**项目视图文档**

|  |  |
| --- | --- |
| 学 院： | 软件学院 |
| 学 号： | 2020302878 |
| 姓 名： | 楚逸飞 |
| 专 业： | 软件工程 |
| 实验时间： | 2023.5.5 |
| 实验地点： | 教学西楼A202 |
| 指导教师： | 武君胜 |

西北工业大学

**2023年5月**

**项目概况与背景：**

我选择的项目是一个在线购物网站，名为“ShopOnline”。该网站旨在为用户提供一个便捷的购物平台，用户可以在该网站上浏览商品、添加到购物车、下订单等。该网站采用敏捷开发模式，以快速响应市场需求，不断迭代优化产品。我们的目标是成为顾客最喜爱的购物网站之一。

1. **用例视图：**

**描述了“ShoppingOnline”在线购物网站的各个用例以及它们之间的关系**

**1.1注册：用户可以在“ShoppingOnline”网站上进行注册，以便可以购买商品和查看订单历史记录。**

·前置条件：用户打开“ShoppingOnline”网站。

·基本流程：

用户点击网站首页上的“注册”按钮。

用户填写注册表格，包括用户名、密码、电子邮件地址和联系电话等个人信息。

用户点击“提交”按钮，将个人信息保存到系统中。

·后置条件：用户成功注册并登录到系统中。

**1.2登录：已注册的用户可以通过用户名和密码登录到“ShoppingOnline”网站。**

·前置条件：用户打开“ShoppingOnline”网站。

·基本流程：

用户点击网站首页上的“登录”按钮。

用户输入用户名和密码。

用户点击“登录”按钮，系统验证用户名和密码。

·替代流程：

用户可以选择使用第三方账号（如微信、支付宝等）登录。

·后置条件：用户成功登录到系统中。

**1.3搜索商品：用户可以在“ShoppingOnline”网站上搜索自己需要的商品。**

·前置条件：用户已登录到“ShoppingOnline”网站。

·基本流程：

用户在网站首页上的搜索框中输入商品关键字。

用户点击“搜索”按钮。

系统根据关键字查询商品库存信息，返回搜索结果。

·后置条件：用户成功搜索到商品并查看商品详细信息。

**1.4添加商品到购物车：用户可以将自己需要购买的商品添加到购物车中。**

·前置条件：用户已登录到“ShoppingOnline”网站。

·基本流程：

用户在商品详细页面中点击“加入购物车”按钮。

系统将商品添加到用户的购物车中。

·后置条件：商品成功添加到用户的购物车中。

**1.5提交订单：用户可以提交自己的购物车中的商品订单。**

·前置条件：用户已登录到“ShoppingOnline”网站并且购物车中存在商品。

·基本流程：

用户在购物车页面中选择要购买的商品。

用户点击“提交订单”按钮。

系统根据用户选择的商品生成订单，将订单保存到数据库中。

·后置条件：用户成功提交订单并可以查看订单历史记录。

**1.6用户查看订单信息**

·参与者： 普通用户

·描述： 普通用户可以在系统中查看他们的订单信息。

·前置条件： 用户已经登录系统并有有效的订单。

·基本流程：

用户进入“我的订单”页面。

系统列出用户的所有订单。

用户点击一个订单，以查看详细信息。

系统显示该订单的详细信息，包括订单编号、订单状态、订单金额、订单时 间、收货地址、商品信息等。

·替代流程：

如果用户没有订单，则系统显示“您还没有订单”的提示信息。

·后置条件： 无

·异常情况：

如果用户没有登录，则系统显示“请先登录”的提示信息。

如果用户的订单不存在或已被删除，则系统显示“订单不存在”的提示信息。

**1.7用户修改个人信息**

·参与者： 普通用户

·描述： 普通用户可以在系统中修改他们的个人信息。

·前置条件： 用户已经登录系统。

·基本流程：

用户进入“个人信息”页面。

用户点击“编辑”按钮。

系统显示当前用户的个人信息。

用户修改个人信息，如用户名、手机号码、电子邮件地址、密码等。

用户点击“保存”按钮。

系统验证用户输入的信息的合法性。

如果输入信息合法，则系统保存用户修改的个人信息，同时提示用户修改成功。

如果输入信息不合法，则系统提示用户输入正确的信息。

·替代流程：

如果输入的手机号码或电子邮件地址已经被其他用户使用，则系统提示用户输入其他的手机号码或电子邮件地址。

后置条件： 无

·异常情况：

如果用户没有登录，则系统显示“请先登录”的提示信息。

**2.逻辑视图**

**2.1 用户界面层**

用户界面层是系统的前端部分，它为用户提供了与系统交互的界面。用户界面层由以下组件构成：

·登录页面：允许用户输入用户名和密码以登录系统。

·注册页面：允许用户注册新账号。

·商品列表页面：列出所有可购买的商品。

·商品详情页面：显示商品的详细信息，包括商品名称、价格、描述和图片。

·购物车页面：列出当前用户的购物车中的商品，允许用户添加、删除和修改商品。

·订单页面：列出当前用户的订单信息。

·个人信息页面：显示当前用户的个人信息，允许用户修改个人信息。

·支付页面：用于处理用户的支付请求。

**2.2 应用逻辑层**

应用逻辑层是系统的核心部分，它包含了系统的业务逻辑和流程。应用逻辑层由以下组件构成：

·用户管理组件：用于管理系统中的用户，包括用户注册、登录、修改个人信息等功能。

·商品管理组件：用于管理系统中的商品，包括商品的查询、添加、删除、修改等功能。

·购物车管理组件：用于管理用户的购物车，包括商品的添加、删除、修改等功能。

·订单管理组件：用于管理用户的订单，包括订单的生成、查询、修改、取消等功能。

·支付管理组件：用于管理用户的支付请求，包括支付请求的验证、处理等功能。

**2.3 数据持久化层**

数据持久化层用于存储系统的数据，并提供对数据的访问接口。数据持久化层由以下组件构成：

·用户数据存储组件：用于存储系统中的用户数据，包括用户的基本信息、订单信息等。

·商品数据存储组件：用于存储系统中的商品数据，包括商品的名称、价格、描述、图片等信息。

·购物车数据存储组件：用于存储用户的购物车数据，包括购物车中的商品信息、数量等。

·订单数据存储组件：用于存储用户的订单数据，包括订单的基本信息、商品信息、价格、时间等。

·支付数据存储组件：用于存储用户的支付请求数据，包括支付的金额、时间、状态等信息。

**3.物理视图：**

ShoppingOnline系统的物理视图描述了该系统的物理组件及其之间的关系。该视图将系统划分为多个物理节点，每个节点都是独立运行的，并包含了系统的物理组件。

**3.1 节点**

**·3.1.1 Web服务器**

Web服务器节点承载了ShoppingOnline系统的Web应用程序，负责接收和响应用户的请求，并将这些请求发送到应用程序服务器处理。Web服务器还提供了一些静态资源，例如网站图标、CSS文件等。这个节点使用Apache服务器作为Web服务器软件，并运行在Linux操作系统上。

**·3.1.2应用程序服务器**

应用程序服务器节点包含了ShoppingOnline系统的业务逻辑组件，用于处理来自Web服务器的请求，并响应结果。该节点使用Java应用服务器作为运行环境，并运行在Linux操作系统上。

**·3.1.3数据库服务器**

数据库服务器节点承载了ShoppingOnline系统的数据库。该节点使用MySQL数据库管理系统作为数据库软件，并运行在Linux操作系统上。

**3.2 组件**

**·3.2.1 Web应用程序**

Web应用程序组件是ShoppingOnline系统的用户界面。它是一个基于Web的应用程序，由Web服务器节点承载并提供给用户。Web应用程序使用HTML、CSS和JavaScript等技术实现，与应用程序服务器节点通过HTTP协议进行通信。用户可以使用Web应用程序浏览商品、添加到购物车、下订单等操作。

**·3.2.2 应用程序组件**

应用程序组件包含了ShoppingOnline系统的业务逻辑，实现了用户下订单、支付、库存管理等功能。它是由应用程序服务器节点承载并运行的。该组件使用Java语言编写，并通过远程方法调用(RMI)与Web应用程序进行通信。

**·3.2.3数据库**

数据库组件存储了ShoppingOnline系统的所有数据，包括商品信息、用户信息、订单信息等。该组件使用MySQL数据库管理系统进行管理，并在数据库服务器节点上运行。

**·3.2.4关系**

Web应用程序与应用程序组件之间通过RMI进行通信，应用程序组件使用Java语言编写并运行在应用程序服务器节点上。

应用程序组件与数据库之间通过JDBC进行通信，数据库使用MySQL数据库管理系统，并运行在数据库服务器节点上。

**4.部署视图**

**4.1 物理硬件**

·Web 服务器：使用 Linux 操作系统，配备 8 核 CPU、32GB 内存、1TB 存储空间和 10 GbE 网卡。

·数据库服务器：使用 Windows 操作系统，配备 16 核 CPU、128GB 内存、5TB 存储空间和 10 GbE 网卡。

·缓存服务器：使用 Linux 操作系统，配备 4 核 CPU、16GB 内存、500GB 存储空间和 10 GbE 网卡。

**4.2 软件资源**

·Web 应用程序：使用 Java 语言编写，使用 Spring Boot 框架和 Thymeleaf 模板引擎。Web 应用程序使用 Tomcat 作为 Web 服务器，并且可以在多个实例上运行，每个实例都有自己的 JVM。

·数据库服务器：使用 Microsoft SQL Server 数据库管理系统，采用主从复制的方式进行数据备份和恢复。

·缓存服务器：使用 Redis 内存数据库作为缓存服务器。

**4.3 部署拓扑**

·Web 应用程序：部署在 2 台 Web 服务器上，每台服务器上部署 4 个实例。

·数据库服务器：部署在 2 台数据库服务器上，每台服务器都运行一个主数据库和一个从数据库。

·缓存服务器：部署在 2 台缓存服务器上，每台服务器运行一个 Redis 实例。

大致拓扑图如下：

┌──────────┐

┌──────────┐ ┌──────────┐ │Web Server│

│Web Server│ │Web Server│ ├──────────┤

│Instance 1│ │Instance 1│ │Instance 1│

├──────────┤ ├──────────┤ ├──────────┤

│Instance 2│ │Instance 2│ │Instance 2│

├──────────┤ ├──────────┤ ├──────────┤

│Instance 3│ │Instance 3│ │Instance

1. **运行时视图**

ShoppingOnline的运行时视图描述了在系统运行时期间，不同的组件之间是如何相互协作的，以及它们在系统中如何进行交互。这个视图通常用于描述系统的运行时架构。

在ShoppingOnline项目中，主要的组件包括客户端、Web服务器、应用服务器、数据库服务器等。这些组件在系统运行时通过不同的接口进行通信。

**5.1 登录场景**

用户在浏览器中输入网站地址后，网站通过Web服务器接受用户请求。Web服务器将请求转发到应用服务器上，应用服务器通过数据库服务器验证用户的用户名和密码。如果验证成功，应用服务器返回登录成功的信息，并将用户信息保存在Session中。Web服务器接收到应用服务器的响应后，将响应返回给用户的浏览器。用户成功登录后，可以浏览商品、添加到购物车等操作。

**5.2 下订单场景**

用户在购物车页面点击“结算”按钮后，Web服务器接收到用户请求，将请求转发给应用服务器。应用服务器从Session中获取用户信息和购物车信息，生成订单并将订单信息存储到数据库中。应用服务器返回下单成功的信息，Web服务器将响应返回给用户的浏览器。用户可以在订单页面查看订单信息。

ShoppingOnline还包括其他的一些组件，如支付网关、物流系统等，这些组件在系统运行时期间也将参与到系统中，以实现在线购物的功能。在系统的运行过程中，客户端向Web服务器发送请求，Web服务器将请求转发给应用服务器进行处理，并且将处理结果返回给客户端。应用服务器通过数据库服务器来存储和检索数据，并与其他的组件进行交互以完成业务流程。支付网关将负责处理支付事务，物流系统将负责处理订单的配送。所有的组件都通过网络接口进行通信，实现了系统的高效运行和数据的安全性。