

**本科生实习报告**

**实习类型 综合实习**

**题 目 软件工程专业实习**

**学院名称 计算机与网络安全学院（牛津布鲁克斯学院）**

**专业名称 软件工程**

**学生姓名 尚伟维**

**学生学号 201701050213**

**指导教师 李军**

**实习地点 杭州哈行网络科技有限公司**

**实习成绩**

**二〇二一年 七月 ---- 二〇二一年 十月**

目录

[第1章 前言 4](#_Toc97137798)

[1.1 公司介绍 4](#_Toc97137799)

[1.2 本人实习岗位职责说明 4](#_Toc97137800)

[1.3 身份证明 4](#_Toc97137801)

[1.4 实习项目介绍 5](#_Toc97137802)

[1.4.1 总体介绍 5](#_Toc97137803)

[1.4.2 开发技术 5](#_Toc97137804)

[第2章 相关技术研究 6](#_Toc97137805)

[2.1 前后端分离开发模式 6](#_Toc97137806)

[2.1.1 前后端人员解耦 6](#_Toc97137807)

[2.1.2 前后端代码解耦 6](#_Toc97137808)

[2.1.3 可扩展性 6](#_Toc97137809)

[2.1.4 资源优化 7](#_Toc97137810)

[2.1.5 模块化 7](#_Toc97137811)

[2.1.6 API 的整合 7](#_Toc97137812)

[2.2 前端开发相关技术 7](#_Toc97137813)

[2.2.1 Vue 7](#_Toc97137814)

[2.2.2 Element UI 8](#_Toc97137815)

[2.2.3 webpack 8](#_Toc97137816)

[2.3 后端开发相关技术 8](#_Toc97137817)

[2.3.1 Node.js 8](#_Toc97137818)

[2.3.2 Koa 9](#_Toc97137819)

[2.3.3 Spring Boot 9](#_Toc97137820)

[2.3.4 MySQL 9](#_Toc97137821)

[2.3.5 MongoDB 9](#_Toc97137822)

[第3章 需求分析 10](#_Toc97137823)

[3.1 系统功能性需求分析 10](#_Toc97137824)

[3.1.1 可视化工具 11](#_Toc97137825)

[3.1.2 开箱即用功能 11](#_Toc97137826)

[3.1.3 拖放功能 11](#_Toc97137827)

[3.1.4 访问控制功能 11](#_Toc97137828)

[3.1.5 新建页面功能 11](#_Toc97137829)

[3.1.6 发布上线功能 11](#_Toc97137830)

[3.2 非功能性需求分析 12](#_Toc97137831)

[3.2.1 可重用性 12](#_Toc97137832)

[3.2.2 安全性 12](#_Toc97137833)

[3.2.3 低成本 12](#_Toc97137834)

[第4章 整体设计 12](#_Toc97137835)

[4.1 系统架构设计 12](#_Toc97137836)

[4.2 页面管理模块设计 13](#_Toc97137837)

[4.3 可视化工具模块设计 14](#_Toc97137838)

[4.3.1 权限功能设计 14](#_Toc97137839)

[4.3.2 登陆注册功能设计 14](#_Toc97137840)

[第5章 测试 14](#_Toc97137841)

[5.1 功能性测试 15](#_Toc97137842)

[5.2 性能测试 16](#_Toc97137843)

[5.2.1 性能测试概要 16](#_Toc97137844)

[5.2.2 性能测试方法 16](#_Toc97137845)

[5.3 安全性测试 16](#_Toc97137846)

[5.3.1 安全性测试概要 16](#_Toc97137847)

[5.3.2 安全性测试方法 17](#_Toc97137848)

[5.3.3 测试结果 17](#_Toc97137849)

[第6章 实习体会 17](#_Toc97137850)

# 前言

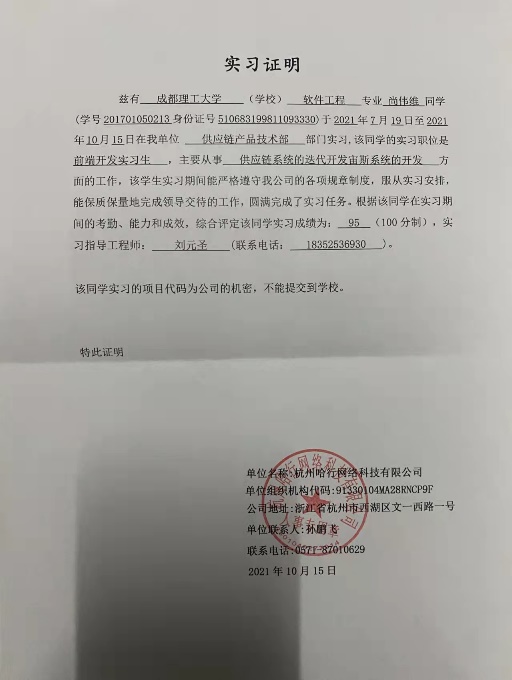
## 公司介绍

我所实习的公司为杭州哈行网络科技有限公司。哈啰出行是国内专业的本地出行及生活服务平台，致力于应用数字技术的红利，为人们提供更便捷的出行以及更好的普惠生活服务。公司成立于2016年9月，总部位于上海，从大家熟悉的共享单车业务起步。目前哈啰出行主要提供移动出行服务及新兴本地服务，其中，移动出行服务包括两轮共享服务（哈啰单车、哈啰助力车）和四轮出行服务（哈啰顺风车、哈啰打车）；新兴本地服务包括自主品牌哈啰电动车，以及与蚂蚁集团、宁德时代合资建立的小哈换电。此外，哈啰出行还在特定城市开发和试点包括酒店预定、到店团购等的产品和服务。

## 本人实习岗位职责说明

我从事JavaScript前端开发的实习工作。平时主要负责宙斯Serverless平台的开发、ERP系统的页面维护和波塞冬低代码平台的工作。

## 身份证明



## 实习项目介绍

### 总体介绍

ERP系统是把供应链系统SCM供应链管理系统迁移出来和物流管理、履约管理、电池管理、资产管理、新业务使用微前端技术重新整合成一个系统。SCM供应链管理是一种集成的管理思想和方法，它执行供应链中从供应商到最终用户的物流的计划和控制等职能。从单一的企业角度来看，是指企业通过改善上、下游供应链关系，整合和优化供应链中的信息流、物流、资金流，以获得企业的竞争优势。

波塞冬低代码平台，为公司ERP系统提供了一个可视化搭建页面的系统，通过配置的方式去生成前端页面，同时也支持代码编写，大大加快了前端页面的编写速度。低代码需要解决的核心功能就6个：表单、视图、工作流、报表、报告、组织架构和权限。

宙斯Serverless项目，主要方向是Fass云函数。为公司内部提供了一个云函数平台，让很多接口可以部署在该项目上。云函数是直接部署在服务器上面的接口方法，很多接口逻辑不需要写到项目里，很多项目共同的接口也直接可以共用一个云函数。云函数可以随时部署替换，不需要审核发布，节省了项目发布等时间。除此之外还有云函数的专属的方法以及操作数据的权限。

### 开发技术

ERP项目采用B/S系统架构，后端采用Java语言进行开发。后端整体使用Spring boot、MyBatis，并使用了Docker容器和负载均衡技术。数据库使用了MySQL，还使用了Redis缓存技术。前端主要使用Vue2.0和Element UI框架，在项目中也使用了qiankun微前端框架。

波塞冬低代码平台项目采用了B/S系统架构，前后端均采用了JavaScript语言开发，前端使用了Vue2.0和Element UI框架，后端使用了Node.js，数据库使用了MySQL。

宙斯Serverless项目采用B/S系统架构，前后端均采用JavaScript语言开发，后端采用Node.js,并使用了Express.js、Koa.js框架，数据库使用了MySQL，还使用了Zookeeper和Docker技术。前端使用了Vue2.0、ECharts和Element UI框架。

# 相关技术研究

## 前后端分离开发模式

前端和后端的分离更多是一种架构概念。在传统的Web架构中，经典的MVC中可以分为数据层、逻辑层和视图层。在如今时代的演变中这个视图层变成了现在的前端，它映射到代码层面，即HTML、JS、CSS等代码文件，数据层和逻辑层是比较后端的部分，比如我们Java，Go，Python等代码文件, 这些文件将在一个项目中，不会单独开发、测试和部署。在前后端分离的架构中，前后端分离在不同的项目中。前端有专门的前端开发人员进行开发和测试，后端有后端开发人员进行开发和测试。它们通过 API 相互交互。前端分离的常用认证方式有以下几种：会话Cookie、令牌验证、OAuth（开放授权）。令牌验证一般指JSON Web Token，是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准(RFC 7519)。该token被设计为紧凑且安全的，特别适用于分布式站点的单点登录（SSO）场景。JWT的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便于从资源服务器获取资源，也可以增加一些额外的其它业务逻辑所必须的声明信息，该token也可直接被用于认证，也可被加密。在前后端分离架构中，可以和Shiro配合使用。

### 前后端人员解耦

让前端和后端由更擅长的人来做，细化工种，更专业化。前端人员关心用户体验、UI设计、交互渲染；后端人员更关注业务逻辑、性能保障、安全等方面。在项目进度方面，前端和后端可以并行开发，互不影响，从而加快整体项目进度。

### 前后端代码解耦

后端只需要提供API服务，不再与静态文件交互。后端可以使用更复杂的分布式和微服务架构来提供更好的性能和稳定性。同时，除了PC端，移动端也可以使用同一套后端服务。

### 可扩展性

在前后端分离中扩展应用程序更容易。由于代码分为两部分，可以更快地优化代码。其次，您可以以单独的速度增加前端和后端的资源，因为随着您的增长，后端需要以相对更快的速度增加。

### 资源优化

在不分离的项目中，您的服务器完成所有工作，从解析请求、从数据库中检索相关数据、根据请求对数据应用计算或修改，然后使用模板以 HTML 格式将它们很好地组合在一起并将其发送到浏览器。而在前后端分离中，服务器只是将原始数据（主要是 JSON 格式）发送到浏览器，浏览器将它们很好地无缝地组合在一起。服务器需要更少的工作并且用户体验更好，因为整个页面不会随着每个请求而更新。

### 模块化

在前后端分离中，在一个模块上工作变得更容易，而另一个模块则保持不变。模块化还使两个团队或两个人更容易同时在前端和后端工作，而不必担心覆盖或弄乱其他人的工作。他们也可以以不同的速度工作。

### API 的整合

在这个不断增长的设备世界中，您需要管理越来越多的代码版本，您的电脑网页、手机网页，小程序，iOS 应用程序、Android 应用程序等。在大多数情况下，您向所有这些应用程序提供相同的信息。因此，只有从相同的代码库为它们提供服务才有意义。拥有一个基于 API 的网站可以让您的开发人员的生活变得更简单。处理更少的代码使其更易于管理。

## 前端开发相关技术

实习公司使用的前端技术框架主要是Vue2.0和Element UI框架。工程化的工具使用的是webpack。在前端中目前使用的是HTML，CSS和JavaScript技术，而Vue2.0是JavaScript的框架，目前Vue的正式版本在Vue3.0，但是目前大部分企业为了适配，都是继续沿用Vue2.0，用Vue3.0重构Vue2.0的成本太高。

### Vue

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

### Element UI

Element UI是一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，包含了一套设计原则、组件和组件，还提供如Axure组件、Sketch模板的设计资源。Element UI由饿了么前端开源维护，是最流行的Vue框架之一。

### webpack

webpack 是代码编译工具，有入口、出口、loader 和插件。webpack 是一个用于现代 JavaScript 应用程序的静态模块打包工具。当 webpack 处理应用程序时，它会在内部构建一个依赖图(dependency graph)，此依赖图对应映射到项目所需的每个模块，并生成一个或多个 bundle。

## 后端开发相关技术

实习公司在波塞冬低代码平台使用的是Node.js后端，框架是Koa2，在平时业务中使用的是Sprint boot框架。Node.js 是一个开源和跨平台的 JavaScript 运行时环境。数据库使用的是MySQL和MongoDB。

### Node.js

Node.js 在浏览器之外运行 V8 JavaScript 引擎（Google Chrome 的内核）。这使得 Node.js 的性能非常好。Node.js 应用程序在单个进程中运行，无需为每个请求创建新的线程。

Node.js 在其标准库中提供了一组异步的 I/O 原语，以防止 JavaScript 代码阻塞，通常，Node.js 中的库是使用非阻塞范式编写的，使得阻塞行为成为异常而不是常态。

当 Node.js 执行 I/O 操作时（比如从网络读取、访问数据库或文件系统），Node.js 将在响应返回时恢复操作（而不是阻塞线程和浪费 CPU 周期等待）。这允许 Node.js 使用单个服务器处理数千个并发连接，而​​不会引入管理线程并发（这可能是错误的重要来源）的负担。

Node.js 具有独特的优势，因为数百万为浏览器编写 JavaScript 的前端开发者现在无需学习完全不同的语言，就可以编写除客户端代码之外的服务器端代码。

在 Node.js 中，可以毫无问题地使用新的 ECMAScript 标准，因为你不必等待所有用户更新他们的浏览器，你负责通过更改 Node.js 版本来决定使用哪个 ECMAScript 版本，你还可以通过运行带有标志的 Node.js 来启用特定的实验性功能。

### Koa

Koa 是一个新的 web 框架，致力于成为 web 应用和 API 开发领域中的一个更小、更富有表现力、更健壮的基石。通过利用 async 函数，Koa 帮你丢弃回调函数，并有力地增强错误处理。Koa 并没有捆绑任何中间件，而是提供了一套优雅的方法，帮助您快速而愉快地编写服务端应用程序。

### Spring Boot

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

### MySQL

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，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，一般中小型网站的开发都选择 MySQL 作为网站数据库。

### MongoDB

MongoDB是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似JSON的BSON格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

# 需求分析

波塞冬低代码平台的出现是为了使用用户业务快速发展对数字化系统的需求、降低开发成本和减少对高技术和人员的依赖。在如今社会向数字化转型的大趋势下，很多的数字化系统需要大量的技术人员。 与其编写成千上万行的复杂代码和语法，不如使用低代码来构建具有现代用户界面、集成、数据和逻辑的完整应用程序，这种方法快速且可视化。应用程序交付得更快，而且只需要最少的代码编写。并且能够用于在测试、阶段化和生产中构建、调试、部署和维护应用程序的自动化工具

在我们传统的系统开发中，会需要比低代码平台更多的成本、人力和时间。在大多数场景下，低代码平台不需要开发人员掌握传统的编程语言知识，通过可视化的开发搭建出页面。这种拖拽的方式降低了软件开发的技术门槛，让很多角色头都可以投入到开发任务中。我所在的部门主要就是BPM、SCM、ERP系统的维护，低代码系统设置起来很简单，通常托管在云中。对于完整的本地配置和部署，传统BPM、SCM、ERP系统的维护，可能需要长达6个月的时间。定价也可能是一个很大的不同。对于企业来说，传统的 BPM、SCM、ERP系统解决方案的实现和运行都非常昂贵。有一些新的 BPM、SCM、ERP系统 解决方案可以减少这个数字，但是它们可能没有您所需要的灵活性。总的来说，低代码系统允许您构建任何类型的应用程序。BPM、SCM、ERP系统 解决方案可能有也可能没有对其进行自定义的编码能力，但应用程序必须与工作流相关。

## 系统功能性需求分析

波塞冬低代码平台的主要功能是可视化工具、开箱即用功能、拖放功能、访问控制功能、新建页面功能、发布上线功能。与使用复杂的编程语言不同，您可以在低代码开发平台中使用具有基本逻辑和拖放功能的可视化接口。低代码创建框架为用户提供了将构建块合并到工作流和应用程序中的能力。这些基本元素抽象了操作和命令背后的代码，允许用户创建接口和业务应用程序，而无需手工编码。

### 可视化工具

与使用代码开发应用程序相比，使用可视化方法和模板创建应用程序花费的时间更少。低代码系统配备了可视化建模特性，使用内置模块以每个人都可以理解的方式反映细节——从非技术的用户到专业开发人员。

### 开箱即用功能

开箱即用软件是由领先的低代码平台提供的，从一开始就不需要为应用程序创建关键组件。例如，一些低代码系统提供数据存储或面向客户的应用程序组件，如服务管理或销售过程管理。

### 拖放功能

拖放功能可以在任何已知的低代码平台上使用。它是促进生产过程的最有关键和最有价值的功能之一。在整个应用程序开发过程中，方便的拖放功能为普通开发人员和技术开发人员提供了支持。

### 访问控制功能

业务用户可以使用低代码平台方便地授予对其程序的访问权限。在广义和微观层面上，用户都可以监控谁可以访问应用程序。应用程序创建者可以将用户指定为管理员、共同所有者或最终用户。

### 新建页面功能

波塞冬低代码平台提供了两种网页模版，一种是查询基础页和产品的详情页面。前者由一个表单和一个表格组成，在其中也可以去插入一些功能按钮。表单和表格支持大部分需求，通过简单的配置项可以完成一个完整的页面。

### 发布上线功能

波塞冬低代码平台有专门的发布上线通道，和公司普通的上线不同，在该系统上进行修改，点击发布之后就可以选择发布到测试环境或者上线环境。简化了大部分流程，不用走完整的发布流程，让流程变得不繁琐。

## 非功能性需求分析

波塞冬低代码平台是一款图形化、托拉拽方式快速实现企业数字化转型中的创新应用、支持用少量代码扩展实现个性化需求的数字技术工具平台。由于低代码大大降低了软件开发的复杂性，任何规模的公司采用这种方法都有能力提高开发人员的生产力和速度。它提升了开发人员的价值，使敏捷团队能够利用他们对如何创建和维护高质量的网络和移动应用程序的理解，同时通过尝试新技术来展示他们的翅膀。

### 可重用性

低代码平台的即用型功能的一个主要方面是为应用程序使用预配置的模块和功能。这些模块具有几个应用程序所需的通用核心功能，并且可以复用以更快地开发不同的解决方案。能够重用预先构建或新开发的模块、插件，以更快地开发必要的应用程序。

### 安全性

不管低代码平台的功能和用户友好程度如何，如果它不安全，那么它都不是完美的解决方案。在开始使用低代码开发平台之前，确保其具有适当的安全性，以保护正在构建的应用程序和整个平台。

### 低成本

传统的软件制作方法——即手动编码系统的每个部分是困难的，需要时间和成本。另一方面，低代码平台允许在更短的时间内开发更多的应用程序，从而降低了开发过程的成本。低代码系统是民主化和权力的完美结合。

# 整体设计

## 系统架构设计

波塞冬低代码平台框架如图3-1所示。

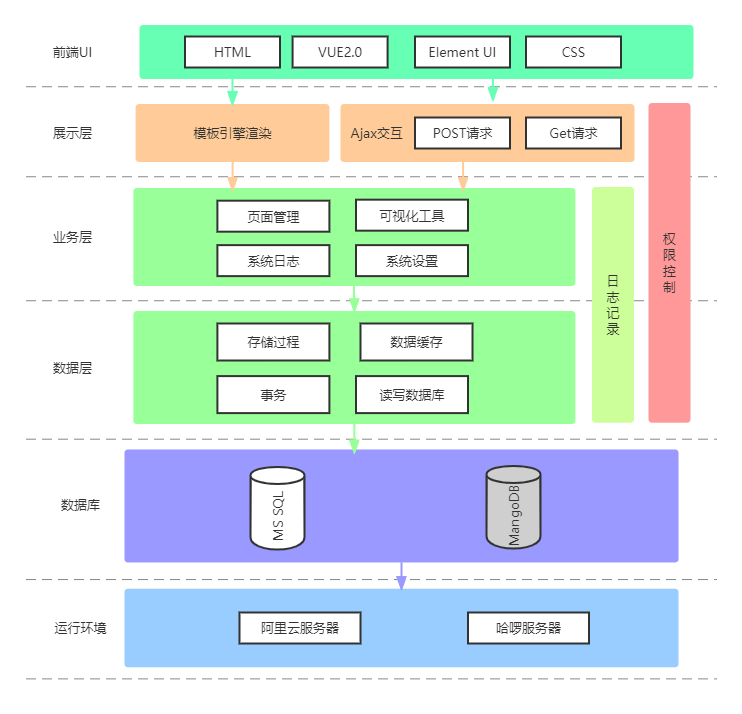


图3-1 系统架构图

在波塞冬低代码平台结构框架图中，使用HTML和JavaScript语言实现视图层，实现向用户展示的软件界面，使用 Node.js实现表示层，当点击视图层相应的按钮或者组件后，触发相应的事件，表示层根据视图层提交的响应事件完成相应的逻辑业务。使用 Node.js 实现模型层，表示层完成相应的逻辑业务任务后向模型层提交数据操作请求，模型层根据操作请求的类型和识别码对MySQL数据库进行访问和操作，执行操作后，MySQL数据库返回给模型层相应的操作信息和数据，模型层接收信息并将信息返回给表示层，表示层接收模型层返回的信息，对视图层进行相应的编辑操作。

## 页面管理模块设计

页面管理包括对页面的新建，删除，修改，查看。用户点击新建页面跳转到新建页，新建页面（可视化工具页面），用户点击删除后弹出弹窗让用户二次确认是否删除，点击修改按钮之后进入到编辑页面中，点击查看按钮后跳转到查看页面。需要有该人员有权限才可以进行相应的操作。

## 可视化工具模块设计

可视化工具包括三个部分：组件选择部分、页面编辑部分、组件页面配置部分。组件有各种简单编辑的组件，包括UI组件、逻辑组件、数据相关组件、API组件。用户在组件功能区域去选择组件拖动带页面编辑部分的对应位置，在页面编辑部分可以点击组件进行配置，在配置区域可以配置组件的数据来源，配置组件的外观大小，配置组件的属性名字。

### 权限功能设计

在SSO系统中项目管理员就是波塞冬低代码平台的超级管理员，拥有页面和人员的全部控制权限。超级管理员分配的人员才有新建页面的权限，用户新建的页面拥有完全控制权。该用户可以给其他用户分配该新建的页面权限。

### 登陆注册功能设计

使用公司内部登陆机制SSO验证进行登陆，用户需要在入职时找到管理员开通账号才能登陆波塞冬低代码平台系统。

# 测试

在本章中考虑低代码开发平台中，从普通商业用户和经验丰富的应用程序开发人员的角度进行了测试。独立测试，试图了解同一个工具如何处理不同级别的开发专业知识和一组不同的需求，具体取决于要构建的应用程序的类型。

为了从普通B端业务用户的角度进行测试，使用每个相应的低代码工具来构建相同的基本调度应用程序。从开发人员的角度进行测试时，使用每个工具构建的标准应用程序有点复杂。这个应用程序成为一个简单的页面管理系统，具有新建列表页面、列表详细信息页面和管理新页面。能够引入第三方服务并向应用程序添加任何附加功能或自动化逻辑是一个优势。我们需要一个稍微复杂一点的应用程序，无论是在桌面还是移动设备上都有用。

根据几个因素衡量。开发人员是否能够实现完整的功能集，并随着时间的推移模拟应用程序的更改。经常需要将修复和更新推送到业务应用程序，模拟流程的项目维护方面，开发人员还测试处理向数据模型添加新字段并将该更改推送到应用程序，以及更改现有字段以查看更改在没有应用程序错误的情况下反映。构建一个基本的、可工作的应用程序，与传统编码相比，基于表单和拖放对象建模的 UI 是否更容易、更省时。在低代码开发过程中提供了定制特性和附加功能，平台在构建应用程序时需要任何编码。

我们从三个方面进行了测试，分别是功能性测试、性能测试和安全性测试

## 功能性测试

功能性测试对波塞冬低代码平台的设计文档所提及的功能进行测试，通过对波塞冬低代码平台所涉及到的功能进行使用，看其是否满足设计需求，测试的功能模块如表5-1所所示。

表5-1 功能组件测试

|  |  |  |  |
| --- | --- | --- | --- |
| 功能组件 | 测试内容 | 测试结果 | 异常情况 |
| 页面管理 | 新建页面 | 通过 | 无 |
| 删除页面 | 通过 | 无 |
| 改动页面 | 通过 | 无 |
| 查看页面 | 通过 | 无 |
| 可视化工具 | 拖动组件 | 通过 | 无 |
| 删除组件 | 通过 | 无 |
| 修改组件 | 通过 | 无 |
| 组件配置 | 通过 | 无 |
| 权限管理 | 修改页面权限 | 通过 | 无 |
| 新增页面权限 | 通过 | 无 |
| 删除页面权限 | 通过 | 无 |
| 查看页面权限 | 通过 | 无 |
| 修改用户权限 | 通过 | 无 |
| 新增用户权限 | 通过 | 无 |
| 登陆功能 | 登陆系统 | 通过 | 无 |

## 性能测试

### 性能测试概要

系统性能测试有两部分组成，将分别对单一场景下的系统使用和综合场景下的波塞冬低代码平台使用进行测试，验证系统是否满足设计需求，同时测试系统的处理能力。

### 性能测试方法

测试使用的软件是Jmeter，Jmeter是Apache提供的一款功能比较齐全的性能测试工具。用例既可以通过GUI进行编写，也可以通过录制脚本的方式创建。另外，Jmeter也可以作为应用的功能/回归测试，通过设定结果断言，脚本会自动判断请求是否返回了正确结果。Jmeter GUI是Jmeter的图形模式，提供可视化的编辑方式，多样的监听器，适合于创建测试计划以及脚本调试。软件可以通过模拟大量的用户同时实施操作，模拟大并发的场景，测试系统在大并发情况下的系统性能，并提供详细的测试结果，通过这些结果，我们可以盘查问题，解决问题。然后根据系统需求的能力要求，添加能实现系统需求的虚拟用户，然后让用户连续进行模拟真实用户的操作。这样测试系统在单一场景下，各个业务模块的性能；在综合场景下，单节点服务器的性能能够满足系统设计同时在以后对系统有升级需求时，可以兼容，性能要求如表5-2所示。

表5-2 性能测试结果

|  |  |  |
| --- | --- | --- |
| 响应时间 | 平均响应时间 | 每秒事务处理数（TPS） |
| 0.4 | 0.4 | 30 |

## 安全性测试

### 安全性测试概要

安全性测试主要依据CVE（Common Vulnerabilities Exposures）进行，主要为了发现安全漏洞和潜在的安全隐患，最大限度的降低系统的上线风险。安全性测试主要分为两类，一类是对整个系统进行安全性测试，另一类是测试某些特定的功能进行的安全性测试。本次安全性测试时全面的安全性的测试，测试整个系统。

测试主要针对外网可以通过链接进行访问的网址，测试采用安全工具扫描和手动扫描两种扫描方法，安全工具负责对系统进行全面细致的扫描，因为手动扫描耗时问题，所以只针对关键功能进行手动扫描，同时扫描安全工具扫描出的问题，进行二次确认。

### 安全性测试方法

针对波塞冬低代码平台，计划将测试分为两个部分，分别是渗透测试和漏洞扫描测试。

在渗透测试中，通过利用专业的安全渗透测试软件进行，选用AppScan作为本次测试主要使用的软件。检测包括：跨站脚本攻击（CSS和XSS）、SQL注入攻击，远程命令执行、系统路径遍历、文件包含、脚本代码暴露、URL重定向、Cookie篡改、数据攫取等方面内容。

### 测试结果

通过利用AppScan对系统进行检测，在目前的检测手段下，波塞冬低代码平台能抵抗目前常见的攻击方式，能有效的预防不法分子对系统的攻击。

# 实习体会

作为供应链前端的一名实习生，在哈啰出行短短三个月的时间里，不一定有很大的进步，但是也会有很多的收获。在这里我做了两部分的需求：第一部分是技术项目部分，另一部分是业务需求部分。

在技术项目上我一共完成了7个需求，4个前端需求，3个后端需求。第一个是这是优化调试面板，对一个数组数据完成拆分和格式化。这是对我是一个有挑战的需求。第二个是目录树，包括目录的展示，新建和删除。第三个是JSON格式校验，这个是当用户输入的格式不正确时，不允许发起AJAX请求。第四个是切换云函数, 因为Element UI没有合适的组件，自己写了一个这样的组件。最后三个就是后端部分了，用node实现其中的功能。

在业务需求上面，一共完成了9个需求，这个8个中有3个基础的优化需求，分别是ECN&物料认证优化、物料领还优化和SIM卡发货迭代，有4个在项目中添加新页面分别是车辆管理、电动车物料维护、电动车发货专项支持、电池资产位置信息。1个改造项目成品入库改造。

在这些需求中，我遇到了两个比较棘手的需求，优化调试面板和成品入库改造。

优化调试面板这个需求，我先把这个需求拆解成了三个部分。第一个部分是数据的格式化，把一个一维的数组，拆分为时间戳和语句对应的键值对。第二个部分是，控制展示的数据缩进和换行。第三个部分是，展示的数据一些某些特定的字符需要上色(比如null， undefined，true，false，字符串，数组)。一共有两种数据：类似JSON的字符串，但是在其中存在回车的转义字符\n和JSON的格式的对象，在Vue可以直接利用Mastache语法，也就是{{}}直接在界面上渲染对象和字符串。但是他们不会主动的换行, 我的思路陷入了每一行单独渲染的错误, 通过请教师兄我想到了解决办法，因为在渲染字符串的时候，浏览器会自动忽略里面的转义字符，利用 CSS 属性 white-space: pre-wrap, 让\n 转义字符生效，于是就解决核心的换行的问题。

缩进相对于比较简单，根据回车的位置去插入空格即可。第三个部分上色，因为需要使用一种解决办法解决两种数据格式的上色，我首先需要去控制这两种格式一致。因为对象是一个整体, 他会一起渲染到页面上，没有办法拆分，我就必须放弃对象直接渲染。就需要把对象转为类似JSON格式的字符串, 并且在其中插入 \n, 控制\n位置非常困难，最后通过后端返回同一种数据结构解决了。到了后面的上色轻松了许多，只需要用正则匹配到需要上色的位置包裹住该部分插入<span color: "对应的颜色"></span>就完成了。

第二个比较大的挑战就是成品入库的改造，这个需求不像之前的需求有非常完整的PRD，只有本次迭代的新需求。最困难的地方就是查看老代码去理解出完成逻辑。老代码问题存在很多问题：经手过很多人、代码风格会多变、几乎没有注释。导致很难从中理解出以前的逻辑，并且要自己重新编写新代码完成以前的逻辑。通常情况下，我们只能看到页面上存在的逻辑，往往会看不到一些隐藏的逻辑（比如通过后端某个参数,在界面需要展示不同的几个部分）。在这种情况下只能完全相信测试的同学会帮助发现所有的流程和问题。通过测试同学提出bug后,发现一些隐藏的逻辑，可能会导致新代码，又会重新构思整个页面。会导致重构的代码质量下降。

对于这样一个比较有挑战性的需求, 我得到了一些体会, 请教是我们成长较快的途径，还有一个感受就是 沟通是解决问题较快的途径，很大的会影响工作效率。我们功能的要求经过测试很严，代码方面要求并不是很严格，命名规范, 注释, 文件放置位置。

在工作中，习惯了哈啰出行的作息节奏, 锻炼出一定的抗压能力，转变了之前到点下班的思想, 现在是以需求为第一目标。在技术方向上，了解到一些新的技术方向，比如微前端、Serverless Fass和低代码平台，到后面慢慢的自己可以独立去完成一个需求，自己独立和产品和后端沟通。

|  |  |
| --- | --- |
| **学生**  **实习**  **心得** | 实习过程中自己不仅仅学习到了规范的开发流程，项目上线流程，也提升了自己对代码的编码能力，阅读能力。在师兄的帮助下也了解到了前端的未来发展方向，微前端，Serverless和低代码，有利于自己对未来发展的职业规划。短短三个月更加深刻的掌握了Vue框架和Node.js。  学生（签名）：  年 月 日 |
| **诚信承诺** | 本人郑重声明所呈交的实习报告是本人在指导教师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注的地方外，报告中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同学对本文研究所做的贡献均已在报告中作了明确的说明并表示谢意。  学生（签名）： |
| **指导**  **教师**  **评语** | 成绩评定：  指导教师（签名）：  年 月 日 |