**快递物流系统**

**体系结构设计文档**

学 院：南京大学软件学院

成 员：潘凌伟 申彬 谭昕玥 杨雁飞

完成日期：2015 年10月26日

**目录**

[更新历史 3](#_Toc433658202)

[1、引言 4](#_Toc433658203)

[1.1 编制目的 4](#_Toc433658204)

[1.2 词汇表 5](#_Toc433658205)

[1.3 参考资料 6](#_Toc433658206)

[2、产品概述 6](#_Toc433658207)

[3、逻辑视角 7](#_Toc433658208)

[4、组合视角 9](#_Toc433658209)

[4.1 开发包图 10](#_Toc433658210)

[4.2 运行时进程 15](#_Toc433658211)

[4.3 物理部署 16](#_Toc433658212)

[5、接口视角 17](#_Toc433658213)

[5.1 模块的职责 17](#_Toc433658214)

[5.2 用户界面层的分解 20](#_Toc433658215)

[5.2.1 用户界面层模块的职责 22](#_Toc433658216)

[5.2.2用户界面层模块的接口规范 22](#_Toc433658217)

[用户界面层需要的服务接口如表5.2.2-2所示。 25](#_Toc433658218)

[5.2.3 用户界面模块设计原理 25](#_Toc433658219)

[5.3 业务逻辑层的分解 25](#_Toc433658220)

[5.3.1 业务逻辑层模块的职责 26](#_Toc433658221)

[5.3.2 业务逻辑层模块的接口规范 27](#_Toc433658222)

[5.4 数据层的分解 52](#_Toc433658223)

[5.4.1 数据层模块的职责 53](#_Toc433658224)

[5.4.2 数据层模块的接口规范 53](#_Toc433658225)

[6、信息视角 67](#_Toc433658226)

[6.1 数据持久化对象 67](#_Toc433658227)

[6.2 数据库表 72](#_Toc433658228)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 全体人员 | 2015-10-17 | 最初草稿（创建文档模板） | V1.0 |
| 谭昕玥 | 2015-12-30 | 总经理制定城市距离用例修改 | v1.0.1 |

# 1、引言

## 1.1 编制目的

本报表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，同时实现和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而编写，是了解系统的导航。

## 1.2 词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |
| RMI | 表示远程方法调用 |  |

## 1.3 参考资料

1. IEEE std 1471-2000
2. 丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版2012：134—182

# 2、产品概述

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软件需求规格说明文档中对产品的概括描述。互联网酒店预订系统主要是应用于客户方便快捷的在网络预订酒店，主要功能见用例图如下。

****

# 3、逻辑视角

互联网酒店预订系统中，选择了分层体系结构风格，将系统分为3层(展示层、业务逻辑层、数据层)能够很好地示意整个高层抽象。展示层包含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。



图一 参照体系结构风格的包图表达逻辑视角



图二 软件体系结构逻辑设计方案

# 4、组合视角

[软工2 166]与抽象的逻辑设计相比，实现物理设计要考虑更多的实现细节，这些细节有：

1. presentation层与logic层被置于客户端，data层被置于服务器端，那么logic层的开发包不可能依赖于data层的开发包。使用RMI技术，将data层开发包分解为置于客户端的dataservice接口包和置于服务器端的data层开发包。这样一来，logic层开发包依赖于dataservice包，dataservice包和data层的开发包都依赖于RMI类库包。
2. 所有的data层开发包都需要进行数据持久化（例如读写数据库、读写文件等），所以它们会有一些重复代码，可以将重复代码独立为新的开发包，然后所有的data层开发包都依赖于databasetility。datavaseutility会依赖于JDBC类库包。
3. 所有的presentation层开发包都需要使用图形类型建立界面，都要依赖于图形界面类库包。
4. 此外，presentation层实现时，由mainui包负责整个页面之间的跳转逻辑。其他各包负责各自页面自身的功能。
5. 在分层风格的典型设计中，不希望高层直接依赖于低层，而是为低层建立接口包，实现依赖倒置原则，所以应该调整为：各presentation层开发包（调用）依赖于logic层接口包businesslogicservice包，logic层开发包也依赖于（实现了）logic层接口包businesslogicservice包。
6. 在分层风格的典型设计中，presentation层与logic层之间、logic层与data层之间可能会传递复杂数据对象，那么相邻两层都需要使用数据对象声明，所以需要将数据对象声明独立为开发包（VO包和PO包）。
7. 使用依赖倒置原则消除包的循环依赖现象，将循环依赖变为单向依赖：

* Order和Manage:将部分Manege类抽象接口manegeInfoService置入Order包，这样Manage单向依赖于Order（实现接口+调用）
* Inventory和Manage:将部分Inventory类抽象接口inventoryInfoService置入Manage包，这样Inventory单向依赖于Manage（实现接口+调用）

8）在logic层中，初始化和业务逻辑层上下文的工作被分配到utility包中。

经过细节改进，最终建立的互联网酒店预订系统开发包设计如表4.1-1，其局部包图如图4.1-1和4.1-2所示：

## 4.1 开发包图

表4.1-1 快递物流系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| Mainui | OrderHanui,OrderViewui,Userui,Promotionui,Searchui,Hotelui |
| OrderHanui | OrderHanblservice,界面类库包,vo |
| OrderHanbl | OrderHanblservice,OderHandataservice,po,Userbl,Promotionbl  Hotelbl |
| OrderHanblservice |  |
| OrderHandata | databaseutility,po, OrderHandataservice |
| OrderHandataservice | Java RMI,po |
| OrderViewui | OrderViewblservice,界面类库包,vo |
| OrderViewbl | OrderViewblservice,OderViewdataservice,po |
| OrderViewblservice |  |
| OrderViewdata | databaseutility,po, OrderViewdataservice |
| OrderViewdataservice | Java RMI,po |
| Userui | Userblservice,界面类库包,vo |
| Userbl | Userblservice,Userdataservice,po, |
| Userblservice |  |
| Userdata | databaseutility,po, Userdataservice |
| Userdataservice | Java RMI,po |
| Promotionui | Promotionblservice,界面类库包,vo |
| Promotionbl | Promotionblservice,Promotiondataservice,po, |
| Promotionblservice |  |
| Promotiondata | databaseutility,po, Promotiondataservice |
| Promotiondataservice | Java RMI,po |
| Searchui | Searchblservice,界面类库包,vo |
| Searchbl | Searchblservice,Searchdataservice,po,Userbl,Hotelbl |
| Searchblservice |  |
| Searchdata | databaseutility,po, Searchdataservice |
| Searchdataservice | Java RMI,po |
| Hotelui | Hotelblservice,界面类库包,vo |
| Hotelbl | Hotelblservice,Hoteldataservice,po, |
| Hotelblservice |  |
| Hoteldata | databaseutility,po, Hoteldataservice |
| Hoteldataservice | Java RMI,po |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| Databaseutility | JDBC |

图4.1-1 互联网酒店预订系统客户端开发包图



图4.1-2 互联网酒店预订系统服务器端开发包图

## 4.2 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4.2所示。结合部署图，客户端进程实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4.2 进程图

## 4.3 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器上，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JavaRMI构件属于JDK6.0的一部分。所以，在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图4.3所示。



4.3 部署图

# 5、接口视角

## 5.1 模块的职责

客户端模块和服务器端模块视图分别如图5.1-1和图5.1-2所示。客户端各层和服务器端各层的职责分别如表5.1-1和表5.1-2所示。



图5.1-1 客户端模块视图



图5.1-2 服务器端模块视图

表5.1-1 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 | 基于窗口的进销存系统客户端用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务 |

表5.1-2 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5.1-3所示。

表5.1-3 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| order\_blservice  user\_blservice  promotion\_blservice  search\_blservice  hotel\_blservice | 客户端展示层 | 客户端业务逻辑层 |
| order\_data\_service  user\_data\_service  promotion\_data\_service  search\_data\_service  hotel\_data\_service | 客户端业务逻辑层 | 服务器端数据层 |

借用用户管理用例来说明层之间的调用，如图5.1-3所示。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。UserBLService提供了User界面所需要的所有业务逻辑功能UserDataService提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。



图5.1-3 用户管理用例层之间调用的接口

## 5.2 用户界面层的分解

根据需求，系统存在19个用户界面：登录界面、客户主界面、酒店工作人员主界面、网站营销人员主界面、网站管理人员主界面、个人信息界面、酒店详情界面、酒店搜索界面、酒店搜索记录界面、酒店评价界面、会员登记界面、酒店信息维护界面、客房信息维护界面、酒店优惠政策制定界面、信用值修改界面、网站优惠政策制定界面、调整用户界面、订单浏览界面、订单处理界面。

界面跳转如图5.2所示。



图5.2 用户界面跳转

服务器端和客户端的用户界面设计接口是一致的，只是具体的页面不一样。用户界面类如图5.2-2所示。



图5.2-2 用户界面类

### 5.2.1 用户界面层模块的职责