面向商场的智能推荐系统

软件构架文档

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 |
| 23/11/2015 | 1.0 | 添加软件架构设计(逻辑视图) | 石嘉昊 |
| 23/11/2015 | 1.1 | 添加简介及用例视图 | 周炯昱 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 用例视图 4

2.1 用例实现 4

2.2 用例展示（部分） 5

2*.2.1* 输入订单 5

2*.2.2* 查询工作单 6

2*.2.3* 查询订单流程信息 6

2*.2.4* 利润查询 7

3. 逻辑视图 8

3.1 概述 8

3.2 系统第零级架构设计 9

3.3 详细架构描述 10

3*.3.1* UI\_Package 10

3*.3.2* Action\_Package 13

3*.3.3* Logic\_Package 13

3*.3.4* Hibernate\_Package 14

4. 部署视图 17

5. 实现视图 17

5.1 概述 17

6. 数据视图 18

6.1 概述 18

7. 约束条件 20

8. 质量目标 20

软件构架文档（简化版）

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面做出的重要决策。

项目成员应根据此文档进行详细设计和开发。

## 范围

此文档适用于“运输运力管理系统”项目。

## 定义、首字母缩写词和缩略语

配送方：负责配送的部门人员

工作单：记录工人数量，车队ID等具体工作信息

路线单：记录车队的具体路线

调度单：记录托运单对应的工作单和路线单

托运单：根据配送单生成的具体运送信息

配送单：配送方提供

订单：即配送单

## 参考资料

1. 《立项申请书》
2. 《SRS需求规约》
3. 《软件开发计划》
4. 《风险列表》

## 概述

此软件架构文档包含以下内容：

* 用例视图：列举用例模型中的部分用例对其进行描述
* 逻辑视图：说明设计模型在架构方便具有重要意义的部分
* 部署视图：说明用来部署和运行该软件的一种或多种物理网络（硬件）配置
* 实现视图：说明实现模型的整体结构、软件分解为实现模型中的层和子系统的情况，以及所有在构架方面具有重要意义的构件
* 数据视图：从永久性数据存储方面来对系统进行说明

# 用例视图

## 用例实现

本系统主要以配送方的actor输入配送单和运输系统内部人员查询维护信息为主。

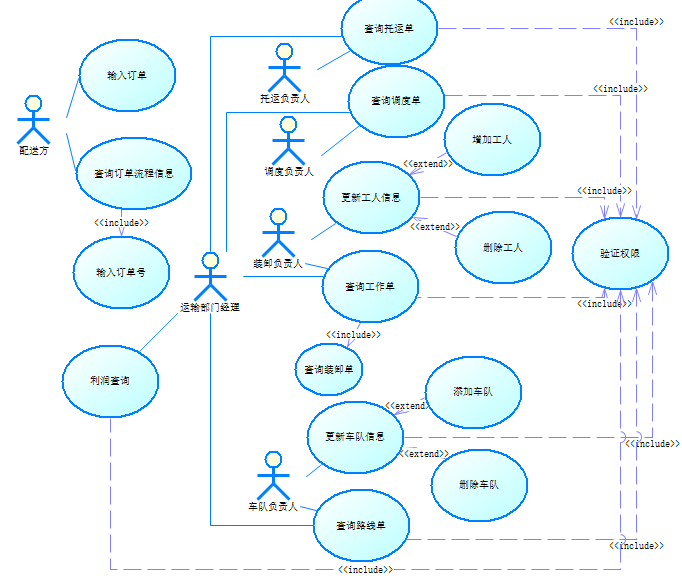


图1系统用例图

## 用例展示（部分）

### 输入订单

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号： | UC101 | 用例名称 | 输入订单 |
| 描述 | 配送方输入订单，运输系统开始运作 | | |
| 执行者 | 配送方 | | |
| 前置条件 | 无 | | |
| 后置条件 | 无 | | |
| 基本流 | 1. 配送方选择输入订单 2. 系统要求输入详细信息 3. 输入详细信息 4. 系统返回唯一订单编号，提示成功 5. 配送方退出系统，用例结束 | | |
| 备选流 | 1-2.a 用户选择退出系统  用例结束，系统退出  4.b 系统提示生成订单失败  系统提示重新检查信息返回3  5.b配送选择继续添加订单，返回1 | | |
| 扩展点 | 待定 | | |
| 非功能需求 | 系统响应客户时间不超过3秒 | | |
| 业务规则 |  | | |

### 查询工作单

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号： | UC204 | 用例名称 | 查询工作单 |
| 描述 | 装卸负责人或运输部经理查询工作单详情 | | |
| 执行者 | 装卸负责人或运输部经理 | | |
| 前置条件 | 无 | | |
| 后置条件 | 无 | | |
| 基本流 | 1. 用户选择查询工作单 2. 系统验证权限 3. 系统提示权限验证成功 4. 用户选择根据工作单id查询详细信息 5. 系统显示详细信息 6. 用户选择退出系统，用例结束 | | |
| 备选流 | 1.a 用户选择退出系统  用例结束，系统退出  3.b 系统提示权限不够，系统退出  4.b用户选择根据时间查询工作单  5.b 系统提示工作单id错误，返回4重新输入  6.c 用户选择继续查询，返回4 | | |
| 扩展点 | 待定 | | |
| 非功能需求 | 系统响应客户时间不超过3秒 | | |
| 业务规则 |  | | |

### 查询订单流程信息

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号： | UC102 | 用例名称 | 查询订单流程信息 |
| 描述 | 配送方查询已下订单的流程信息 | | |
| 执行者 | 配送方 | | |
| 前置条件 | 无 | | |
| 后置条件 | 无 | | |
| 基本流 | 1. 用户选择查询订单流程信息 2. 系统提示输入订单号 3. 系统提示查询成功，输出该订单信息 4. 用户选择退出系统，用例结束 | | |
| 备选流 | 1.a 用户选择退出系统  用例结束，系统退出  3.b 系统提示订单号不存在，重新输入订单号，返回2  4.b用户选择继续查询订单，返回2 | | |
| 扩展点 | 待定 | | |
| 非功能需求 | 系统响应客户时间不超过3秒 | | |
| 业务规则 |  | | |

### 利润查询

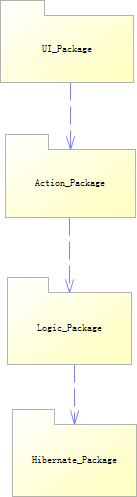
|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号： | UC205 | 用例名称 | 利润查询 |
| 描述 | 运输部经理查询利润 | | |
| 执行者 | 运输部经理 | | |
| 前置条件 | 无 | | |
| 后置条件 | 无 | | |
| 基本流 | 1. 用户选择查询利润 2. 系统验证权限 3. 系统提示权限验证成功 4. 用户选择查询当天利润 5. 系统提示查询成功，显示详细信息 6. 用户选择退出系统，用例结束 | | |
| 备选流 | 1.a 用户选择退出系统  用例结束，系统退出  3.b 系统提示权限不够，系统退出  4.b用户选择查询其他日期的利润  5.c 用户选择继续查询，返回4 | | |
| 扩展点 | 待定 | | |
| 非功能需求 | 系统响应客户时间不超过3秒 | | |
| 业务规则 |  | | |

# 逻辑视图

## 概述

本系统可作为公司系统的子系统，亦可作为公司管理系统的子模块。以下设计为按照子系统设计使用SSH框架的B/S架构。由于View模块（即UI模块）与逻辑模块的分离，也可以通过简单的底层修改使本系统作为运输模块融入主系统的C/S架构中。

## 系统第零级架构设计



**系统0层架构设计图**

系统主要分成四个模块，分别是UI模块，动作模块，逻辑模块，Hibernate持久化模块。与一般的ssh框架相比，添加了中间的逻辑模块，将复杂的动作逻辑按照流程拆分成小模块，增加代码的复用度和容错率。

## 详细架构描述

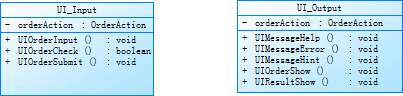
### UI\_Package



**UI包组件**

UI包分为3个子包作为其组件，分别是：UI传输动作包、UI查询动作包、UI管理动作包。其中UI传输包具有显示订单输入、开始运输按键、显示运输流程及结果的功能；UI查询动作包提供显示指定的数据库数据的功能；UI管理动作包辅助管理人员对于车辆等需要管理的数据进行更改和修复等操作。

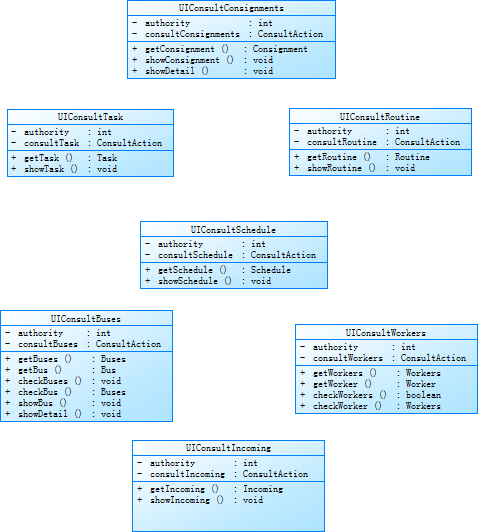
#### UITransportPackage



**UITransportPackage类图**

UITransportPackage包对应用户输入提交订单界面，其中主要包含用户提交订单时的输入输出类，包括对用户订单的检查、提交，以及返回的提示信息、订单所处的流程信息的显示等功能。

#### UIConsultPackage



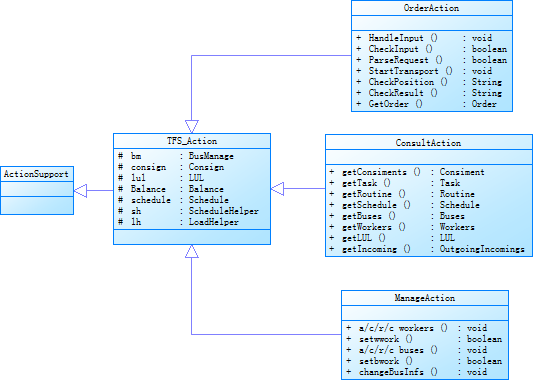
**UIConsultPackage类图**

UIConsultPackage包对应用户/系统工作人员查询各类相关信息的界面。

#### UIManagePackage



**UIManagePackage类图**

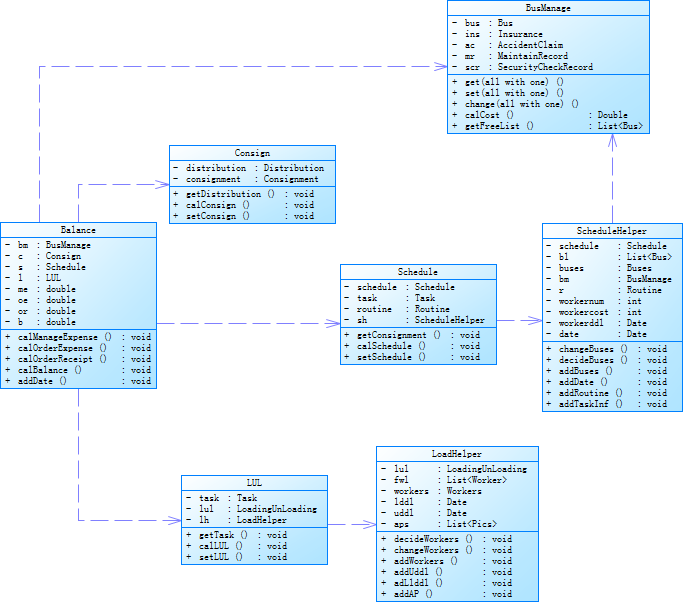
UIManagePackage主要对应管理人员对于车辆、工队、车队信息的登记与修改界面，其中主要包含对于这三者数据详细信息的增删改操作按键。

### Action\_Package

**ActionPackage类图**

Action包为前端UI提供了对应的动作实现。

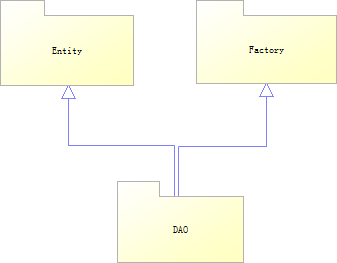
### Logic\_Package



**LogicPackage图**

逻辑包按照运输流程将协作操作、调度操作以及数据库操作分解为小模块，以供Action包相关类方法组合调用实现具体动作。

### Hibernate\_Package

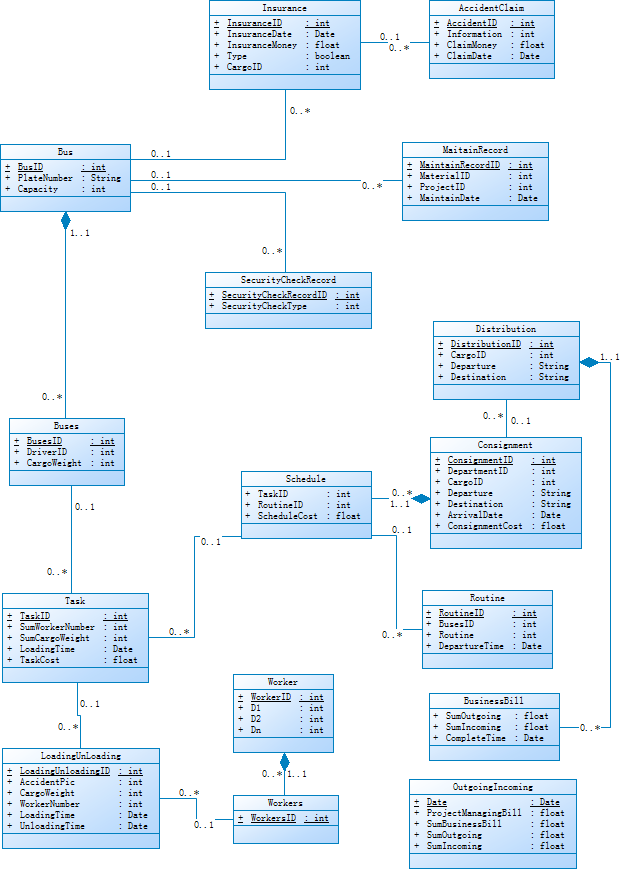


**Hibernate包组件**

Hibernate包包含了三个包，Entity为实体类、DAO为数据库交互类、Factory为工厂类，用于产生Session。

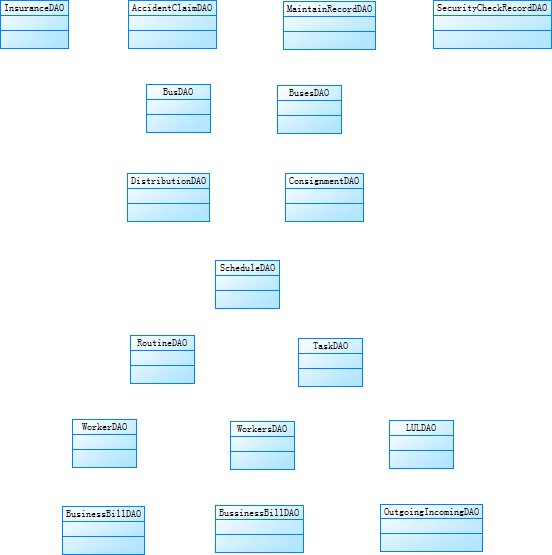
#### Entity Package

Entity包包含了各个实体类，将数据库中每一个表抽象为一个类，将对数据库表的操作转化为对实体对象的操作。

f

**Entity类图**

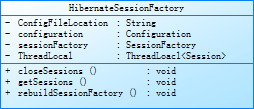
#### DAO Package



**DAO类图**

DAO包包含了对各个实体类的数据访问接口。

#### Factory Package



**Factory类图**

HibernateSessionFactory类用于获得Session、关闭Session、获得SessionFactory以及获得配置信息等。

# 部署视图



图14系统部署视图

由系统部署视图可以知道，系统需要提供两台服务器，一台为web服务器，用于运行系统的后台，一台为数据库服务器，用于保存系统的数据信息。同时客户端Android系统设备通过无线网络访问后台服务器，如果需要数据访问，则访问后台数据库服务器。

# 实现视图

## 概述

本系统采用C/S框架搭建架构体系。C代表客户端，S代表服务器端。客户端主要负责向服务器端请求数据与提交数据，服务器端主要负责相应客户端请求，并进行相关数据库操作，服务器端数据库实现采用Hibernate,将数据库表抽象为实体类，通过操作实体类达到操作数据库的目的。服务器端与APP客户端通过HTTP进行交互，使用JSON字符串，方便服务器端和APP进行数据封装和数据解析。

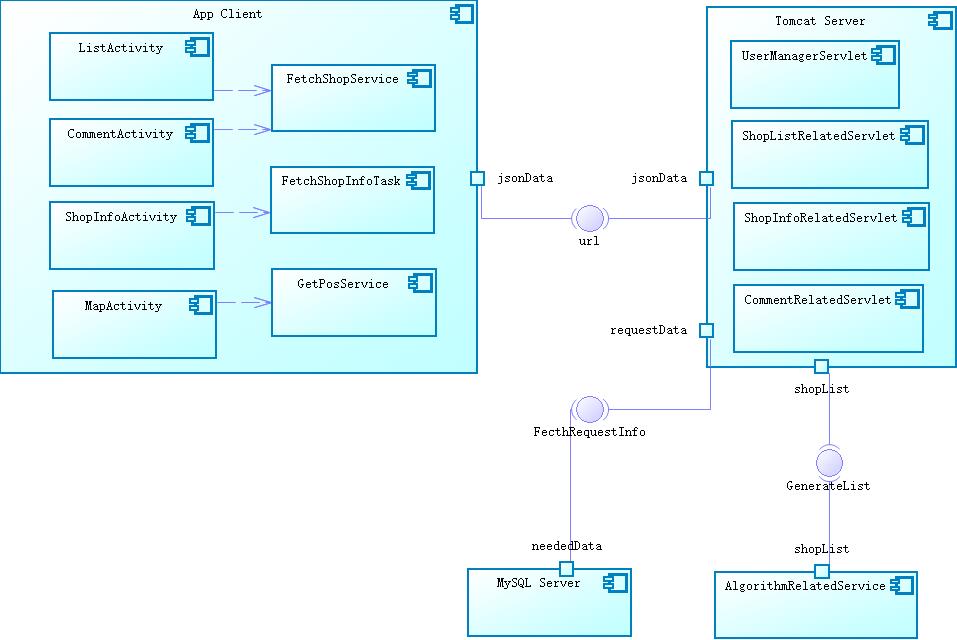


图12系统实现视图

# 数据视图

## 概述

本系统的数据库设计中User、Shop、Goods、BehaviorPoint分别表示用户、店铺、商品、用户行为点，ShoppingBehavior、BrowsingBehavior、RegistraionBehavior分别代表用户的购买商品行为、浏览商品行为、用户签到行为。本系统的物理数据模型图3所示。

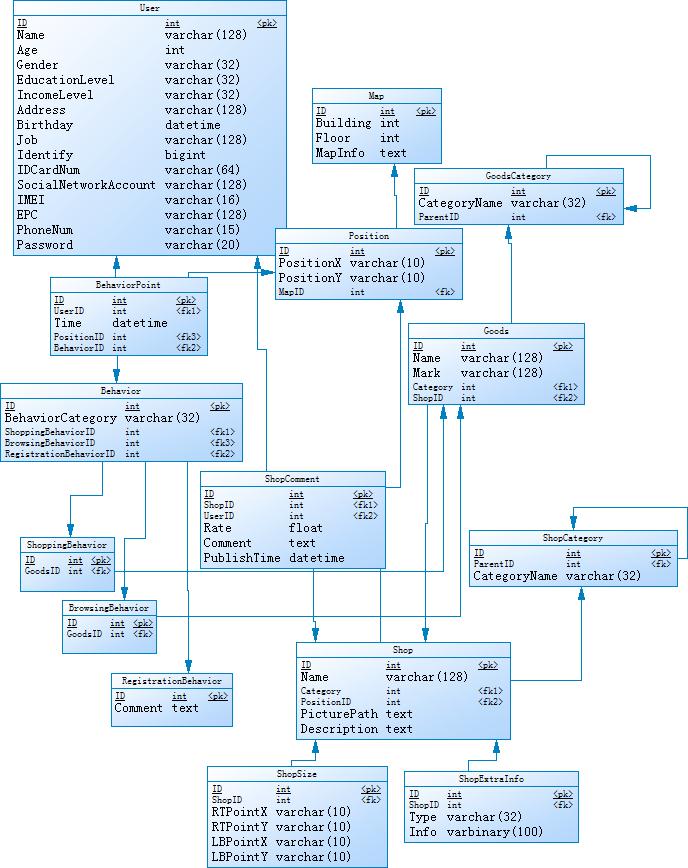


图14系统数据物理模型图

# 约束条件

文档描述的面向商场的智能推荐系统服务器端采用Java语言进行开发，客户端在Android平台同样采用Java语言进行开发。

平台提供用户在商场中实时查看自己位置、查看店铺信息、对店铺进行评论以及智能为用户推荐店铺。

小组目前有6名成员，开发周期为2014-10-27 ~ 2015-1-10两个半月

# 质量目标

可用性：系统要求操作简单易懂，普通用户无需参加培训即可使用本系统的基本功能；系统的界面应明确展示产品功能，方便使用者使用。

可靠性：系统平均故障间隔时间至少6个月；系统平均修复时间不超过2天

性能：网络状况正常的情况下A加载时间不超过2s，平台事物处理时间不超过3s，信息检索时不超过3s；系统能够承载1000个用户同时访问

可支持性：系统应易于维护，代码编写应符合JAVA编写规范