基于Web设计实习报告

BS00580_

**课程设计题目： 教材定购系统**

**桂林理工大学 信息科学与工程学院**

**班 级： 软件三班**

**报告人姓名： 朱晓锋**

**学 号： 3172052051626**

**承担角色 ： 组长**

**同组组长 ： 朱晓锋**

**同组其他成员： 顾进原**

**实验指导教师： 邓酩**

**实验地点： 02524**

**完成起止日期： 2020/1/8**

目录

[基于Web设计实习报告 1](#_Toc29378628)

[一、系统设计概述 3](#_Toc29378629)

[1.1目的 3](#_Toc29378630)

[1.2 任务 3](#_Toc29378631)

[1.2.1要完成的功能： 3](#_Toc29378632)

[1.2.2 小组实习进度安排 3](#_Toc29378633)

[1.2.3 小组分工说明 4](#_Toc29378634)

[1.2.4 小组完成情况 5](#_Toc29378635)

[1.3 开发环境 5](#_Toc29378636)

[1.4参考资料 5](#_Toc29378637)

[二、系统需求分析 6](#_Toc29378638)

[三、系统设计的基本原理和采用的主要方法与技术 6](#_Toc29378639)

[基本原理： 6](#_Toc29378640)

[主要方法与技术： 6](#_Toc29378641)

[Spring 6](#_Toc29378642)

[Mybatis 7](#_Toc29378643)

[vue.js 7](#_Toc29378644)

[iviewUI 8](#_Toc29378645)

[Nginx 9](#_Toc29378646)

[JSON 9](#_Toc29378647)

[四、本人在此次实习负责的系统模块设计与实现。 9](#_Toc29378648)

[1需求分析 9](#_Toc29378649)

[2、页面参考 同类网站参考 http://www.gxjcy.cn 9](#_Toc29378650)

[3、 详细设计 11](#_Toc29378651)

[4、编码原理 11](#_Toc29378652)

[1项目结构图： 11](#_Toc29378653)

[2程序流程图： 13](#_Toc29378654)

[3数据结构： 13](#_Toc29378655)

[4核心代码 15](#_Toc29378656)

[五、个人遇到的困难与获得的主要成果 16](#_Toc29378657)

[遇到的困难有： 16](#_Toc29378658)

[主要成果： 17](#_Toc29378659)

[个人经验获得与总结： 19](#_Toc29378660)

[六、测试与运行记录 19](#_Toc29378661)

[6.1 测试 19](#_Toc29378662)

[前端自行测试: 19](#_Toc29378663)

[前后端对接测试： 21](#_Toc29378664)

[6.2运行记录： 22](#_Toc29378665)

[七、系统设计完成结果分析与个人小结 22](#_Toc29378666)

[完成结果分析： 23](#_Toc29378667)

[缺点： 23](#_Toc29378668)

[小结： 23](#_Toc29378669)

[八、*附录* 23](#_Toc29378670)

[软件配置： 23](#_Toc29378671)

[个人完成的程序模块： 24](#_Toc29378672)

[文档清单： 24](#_Toc29378673)

0

# 一、系统设计概述

## 1.1目的

本系统基于信息技术建立的高校教材订购系统。

## 1.2 任务

### 1.2.1要完成的功能：

教材订购系统，应当从学生、老师、采购员、审核员等不同用户角度展开分析,

分析并实现以下功能：

首先由教师或学生提交购书单,经教材发行人员审核是有效购书单后,开发票,登记并返给教师或学生领书单,教师或学生可以到书库领书.

采购系统的主要工作过程为：若是教材脱销,则登记缺书,发缺书单给书库采购人员;一旦新书入库后,即发进书通知给教材发行人员.

### 1.2.2 小组实习进度安排

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 日程内容 | 时间期限 | 应完成的内容 |
| 2019/12/22 | 完成题目的选择 | 一天 | 交付题目，选出组长 |
| 2019/12/23 | 确定分工 | 一天内 | 确定文档的撰写，整个系统的开发方向 |
| 2019/12/24 | 开始讨论需求分析，编码工具的选择，以及框架的选择，并开始学习，期间相互交流以免产生偏差过大。概要设计 应该包含数据的设计，以及合理性评估。 | 三天内 | 完成框架的初步学习，确定了整个教材订购系统的框架和模样，完成 需求分析和概要述设计， |
| 2019/12/27 | 开始负责撰写详细设计文档，以及开发接口文档撰写完成文档的同时搭建开发环境。 | 三天内 | 应当熟悉分工所需要做的功能。  清楚明白，前后端交互所需要的数据格式的作用。  文档应当清楚详细的描述过程。和代名词的含义。  组员应当简单了解到框架的运行原理。并成功运行简单的例子。 |
| 2019/12/31 | 开始编码，完成各模块的功能 | 六天内完成 | 应该完成相应的模块编码，并撰写相应文档。功能能正常运行。自动测试。 |
| 2020/1/6 | 开始测试，进行前后端对接 | 两天内完成 | 编写测试文档，保存测试数据，修改相应代码使得项目够运行，和健壮性评估。完善代码结构和逻辑。 |
| 2020/1/8 | 完成个人实习报告的撰写，等待提交检查 | 至检查 | 准备好文档和项目，调试好，等待检查 |

### 1.2.3 小组分工说明

**组长负责：**

全面协调小组工作；负责整个项目开发工作,网站需求分析,系统测试与调试和小组文档核查等工作

1、开题(问题定义与可行性研究)

2、用户类型与业务流程图分析图

3、网站原型需求分析小组报告

4、负责美工,网页设计与实现等工作

5、著名同类网站考察,分析小组报告

6、网站文字与素材,网页设计文档

7、个人课程设计报告

8、小组课程设计最终报告大纲

9、个人课程设计报告

**组员：**

**组员1：**

1数据库设计小组报告(含 E­R 图)、

2、系统总体测试与调试小组报告

3、网站后台服务模块集成测试

4、数据库设计与实现

5、后端业务层的逻辑的控制层的逻辑实现

### 1.2.4 小组完成情况

⑴ 问题理解和分析。

2019/12/24 组员完成了相应的任务，并提交了相应文档。共同撰写了需求文档

⑵ 确定解决问题的方法（技术）。

2019/12/25 考虑到组员的能力，确定了采用前后端分离的方法进行开发。使用流行的框架，文档资料较多的开发软件IDEA。

⑶ 详细设计和编码（设计）。

2019/12/30 共同制定和撰写了详细设计，数据库的设计，以及约定了接口文档的参数和形式。

2020/1/5 完成了编写代码。各功能模块。

⑷调试和运行（调试）

2020/1/7 共同进行测试，完成了前后端对接的任务。成功地运行了项目。

## 1.3 开发环境

经过小组讨论，决定采取前后端分离的形式开发，前端使用vue+iviewUI,后端使用SpringBoot+Mybatis。

**开发平台：Window10**

**开发工具**：后端：IDEA 前端： WebStorm

**服务器**：Nginx、tomcat

**原型软件：ROSE、亿图、Navicat**

## 1.4参考资料

《软件工程案例与实践》第三版 韩万江

《软件工程》作 者 ：宋开旭

百度搜索：[www.baidu.com](http://www.baidu.com)

csdn博客：[www.csdn.com](http://www.csdn.com)

中国知网：<https://kns.cnki.net>

vue框架文档：<https://cn.vuejs.org>

iviewUi框架文档：<https://iviewui.com>

Mybatis框架文档：<https://mybatis.org/mybatis-3/zh/index.html>

Spring：<https://spring.io>

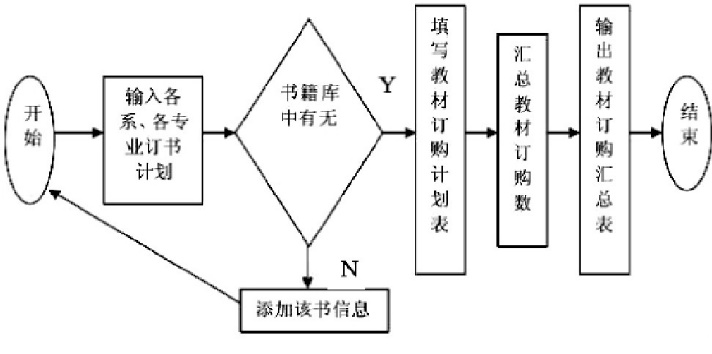
# 二、系统需求分析

详见附录文档

# 三、系统设计的基本原理和采用的主要方法与技术

基于web的MVC模式开发

教材订购系统可实现的系统功能主要为学校教务处及各院系教材管理员提供管理使用, 也就是对各院系教材订购计划进行审核管理, 查询现有教材信息数据, 审核统计不同院系教材订购情况, 设置到书信息管理;实现对教材的征订、统计;联系教材出版社及供书商, 对教材费用进行统计。教材订购系统对应输入的信息主要为:不同院系教材计划, 包括书籍名称、ISBI、出版社、出版时间、价格、订购数量、班级等。教材订购系统对应输出的信息则为:不同院系教材订购记录。教材订购管理处理流程, 如图1所示。



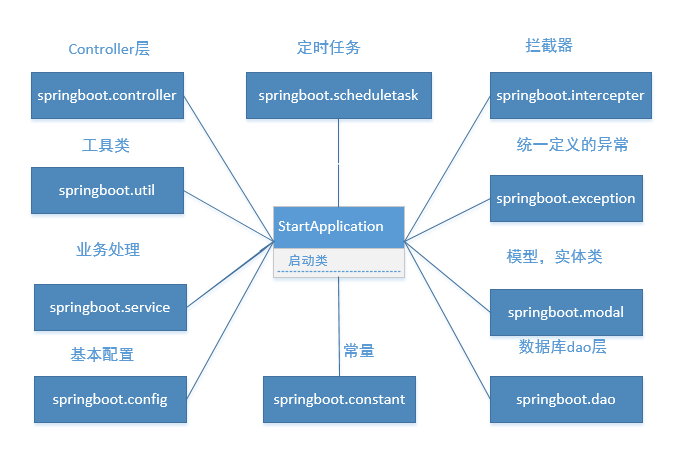
## 基本原理：

## 主要方法与技术：

### Spring

Spring框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，Spring框架具有面向切面编程（AOP）框架.

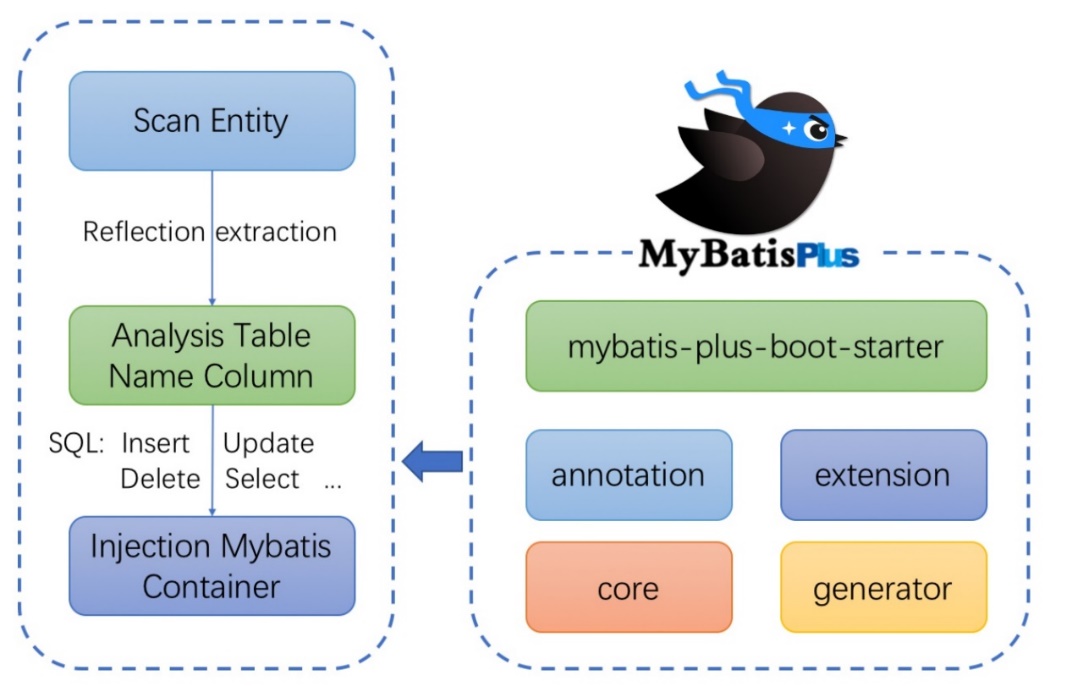
技术结构如图所示：



### Mybatis

MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生类型、接口和 Java 的 POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

结构图如下：



### vue.js

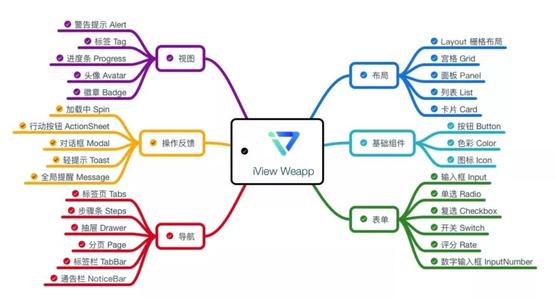
Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。

**结构图如下：**



### iviewUI

一套基于 Vue.js 的高质量UI 组件库



### Nginx

Nginx是一款轻量级的Web服务器，也是一款轻量级的反向代理服务器， 高稳定、高性能、资源占用少、功能丰富、模块结构化、支持热部署

**本系统主要使用Nginx的反向代理解决跨域问题：**

Nginx 反向代理模块 proxy\_passproxy\_pass 后面跟着一个 URL，用来将请求反向代理到 URL 参数指定的服务器上，就可以完成反向代理。

server {

listen 80;

location /api {

proxy\_pass https://api.shanbay.com;

}

}

### JSON

JSON 是用于存储和传输数据的格式。JSON 通常用于服务端向网页传递数据 。

**本系统使用JSON作为前后端的数据交互类型。并作为API接口的数据形式交换**

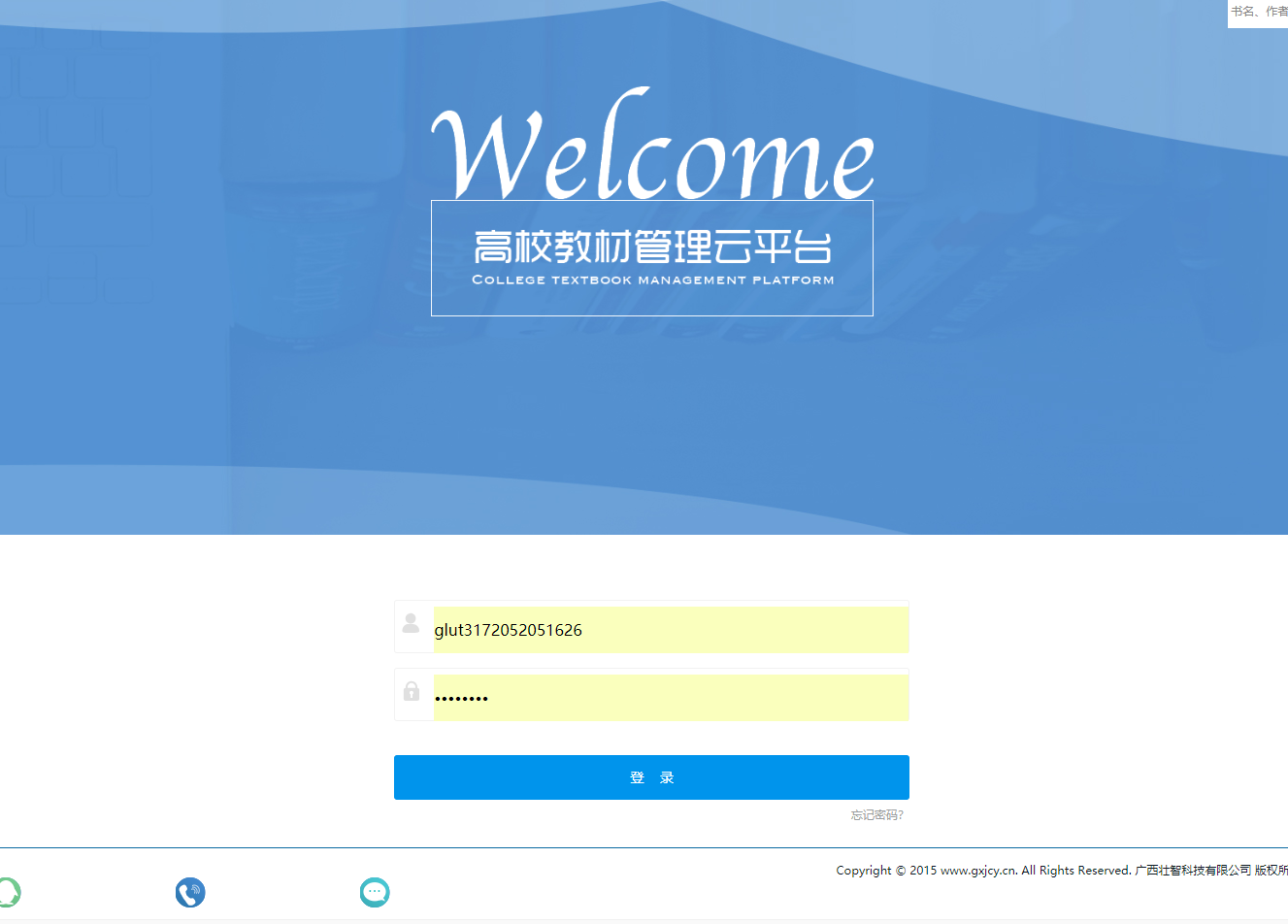
# 四、本人在此次实习负责的系统模块设计与实现。

## 1需求分析

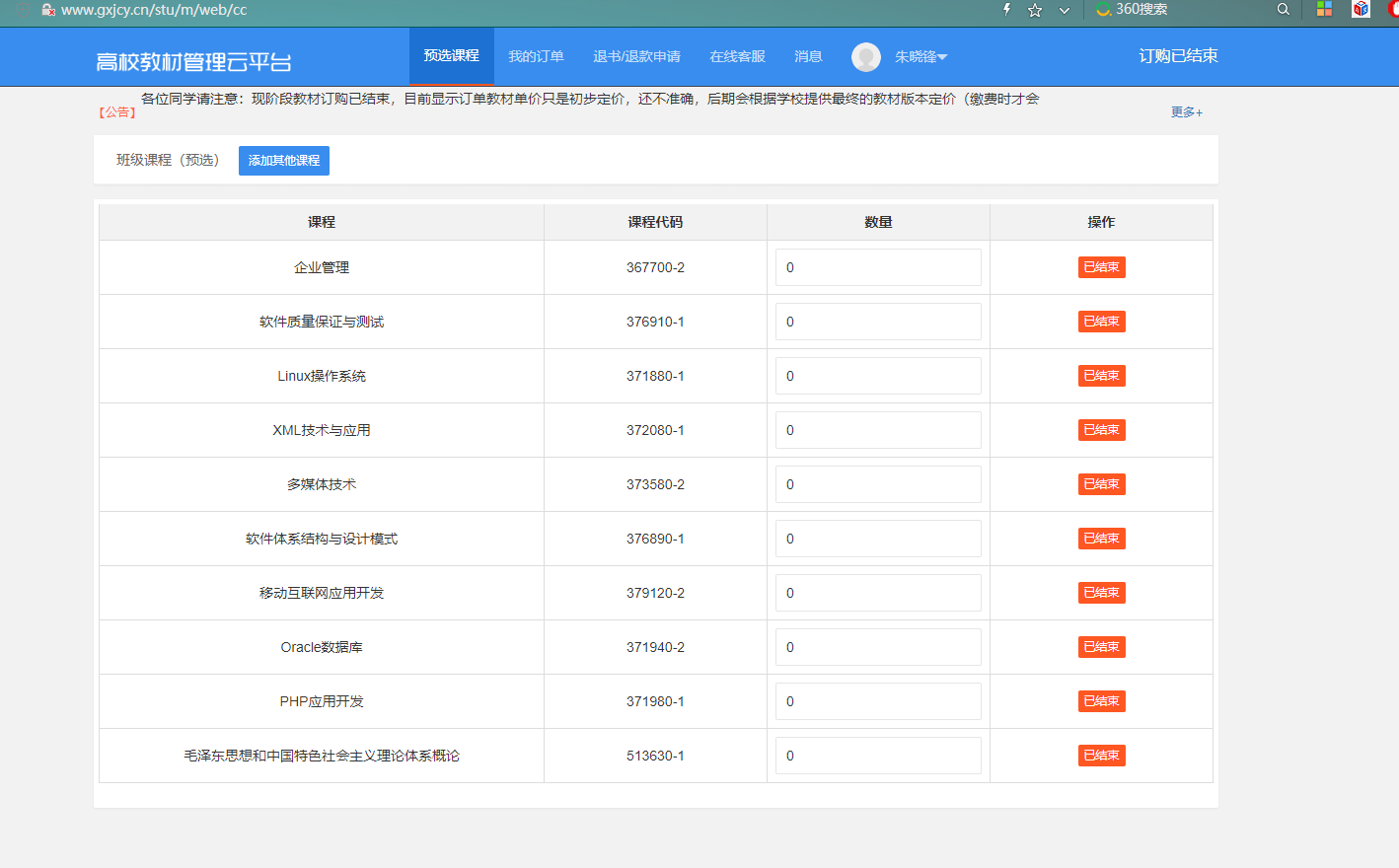
美观大方，简洁、实用性高的网站

## 2、页面参考 同类网站参考 <http://www.gxjcy.cn>

登录页面参考



订购系统参考



历史订单参考：



## 详细设计

参照详细设计

## 4、编码原理

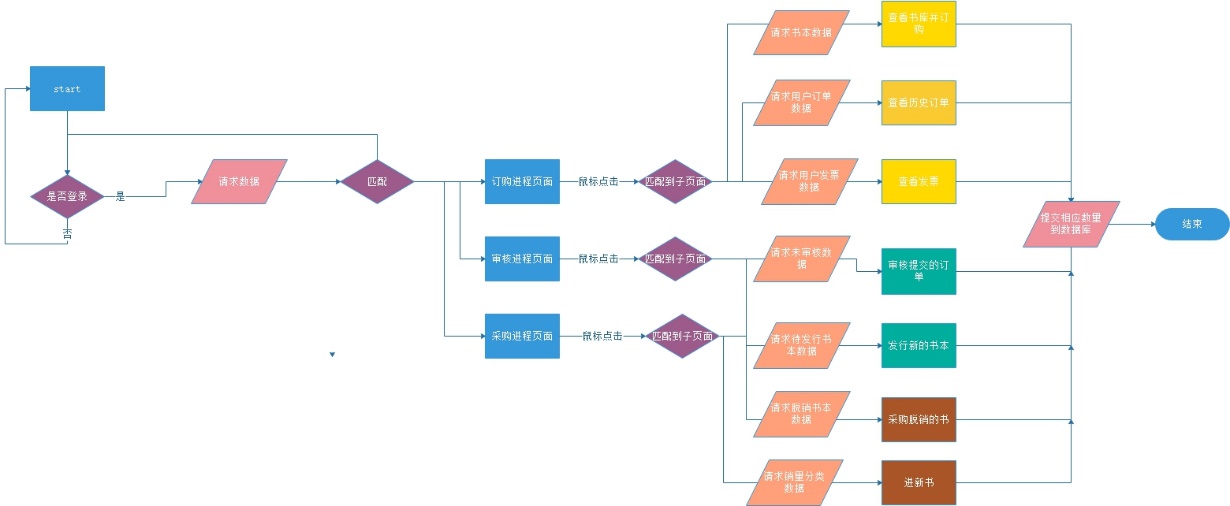
采用vue 的数据驱动型，来更新页面

编码遵循着结构清晰，职责单一，面向对象的原则进行开发。

### 1项目结构图：



### 2程序流程图：



### 3数据结构：

与后端交互的数据格式：



前端数据结构：

动态绑定数据html数据：



所有的数据做为属性，并和所有方法组成对象给组件引用。

只修改对应对象的数据和操纵方法即可完成业务逻辑。

如：



其中export 导出这个对象，因为JS中对象都是引用会造成污染。使用export可以导出新的对象。data 为该对象的属性，即数据，method为该对象的方。

通过方法来更新数据达到页面更新的目的。vue会检测数据的变化自动渲染。避免我们直接用js操作DOM元素带来的不可避免的错误和麻烦。

### 4核心代码

登录：

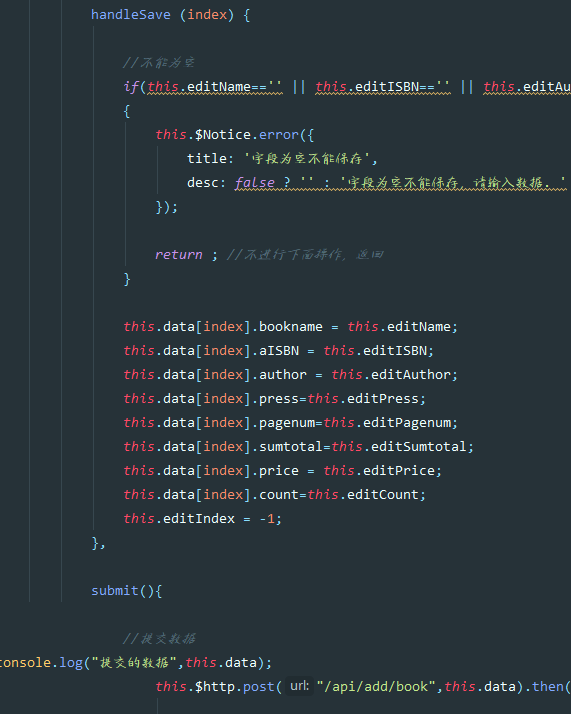
说明：向后端请求该用户的信息。请求成功就存储一下该用户的信息。

*this*.$http.post('/api/login',dataLogin).then((res)=> {  
 console.log("loginpost",res);  
  
 *//存储一下用户数据：  
 // window.localStorage.setItem('token', token)// 这个长期保存* sessionStorage.setItem("username", *this*.account);*//临时会话保存* sessionStorage.setItem("role",res.data);*//保存角色代码*

注册同理。

订购模块：

说明：由于代码过长，已折叠代码显示



*this*.$http.get('/api/audit/submit/textbookstatus?userId='+row.userId+'&orderNum='+row.orderNum).then((res)=>{  
 *if*( res.data=="600"){  
 *this*.$Notice.success({  
 title: '该订单审核已通过',  
 desc: *false* ? '' : res.data  
 });  
 *//删除当前行  
 // this.data9.splice(index, 1);* } *else* {  
 *this*.$Notice.error({  
 title: '提交通过审核过程发生了某些错误',  
 desc: *false* ? '' : res.data }); }  
}).catch((err)=> {  
 console.log(err);  
 *this*.$Notice.error({  
 title: '通过发生了某些错误',  
 desc: *false* ? '' : '发生了某些错误'  
 });  
});

# 五、个人遇到的困难与获得的主要成果

## 遇到的困难有：

|  |  |  |
| --- | --- | --- |
| 编号 | 时间 | 详情 |
| 001 | 2019/12/24 | 首先是选择题目的问题是，小组有不同的看法。难以决定。一种想做有新意一点的，一种想选择简单点的来做。意见难统一 |
| 002 | 2019/12/27 | 刚开始学习框架，困难程度太大。几乎没学过。环境都搭不起来。花费了许多时间。后来知道学习一个框架应该先去了解这个框架的运行流程，核心的设计理念。 |
| 003 | 2019/12/29 | 遇到bug，修了一整天。Eslint语法检查太严格，由于是新手，关掉花了诸多时间。配置文件也是件困难的事。 |
| 004 | 2020/1/2 | 由于设计不合理，导致不得不重构，花了大量时间。不符合编码规范。 |
| 005 | 2020/1/3 | 由于实习在实验室的时间少，小组间的交流只能通过QQ等通信工具。因此只能抽出时间，在图书馆交流，但又不能大声讨论，只能寻找没有人的教室（然后都有人上课）。是件麻烦的事情。因此会导致后面对接的时候出现，期望与事实相差太大的问题，而导致对接不上，不得不改代码的现状 |
| 006 | 2020/1/5 | 开即前后端对接，发行，自己以前写的代码因为没有写逻辑，而导致了看不懂的情况，需要花许时间来调试一确定。  发现一个文件代码超过200行，逻辑就会变得复杂难懂。  在进行前后端对接的时候，发现，可能因为在初始阶段没有商量好的原因，导致，后端传过来的数据不并与我前端对应。因此不得修改逻辑代码。重构导致大量bug。  因为前后端对接 ，一开始很难说是谁的错，如果不能协调好，相互修改代码以确定是谁的错，是个问题 |
| 007 | 2020/1/7 | 采购页面提交后，表单数据表格某些行发生了空白的现象，排查了好久都没有发现问题。到下午，才百度Google 发现，这是由于框架的深浅拷贝和变量作用域的问题，引用过多，又对其进行修改的话，会造成不可预见的结果 ，我用Object.assgin 进行深拷贝就解决了，js 变量 尽量用 let ,var是全局，let 是局部有效。 |

## 主要成果：

|  |  |  |
| --- | --- | --- |
| 编号 | 时间 | 详情 |
| 001 | 2019/12/26 | 确定了开发的方向和工具选择，这过程认识到各种工具的特点，为以后的实践选择不再选择困难症。快捷选出工具。 |
| 002 | 2019/12/28 | 完成了系统需求分析，系统概要设计，系统的详细设计开始和接口开发文档的编写完成50%，学习了UML ROSE的面向对象建模方法。学习亿图软件的快速画图（试用版）。以及了解到Navicat的操作数据库方便性。 |
| 003 | 2019/12/30 | 完成所有的编码前的文档，并对框架进行了一段时间的学习，完成的环境的搭建。知道了前端也分层的概念，MVVM模式。使得，业务逻辑与页面设计 进一步分离。不会直接操纵DOM元素带来的麻烦 |
| 004 | 2019/12/31 | 完成登录注册功能，并用测试数据完成了初步的测试。  知道了登录注册的安全性问题，在商业中，使用Token 令牌来验证用户。  使用 seesionID,来防止用户输入地址栏直接进来。  使用 双钥密码技术与MD5，消息杂凑函数，以及加上，时间戳来防止中间人攻击，避免明文传输。 |
| 005 | 2020/1/2 | 完成了订购系统的页面开发，并同步测试。  使用组件式开发，使得，页面切换流畅，不需要进行页面的跳转，用户体验非常好。  学到了懒加载和分页技术。 |
| 006 | 2020/1/3 | 完成了审核系统的页面开发，并进行了使用测试数据测试 |
| 007 | 2020/1/4 | 完成了采购的系统的页面开发，用mock.js数据进行测试，了解到了mock,js 这个强大模拟数据工具的用法。 |
| 008 | 2020/1/5 | 优化了代码，删除无用的代码，了解职责单一，不与陌生人交流的原则 在实战中的用法。 |
| 009 | 2020/1/6 | 增加一些图标和美化了页面，更改了背景图片，调整页面元素的样式。使之看起来有了很大的改善。显得更简洁和美观。了解到CSS的一些语法，感觉很难，美化对于做后端的人来说是这样的 |
| 010 | 2020/1/7 | 开始前后端对接，并开始测试和同步数据 |
| 011 | 2020/1/8 | 完成测试，开始撰写测试文档和个人实习报告  了解到了测试文档的写法，黑白盒的测试。 |
| 012 | 2020/1/9 | 完成实习，等待检查 |

### 个人经验获得与总结：

通过参加小组团队的开发实践，为毕业后适应团队合作开发模式打下基础，了解项目管理、团队合作、文档编写、口头与书面表达的重要性。

并在课程设计实践中，提高了自学能力，书面与口头表达能力，创造能力和与团队其他成员交往和协作开发软件的能力，提高今后参与开发稍大规模实际软件项目和探索未知领域的能力和自信心。

作为组长应该对整个系统有一定的把握和了解。能及时了解到组员的进度和完成情况。

能够及时纠正因理解不到位而引发的开发方向偏离的问题。

在系统出现问题的时候，可以及时的和组员反应并处理。

经过这次的学习。了解到了前端开发的方向。分解，粘合度降低，的重要性，MVC开发的原理，和实践学习。vue 框架的响应式原理用来适应不同的屏幕如手机，电脑屏幕尺寸出现的问题，而改变 大小，数据双向绑定，页面可以动态刷新。Ajax 并不是一门新技术，而是各种技术的简写简称，即各种技术的组合，使得，在不刷新页面的情况下，可以局部更新，而vue 正是大量使用了这种技术 和watch 监听器技术。像在用Ajax 与后端交互，前端可以接受刷新数据。本项目开发，采用前后端，也正是采用了这种技术。使用用户体验上升（不用刷新页面等待。特别是在提交订单的时候，或者加载数据库的信息过于庞大的时候，vue 的懒加载）。

这是我对这个vue 框架的这次实习的理解。至于iview 是个UI框架，主要提供界面，所以并没有深入了解，只有出现bug的时候，调试进了源码里。

# 六、测试与运行记录

## 6.1 测试

### 前端自行测试:

Mock.js 模拟数据：

测试数据样例：

**样例一**：

*//提交书单的返回码  
const* data4={  
 "code":"@integer(400, 401)",  
};  
Mock.mock('/api/buy/submit/textbook',data4);

**说明**：位于项目的订购模块的购买教材部分。

**含义说明**： code :状态码：，@integer(400, 401) 是随机返回400 到401 之间的整数。

Mock.mock函数指 对应的api,应该返回的数据 data4;

**样例二**：

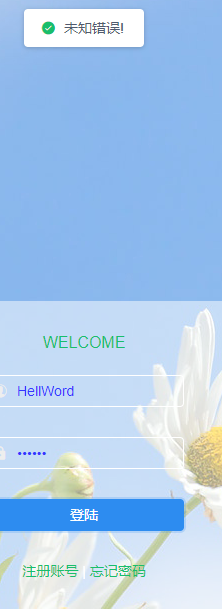
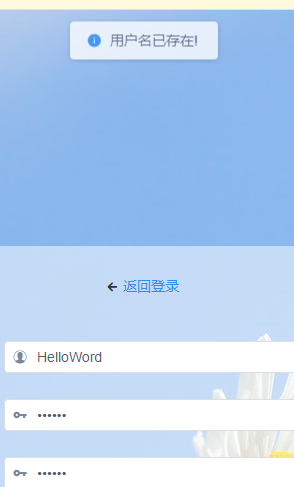
*//请求书单  
const* data3= [{  
 "name":"@ctitle(3, 8)",  
 "ISBN":"@guid",  
 "id|1-5": "@string",  
 "price":"@float(10, 200, 3, 5)",  
 "count":"@integer(10, 203)",  
  
 }, //因为代码过长。此处n本书的内容。

];  
  
Mock.mock('/api/buy/textbook', 'post', data3);  
*// Mock.setup({  
// timeout: '60-100' // 表示响应时间介于 200 和 600 毫秒之间，默认值是'10-100'。全局？  
// })*

**说明**：位于项目的订购系统的查询书库信息

**含义说明**：依次是书名，ISBN，id, 价格，剩余数据，mock 为api ，post方式提交，返回数据为data3;

**测试效果样例**：

随机返回状态码。

横拟的数据提交效果：



**依次测试完成整个项目。确认提交的结果符合接口文档。**

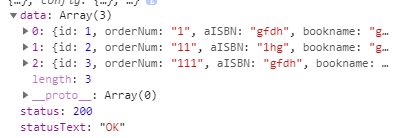
### 前后端对接测试：

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 编号 | 测试步骤 | 预测结果 |
| 登录 | 001-1 | 登录框输入正确的用户名:student及其密码:student | 进入项目网站首页面 |
| 001-2 | 登录框输入错误的用户名:student及其密码:std | 登录失败，提示重新登录 |
| 注册 | 002-1 | 登录输入时回车刷新主页面 | 页面正常刷新 |
| 002-2 | 通过快速注册注册用户std，密码std，以此用户登录 | 进入SPM项目网站首页面 |
| 002-3 | 注册时确认密码:123456和密码:123写不一致 | 系统阻止此次注册 |
| 采购页面 | 003-1 | 分别以学生、教师、管理员身份点击登录注册 | 进入相应页面 |
| 003-2 | 以采购员点击确认选课 | 进入相应页面 |
|  | 点击保存，随意填写内容 | 成功显示试题 |
| 订购页面 | 004-1 | 以教师身份点击实践指导书 | 页面显示界面相关正文,并能够订购、查看历史、订单、查看发票 |
| 审核页面 | 005-1 | 以审核员的身份登录 | 显示相应页面，能查看审核的订单 |

## 6.2运行记录：

浏览F12 查看记录：

样例一：



说明：这是查询书库返回的信息数据。

样例二：



说明：Nginx服务器产生的访问日志记录样本，没有显示异常。api 为向后端访问的记录

# 七、系统设计完成结果分析与个人小结

## 完成结果分析：

在2020/1/8 进入了整个项目尾声，教材订购系统成功地部署和运行。订购系统、审核系统、采购系统均能正常运行。

### 缺点：

健壮性有待提高，比如：可以直接通过地址栏进入没有相应权限进入的页面。

虽然有开发环境下有一定的方便快捷性调试，但在生产环境中不需要。虽然有相应的技术，但由于时间能力有限的情况下，就暂时不实现。

用户体验有待优化，界面不友好，只能按照选定的路线进行操作，如果出现非法操作，系统可能出现无法预估的状况，bug.

## 小结：

在经过小组成员的三周27天的努力下，完成了本次实习，在经过这次的实习，深刻体会到了小组团队协作能力的重要性。总的说，系统虽然有待完善空间很大，但学习到了这些前端框架：vue +iview 还是收获良多。了解主流系统开发的流程和经验。

# 八、*附录*

### 软件配置：

**前端**：Nginx 1.16.1

conf配置文件：



注：proxy\_pass 指向后端服务器的IP 和端口。

dist 里的文件放到nginx 的html根目录下。

访问Nginx服务器

**后端**：IDEA 2019.1版 maven 3.0 版本 SpringBoot 2.3 JDK 1.8

### 个人完成的程序模块：

前端的各模块页面文件：

登录注册模块login.vue register.vue

订购模块：bookForm.vue order.vue historyorder.vue

审核模块：reviewer.vue extend.vue

采购模块：procurement.vue chart.vue

后端的是项目的各模块的业务逻辑，数据库设计.详见组员的个人实习报告。

### 文档清单：

|  |  |  |
| --- | --- | --- |
| 文件名 |  | 说明 |
| 需求分析文档.docx |  | 项目第一阶段产生的文档，描述了系统项目的发展方向 |
| 概要设计文档.docx |  | 项目第二阶段产生的文档，描述了整个项目大致的框架体系 |
| 详细设计文档.docx |  | 项目第三阶段产生的文档，进一步描述了项目的模块的内容，使得整个系统清晰起来 |
| 测试文档.docx |  | 项目第四阶段产生的文档，测试系统所产生的数据，记录，报告 |
| 开发接口文档.docx |  | 该文档描述了前后端对接所应遵循的规范和数据交换规则 |
| 数据库设计文档.docx |  | 文档描述了数据库整体设计理念的模块， |

注：文档的一些图表，需要使用Office 2016 以上版本才能打开。