【摘要】随着人们的生活娱乐方式逐渐变得丰富，人们看视频、浏览网站不再愿意经过漫长的等待，而是升级网络带宽以及升级自己电脑的配置，此时云主机的出现满足了用户的需求。本文重点阐述通过前后端分离的开发方式，使用Vue+Keystone 开发框架，Keystone后台自带数据库，Sublime开发工具来完成整个能力平台系统。且测试结果良好，用户可以注册、登录网站后，添加订单到购物车，通过修改CPU核数、内存大小、系统盘类型、数据盘类型及大小等配置，提交订单至管理员审核。管理员可通过订单或不通过订单，通过的订单可以在用户的资产中查看到，不通过的订单则出现在用户的购物车，用户可继续修改配置直至资源池中有该配置，管理员通过该订单。用户可进入个人管理平台修改自己的密码，管理员亦可帮助用户修改密码为指定密码。

【关键词】 Vue开发框架 Keystone内容管理系统 能力平台

【Abstract】As people's life and entertainment methods gradually become richer, people watch videos and browse websites are no longer willing to wait for a long time, but upgrade network bandwidth and upgrade the configuration of their own computers. At this time, the emergence of cloud hosting meets the needs of users. This article focuses on the separation of front and back-end development methods, the use of Vue + Keystone development framework, Keystone back-end database, Sublime development tools to complete the entire capacity of the platform system. And the test results are good, users can register, login website, add orders to the shopping cart, modify the CPU core count, memory size, system disk type, data disk type and size and other configuration, submit the order to the administrator for review. The administrator can pass the order or not. The passed order can be viewed in the user's assets. The non-passed order appears in the user's shopping cart. The user can continue to modify the configuration until the configuration is available in the resource pool,and the administrator passes the order. Users can enter their personal management platform to modify their own passwords, and administrators can help users change their passwords to specify their passwords.

【Key Words】Vue Keystone Capability Platform

一、绪论

1.1选题背景

科技发展与我们的社会生活息息相关，相辅相成。人们渐渐意识到，科学技术已经深深地影响着我们的日常生活，给我们带来了很多便利，在经济社会发展扮演者不可或缺的角色。21世纪以来，科学技术，尤其是计算机网络技术、电子信息技术的飞速发展，使得手机、电脑那些昂贵的奢侈品步入寻常百姓家，成为我们生活中的必需品。科技发展让人们意识到科技在生活中能发挥的作用，人们的需求又促进了科技的发展。随着人们的生活娱乐方式逐渐变得丰富，人们看视频、浏览网站不再愿意经过漫长的等待，而是升级网络带宽以及升级自己电脑的配置，此时云主机的出现满足了用户的需求。

随着计算机技术的高速发展，交互设计成为产品设计中不可或缺的部分，而创新、有趣的情感化体验是交互设计考虑的重要因素。情感化体验是一种人性化、丰富、复杂、不可确定的体验，因为情感不是一种理性认知、逻辑推理的过程【欧静,赵江洪.多维情感-动作与产品形态的交互设计研究[J].包装工程.2015,(18):49-53.】。终端是否完备，不仅是决定此设计的自身，还涉及到了产品相关联的服务带给人们的完善体验【许晓云,王一琳.扁平化——以简约为核心的产品交互设计研究[J].设计. 2015,(19):80-81.】。

用户体验和人与产品的交互密切相关,交互行为无处不在,但并非都能达到预期目标,主要是由于存在“认知摩擦”。理解和应用交互设计方法,设计交互式产品是解决“认知摩擦”的有效途径。【李世国,华梅立,贾锐.产品设计的新模式———交互设计[J].包装工程.2007,28(4):90-92.】。

交互的主要含义就是规划和描述事物的行为。和以前传统的设计模式相比较，交互设计在规划和描述事物行为的时候，主要是对事物的内容和内涵进行规划和描述。交互设计的宗旨是对事物的行为和界面进行相应的改造，它是连接时尚和科技、生活和艺术的纽带【关晓娜.交互设计时代平面设计的艺术表达[J].大舞台.2015,(6):88-89.】。

交互设计改变了设计中以物为对象的传统，直接把人类的行为作为设计对象。【王玉梅,胡伟峰,汤进,李世国.产品交互设计中场景理论研究[J].包装工程. 2017,(6):76-80 .】。

交互设计不仅关注如何能更简单的识别产品和更轻松地进行操作，更重视如何使用户获得情感上的满足，这种对情感的需求正是用户追求体验之使然。【王玉珊,李世国.情感记忆在交互设计中的价值与应用[J].包装工程.2011,(2):56-59.】

人机界面,传统意义上指的是计算机软硬件系统中用户看得见、摸得着的部分,如屏幕上的图像、文字和键盘、鼠标等。扩展到普遍意义的产品设计领域,其概念有了进一步的延伸。好的人机界面美观易懂、操作简单且有引导功能,使用户增强兴趣,提高使用效率。在产品使用与功能实现中,界面是人与机器 - 环境发生交互关系的具体表达形式,是实现交互的必要手段。如果把功能可用性比喻为产品的灵魂,界面则可称其为负载灵魂的肉体。人机交互中的界面形成,是一个科学的设计过程,需要整合多方面知识,依据目标功能的实现途径进行逻辑性的规划创造,而绝不能理解为交互信息和载体的单一美化【刘永翔.基于产品可用性的人机界面交互设计研究[J].包装工程.2008,29(4):81-83.】。

在交互设计过程中，通常设计师所面临的用户的需求是多种多样的，对于不同类型的设计对象以及在不同阶段的设计周期，设计所要着重解决的需求也不一 样，因此，理解各层次需求对于用户的意义，根据设计中的实际需求来调整设计的重点，对于交互设计的完成具有实际意义。高层次需求的满足以低层次需求的实现为基础，各层次需求需要统筹考虑，灵活地调整需求实现的侧重，高层次的需求必须得到重视【叶冬冬，李世国.交互设计中的需求层次及设计策略[J].包装工程.2013,(8):75-78.】。

1.2国内外研究现状

交互设计作为正式的研究方向早在上个世纪60年代就开始了。早期的交互设计主要关注的还是人与机器，尤其是与计算机的交互，其主要目的是通过提供简单易懂的操作界面使计算机和其他新兴的数字产品能够为更多的普通用户所接受。随着信息技术的不断成熟，互联网行业的发展以及用户体验对消费决策影响力的不断增强，一时间，交互设计几乎覆盖了包括信息技术、互联网、物联网、服务与体验等不同领域、不同行业的产品研发【辛向阳.交互设计:从物理逻辑到行为逻辑[J].装饰.2015,(1):58-62 .】。

美国卡根研究了不同文化情境下的动作文法特征，荷兰学者罗斯研究了动作对产品造型的影响，国内李世国研究了动作与隐喻物的对应关系，赵江洪研究了潜意识行为对产品设计的影响【欧静,赵江洪.多维情感-动作与产品形态的交互设计研究[J].包装工程.2015,(18):49-53.】。

斯坦福大学教授Terry Winograd认为交互设计“是人类交流和交互空间的设计”，强调的是用户与产品使用环境的共存以及交互场所与空间的构建；《交互设计———超越人机交互》的作者Preece等人认为交互设计是“设计支持人们日常工作与生活的交互式产品”，强调的是设计结果;创立交互设计制作室的Salolamon认为交互设计是“设计能够随着时间而逐步展现自己的产品”。强调使用产品带来的用户体验;也有人认为交互设计是一种把技术化产品变成智能化产品的一种设计方法,强调的是使技术适应人【李世国,费钎. 和谐视野中的产品交互设计[J].包装工程.2009,(1):137-140.】。

国外交互设计思想和交互式产品设计已不再囿于计算机科学领域的软件产品,正向有形的实体产品设计与开发领域渗透。美国的麻省理工学院(MIT)、卡耐基-梅隆大学(CMU)、加拿大的西蒙菲莎大学(SFU)和瑞典于默奥大学(UMEA UN IVERSITY)等高校都在开展交互式产品设计方面的相关研究,有的还设有交互设计方面的专业或研究方向。国内在这方面的研究多集中在计算机科学领域,交互设计在工业设计方面的应用和研究才刚刚起步【李世国,华梅立,贾锐.产品设计的新模式———交互设计[J].包装工程.2007,28(4):90-92.】。

React最早的时候是Facebook公司的自己做的一个项目，当时的前端框架都是基于传统的MVC模式，但是传统的MVC却很难满足F a R o k 公司当时的要求。因为当时项目是一个很庞大的项目，包含非常复杂的逻辑结构和相当多的程序代码，所以便诞生了React.js。Vue.js与React.js是有很多相似之处：

1)使用Virtual DOM。

2)提供了响应式（Reactive)和组件化（Composable)的视图组件。

3)将注意力集中保持在核心库，伴随于此，有配套的路由和负责处理全局状态管理的库。【徐頓,朱广华,贾瑶.基于VueJs的WEB前端开发研究[J].科技风.2017,(14):69-69.】

轻量级响应式框架 Vue.js 应用分析

Vue.js 是一个非常灵活开放的解决方案。它允许你以希望的方式组织应用程序，着重关注 MVVM 前端 View 和Model 部分，允许嵌入一个现有页面，而不一定要做成一个庞大的单页应用。Vue.js 并不是一个入侵式前端框架库，在配合其他库方面它给了较大的空间。例如，Vue.js 核心默认不包含路由和Ajax 功能，并且通常假定在应用中使用了一个模块构建系统。而 AngularJS 和 ReactJS 都过于庞大和全面，两者涉及了一个页面应用的所有阶段和所有功能，一旦使用，就需要抛弃现有的成熟框架，否则基础功能均导致重复冗余，降低客户的体验效果。更换基础框架的学习和维护成本巨大。Vue.js 具有以下优点：响应式编程；组件化；模块化。【麦冬,陈涛,梁宗湾.轻量级响应式框架Vue.js应用分析[J].信息与电脑理论版.2017,(7):58-59.】

基于Vuejs的WEB前端开发研究

React最早的时候是Facebook公司的自己做的一个项目，当时的前端框架都是基于传统的MVC模式，但是传统的MVC却很难满足F a R o k 公司当时的要求。因为当时项目是一个很庞大的项目，包含非常复杂的逻辑结构和相当多的程序代码，所以便诞生了React.js。Vue.js与React.js是有很多相似之处：

1)使用Virtual DOM。

2)提供了响应式（Reactive)和组件化（Composable)的视图组件。

3)将注意力集中保持在核心库，伴随于此，有配套的路由和负责处理全局状态管理的库。【徐頓,朱广华,贾瑶.基于VueJs的WEB前端开发研究[J].科技风.2017,(14):69-69.】

传统DOM处理需要编写繁杂的选择器，逐级操作还原服务器需要的 JSON 数据格式，不但操作繁琐且易出现致命错误。而通过 Vue.js 的响应式双向绑定数据，实时反映数据的真实变化并映射到数据源上，避免前端页面开发中DOM 选择器繁杂的操作，简化 Web 前端开发流程和降低开放难度，提升前端开发效率，降低开发成本和周期，提升产品使用的流畅性【麦冬,陈涛,梁宗湾.轻量级响应式框架Vue.js应用分析[J].信息与电脑理论版.2017,(7):58-59.】。

MVC模式（ModelViewController，简称MVC）是软件工程中的一种软件架构模式，把软件系统分为模型（Model）、视图（View）和控制器（Controller）3个部分。Model对象包含数据；View对象负责显示有模型包含的数据，用于与用户交互；Controller对象是介于Model与View之间的桥梁，它可以分发和处理用户的请求，选择适当的视图用于显示模型包含的数据返回给用户。Spring框架提供了构建Web应用程序的全功能MVC模块，是一种高度可配置的MVC框架，可以定制本地化和主题解析，并提供多种视图技术，实现了控制器、模型对象、分派器以及处理程序对象的多角色分离，这种分离让它们更容易进行定制。MVC模式强制性地将应用程序的输入、处理和输出分开。在MVC架构中，通过把系统分为3个基本部分，使应用系统结构更清晰，升级、维护更方便【薛峰，梁锋，徐书勋，王彪任.基于Spring MVC框架的Web研究与应用[J].合肥工业大学学报自然科学版.2012,(3):337-340.】。

交互设计需要遵循以下四个原则【谢伟,辛向阳,李世国.无意识认知交互设计探讨[J].包装工程.2015,(22):57-61.】：

1、一致性原则：一致性原则是指界面设计过程中，无论采用何种风格或形式，向用户传达的交互知识、规范或核心理念应保持不变。

2、图形优先化原则：在界面设计中，尽量使用图形来传达信息，发挥图形的优势效应，从而使人们能更直观、高效地与产品进行交互。

3、及时有效的反馈原则：要求系统界面针对用户的每一步操作行为，都能及时给予信息反馈，让用户准确获得操作行为的结果，从而帮助用户在与界面的交互中能无意识地学习交互规则和规范。

4、流畅性原则：个体对事物进行认知加工过程中所感受到的难易程度。

轻量级Spring框架提供了对MCV模式的实现机制，它主要围绕分发器进行设计，包括可配置的处理器映射、视图解析、本地化、主题解析、支持文件上传等。该机制具有角色划分清晰、可重用业务代码等有点，但在具体应用与开发中，其实现难度大、配置复杂，且没有提供实用的数据持久化方法。

Spring框架提供了数据持久化机智，其实现方式分为2种，一种以对象关系映射（Object Relational Mapping，ORM）工具为基础，另一种则对Java数据库链接（Java Data Base Connectivity，JDBC）类进行封装和抽象。前者能有效解决面向对象与关系数据库之间的不匹配问题，但必须依赖ORM工具。后者虽然简单易用，但不能有效解决面向对象与关系数据库之间的不匹配问题。【张宇,王映,张翔南.基于Spring的MVC框架设计与实现[J].计算机工程.2010,(4):59-62.】

1.3目标和内容

本课题的主要功能包括以下几个部分：

用户注册：用户进入平台主页点击注册按钮，输入用户名、账号（工号或手机号码）、密码、确认密码。

用户登录：用户进入平台主页点击登录按钮，输入账号（工号或手机号码）、密码。

退出登录：用户登录后，鼠标移至用户名称下，显示退出按钮，点击按钮，确定后可退出登录。

商品订购：用户在首页选择需要购买的商品进入商品界面，商品界面点击加入购物车按钮进行商品配置选择及订购。

购物车：用户在购物车界面中进行商品的配置以及购物车中的订单删除、订单查询、订单提交。

费用计算：系统根据产品不同配置以及折扣使用价格公式计算订单价格。

订单提交：系统判断订单状态是否满足提交条件，提交订单至管理员，待管理员审核。审核通过的订单会在用户资产列表中显示，审核不通过的订单则回到购物车订单列表，继续修改直至资源池内存在符合的资源，并且管理员审核通过该订单。

密码修改：用户进入管理平台-个人管理平台界面，输入原密码、新密码、确认密码即可修改密码。

重置密码：管理员进入管理平台-个人管理平台-重置密码界面，输入用户账号、确认用户账号、新密码、确认密码，即可修改、重置用户密码，进行用户管理。

资源审核：管理员进入管理平台-个人管理平台-资源审核界面，查询订单、查看订单配置，进行订单的通过、不通过操作。

资产：用户登录后，鼠标移至用户名称下，显示资产按钮，点击该按钮，用户可在资产界面中，查询订单、续订订单、变更订单、退订订单。

续订订单：用户在资产界面中，点击续订按钮，在弹框中输入续订时长，点击确定按钮即可完成订单的续订工作。

变更订单：用户在资产界面中，点击变更按钮，点击确定按钮后将跳转到购物车进行变更订单。

退订订单：用户在资产界面中，点击退订按钮，阅读退订订单的配置，确定后进行云产品的退订。

二、系统的开发工具和框架技术的介绍

2.1系统的开发工具简介

本系统主要采用以下开发工具进行开发：Sublime代码编辑器作为整个系统的开发环境，在数据存储方面采用后台Keystone内容管理系统自带的MongoDB数据库和自带的后台以及Mongoclient作为数据库可视化工具。在整个开发过程中项目版本控制方面采用gitHub服务器，并使用SourceTree作为项目依赖管理工具。

2.1.1Subline代码编辑器介绍

Sublime Text 是一个代码编辑器（Sublime Text 2是收费软件，但可以无限期试用），也是HTML和散文先进的文本编辑器。Sublime Text是由程序员Jon Skinner于2008年1月份所开发出来，它最初被设计为一个具有丰富扩展功能的Vim。

Sublime Text具有漂亮的用户界面和强大的功能，例如代码缩略图，Python的插件，代码段等。还可自定义键绑定，菜单和工具栏。Sublime Text 的主要功能包括：拼写检查，书签，完整的 Python API ， Goto 功能，即时项目切换，多选择，多窗口等等。Sublime Text 是一个跨平台的编辑器，同时支持Windows、Linux、Mac OS X等操作系统。

2.1.2Keystone内容管理系统介绍

KeystoneJS使得在node.js中搭建数据库驱动的网站、应用程序和API变得容易了。

KeystoneJS在底层使用了express.js web服务器框架，并通过对象建模框架mongoose使用MongoDB数据库。

2.1.3MongoDB数据库介绍

MongoDB是当前最流行的Nosql数据库之一。

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。

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。他支持的数据结构非常松散，是类似json的bjson格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他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。

2.1.4 gitHub介绍

gitHub是一个面向开源及私有软件项目的托管平台，因为只支持git 作为唯一的版本库格式进行托管，故名gitHub。

2.1.5SourceTree介绍

SourceTree 是 Windows 和Mac OS X 下免费的 Git 和 Hg 客户端管理工具，同时也是Mercurial和Subversion版本控制系统工具。支持创建、克隆、提交、push、pull 和合并等操作。

SourceTree拥有一个精美简洁的界面，大大简化了开发者与代码库之间的Git操作方式，这对于那些不熟悉Git命令的开发者来说非常实用。

SourceTree拥有完整的Git功能：

通过一个简单的用户界面即可使用所有的Git命令

通过一次单击，即可管理所有的Git库，无论是托管的还是本地的

通过一次单击，即可进行commit、push、pull、merge等操作

一些先进的功能，如补丁处理、rebase、shelve、cherry picking等

可以连接到你托管在Bitbucket、Stash、Microsoft TFS或GitHub中的代码库

2.2框架技术的介绍

本系统总体采用前后端分离的开发方式，前端使用Vue开发框架，后端以及数据库使用Keystone开发框架。

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。

Vue.js 的核心是一个允许采用简洁的模板语法来声明式地将数据渲染进 DOM 的系统。数据和 DOM 建立关联，所有东西都是响应式的。Vue可以分成三个部分，第一部分<template></template>中写html代码，第二部分<script></script>中写JavaScript代码，第三部分<style></style>中写css代码。methods是Vue组件的重要选项，两个方法之间通过逗号隔开，data中的数据也是如此。

数据库建立在Keystone中的models文件夹下，使用JavaScript编写的非关系型数据库。

后台部分是Keystone中的routes-api文件夹下，使用JavaScript编写的接口文件。

三、系统的需求分析和整体设计

3.1系统的需求及功能

本系统的核心功能是用户选择商品，修改配置，提交审批，查看资产，对资产中的订单进行续订、变更、退订等操作。管理员平台作为系统的辅助平台，负责维护用户数据以及订单资源的审批。

3.2系统的整体框架和模块设计

本系统分为两大平台，用户平台、管理员平台。本系统的总体框架如下图3.1所示：

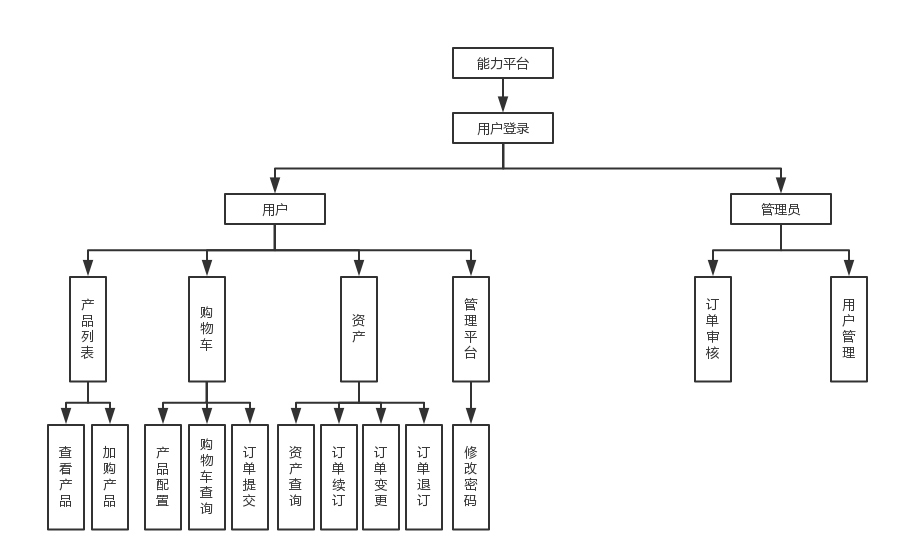


图3.1系统总体框架图

3.2.1用户平台

用户平台主要分为以下四个模块：产品模块、购物车模块、管理平台模块。其中产品模块可查看产品相关信息以及架构产品。购物车模块可查询购物车中的订单、配置订单列表中的产品以及提交订单审批。资产模块可查询资产以及续订、变更、退订订单。管理平台模块可修改用户密码。

3.2.2管理员平台

管理员平台可分为两个模块：订单资源审核与用户管理。订单资源审核主要是根据订单的配置与资源池中的资源，选择是否通过该订单的申请，通过的订单将在用户的资产中显示，不通过的订单回到购物车中，待用户修改后继续提交。用户管理主要包括用户密码的修改与重置。

3.3系统的流程设计

3.3.1用户平台

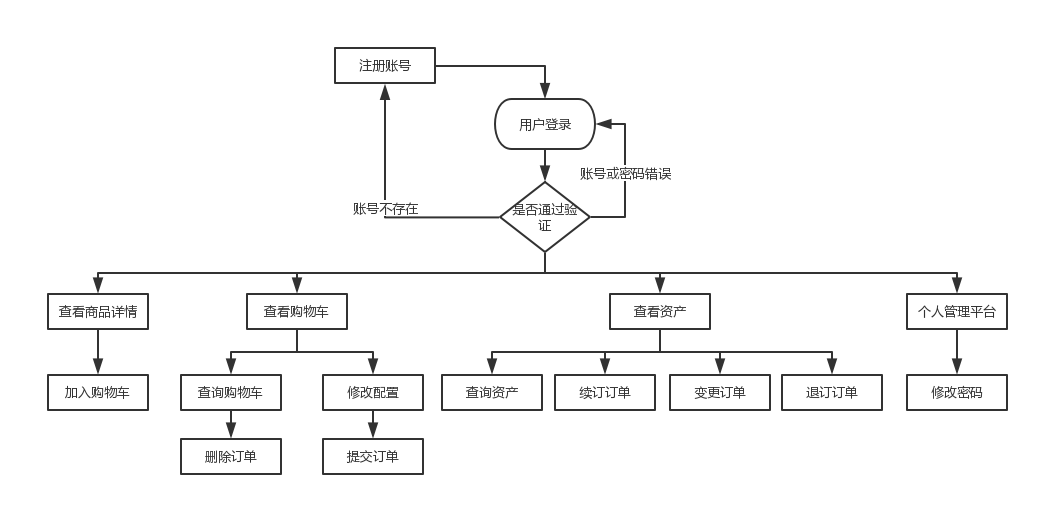


图3.2用户平台流程图

用户登录平台，如需查看产品或其他操作，需要进行账号密码登录，如没有账号可注册账号。注册账号需要用户进入平台主页点击注册按钮，输入用户名、账号（工号或手机号码）、密码、确认密码。成功登录之后，可以在主界面中选择云主机产品，查看产品详细信息。在产品列表中可点击加入购物车按钮进行产品的加购。点击加入购物车按钮后会自动跳转到购物车界面，用户可以根据不同的条件查询订单，对订单进行配置修改。确认修改好配置后，用户勾选需要提交的订单，点击提交审批按钮。若订单审批通过将在资产中显示该订单，若订单审批不通过该订单会回到购物车，待用户修改配置后继续提交，直至资源池中有该资源，并且管理员通过后将会显示在资产列表中。用户可查看以及查询自己的资产中的订单，并对资产中的订单进行续订、变更、退订操作。用户在资产界面中，点击续订按钮，在弹框中输入续订时长，点击确定按钮即可完成订单的续订工作。用户在资产界面中，点击变更按钮，点击确定按钮后将跳转到购物车进行变更订单。用户在资产界面中，点击退订按钮，阅读退订订单的配置，确定后进行云产品的退订。用户可在管理平台-个人管理平台进行修改密码，用户进入管理平台-个人管理平台界面，输入原密码、新密码、确认密码即可修改密码。用户登录后，鼠标移至用户名称下，显示退出按钮，点击按钮，确定后可退出登录。退出登录后，界面将强制跳转到首页。

3.3.2管理员平台

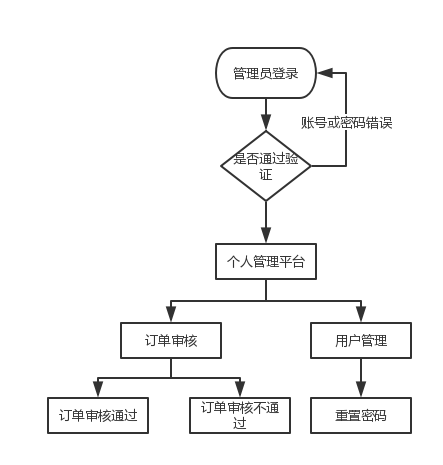


图3.3管理员平台流程图

管理员登录管理员平台之前需要进行账号密码登录，成功登录之后，点击个人管理平台，可以再管理员界面上选择相应的功能管理模块，例如重置密码、资源审核等。其中重置密码需要输入两遍用户账号（工号或手机号），密码需要输入新密码以及确认密码。资源审核时，若订单审批通过将在用户资产中显示该订单，若订单审批不通过该订单会回到用户购物车，待用户修改配置后继续提交，直至资源池中有该资源，并且管理员通过后将会显示在用户资产列表中。管理员可在管理平台-个人管理平台进行修改密码，管理员进入管理平台-个人管理平台界面，输入原密码、新密码、确认密码即可修改密码。管理员登录后，鼠标移至用户名称下，显示退出按钮，点击按钮，确定后可退出登录。退出登录后，界面将强制跳转到首页。

3.4系统数据库设计

本系统一共设计了7张数据库表：

CPU表，表名CPU，具体表结构如下表3.1所示：

表3.1CPU表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CPU：CPU表 | | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| CPU\_ID | CPU标识 | 记录CPU的主键 | String |
| CPU\_NUM | CPU数量 | 记录CPU的数量 | String |
| MEMORY\_ID | 内存标识 | 记录内存的主键 | Relationship |

其中，此表中的CPU\_ID为自增的主键标识，没有实际意义，CPU\_NUM为CPU数量。MEMORY\_ID为MEMORY的主键，是一个关系类型的外键，与CPU核数相关。

内存表，表名Memory，具体表结构如下表3.2所示：

表3.2 内存表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEMORY：内存表 | | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| MEMORY\_ID | 内存标识 | 记录内存的主键 | String |
| MEMORY\_SIZE | 内存大小 | 记录内存的大小 | String |

其中，此表中的MEMORY\_ID为自增的主键标识，没有实际意义。MEMORY\_SIZE为内存大小。

系统类型表，表名OperateSystemType，具体表结构如下表3.3所示：

表3.3 系统类型表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OperateSystemType：系统类型表 | | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| OperateSystemType\_ID | 系统类型标识 | 记录系统类型的主键 | String |
| OperateSystemType\_NAME | 系统类型名称 | 记录系统类型的名称 | String |
| OperateSystemType | 系统类型 | 记录系统的类型 | Relationship |

其中，此表中的OperateSystemType\_ID为自增的主键标识，没有实际意义，OperateSystemType\_NAME为系统类型名称，例如Windows、linux。OperateSystemType为该系统下的系统类型，与系统类型表是从属关系，是一个关系类型的外键。

系统表，表名OperateSystem，具体表结构如下表3.4所示：

表3.4 系统表

|  |  |  |  |
| --- | --- | --- | --- |
| OperateSystem：系统表 | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| OperateSystem\_ID | 系统标识 | 记录系统的主键 | String |
| OperateSystem\_NAME | 系统名称 | 记录系统的名称 | String |

其中，此表中的OperateSystem\_ID 为自增的主键标识，没有实际意义，OperateSystem\_NAME为系统名称，例如Windows7、Windows10、Ubuntu 16.04等。

硬盘类型表，表名为HardDiskType，具体表结构如下表3.5所示：

表3.5硬盘类型表

|  |  |  |  |
| --- | --- | --- | --- |
| HardDiskType：硬盘类型表 | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| HardDisk\_ID | 硬盘类型标识 | 记录硬盘类型的主键 | String |
| HardDiskType | 硬盘类型 | 记录硬盘类型的主键 | String |

其中，此表中的HardDisk\_ID为自增的主键标识，没有实际意义。HardDiskType为系统类型标识，例如SATA、SSD。

用户基本信息表，表名Customer，具体表结构如下表3.6所示：

表3.6 客户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| CUSTOMER：客户基本信息 | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| CUST\_ID | 客户标识 | 主键 | String |
| CUST\_NAME | 客户名称 | 记录客户名称 | String |
| CUST\_TELEPHONE | 客户电话 | 客户电话 | String |
| PWD | 密码 | 用户密码 | String |

其中，此表中的CUST\_ID为自增的主键标识，没有实际意义。CUST\_NAME为客户名称，CUST\_TELEPHONE为用户账号，默认用户手机号码或管理员工号。PWD为用户密码。

订单表，表名Order Item，具体表结构如下表3.7所示：

表3.7 订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDER：订单信息 | | | | |
| 字段名 | 字段中文名 | 字段描述 | 字段类型 |
| ORDER\_ID | 订单标识 | 记录订单标识 | String |
| ORDER\_CREATE\_DATE | 创建时间 | 记录创建的时间 | DT |
| period | 订购时长 | 记录订购时长 | String |
| CPU | CPU数量 | 记录CPU数量 | Relationship |
| memory | 内存大小 | 记录内存大小 | Relationship |
| operateSystemType | 系统类型 | 记录系统类型 | Relationship |
| operateSystem | 系统 | 记录系统 | Relationship |
| systemHardDiskType | 系统盘类型 | 记录系统盘类型 | Relationship |
| dataHardDiskType | 数据盘类型 | 记录数据盘类型 | Relationship |
| dataHardDiskSize | 数据盘大小 | 记录数据盘大小 | String |
| bandWidth | 带宽大小 | 记录带宽 | String |
| status | 订单状态 | 记录订单状态 | String |
| userID | 用户 | 记录订单所属用户 | Relationship |

其中，此表中的ORDER\_ID为自增的主键标识，没有实际意义。ORDER\_CREATE\_DATE 为订单创建时间，period为订单创建的时间，dataHardDiskSize为数据盘大小，bandWidth为带宽大小，status为订单状态。其余的属性为关系类型的外键，CPU为CPU数量，memory为内存大小，operateSystemType为系统类型，operateSystem为系统，systemHardDiskType为系统盘类型，dataHardDiskType为数据盘类型，userID为订单所属用户。

四、用户平台的详细设计与开发

4.1用户注册模块

4.2用户登录模块

4.3用户个人信息管理模块

4.4产品订购模块

4.5购物车模块

4.6资产模块

五、管理员平台的详细设计与开发

5.1登录模块

5.2管理模块

5.2.1用户账号管理

5.2.2订单管理

六、结论

参考文献

【欧静,赵江洪.多维情感-动作与产品形态的交互设计研究[J].包装工程.2015,(18):49-53.】

【许晓云,王一琳.扁平化——以简约为核心的产品交互设计研究[J].设计. 2015,(19):80-81.】

【李世国,华梅立,贾锐.产品设计的新模式———交互设计[J].包装工程.2007,28(4):90-92.】

【徐頓,朱广华,贾瑶.基于VueJs的WEB前端开发研究[J].科技风.2017,(14):69-69.】

【麦冬,陈涛,梁宗湾.轻量级响应式框架Vue.js应用分析[J].信息与电脑理论版.2017,(7):58-59.】

【薛峰，梁锋，徐书勋，王彪任.基于Spring MVC框架的Web研究与应用[J].合肥工业大学学报自然科学版.2012,(3):337-340.】

【谢伟,辛向阳,李世国.无意识认知交互设计探讨[J].包装工程.2015,(22):57-61.】

【张宇,王映,张翔南.基于Spring的MVC框架设计与实现[J].计算机工程.2010,(4):59-62.】

李世国,费钎. 和谐视野中的产品交互设计[J].包装工程.2009,(1):137-140.

关晓娜.交互设计时代平面设计的艺术表达[J].大舞台.2015,(6):88-89.

辛向阳.交互设计:从物理逻辑到行为逻辑[J].装饰.2015,(1):58-62 .

王玉梅,胡伟峰,汤进,李世国.产品交互设计中场景理论研究[J].包装工程. 2017,(6):76-80 .

王玉珊,李世国.情感记忆在交互设计中的价值与应用[J].包装工程.2011,(2):56-59.

刘永翔.基于产品可用性的人机界面交互设计研究[J].包装工程.2008,29(4):81-83.

叶冬冬，李世国.交互设计中的需求层次及设计策略[J].包装工程.2013,(8):75-78.