快递管理系统 MSE(Management System of Expressage)

软件详细设描述文档

V1.0

完整版

一组

成员：吕德超、李昊朔、阙俊杰、罗铉斌

2015-11-5

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 全体人员 | 2015-10-21 | 最初草稿（不完整） | V1.0 草稿 |

# 引言

## 编制目的

本报告详细完成对快递管理系统的详细设计，达到指导后续软件构造的目的，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。

## 词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| MSE | 快递管理系统 |  |
| checklist | 订单查询 | 订单查询 |
| listin | 单据输入 | 订单输入、收件人信息输入 |
| transfer | 中转接收 | 接收与派件、中转接收、录入中转单 |
| load | 装运管理 | 车辆装车管理、装运管理 |
| vehicle | 车辆管理 | 司机信息管理、车辆信息管理 |
| finance | 财务管理 | 收款单管理，结算管理，成本管理，查看成本收益表，统计报表，查看经营情况 |
| store | 仓库管理 | 入库管理，出库管理，库存管理 |
| bill | 期初建账 | 期初建账 |
| account | 账户管理 | 账户管理，帐户查询 |
| staff | 员工管理 | 人员机构管理，用户信息管理 |
| policy | 策略管理 | 策略管理 |
| approve | 审批单据 | 审批单据 |
| user | 用户 |  |
| order | 订单 |  |
| receiver | 收件人信息 |  |
| loadorder | 装车单 |  |
| driver | 司机信息 |  |
| arriveorder | 到达单 | 中转中心到达单 |
| receiveorder | 接收单 | 营业厅接收单 |
| sendorder | 派件单 |  |
| payeeorder | 收款单 |  |
| changeorder | 中转单 |  |
| storeinorder | 入库单 |  |
| storeoutorder | 出库单 |  |
| payorder | 付款单 |  |
| startinfo | 期初信息 |  |
| salary | 薪水信息 |  |
| constant | 常量信息 |  |

## 参考资料

1. IEEE标准
2. 快递管理系统MSE用例文档
3. 快递管理系统MSE结构逻辑设计文档

# 产品概述

参考快递管理系统用例文档和快递管理系统软件需求规格说明文档中对产品的概括描述。

# 体系结构设计概述

参考快递管理系统概要设计文档中对体系结构文档的概述。

# 结构视角

## 业务逻辑层的分解

业务逻辑层的开发包图参见软件体系结构文档图3

### Userbl模块

#### 4.1.1.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Checklistbl模块

#### 4.1.2.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Listinbl模块

#### 4.1.3.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.3.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Transferbl模块

#### 4.1.4.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.4.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.4.3具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.4.4具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Loadbl模块

#### 4.1.5.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.5.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Vehiclebl模块

#### 4.1.6.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.6.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Financebl模块

#### 4.1.7.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.7.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.7.3具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.7.4具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Storebl模块

#### 4.1.8.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.8.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.8.3具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Billbl模块

#### 4.1.9.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Accountbl模块

#### 4.1.10.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Staffbl模块

#### 4.1.11.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Policybl模块

#### 4.1.12.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

#### 4.1.12.2具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

### Approvebl模块

#### 4.1.13.1具体用例

###### 模块描述

**自己补全**

###### 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如逻辑层和业务逻辑层之间，**自己补全**

###### 模块内部类的接口规范

**自己补全**

###### 业务逻辑层的动态模型

**自己补全**

###### 业务逻辑层的设计原理

**自己补全**

# 依赖视角



图3 快递管理系统客户端开发包图



图4 快递管理系统服务器端开发包图