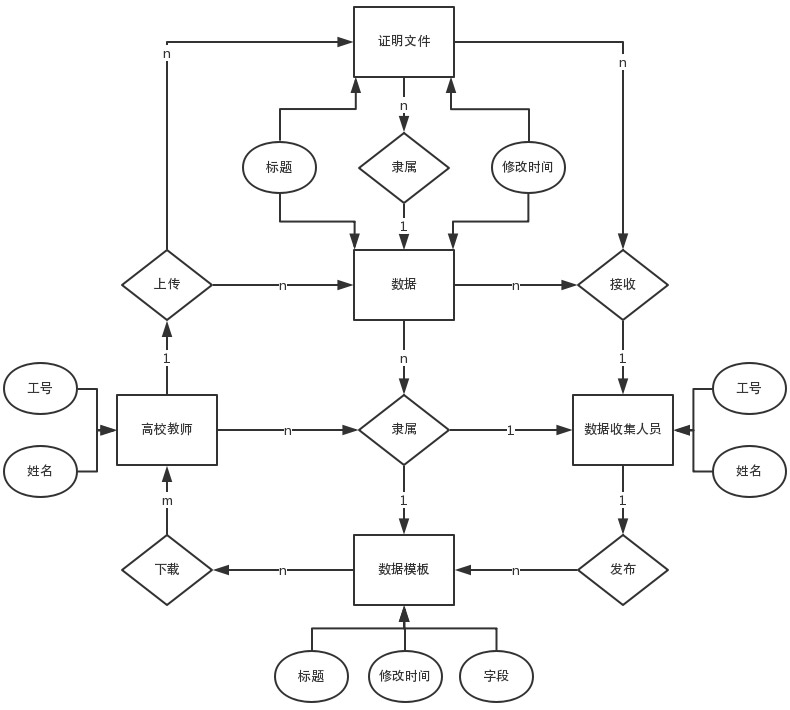


图6-1 高校教师数据采集与汇总系统E-R图

* 1. 逻辑结构设计

数据库的逻辑结构设计需要将概念设计阶段所完成的“实体-关系图”转换为符合数据库数据模型的逻辑结构。本系统采用的是MongoDB数据库，虽然是非关系型的数据库，这依赖于它强大的自由度；但是在实际使用中，通过Node.js中的mongoose依赖组件，能够轻易地将MongoDB自由的使用方式转换为我们熟悉的关系型数据库使用方式。

也因此，我们能够将图6-1的E-R（Entity-relationship）图，转换为如下形式：

高校教师（\_id、工号、姓名、账号类型）；

数据收集人员（\_id、工号、姓名、账号类型）；

数据模板（\_id、标题、修改时间、关键字）；

数据（\_id、标题、修改时间、依赖模板、数据）；

证明文件（\_id、标题、修改时间、依赖数据记录、存储路径）。

* 1. 物理结构设计

数据库的物理结构设计，在使用关系型数据库的时候，一般指存取方法和存取结构。在高校教师数据采集与汇总系统中，采用MongoDB，使用方法符合关系型数据库。

本系统采用索引存取方法，对每位教师用户建立唯一索引形如XXX\_0000\_D，给每一个数据模板建立唯一索引\_mId形如model\_0，给每一条数据记录建立组合索引形如（XXX\_0000\_D，model\_0，msgId\_0）。

1. 系统测试
   1. 测试方案
      1. 测试平台

Windows10，Chrome

* + 1. 测试内容

1. 测试页面间跳转是否正常
2. 页面数据获取
3. 文件数据读取是否准确、上传是否异常
4. 数据导出是否正常
   1. 运行结果

表7-1 测试运行结果表

|  |  |  |
| --- | --- | --- |
|  | 测试内容 | 测试结果 |
| 1 | 登录 | 正常 |
| 2 | xlsx/xls读取 | 正常 |
| 3 | 数据上传 | 正常 |
| 4 | 文件上传 | 正常 |
| 5 | 模板生成 | 正常 |
| 6 | 模板下载 | 正常 |
| 7 | 数据排序 | 正常 |
| 8 | 数据导出 | 正常 |

* 1. 应用展示
     1. 客户端首页

普通教师用户进入页面，会出现在消息列表页面，如图7-1所示。



图7-1 消息列表

* + 1. 模板下载列表

通过首页的导航按钮，进入模板下载页面，如图7-2所示。



图7-2 模板下载列表

* + 1. 数据上传

数据上传页面用户可以上传xlsx文件或者xls文件，数据会直接读取到Javascript脚本，通过点击“上传”按钮进行上传，如图7-3所示。



图7-3 数据上传

* + 1. 文件上传

文件上传页面与数据上传同属一类，功能上确需要区分，不同的文件需与相对应的信息绑定，以便管理，通过列表选项可绑定到相应的记录，如图7-4所示。



图7-4 文件上传

* + 1. 已上传数据列表

已上传数据列表是用户查看自己所上传的数据的窗口，在这里能够看到自己上传的数据，如图7-5所示。



图7-5 已上传数据列表

* + 1. 后台首页

后台是数据采集汇总人员的工作台，工作流程是发布数据收集模板后，检查收集情况，最后验收导出。首先看到的便是收集到的数据列表，如图7-6所示。

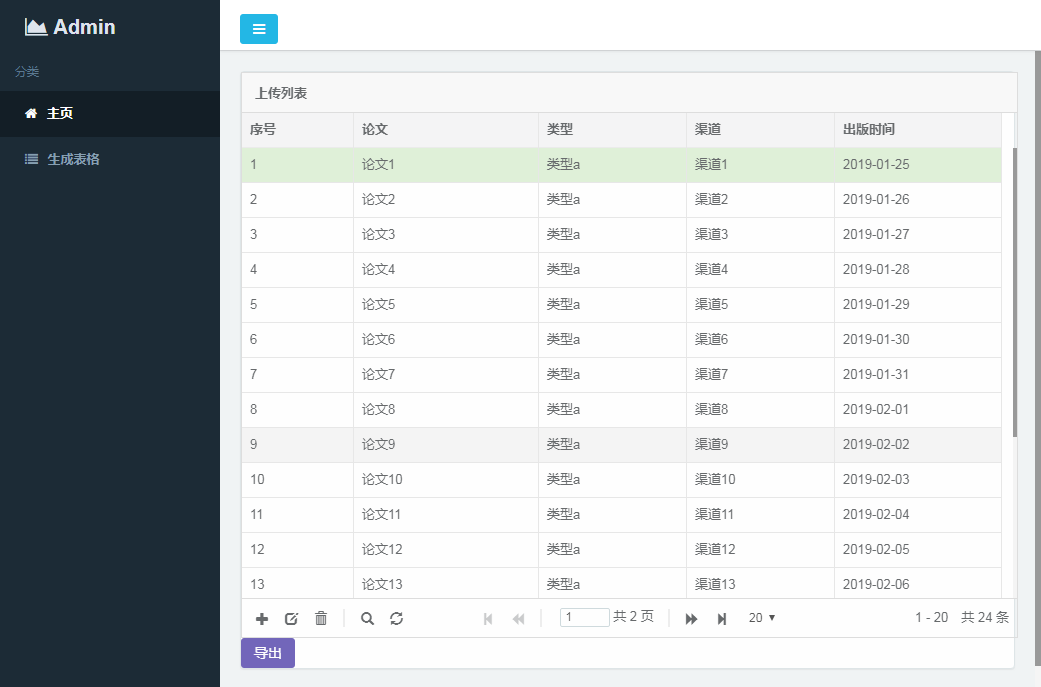


图7-6 数据收集列表

* + 1. 模板生成

模板生成页面通过几个字段，工作人员能够简单地生成一个数据收集模板，如图7-7所示。

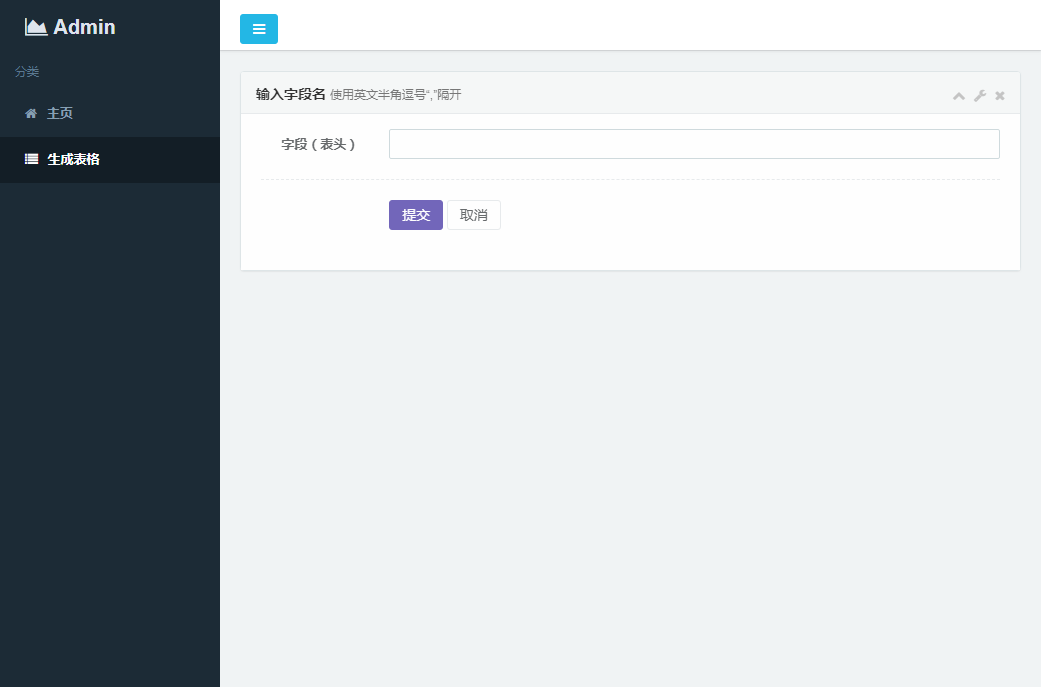


图7-7 模板生成页面

参考文献

[1]林朝炽.基于Elastic Stack的分布式数据采集与分析平台探索[J].网络安全技术与应用,2019(02):38+61.

[2]周永辉. 基于网络的短信收发系统的设计与实现[D].郑州大学,2017.

[3]杨诚. 基于Android的心理咨询服务系统的设计与实现[D].电子科技大学,2013.

[4]张志恒,张显库,杨光平,马超,冯永孝.基于Visual C++的Excel工作簿数据处理[J].软件导刊,2017,16(01):135-137.

[5]高小茜. 高校业务数据管理系统的研究与实现[D].西安科技大学,2016.

[6]罗文,刘星,兰全祥.基于Node.js的创新团队管理系统的设计与实现[J].赤峰学院学报(自然科学版),2018,34(10):83-85.

[7]陈健.广东高校图书馆数字电子期刊资源分布与利用研究[J].图书馆学刊,2017,39(02):39-44.

[8]胡佳静. 基于electron的待办事项管理app开发[D].华中科技大学,2018.

[9] 不止前端. electron入门心得[Z]. https://www.cnblogs.com/buzhiqianduan/p/7620099.html,2017

[10] 秋雨仲夏. 记录Python模拟登录爬取Mysise学生管理系统的信息[Z]. https://www.jianshu.com/p/3576b0d4c457, 2017.

[11]罗贤春,余波,姚明.信息价值研究的热点领域可视化分析[J].现代情报,2015,35(11):120-126.

[12]于文洋.针对办公自动化系统的信息安全问题的探讨[J].办公自动化,2018,23(24):44-45.

[13] muffinfish.Node.js：浅析高并发与分布式集群[Z].https://segmentfault.com/a/1190000015841624，2018.

致 谢

我历时将近两个月时间终于把这篇论文写完了，在这段充满奋斗的历程中，带给我的学生生涯无限的激情和收获。在论文的写作过程中遇到了无数的困难和障碍，都在同学和老师的帮助下度过了。在校图书馆查找资料的时候，图书馆的老师给我提供了很多方面的支持与帮助，尤其要强烈感谢我的论文指导老师——袁冠远老师和刘生建老师对我进行了不厌其烦的指导和帮助，无私的为我进行论文的修改和改进，就没有我这篇论文的最终完成。在此，我向指导和帮助过我的老师们表示最衷心的感谢！

同时，我也要感谢本论文所引用的各位学者的专著，如果没有这些学者的研究成果的启发和帮助，我将无法完成本篇论文的最终写作。至此，我也要感谢我的朋友和同学，他们在我写论文的过程中给予我了很多有用的素材，也在论文的排版和撰写过程中提供热情的帮助！金无足赤，人无完人。由于我的学术水平有限，所写论文难免有不足之处，恳请各位老师和同学批评和指正！