

体系结构描述文档

灯具照明行业进销存管理系统



2014-10-14

小百合工作室

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 陈云龙 | 2014-10-14 | 最初草稿 | V1.0 |
| 陈云龙 | 2014-10-15 | 部分文档 | V1.1 |
| 曾婧 | 2014-10-19 | 添加开发包图 | V1.2 |
| 景诗维  陈云龙 | 2014-10-22 | 添加用户界面层分解、数据层分解和信息视角 | V1.3 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 1](#_Toc401802261)

[1.1. 编制目的 1](#_Toc401802262)

[2. 产品描述 1](#_Toc401802263)

[3. 逻辑视角 1](#_Toc401802264)

[4. 组合视角 2](#_Toc401802265)

[4.1. 开发包图 2](#_Toc401802266)

[4.2. 运行时进程 6](#_Toc401802267)

[4.3. 物理部署 7](#_Toc401802268)

[5. 接口视角 7](#_Toc401802269)

[5.1. 模块的职责 7](#_Toc401802270)

[5.2. 用户界面层的分解 7](#_Toc401802271)

[5.2.1. 用户界面层模块的职责 8](#_Toc401802272)

[5.2.2. 用户界面模块的接口规范 8](#_Toc401802273)

[5.2.3. 用户界面模块设计原理 9](#_Toc401802274)

[5.3. 业务逻辑层的分解 9](#_Toc401802275)

[5.4. 数据层的分解 9](#_Toc401802276)

[5.4.1. 数据层模块的职责 9](#_Toc401802277)

[5.4.2. 数据层模块的接口规范 9](#_Toc401802278)

[6. 信息视角 13](#_Toc401802279)

[6.1. 数据持久化对象 13](#_Toc401802280)

[6.2. 文件格式 14](#_Toc401802281)

# 引言

## 编制目的

本报告详细完成对灯具照明行业进销存系统的概要设计，达到指导详细设计和开发的目的，同时实现测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户编写，是了解系统的导航。

# 产品描述

参考灯具照明行业进销存系统用例文档和灯具照明行业进销存系统软件规格说明中对产品的概括描述。

# 逻辑视角

灯具照明行业进销存系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图1和图二所示。

图 1参照体系结构风格的暴徒表达逻辑视角

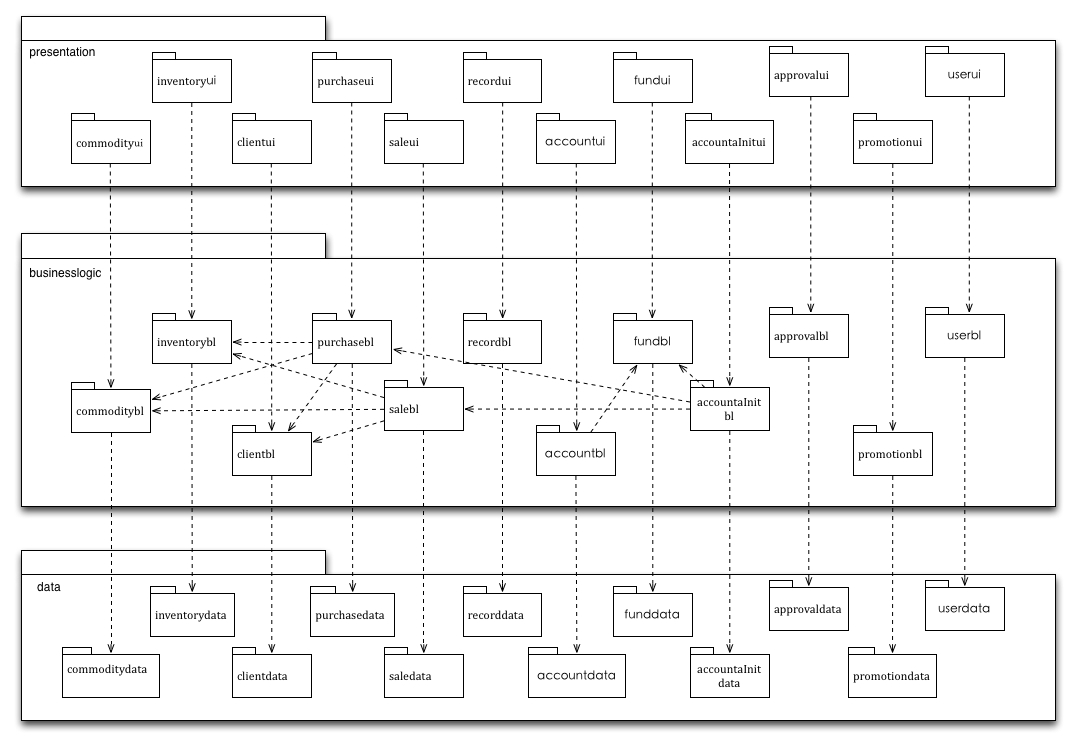


图 2 软件体系结构逻辑设计方案

# 组合视角

## 开发包图

灯具照明行业进销存系统的最终开发包设计如表1所示

**表1 灯具照明行业进销存系统的最终开发包设计**

|  |  |
| --- | --- |
| 开发（物理包） | 依赖的其他开发包 |
| mainui | commodityui, inventoryui, clientui, purchaseui, saleui, recordui, accountui, approvalui, promotionui, accountainitui, userui, fundui, vo |
| commodityui | commodityblservice, vo, 界面类库包 |
| commodityblservice | vo |
| commoditybl | commodityblservice, po |
| commoditydataservice | Java RMI, po |
| commoditydata | commoditydataservice, databaseutility |
| inventoryui | inventoryblservice, vo, 界面类库包 |
| inventoryblservice | vo |
| inventorybl | inventoryblservice, po |
| inventorydataservice | Java RMI, po |
| inventorydata | inventorydataservice, databaseutility |
| clientui | clientblservice, vo, 界面类库包 |
| clientblservice | vo |
| clientbl | clientblservice, po |
| clientdataservice | Java RMI, po |
| clientdata | clientdataservice, databaseutility |
| purchaseui | purchaseblservice, vo, 界面类库包 |
| purchaseblservice | vo |
| purchasebl | purchaseblservice,inventorybl,commoditybl, clientbl, po |
| purchasedataservice | Java RMI, po |
| purchasedata | purchasedataservice, databaseutility |
| saleui | saleblservice, vo, 界面类库包 |
| saleblservice | vo |
| salebl | saleblservice, inventorybl, commoditybl, clientbl, po |
| saledataservice | Java RMI, po |
| saledata | saledataservice, databaseutility |
| recordui | recordblservice, vo, 界面类库包 |
| recordblservice | vo |
| recordbl | recordblservice, po |
| recorddataservice | Java RMI, po |
| recorddata | recorddataservice, databaseutility |
| accountui | accountblservice, vo, 界面类库包 |
| accountblservice | vo |
| accountbl | accountblservice, fundbl, po |
| accountdataservice | Java RMI, po |
| accountdata | accountdataservice, databaseutility |
| fundui | fundblservice, vo, 界面类库包 |
| fundblservice | vo |
| fundbl | fundblservice, po |
| funddataservice | Java RMI, po |
| funddata | funddataservice, databaseutility |
| accountaintui | accountaintblservice, vo, 界面类库包 |
| accountaintblservice | vo |
| accountaintbl | accountaintblservice, fundbl, purchasebl, salebl, po |
| accountaintdataservice | Java RMI, po |
| accountaintdata | accountaintdataservice, databaseutility |
| approvalui | approvalblservice, vo, 界面类库包 |
| approvalblservice | vo |
| approvalbl | approvalblservice, po |
| approvaldataservice | Java RMI, po |
| approvaldata | approvaldataservice, databaseutility |
| promotionui | promotionblservice, vo, 界面类库包 |
| promotionblservice | vo |
| promotionbl | promotionblservice, po |
| promotiondataservice | Java RMI, po |
| promotiondata | promotiondataservice, databaseutility |
| userui | userblservice, vo, 界面类库包 |
| userblservice | vo |
| userbl | userblservice, po |
| userdataservice | Java RMI, po |
| userdata | userdataservice, databaseutility |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility |  |

****

图 3 客户端开发包图



图 4 服务器端开发包图

## 运行时进程

在灯具照明行业进销存系统中，会有多个客户端进程和一个服务器端进程，其进程图如图5所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行。



图 5 进程图

## 物理部署

灯具照明行业进销存系统中客户端构件是放在客户端机上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于Java RMI 构件属于JDK 1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如图6所示。



图 6 部署图

# 接口视角

## 模块的职责

## 用户界面层的分解

系统存在30个用户界面,他们分别为登录界面、库存管理人员界面组（库存管理人员主界面、商品分类管理界面、商品管理界面、库存查看界面、商品详细信息界面、商品分类详细信息界面、库存赠送界面、警报类单据界面以及库存盘点界面）、进货销售人员界面组（进货销售人员主界面、制定进货单界面、制定进货退货单界面、制定销售单界面、制定销售退货单界面、客户管理界面以及历史单据查看界面）、财务人员界面组（财务人员主界面、制定现金单界面、制定收款单界面、制定付款单界面、销售明细表界面、经营历程表界面、经营情况表界面、账户管理界面、期初建账界面）、总经理界面组（总经理主界面、审批单据以及促销策略界面）以及管理员界面组（用户权限与信息管理界面）。其用户界面之间的跳转如下图所示：



其中所有用户界面通过同一个类处理，即主用户界面类（MainFrame）



### 用户界面层模块的职责

MainFrame主要用于负责界面的显示和界面之间的跳转工作。

### 用户界面模块的接口规范

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | init(args:String[]) |
| 前置条件 | 用户触发相应需要显示界面的条件 |
| 后置条件 | 显示Frame并且加载Panel |

### 用户界面模块设计原理

用户界面你用java的Swing和AWT库来实现。

## 业务逻辑层的分解

## 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。持久化数据保存采用序列化文件

### 数据层模块的职责

数据层模块职责如下表所示

**数据层模块职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| InventoryDataService | 提供库存单据的提交服务（赠送单、报溢单、报损单、报警单） |
| PurchaseDataService | 提供进货数据集体载入、保存、增加、删除、查找 |
| RecordDataService | 这里是查看操作，是不是就直接调用其他类的方法就可以了，比如查看销售明细表就直接调用销售里查看销售单方法 |
| FundDataService | 这个是什么东西…… |
| ApprovalDataService | 提供单据的集体载入、保存、查看修改操作 |
| UserDataService | 提供用户集体载入、保存、增删改查服务 |
| CommodityData | 提供商品集体载入、保存、增删改查服务 |
| SaleDataService | 提供销售数据集体载入、保存、增加、删除、查找 |
| AccountDataService | 提供账户数据集体载入、保存、增删改查服务 |
| ClientDataService | 提供客户数据集体载入、保存、增删改查服务 |
| AccountaInitData | 提供期初建账数据集体载入、保存、增加、查看服务 |
| PromotionDataService | 这块有什么东西需要保存么？ |

### 数据层模块的接口规范

数据层模块的接口规范如下表

**数据层模块的接口规范**

InventoryDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| InventoryDataService.submit | 语法 | public boolean submit(InventoryBillPO po) |
| 前置条件 | 无 |
| 后置条件 | 返回提交是否成功 |

PurchaseDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PurchaseDataService.insert | 语法 | public boolean insert(PurchasePO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| PurchaseDataService.find | 语法 | public boolean find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| PurchaseDataService.delete | 语法 | public boolean delete (PurchasePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| PurchaseDataService.update | 语法 | public boolean update (PurchasePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| PurchaseDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

ApprovalDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ApprovalDataService.update | 语法 | public boolean update (PersistentObject po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| ApprovalDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

UserDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataService.insert | 语法 | public boolean insert(UserPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| UserDataService.find | 语法 | public boolean find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| UserDataService.delete | 语法 | public boolean delete (UserPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| UserDataService.update | 语法 | public boolean update (UserPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| UserDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

CommodityDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CommodityDataService.insert | 语法 | public boolean insert(CommodityPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| CommodityDataService.find | 语法 | public boolean find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| CommodityDataService.delete | 语法 | public boolean delete (CommodityPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| CommodityDataService.update | 语法 | public boolean update (CommodityPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| CommodityDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

SaleDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SaleDataService.insert | 语法 | public boolean insert(SalePO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| SaleDataService.find | 语法 | public boolean find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| SaleDataService.delete | 语法 | public boolean delete (SalePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| SaleDataService.update | 语法 | public boolean update (SalePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| SaleDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

AccountDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AccountDataService.insert | 语法 | public boolean insert(AccountPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| AccountDataService.find | 语法 | public boolean find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| AccountDataService.delete | 语法 | public boolean delete (AccountPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| AccountDataService.update | 语法 | public boolean update (AccountPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| AccountDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

ClientDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ClientDataService.insert | 语法 | public boolean insert(ClientPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| ClientDataService.find | 语法 | public boolean find(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| ClientDataService.delete | 语法 | public boolean delete (ClientPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| ClientDataService.update | 语法 | public boolean update (ClientPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| ClientDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

AccountaInitDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AccountaInitDataService.insert | 语法 | public boolean insert(AccountaInitPO po) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| AccountaInitDataService.init | 语法 | public void init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

# 信息视角

## 数据持久化对象

系统的PO类就是对应的相关的实体类，如下所示。

* PersistentObject 所有po类的父类，包括时间，编号
* UserPO包括用户名、密码、身份属性。
* ClientPO包括编号、分类（进货商、销售商）、级别（五级，一级普通用户，五级VIP客户）、姓名、电话、地址、邮编、电子邮箱、应收额度、应收、应付、默认业务员。
* CommodityPO包括编号、名称、型号、库存数量、进价、零售价、最近进价、最近

零售价。

* PurchasePO保存进货时候数据的类，包括时间、状态、单据编号（格式为：JHD-yyyyMMdd-xxxxx，后五位每天从1开始编号，所以一天最多可以生成99999条单子），供应商，仓库，操作员，入库商品列表，备注，总额合计。其中入库商品列表包含的信息有：商品编号，名称（从商品选择界面进行选择），型号，数量（手动输入），单价（默认为商品信息中的进价），金额，备注（手动输入）。没有下划线的部分是自动计算并填充进去的
* SalesPO就是保存销售时候数据的类，包括时间、状态、单据编号（XSD-yyyyMMdd-xxxxx），客户（仅显示销售商），业务员（和这个客户打交道的公司员工，可以设置一个客户的默认业务员），操作员（当前登录系统的用户），仓库，出货商品清单，折让前总额，折让，使用代金卷金额，折让后总额，备注。出货商品清单中要显示商品的编号，名称（从商品选择界面选择），型号，数量（手工输入），单价（默认为商品信息里的销售价，可修改），金额（自动生成），商品备注。
* AccountPO包括创建时间、账户名称和金额
* AccountProceedsBillPO 包括时间、状态，单据编号（SKD-yyyyMMdd-xxxxx），客户（同时包含供应商和销售商），操作员（当前登录用户），转账列表，总额汇总。转账列表中的一项包含：银行账户，转账金额，备注
* AccountPaymentBillPO 包括时间、状态、单据编号（XJFYD-yyyyMMdd-xxxxx）,操作员（当前登录用户），银行账户，条目清单，总额。条目清单中包括：条目名，金额，备注。
* InventoryBillPO 包括时间、单据类型、商品、商品数量。
* AccountaInitPO 包括创建时间、添加商品信息（商品分类，某一商品的名称，类别，型号，进价/售价(默认为上年平均)，最近进价和最近售价留空），客户信息（客户分类，某一个客户的名称，联系方式等，应收应付(之前遗留)），银行账户信息（名称，余额）。

## 文件格式

统一用\*.dat格式，比如Client.dat