****

**《ERP SD系统研发与设计报告》**

**（2025）**

**专 业：** **信息管理与信息系统、物流管理**

**指导老师： 吴冰**

**小 组： 第1组**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **学号** | **姓名** | **学号** |
| **林哲扬** | **2251357** | **冷芸婷** | **2252997** |
| **陆心语** | **2251234** | **田晓琪** | **2252598** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**日 期： 2025年7月 10日**

**小组具体分工**

**系统逻辑模型设计报告：**

1. 林哲扬
2. 陆心语

**系统分析设计报告：**

1. 田晓琪

2. 冷芸婷

**代码开发：**

1.

**工作量：小组共10名成员，每人约10%，总计100％**

目录

[1 信息系统设计的目标 6](#_Toc203085088)

[1.1 提高效率 6](#_Toc203085089)

[1.2 提高关联性 6](#_Toc203085090)

[1.3 提高灵活性和适应性 6](#_Toc203085091)

[1.4 提高用户友好性 7](#_Toc203085092)

[1.5 提高安全性与准确性 7](#_Toc203085093)

[2 SD 系统平台配置报告 7](#_Toc203085094)

[2.1网络结构体系 7](#_Toc203085095)

[2.1.1网络拓扑结构 7](#_Toc203085096)

[2.1.2系统架构 8](#_Toc203085097)

[2.1.3传输介质 8](#_Toc203085098)

[2.1.4组网方式 8](#_Toc203085099)

[2.1.5网络设备 8](#_Toc203085100)

[2.1.6传输协议 8](#_Toc203085101)

[2.1.7网络操作系统 9](#_Toc203085102)

[2.2硬件配置 9](#_Toc203085103)

[2.2.1 服务器 9](#_Toc203085104)

[2.2.2 工作站 10](#_Toc203085105)

[2.2.3外部设备 10](#_Toc203085106)

[2.3软件选择 10](#_Toc203085107)

[2.3.1操作系统 10](#_Toc203085108)

[2.3.2数据库管理系统 10](#_Toc203085109)

[2.3.3开发平台与工具 10](#_Toc203085110)

[2.3.4Web框架 11](#_Toc203085111)

[2.3.5前后端数据交互 11](#_Toc203085112)

[2.4云计算支持（可选扩展） 11](#_Toc203085113)

[2.5配置说明 11](#_Toc203085114)

[3 数据库设置清单及说明 12](#_Toc203085115)

[3.1 数据库结构设计 12](#_Toc203085116)

[3.1.1 用户模块（EUser） 12](#_Toc203085117)

[3.1.2 客户模块（Customer） 13](#_Toc203085118)

[3.1.3 库存管理模块 (Stock, MaterialItem, StockHistory): 13](#_Toc203085119)

[3.1.4 销售订单申请模模块(SalesRequisition, RequisitionItem) 15](#_Toc203085120)

[3.1.5 报价管理模块(Quotation) 16](#_Toc203085121)

[3.1.6 销售订单模块(SalesOrder, OrderItem) 17](#_Toc203085122)

[3.1.7 发货管理模块（GoodDelivery） 18](#_Toc203085123)

[3.1.8 发票管理模块(Invoice) 19](#_Toc203085124)

[3.1.9 财务记账模块(Account, AccountDetail) 19](#_Toc203085125)

[3.2数据库优点 20](#_Toc203085126)

[3.2.1 关系管理和业务逻辑体现 20](#_Toc203085127)

[4代码赋值清单 21](#_Toc203085128)

[4.1公司代码（CompanyCode） 21](#_Toc203085129)

[4.2国家代码（CountryCode） 21](#_Toc203085130)

[4.3货币代码（CurrencyCode） 22](#_Toc203085131)

[4.4语言代码（LanguageCode） 22](#_Toc203085132)

[4.5计量单位代码（MeaunitCode） 23](#_Toc203085133)

[4.6工厂代码（PlantCode） 24](#_Toc203085134)

[4.7销售组织代码 24](#_Toc203085135)

[4.8付款方式代码（TPtyprCode） 25](#_Toc203085136)

[4.9结论 25](#_Toc203085137)

[5 输入输出设计 25](#_Toc203085138)

[5.1 输入设计 25](#_Toc203085139)

[5.1.1 输入格式设计 26](#_Toc203085140)

[5.1.2 输入格式的正确性检验 27](#_Toc203085141)

[5.2 输出设计 27](#_Toc203085142)

[5.3 输入输出的异常处理 29](#_Toc203085143)

[6 用户界面设计 30](#_Toc203085144)

[6.1用户界面设计原则 30](#_Toc203085145)

[6.2 登录注册界面 30](#_Toc203085146)

[6.3 主页 30](#_Toc203085147)

[6.4客户管理系统界面 31](#_Toc203085148)

[6.2.1客户、联系人的创建与维护 31](#_Toc203085149)

[6.2.2关系的创建 31](#_Toc203085150)

[6.3发货管理系统界面 31](#_Toc203085151)

[6.3.1 物料出库登记 31](#_Toc203085152)

[6.3.2 物料库存查询 32](#_Toc203085153)

[6.4销售订单管理系统界面 32](#_Toc203085154)

[6.5 财务管理系统界面 32](#_Toc203085155)

[6.6 文档流 33](#_Toc203085156)

[7 结构图及模块说明书 33](#_Toc203085157)

[7.1结构图 33](#_Toc203085158)

[7.2 模块说明书 33](#_Toc203085159)

[8 信息系统安全设计​ 37](#_Toc203085160)

[8.1 概述​ 37](#_Toc203085161)

[8.2 人员安全管理​ 37](#_Toc203085162)

[8.3 用户标识与认证​ 38](#_Toc203085163)

[8.4 数据完整性与有效性控制​ 39](#_Toc203085164)

[8.4.1数据输入与验证 39](#_Toc203085165)

[8.4.2数据存储与处理 39](#_Toc203085166)

[8.4.3数据输出与安全 39](#_Toc203085167)

[8.5 逻辑访问控制​ 40](#_Toc203085168)

[8.6 审计与跟踪​ 40](#_Toc203085169)

[9 SD 系统实施费用估算报告 41](#_Toc203085170)

[9.1软件开发成本 41](#_Toc203085171)

[9.2硬件设备采购 42](#_Toc203085172)

[9.3人员培训成本 43](#_Toc203085173)

[9.4系统维护和支持成本 43](#_Toc203085174)

[9.5其他间接成本 43](#_Toc203085175)

[9.6总费用估算 43](#_Toc203085176)

[10参考文献 44](#_Toc203085177)

# 1 信息系统设计的目标

在本小学期的系统设计实践中，我们的核心任务是构建一个简化版的SD系统，重点满足企业销售管理领域对核心业务处理的功能需求。该系统以提升销售管理效能为目标，通过整合订单处理、库存监控等关键业务环节，有效降低了人工成本与跨部门协作成本。系统通过信息流的实时更新与全流程共享优化了整体业务运转效率。我们的设计不仅着眼于当前的业务需求，还为未来的扩展和升级预留了足够的空间。以下将详细阐述本系统的具体设计目标：

## 1.1 提高效率

我们设计了4个核心模块，分别是：客户管理（Create New Customer, Create Contact Person for Customer，Create BP relationship）、订单管理（Create Customer Inquiry，Create Customer Quotation，Create Sales Order Referencing a Quotation）、发货管理（Start Delivery Process，Post Goods Issue）、财务管理(Create Invoice for Customer，Post Receipt of Customer Payment，Review Document Flow)。

通过集成这些功能模块，能够大幅缩减手工操作所耗费的时间与人力投入，有效降低人为失误出现的概率。此外，该系统具备对各模块所涵盖的核心业务数据实施创建、修改及查询操作的功能，保障用户可借助统一平台高效处理日常事务，进而显著提升企业整体业务运作效率。

## 1.2 提高关联性

在系统设计环节，保障各模块间的关联性以及数据的一致性，是整个设计工作的要点，会对系统内部的逻辑架构以及用户的使用操作体验产生影响。

在本系统中，在订单管理中会自动调用客户的相关信息；而在发货管理模块中，订单又会自动与对应的发货记录建立关联。除此之外，文档流展示模块会全程融入整个业务流程，为用户提供清晰明了的业务脉络以及便捷的数据追溯途径。这种模式能够切实保障业务流程之间的顺畅衔接，维护各模块间数据的一致性。

## 1.3 提高灵活性和适应性

业务需求通常是动态变化的，这就要求信息系统具备充分的扩展性、灵活性和易维护性。在设计SD系统时，我们运用了模块化的架构理念，将系统拆分为多个独立的功能模块。各个功能模块能够独立开展开发、测试、更新以及部署工作，且不会对系统的其他模块产生干扰。这种模块化的设计方式，极大地提升了系统的可维护性。同时，也为系统未来的功能拓展与个性化定制提供了便利条件。例如，企业在拓展海外市场过程中，发现现有的客户管理模块无法满足对海外客户特殊信息（如海外税务登记号、当地合规认证信息等）的记录和管理需求。借助SD系统的模块化设计，开发团队只需在客户管理模块中独立进行功能扩展，添加相应的字段和信息录入界面，就能轻松实现对海外客户特殊信息的管理。又如，随着企业业务模式的创新，推出了一种全新的预售订单业务。在订单管理模块中，开发人员可以通过独立的模块开发，新增预售订单的处理逻辑，包括预售订单的创建、状态跟踪、与正式销售订单的转换等，而不会影响发货管理和财务管理模块的正常运行。这种灵活的设计使得SD系统能够快速响应业务需求的变化，确保系统始终与企业的业务发展保持同步，保持系统的先进性和竞争力，为企业的持续发展提供有力的信息技术支持。

## 1.4 提高用户友好性

在界面设计方面，我们运用了一致性设计原则，保障不同模块在操作逻辑与视觉呈现风格上达成统一，期望能提升用户友好性，助力用户迅速熟悉系统操作，降低用户操作失误的概率。除此之外我们精简了用户的操作流程，在保证系统功能完备性的基础上，使用户能够以更为轻松、高效的方式完成各项工作任务。系统界面的响应效率以及交互设计均经过了深度优化。即便在面对大规模数据处理以及复杂操作场景时，系统依旧能够为用户提供流畅、无卡顿的使用体验。

## 1.5 提高安全性与准确性

在系统设计阶段，保障数据的安全性与准确性是至关重要的目标之一。鉴于系统需存储大量敏感的商业数据以及用户信息，防止这些关键数据遭受未授权的访问与恶意篡改是系统安全防护的核心任务。为此，我们所设计的系统综合运用了多重身份验证和权限管理等多种安全防护机制。同时定期开展全面的安全检查，对系统的操作记录、访问日志等进行审查，及时发现潜在的安全隐患。同时，及时修复发现的系统漏洞，确保系统保持安全稳定的运行状态。在数据质量保障方面，系统在数据输入和处理环节设置了严格的验证和校验机制。在数据输入时，对数据的格式、类型、范围等进行实时检查，确保输入数据的合法性和规范性。在数据处理过程中，通过数据比对等方式，保证数据的一致性和完整性，有效减少因数据错误引发的业务风险，为企业的决策提供可靠的数据支持。

综合而言，本系统设计的核心目标在于打造一套简化操作、易于使用且具有良好扩展性的ERP解决方案。该方案不仅能够充分满足企业在销售方面的核心需求，提升效率，还为未来信息系统的升级和拓展奠定基础，助力企业在数字化时代实现可持续发展。

# 2 SD 系统平台配置报告

## 2.1网络结构体系

### 2.1.1网络拓扑结构

采用星型拓扑结构，以核心交换机为中心节点，连接应用服务器、数据库服务器及各部门工作站。该结构符合 Global Bike（GBI）公司销售、仓库、财务等多部门集中协作的场景，便于故障排查和节点扩展，与案例中销售代表、仓库员工、会计等多角色协同处理订单的业务模式匹配。

### 2.1.2系统架构

本信息系统采用B/S（Browser/Server）架构。这种架构的优势在于简化了客户端的复杂性，所有的业务逻辑和数据处理都集中在服务器端进行，客户端通过浏览器访问系统并与服务器进行交互。整个系统由以下几个关键组件构成：

**·**客户端：用户通过浏览器访问系统，无需安装任何额外的软件。浏览器负责向服务器发送请求并展示返回的数据。

**·**服务器端：负责处理来自客户端的请求，执行业务逻辑，并访问数据库进行数据的存储与检索。服务器端的核心功能包括用户认证、数据处理、业务规则执行以及页面的动态生成。

由于B/S架构的特点，系统的可扩展性较强，后续可以根据需要对服务器端进行升级或扩展，而无需对客户端进行大规模的调整。

### 2.1.3传输介质

主要采用千兆以太网（Cat 6）网线作为物理传输介质，以保证高效的数据传输速率与稳定性。局部区域可辅以Wi-Fi 6无线网络，便于移动设备访问。

### 2.1.4组网方式

**·**内部组网：基于局域网（LAN）实现各部门互联互通，支持销售订单创建后仓库实时查看库存、财务同步生成发票等跨部门流程。

**·**远程接入：通过 VP支持异地办公，如销售人员在外处理客户询价等场景。

### 2.1.5网络设备

**·**核心交换机：1 台企业级千兆交换机（支持 VLAN 划分），隔离销售、仓库、财务网段，保障数据安全（如客户财务信息与库存数据分离）。

**·**接入交换机：3 台百兆交换机，分别部署于销售部、仓库、财务部，连接各部门工作站。

**·**路由器：1 台支持 VPN 的企业级路由器，负责内外网数据转发。

**·**防火墙：1 台下一代防火墙，过滤非法访问，保护客户主数据、订单金额等敏感信息。

### 2.1.6传输协议

在客户端与服务器端，或各业务模块之间进行数据交互的过程中，系统采用了以下几种主流的网络通信协议，以保障通信的稳定性、安全性与互操作性。

**·HTTP/HTTPS 协议**

系统界面基于 Web 技术构建，主要操作通过浏览器发起，因而数据传输使用超文本传输协议HTTP和 安全超文本传输协议HTTPS。HTTP 协议为 Web 应用提供基本的数据传输通道，而 HTTPS 则在其基础上增加了 SSL/TLS 加密机制，有效防止数据在传输过程中的窃听、伪造和篡改，确保用户操作、客户信息、发票数据等敏感信息的安全。该协议对于需要用户身份认证、付款记录传输等高安全性场景尤为关键，是本系统信息安全体系的重要组成部分。

**·TCP/IP 协议**

所有网络通信均建立在 TCP/IP 协议栈之上。传输控制协议TCP确保数据包可靠、有序地传输，即便在网络波动或延迟较大时也能保证数据完整性；网际协议IP负责将数据正确路由至目标设备。借助 TCP/IP 协议，本系统实现了跨模块、跨服务器的数据联通，是分布式系统部署的技术基础。

**·SSL VPN 协议**

系统支持通过安全套接层虚拟专用网络 SSL VPN实现远程接入，便于销售人员或管理员在异地访问 SAP S/4HANA 系统后台。该协议结合了 HTTPS 的加密能力与 VPN 的网络隔离机制，为远程办公或异地数据审核提供安全保障，有效防止非授权访问。

**·LDAP 协议**

系统采用 轻量目录访问协议LDAP进行用户身份认证与访问控制。通过 LDAP 协议，系统能够对接企业现有的目录服务（如 Microsoft Active Directory），实现用户角色统一管理、权限分级控制以及登录日志跟踪，增强系统的可控性与可审计性。

通过上述多层级协议体系的协同配合，系统在保障**数据传输效率**的同时，也实现了**信息安全、身份认证和远程接入**等多方面的业务需求，为整个SD系统的稳定运行提供了强有力的网络通信支持。

### 2.1.7网络操作系统

采用 Windows Server 2019，支持 Active Directory 域管理，可根据职员在业务中的角色分配操作权限，确保数据访问安全。

## 2.2硬件配置

### 2.2.1 服务器

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 设备类型 | 机型 | 性能指标 | 数量 | 所属部门 | 用途 |
| 应用服务器 | Dell PowerEdge R650 | 12 核 CPU，32GB 内存，512GB SSD | 1 台 | 信息技术部 | 运行客户管理、订单管理等 SD 模块，支持 50 并发用户操作 |
| 数据库服务器 | IBM System x3650 | 16 核 CPU，64GB 内存，2TB RAID5 存储 | 1 台 | 信息技术部 | 存储客户主数据、订单记录、库存信息等，支持数据实时读写 |
| 备份服务器 | HP ProLiant DL380 | 8 核 CPU，16GB 内存，1TB 存储 | 1 台 | 信息技术部 | 每日自动备份业务数据，防止订单、付款记录丢失 |

### 2.2.2 工作站

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 设备类型 | 机型 | 性能指标 | 数量 | 所属部门 | 用途 |
| 销售工作站 | Lenovo ThinkCentre M900 | 6 核 CPU，8GB 内存，256GB SSD | 4 台 | 销售部 | 执行客户创建、询价处理、订单录入等操作 |
| 仓库工作站 | Dell OptiPlex 7080 | 6 核 CPU，8GB 内存，256GB SSD | 2 台 | 仓库 | 处理发货流程、库存检查、货物出库等任务 |
| 财务工作站 | HP EliteDesk 800 G6 | 6 核 CPU，8GB 内存，256GB SSD | 2 台 | 财务部 | 负责发票创建、付款记录、财务对账等工作 |

### 2.2.3外部设备

**·**激光打印机：2 台（销售部、财务部各 1 台），用于打印订单、发票等单据（如案例中的客户报价单、付款凭证）。

**·**条码扫描器：2 台（仓库），配合发货管理模块快速录入出库商品信息，提升发货效率。

**·**UPS 不间断电源：1 套，保障服务器在断电时完成数据保存，避免订单处理中断。

## 2.3软件选择

### 2.3.1操作系统

在本项目中，操作系统的选择为Windows系列。开发环境和服务器均采用Windows操作系统，以确保一致的兼容性和便捷的管理。

·工作站操作系统: Windows 10 或 Windows 11

·服务器操作系统: Windows Server，配置为2核CPU和1GB内存，适用于轻量级应用的开发和部署。

### 2.3.2数据库管理系统

为了满足数据存储和管理的需求，本项目选用了MySQL数据库管理系统。MySQL以其开源、稳定、高效的特点广泛应用于Web应用开发中，能够满足本项目中数据存储、查询和管理的需求。

### 2.3.3开发平台与工具

开发团队使用了PyCharm作为主要的开发平台和IDE（集成开发环境）。PyCharm为Python开发提供了强大的支持，包括代码补全、调试、测试集成和版本控制，极大提高了开发效率。

### 2.3.4Web框架

在Web开发中，采用了Django作为Web框架。Django是一个高级Python Web框架，具有快速开发、高效、安全的特点，适用于构建复杂的数据驱动应用。

### 2.3.5前后端数据交互

在前后端进行数据交互时，主要有GET和POST两种方式。对于数据交互的手段，我们主要采用的都是Django框架所自带的方法。例如在数据上传时，由于Django 提供的ModelForm极大地简化了这一过程，所以我们主要通过由ModelForm实例化的form对象进行。而对于数据的取出，我们主要使用的是objects.filter的方法，代替了SQL略显繁杂的查询语句，从而能更方便地得到数据，并将其放在python文件中，利用python功能强大的各种库进行处理。

## 2.4云计算支持（可选扩展）

为适配课程项目的资源限制，可选用阿里云轻量应用服务器：

**·**云服务器：2 核 4GB 配置，部署应用程序，支持 10 人同时在线演示。

**·**云数据库：RDS for MySQL，免去本地硬件维护，保障数据可用性。

## 2.5配置说明

本方案基于课程项目需求及案例业务场景设计，核心考虑以下因素：

性能匹配：服务器配置满足 10 人小组（2 名设计 + 6 名开发 + 2 名演示）的并发操作需求，响应时间≤2 秒，适配订单创建、库存查询等高频操作。

成本控制：选用开源软件（MySQL、Tomcat）和中端硬件，平衡性能与预算，符合课程项目的资源约束。

角色适配：按案例中的部门分工（销售、仓库、财务）配置工作站，确保各角色可独立完成对应模块操作（如仓库人员处理发货，财务人员管理发票）。

可扩展性：支持后续添加移动客户端（如销售人员用平板录入订单），适配案例中 “客户订单全流程管理” 的扩展需求。

通过以上配置，系统可稳定运行客户管理、订单管理、发货管理、财务管理等核心模块，满足课程项目的演示与开发要求。

# 3 数据库设置清单及说明

我们的这个系统使用了Django框架,采用了模型-视图-控制器（MVC）架构。代码定义了多个模型（Models）,这些模型对应数据库中的表,描述了系统的数据结构。

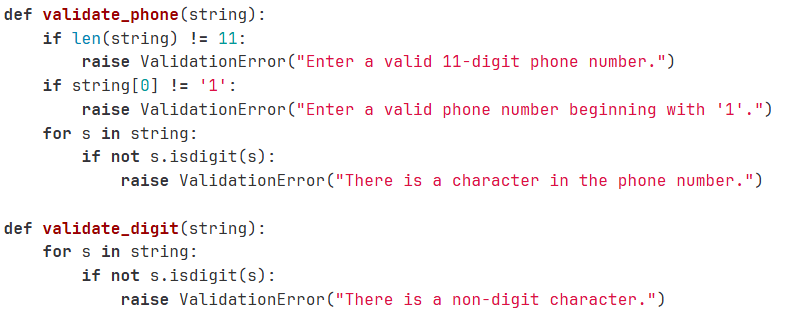
## 3.1 数据库结构设计

数据库共有13张表，可依据相关的功能分为9个主要模块，主要包括了用户管理（EUser模型）、客户管理模块（Customer）、物料主数据模块（Material）（删）、库存管理模块（Stock, MaterialItem, StockHistory）、销售订单申请模块（SalesRequisition, RequisitionItem）、报价管理模块（Quotation）、销售订单模块（SalesOrder, OrderItem）、发货管理模块（GoodDelivery）、发票管理模块（Invoice）、财务记账模块（Account, AccountDetail）等共9个主要模块。后期打算加入使用记录和收藏夹等的支持性模块。

### 3.1.1 用户模块（EUser）

该模块继承了Django的User模型，扩展了用户属性（如sector, phone, security questions等），主要相关的功能有用户注册、登录，查看和更新用户个人信息，在需要密码重置等额外安全验证时进行安全问题验证。

用于验证电话号码和邮政编码的正确性的自定义验证器部分代码：



用户模型的具体代码：



主要包含以下字段：

Uid，用户ID，自增的主键字段。

Sector，用户所属部门，最大长度为20个字符的字符串字段。

Phone，用户手机号，11位字符，使用自定义的手机号验证函数进行验证。

question1，密保问题1，最大长度为50个字符的字符串字段。

answer1，密保答案1，最大长度为50个字符的字符串字段。

question2，密保问题2，最大长度为50个字符的字符串字段。

answer2，密保答案2，最大长度为50个字符的字符串字段。

### 3.1.2 客户模块（Customer）

该模块定义了客户的基本信息，如cname（客户名称），city（城市），address（地址），postcode（邮政编码），country（国家），以及与客户相关的其他信息。客户与用户（EUser）之间存在一对多的关系，即一个用户可以创建多个客户。

主要相关的功能有客户信息的创建、更新，根据不同条件搜索和查看客户信息，记录和更新客户的表现评分。

客户模型的主要代码：



主要包含以下字段：

Cid，客户ID，自增的主键字段。

Euser，关联到用户模型的外键字段，如果用户被删除，客户记录也将被删除。

Cname，客户名称，最大长度为20个字符的字符串字段，必须唯一。

City，客户所在城市，最大长度为20个字符的字符串字段。

Address，客户地址，最大长度为50个字符的字符串字段，可以为空。

Postcode，邮政编码，最大长度为6个字符的字符串字段，使用自定义的数字验证函数进行验证。

Country，客户国家，最大长度为10个字符的字符串字段。

Language，客户语言，最大长度为10个字符的字符串字段。

glAcount，客户的总账科目，最大长度为6个字符的字符串字段。

phone，客户电话，11位字符，使用自定义的手机号验证函数进行验证，可以为空。

fax，客户传真，最大长度为10个字符的字符串字段，使用自定义的数字验证函数进行验证，可以为空。

tpType，交易类型，最大长度为1个字符的字符串字段，可以为空。

companyCode，公司代码，最大长度为4个字符的字符串字段。

sOrg，销售组织代码，最大长度为4个字符的字符串字段。

currency，客户货币，最大长度为10个字符的字符串字段，可以为空。

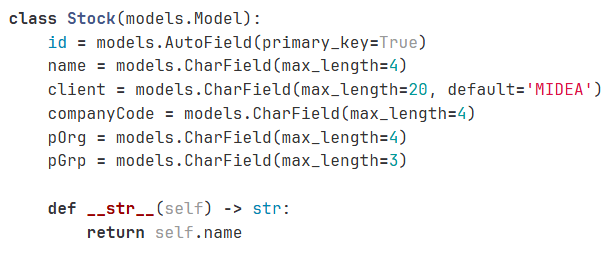
### 3.1.3 库存管理模块 (Stock, MaterialItem, StockHistory):

这组模块协同工作,管理系统中的库存信息和变动历史。主要相关的功能包括查看任意物料在任意库存地点的库存状况，根据业务操作(发货)更新库存数量，查看库存变动历史,便于审计和问题追踪和区分不同状态的库存,如可用库存、质检库存等。

（1）Stock模型

该模块描述了库存的基本信息，包括库存名称name，所属公司companyCode，销售组织sOrg等。

库存模型（stock）代码：



主要包含以下字段：

Id，库存ID，自增的主键字段。

Name，库存地点名称，最大长度为4个字符的字符串字段。

Client，客户名称，最大长度为20个字符的字符串字段，默认值为“JiTong”。

companyCode，公司代码，最大长度为4个字符的字符串字段。

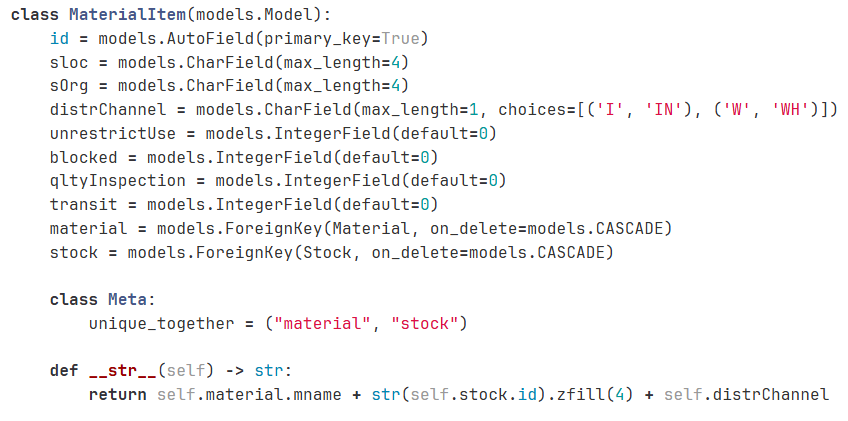
sOrg，销售组织代码，最大长度为4个字符的字符串字段。

sGrp，销售组代码，最大长度为3个字符的字符串字段。

（2）物料库存项模型（MaterialItem）

该模块表示特定库存中的特定物料，管理库存中物料的不同状态，如unrestrictUse（可用库存），blocked（冻结库存），qltyInspection（质检库存）等。与Material和Stock模型存在外键关系。

物料库存项模型代码：



主要包含以下字段：

id物料项目的ID，自增的主键字段。

sloc，存储位置，字符字段，最大长度为4个字符。

sOrg，销售组织，字符字段，最大长度为4个字符。

distrChannel，分销渠道，字符字段，最大长度为1个字符，选择‘I’表示内部(IN)，选择‘W’表示仓库(WH)。

unrestrictUse，无限制使用数量，整数字段，默认值为0。

blocked，被锁定的数量，整数字段，默认值为0。

qltyInspection，质量检查数量，整数字段，默认值为0。

transit，在途库存数量，整数字段，默认值为0。

material，外键，关联到 `Material` 模型。表示此物料项目所属的物料。

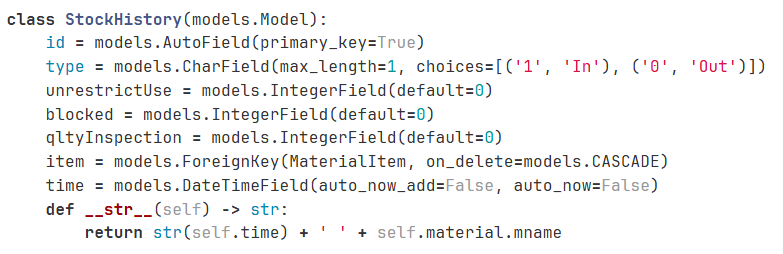
stock，外键，关联到 `Stock` 模型。表示此物料项目所属的库存。

class Meta，元类定义，确保物料和库存的组合是唯一的。

（3）库存历史模型（StockHistory）

该模块记录了库存中物料的进出库历史，包含库存类型type（入库或出库），操作时间time等信息。与MaterialItem模型关联。

库存历史模型代码：



主要包含以下字段：

Id，库存历史记录的ID，自增的主键字段。

type，记录类型，表示库存是入库（'1'）还是出库（'0'）。最大长度为1个字符，使用选择项来限定可能的值。

unrestrictUse，可用库存数量，整数字段，默认值为0。

blocked，被锁定的库存数量，整数字段，默认值为0。

qltyInspection，质量检验中的库存数量，整数字段，默认值为0。

item，关联到物料项模型的外键字段，如果物料项被删除，库存历史记录也将被删除。

time，记录发生的时间，日期时间字段。auto\_now\_add，表示不会自动设置为创建时的时间，`auto\_now=False`表示不会自动设置为每次保存时的时间。

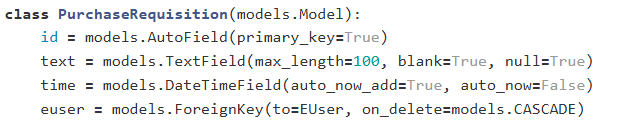
### 3.1.4 销售订单申请模模块(SalesRequisition, RequisitionItem)

主要相关的功能包括创建销售订单申请，在销售订单申请中添加多个物料项目，跟踪每个申请项目的处理状态，允许用户查询和修改未处理的销售订单申请等。

（1）销售订单申请模型（SalesRequisition）

该模块定义了销售订单申请的基本信息，包括申请内容text，申请时间time，以及与用户的关联关系。

销售订单申请模型代码：



主要包含以下字段：

validate id，销售订单申请的唯一标识，自增主键。

text，销售订单申请的描述性文本，可选字段，最大长度为100个字符。

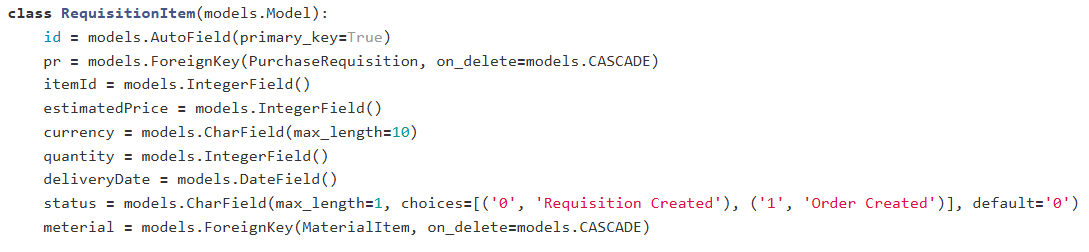
time，创建销售订单申请的时间，自动设置为创建时的时间。

euser，外键，关联到 `EUser` 模型，表示提交此销售订单申请的用户。如果用户被删除，相应的销售订单申请也会被删除。、

（2）销售订单申请项模型（RequisitionItem）

该模块描述销售订单申请中的具体物品信息，包括预估价格estimatedPrice，数量quantity，交货日期deliveryDate等。与SalesRequisition模型关联。

销售订单申请项模型代码：



主要包含以下字段：

Id，销售订单申请项的唯一标识，自增主键。

pr，外键，关联到 `SalesRequisition` 模型，表示该项属于哪个销售订单申请。

itemId，销售订单项的项目ID，用于区分同一销售订单申请中的不同项目。

estimatedPrice，预估的销售订单价格，以整数形式表示。

currency，货币类型，字符字段，最大长度为10个字符。

quantity，销售订单数量，以整数形式表示。

deliveryDate，交货日期，日期字段，表示预期的交货时间。

status，销售订单申请项的状态，'0'表示销售订单申请已创建，'1'表示订单已创建。

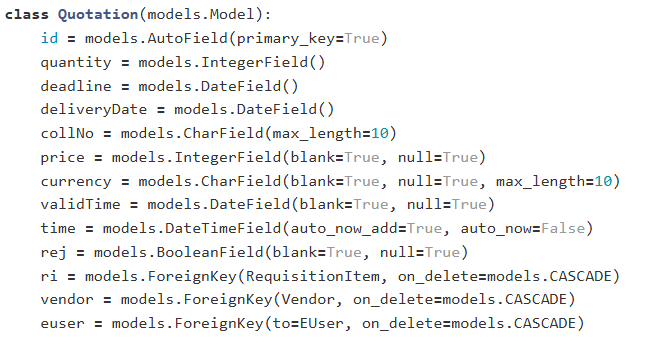
meterial，外键，关联到 `MaterialItem` 模型，表示该销售订单项涉及的具体物料项目。

### 3.1.5 报价管理模块(Quotation)

该模块管理客户对销售订单申请的报价信息。记录客户针对销售订单申请所提供的报价信息，包括数量quantity，价格price，货币currency等。与Customer（客户）和RequisitionItem（销售订单申请项）模型关联。

主要相关的功能有记录客户的报价信息，比较多个客户的报价，审核和处理客户报价，查看历史报价信息等。

报价管理模块代码：



主要包含以下字段：

Id，报价的唯一标识，自增主键。

quantity，报价的数量，以整数形式表示。

deadline，报价的截止日期，日期字段。

deliveryDate，预计交货日期，日期字段。

collNo，收集编号，字符字段，最大长度为10个字符。

price，报价的价格，以整数形式表示，可以为空。

currency，货币类型，字符字段，最大长度为10个字符，可以为空。

validTime，报价有效期，日期字段，可以为空。

time，创建报价的时间，自动设置为创建时的时间。

rej，是否被拒绝的标志，布尔字段，可以为空。

ri，外键，关联到 `RequisitionItem` 模型，表示该报价与哪个销售订单申请项关联。

customer，外键，关联到 `Customer` 模型，表示提供此报价的客户。

euser，外键，关联到 `EUser` 模型，表示提交此报价的用户。

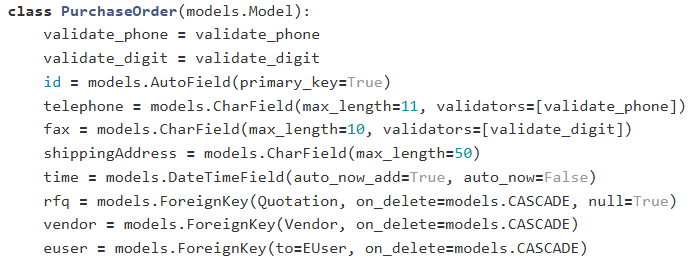
### 3.1.6 销售订单模块(SalesOrder, OrderItem)

这组模块管理正式的销售订单信息。主要相关的功能包括基于已批准的报价创建正式销售订单，跟踪每个订单项目的处理状态，允许在特定条件下修改订单，记录和更新客户在每个订单项上的表现评分等。

（1）销售订单模型（SalesOrder）

该模型定义了销售订单的字段，包括客户的联系方式telephone，发货地址shippingAddress，下单时间time等。与Customer（客户）和Quotation（报价）模型关联。

销售订单模型代码：



主要包含以下字段：

Id，销售订单的唯一标识，自增主键。

telephone，联系电话号码，字符字段，最大长度为11个字符，使用自定义验证器

validate\_phone，验证输入的格式。 fax，传真号码，字符字段，最大长度为10个字符，使用自定义验证器

validate\_digit，验证输入的格式。

shippingAddress，送货地址，字符字段，最大长度为50个字符。

time，创建销售订单的时间，自动设置为创建时的时间。

rfq，外键，关联到 `Quotation` 模型，表示该销售订单关联的报价。如果相关报价被删除，相应的销售订单也会被删除。此字段可以为空。

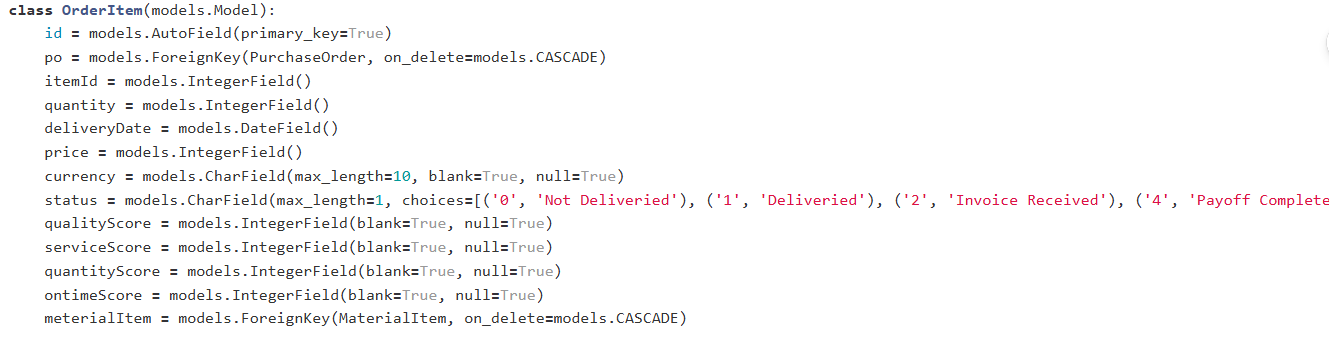
customer，外键，关联到 `Customer` 模型，表示客户。

euser，外键，关联到 `EUser` 模型，表示提交此销售订单的用户。

（2）订单项模型（OrderItem）

该模型定义了订单中的具体物品信息，包含数量quantity，交货日期deliveryDate，单价price等。与SalesOrder（销售订单）和MaterialItem（物料库存项）模型关联。

订单项模型代码：



主要包含以下字段：

Id，订单项的唯一标识，自增主键。

po，外键，关联到 `SalesOrder` 模型，表示该项属于哪个销售订单。

itemId，订单项的项目ID，用于区分同一销售订单中的不同项目。

quantity，销售数量，以整数形式表示。

deliveryDate，预计交货日期，日期字段。

price，订单项的价格，以整数形式表示。

currency，货币类型，字符字段，最大长度为10个字符，可以为空。

status，订单项的状态，'0'表示未交货，'1'表示已交货，'2'表示已收到发票，'4'表示付款完成。

qualityScore，质量评分，整数字段，可以为空。

serviceScore，服务评分，整数字段，可以为空。

quantityScore，数量评分，整数字段，可以为空。

ontimeScore，准时交货评分，整数字段，可以为空。

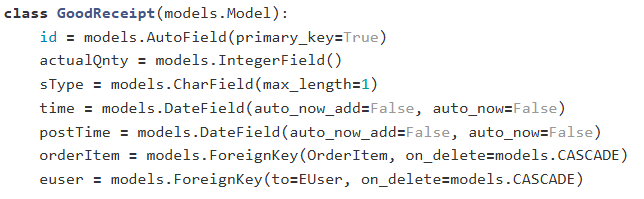
meterialItem，外键，关联到 `MaterialItem` 模型，表示该订单项涉及的具体物料项目。

### 3.1.7 发货管理模块（GoodDelivery）

该模块管理实际发货过程。记录发货时的实际数量actualQnty，发货时间time等信息。与OrderItem（订单项）模型关联。

主要相关的功能包括记录实际发出的货物信息，比较实际发货数量与订单数量，集成质量检验流程，发货后自动更新相关库存等。

发货模型代码：



主要包含以下字段：

Id，发货的唯一标识，自增主键。

actualQnty，实际发出的数量，以整数形式表示。

sType，发货类型，字符字段，最大长度为1个字符。

time，发出货物的时间，日期字段。

postTime，发货单记录时间，日期字段。

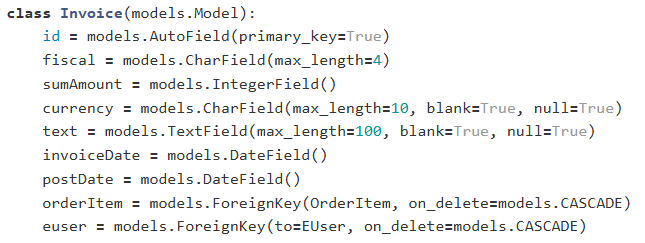
orderItem，外键，关联到 `OrderItem` 模型，表示该发货单与哪个订单项关联。

euser，外键，关联到 `EUser` 模型，表示提交此发货操作的用户。

### 3.1.8 发票管理模块(Invoice)

该模块处理客户发票信息。定义了发票的相关信息，包括总金额sumAmount，开票日期invoiceDate等。与OrderItem（订单项）模型关联。主要相关的功能包括记录收到的客户发票信息，将发票与订单和发货单匹配，审核发票金额和内容，为财务部门准备付款所需信息等。

发票管理模块代码：



主要包含以下字段：

Id，发票的唯一标识，自增主键。

fiscal，财政年度，以4个字符的形式表示。

sumAmount，发票的总金额，以整数形式表示。

currency，货币类型，字符字段，最大长度为10个字符，可以为空。

text，发票的描述或备注，文本字段，最大长度为100个字符，可以为空。

invoiceDate，发票的开具日期，日期字段。

postDate，发票的过账日期，日期字段。

orderItem，外键，关联到 `OrderItem` 模型，表示该发票与哪个订单项关联。

euser，外键，关联到 `EUser` 模型，表示提交此发票的用户。

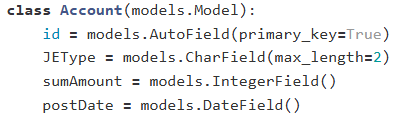
### 3.1.9 财务记账模块(Account, AccountDetail)

这组模块处理与订单相关的财务记账。主要相关的功能包括基于发货和发票自动生成会计分录，提供订单、库存和财务之间的对账功能，协助进行归集和分析，为财务报告提供必要的数据支持等

（1）账户模型（Account）

记录财务相关的帐务信息。

账户模型代码：



主要包含以下字段：

Id，会计凭证的唯一标识，自增主键。

JEType，凭证类型，字符字段，最大长度为2个字符。

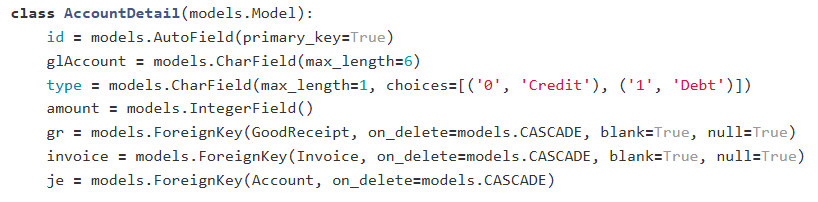
sumAmount，凭证的总金额，以整数形式表示。

postDate，凭证的过账日期，日期字段。

（2）账户明细模型（AccountDetail）

记录具体的财务明细，包括总账科目glAccount，金额amount等。与GoodDelivery（发货）和Invoice（发票）模型关联。

账户明细模型代码：



主要包含以下字段：

Id，会计明细的唯一标识，自增主键。

glAccount，总账科目，以6个字符的形式表示。

Type，明细类型，'0'表示贷方，'1'表示借方。

Amount，金额，以整数形式表示。 Gr，外键，关联到 `GoodDelivery` 模型，表示该明细与哪个货物发货单关联，可以为空。

Invoice，外键，关联到 `Invoice` 模型，表示该明细与哪个发票关联，可以为空。

Je，外键，关联到 `Account` 模型，表示该明细与哪个会计凭证关联。

## 3.2数据库优点

### 3.2.1 关系管理和业务逻辑体现

我们巧妙地利用了Django的外键关系来反映实际业务中的关联。例如，Customer模型与EUser模型之间建立的外键关系，不仅体现了客户与系统用户之间的联系，还为复杂的跨表查询和数据分析提供了可能。同时，模型中包含了如客户评分（score）、公司代码（companyCode）等字段，直接体现了客户管理、多公司运营等复杂的业务逻辑。这种设计使得数据模型不仅是数据的容器，更成为了业务逻辑的载体。

# 4代码赋值清单

代码是指表示事物名称、属性、状态的符号和记号。在我们的SD系统中，我们依据系统中的不同对象，进行了代码值的设计。下面是系统中各个主要对象的代码赋值清单：

## 4.1公司代码（CompanyCode）

代码结构为XX-YY。XX表示该公司所在的国家，YY表示公司在这个国家中的序号。

代码赋值示例：

|  |  |
| --- | --- |
| 公司代码 | 公司所在国家 |
| CN00 | 中国 |
| VN00 | 越南 |
| EG00 | 埃及 |
| BR00 | 巴西 |
| AR00 | 阿根廷 |
| IN00 | 印度 |
| JP00 | 日本 |
| TH00 | 泰国 |
| US00 | 美国 |

## 4.2国家代码（CountryCode）

代码结构为XX。这是国家的英文的缩写，我们的系统共包括了共239个国家和地区。

代码赋值示例：

|  |  |  |
| --- | --- | --- |
| 代码 | 国家（中文） | 国家（英文） |
| AL | 阿尔巴尼亚 | Albania |
| DZ | 阿尔及利亚 | Algeria |
| AF | 阿富汗 | Afghanistan |
| AR | 阿根廷 | Argentina |
| AE | 阿联酋 | United Arab Emirates |
| AW | 阿鲁巴 | Aruba |
| OM | 阿曼 | Oman |
| AZ | 阿塞拜疆 | Azerbaijan |
| EG | 埃及 | Egypt |
| ET | 埃塞俄比亚 | Ethiopia |
| IE | 爱尔兰 | Ireland |
| EE | 爱沙尼亚 | Estonia |
| AD | 安道尔 | Andorra |
| AO | 安哥拉 | Angola |
| AI | 安圭拉 | Anguilla |
| AG | 安提瓜和巴布达 | Antigua and barbuda |
| AT | 奥地利 | Austria |
| AU | 澳大利亚 | Australia |
| CN | 中国澳门 | Macau |
| BB | 巴巴多斯 | Barbados |
| PG | 巴布亚新几内亚 | Papua New Guinea |
| BS | 巴哈马 | Bahamas |
| PK | 巴基斯坦 | Pakistan |
| PY | 巴拉圭 | Paraguay |
| PS | 巴勒斯坦 | Palestine |
| BH | 巴林 | Bahrain |
| PA | 巴拿马 | Panama |
| BR | 巴西 | Brazil |
| BY | 白俄罗斯 | Belarus |

## 4.3货币代码（CurrencyCode）

代码结构为XXX。XXX表示货币的简称代码，我们的系统共包括了169种货币

代码赋值示例：

|  |  |  |  |
| --- | --- | --- | --- |
| 货币代码 | 货币（中文） | 货币（英文） | |
| CNY | 人民币 | Chinese Yuan | |
| USD | 美元 | United States Dollar | |
| EUR | 欧元 | Euro |  |
| GBP | 英镑 | British Pound | |
| AUD | 澳元 | Australia Dollar | |
| CAD | 加元 | Canadian Dollar | |
| JPY | 日元 | Japanese Yen | |
| HKD | 港币 | Hong Kong Dollar | |
| INR | 印度卢比 | Indian Rupee | |
| ZAR | 南非兰特 | South African Rand | |
| TWD | 新台币 | New Taiwan Dollar | |
| MOP | 澳门元 | Macau Pataca | |
| KRW | 韩元 | South Korean Won | |
| THB | 泰铢 | Thai Baht | |
| NZD | 新西兰元 | New Zealand Dollar | |
| SGD | 新加坡元 | Singapore Dollar | |
| AED | 阿联酋迪拉姆 | United Arab Emirates Dirham | |
| AFN | 阿富汗尼 | Afghan Afghani | |
| ALL | 阿尔巴尼列克 | Albania Lek | |
| AMD | 亚美尼亚德拉姆 | Armenia Dram | |
| ANG | 荷兰盾 | Dutch Guilder | |
| AOA | 安哥拉宽扎 | Angola Kwanza | |
| ARS | 阿根廷比索 | Argentina Peso | |

## 4.4语言代码（LanguageCode）

代码结构为XX。XX是国家的简称代码，我们用国家的代码代指语言，我们的系统中共包括了41种语言。

代码赋值示例：

|  |  |
| --- | --- |
| 语言代码 | 语言 |
| af | 南非语 |
| ar | 阿拉伯语 |
| bg | 保加利亚语 |
| cs | 捷克语 |
| da | 丹麦语 |
| de | 德语 |
| dz | 不丹语 |
| en | 英语 |
| es | 西班牙语 |
| fa | 波斯语 |
| fi | 芬兰语 |
| fr | 法语 |
| ga | 爱尔兰语 |
| hu | 匈牙利语 |
| is | 冰岛语 |
| it | 意大利语 |
| ja | 日语 |
| kg | 刚果语 |
| kk | 哈萨克语 |
| kl | 格陵兰语 |
| ko | 朝鲜语、韩语 |
| la | 拉丁语 |
| lo | 老挝语 |

## 4.5计量单位代码（MeaunitCode）

计量单位代码没有统一的结构，主要为计量单位的国际通用的简称。我们的系统中共包含了168种计量单位。

代码赋值示例：

|  |  |
| --- | --- |
| 计量单位代码 | 计量单位（英文） |
| '' | Inch |
| ''2 | Square inch |
| ''3 | Cubic inch |
| % | Percentage |
| %O | Per mille |
| 0 | Meter/Minute |
| 1 |  |
| ACR | Acre |
| AU | Activity unit |
| BAG | Bag |
| BT | Bottle |
| C3S | Cubic centimeter/second |
| CAN | Canister |
| CAR | Carton |
| CCM | Cubic centimeter |
| CD3 | Cubic decimeter |
| CL | Centiliter |
| CM | Centimeter |
| CM2 | Square centimeter |
| CMS | Centimeter/second |
| CRT | Crate |
| CV | Case |
| D | Days |
| DAY | Days |
| DEG | Degree |

## 4.6工厂代码（PlantCode）

代码结构为XX-YY。XX表示工厂基地所在的省区，YY序号表示在所在省区包含的工厂中的序号。我们的系统中共包含了25个工厂。

代码赋值示例：

|  |  |  |
| --- | --- | --- |
| 工厂代码 | 工厂基地名称 |  |
| GD00 | 佛山顺德基地 | |
| AH00 | 安徽芜湖基地 | |
| GD01 | 广东中山基地 | |
| AH01 | 安徽合肥基地 | |
| HB00 | 湖北武汉基地 | |
| GD02 | 广州南沙基地 | |
| JS00 | 江苏淮安基地 | |
| CQ00 | 重庆基地 | |
| JS01 | 江苏苏州基地 | |
| HB01 | 湖北荆州基地 | |
| JS02 | 江苏无锡基地 | |
| SX00 | 山西临汾基地 | |
| HD00 | 河北邯郸基地 | |
| JX00 | 江西贵溪基地 | |
| JX01 | 江苏常州基地 | |
| BD00 | 越南平阳省基地 | |
| SZ00 | 埃及苏伊士湾基地 | |
| PA00 | 巴西波苏阿雷格里市基地 | |

## 4.7销售组织代码

代码结构为XX-YY。XX表示所在地区的简称，YY表示序号。

代码赋值示例：

|  |  |  |  |
| --- | --- | --- | --- |
| 销售组织代码 | 分销渠道简称 | 所在地区 | 分销渠道全称 |
| CN00 | IN | 中国区 | Internet |
| CN00 | WH | 中国区 | Wholesale |
| AF00 | WH | 非洲区 | Wholesale |
| SA00 | WH | 南美洲区 | Wholesale |
| AS00 | IN | 海外亚洲区 | Internet |
| AS00 | WH | 海外亚洲区 | Wholesale |

## 4.8付款方式代码（TPtyprCode）

代码结构为顺序码，每个数字表示一种付款方式。我们的系统共包括8种付款方式。

代码赋值示例：

|  |  |
| --- | --- |
| 付款方式代码 | 付款方式全称 |
| 1 | 付现 |
| 2 | 交货付款 |
| 3 | 交单付款 |
| 4 | 记账付款 |
| 5 | 托收付款 |
| 6 | 信用证付款 |
| 7 | 分期付款 |
| 8 | 延期付款 |

## 4.9结论

本系统通过对各主要对象的代码结构和代码值的设计，使得系统内部数据的管理更加规范、易于维护，同时便于通过代码快速识别和定位相关数据对象。这些代码不仅便于识别和管理，还保证了数据的一致性和准确性，在不同的模块间可以进行传递和使用。以上就是代码赋值清单。

# 5 输入输出设计

输入输出是用户与计算机的界面。手工编制的凭证，通过系统输入，经过计算机加工处理，将有关信息以报表、图形等形式提供给管理人员。输入与输出设计是数据交互流程里的核心构成部分，对系统整体性能表现、运行稳定可靠性以及用户体验均有着重要的作用。具体而言，本系统的输入输出设计将从多个维度展开深入探讨。以下将围绕具体的各个方面，对本系统的输入输出设计进行全面的阐述：

## 5.1 输入设计

在信息系统分析阶段，输入设计的任务是确定输入的内容。而在信息系统设计阶段，输入设计的内容是输入格式的设计、输入数据的正确性检验，以及联机系统的输入界面设计。

### 5.1.1 输入格式设计

输入格式首先指的是需要用户在界面上输入的数据的排列格式。本系统数据的排列格式主要有如下几项。

（1）界面中，按照从上到下、从左到右的形式排列所有的数据项，重要数据单独成行，相关数据排为一行，如国家、地区两个数据在一行录入.

（2）界面中，数据采取相关数据相邻出现的方式，将所有需要录入的数据分组，并分栏显示。如下图，在维护客户关系界面中，将与地址相关的街道、邮编、国家、地区等信息统一归为“街道地址”一栏，使用户同时输入。

（3）界面中，同时使用了“填空”和“选择”两种输入方式。对于取值较多的数据，采取用户自由输入的形式进行录入，而对于国家、地区等取值有限的数据，用户既可以从选项中选择，又可以自行输入，可以加快输入的效率。

（4）界面中，充分考虑了终端屏幕所允许的最大宽度和最大长度。宽度方面，界面中，同一行数据项一般为一至二项，以保证数据项不会超过一行的宽度，长度方面，界面中，所有数据都尽可能在同一屏内展示出，以最大程度的减小用户移屏的次数，提高用户输入的效率。

另一方面，输入格式还包括前端传递给后端的数据的格式。前端主要通过Get和Post两种方式传递数据给后端。对于Get方式，直接使用键值对的方式进行传值，而对于Post方式，则采用表单的方式传递数据。此处，将客户关系管理、物料及库存管理部分所有功能使用到的输入数据格式整理如下。

|  |  |  |
| --- | --- | --- |
| 功能 | 传入参数 | 传入方式 |
| 通过地址查询Business Partner | Street | Get |
| Postal Code |
| Country |
| City |
| 通过名称查找Business Partner | Name | Get |
| 通过编号查找Business Partner | Search Term | Get |
| 通过Search Term查找Business Partner | Search Term | Get |
| 保存Business Partner | 数据库Customer表中所有字段 | Post |
| 更新Business Partner | 数据库Customer表中所有字段 | Post |
| 保存联系人 | 数据库contact\_person表中所有字段 | Post |
| 更新联系人 | 数据库contact\_person表中所有字段 | Post |
| 保存联系 | 数据库relationship表中所有字段 | Post |
| 创建物料 | 数据库material\_SD表中所有字段 | Post |
| 查询物料 | 数据库material\_SD表中所有字段 | Get |
| 保存物料更改 | 数据库material\_SD表中所有字段 | Post |
| 确认物料入库 | 入库时间 | Get |
| 操作者 |
| 物料名称 |
| 入库量 |
| 入库工厂 |
| 入库库位 |
| 查询物料库存 | 物料编号 | Get |
| 工厂 |
| 查询低于安全库存的物料 | 无参数 | Get |

### 5.1.2 输入格式的正确性检验

引起数据输入错误的原因有两类：一类是数据采集过程中发生的错误，如原始凭证丢失、填写不准确、凭证传递延误等引起的数据输入错误，对于这类错误可以通过建立严格的凭证管理制度来避免；另一类是在录入数据阶段，由数据录入员的错读、漏读、误操作等原因引起的数据输入错误，对于这类错误，可以采用检验方法来解决：

（1）重复检验。将同一数据先后输入两次，然后由计算机程序进行自动对比，若两次输入不一致，则计算机显示出错。例如，系统中设置和修改密码的操作要求用户输入两次密码。

（2）视觉检验。在手工输入的同时，由计算机打印或显示输入数据，然后与原始单据进行比对，找出差错。

（3）逻辑检验。检查业务上各种数据是否符合逻辑。例如，本部分涉及到较多关于国家、地区的录入，工厂、库位的录入，销售组织、分销渠道的录入等，不同的国家对应不同的地区，不同的工厂可能涉及到的库位不相同，而不同的销售组织有权销售的渠道可能也不尽相同，因此，在保存数据时，就需要检验是否有录入地区不属于录入国家、录入库位不属于录入工厂、录入分销渠道不属于录入销售组织等情况出现。具体操作过程中，对于新建时的录入，采用表格选择的方式，即用户只能在所有可能的选项中选择一项，而不能随意录入；而对于修改时的录入，则在保存时进一步检验是否有上述违反逻辑的情况发生，如果有，则提示用户。

（4） 必填项检验。录入数据时，某些数据字段为必填项，而其他数据项则为选填项。所有必填数据字段均在页面中用红色星号表示，同时，对于必填数据字段，需要在保存时进行检验，如果没有填写，则需要提示用户录入该字段。

## 5.2 输出设计

输出设计是系统为用户和管理层提供信息反馈、支持决策的核心部分。输出数据的准确性、完整性和展示的直观性直接关系到系统的实用性和用户体验。

系统的输出数据类型主要包括用户信息、业务流程文档、库存报告和财务报表等。这些数据不仅通过前端界面进行实时展示，还可以导出为文档或表格形式，供用户进一步处理或保存。具体来说，用户个人信息和权限列表如用户名、所属部门和角色等信息，可以随时从系统中导出，以便进行权限审查和管理。而在业务流程中，系统能够生成订单的明细表，包括订单编号、客户、物料项、数量和价格等。这些明细表不仅可以通过界面查看，还可以导出为PDF文件，便于存档和审核。

系统的输出方式不仅限于在界面上展示，还支持将关键业务数据导出为各种格式的文档。通过PDF导出功能，系统能够将订单、库存报告、发票信息等生成可打印的文档，供用户归档和审查。而Excel导出功能则使用户能够将这些数据转化为可编辑的表格形式，便于进行进一步的数据分析和处理。

为了增强系统的集成性和数据共享能力，系统同样提供了RESTful API接口，支持外部系统调用获取系统数据。例如，外部系统可以通过这些接口获取订单状态、库存信息等，提升了跨系统的数据共享与业务协同能力。为了确保管理层能够及时获取最新的业务信息，系统设计了自动化报告生成功能，支持定时任务。这一功能可以根据预设的时间间隔，自动生成并发送业务报告（如库存日报、订单处理情况报告等）至指定用户的邮箱，确保管理层能够随时掌握业务动态，做出及时的决策。

功能模块输出格式:

1. 客户与联系人管理

|  |  |  |
| --- | --- | --- |
| 功能 | 输出格式 | 说明 |
| 查找Customer | Customer对象 | 包含客户主数据（名称、地址等） |
| 查找ContactPerson | ContactPerson对象 | 包含联系人信息（姓名、电话、职位等） |
| 保存BP/更新BP | 1或-1 | BP操作结果 |
| 保存Contact Person | 1或-1 | 联系人保存结果 |

2. 物料管理

|  |  |  |
| --- | --- | --- |
| 功能 | 输出格式 | 说明 |
| 查询物料 | MaterialSd对象 | 包含物料主数据（描述、单位、价格等） |
| 创建物料 | 1或-1 | 物料主数据创建结果 |
| 保存物料更改 | 1或-1 | 物料修改操作结果 |
| 查询物料库存 | JSON数组（示例见下） | 包含库存状态（可用量、计划交付量等） |

库存查询示例：

json

|  |
| --- |
| [ |
| { |
| "Unrestricted": "530", // 可用库存 |
| "SchedForDel": "0", // 计划交付量 |
| "id": "1", // 物料ID |
| "Client": "full", // 工厂/存储地点 |
| "SalesOrder": "0" // 预留量 |
| }, |
| ... |
| ] |

3. 库存预警

|  |  |  |
| --- | --- | --- |
| 功能 | 输出格式 | 说明 |
| 展示unrestricted库存低于安全库存的物料 | mid, plant, unrestricted | 返回物料ID、工厂及当前可用库存量 |

示例输出：

|  |
| --- |
| 物料ID: 1001, 工厂: MI00, 可用库存: 45 （假设安全库存为50） |

4. 业务操作

|  |  |  |
| --- | --- | --- |
| 功能 | 输出格式 | 说明 |
| 保存Relationship | 1或-1 | 客户与物料关联关系保存结果 |
| 确认货物入库 | 1或-1 | 库存移动（如发货）操作结果 |

## 5.3 输入输出的异常处理

在信息系统的设计进程中，保障系统的稳定运行以及数据的精准无误非常重要，而构建完善的异常处理机制则是达成这一目标的重要支撑。基于我们已完成的信息系统设计方案，异常处理机制贯穿于系统的输入与输出环节，旨在数据交互过程中及时察觉并妥善处理潜在的错误或异常状况，进而确保系统的可靠性，提升用户体验。

针对数据输入流程，系统搭建了实时监控与严格验证机制。当用户在系统中录入数据时，系统会即刻对输入数据的格式展开细致检查。以用户输入电话号码、邮政编码等关键信息为例，系统借助自定义的验证器（如validate\_phone和validate\_digit）进行格式校验。依托Django框架的内置功能，系统能够迅速识别格式错误，并通过前端界面及时向用户反馈错误提示，明确指出输入数据存在的具体问题。这种设计模式有效阻止了错误数据流入系统，降低了后续数据修正的成本，同时保障了数据进入系统时的规范性与准确性。

此外，系统对重复数据输入问题也给予了考量。在我们研发的系统中，数据的唯一性十分重要。当用户输入的新客户名称或物料名称与系统中现有记录重复时，系统会即刻发出警告，并提示用户可能存在重复输入的情况。系统不仅允许用户对重复数据进行修改，还提供了合并选项，以便在必要时将重复的数据记录进行整合。这一机制有效避免了数据冗余，维持了数据库中数据的一致性与可用性。

在数据输出阶段，系统同样构建了详尽的异常处理机制。在生成报告或导出数据时，系统可能会遭遇数据缺失或处理错误等问题。针对此类情况，系统设计了精细的错误日志记录功能。每当系统检测到输出过程中的异常，便会自动生成日志记录，详细记录异常发生的时间、涉及的数据以及错误的具体情形。这些日志为系统管理员提供了关键的调试依据，助力他们快速定位问题根源，并采取针对性的纠正措施。

同时，当系统检测到异常时，会通过前端界面向用户发出提示，说明操作未成功的原因，并提供可行的解决方案，例如建议用户重新尝试导出操作，或联系技术支持团队进行进一步处理。这种设计提升了用户体验和系统的易用性。

对于系统与外部应用的集成，系统提供了RESTful API接口。然而，在调用API过程中，可能会面临网络连接不稳定或数据不一致等问题。为此，系统设计了详细的错误码和错误信息返回机制。当外部系统调用API时，若发生异常，系统将返回具体的错误码，并附带详细的错误信息，帮助调用方快速识别并解决问题。通过这种精细化的异常处理设计，系统在与外部应用的交互中保持了高度的稳定性与可靠性。

综上所述，我们系统中的异常处理设计不仅确保了数据输入的规范性以及数据输出的完整性，还具备强大的日志记录和错误处理能力，有力保障了系统在各种复杂场景下的稳定运行。这些机制的实施遵循了信息系统设计的最佳实践，确保了系统能够在企业的日常运营中充分发挥其效能，为企业的管理决策提供坚实可靠的支持。

# 6 用户界面设计

## 6.1用户界面设计原则

用户界面是人机联系的桥梁。用户界面设计的优劣关系到软件产品的竞争力。本系统使用的用户界面设计原则主要有以下几项：

(1)一致性：一致性的外观和功能界面是用户界面重要的设计目标之一。信息在窗体上的组织方式、 菜单项的名称及其排列、图表的大小和形状、字号和字体、按钮在窗口的位置、提示用词、 界面色调以及任务的执行次序保持统一，并贯穿信息系统的始终。

(2)提供有效反馈：用户在系统中进行的操作，都需要及时获得反馈是否执行成功。对于查询等操作，用户按下按钮时界面会显示查询结果，因此用户自然可以知道操作是否执行成功，但是对于保存等操作，由于按下按钮后操作是在数据库中进行的，界面上不会有任何变化，因此，需要通过提示消息的方式告知用户操作是否成功进行。例如，保存联系人的操作中，当保存成功时，后端会返回“1”，而保存失败时，后端会返回“-1”，（改）而前端则根据后端返回的内容来判断是否保存成功。

(3)提供错误处理机制。为了防止用户出错，本系统也提供了一系列错误处理机制，如逻辑检验、必填项检验等。这部分内容在5.1.2节输入格式的正确性校验中已经详细说明，此处不再赘述。

(4)减少短期记忆负担。由于用户在同一时间内能记忆的信息是有限的，所以本系统尽量减少不必要的用户记忆。例如，对于用户创建的联系人、物料等编号，在输出框旁全部提供查询按钮，用户可以用较容易记忆的Search Term等筛选条件查询到它们的编号。

(5)允许撤销操作

## 6.2 登录注册界面

当用户进入我们的网站时，需要点击右上角的登录注册按钮，此时会跳转到登录界面，在文本框里输入正确的账号和密码即可开始使用网站。

如果用户忘记了密码，则可以选择“忘记密码”，然后通过邮箱地址和密保问题，重新设置一个密码，然后进行登录。

如果用户尚未拥有账号，需要点击登录按钮下方的“注册账号”进行账号注册。账号注册的第一步是填写个人信息，用户在此页面输入想要设置的用户名和姓名，如果用户名已被注册则会在注册时提示。点击“下一步”进入账号信息填写。

账号注册的第二步为填写账号信息，需要填写包括邮箱地址，部门和电话等信息，填写完成后点击“下一步”，也可以点击“上一步”回到个人信息填写。

账号注册的第三步是设置密码，需要填写密码、确认密码、两个密保问题及其答案等内容。然后点击“立即注册”即可注册成功。注册成功后即可回到登录界面，登录后就可以使用网站功能。

## 6.3 主页

主页包括了侧边栏和主界面的模块跳转。在侧边栏列出了我们系统的模块功能，主要包括了“客户管理”、 “销售订单管理”、“发货管理”、“文档流”等模块。主界面则是将这些模块以及这些模块下包含的功能全部进行了展示，用户可以点击这些模块跳转到对应的功能。

右上角则是个人ID，用户可以点击“退出”退出登录。

## 6.4客户管理系统界面

本部分主要包括创建、查找、修改客户信息。

### 6.2.1客户、联系人的创建与维护

按照功能可以分为三部分。上方按钮构成的区域用于切换创建客户和联系人、切换显示状态等，在录入客户或联系人信息时，根据录入的角色不同还会展示不同的按钮以供用户录入不同的信息。下方左侧是查询客户的区域，右侧是新建、修改和展示客户信息的区域。

对于左侧的查询区，界面既允许用户查询客户又允许客户查询联系人（也可以二者同时查询）。

同时，界面还提供了四种不同的查询方式，分别是根据地址查询、根据名称查询、根据编号查询和根据搜索项查询。每种查询方式会对应不同的查询条件。

按下“start”键进行查询后，下方表格中会显示出所有的查询结果。查询结果由两部分构成，左侧一列代表编号，而右侧一列则包含该客户或联系人的简要信息，包括名称、城市、邮编等内容。

双击其中某一条记录，右侧区域可以展示出该客户详细的信息。默认状态下是只读模式，单击上方按钮“Switch Between Display and Change”后，界面变为可编辑模式。如果需要保存客户信息，可以点击下方“Save”按钮进行保存。

对于创建客户，首先点击上方按钮区中的“Organization”。首先录入“FI Customer”角色下的信息。

接下来点击上方“Company Code”按钮录入Company Code相关数据。首先输入Company Code并回车，此时下方两个页面输入栏变为可编辑状态。

然后切换到Customer角色，并单击“Sales and Distribution”按钮。首先选择Sales Organization等与Sales Area相关的数据，然后依次输入下方三个标签页中的数据。最终，单击“Save”按钮，页面会提示已经成功保存相关信息，即可完成客户的创建。

对于联系人的创建与维护，与客户类似，此处以创建联系人为例。只需要点击上方“Person”按钮，并录入相关信息，即可完成联系人的创建。

### 6.2.2关系的创建

对于每一项数据，都提供了查找框，用户无需记忆繁琐的代码与编号，只需要根据相关条件进行搜索即可。

## 6.3发货管理系统界面

### 6.3.1 物料出库登记

界面中，首先需要用户选择出库的时间。接下来，在下方以表格形式列出了登记出库需要录入的信息，每一行对应一种物料在一个工厂的出库。可以通过单击下方“Add Row”按钮来新增一行。录入完所有需要出库的物料之后，单击“Post”按钮，即可保存所有的出库记录，同时，库存也会自动增加。

除此之外，本界面还设有预警功能。右侧的“Materials with low inventory”按钮提供了库存低于安全库存的物料预警。右上角的数字代表低于安全库存的物料种数，单击该按钮，可以查看详细信息。

### 6.3.2 物料库存查询

提供了查询物料库存的功能。上方需要指定查询的物料和物料所属的工厂，单击“Execute”按钮，下方即可查看物料库存。库存中共分为四级，最上方的一级“Full”表示输入物料所有工厂的库存之和，第二级表示输入工厂所属公司的物料总库存，第三级表示输入工厂的总库存，第四级表示输入工厂该物料所属的库位。

## 6.4销售订单管理系统界面

首先进入销售订单管理的销售订单申请模块主页。

点击“创建销售订单申请”功能，输入商品编码、名称等基本信息创建销售订单申请。

点击“管理销售订单申请”功能，使用销售订单申请编码、创建者编码等信息搜索销售订单列表。下方会列出相符合的销售订单申请的概括信息。点击后可进入查看详情。

点击“修改销售订单申请”功能，可以修改销售订单信息的部分信息。

点击进入“创建报价申请”功能。通过销售订单申请编码等信息找到相应的销售订单申请，将销售订单申请导入到报价申请，然后添加客户等报价信息来创建报价申请。

点击进入“查询报价申请”功能，通过输入报价申请编码、客户编码等信息查询报价申请，然后点击进入查看报价申请的详情。

点击进入“添加客户报价”功能，通过报价申请编码等查找报价申请，然后将报价申请导入转化，添加客户报价。

点击进入“修改客户报价”功能，使用报价申请编码等信息查找客户报价，进入修改。

然后进入销售订单订单管理主页。进入“系统创建销售订单”。

通过销售订单申请编码等查找报价申请，然后点击转化即可将报价申请转化为销售订单。

查看并填写转化销售订单的具体信息。

点击“查看销售订单”功能，通过销售订单编码等信息查找销售订单，然后查看或修改具体信息。

## 6.5 财务管理系统界面

进入“创建产品发货单”功能，通过输入销售订单编码等信息查找销售订单，然后查看销售订单详情，创建产品发货单。

点击“查看商品发货单”功能，通过输入发货单编码等信息查看对应的商品发货单，查看详情。

点击“创建发票收据”功能，通过输入销售订单编码等信息搜索对应的销售订单列表，然后查看详情，创建发票收据。

点击“查看发票”功能，使用发票编码等信息查看发票具体信息。

## 6.6 文档流

最后是文档流功能模块。点击进入，使用商品项编码等查找对应的销售订单，点击查看详情，可以查看销售订单、商品发货单、交易发票等的相关文档流。

# 7 结构图及模块说明书

## 7.1结构图

本信息系统的结构图如下图所示，我们所设计的ERP系统主要包括客户管理，发货管理，订单管理，财务管理，文件流管理及其子模块等功能。

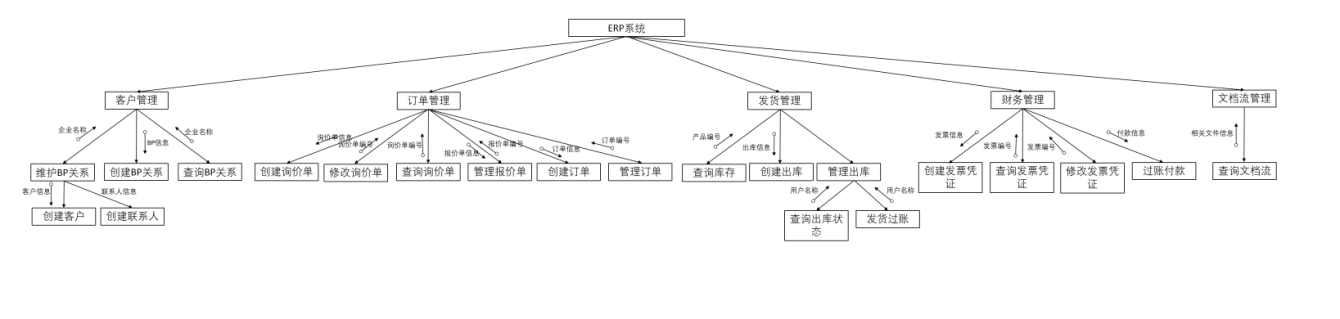


图 结构图

## 7.2 模块说明书

各模块说明书依次介绍如下。

1. 客户管理模块说明书

|  |
| --- |
| 模块名称：客户管理 |
| 对每一个客户  调用“维护BP关系”，创建新客户信息，并为新客户创建联系人  调用“创建BP关系”，进行客户的商业伙伴关系创建  调用“查询BP关系”，进行客户的商业伙伴关系查询 |
| 模块名称：维护BP关系 |
| 显示新客户创建和客户联系人创建界面  输入客户名称，客户所处国家等信息  如果客户信息合法  则完成客户信息创建  否则  返回错误信息提示  找到想为其创建联系人的客户，输入联系人姓名，所处国家等信息  如果联系人信息合法  则完成客户联系人信息创建 |
| 模块名称：创建BP关系 |
| 显示BP关系创建界面  输入想要创建BP关系的客户名称，填入相关BP信息  若合法  则完成BP关系创建  否则  返回空信息 |
| 模块名称：查询BP关系 |
| 显示BP关系查询界面  输入想要查询BP关系的客户名称  若合法  则返回BP关系信息  否则  返回空信息 |

1. 订单管理模块说明书

|  |
| --- |
| 模块名称：订单管理 |
| 对每一个订单  调用“创建询价单”，创建询价单信息  调用“修改询价单”，进行询价单信息的修改  调用“查询询价单”，进行询价单信息的查看  调用“管理报价单”，进行报价单信息的创建、查看和修改  调用“创建订单”，创建订单信息  调用“管理订单”，进行订单信息的查看和修改 |
| 模块名称：创建询价单 |
| 显示询价单创建界面  输入商品名称，价格等信息  如果询价单信息合法  则完成询价单信息创建  否则  返回错误信息提示 |
| 模块名称：修改询价单 |
| 显示询价单修改界面  输入询价单编号  若合法  则返回询价单信息并允许修改  否则  返回空信息 |
| 模块名称：查询询价单 |
| 显示询价单查询界面  输入询价单编号  若合法  则返回询价单信息否则  返回空信息 |
| 模块名称：管理报价单 |
| 显示报价单创建、查看或修改界面  输入商品名称，价格等报价单信息  如果报价单信息合法  则完成报价单信息创建  否则  返回错误信息提示 输入报价单编号  若合法  则返回商品信息并允许修改  否则  返回空信息 |
| 模块名称：创建订单 |
| 显示销售订单创建界面  输入商品名称，商品数量等信息或通过调用采报价单进行销售订单的创建  若合法  则成功创建销售订单，返回销售订单编号  否则  提示错误信息 |
| 模块名称：管理订单 |
| 显示销售订单的查询或修改界面  输入销售订单编号  若合法  则返回销售订单信息并允许修改  否则  提示错误信息 |

1. 发货管理模块说明书

|  |
| --- |
| 模块名称：查询库存 |
| 查询库存状态  输入产品编号等信息  若合法  则返回库存信息  否则  返回空信息 |
| 模块名称：创建出库 |
| 显示创建出库界面  输入出库信息  如果合法  则完成出库信息创建  否则  返回错误信息提示 |
| 模块名称：管理出库 |
| 调用“查询出库状态”  调用“发货过账” |
| 模块名称：查询出库状态 |
| 查询库存状态  输入用户名称  若合法  则返回出库状态  否则  返回空信息 |
| 模块名称：发货过账 |
| 输入用户名称  若合法  则显示发货过账操作，用户可进行发货过账  否则  返回空信息 |

1. 财务管理模块说明书

|  |
| --- |
| 模块名称：创建发票凭据 |
| 显示发票凭据创建界面  输入商品编号，订单编号等信息  若合法  则进行发票凭据的创建  否则  提示错误信息 |
| 模块名称：查询发票凭据 |
| 显示发票凭据查询界面  输入发票编号等信息  若合法  则返回发票信息  否则  返回空信息 |
| 模块名称：修改发票凭据 |
| 显示发票凭据修改界面  输入发票编号等信息  若合法  则返回发票凭据信息并允许修改  否则  返回空信息 |
| 模块名称：过账付款 |
| 显示过账付款界面  输入销售订单编号，发票编号等信息  若合法  则显示过账付款操作，用户可进行过账付款  否则  返回空信息 |

1. 文档流管理模块说明书

|  |
| --- |
| 模块名称：文档流管理 |
| 对每一个销售订单  调用“展示文档流”，进行各相关文件的查看 |
| 模块名称：展示文档流 |
| 显示文档流展示界面  输入商品编号或销售订单编号等信息  若合法  则展示所有相关文件信息  否则  返回空信息 |

# 8 信息系统安全设计​

## 8.1 概述​

在销售与分销（SD）系统中，信息系统安全是保障企业业务顺畅运行的核心支柱。作为连接客户管理、订单处理、发货管理和财务管理的集成平台，SD 系统存储和处理着大量敏感信息，包括客户的联系方式、付款条件、订单价格以及财务对账记录等。这些信息的安全性直接关系到企业的商业利益、客户信任乃至市场竞争力。信息系统安全的核心目标在于确保数据的保密性、完整性和可用性：保密性要求客户信息、价格策略等敏感数据仅对授权人员开放，避免因信息泄露造成商业损失；完整性旨在保证订单数量、发货记录等数据在传输和存储过程中不被篡改，防止业务流程因数据错误出现混乱；可用性则需确保系统在面对网络攻击、系统故障或其他突发事件时均能稳定运行，保证系统的服务能够持续提供，不中断企业的正常业务活动。

SD 系统的各模块间存在紧密的数据关联，例如客户创建后自动同步至订单管理模块，订单生成后实时触发库存检查等。这种高度集成的特性意味着任何一个环节的安全漏洞都可能引发连锁反应：若客户主数据被未授权访问，可能导致竞争对手获取客户资源；若订单数据被恶意修改，可能造成发货数量与合同不符，引发客户纠纷；若财务记录出现异常，可能影响企业资金管理的准确性。因此，SD 系统的安全设计需覆盖从数据录入到最终归档的全流程，通过技术手段与管理制度的结合，构建多层次、全方位的安全防护体系，为企业的销售与分销业务提供坚实保障。​

## 8.2 人员安全管理​

人员作为系统操作的主体，其行为直接影响信息安全，因此人员安全管理是 SD 系统安全设计的基础环节。在制度设计上，需严格遵循授权最小化、分散化和规范化原则，确保每个操作人员仅能获得完成本职工作所必需的最小权限。例如，销售代表的权限应限定在创建本区域客户、录入订单和生成报价单范围内，无权修改客户的付款条件或查看其他区域的订单数据；仓库管理员仅能处理发货流程和库存查询，无法参与发票生成或收款记录录入；财务人员则专注于发票管理和付款对账，不涉及订单创建或库存调整。这种权限划分不仅能减少误操作的风险，更能从源头防范内部人员的恶意行为。​

对于关键任务，需通过功能拆分实现多人协同，避免单一人员掌握完整操作权限。以大额订单的审批流程为例，销售人员仅能发起订单申请，需由销售经理审核价格合理性，再经财务主管确认客户信用额度，最终由仓库主管安排发货，通过多岗位的相互制约确保业务合规。同时，企业需建立严格的权限生命周期管理制度：新员工入职时，需由部门主管提交权限申请，经 IT 部门审核后分配对应角色权限；员工岗位变动时，IT 部门需在 24 小时内调整其权限范围；员工离职时，立即冻结账号并回收所有权限，防止账号被滥用。此外，定期开展安全培训也至关重要，通过案例讲解、操作演练等方式，提升员工的安全意识，例如识别钓鱼邮件、妥善保管账号密码等，从人员层面筑牢安全防线。​

## 8.3 用户标识与认证​

用户标识与认证是防止未授权用户进入系统的第一道技术防线，其核心在于确保用户身份的唯一性与合法性。在用户标识设计上，SD 系统采用员工工号作为唯一账号，账号格式包含部门标识和序号， 其中标识“SL” 代表销售部、“WH” 代表仓库、“FI” 代表财务，和序号则如 “SL-001”“WH-002”，既便于身份识别，又能直观反映用户所属部门。系统会为每个账号关联如 销售代表、仓库管理员的角色信息，角色与权限矩阵绑定，确保用户仅能访问其角色范围内的功能模块。

在授权机制方面，系统采用了基于角色的访问控制（RBAC）模型，不同的用户角色（如管理员、采购员、仓库管理员等）拥有不同的权限集，以确保各用户只能执行与其角色权限相符的操作。系统通过Django的权限控制装饰器，精确地限制了视图函数和类视图的访问权限，确保敏感操作只能由具有相应权限的用户执行。

在认证机制方面，系统采用了标准的用户名/密码组合作为用户身份验证的基础。为增强密码安全性，系统在存储用户密码时，使用了SHA-256等安全哈希算法进行加密处理。系统强制要求密码长度不低于 8 位，包含大小写字母、数字和特殊字符（如 “Sales@2024!”），且每 90 天需强制更换，同时禁止使用近 5 次内的历史密码。对于修改客户付款条件一类的高风险操作，系统启用多因素认证，用户在输入密码后，还需通过手机短信验证码或企业微信扫码完成二次验证，进一步降低账号被盗用的风险。为增强登录安全性，系统设置了多项防护措施：连续 5 次输错密码时，账号自动锁定 30 分钟；从非常用 IP 地址访问进行异地登录时，系统会发送提醒邮件至用户邮箱，用户确认后才可继续操作；登录成功后，系统自动显示最近登录记录，包括时间、IP 地址，便于用户发现异常登录。通过这些措施，既能确保合法用户便捷访问系统，又能有效阻挡未授权用户的非法入侵。​

from django.contrib.auth.models import User

from django.contrib.auth import authenticate

# 用户认证代码示例

username = request.POST['username']

password = request.POST['password']

user = authenticate(request, username=username, password=password)

if user is not None:

# 认证成功，用户已登录

login(request, user)

else:

# 认证失败，返回错误信息

return HttpResponse('Invalid login credentials')

## 8.4 数据完整性与有效性控制​

数据完整性与有效性是确保 SD 系统业务流程准确运行的关键，需通过技术手段防止数据被篡改、破坏或丢失。本系统主要从数据输入与验证、数据存储与处理、数据输出与安全三个方面对数据完整性与有效性进行控制。

### 8.4.1数据输入与验证

在本系统的设计中，数据输入是保证业务流程正常运作的第一步。用户通过系统界面录入各类业务数据，包括客户信息、订单信息、发货信息和发票等。为了确保录入数据的准确性和可靠性，系统对输入数据进行了严格的验证和校验。

具体而言，系统采用了Django框架中的模型验证机制，通过自定义验证器对用户输入的数据进行实时检查。例如，在客户信息模块中，系统通过自定义的手机号验证器（validate\_phone）确保输入的电话号码符合规范，同时使用邮政编码验证器（validate\_digit）确保邮政编码的有效性和一致性。此外，为了防止数据冗余和潜在的业务冲突，系统对户名称（cname）等关键字段设置了唯一性约束。这种设计不仅提高了数据的准确性，还有效防止了重复记录的产生，从而保障了系统数据的一致性和完整性。

通过以上验证机制的实施，系统能够在数据录入的早期阶段就防范潜在的错误，确保后续业务流程能够基于准确、可靠的数据顺利执行。

### 8.4.2数据存储与处理

本系统的数据存储采用了基于Django ORM（对象关系映射）的设计模式，利用关系型数据库存储和管理各类业务数据。系统设计了多个核心数据表，其中包括EUser（用户信息）、Customer（客户信息）、SalesOrder（销售订单）、GoodDelivery（发货信息）、Invoice（发票信息）以及Account（财务记账）等。每个数据表通过外键关系紧密相连，形成了一个结构化的数据网络，确保了业务数据的内在一致性。

在数据处理方面，系统实现了各业务模块间的高效协同。举例来说，在用户生成发货订单时，系统会自动触发相应的库存更新操作，记录在MaterialItem和StockHistory表中。同时，系统还会在订单生成后，记录采购的详细项目信息，并在订单确认后生成收货记录。这一系列操作通过事务管理机制保证了数据的一致性，避免了因并发操作或系统故障导致的数据不一致问题。

此外，在发票生成模块中，系统根据发货信息和发货订单自动生成相应的财务记录，并将这些记录存储在Account和AccountDetail表中。这种数据处理的自动化不仅减少了人工干预的必要性，还提高了业务流程的效率和准确性

### 8.4.3数据输出与安全

数据输出是本系统中一个至关重要的环节，直接影响到用户的决策支持能力。系统提供了多种数据输出形式，包括详细的业务报表、订单确认单、库存状态报告和财务对账单等。这些报表和文档通过系统的格式化处理，确保了数据的可读性和逻辑性，使用户能够方便地获取所需的业务信息。

为了保护系统中存储和处理的敏感数据，系统在数据安全性设计方面也采取了多层次的保护措施。首先，所有涉及用户身份验证的密码数据都经过SHA-256哈希算法加密后存储，从而防止了潜在的密码泄露风险。其次，系统采用了严格的角色权限管理机制，确保不同用户只能访问和操作与其权限相符的数据。通过这种精细化的权限控制，系统有效防止了未经授权的数据访问和修改，保障了企业核心数据的安全性。

此外，系统还定期进行数据备份，以应对可能的系统故障或数据丢失情况。在发生数据损坏或丢失时，系统能够通过备份数据进行快速恢复，确保业务连续性和数据完整性。这些安全措施共同构成了系统数据处理过程中的强有力保障，为企业的日常运营提供了可靠的技术支持。

## 8.5 逻辑访问控制​

逻辑访问控制基于系统的安全机制，确定用户或进程对特定资源的访问权限，是防止越权操作的核心手段。SD 系统采用基于角色的访问控制（RBAC）模型，预先定义角色集合（如销售代表、仓库主管、财务专员），每个角色关联明确的权限清单。例如，销售代表的权限包括 “创建客户”“录入订单”“查询库存”，但无 “修改发票金额”“审批折扣” 的权限；仓库主管可 “处理发货”“确认出库”，但无法 “删除订单记录”。权限分配需经过严格审批，由部门主管提交申请，IT 部门审核后执行，且每季度进行权限审计，回收不再需要的权限。​

数据加密是保护敏感信息的重要技术，本系统通过一系列的加密措施，确保了数据在存储和传输过程中的安全性，防止未经授权的访问和数据泄露。

系统首先对敏感信息如用户密码、密保答案等进行加密处理。采用SHA-256等哈希算法，并结合加盐技术，以增强密码的抗破解能力。这种加密策略有效防止了彩虹表攻击等常见解密手段。为了进一步提升数据的安全性，系统还可以利用数据库自带的加密功能，确保即便数据库文件被泄露，敏感信息也不会被轻易破解。

在传输安全方面，系统采用了HTTPS协议来确保客户端和服务器之间的数据传输受到SSL/TLS加密保护。通过这种方式，系统有效防止了数据在传输过程中遭受窃听或中间人攻击。此外，为防止会话劫持和重放攻击，系统使用了加密令牌（如JWT）来验证请求的合法性，进一步保障了数据传输的安全性。

## 8.6 审计与跟踪​

审计与跟踪系统，维护一个或多个系统运行的日志记录文件，记录系统应用和维护活动等用户活动，是进行信息系统安全控制的重要手段。本SD 系统的审计日志需覆盖全业务流程，记录内容包括：用户账号、操作时间、操作模块、操作类型、操作内容、操作结果、客户端 IP 地址等。日志采用不可篡改的格式存储，写入后即变为只读状态，防止被删除或修改。​

日志管理需建立规范流程，日志文件由专门的审计服务器存储，保留期限为 1 年，仅安全管理员有权查看和导出。系统每日生成审计报告，重点分析异常操作：如凌晨时段的批量订单修改、不同账号在同一 IP 地址登录、短时间内多次失败的付款记录录入等。对于高风险操作，系统会实时推送告警至安全管理员，管理员需在 24 小时内核查并记录结果。定期开展日志审计，每月由内部审计部门抽查日志记录，检查是否存在越权操作、违规修改等情况，发现问题及时追溯责任并优化安全策略。通过完善的审计与跟踪机制，既能威慑潜在的恶意行为，又能为安全事件的事后分析提供可靠依据。​

# 9 SD 系统实施费用估算报告

在信息系统的实施过程中，准确的费用估算是确保项目成功的关键因素之一。本文基于《管理信息系统》教材中经济可行性分析框架，结合 SD 系统的开发需求，在估算过程中，参考了多项行业报告和学术研究，通过分析各类从软件到硬件到人员等成本，力求对信息系统实施费用进行科学、合理的评估。

## 9.1软件开发成本

软件开发作为 SD 系统实施的核心环节，涉及多岗位协同作业。依据轻量级erp开发团队配置，结合本项目一个月的开发周期以及项目团队成员，具体人员配置及费用如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 岗位 | 人数 | 月薪（元） | 开发周期 | 总费用（元） | 职责说明 |
| 项目经理 | 1 | 18000 | 1 个月 | 18000 | 全面统筹开发流程，协调各模块开发进度，确保客户管理、订单处理等业务流程顺畅衔接、 |
| 业务分析师 | 1 | 12000 | 1个月 | 12000 | 明确详细的业务需求，分析现有业务流程中的痛点与问题，为 ERP 系统的功能设计提供准确依据。 |
| 系统架构师 | 1 | 16000 | 1 个月 | 16000 | 负责设计系统整体架构，保障模块间数据交互高效稳定，如实现订单与库存数据的实时联动 |
| 后端开发工程师 | 3 | 12000 | 1 个月 | 12000 | 专注开发客户管理、订单创建、财务对账等核心业务逻辑，为系统功能实现提供后端支持 |
| 前端开发工程师 | 3 | 12000 | 1 个月 | 12000 | 负责设计并搭建 Web 界面，打造简洁易用的操作界面，提升用户体验 |
| 测试工程师 | 1 | 10000 | 1 个月 | 10000 | 针对系统各项功能编写测试用例，全面验证订单流程、库存扣减等功能的准确性与稳定性 |
| 小计  （人力成本） | - | - | - | 80000 | - |

此外，开发过程中所需的工具及软件许可费用，参考行业信息按开发成本的 5% 估算：

工具与许可费用：80000×5% = 4000 元

软件开发总成本：80000 +4000 = 84000 元

## 9.2硬件设备采购

硬件设备需满足系统部署、数据存储及网络互联等关键需求。参考 2024 年华为云设备采购指南及市场行情，具体配置及费用如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 设备类型 | 规格参数 | 单价（元） | 数量 | 总费用（元） | 用途说明 |
| 应用服务器 | 8 核 16G，支持高并发访问 | 45000 | 1 | 45000 | 用于部署 SD 系统前端界面及后端服务，保障系统流畅运行 |
| 数据库服务器 | 16 核 32G，支持 TB 级存储 | 50000 | 1 | 50000 | 负责存储客户信息、订单数据、库存记录等关键业务数据 |
| 网络设备 | 企业级交换机 + 防火墙 | 15000 | 1 | 15000 | 构建安全稳定的网络环境，确保服务器与前端界面间的数据安全传输 |
| 备份设备 | 自动备份系统，支持数据恢复 | 20000 | 1 | 20000 | 定期备份业务数据，防止订单、财务等重要数据丢失，保障数据安全 |
| 硬件设备总费用 | - | - | - | 130000 | - |

## 9.3人员培训成本

系统上线前，针对销售、仓库、财务等关键岗位人员开展操作培训至关重要。依据行业报告，企业培训费用呈上升趋势。考虑到本项目涉及的业务流程复杂度，培训费用估算如下：

**·**培训对象：销售代表（4 人）、仓库管理员（3 人）、财务人员（2 人）、管理人员（2 人），共 11人

**·**培训费：参考市场行情，每人培训费用按 2000 元计算，总计 2000×11 = 22000 元

培训内容将全面覆盖客户创建流程、订单录入操作、库存查询方法、发票生成步骤等系统核心操作，确保员工能够熟练运用系统，提升工作效率。

## 9.4系统维护和支持成本

系统上线后，为保障其长期稳定运行，需持续投入维护资源。系统维护成本通常占开发成本的10%至15%之间，按保守的 10% 估算：

维护与支持费用（首年）：80000×10% = 8000 元

涵盖内容包括日常漏洞修复，如处理订单数据异常；功能优化，如根据业务需求新增折扣计算逻辑；以及为用户提供技术支持，及时解决操作问题等，确保系统稳定运行。

## 9.5其他间接成本

其他间接成本包含项目管理费用，如组织开发会议、使用项目进度跟踪工具；文档编制费用，用于编写用户手册、开发文档；以及用户反馈收集与系统改进费用等。参考项目管理学会（PMI）2024 年发布的相关资料，间接成本一般占总开发成本的 8%-10%，取中间值 9% 估算：

间接成本：80000×9% = 7200 元

## 9.6总费用估算

|  |  |  |
| --- | --- | --- |
| 费用类别 | 金额（元） | 占比 |
| 软件开发成本（含工具） | 80000 | 32.36% |
| 硬件设备采购 | 130000 | 52.59% |
| 人员培训成本 | 22000 | 8.90% |
| 系统维护和支持 | 8000 | 3.23% |
| 其他间接成本 | 7200 | 2.91% |
| 总费用 | 247200 | 100% |

# 10参考文献

[1]刘仲英主编；王洪伟，吴冰副主编；刘仲英，王洪伟，吴冰，邵建利，徐德华，李沁芳编；薛华成主审.“十二五”普通高等教育本科国家级规划教材 管理信息系统 第3版[M].北京：高等教育出版社,2017

[2]《SAP ERP Materials Management case study》

[3]Pamela J. Schmidt,Ronald D. Freeze. (2019). Enterprise Job Roles and Resistance to ERP Use: Actual Usage as an Antecedent to ERP Resistance. Hawaii International Conference on System Sciences.

[4]M. Haddara. (2022). ERP systems selection in multinational enterprises: a practical guide. International Journal of Information Systems and Project Management.

[5]Ying Gao,Qin Su,Yue Fang,Pinyan Lai. (2020). Construction of Cloud ERP Security Evaluation Index System Based on Text Mining..

[6]张巍钟,张震. (2012). ERP系统权限管理. 中国管理信息化(3).