

论文检测全文报告

基本信息

报告编号: 2022022271535698B3C382BB1

文档名称: 1901050108程芳杰19计应(创新卓越)班
提交方式: 上传文档检测
提交时间: 2022年02月22日
正文字符数: 9137
检测范围: 大雅全文库

过滤操作: 已过滤参考文献

总体结论

文献相似度: 21.01%
去除参考文献相似度: 21.01%
去除本人已发表论文相似度: 21.01%
重复字符数: 1920
过滤前的相似度: 22.26%

文献原创度: 78.99%
单篇最大相似度: 3.53%
单篇最大重复数: 336
最相似文献题名: 毕业论文工作总结范文集锦六篇
过滤前的重复字数: 1920

相似片段分布



典型相似文献

相似图书

序号	题名	作者	出处	相似度
1	互联网+智慧城市 核心技术及行业应用	强磊;勾善;林明;曲哲;王红安	北京: 人民邮电出版社, 2018.10	2.04%
2	HTML5移动Web+Vue.js应用开发实战	吕鸣	北京: 清华大学出版社, 2020.05	1.76%
3	HTML5 实战宝典	山西优逸客科技有限公司	北京: 机械工业出版社, 2017.02	1.76%
4	JavaScript实战 JavaScript、jQuery、HTML5、Node.js实例大全	张泽娜	北京: 清华大学出版社, 2014.08	1.65%
5	云应用系统开发	邱锦明	北京: 人民邮电出版社, 2019.03	1.63%
6	JavaScript实战 JavaScript、jQuery、HTML5、Node.js实例大全	张泽娜	北京: 清华大学出版社, 2018.06	1.35%
7	大数据架构商业之路 从业务需求到技术方案	黄申	北京: 机械工业出版社, 2016.05	1.31%
8	云计算软件工程 云软件自动生成原理及方法	周相兵	北京: 科学出版社, 2014.11	1.31%
9	前端工程师必备技能 Vue移动开发实战技巧	李利德;徐辛承	北京: 电子工业出版社, 2018.01	1.23%
10	Node.js实战	赵坤;寸志;雷宗民	北京: 机械工业出版社, 2014.05	1.18%
11	React+Redux前端开发实战	徐顺发	北京: 机械工业出版社, 2019.07	1%



12	Vue.js入门与商城开发实战	黄菊华	北京: 机械工业出版社, 2020.09	1%
13	中国林业物联网 思路设计与实践探索	李世东	北京: 中国林业出版社, 2017.12	0.99%
14	信息类专业复杂工程问题实践教程	缪新颖;李恩松;于红	北京: 清华大学出版社, 2020.07	0.88%
15	全栈开发实战宝典	优逸客科技有限公司	北京: 机械工业出版社, 2019.01	0.86%
16	云计算原理与实践	过敏意;吴晨涛;李超;阮娜	北京: 机械工业出版社, 2017.10	0.78%
17	龙芯应用开发标准教程	靳国杰;张戈;胡伟武	北京: 人民邮电出版社, 2018.12	0.78%
18	自学区块链 原理、技术及应用	范凌杰	北京: 机械工业出版社, 2019.05	0.7%
19	Web前端开发技术	吴志祥;雷鸿;李林;肖建芳;黄金刚	武汉: 华中科技大学出版社, 2019.03	0.7%
20	New Internet 大数据挖掘	谭磊	北京: 电子工业出版社, 2013.03	0.69%
21	大数据系统安全技术实践	尚涛;刘建伟	北京: 电子工业出版社, 2020.01	0.69%
22	信息系统项目管理师教程第3版	全国计算机专业技术资格考试办公室组编,谭志彬,柳纯录,周立新,卢光明	北京: 清华大学出版社, 2005.03	0.69%
23	程序员面试笔试宝典 第3版	何昊;叶向阳;窦浩	北京: 机械工业出版社, 2018.06	0.68%
24	Vue.js项目开发实战	张帆	北京: 机械工业出版社, 2018.08	0.67%
25	能源大数据技术与应用	任庚坡;楼振飞	上海: 上海科学技术出版社, 2018.06	0.62%
26	Node.js Web开发实战	李翠霞;严晓龙;王勇	北京: 人民邮电出版社, 2019.12	0.62%
27	软件工程案例教程 第2版	魏雪峰;葛文庚;王春华;刘会超;张银玲;耿红琴	北京: 电子工业出版社, 2018.08	0.57%
28	软件设计师软考辅导 3年真题精解与闯关密卷	郭春柱	北京: 机械工业出版社, 2014.04	0.56%
29	2013软件设计师软考辅导-3年真题精解与实战模拟	郭春柱	北京: 机械工业出版社, 2013.06	0.56%
30	Vue.js从入门到项目实战	刘汉伟	北京: 清华大学出版社, 2019	0.55%
31	JS全书	高鹏	北京: 清华大学出版社, 2019.12	0.55%
32	JavaScript设计模式与开发实践	曾探	北京: 人民邮电出版社, 2015.05	0.55%
33	微店这样玩就赚了! 开店运营+营销推广+成功案例 第2版	徐茜	北京: 人民邮电出版社, 2017.09	0.52%
34	React与Redux开发实例精解	刘一奇	北京: 电子工业出版社, 2016.11	0.51%
35	信息组织	马建锋;魏强	北京: 国防工业出版社, 2019.11	0.51%
36	JavaScript设计模式	张容铭	北京: 人民邮电出版社, 2015.08	0.39%
37	物联网云平台设计与开发	林勇;农国才;郭炳宇;姜善永	北京: 人民邮电出版社, 2019.02	0.38%
38	Node.js开发实战教程	钟小平	北京: 人民邮电出版社, 2020.07	0.35%
39	Ionic3与CodePush初探 支持跨平台与热更新的App开发技术	陈杰浩;张成;吴曦;史继筠	北京: 北京理工大学出版社, 2018.04	0.34%
40	Vue.js从入门到实战	孙鑫	北京: 中国水利水电出版社, 2020.03	0.3%
41	Neo4j权威指南	张帜;庞国明;胡佳辉;苏亮;赵炳	清华大学出版社, 2017.09	0.27%
42	Javascript程序设计教程	李林;施伟伟	北京: 人民邮电出版社, 2008.05	0.26%
43	ActionScript 3.0设计模式	WilliamSanders;Cb andimaCumaranat unge	北京: 中国电力出版社, 2009.09	0.17%

相似期刊

序号	题名	作者	出处	相似度
----	----	----	----	-----



1	基于Node.js的高性能站群系统设计与实现	童敏;张文盛;钱立三	软件, 2018, 第8期	1.9%
2	基于react-native的移动端周界入侵定位系统	张芳杰;刘太君;尚爱民;叶焱	无线通信技术, 2019, 第3期	1.45%
3	基于NGRAIN技术的民族文化遗产桌面三维VR系统的研究与实现	陈建强	贵州民族大学学报(哲学社会科学版), 2017, 第6期	1.3%
4	基于MongoDB的物联网开放平台用户信息管理	杨洁;赵启飞	现代电子技术, 2018, 第2期	1.03%
5	基于Web的色质数据解析平台设计与开发	孔波;邹有;卢红兵;杨华武;虞苏行	计算机技术与发展, 2019, 第12期	1%
6	Web页面开发模式演变历史研究*	刘敏娜;黄素萍;李延香	数字技术与应用, 2019, 第10期	0.99%
7	基于WebSocket与WebGL的城市三维协同规划	徐瑞	测绘地理信息, 2019, 第6期	0.96%
8	构建分布式负载均衡集群的企业邮件网络平台	伊雯雯	计算机安全, 2011, 第6期	0.7%
9	基于Vue.js和Javalin框架的学生宿舍社交平台WebAPP应用程序的设计与开发	黄浩铭;刘成珏;郑滢;王璐越;彭泰	软件工程与应用, 2019, 第6期	0.63%
10	现代Web 技术在MES 系统中的应用	郭政健;杨军	新一代信息技术, 2018, 第5期	0.62%
11	基于Web系统的大数据搜索技术的实现和优化	慕慧;徐晓慧	计算技术与自动化, 2021, 第1期	0.55%
12	基于Echarts的智能车间数据可视化系统	郑嘉怡;陈涛;齐世玲;朱东	电脑知识与技术, 2021, 第25期	0.55%
13	基于MongoDB的蛋白质组学大数据存储系统的设计	张琳;谭军;白明泽	计算机应用, 2016, 第A1期	0.52%
14	基于Vue.js的在线设计开放平台研究与实现	王璐;崔保磊;潘红霞;赵莉;田宇	信息技术与信息化, 2019, 第11期	0.47%
15	基于移动平台的微应用开发框架技术的研究与应用*	蔡璟;丁宗银	通信技术, 2018, 第5期	0.31%
16	运载火箭参数数据库开发与应用	徐利杰	航天工业管理, 2018, 第4期	0.3%
17	BIM+GIS集成可视化性能优化技术	陈明娥;崔海福;黄颖;郭明强;庄灿	地理信息世界, 2020, 第5期	0.27%
18	"天地图·陕西"新冠肺炎疫情动态地图设计与实现	王乃生;王凯;申传庆	测绘技术装备, 2020, 第3期	0.26%
19	基于卡尔曼滤波的牲畜动态称重系统的设计	李琦;夏云龙	黑龙江畜牧兽医, 2017, 第23期	0.25%
20	渐进式JavaScript框架Vue.js的全家桶应用	吕英华	电子技术与软件工程, 2019, 第22期	0.24%
21	校园智能导游系统的设计与实现	黄秋凤;廖明惠;李孟欢;杨丽;高茂兰	电子技术与软件工程, 2021, 第16期	0.24%
22	基于DevOps能力模型的持续集成方法	董昕;郭勇;王杰	计算机工程与设计, 2018, 第7期	0.24%
23	基于图像纹理识别的澳门汽车牌照定位提取	李雪涵	信息技术与信息化, 2020, 第11期	0.19%
24	基于SPA技术的网络随访系统的设计与实现	陈洞天;任宇飞	中华医学图书情报杂志, 2019, 第7期	0.17%
25	利用设计模式进行Email系统的设计	郑琪;方思行	济南大学学报(自然科学与医学版), 2003, 第3期	0.17%

相似网络文档

序号	题名	作者	相似度
1	毕业论文工作总结范文集锦六篇		3.53%
2	移动运营商能力开放平台的设计与实现	王伟	1.4%
3	基于微前端的旅行社业务支撑系统的设计与实现	杨彩芳	0.99%



4	组件化自助微商城的设计与实现	杨雪	0.87%
5	去哪儿网酒店供应链系统的设计及实现	姚光艳	0.86%
6	苗圃智能化信息管理系统研究	原阳晨	0.84%
7	农村宅基地管理信息系统的设计与实现	王金明	0.67%
8	基于React的前端广告组件库的设计与实现	裴春艳	0.63%
9	智能家居服务系统的设计与实现	刘历	0.61%
10	基于RESTful Web服务的配电设备状态监测系统研究与开发	李俊杰	0.58%
11	面向在线教育的电话销售系统设计与实现	陶林涛	0.58%
12	基于携程旅行APP特卖汇项目的前端优化设计与实现	郭慧敏	0.57%
13	基于图像分割算法的标注系统的设计与实现	安郝敏	0.55%
14	高职院校数字化校园工程规划及相关技术研究	李鹏	0.48%
15	基于MongoDB的旅游垂直搜索系统的设计与实现	费华辉	0.43%
16	基于深度学习的三维机房设备信息采集系统的研究与实现	张雅茹	0.42%
17	安全监测监控联网平台模块设计	李志强	0.42%
18	移动授权与验证系统的设计与实现	陈鹏辉	0.35%
19	基于Vue.js的表单可视化构建系统的设计与实现	牛仁腾	0.34%
20	基于用户角色的数据包过滤系统的设计与实现	成锐	0.3%
21	面向循环内燃烧反馈控制的双核集成式发动机控制器开发	高国景	0.3%
22	反映学生需求的评教指标系统研究	段媛媛	0.29%
23	基于ZigBee和超声波的室内定位系统	程正瑞	0.28%
24	SDN安全渗透测试系统的设计与实现	李建武	0.27%
25	基于SOA的门户框架的研究与实现	王前	0.26%
26	基于FIDO架构在线指纹识别系统客户端的设计与实现	王耀龙	0.25%
27	广东大学生三下乡实践报告		0.25%
28	面向专用协议的无线传感器网络测试系统的设计与实现	闻婷婷	0.23%
29	杭州市下城区数字监控综合管理平台的研究与实现	丁家麒	0.23%
30	PXIe总线板卡的设计与实现	丁泽峰	0.22%
31	基于图像匹配的景区虚拟留言板系统	常海峰	0.22%
32	HCV基因分型及基于基因型的抗原表位预测研究	万祥辉	0.2%
33	渣土车管理系统的研究与实现	张春艳	0.19%
34	机器人视觉框架及视觉原型系统的设计与实现	胡心滢	0.17%
35	基于氧化海藻酸钠载药微球的制备与研究	王秋艳	0.15%

全文对比

摘要

随着互联网技术的发展，传统的多页面开发方式早已成为过去式，而基于单页面开发技术已经占据主流，单页面是指只有一个主页面的应用，浏览器一开始要加载所有必须的 html, js, css。所有的页面内容都包含在这个所谓的主页面中。但在写的时候，还是会分开写（页面片段），然后在交互的时候由路由程序动态载入。相比之下，传统的多页面应用每个页面（只说动态页面）都是使用服务器端模板编写，然后请求这个页面的时候由服务器渲染成 html 再返回。两者对比，一个很明显的区别就是，多页面应用的 server 端要干两件事：提供数据+渲染，而单页面应用把渲染拿到浏览器端做了，服务器只提供数据就可以了。

在webpack统治的期间，基于vue-cli脚手架工具开发的Vue2.x项目更是成为了H5和移动开发的大头。webpack就是前端资源模块化管理和打包工具，它可以很多松散的模块按照依赖和规则打包成符合生产环境部署的前端资源，还可以将按需加载的模块进行代码分割，等到实际需要的时候再异步加载，而要它自动实现这些功能，你得提前编辑好配置文件。vue-cli是脚手架工具，其作用就是用配置好的模板迅速搭建起一个项

目工程来，省去自己配置webpack配置文件的基本内容。

在2016年10月Vue2.x正式发布，vue是一个渐进式的JS框架。他易用，灵活，高效；可以把一个页面分隔成多个组件；当其他页面有类似功能时，直接让封装的组件进行复用；他是构建用户界面的声明式框架，只关心图层；不关心具体是如何实现的

vue cli是一个基于Vue.js进行快速开发的完整系统，Vue CLI致力于将Vue生态中的工具基础标准化，它确保了各种构建工具能够基于智能的默认配置即可平稳衔接。

我将用最新的Vue2.x + Vue CLI开发该系统，得到前所未有的开发体验，运行速度，打造一个小型的外卖平台。

关键词：外卖；Vue2；Vue CLI；Node；MongoDB；

目录

摘要.....	1
第1章.....引言.....	1
1.1选题背景与意义.....	1
1.2研究内容.....	1
1.3相关技术介绍.....	1
1.4研究内容与主要工作.....	6
1.5论文结构与章节安排.....	6
第2章.....需求分析.....	8
2.1.....用户与基础行为分析.....	8
2.2.....功能需求分析.....	8
2.3.....环境需求分析.....	10
2.4.....本章小结.....	12
第3章.....设计.....	13
3.1.....平台架构总体设计.....	13
3.2.....前后端交互设计.....	14
3.3.....前端设计.....	15
3.4.....功能模块业务逻辑设计.....	16
3.5.....数据库设计.....	17
3.6.....本章小结.....	19
第4章.....总结与展望.....	20
4.1.....工作总结.....	20
致谢.....	21



参考文献.....22

引言

选题背景与意义

随着网络的发展，外卖已经成为大多数人不得不使用的一个程序，因为许多用户喜欢它，因为不必下楼去跑腿或排队，在手机上订单并直接送货上门。对于用户来说不用下楼，不用排队，手机点菜快，方便快捷；用一只手控制周围的餐馆，选择更广泛的美食；通过参与会员积分和全额折扣来订餐更具成本效益。更多的销售、评估、位置距离等筛选，更放心。对于老板来说移动订购已经成为一种趋势，建立自己的订购和交付平台，以扩大在线流量；通过平台，可以轻松举办各种营销活动，如优惠待遇、全额优惠、折扣、限时优惠、团体竞争等。，从而大大增加了回头客的比例。新产品信息、特殊事件等。可以发送到用户的手机上进行准确的营销，而过去只能发送传单。对于企业家来说可以组建自己的分销团队，赚取分销成本。

研究内容

用户可以在线点餐，用户可以筛选符合自己要求的商家店铺以及配送地址的增删改，订单查询，点餐后可以进行评价等。

相关技术介绍

本节通过介绍Vue2、Vue-Cli、mint-ui、JavaScript、Axios、Node、Express、MongoDB、开发工具以及开发环境，介绍了整个平台开发所涉及的相关技术。

Vue2简介

随着ES2015标准的发布，Javascript（正式称为ECMAScript，简称ES）得到了重大改进，主流浏览器终于开始为这些新添加的特性提供适当的支持。其中一些特性特别地为我们提供了极大提升Vue能力的机会。

Vue.js目前最火的一个前端框架，**三大主流前端框架之一**。Vue.js是一套构建用户界面的框架（一套完整的解决方案，对项目侵入性大，中途需要更换框架则需要重构整个项目），只关注视图层，易上手，有配套的第三方类库。提高开发效率，帮助减少不必要的dom操作；双向数据绑定，通过框架提供的指令，**前端只需要关注业务逻辑，不再关心dom如何渲染**。前端的MVVM前端视图层的概念，主要把每个页面分成了M（Model）、V（View）、VM（VM ViewModel）。VM是其中核心，M和V间的调度者。**M，保存的是每个页面中单独的数据（比如要渲染页面表格，ajax请求到后台的你个数组，此数据即为M）**。V，每个页面的html结构。VM，一个调度者，分割了M和V，M和V不直接关联，通过中间的VM。V想要保存数据到M，都要有VM做中间处理；V想要渲染页面，需要调用VM，VM从M中取数据。前端中使用MVVM思想，主要让开发更方便，MVVM提供了数据的双向绑定（由VM提供）。

在Vue2.0中，组件的通信是双向的，这简直是amazing的创造。在Vue2.0中，父组件的数据通过props流向子组件，这个传值过程中，Vue提供了props数据类型验证。而子组件的数据可以通过\$emit()传递给父组件，这样你就可以通过事件的机制进行子向父的数据传递。个人感觉，这个构思和实现方式，是基于浏览器的事件响应机制。

除了双向数据流之外，组件缓存也绝对能够让你惊奇。缓存简单说就是数据暂缓的存放，缓存的好处就是在数据不变化的情况下，数据需求单位可以从读取更快的缓存拿到数据，而不需要重复请求或者计算获取。这样以下两种场景来说，是不二之选。第一种是你从后端拿到较为row的数据，而这个数据离你实际渲染的格式还需要经过好久的计算和格式化，那想一下有了组件缓存，只要组件的流入数据不变，就不需要经过这个过程。

Vue2优点:

提出渐进式框架的概念 根据官方口径，Vue 2 保留了 Vue 1 90% 的 API，Vue 2 的核心还是数据驱动和组件化，在此基础上提出了“渐进式框架”的概念。

引入虚拟 DOM 移植自snabbdomsnabbdom 的虚拟 DOM 使得初始渲染速度更快，同时使得 Server-Side Rendering — Vue.js 成为可能，支持流式 SSR 渲染，此外，由于 Vue 采用Object.defineProperty 的方式实现响应式，并不需要使用 diff 进行虚拟DOM 的比较后进行更新，效率更高。

提供两种不同构建 runtime-only 构建可用于手写 render 的场景或通过 webpack 打包预先编译的场景，减少代码大小，优化页面加载速度。



添加更为丰富的修饰符 表单添加了 trim 修饰符, 事件添加 once、alt、ctrl、shift、meta、(鼠标键的) left、right、middle (其中前五个是在下贡献的), 使得用户可以用更少的声明式代码来实现这些基础逻辑, 将更多精力放在复杂业务逻辑的开发上。

移除不少可能埋雷的坑 根据 Migration from Vue 1.x — Vue.js 这篇教程, 移除了不少 Vue 1的特性, 在继续保持 API 简洁风格的同时, 避免一些过于“灵活”的写法所带来的隐患。

Vue-Cli简介

Vue CLI是用于搭建Vue.js应用程序的简单工具, 即设置基本的开发环境和项目结构。

Vue-Cli的优点:

开箱即用, 但强制性强, 需要按照它的规则进行

开箱即用, 简单易用

抽离cli service层

插件化

快速原型开发

Mint-ui简介

Mint UI 是饿了么前端团队出品的移动端 UI 组件库, 基于 Vue 2.0 构建, 继桌面 UI 组件库 Element UI 后又一个优秀的开源 UI 组件库, 包含丰富的 CSS 和 JS 组件, 能够快速构建出风格统一的页面, 提升开发效率。

Mint-ui的特性:

UI 设计风格中性, 不需要改样式就能适应很多产品项目

中小项目能大大减少打包体积使用更轻量更通用的时间日期解决方案 Day.js

JavaScript简介

JavaScript诞生于1995年。布兰登·艾奇 (Brendan Eich, 1961年~), 1995年在网景公司, 发明的JavaScript。JavaScript是由网景公司发明, 起初命名为LiveScript; 1995年年12月SUN公司合作, 因市场宣传需要, 改名为 JavaScript。同时期还有其他的网页语言, 比如VBScript、JScript等等, 但是后来都被JavaScript打败了, 所以现在的浏览器中, 只运行一种脚本语言就是JavaScript。JavaScript是世界上用的最多的脚本语言。

Vue-router简介

Vue路由器是Vue.js的官方路由器, 它与Vue.js核心深度集成, 使用Vue轻松构建单页应用程序变得轻而易举。功能包括:

嵌套路线映射

动态路由

模块化, 基于组件的路由器配置

路由参数, 查询, 通配符

查看由Vue.js过渡系统提供动力的过渡效果

细粒度的导航控制

带有自动活动CSS类的链接

HTML5历史记录模式或哈希模式



可自定义的滚动行为

网址的正确编码

Axios简介

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。

Axios的优点：

支持 Promise API

拦截请求和响应

转换请求数据和响应数据

取消请求

自动转换 JSON 数据

Node简介

Node.js 是一个 JavaScript 运行时环境。它又是如何运作的？Node 运行时环境包含执行 JavaScript 程序所需要的一切。如果你了解 Java 的话，会发现它们有点像。JavaScript 原来是只能在浏览器中运行的，当把它扩展成为可以在你的计算机上作为独立的程序运行时，Node.js 就出现了。现在你可以用JavaScript 做更多的事情，而不仅仅是用在网站的互动和特效上。JavaScript 现在能够去做其他脚本语言（如Python）可以执行的操作。你 Chrome 浏览器中的 JavaScript 和 Node.js 都在 V8 引擎上运行。该引擎将你的 JavaScript 代码转换为更快的机器代码。机器代码是低级代码，计算机可以直接运行而无需先解释它。

Express简介

Express 是一个简洁而灵活的 node.js Web应用框架, 提供了一系列强大特性帮助你创建各种 Web 应用, 和丰富的 HTTP 工具。

使用 Express 可以快速地搭建一个完整功能的网站。FRP 简介

MongoDB

MongoDB是一个基于分布式文件存储的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

开发工具以及开发环境

开发工具：Visual Studio Code

Vue版本：2.6.11

Vue-cli版本：4.5.0

Express版本：4.17.1

MongoDB版本：4.0.0

Node版本：v14.17.1

Npm版本：6.14.4

研究内容与主要工作

本文主要在学习和掌握前端开发的相关技术和语言后，进行需求分析提炼出需要实现的功能、后台系统设计、web端的设计、前后端交互的方案、整个前后端的具体实现以及测试，完成了该系统的构建。



论文结构与章节安排

本文分为六章节，每章内容如下：

第一章：引言。第一章介绍课题研究的背景与意义、国内和国外的发展、技术和本文的研究内容与主要工作以及论文和章节安排。

第二章：需求分析。第二章主要对平台的用户与基础行为分析、功能需求分析、环境需求分析、界面需求分析。

第三章：设计。第三章主要对平台架构的总体设计、前后端交互设计、前端界面设计、功能模块业务逻辑设计和数据库设计。

第四章：总结与展望。

需求分析

本章主要采用了图、模型等元素的方式分析和讲述。整体工作内容包括用户分析、功能分析、界面分析等。

用户与基础行为分析

在该外卖平台中分为商家、用户。基于这两个不同角色用户的特性分析出其所需的功能服务，并从其特征中抽象出来投入到平台的设计中。

功能需求分析

主要用户面向大众群体，所以我们分为了两种角色，商家、用户，大致需求如下：

图2-1 需求分析大纲

2.2.1需求分析

表2-1从登录到点餐列表、配送服务、评价系统等方面分析了商家最基本的需求。

用户类型子功能

商家

登录

点餐列表

配送服务

评价系统

商家信息

表2-1 商家需求分析大纲

表2-2从点餐、地址管理、用餐评价、订单管理等方面分析了用户的基本需求。

用户类型子功能

用户

登录

点餐

地址管理

用餐评价



订单管理

表2-2 用户需求分析大纲

2.2.2网站功能描述

表 2-3 是针对整个仿饿了么后台管理端的功能进行功能描述、操作序列和补充说明。

名称1商家登录

功能描述商家通过该功能可以登录后台管理系统

操作序列1.商家进入后台管理登录页面;

2.商家输入特定的账号密码;

3.页面跳转到商家页面;

补充说明特定账号admin 密码246611

名称2统计用户数量

功能描述登录

操作序列1.商家在后台页面进行登录;

2.登陆成功后请求后台将汇总信息显示在页面;

补充说明无

名称3菜单管理

功能描述商家通过该功能可以对菜单列表信息编辑、删除、查询。

操作序列1.商家在后台页面进行登录;

2.登陆成功后切换到用户页面;

3.请求后台将菜单信息显示在页面，可以对单个菜单信息进行操作

补充说明菜单信息操作不可逆，只能再次修改，不能回退

名称4公告管理

功能描述商家通过该功能可以对商家信息编辑、删除。

操作序列1.商家在后台页面进行登录;

2.在商家界面切换到公告管理页面;

3.请求后台将公告信息显示在页面;

补充说明操作不可逆，只能再次修改，不能回退

表2-3 网站需求分析

环境需求分析

对于改系统所需要的环境需求，通过考虑得出下面4点：



(1) 单页Web应用 (single page web application, SPA) : SPA 是一种特殊的 Web 应用, 是加载单个 HTML 页面并在用户与应用程序交互时动态更新该页面的。

(2) 有良好的交互体验能提升页面切换体验, 用户在访问应用页面是不会频繁的去切换浏览页面, 从而避免了页面的重新加载;

(3) 服务器方面, 本系统后台将使用Node的Express, Express 是一个简洁而灵活的 node.js Web应用框架, 提供了一系列强大特性帮助你创建各种 Web 应用, 和丰富的 HTTP 工具。

(4) 数据库方面, MongoDB是一个基于分布式文件存储的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

图2-2 MongoDB流程图

本章小结

本章借助图、模型等元素成功完成用户分析以及用例图的制作, 并在此基础上完成了用户功能分析、功能细化描述和界面分析, 为后文中整个平台设计提供了基础的理论支撑。

设计

本章主要工作内容包括平台架构总体设计、前后端交互设计、前端界面设计、功能模块业务逻辑设计和数据库系统设计, 部分设计描述的时候将采用UML时序图辅助描述。

平台架构总体设计

与以往的MVC三层架构不同, 后台页面使用了MVVM模式, 它本质上就是MVC 的改进版。

* MERGEFORMATINET

图3-1 MVVM模型介绍

前后端交互设计

前后端交互的数据格式统一使用JSON格式, 接口采用Restful风格。

状态码描述

200请求成功

400数据提交有问题

401无权限访问

403访问被拒绝

404访问的资源不存在

405请求方法不合法

413请求过于频繁被限制访问

500服务器错误

502服务端暂时离线

503服务器正在维护

表3-1 HTTP状态码



3.3.1接口请求数据分为成功、失败两种状态:

两种状态是指请求并响应回HTTP状态码200，并且包含一个state属性，该属性描述是否成功。

(1) 如果 state=success 则表示请求成功，并且在data属性中包含响应的参数或提示，例如:

```
{  
  
  "data": {  
  
    "price":190  
  
    "description":"单品定价"  
  
    "icon_name": "特"  
  
  },  
  
  "meta": {  
  
    "msg": "查询成功!",  
  
    "status": 200  
  
  }  
  
}
```

Data为接口请求的内容存放的位置 如果是get请求获取到的所有内容都会存放在这个键下

Meta则是返回状态的一种封装里面包含status以及message

Status为遵守http状态码设计规范设计的一个返回状态码,该码代表了当前请求的接口的响应状态

Message为返回的消息如果接口请求出现了异常则会将对返回消息返回到前端

(2) 如果 data.meta.status!=200 则表示请求失败，并且会包含code、message 属性，code属于自定义标识code（可见表3-2）、message 属性中包含有错误的描述,例如:

```
{  
  
  "data": {},  
  
  "meta": {  
  
    "msg": "缺少token!",  
  
    "status": 401  
  
  }  
  
}
```

识别CODE描述

400请求参数不可读

401未登录或未授权用户认证失败，需要重新登录



404资源已经移除

500服务器繁忙!

501服务器繁忙!

表3-2 自定义状态码

前端设计

工厂模式 Factory Pattern: 工厂模式是一种很常用的模式。由一个工厂对象决定创建某一种产品对象类的实例，主要来创建同一类型产品对象。

单例模式Singleton Pattern: 单例模式是保证一个类只有一个实例，字面意思：我要你一个就够了。并且提供一个访问它的全局访问点。

前端实现：if没有，返回创建一个新的。else，返回刚才创建的那个。

观察者模式ObserverPattern: 观察者模式是一种一对多的依赖关系。字面意思：谁关注我，谁就看得到我动态。

前端实现：定义一个Array，存放订阅者，当我有新消息时，通知订阅者。这是极简的观察者模式。

解决什么问题呢？

当Vue, React项目中，跨组件通信时，有时候会非常不方便。比如Car项目里是由一个父容器，包含所有的异步路由子容器的结构。总不能把所有通信方法都写在父容器上，所以一般会引用第三方的event库。

策略模式：策略模式指的是定义一系列的算法，把他们一个个封装起来，就是将算法的使用与算法的实现分离开来，避免多重判断条件，更具有扩展性。

传统的前端数据交互是用Ajax从服务端获取数据，然后操作DOM来改变视图；或者前端交互要改变数据时，又要再来一次上述步骤，而手动操作DOM是一个繁琐的过程且易出错。

Vue是一种MVVM框架。而DOM是数据的一个种自然映射。传统的模式是通过Ajax请求从model请求数据，然后手动的触发DOM传入数据修改页面。Vue中，Directives对view进行了封装，当model里的数据发生变化是，Vue就会通过Directives指令去修改DOM。同时也通过DOM Listener实现对视图view的监听，当DOM改变时，就会被监听到，实现model的改变，实现数据的双向绑定。

图3-2 数据模型驱动流程

功能模块业务逻辑设计

3.4.1 登录功能列举

登录功能

用户进入平台的登录页面，输入账号密码之后点击登录按钮发起请求然后解析请求数据解析并进行认证生成token返回，时序图如图3-2。

图3-2 用户登录时序图

数据库设计

MongoDB 支持丰富的查询操作，MongoDB 几乎支持SQL中的大部分查询。强大的聚合工具。

图3-3 关系型数据库与非关系型数据库区别

3.5.1 MongoDB数据库的设计规范

MongoDB是非关系型数据库，数据显示，近年来，MongoDB在NoSQL领域一直独占鳌头。并且可以多方面的弹性拓展。我们从集合设计规范、数据库设计规范等2个方面进行阐述和要求。



集合设计规范

集合名称约定为小写。集合名称的命名规则和MySQL数据库表的命名规则相同。固定集合可以用于记录日志，其插入数据更快，指定Size的大小，还能够指定文档的数量。集合不管先达到哪一个限制，之后插入的新文档都会把最老的文档移出。索引命名：idx_构成索引的字段名。如果字段名字过长，可采用字段缩写。

数据库设计规范

数据库名约定为小写。数据库名称不能包含除'_'以外的特殊字符，例如：. "\$。数据库名称最多为64个字符。数据库上线需经过DBA评审。

3.5.2数据表关系结构

由于MongoDB是非关系型数据库，是以文档形式存储，这里就不展示关系结构。

本章小结

本章借助图片、文字表述等方式展现架构总体设计、前后端交互设计、前端界面设计、功能模块设计和数据库设计，为在平台具体实现和开发的过程中提供充分的指导作用

总结与展望

本章总结了整个毕业设计以及毕业论文，同时提出今后工作的展望。

工作总结

在将近两个月的时间里，已基本完成了毕业设计。在编程的过程中，学到了很多知识，不仅是编程开发方面的专业技能，更锻炼了自己的毅力和耐力。

经过一段时间的设计和开发，外卖平台基本开发完毕，平台功能基本符合外卖平台的需求，由于时间比较紧迫，该平台还有些不足之处，比如有些界面做的不够漂亮，不够人性化，有的操作有点繁琐，这些都是后续工作可以更好完善的。因为对外卖平台的调研不够精确，所以有些问题没有考虑到，导致有些实现的功能不够周全，还需要进一步的完善。对于一些复杂的代码还需要进一步的修改，使之更加简捷易懂。

在长达的几个月中，毕业设计已经基本完成，在编写毕业论文的过程中，出现许多问题与困难，遇到困难时也会有所烦躁但烦躁解决不了问题还是要沉下心来解决问题，针对自己遇到的问题通过百度查找了大量的资料和论坛上一些牛人的解答下我的问题也得到了解决。在论文的编写与作品的开发中我学习到了很多东西，比如提高了编程的开发的还有写作能力等。通过这次的写作与开发更锻炼了我自己的毅力与耐力。

在这个月中的时间里我完成了开发与论文的编写，平台也已经基本完善，平台功能也基本符合外卖平台的需求，由于个人能力有限，平台还是有一些不完美的地方，比如一些页面不够完美，有的功能操作有点繁琐，在往后的时间里会不断对该平台继续完善。由于对外卖平台的调研不够准确，有些细节问题没有注意到，导致有些功能不够完美，还需要进一步的完善。一些复杂的代码还需要进一步的改良，使之更加简捷易懂。

致谢

三年时间说长不长说短不短，一眨眼三年时间已剩不多，这三年有太多值得留念的事情。还清晰记得入学的那年懵懵懂懂的选择了计算机方向的专业，而我对于这个专业的认知仅仅是在百度所查找得到的信息。在学校安排的一次演讲中让我对这个专业有了更深层次的认知。也是因为这场演讲让我知道了科创工作室的存在，科创对于我来说就好比是灯塔，是他照亮了路的方向，使我不用在黑夜里探索方向。工作室的师兄师姐以及老师给予我很大的帮助，如果不是你们我的大学生活应该是懵懵懂懂的入学到懵懂的毕业。在工作室的时间里，我收获了很多，比如认识了许多志同道合的同学以及朋友，工作室的氛围给我的感觉就像是这群人不会轻易向命运低头，努力了就会有收获。还有最值得感谢的一位老师，汤老师！是您我们才有机会参加各种比赛，为我们忙前忙后。感谢您的辛苦付出。

最后祝母校，蒸蒸日上，更创新高！

参考文献

- [1] 邹琼俊. Vue.js 2.x实践指南. 清华大学出版社, 2020.
- [2] 霍多罗夫, (美) 迪洛尔夫. MongoDB权威指南. 人民邮电出版社, 2011.



- [3] (美)(Nicholas C.Zakas)扎卡斯. JavaScript高级程序设计.人民邮电出版社, 2006.
- [4] 陆凌牛. HTML 5与CSS 3权威指南. 机械工业出版社, 2011.
- [5] 刘振涛. 深入理解ES6.电子工业出版社, 2017.
- [6] 朴灵. 深入浅出Node.js. 人民邮电出版社, 2013.
- [7] 阮一峰. ES6标准入门 (第3版) .电子工业出版社, 2017.
- [8] Nicholas C. Zakas (尼古拉斯.泽卡斯) .高性能JavaScript.电子工业出版社, 2015.
- [9] 韩骏. Visual Studio Code 权威指南.电子工业出版社, 2020.
- [10] [美]安德鲁·福克纳 (Andrew Faulkner) 、康拉德·查韦斯 (Conrad Chavez) . Adobe Photoshop CC 2018经典教程 (彩色版) . 人民邮电出版社, 2020.
- [11] 菜鸟教程: <https://www.runoob.com>
- [12] Vue.js: <https://cn.vuejs.org>

说明:

- 1.文献相似度 = 送检文章中与检测范围所有文献的相似字符数/送检文章正文字符数
- 2.去除参考文献相似度 = 送检文章中检测范围所有文献 (不包括参考文献) 的相似字符数/送检文章正文字符数
- 3.去除本人已发表论文相似度 = 送检文章中与检测范围所有文献 (不包括自引) 的相似字符数/送检论文正文字符数
- 4.单篇最大相似度: 送检文章与某一文献的相似度高于全部其他文献
- 5.正文字符数: 送检文章正文部分的总字符数, 包括汉字、非中文字符、标点符号、阿拉伯数字 (不计入空格), 正文不包括关键词、目录、图片、附录、参考文献等

