目录

[更新历史 2](#_Toc463380462)

[一、 引言 3](#_Toc463380463)

[(一) 编制目的 3](#_Toc463380464)

[(二) 词汇表 3](#_Toc463380465)

[(三) 参考资料 3](#_Toc463380466)

[二、 产品描述 4](#_Toc463380467)

[三、 逻辑视角 5](#_Toc463380468)

[四、 组合视角 6](#_Toc463380469)

[(一) 开发包图 6](#_Toc463380470)

[(二) 运行时进程 6](#_Toc463380471)

[(三) 物理部署 6](#_Toc463380472)

[五、 接口视角 7](#_Toc463380473)

[(一) 模块的职责 7](#_Toc463380474)

[(二) 用户界面层的分解 9](#_Toc463380475)

[1. 用户界面层模块的职责 9](#_Toc463380476)

[2. 用户界面模块的接口规范 9](#_Toc463380477)

[3. 用户界面模块设计原理 10](#_Toc463380478)

[(三) 业务逻辑层的分解 10](#_Toc463380479)

[1. 业务逻辑层模块的职责 10](#_Toc463380480)

[2. 业务逻辑模块的接口模范 10](#_Toc463380481)

[(四) 数据层的分解 10](#_Toc463380482)

[1. 数据层模块的职责 10](#_Toc463380483)

[2. 数据层模块的接口规范 11](#_Toc463380484)

[六、 信息视角 11](#_Toc463380485)

[(一) 数据持久化对象 11](#_Toc463380486)

[(二) 文件格式 11](#_Toc463380487)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 冯俊杰 | 2016/10/4 | 最初草稿 |  |
|  |  |  |  |

# 引言

## 编制目的

本报告详细完成对互联网酒店预订系统的概要设计，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。

## 词汇表

|  |  |
| --- | --- |
| **缩写或单词** | **解释** |
| ELMS | 互联网酒店预订系统 |
| loadDoc | 装车单 |
| arriveZZDoc | 到达单（营业厅和中转中心流转） |
| transferDoc | 中转单 |
| arriveYYDoc | 接收单 |
| sendGoodDoc | 派件单 |
| Bill | 期初账单 |
| order | 订单 |
| inStoreDoc | 入库单 |
| outStoreDoc | 出库单 |
| orderBarCode | 订单的条形码号 |
| DTManage | 司机车辆信息管理 |

## 参考资料

1. 互联网酒店预订系统用例文档
2. 互联网酒店预订系统需求规格说明文档

# 产品描述

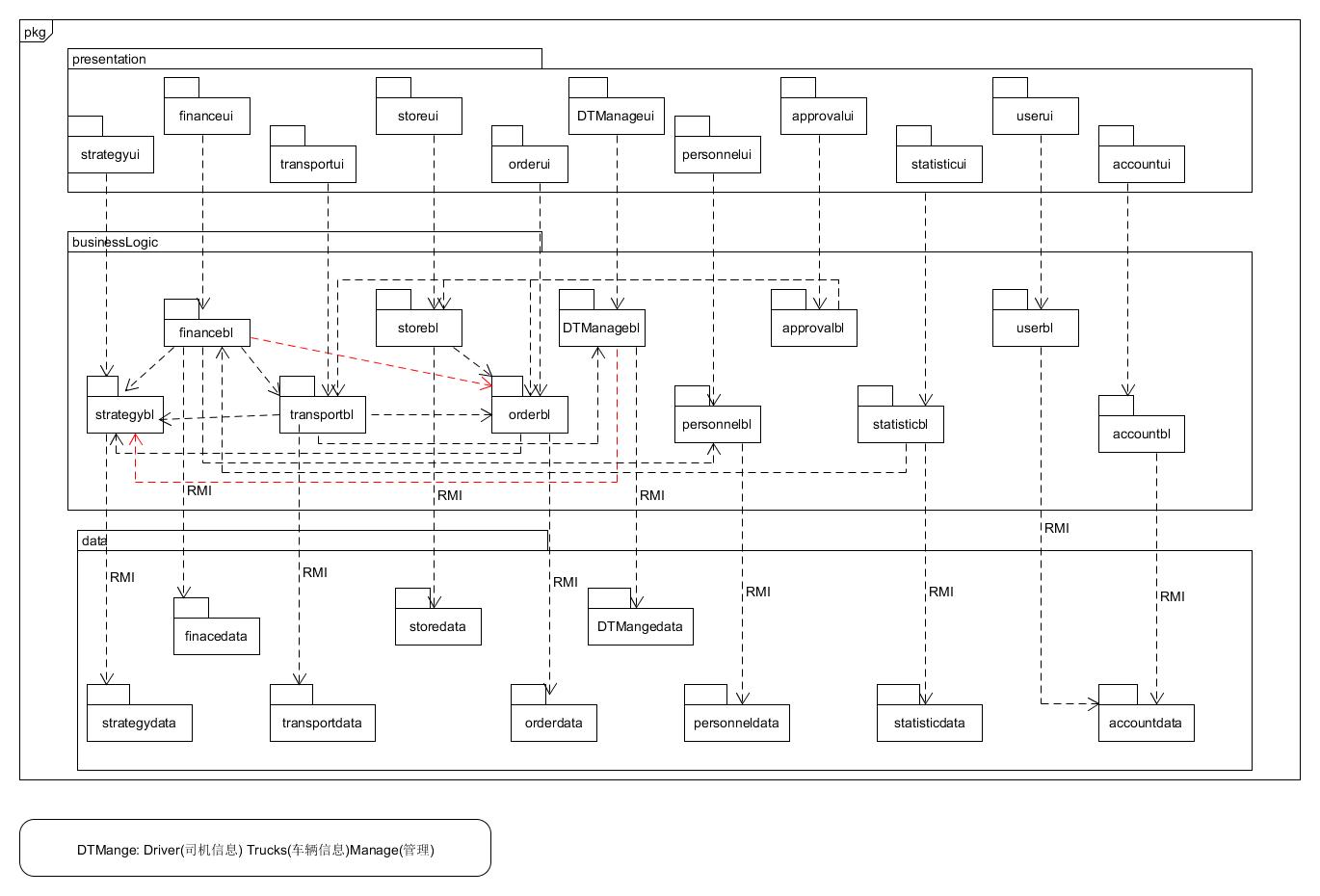
参考《互联网酒店预订系统用例文档》和《互联网酒店预订系统需求规格说明文档》中对产品的概括描述。

# 逻辑视角

互联网酒店预订系统中，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2所示。



**图1：分层结构**



**图2：逻辑设计方案**

# 组合视角

## 开发包图

互联网酒店预订系统的最终开发包设计如表1所示：

表1 互联网酒店预订系统

|  |  |
| --- | --- |
| mainui | userui, hotelui, memberui, orderui promotionui , marketerui, creditui, vo |
| userui | userblservice , 界面类库包 |
| userblservice |  |
| userbl | userblservice, userdataservice, po, memberbl |
| userdataservice | Java RMI, po |
| userdata | Databaseutility, po, userdataservice |
| Hotelui | Hotelblservice, 界面类库包 |
| Hotelblservice |  |
| Hotelbl | Hotelblservice, Hoteldataservice, po, promotionbl, orderbl |
| Hoteldataservice | Java RMI, po |
| Hoteldata | Databaseutility, po, Hoteldataservice |
| Memberui | Memberblservice, 界面类库包 |
| Memberblservice |  |
| Memberbl | Memberblservice, Memberdataservice, po |
| Memberdataservice | Java RMI, po |
| Memberdata | Databaseutility, po, Memberdataservice |
| Orderui | Orderblservice, 界面类库包, vo |
| Orderblservice |  |
| Orderbl | Orderblservice, Orderdataservice, po, userbl, hotelbl, memberbl, promotionbl |
| Orderdataservice | Java RMI, po |
| Orderdata | Databaseutility, po, Orderdataservice |
| Promotionui | Promotionblservice, 界面类库包 |
| Promotionblservice |  |
| Promotionbl | Promotionblservice, Promotiondataservice, po |
| Promotiondataservice | Java RMI, po |
| Promotiondata | Databaseutility, po, Promotiondataservice |
| marketerui | marketerblservice, 界面类库包, vo |
| marketerblservice |  |
| marketerbl | marketerblservice, marketerdataservice, po, orderbl, memberbl, promotionbl |
| marketerdataservice | Java RMI, po |
| marketerdata | Databaseutility, po, Webworkerdataservice |
| creditui | creditblservice, 界面类库包, vo |
| creditblservice |  |
| creditbl | creditblservice, creditdataservice, po, hotelbl, guestbl, marketerbl |
| creditdataservice | Java RMI, po |
| creditdata | Databaseutility, po, creditdataservice |
| Vo |  |
| Po |  |
| Utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| Databaseutility | JDBC |

## 运行时进程

在互联网酒店预订系统中，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。

## 物理部署

互联网酒店预订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。

# 接口视角

## 模块的职责

客户端模块和服务器模块视图分别如图7和图8所示。客户端各层和服务器端各层的职责分别如表2和表3所示。

**启动模块**

**业务逻辑层**

**展示层**

**网络模块**

**图7 客户端模块视图**

**网络模块**

**启动模块**

**业务逻辑层**

**图8 服务器端模块视图**

**表2 客户端各层的职责**

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的进销存管理系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 客户端网络模块 | 利用java RMI机制查找RMI服务 |

**表3 服务器端各层的职责**

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 数据层 | 负责数据的持久化及数据访问接口 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| UserMesbl  Accountbl  Personnelbl  Statisticbl  Approvalbl  Stretegybl  Orderbl  Financebl  Storebl  DTmanagebl  transportbl | 客户端展示层 | 客户端业务逻辑层 |
| Accountdataservice  StatisticDataService  PersonnelDataservice  StretegyDataService  financedataservice  storedataservice  DTmanageDataService  TransportDataService | 客户端业务逻辑层 | 服务器端数据层 |

## 用户界面层的分解

根据需求，系统存在38个用户界面（加粗为主界面）：**登录界面**、**查询订单界面**、**管理员主界面**、**总经理界面**、机构管理界面、制定常量界面、制定薪水策略界面、人员管理界面、审批单据界面、统计分析界面、**财务人员界面**、交易审核界面、增加成本收益表界面、期初建账界面、银行账户管理界面、新建付款单界面、成本管理界面、财务人员主界面、**快递员主界面**、增加订单界面、查询订单界面、收件信息录入界面、**营业厅业务员主界面**、装车单界面、派送单界面、接收单界面、收款单界面、车辆信息管理界面、司机信息管理界面、**中转中心业务员界面**、**中转中心仓库管理人员界面**、到达单界面、中转单界面、库存查看界面、报警恢复界面、设置报警值界面、库存盘点界面、新建入库单界面、新建出库单界面。

### 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 界面frame，负责界面的显示和界面的跳转 |

### 用户界面模块的接口规范

1. **用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 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LoginPanel |

1. **用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| bl.accountservice | 登录界面的业务逻辑接口 |
| bl.\*blservice | 每个ui模块会有对应的数据逻辑接口 |

### 用户界面模块设计原理

用户界面拟用java的Swing和AWT库来实现。

## 业务逻辑层的分解

业务逻辑层包括多个针对界面及业务逻辑处理对象。例如User对象负责处理登录界面

的业务逻辑。业务逻辑层的设计如下图所示：

### 业务逻辑层模块的职责

### 业务逻辑模块的接口模范

## 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。例如，AccountBusinessService的服务主要有AccountDataService提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。以AccountDataService为例，数据层模块的具体描述下图所示：

### 数据层模块的职责

### 数据层模块的接口规范

# 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。本系统共包含23个PO类，在此

不一一赘述，只做部分介绍：

AccountPO类为账户信息，包括ID、密码、账户类型、个人其他信息。

ConstPO类为系统常量，包括城市间距离，交通方式运费，包装费用等等。

DocPO类为所有单据父类，储存了单据ID、生成时间、单据类型、单据状态。

持久化对象ReceivePO类的定义如图所示：

## 文件格式

本系统依据不同情况分别采用数据库、序列化、txt文件持久化保存三种方式保存文件