**中南财经政法大学**

**信息系统分析与设计**

**课程报告**

快递管理信息系统

分析与设计

|  |  |
| --- | --- |
| **组 长** | **项英（1309030118）** |
| **组 员** | **郭雪（1309030129）**  **陈雅楠（1309030139）**  **曾子轩（1309030109）**  **许子吟（1309030119）**  **李和嫒（1309030121）** |
| **专 业** | **信息管理与信息系统(2013级)** |
| **指导教师** | **余传明** |
| **完成日期** | **2016年1月** |

目 录

[第一部分 项目管理与计划 2](#_Toc439678256)

[实验1 指定项目计划 2](#_Toc439678257)

[实验2 项目可行性分析 5](#_Toc439678258)

[第二部分 需求分析 8](#_Toc439678259)

[实验3 项目需求收集 8](#_Toc439678260)

[实验4 用例建模 11](#_Toc439678261)

[实验5 通过用例获取概念数据模型 24](#_Toc439678262)

[实验6 将概念数据模型转化为对象关系模型 25](#_Toc439678263)

[实验7 分析类图建模 25](#_Toc439678264)

[实验8三种可选方案选择分析 46](#_Toc439678265)

[第三部分 系统设计 52](#_Toc439678266)

[实验9物理数据库设计 52](#_Toc439678267)

[实验10确定系统构架等设计元素、设计类图建模 59](#_Toc439678268)

[实验11界面设计 67](#_Toc439678269)

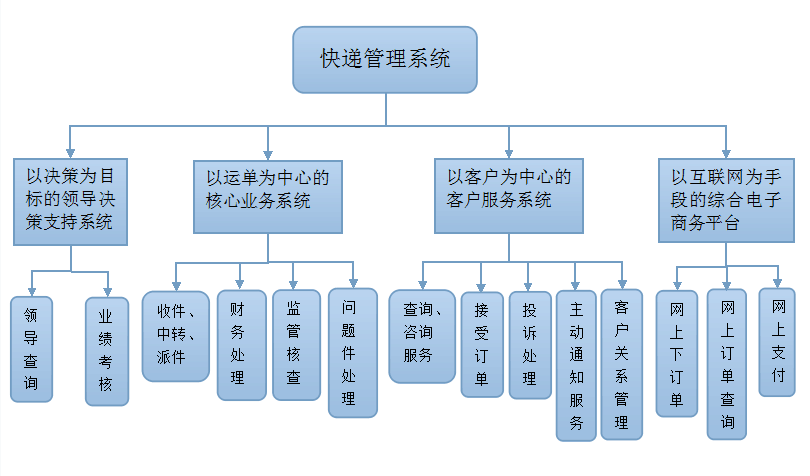
[实验12 代码实现 74](#_Toc439678270)

# 第一部分 项目管理与计划

## *实验1 指定项目计划*

* 1. 项目的工作分解结构（WBS），所需资源

|  |  |
| --- | --- |
| WBS | 所需资源 |
| 1 需求开发  1.1 需求获取  1.2 需求定义  1.3 需求验证  1.4 需求规格说明书 | 一定的用户群体需求调查资料 |
| 2 总体设计  2.1 系统方案设计  2.2 总体进度规划  2.3 风险评估 | 同类别系统相关信息 |
| 3 详细设计  3.1 结构设计  3.2 人机界面设计  3.3 接口设计  3.4 网络构架设计  3.5 数据库设计 | 相应的设计人员 |
| 4 设计实施  4.1 用户管理模块  4.1.1 添加用户  4.1.2 修改个人信息  4.1.3 删除用户  4.1.4 系统退出  4.2 人事管理模块  4.2.1 领导查询  4.2.2 业绩考核  4.3 业务管理模块  4.3.1 寄件下单  4.3.2 物流查询服务  4.3.3 主动通知服务  4.3.4 投诉处理  4.3.5 客户关系管理  4.4 快件管理模块  4.4.1 收件、中转、派件  4.4.2 财务处理  4.4.3 监管核查  4.4.4 问题件处理  4.5 下单操作  4.5.1 购物车管理  4.5.2 结算管理  4.6 订单管理  4.6.1 处理订单  4.6.2 修改订单  4.6.3 查找订单  4.6.4 删除订单 | XX快递公司信息  基础业务信息  实施人员  管理人员 |
| 5 测试  5.1 集成测试  5.2 环境测试  5.3 系统测试  5.4 缺陷跟踪  5.5 测试总结 | 测试工具，环境 |
| 6 交付  6.1 验收测试  6.2 产品提交  6.3 用户培训 | 交洽人员，场地 |



整个快递管理信息系统分为四大部分：以决策为目的的领导决策支持系统、以运单为中心的核心业务系统、以客户为中心的客户服务系统和以互联网为手段的综合电子商务平台，其中，核心业务系统是重点。下面展开讲述以下这四个部分的具体功能：

1. 以决策为目的的领导决策支持系统

1)领导查询：原则上，普通员工能看到的信息领导都可以看到，领导桌面的信息可定制。2)业绩考核：根据设定的管理指标，实时反映员工业绩情况。

1. 以运单为中心的核心业务系统

1)收件：短信通知业务员收件、运单录入。2)中转：中转到件、中转发件、中转报关。3)到件：派件清单、派件回单（标志快件周期 结束）。4)监管核查：对每一单实时跟踪，及时发现问题，报警，对运单状态全程监控。5)财务处理：发票、对账单、应收／应付、核查、收／付款、财务核销等。6)问题件处理：对各种问题件进行分类处理。

1. 以客户为中心的客户服务系统

1)查询咨询服务：可通过电话、Internet等方式查询运单状态，咨询业务。2)投诉处理：受理对服务的投诉、赔偿要求等。3)接受定单：通过电话、传真等方式接受定单委托、预约等。4)主动通知服务：定制服务要求。如：货到后通过短信通知发货方。5)客户关系管理：客户的基本资料、历史交易情

1. 以互联网为手段的综合电子商务平台

1）网上下订单：客户在网上填写订单信息2）网上订单查询：通过订单号等查询订单信息3）网上支付：线上支付运费

* 1. 通过PERT技术估算的项目持续时间

PERT公式

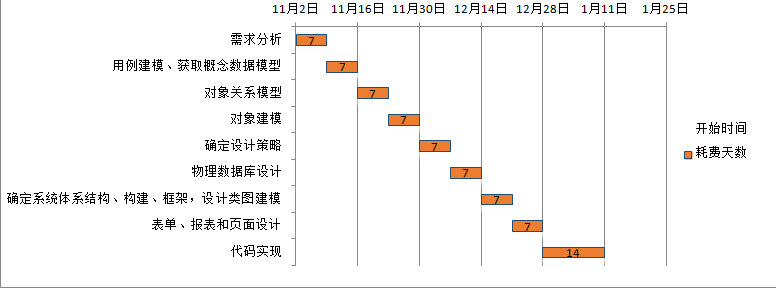
IMG_256

（其中a为乐观时间，m为最可能时间，b为悲观时间）

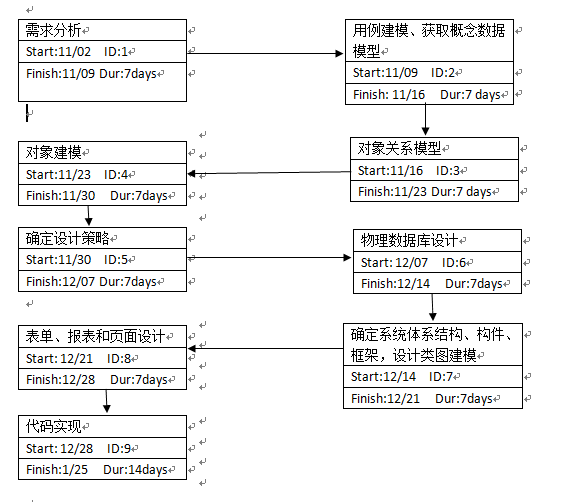
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块 | 乐观时间（day） | 最可能时间 | 悲观时间 | PERT估计期望时间 |
| 需求分析 | 7 | 11 | 15 | 11 |
| 用例建模、获取概念数据模型 | 5 | 7 | 11 | 7 |
| 对象关系模型 | 5 | 7 | 11 | 7 |
| 对象建模 | 6 | 7 | 11 | 8 |
| 确定设计策略 | 5 | 7 | 11 | 7 |
| 物理数据库设计 | 5 | 7 | 11 | 7 |
| 确定系统体系结构、构件、框架、设计类图建模 | 5 | 7 | 11 | 7 |
| 表单报表页面设计 | 6 | 7 | 11 | 8 |
| 代码实现 | 10 | 14 | 20 | 14 |

* 1. 甘特图和网络图

甘特图：



网络图：



## *实验2 项目可行性分析*

* 1. 可行性分析
     1. 经济可行性

1. 项目收益：

本系统可以为快递公司提供针对领导决策层，管理层，普通员工，用户的不同功能。（提供决策支持，减少错误，鼓舞士气，削减人工费用，简化繁琐的统计，提供便捷服务，提高公司市场竞争力）

1. 针对领导决策层：提供决策支持系统，根据系统收集的各种数据，提供领导查询（包括整个系统的各种功能和数据的查询）和员工业绩考核功能，省去人工统计的繁琐和避免差错，为领导层提供各种可供参考的数据，可根据这些数据制定未来发展战略，其中业绩考核功能有利于公司制定针对员工的激励或惩罚措施，提高整个公司的士气；
2. 针对管理层：提供运单监管功能，针对每一份订单都能及时追踪，有利于发现问题订单，减小损失发生的可能性；提供收件，中转，派件等的一体化查询管理，对公司整体资源实现统一调度，减小不必要的人员浪费；实现财务处理功能，对账单，发票等的核查变得更加方便快捷；
3. 针对普通员工：提供网上接单功能，减少员工的工作量，削减人工方面的开支，员工也可以很方便的查询自己的业绩等，有利于激励员工；
4. 针对用户：为用户提供准确的订单状态查询功能，网上下单和支付功能，给用户提供更加方便快捷的服务，增加公司的竞争力，在市场竞争中处于一个有利地位；
5. 项目成本：

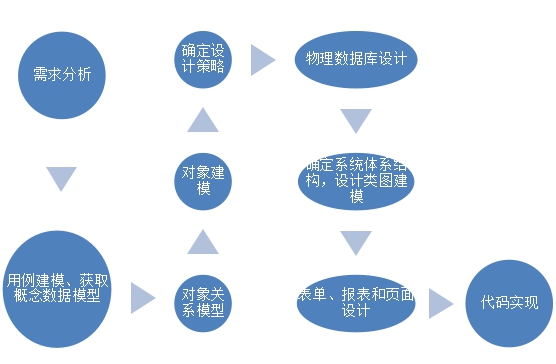
整个系统的开发费用（一次性成本）和软件维护费用（续生成本）是主要的开支，设备主要是PC机，处于现在的大数据时代，对于公司的领导层，管理层，普通员工已经用户来说，系统的使用并不是一件困难的事，所以在用户培训方面开支较小。随着数据逐渐增多，数据的存储和维护费用（续生成本）也会相应增加。

* + 1. 技术可行性

本系统将针对快递公司的业务过程展开设计，我们选择access来支持对数据库进行的操作,选择c#来支持系统的开发需求，开发环境为Windows7。本系统以订单的接收、处理、派送、查询为主要功能，辅助以领导决策支持、客户服务、网上平台等功能。在此之前，我们做过火车票订票系统、网上购物系统的实验，对其中的一些步骤有所了解。

* + 1. 进度可行性

进度图：



我们用两周时间进行代码实现部分，其余部分用一周时间实现，时间安排合理。

* + 1. 法律与契约可行性

关于合同责任，符合国家标准的合同，经双方签字后生效，所用软件都是正版，无侵犯版权专利权的情况，在快递管理信息系统中，需要客户填写他们的地址，联系方式，为了保护客户的隐私权，该系统限制了非管理员对客户信息的访问，防止客户隐私泄露。综上所述该信息系统在法律方面是可行的。

* + 1. 运营可行性

1. 客户除了可以通过电话预约还可以在网络自助下单通知快递公司，从而客户多了选择权利，也减少了因电话占线而损失客户的机会和电话沟通的时间，提高了效率，降低了成本。
2. 在结算环节上，该系统为客户提供了多种结算方式，如：寄方支付，到方支付，第三方支付等。
3. 该系统还提供了限时派送业务，在承诺的时间内把快件安全、准确的送达客户手中。
4. 为了完善服务，提高客户对公司的满意度，该系统还设计了业绩考核功能，可以记录客户对快递员的评价、快递员的派件速度、派件数量等，方便领导对员工的考核。
5. 为方便管理客户，系统设置客户管理功能，录入客户基本资料、历史交易情况、服务约束等。
6. 在快递业务中，经常会遇到问题件，这些快件的问题多种多样，为了方便统计，该系统提供了问题件处理功能，对问题件进行分类处理，提高公司监管效率。
7. 快递工作繁杂，业务员有时会忘记收件，运单录入，该系统设计了短信通知功能来通知业务员收件、运单录入。
8. 为方便监管核查，系统对每一单实时跟踪，及时发现问题，报警，对运单状态全程监控。

2.2． 进一步的细化的项目进度安排

|  |  |  |
| --- | --- | --- |
| 项目 | | 时间 |
| 需求分析 | 访谈笔录 | 11月2-3日 |
| 观察笔记 | 11月4-5日 |
| 会议记录 | 11月6日 |
| 企业业务表单、操作手册 | 11月7-8日 |
| 用例建模，建立概念数据模型 | 用例图 | 11月9-10日 |
| 用例描述 | 11月11-12日 |
| 建立概念数据模型 | 11月13-16日 |
| 建立关系对象模型 | 建立关系对象模型 | 11月17-23日 |
| 对象建模 | 实体类，边界类，控制类的划分 | 11月24日 |
| 画顺序图 | 11月25-26日 |
| 画分析类图 | 11月27-28日 |
| 画活动图和状态图 | 11月29-30日 |
| 确定设计策略 | 了解三种可选设计策略 | 12月1日 |
| 对三种可选设计策略定性定量分析 | 12月2-4日 |
| 更新后的BPP | 12月5-7日 |
| 物理数据库的设计 | 域的设计 | 12月8日 |
| 物理表的设计和反规范化 | 12月9-10日 |
| 文件组织 | 12月11-12日 |
| 对象关系特征的设计 | 12月13-14日 |
| 确定系统体系结构、构件、框架，设计类图建模 | 确定系统体系结构、构件、框架 | 12月15-17日 |
| 设计类图建模 | 12月18-21日 |
| 表单、报表和页面设计 | 表单 | 12月22-23日 |
| 报表 | 12月24日 |
| 页面设计 | 12月25-26日 |
| 代码实现 | 代码实现 | 12月27-1月3日 |

# 第二部分 需求分析

## *实验3 项目需求收集*

3.1. 访谈笔录、观察日记、会议记录等

3.1.1. 访谈笔录

1. 背景资料
2. 访谈目的

了解公司员工对快递管理信息系统的看法，以及对员工的工作的影响和变化。

1. 访谈对象

公司员工

描述：被访谈者为公司的管理层，现正在这家公司里参与快递管理信息系统这个项目的策划。

访谈思路：先了解被访者的工作态度，再询问公司对员工的要求，其次谈员工对快递管理信息系统的看法以及它对公司的影响，再次细问关于系统的问题。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **被访谈人** | | 黄先生（公司员工） | | | |
| **参 会 人** | | 公司所有人员 | | | |
| **访谈时间** | |  | | **联系方式** |  |
| **访 谈 人** | | 胡先生（公司员工） | | **记 录 人** | 李先生（公司员工） |
| 1 | 您是如何找到工作？ | | 是通过熟人介绍的，经过数月实习，通过自己的努力，从销售一步一步干到了主管。 | | |
| 2 | 刚参加工作时会有什么培训？ | | 刚进入工作，公司会对企业文化和公司产品经行培训。 | | |
| 3 | 对于快递信息管理系统这个新的项目，您的态度如何？ | | 很满意，觉得这是一个非常有发展前景的项目。 | | |
| 4 | 您觉得这个系统对公司有何作用？ | | 完善公司的服务机制，大大提高了公司的运行速度，减少了错误的出现，实现了客户公司一体化的现象，加强了公司与客户的联系。 | | |
| 5 | 这个系统是否会增加客户的利益？ | | 从系统定制开发，我们希望解决客户的个性化需求。我个人感觉这个系统实际上有倾向。 | | |
| 6 | 如果运行过程中，出现系统方面的问题怎么处理？ | | 公司会通知各个部门人员进行检测，务必争取系统的完善和系统的完美。 | | |
|  |  | |  | | |
| **访谈总结：**   1. 每个员工积极争取学习和进步的机会。 2. 公司对员工的素质和能力的要求严格。 3. 员工对快递信息管理系统这个新的项目十分看好。 4. 快递信息管理系统这个新的项目符合公司的要求，满足公司的业务需求 5. 快递信息管理系统这个新的项目适应于客户的需求和个性化要求。 6. 公司对快递信息管理系统这个新的项目精益求精。 | | | | | |

3.1.2. 观察日记

总体观察

访谈过程比较轻松，因为关系较好，且他情绪很好，主动陈述的地方比较多。希望能有更多的经历。

会议记录

会议时间：2015年12月28日星期一9：00

会议地点：快递公司的办公室

参会人员：快递公司的员工、老板

会议目的：收集员工的资料

达成完善系统的一致意见

探究员工对系统看法的领域

会议内容：

（1）公司成员相互介绍和认识

（2）讨论并进行项目名称拟定

（3）加强劳动纪律、整顿工作作风、提高服务质量

在会议中指出现各部门出现的种种问题，提出上班期间禁止扎堆聊天，各部门禁止上班期间擅自离岗以及夜班期间禁止到处串岗等要求，要求各部门应紧抓劳动纪律，整顿工作作风。上班期间禁止擅自离岗逛街（夜班雷同），一经发现扣除罚金50元。

（4）各部门负责人应带好头，加强责任心

员工的考核需通过部门负责人进行评比考核，部门工作人员需听取部门负责人的领导；部门出现的问题应及时向部门负责人上报，部门负责人解决不了方可上报公司；优化请假流程：员工请假必须有部门负责人签字同意后方可执行，禁止越级请假。同时各部门负责人需做好部门人员公休安排，无特殊情况，一个部门一天仅可一人休息。

（5）了解市场情况和竞争对手的情况

（6）讨论所需要的单据及合同以及如何拟定的问题

（7）讨论客户服务标准及如何接受客户咨询以及接收业务

（8）讨论分工问题

会议期间大家提出了各种建议并提出讨论，一致认为要加大各部门的联系，多想办法增加客户来源，使系统更强更好。各部门应遵守劳动纪律，整顿工作作风，提高服务质量。团结合作、协同作战，工作上互相帮助，生活中互相关心，将打造成出具有凝聚力的团队。

3.2. 企业业务表单、操作手册等现有的文档和文件



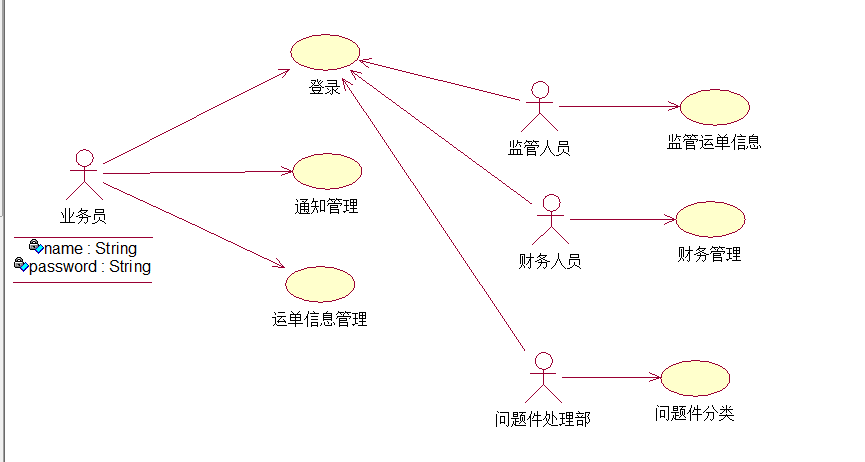




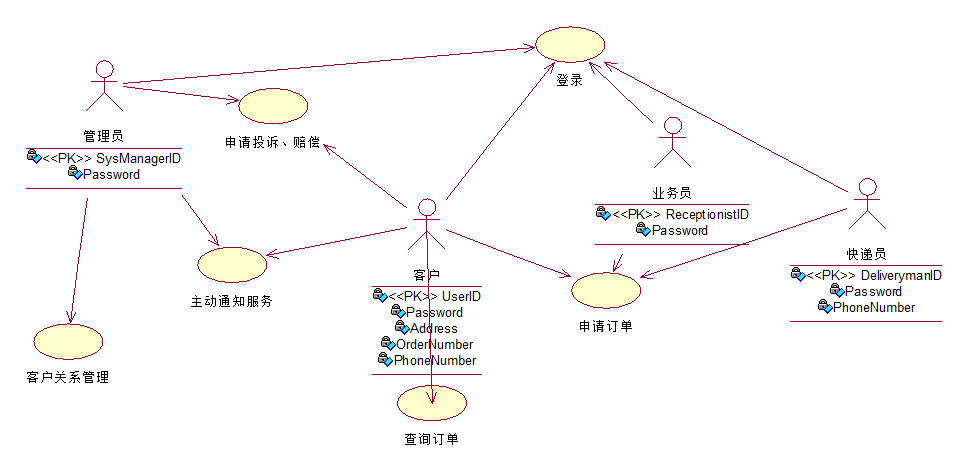
## *实验4 用例建模*

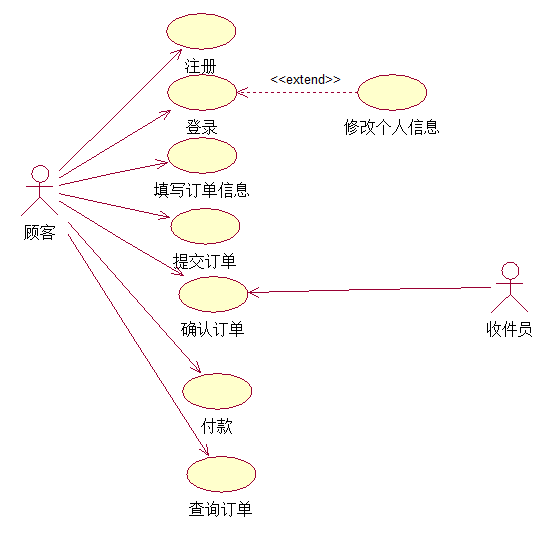
4.1. 用例图

4.1.1． 以运单为中心的核心业务系统

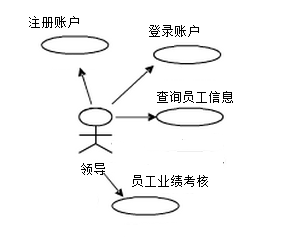


4.1.2． 以客户为中心的客户服务系统



4.1.3． 以互联网为手段的综合电子商务平台

4.1.4. 以决策为目的的领导决策支持系统



4.2. 用例描述

4.2.1. 以运单为中心的核心业务系统

|  |
| --- |
| 用例名称：通知管理 |
| 主参与者：业务员 |
| 层次：海平面（用户目标） |
| 利益相关者：业务员、快递员 |
| 前置条件：业务员必须在线可访问网络 |
| 最低保证：回滚任何未完成事物。系统记录进展日志直到失败 |
| 成功保证：业务员成功的接收到信息，并且信息完整的显示出来 |
| 触发器：业务员登录后点击发送通知 |
| 主要成功情节：   1. 业务员成功登录。 2. 在业务员的硬盘上创建Cookie。 3. 业务员点击查看信息，获得收件或者派件信息。   4 业务员选择要转发的信息。  5 业务员选择要转发的快递员的账号。  6. 业务员可以成功的转发收件或派件消息 |
| 扩展：  1.a 不能成功登录。   |  | | --- | | 用例名称：登录 | | 主参与者：业务员，监管人员，财务人员, 问题件处理部, 快递员 | | 层次：海平面（用户目标） | | 利益相关者：业务员，监管人员，财务人员, 问题件处理部, 快递员 | | 前置条件：图书馆的用户注册页面正常运行 | | 最低保证：回滚任何未完成事物。系统记录进展日志直到失败 | | 成功保证：用户可以成功的登录到主页 | | 触发器：用户点击页面上的登录按钮 | | 主要成功情节：  1.用户进入快递公司的用户注册页面 ；  2.用户输入需要注册的用户名和密码 ；  3.用户将注册信息进行提交；  4.系统对用户提交的用户名和密码进行有效性检查；  5.系统显示用户是否注册成功；  6.用户输入用户名和密码成功登录到相应的页面； | | 扩展：  1-5.a 不能成功注册  1-5.a.1 输入的密码不匹配  1-5.a.1.1 提示用户密码输入不一致  1-5.a.1.2用户离开或者重新输入密码  1-5.a.2 用户名已存在  1-5.a.2.1 提示用户输入的用户名已存在  1-5.a.2.2 用户离开或者重新输入用户名  6.a 不能成功登录。  6.a.1 输入的用户名错误。  6.a.1.1提示用户账号不存在或者错误。  6.a.1.2用户离开或重新输入账号。  6.a.2 输入密码错误。  6.a.2.1提示用户密码输入错误或者是否找回密码。  6.a.2.2用户离开或者重新登录、找回密码。 |   1.a.1业务员被告知账户信息无效。业务员提供有效的信息并成功登录。  1.a.2 业务员被告知账户信息无效。业务员转到服务网站。  2.a 不能创建Cookie。  2.a.1 创建消息向业务员说明因为他或她的Web浏览器不允许创建cookie，所以不可能查看消息。  2.a.2 业务员调整浏览器的cookie设置并重试，或者离开网站。  3.a. 请求的Web页面不能装入或找不到。  3.a.1 业务员在浏览器中得到一个“找不到网页”的错误页面。  3.a.2 业务员点击刷新按钮，请求页面成功加载。  3.a.3 业务员单击刷新按钮，请求页面不能成功加载，业务员离开页面。 |

|  |
| --- |
| 用例名称：运单信息管理 |
| 主参与者：业务员 |
| 层次：海平面（用户目标） |
| 利益相关者：业务员、快递员 |
| 前置条件：业务员必须在线可访问网络 |
| 最低保证：回滚任何未完成事物，系统记录进展日志直到失败 |
| 成功保证：业务员成功将运单信息录入 |
| 触发器：业务员访问运单管理网页 |
| 主要成功情节：   1. 业务员成功填写运单的各项信息。 2. 业务员成功的更新物流信息。 3. 运单信息被成功的保存。 |
| 扩展：  1.a 录入事务被中断。  1.a.1 事务回滚，业务员再次开始。  1.a.2 事务回滚，业务退出网站。  2.a. 请求的Web页面不能装入或找不到。  3.a.1 业务员在浏览器中得到一个“找不到网页”的错误页面。  3.a.2 业务员点击刷新按钮，请求页面成功加载。  3.a.3 业务员单击刷新按钮，请求页面不能成功加载，业务员离开页面。 |

|  |
| --- |
| 用例名称：监管运单信息 |
| 主要参与者：管理人员 |
| 层次：海平面 |
| 利益相关者：管理人员，快递员 |
| 前置条件：管理人员已经登录快递管理系统 |
| 最低保证：回滚任何未完成的事务，系统记录进展日志直到失败 |
| 成功保证：管理人员能够查看运单状态 |
| 触发器：管理人员点击“查看运单状态” |
| 主要的成功情节：   1. 管理人员进行运单查看 2. 若发现问题件，管理人员进入提交问题件页面 3. 管理人员填写问题件信息 4. 管理人员提交问题件信息 |
| 扩展：   1. a查询所有运单信息不成功   1.a.1管理人员得到一个“查询错误”信息  1.a.1重新查询运单信息，成功查询  1.a.2重新查询运单信息，查询失败，退出Web网页  2.a进入问题件提交页面不成功  2.a.1管理人员在进入浏览器中得到一个“找不到网页”的错误页面  2.a.2管理人员点击刷新按钮，请求页面成功装载  2.a.3管理人员点击刷新按钮，请求页面不成功，退出Web网页  4.a提交不成功  4.a.1管理人员得到一个提交不成功提示  4.a.2管理人员重新提交，提交成功  4.a.3管理人员重新提交，提交不成功，退出Web网页 |

|  |
| --- |
| 用例名称：财务管理 |
| 主要参与人员：财务人员 |
| 层次：海平面 |
| 利益相关者：财务人员，客户 |
| 前置条件： |
| 最低保证：回滚任何未完成的事务，系统记录进展日志直到失败 |
| 成功保证：财务人员完成财务报表 |
| 触发器：财务人员点击“绘制财务报表” |
| 主要的成功情节：   1. 为当天创建一张财务报表 2. 财务人员填写编号，日期，借方，贷方，金额等相关信息 |
| 扩展：   1. a创建财务报表不成功   1.a.1点击创建财务报表，得到一个创建失败信息  1.a.2重新创建财务报表，创建成功  1.a.3重新创建财务报表，创建失败，退出Web网页  2.a保存不成功  2.a.1点击保存，得到保存失败信息  2.a.2重新保存，保存成功  2.a.3重新保存，保存失败，回滚到保存前的状态，选择不保存退出 |

|  |
| --- |
| 用例名称：问题件处理 |
| 主要参与人员：问题件处理部门 |
| 层次：海平面 |
| 利益相关者：问题件处理部门，管理人员 |
| 前置条件：收到管理人员发来的问题件处理信息 |
| 最低保证：回滚任何未完成的事务，系统记录进展日志直到失败 |
| 成功保证：完成问题件处理 |
| 触发器：问题件处理部门人员点击“问题件录入” |
| 主要成功情节：   1. 将收到的问题件信息录入到问题件信息处理中 2. 将问题件信息进行分类 3. 处理问题件信息 4. 标注问题件状态 |
| 扩展：  1.a无法进入问题件录入界面  1.a.1点击问题件录入，加载页面失败，收到错误信息  1.a.2重新进入，进入成功  1.a.3重新进入，加载失败，退出Web网页  2.a系统显示填写分类信息错误  2.a.1重新填写分类信息，并保存  2.a.2没有该分类，选择其他分类  2.a.3没有该分类，退出Web网页 |

4.2.2． 以客户为中心的客户服务系统

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 查询订单（分为互联网方式和电话方式） |
| 主要参与者 | | 客户 |
| 层次 | | 海平面 |
| 利益相关者 | | 客户 |
| 前置条件 | | 客户成功登录快递管理系统（或客户成功拨通客服电话） |
| 最低保证 | | 回滚任何未完成的事务，系统记录进展日志直到失败（或电话得到应答） |
| 成功保证 | | 客户成功查看订单信息 |
| 触发器： | 客户选择“查询订单” | |
| 主要成功情节：   1. 选择服务方式。   若为互联网方式：  2.1客户登录。  2.2客户查看订单列表。  2.3客户通过查看快递单得到快递单号。  2.5客户在订单列表上选择要查看的订单。  2.6 客户选择“查看物流信息”。  2.6客户获得订单物流信息。  若为电话方式：  3.1客户通过查看快递单得到快递单号、客服电话。  3.2客户将单号提供给客服。  3.3客户获得订单物流信息。 | | |
| 扩展：  1.a请求的web页面不能载入或不能找到。（互联网方式）  1.a.1 客户在浏览器中得到一个“找不到页面”的错误页面。  1.a.2 客户单击刷新按钮，请求页面成功加载。  1.a.3 客户单击刷新按钮，请求页面成功加载；客户离开网站。  2.a 订单状况不可获得。（互联网方式）  2.a.1 客户退出网站；  2.a.2 客户采取行动获得对订单状况的访问。  3.a 电话中途挂断。（电话方式）  3.a.1 客户重新打电话给客服。 | | |

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 下订单（分为互联网方式和电话方式） |
| 主要参与者 | | 客户 |
| 层次 | | 海平面 |
| 利益相关者 | | 客户、业务员、快递员 |
| 前置条件 | | 客户成功登录快递管理系统（或客户成功拨通客服电话） |
| 最低保证 | | 回滚任何未完成的事务，系统记录进展日志直到失败（或电话得到应答） |
| 成功保证 | | 客户成功下订单 |
| 触发器： | 客户选择“在线下单”（或客户选择下单） | |
| 主要成功情节：   1. 选择服务方式。   若为互联网方式：  2.1客户调用“登录”用例。  2.2为这个事务创建一个订单。  2.3填写寄件信息、收件信息、物品信息、取货时间。  2.5客户收到包含费用的订单。  2.6客户选择“提交订单”。  2.7收费。  2.8业务员受理订单。  2.9把订单任务分配给快递员。  2.10创建客户收据并作为电子邮件发送给用户。  2.11 快递员取件。  若为电话方式：   * 1. 登录。   2. 客户填写寄件信息、收件信息、物品信息、取货时间。   3. 业务员受理订单。   4. 把订单任务分配给快递员。   5. 快递员取件并收费。 | | |
| 扩展：  1.a “登录”用例失败。  1.a．1 客户提供有效的信息并成功登录。  1.a．2 客户转到客户服务网站。  2.a 客户信用卡账户不能成功收费。  2.a．1 客户被要求用一个新的信用卡账户。  3.a Internet连接中断。  3.a．1 系统回滚到顾客交易前的位置。 | | |

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 申请投诉、赔偿 |
| 主要参与者 | | 客户 |
| 层次 | | 海平面 |
| 利益相关者 | | 客户、管理员 |
| 前置条件 | | 客户成功登录快递管理系统 |
| 最低保证 | | 回滚任何未完成的事务，系统记录进展日志直到失败 |
| 成功保证 | | 客户成功提交申请投诉、赔偿信息 |
| 触发器： | 客户选择“申请投诉”或“赔偿” | |
| 主要成功情节：   1. 用户登录。 2. 客户选择“申请投诉”或“赔偿”。 3. 客户提交申请投诉或赔偿信息。 4. 管理员受理投诉或赔偿。 5. 通过邮件反馈结果。 | | |
| 扩展：   1. a客户提交申请投诉、赔偿信息失败。   1.a.1 客户重新提交申请投诉、赔偿信息。  2.a “登录”用例失败。  2.a．1 客户提供有效的信息并成功登录。  2.a．2 客户转到客户服务网站。 | | |

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 主动通知服务 |
| 主要参与者 | | 管理员 |
| 层次 | | 海平面 |
| 利益相关者 | | 客户、管理员 |
| 前置条件 | | 管理员成功登录快递管理系统 |
| 最低保证 | | 回滚任何未完成的事务，系统记录进展日志直到失败 |
| 成功保证 | | 管理员成功通知客户 |
| 触发器： | 管理员选择“主动通知服务” | |
| 主要成功情节：   1. 管理员登录。 2. 管理员选择“主动通知服务”。   3. 管理员获得订单物流信息。  4. 针对可以领取的快递发送收货信息给客户。  5. 通知快递员送货上门。  6. 客户收到快递。 | | |
| 扩展：  1.a管理员获得订单物流信息失败。  1.a.1客户采取行动获得对订单状况的访问。  2.a “登录”用例失败。  2.a．1 管理员提供有效的信息并成功登录。 | | |

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 客户关系管理 |
| 主要参与者 | | 管理员 |
| 层次 | | 海平面 |
| 利益相关者 | | 客户、管理员 |
| 前置条件 | | 管理员成功登录快递管理系统 |
| 最低保证 | | 回滚任何未完成的事务，系统记录进展日志直到失败 |
| 成功保证 | | 管理员成功管理客户资料 |
| 触发器： | 管理员选择“客户关系管理” | |
| 主要成功情节：   1. 管理员登录。 2. 管理员选择“客户关系管理”。 3. 管理员针对用户资料进行增删改查操作。 | | |
| 扩展：  1.a管理员获得用户资料失败。  1.a.1客户采取行动获得对用户资料的访问。  2.a “登录”用例失败。  2.a．1 管理员提供有效的信息并成功登录。 | | |

4.2.3． 以互联网为手段的综合电子商务平台

|  |
| --- |
| 用例名称：在线下快递订单 |
| 主参与者：顾客 |
| 层次：风筝 |
| 利益相关者：顾客、收件员 |
| 前置条件：顾客访问该快递网站 |
| 最低保证：回滚任何未完成的事务 |
| 成功保证：付款成功 |
| 触发器：顾客访问网站主页 |
| 主要的成功情节：   1. 顾客登录 2. 顾客填写订单信息 3. 顾客提交订单 4. 顾客确认订单 5. 顾客付款 |
| 扩展：  1.a 登录失败  1.a.1顾客退出网站  1.a.2顾客采取行动成功登录  2.a 订单信息无法录入  2.a.1顾客采取行动录入正确信息  2.a.2顾客退出网站  3.a 无法提交订单  3.a.1等待后顾客成功提交  3.a.2顾客退出网站  4.a 确认失败  4.a.1顾客取消订单  4.a.2顾客返回修改订单  5.a付款失败  5.a.1顾客采取行动成功付款  5.a.2顾客取消订单退出网站 |

4.2.4. 以决策为目的的领导决策支持系统

|  |  |
| --- | --- |
| 用例名称： | 业绩考核 |
| 执行者： | 领导 |
| 用例说明： | 领导业绩考核员工 |
| 前置条件： | 查询员工信息 |
| 基本事件流： | 1. 查询员工信息 2. 查看员工业绩情况 |
| 其它事件流： | 2.a员工的业绩情况反映公司的业绩 |
| 异常事件流： | 3c.查找时出现系统故障，例如网络故障，数据库服务器故障，系统弹出系统异常页面，则查找不成功。 |
| 后置条件： | 了解员工的业绩情况 |
| 备注说明： | 无 |

|  |  |
| --- | --- |
| 用例名称： | 业绩考核 |
| 执行者： | 领导 |
| 用例说明： | 领导业绩考核员工 |
| 前置条件： | 查询员工信息 |
| 基本事件流： | 1. 查询员工信息 2. 查看员工业绩情况 |
| 其它事件流： | 2.a员工的业绩情况反映公司的业绩 |
| 异常事件流： | 3c.查找时出现系统故障，例如网络故障，数据库服务器故障，系统弹出系统异常页面，则查找不成功。 |
| 后置条件： | 了解员工的业绩情况 |
| 备注说明： | 无 |

## *实验5 通过用例获取概念数据模型*



## *实验6 将概念数据模型转化为对象关系模型*

Logistics（ID,OrderID,Location,first,before1,next,date,time,received)外键为：OrderID

Order(OrderID,Sender,Recipients,STelephone,RTelephone,Saddress,Raddress,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,Charge)

ProblemOrder(ProID,Type,OrderID,State)外键为：OrderID

Account(ID,Date)

SysManager(ID,Name,Password)

Customer(ID,Name,Password,Address,TelePhone,OrderID) 外键为：OrderID

Deman(ID,Name,Password,TelePhone)

Receptionist(ID,Name,Password)

Supervisor(ID,Name,Password)

FinancialStaff(ID,Name,Password)

ProblemDepartment(ID,Name,Password)

Employee(ID,Name,Password,Position,TelePhone,Performance)

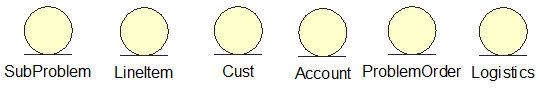
## *实验7 分析类图建模*

7.1. 以运单为中心的核心业务系统

7.1.1. 实体类、边界类、控制类的划分

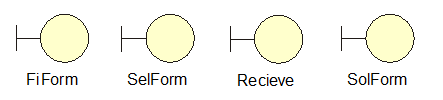
7.1.1.1. 实体类：





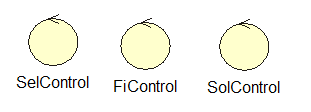
7.1.1.2. 边界类：





7.1.1.3. 控制类：





7.1.2.顺序图

7.1.2.1. 通知管理：

Receptionist

: InformForm

: informControl

: Receptionist

: Deman

//provide login info()

//request login()

//verify login()

//review inform()

//review inform()

get salesman info()

//display inform details()

//email Deman()

//get Deman info()

7.1.2.2. 运单信息管理



7.1.2.3.财务处理：



7.1.2.4. 监管：



7.1.2.5.问题件处理：



7.1.3. 分析类图

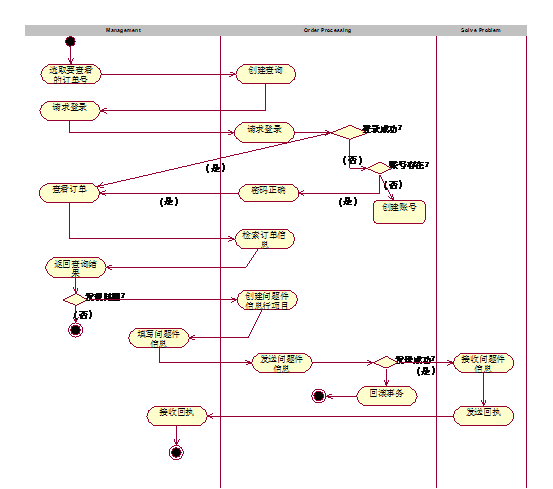
7.1.3.1． 通知管理



7.1.3.2．运单信息管理



7.1.4. 活动图



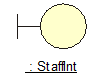
7.2. 以客户为中心的客户服务系统

7.2.1. 实体类、边界类、控制类的划分

7.2.1.1. 实体类：



7.2.1.2. 边界类：



7.2.1.3. 控制类：



7.2.2. 顺序图

* + - 1. 下订单（电话方式）顺序图：

7.2.2.2. 下订单（网络方式）顺序图：



7.2.2.3.查询咨询服务（电话方式）顺序图：

7.2.2.4.查询咨询服务（网络方式）顺序图：



7.2.2.5.主动通知服务（仅电话方式）顺序图：



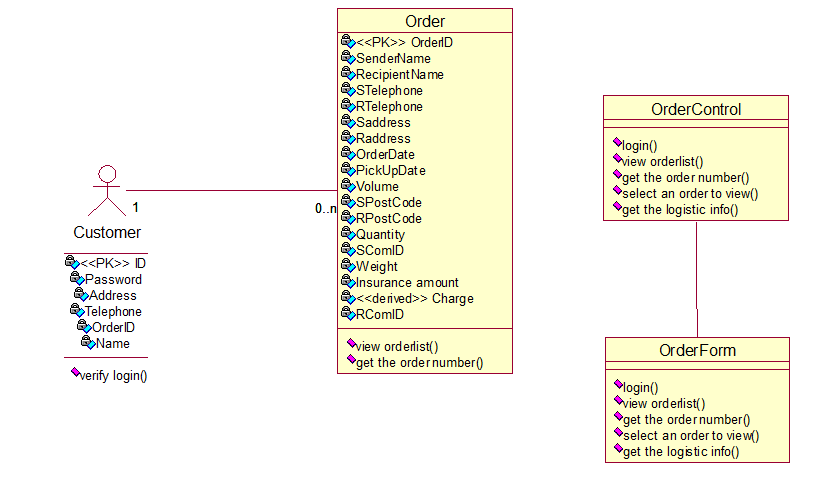
7.2.2.6.客户关系管理顺序图：



7.2.2.7.投诉处理顺序图：



* + 1. 分析类图
       1. 查询订单（网络方式）



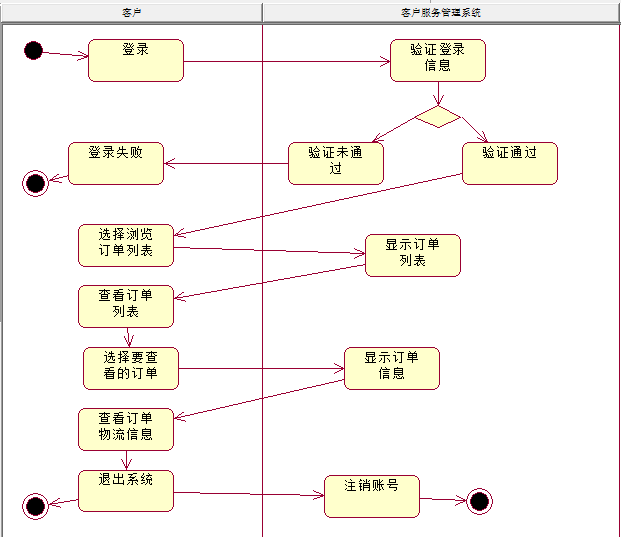
* + - 1. 下订单（网络方式）



* + 1. 活动图
       1. 下订单活动图：

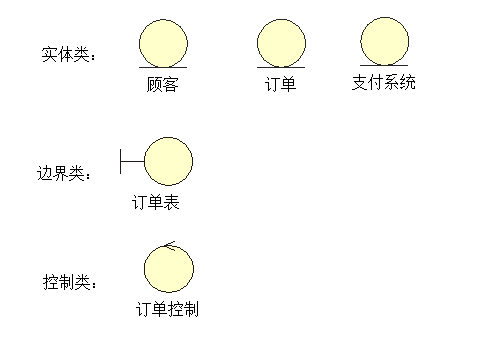


* + - 1. 查询订单活动图：

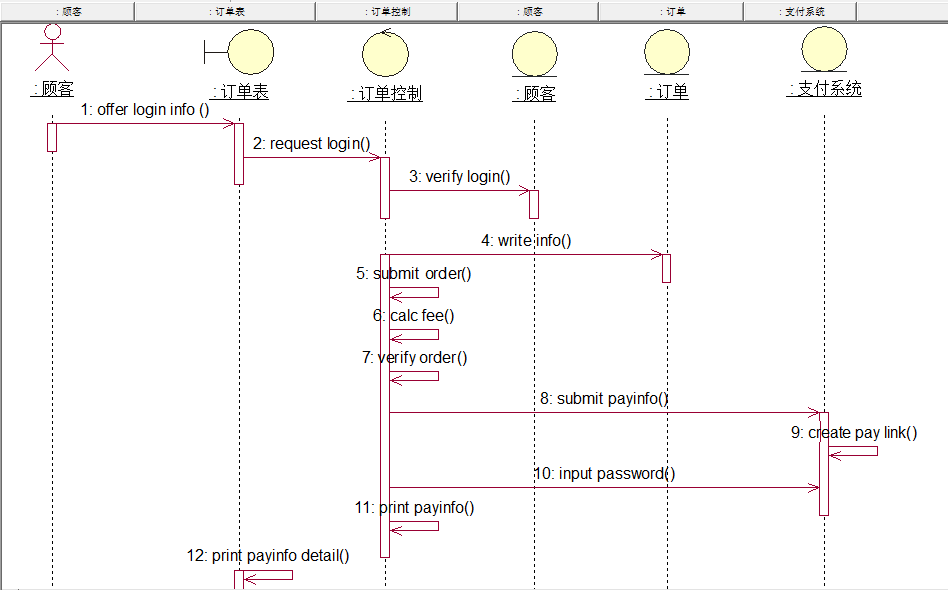


* 1. 以互联网为手段的综合电子商务平台

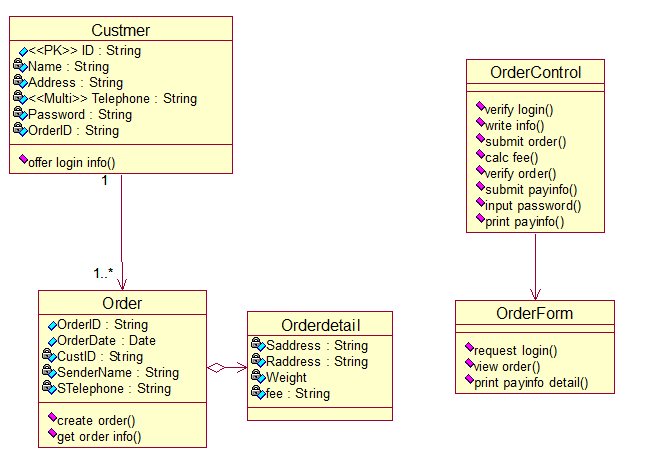
7.3.1. 实体类、边界类、控制类的划分：



* + 1. 顺序图：



* + 1. 分析类图：



* 1. 以决策为目标的领导决策支持系统

7.4.1. 实体类、边界类、控制类的划分

* + - 1. 实体类：

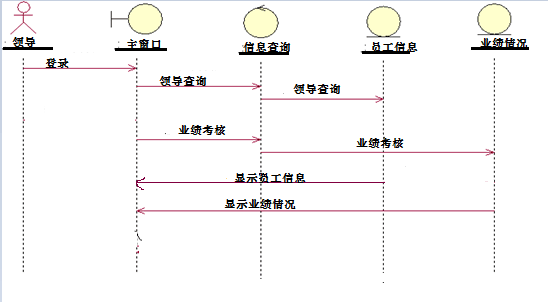
* + - 1. 边界类



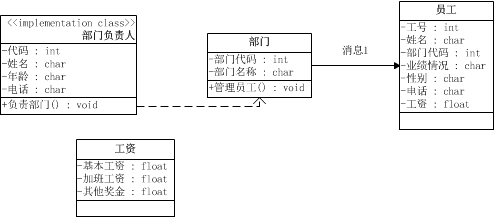
* + - 1. 控制类



7.4.2. 顺序图



7.4.3.分析类图



## *实验8三种可选方案选择分析*

8.1. 三种可选设计策略

**快递管理信息系统的需求和约束**

|  |  |
| --- | --- |
| **需求** | **约束** |
| **强制系统需求：**  四大应用系统能够完全集成  99.9%的正常运行时间和可用性  **基本系统需求：**  未来系统集成的灵活性和伸缩性  有效的和有成本效益的系统管理  **期望的系统需求：**  应急响应  文档资料 | 一月中旬前完成产品交付  少量的开发人员  有限的外部咨询预算  有限的外部支持 |

8.2. 三种可选设计策略定性与定量分析

**三种可选系统优缺点定性分析**

|  |  |  |
| --- | --- | --- |
| 方案 | 优点 | 缺点 |
| 外包给应用服务供应商 | 所有硬件都不设在现场  应用程序的开发不在现场  优秀的应急响应 | 不灵活 |
| 企业资源计划系统 | 稳定性强  文档资料可用性 | 需要有技能的内部员工  昂贵的硬件和软件  较大的学习曲线 |
| 应用服务器/对象框架 | 四大应用系统能较好集成  可伸缩性  灵活性 | 需要内部开发  计划和开发期间文档至关重要 |

**三种可选设计策略加权方法定量分析**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 方案A：外包 | | 方案B：ERP | | 方案C：服务器/对象 | |
|  | 权值 | 等级 | 分数 | 等级 | 分数 | 等级 | 分数 |
| 需求 |  |  |  |  |  |  |  |
| 四大部分集成 | 10 | 4 | 40 | 3 | 30 | 4 | 40 |
| 正常运行时间 | 10 | 4 | 40 | 4 | 40 | 4 | 40 |
| 灵活性和可伸缩性 | 10 | 3 | 30 | 2 | 20 | 4 | 40 |
| 应急响应 | 10 | 1 | 10 | 1 | 10 | 4 | 40 |
| 文档资料 | 10 | 2 | 20 | 2 | 20 | 4 | 40 |
| 有效的系统管理 | 10 | 3 | 30 | 3 | 30 | 4 | 40 |
|  | 60 |  | 140 |  | 150 |  | 240 |
| 约束 |  |  |  |  |  |  |  |
| 一月中旬交付 | 10 | 3 | 30 | 3 | 30 | 4 | 40 |
| 少量开发人员 | 10 | 5 | 50 | 2 | 20 | 3 | 30 |
| 有限外部咨询预算 | 10 | 2 | 20 | 2 | 20 | 2 | 20 |
| 有限外部支持 | 10 | 2 | 20 | 2 | 20 | 2 | 20 |
|  | 40 |  | 120 |  | 90 |  | 110 |
| 合计 |  |  | 280 |  | 240 |  | 350 |

**说明：等级级别1表示还方案不能很好地满足需求或改方案违反了约束，等级5表示该方案满足或超过了需求或者明确地遵守约束。**

分析：如果只考虑需求，方案C是最佳选择；如果只考虑约束，方案A是最佳方案。结合需求与约束来看，方案A是最佳方案。

8.3. 更新后的BPP

基线项目计划（BPP）报告

**1.0 引言**

**A. 项目综述**——90年代以来，快递公司由一种新兴产业逐渐走入千家万户，在加快社会发展脚步的同时，为人们的生活提供了很大便利，也为自己创造了极大的利益和社会价值。在日常生活中，我们和快递公司的接触也很多。基于以上原因，我们选择快递公司作为研究对象，为其分析设计一个适合的管理信息系统。

**A1 项目名称**：快递管理信息系统

**A2 问题说明**：

Internet已经成为一个各方人员沟通的主要桥梁。快递管理信息系统的创建允许快递公司工作人员通过Internet管理大量的快递订单并进行实时监控，同时，它也允许顾客通过Internet在线填写快递订单并实时查询订单的状态与物流信息。为了达到公司目标，创建这个系统是必须的。

**A3 项目目标**：

使顾客通过Internet在线填写快递订单并实时查询订单的状态与物流信息，使快递公司工作人员通过Internet管理大量的快递订单并进行实时监控，由此提高快递公司的工作效率与业务成绩。

**A4 项目描述**：

整个快递管理信息系统分为四大部分：以决策为目的的领导决策支持系统、以运单为中心的核心业务系统、以客户为中心的客户服务系统和以互联网为手段的综合电子商务平台，其中，核心业务系统是重点。

1. 以决策为目的的领导决策支持系统

1)领导查询：原则上，普通员工能看到的信息领导都可以看到，领导桌面的信息可定制。2)业绩考核：根据设定的管理指标，实时反映员工业绩情况。

1. 以运单为中心的核心业务系统

1)收件：短信通知业务员收件、运单录入。2)中转：中转到件、中转发件、中转报关。3)到件：派件清单、派件回单（标志快件周期 结束）。4)监管核查：对每一单实时跟踪，及时发现问题，报警，对运单状态全程监控。5)财务处理：发票、对账单、应收／应付、核查、收／付款、财务核销等。6)问题件处理：对各种问题件进行分类处理。

1. 以客户为中心的客户服务系统

1)查询咨询服务：可通过电话、Internet等方式查询运单状态，咨询业务。2)投诉处理：受理对服务的投诉、赔偿要求等。3)接受定单：通过电话、传真等方式接受定单委托、预约等。4)主动通知服务：定制服务要求。如：货到后通过短信通知发货方。5)客户关系管理：客户的基本资料、历史交易情

1. 以互联网为手段的综合电子商务平台

1）网上下订单：客户在网上填写订单信息2）网上订单查询：通过订单号等查询订单信息3）网上支付：线上支付运费

**A5 业务收益**：增加快递订单

改进顾客对快递订单模式的理解

改进对快递员的利用

**A6 项目交付产品**：快递管理信息系统分析与设计

系统核心程序

系统文档

**A7 估计项目持续时间**：两个月

**B. 建议**——本系统的使用人群设计到快递公司工作人员、快递公司领导层以及顾客，要充分做好权限设计，充分理解不同使用者的不同需求。

2.0 系统描述

**A. 可选方案**——本系统是基于web的在线系统，开发平台是JAVA ，运用mvc框架，所需要的操作系统是XP以上，使用的数据库类型是mysql。

**B. 系统描述**——整个快递管理信息系统分为四大部分：以决策为目的的领导决策支持系统、以运单为中心的核心业务系统、以客户为中心的客户服务系统和以互联网为手段的综合电子商务平台，其中，核心业务系统是重点。

3.0可行性评估

**A. 经济分析**

1. 项目收益：

本系统可以为快递公司提供针对领导决策层，管理层，普通员工，用户的不同功能。（提供决策支持，减少错误，鼓舞士气，削减人工费用，简化繁琐的统计，提供便捷服务，提高公司市场竞争力）

1. 针对领导决策层：提供决策支持系统，根据系统收集的各种数据，提供领导查询（包括整个系统的各种功能和数据的查询）和员工业绩考核功能，省去人工统计的繁琐和避免差错，为领导层提供各种可供参考的数据，可根据这些数据制定未来发展战略，其中业绩考核功能有利于公司制定针对员工的激励或惩罚措施，提高整个公司的士气；
2. 针对管理层：提供运单监管功能，针对每一份订单都能及时追踪，有利于发现问题订单，减小损失发生的可能性；提供收件，中转，派件等的一体化查询管理，对公司整体资源实现统一调度，减小不必要的人员浪费；实现财务处理功能，对账单，发票等的核查变得更加方便快捷；
3. 针对普通员工：提供网上接单功能，减少员工的工作量，削减人工方面的开支，员工也可以很方便的查询自己的业绩等，有利于激励员工；
4. 针对用户：为用户提供准确的订单状态查询功能，网上下单和支付功能，给用户提供更加方便快捷的服务，增加公司的竞争力，在市场竞争中处于一个有利地位；
5. 项目成本：

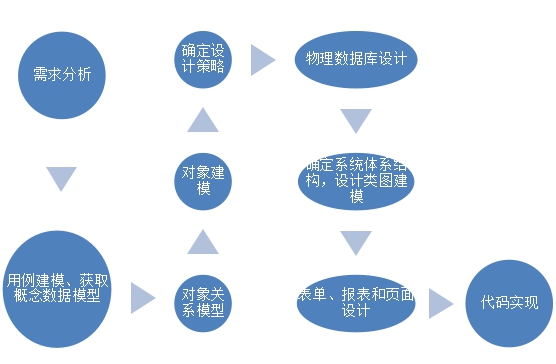
整个系统的开发费用（一次性成本）和软件维护费用（续生成本）是主要的开支，设备主要是PC机，处于现在的大数据时代，对于公司的领导层，管理层，普通员工已经用户来说，系统的使用并不是一件困难的事，所以在用户培训方面开支较小。随着数据逐渐增多，数据的存储和维护费用（续生成本）也会相应增加。

**B. 技术分析**

本系统将针对快递公司的业务过程展开设计，我们选择mysql来支持对数据库进行的操作,选择JAVA来支持系统的开发需求，运用MVC框架，开发环境为Windows7。本系统以订单的接收、处理、派送、查询为主要功能，辅助以领导决策支持、客户服务、网上平台等功能。在此之前，我们做过火车票订票系统、网上购物系统的实验，对其中的一些步骤有所了解。

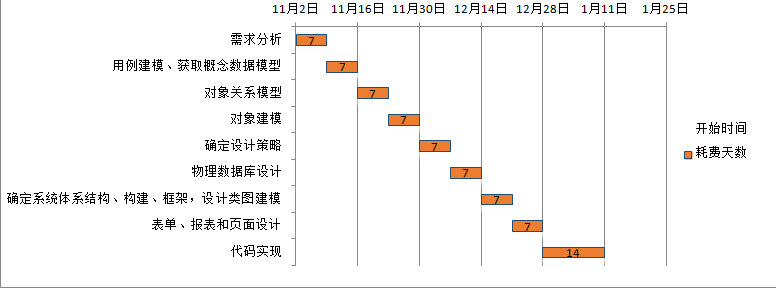
**C.进度分析**

进度图：



我们用两周时间进行代码实现部分，其余部分用一周时间实现，时间安排合理。

甘特图：



**D. 法律与契约分析**

关于合同责任，符合国家标准的合同，经双方签字后生效，所用软件都是正版，无侵犯版权专利权的情况，在快递管理信息系统中，需要客户填写他们的地址，联系方式，为了保护客户的隐私权，该系统限制了非管理员对客户信息的访问，防止客户隐私泄露。综上所述该信息系统在法律方面是可行的。

**E. 运营分析**

1. 客户除了可以通过电话预约还可以在网络自助下单通知快递公司，从而客户多了选择权利，也减少了因电话占线而损失客户的机会和电话沟通的时间，提高了效率，降低了成本。
2. 在结算环节上，该系统为客户提供了多种结算方式，如：寄方支付，到方支付，第三方支付等。
3. 该系统还提供了限时派送业务，在承诺的时间内把快件安全、准确的送达客户手中。
4. 为了完善服务，提高客户对公司的满意度，该系统还设计了业绩考核功能，可以记录客户对快递员的评价、快递员的派件速度、派件数量等，方便领导对员工的考核。
5. 为方便管理客户，系统设置客户管理功能，录入客户基本资料、历史交易情况、服务约束等。
6. 在快递业务中，经常会遇到问题件，这些快件的问题多种多样，为了方便统计，该系统提供了问题件处理功能，对问题件进行分类处理，提高公司监管效率。
7. 快递工作繁杂，业务员有时会忘记收件，运单录入，该系统设计了短信通知功能来通知业务员收件、运单录入。
8. 为方便监管核查，系统对每一单实时跟踪，及时发现问题，报警，对运单状态全程监控。

**4.0 管理问题**

**A. 团队结构和管理**

组长：项英 负责电子商务平台的总体分析设计、三种可选设计策略、对三种可选设计策略的定性与定量分析、更新后的BPP

组员：郭雪 负责网络图、客户服务系统的总体分析设计、物理表的设计和反规范化、对象关系特征的设计、系统体系结构、构件、框架

陈雅楠 负责通过PERT技术估算的项目持续时间、核心业务系统的总体分析与设计、域的设计、文件组织、设计类图

曾子轩 负责进一步细化的项目进度安排、核心业务系统的总体分析与设计、代码实现

许子吟 负责WBS和所需资源、客户服务系统的总体分析设计、表单和报表设计、对话设计

李和嫒 负责领导决策支持系统的总体分析设计、系统需求汇总

**B. 沟通计划**

小组成员尤其是组长要及时的与管理方进行信息沟通，并将所获取的信息及时的通知到小组成员。每周进行一次小组会议，分配任务，并将成员一周来做的系统成果进行讨论，总结和改进，并由组长负责系统性的修改，打印好电子档。

在用户方面，及时的收集用户反馈信息，对系统进行技术上的改进，满足用户的更高需求。

**C. 项目标准和规程**

交付产品，客户在使用之后，他们的满意度将是我们调查产品是否合格的一个最重要依据。首先，我们将在客户之间进行满意度调查，如果客户客户满意度达到95%以上，那么我们的产品即被列为合格。其次，我们要在客户之间建立一个客户权益管理，跟踪追查对客户进行价值分析、忠诚度分析。如若客户的对我们产品的忠诚度达到85%以上，则表明客户已经接受了我们的产品。

# 第三部分 系统设计

## *实验9物理数据库设计*

9.1.域的设计：

域是由系统软件比如编程语言或数据库管理系统识别的应用数据的最小单位。

9.1.1.选择数据类型：

选择一个数据类型要均衡以下四个目标：

最小化存储空间；

表示域的所有可能值；

提高域的数据完整性；

支持域上想要的所有数据操作；

综合上述目标，各个表中选择的数据类型如下所示：

Logistics(物流信息表)：

|  |  |  |
| --- | --- | --- |
| 名称 | 名 | 类型 |
| 物流信息编号 | ID | int |
| 运单编号 | OrderID | varchar |
| 快递目前所在的地点 | Location | varchar |
| 始发地 | first | varchar |
| 快递所在地点的上一站 | before1 | varchar |
| 快递所在地点的下一站 | next | varchar |
| 当前日期 | date | date |
| 当前时间 | time | time |
| 状态 | received | varchar |

Order（运单）

|  |  |  |
| --- | --- | --- |
| 名称 | 名 | 类型 |
| 运单编号 | OrderID | varchar |
| 发件人 | Sender | varchar |
| 发件人电话 | STelephone | varchar |
| 发件人地址 | Saddress | varchar |
| 收件人 | Recipients | varchar |
| 收件人电话 | RTelephone | varchar |
| 收件人地址 | Raddress | varchar |
| 货物名称 | Cargo | varchar |
| 货物的数量 | Quantity | int |
| 货物的重量 | Weight | double |
| 货物的体积 | Volume | double |
| 保价金额 | ProValue | double |
| 填写订单的日期 | OrderDate | date |
| 运费 | Charge | double |

ProblemOrder（问题件）

|  |  |  |
| --- | --- | --- |
| 名称 | 名 | 类型 |
| 问题件编号 | ProID | varchar |
| 问题类型 | Type | varchar |
| 运单编号 | OrderID | varchar |
| 状态（问题件是否被处理） | State | varchar |

Account（账户）

|  |  |  |
| --- | --- | --- |
| 名称 | 名 | 类型 |
| 编号 | ID | int |
| 日期 | Date | date |

Customer(客户)

|  |  |  |
| --- | --- | --- |
| 名称 | 名 | 类型 |
| 编号 | ID | int |
| 用户名 | Name | varchar |
| 密码 | Password | varchar |
| 地址 | Address | varchar |
| 联系方式 | TelePhone | varchar |
| 运单编号 | OrderID | varchar |

Employee（员工）

|  |  |  |
| --- | --- | --- |
| 名称 | 名 | 类型 |
| 编号 | ID | int |
| 用户名 | Name | varchar |
| 密码 | Password | varchar |
| 职位 | Position | varchar |
| 联系方式 | TelePhone | varchar |
| 业绩 | Performance | varchar |

职位：

Deman（快递员），Receptionist(业务员)，Supervisor(监管人员)，FinancialStaff（财务人员），ProblemDepartment（问题件处理部门），SysManage（系统管理员），Leader（领导）

9.1.2.控制数据完整性：

9.1.2.1默认值：

默认值是没有明确输入一个域的值时该域将采用的值。在本系统设计中使用到默认值的例子有：Logistics表中的first（始发地），因为如果快递是从同一个城市发出，始发地都是一样的，给first赋予一个默认值可以减少数据输入时间和数据输入错误。

9.1.2.2格式控制：

有些数据必须遵从规定的模式。在本系统中各个表中的日期，时间都是遵从特定的日期时间格式的。

9.1.2.3范围控制：

范围控制是数字和字母数据可能具有一个有限集合的允许值。当前日期只能是从1月1日到12月31日之间，时间时，分，秒分别只能是0-24,0-60,0-60；

货物的数量，重量，体积，保价金额都必须大于0。

9.1.2.4参照完整性：

在Logistics（物流信息表），ProblemOrder（问题件表），Customer（客户表）中的外键OrderID的值只限于来自Order（运单表）关系OrderID的值的集合，因为问题件以及客户查询到的快递物流信息是必须在运单中存在的快递，否则，这样的快递是没有存在意义的。

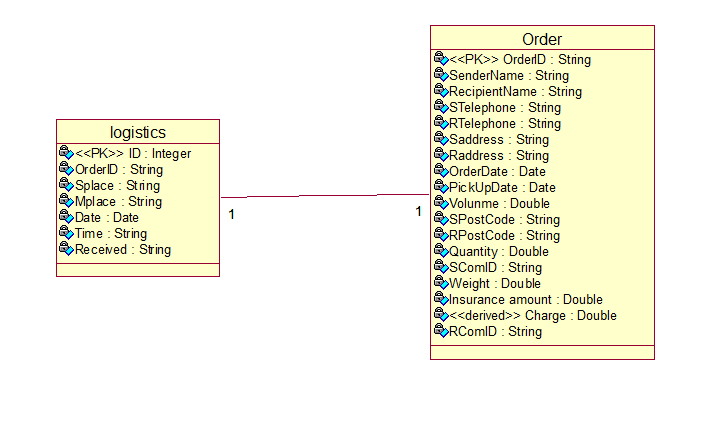
9.1.2.5空值控制

空值是一个特殊的域值，不同于0，空白或任何其他值，它表明缺少或者要不然不知道该域的值。在Logistics(物流信息表) 、ProblemOrder（问题件）、Account（账户）、Customer(客户)、Employee（员工）中，各个域均不为空值；Order（运单）中只有Provalue可以为空。

9.2.物理表的设计与反规范化：

反规范化是根据行和域使用的密切关系并根据标的检索和更新频率将规范化关系合并为物理表的过程。它以功能为代价，优化了某些数据处理功能。在不通标的合并中，当表的检索超过更新相当多时，反规范化最有用。反规范化可能合理的三种情况为：有一个一对一关系的两个实体、有非主键的属性的一个多对多关系（关联实体类）、一个一对多关系。

(1)有一个一对一关系的两个实体



Logistics（物流关系）和 Order(订单信息) 是一对一的关系，可以进行反规范化。

规范化的关系：

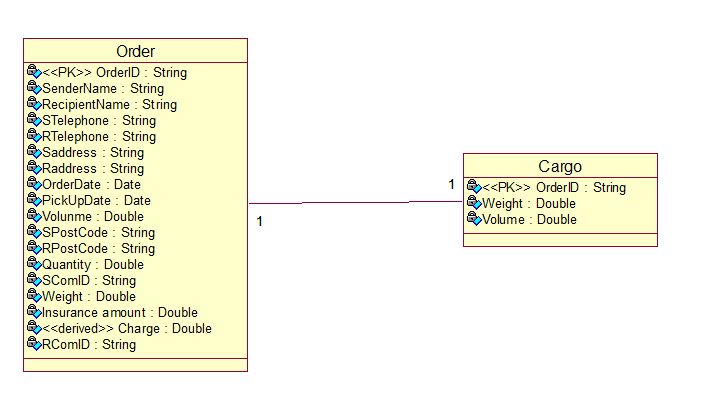
Logistics（ID,OrderID,Splace,Mplace,date,time,received)主键为ID

Order(OrderID,Sender,Recipients,STelephone,RTelephone,SComName,RComName,Saddress,Raddress，SPostCode,RPostCode,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,PickUpDate,Charge)主键为OrderID

反规范化的关系：

Order(OrderID,Sender,Recipients,STelephone,RTelephone,SComName,RComName,Saddress,Raddress，SPostCode,RPostCode,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,PickUpDate,Charge，logistics\_ ID,

OrderID,Splace,Mplace,date,time,received)



Order(订单信息)和Cargo(货物)是一对一的关系，可以进行反规范化。

规范化的关系：

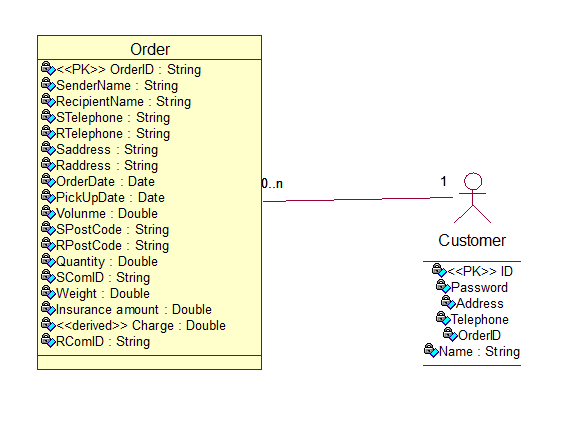
Order(OrderID,Sender,Recipients,STelephone,RTelephone,SComName,RComName,Saddress,Raddress，SPostCode,RPostCode,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,PickUpDate,Charge)主键为OrderID

Cargo(OrderID,Weight,Volume) 主键为OrderID

反规范化的关系：

Order(OrderID,Sender,Recipients,STelephone,RTelephone,SComName,RComName,Saddress,Raddress，SPostCode,RPostCode,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,PickUpDate,Charge)

(2)一个一对多关系



Customer(客户)和Order(订单信息)是一对多的关系，可以进行反规范化。

规范化的关系：

Order(OrderID,Sender,Recipients,STelephone,RTelephone,SComName,RComName,Saddress,Raddress，SPostCode,RPostCode,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,PickUpDate,Charge)主键为OrderID

Customer(ID,Name,Password,Address,TelePhone,OrderID)主键为： ID

反规范化的关系：

Order(OrderID,Sender,Recipients,STelephone,RTelephone,SComName,RComName,Saddress,Raddress，SPostCode,RPostCode,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,PickUpDate,Charge，customer\_ID,Name,Password,Address,TelePhone) 主键为OrderID

9.3.文件组织：

操作系统安排一个文件中的表行的方式被称为文件组织。文件组织包括下列步骤：

1. 估计行的大小和数目。
2. 计算分块因子和块的数目；
3. 计算每个表的扫描时间；
4. 估计Select和Update查询的频率；
5. 将小的频繁使用的表保存在高速缓存中；
6. 确定要索引的域。

针对快递管理信息系统，首先确定各个表的信息如下表

|  |
| --- |
| 表名 行对象数目 每个行对象的字节数 |
| |  | | --- | | Logistics 200000 135  Order 500000 203 |   ProblemOrder 10000 80  Account 300000 11  Customer 500000 104  Employee 700000 104 |

我们选用的块的大小有4000个可用字节，扫描每个块所用时间为2.5ms，计算得到表特征如下：

|  |
| --- |
| **表名 行对象数目 每个行对象的字节数 分块因子（BF） 块数 扫描时间（秒） 是否需要建立索引** |
| |  | | --- | | **Logistics 200000 135 29 6897 17.3 是**  **Order 500000 203 19 26316 65.8 是** |   **ProblemOrder 10000 80 50 200 0.5 否**  **Account 300000 11 363 827 2.1 是**  **Customer 500000 104 38 13158 32.9 是**  **Employee 700000 104 38 18422 46.1 是** |

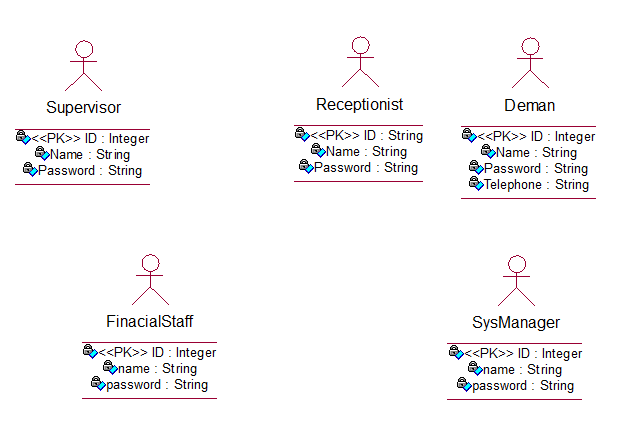
从上面的计算可以看出，Logistics，Order, Customer, Employee，Account是大表，需要建立索引来加快查询的速度，ProblemOrder表只有200个块，可以被存储在数据库服务器的高速缓冲区中。由于主键是被常访问的属性，因此可以索引这五个表的主键，当在主键域输入任何值的时候，都需要检查是否已经存在相同值。

9.4. 对象关系特征设计

对象关系DBMS支持诸如泛化、聚合、多值属性、对象标识符和引用关系等特征，还支持大对象。另外，可以安装扩充（有时被称为模块）来处理特殊的应用程序，比如空间坐标数据。

（1）泛化

泛化基于继承，在Oracle\Informix和其他一些对象关系DBMS中都支持它。



这些工作人员实体的属性很相似，可以创建一个employee的父类型，在它下面创建SysManager、Superviser、Receptionist、Deman、FinancialStaff、SysManager等实体类。

（2）聚合

聚合是一种强连接的关联。如果两个对象时是强连接的，并且对一个进行操作时，可能对另外一个也应用同样的操作。一种选择是将对象簇集在一起（即将它们放在物理上接近的位置）。还可以通过创建一个嵌套表来设计聚合。

Order(订单)和Cargo(货物)是强连接的，将Cargo表创建为Order表内的一种变长行数组是可能的。嵌套的表被定义为主表中的一个列。Order的数据将和Cargo分开存储，而这个关系将通过使用指针维护。这样的组织也提供了从一个订单到连接的货物对象的方便访问。然后嵌套的表和其他表有关联时，做嵌套表可能不很方便。

(3)多值属性

在嵌套表的情况下，嵌套表的多个行对应于主表的一个行。在多值属性的情况下，一个属性的多个值对应于一个表的一行。多值属性在传统的关系模型中是不允许的，所以这个属性的实现代表了从纯关系系统到对象关系系统的重大转移。然而，如果大量的查询是基于多值属性的，那么可以用一种传统的关系方式来吹多值属性（即创建一个附加表）。以这种方式，对象关系系统可以提供一个在最好的关系特征和对象关系特征之间的选择。

如客户信息中，客户的住址、电话都可以是多值的。如果需要基于一个多值属性进行频繁的查询，那么设计者需要在为该属性创建一个分离表的传统方法和对象关系方法中进行选择。再者，设计者需要保证这样的查询在实现环境中可以运转。多值属性情况下的性能益处是显而易见的。多值属性是表的一部分，不需要在原始表和因为多值属性而创建的分离的表之间进行连接。当检索对象时，多值属性也被检索，这就导致了有效的查询处理。

(4)对象标识符

每个行对象有一个对象标识符，在创建该行的时候，它有ORDBMS赋值，并且在改行的生命期间不能被改变。对象标识符用于维护使用引用的关系，不会在查询中被用到。因而将有两个主键，一个是对象标识符，第二个是可能由使用数列产生的传统主键。

(5)引用关系

可以通过使用引用建立关系。在定义这样一种关系之前，先要将被引用的表定义为对象，这样它的行对象就具有对象标识符。引用关系方法提供了一种快速访问的方法。

## *实验10确定系统构架等设计元素、设计类图建模*

10.1. 系统体系结构、构件、框架

（1）系统体系结构

设计体系结构是系统的一个总体蓝图。设计体系结构的确定以诸如应用程序的复杂性、接受变更的能力和项目团队的技术水平等因素为基础。在进行系统设计时，常用的体系结构有：两层体系结构、三层体系结构、N层体系结构。

我们选择的是三层体系结构。三层体系结构具有表示层、业务逻辑层和数据层。表示层处理用户界面，用户界面以一般是图形化的。包括文本框、组合和列表框、标签、网格和图片，还包括数据录入检查和规则。表示层可以用程序设计语言开发，如VB和Java，或者被嵌入在Web浏览器中，仅仅使用HTML和JavaScript这样的脚本语言。业务规则层包括业务逻辑，还包括与一端的表示鞥和另一端的数据库层交互的类和构件。

三层结构具有以下优点:

1. 该体系结构是松散耦合的。可以对一个层进行修改，而且，如果结构不变，修改就应该引起对系统中其它层的最小修改。
2. 该体系结构导致的不仅是数据的复用，而且是业务规则的复用。
3. 处理负载遍布在多个机器上，因此，系统就更加可伸缩。

和两层系统比，三层体系结构更加复杂，并且为了表示数据库和业务逻辑层之间的通信量，必须充分地设计网络。

（2）构件

构件具有下列特征：

1. 一个特定目标。
2. 一组类和接口。
3. 隐藏的实现细节。
4. 即插即用的能力。
5. 通常以二进制代码的形式分布。
6. 总是在第三方购买。
7. 可以在各种各样的应用程序中复用。
8. 在组织内部开发更昂贵。
9. 通常经过良好测试并相对无错误。
10. 可以用任何语言编写，只要该语言支持特定构件标准的接口规范。

我们用到的构件有：JavaBean、Servlet、JSP。JavaBean是以Java语言中的类和对象为基础定义的，主要用于存储状态信息，易于维护、使用、编写，可实现代码的重用性，可移植性强，便于传输，可以以其他部件的模式进行工作。Servlet主要用来处理客户端的请求并将其结果发送到客户端。当用户发送一个请求到某个Servlet的时候，Servlet容器会创建一个ServletRequst和ServletResponse对象。在ServletRequst对象中封装了用户的请求信息，然后Servlet容器把ServletRequst和ServletResponse对象传给用户所请求的Servlet，Servlet把处理好的结果写在ServletResponse中，然后Servlet容器把响应结果传给用户。JSP的优点有：一次编写，可以在很多地方运行；系统的多平台支持；强大的可伸缩性；多样化和功能强大的开发工具支持；支持服务端组件。

（3）框架

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。Model（模型）是应用程序中用于处理应用程序数据逻辑的部分,通常负责在数据库中存取数据。View（视图）是应用程序中处理数据显示的部分,通常依据模型数据创建的。Controller（控制器）是应用程序中处理用户交互的部分。通常负责从视图读取数据，控制用户输入，并向模型发送数据。MVC的优点包括：耦合性低，重用性高，生命周期成本低，部署快，可维护性高，有利于软件工程化管理等。

10.2． 设计类图

设计类图：



























































## *实验11界面设计*

11.1 表单和报表设计

11.1.1. 表单设计

|  |
| --- |
| 表单：客户登录  用户：选择XX快递公司实现快递业务的顾客代表  任务：实现XX快递公司客户服务系统的用户的登录  系统：Microsoft Windows  环境：标准办公环境 |
| C:\Users\Administrator\Desktop\QQ图片20160105232409.jpg |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.35  充分性[1=一致 到 5=不一致] 1.27  准确性[1=一致 到 5=不一致] 1.49 |

|  |
| --- |
| 表单：客户注册  用户：选择XX快递公司实现快递业务的顾客代表  任务：实现XX快递公司客户服务系统的用户的注册  系统：Microsoft Windows  环境：标准办公环境 |
| IMG_256 |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.38  充分性[1=一致 到 5=不一致] 1.26  准确性[1=一致 到 5=不一致] 1.47 |

|  |
| --- |
| 表单：客户服务系统主菜单  用户：选择XX快递公司实现快递业务的顾客代表  任务：支持已登录XX快递公司客户服务系统的用户选择功能实现相应的业务  系统：Microsoft Windows  环境：标准办公环境 |
| IMG_256 |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.34  充分性[1=一致 到 5=不一致] 1.31  准确性[1=一致 到 5=不一致] 1.47 |

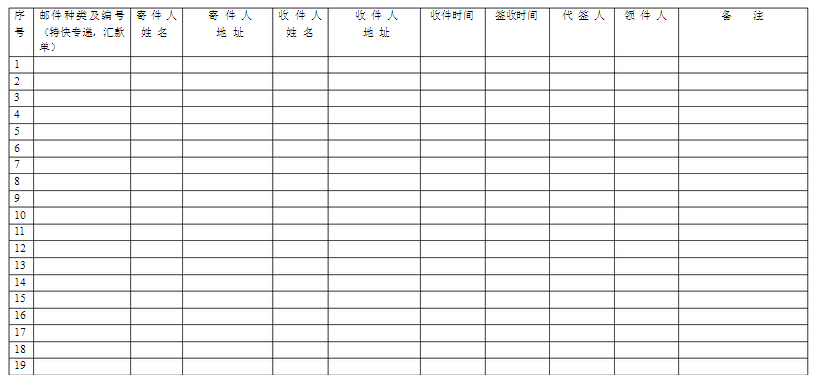
|  |
| --- |
| 表单：客户个人中心  用户：选择XX快递公司实现快递业务的顾客代表  任务：显示已登录XX快递公司客户服务系统的用户的个人及订单信息  系统：Microsoft Windows  环境：标准办公环境 |
| IMG_256 |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.38  充分性[1=一致 到 5=不一致] 1.26  准确性[1=一致 到 5=不一致] 1.47 |

|  |
| --- |
| 表单：客户下订单  用户：选择XX快递公司实现快递业务的顾客代表  任务：支持已登录XX快递公司客户服务系统的用户实现网上下订单功能  系统：Microsoft Windows  环境：标准办公环境 |
| IMG_256 |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.32  充分性[1=一致 到 5=不一致] 1.28  准确性[1=一致 到 5=不一致] 1.39 |

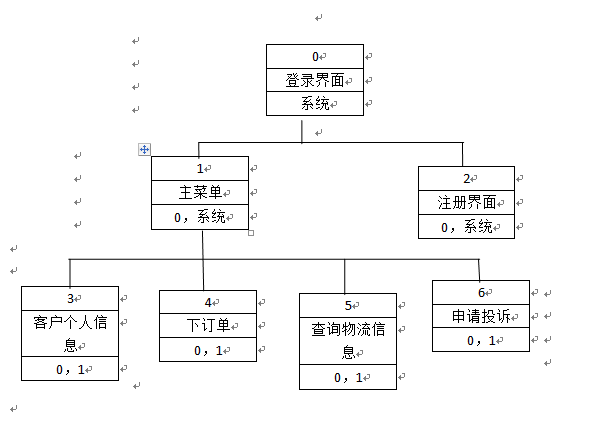
|  |
| --- |
| 表单：客户查询物流信息  用户：选择XX快递公司实现快递业务的顾客代表  任务：支持已登录XX快递公司客户服务系统的用户实现对物流信息的查询  系统：Microsoft Windows  环境：标准办公环境 |
| IMG_256 |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.34  充分性[1=一致 到 5=不一致] 1.28  准确性[1=一致 到 5=不一致] 1.47 |

|  |
| --- |
| 表单：客户申请投诉界面  用户：选择XX快递公司实现快递业务的顾客代表  任务：支持已登录XX快递公司客户服务系统的用户对问题件申请投诉  系统：Microsoft Windows  环境：标准办公环境 |
| IMG_256 |
| 用户评定的感知（平均15个用户）  一致性[1=一致 到 5=不一致] 1.29  充分性[1=一致 到 5=不一致] 1.32  准确性[1=一致 到 5=不一致] 1.44 |

11.1.2. 报表设计



11.2. 对话设计



## *实验12 代码实现*

我们实现了系统的部分功能，以下是结果截图和部分代码。

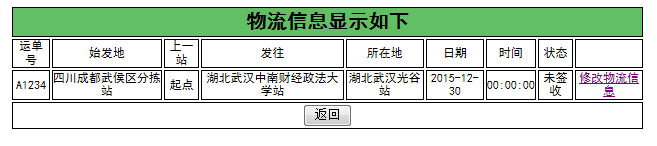
登录页面，输入ID和密码，选择权限进行登录



进入业务员界面，在左上角显示业务员所属地点



点击查看物流信息按钮，显示即将到达此地点的快递信息



点击录入运单信息，可以填写运单信息



登录界面选择监管人员



进入监管人员界面



点击查看订单信息，可以查看所有运单信息





点击录入发货信息，进入录入发货信息界面



代码：

**连接数据库：**

**package** Exp.util;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**publicclass** DButil {

**privatestatic** Connection *connection*;

**private** Statement statement;

**private** ResultSet resultSet;

**private** PreparedStatement preparedStatement;

**static**{

**try**{

//连接数据库

Class.*forName*("com.mysql.jdbc.Driver");

*connection* = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/expressmanagement?&characterEncoding=utf-8","root","MySQL");

**if**(!*connection*.isClosed())

{

//打印连接数据库成功字样

System.***out***.println("success");

}

}

**catch** (ClassNotFoundException e) {

e.printStackTrace();

}

**catch** (SQLException e) {

e.printStackTrace();

}

}

**publicvoid** close(){

//关闭数据库

**try** {

**if** (resultSet!=**null**) {

resultSet.close();

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**try** {

**if** (statement!=**null**) {

statement.close();

}

} **catch** (SQLException e) {

e.printStackTrace();

}

**try** {

**if** (preparedStatement!=**null**) {

preparedStatement.close();

}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**publicvoid** preparSql(Object[] obj,**int**length)**throws** SQLException{

//判断对象所属类型并为sql语句的？赋值

**for**(**int**i=0;i<length;i++){

System.***out***.println(i);

**if**(obj[i].getClass()==String.**class**){

preparedStatement.setString(i+1, obj[i].toString());

}

**elseif** (obj[i].getClass()==Integer.**class**) {

preparedStatement.setInt(i+1, (Integer)obj[i]);

}

**elseif**(obj[i].getClass()==Double.**class**){

preparedStatement.setDouble(i+1, (Double)obj[i]);

}

}

}

**public** ResultSet excuteSql(String sql) {

**try** {

statement=(Statement)*connection*.createStatement();

resultSet=statement.executeQuery(sql);

} **catch** (Exception e) {

e.printStackTrace();

}

**return**resultSet;

}

**publicint** excute(String sql,Object[] obj) {

//带参数的增删改查

**int**row=0;

**int**len=obj.length;

System.***out***.println(len);

**try** {

preparedStatement=*connection*.prepareStatement(sql);

preparSql(obj, len);

row=preparedStatement.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

}

**finally**{

close();

}

**return**row;

}

}

**登录servlet：**

**package** Exp.servlet;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** Exp.dao.UserDao;

/\*\*

\* Servlet implementation class LoginAction

\*/

@WebServlet("/LoginAction")

**publicclass** LoginAction **extends** HttpServlet {

**privatestaticfinallong*serialVersionUID*** = 1L;

**private** UserDao userdao;

**protectedvoid** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doPost(request, response);

}

**protectedvoid** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

String uID=request.getParameter("ID");

String upassword=request.getParameter("password");

String uposition=request.getParameter("level");

userdao=**new** UserDao();

System.***out***.println(uID);

System.***out***.println(upassword);

System.***out***.println(uposition);

**if** ("".equals(uID)||"".equals(upassword))

{

response.sendRedirect("index.jsp");

}

**else**{

**if**(userdao.VerifyLogin(uID, upassword,uposition))

{

**if**("业务员".equals(uposition))

{

String area=userdao.Area(uID);

System.***out***.println(area);

request.getSession().setAttribute("area", area);

response.sendRedirect("Receptionist.jsp");

}

**if**("监管人员".equals(uposition))

{

String area=userdao.Area(uID);

System.***out***.println(area);

request.getSession().setAttribute("area", area);

response.sendRedirect("Supervisor.jsp");

}

}

**else**

{

String message="Failure";

System.***out***.println(message);

request.setAttribute("message", message);

response.sendRedirect("Index.jsp");

}

}

}

}

**业务员servlet：**

**package** Exp.servlet;

**import** java.io.IOException;

**import** java.util.List;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** Exp.bean.LogisticsBean;

**import** Exp.bean.OrderBean;

**import** Exp.dao.ReceptionistDao;

/\*\*

\* Servlet implementation class ReceptionistServlet

\*/

@WebServlet("/ReceptionistServlet")

**publicclass** ReceptionistServlet **extends** HttpServlet {

**privatestaticfinallong*serialVersionUID*** = 1L;

**private** ReceptionistDao receptionistdao;

**protectedvoid** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doPost(request,response);

}

**protectedvoid** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

String str=request.getParameter("str");

**if**("insert".equals(str)){

receptionistdao=**new** ReceptionistDao();

String OrderID1=request.getParameter("OrderID1");

String Sender=request.getParameter("Sender");

String STelephone=request.getParameter("STelephone");

String Saddress=request.getParameter("Saddress");

String Recipients=request.getParameter("Recipients");

String RTelephone=request.getParameter("RTelephone");

String Raddress=request.getParameter("Raddress");

String Cargo=request.getParameter("Cargo");

**int**Quantity=Integer.*parseInt*(request.getParameter("Quantity"));

**double**Weight=Double.*parseDouble*(request.getParameter("Weight"));

**double**Volume=Double.*parseDouble*(request.getParameter("Volume"));

**double**ProValue=Double.*parseDouble*(request.getParameter("ProValue"));

String OrderDate=request.getParameter("OrderDate");

**double**Charge=Double.*parseDouble*(request.getParameter("Charge"));

OrderBean orderbean=**new** OrderBean();

orderbean.setOrderID(OrderID1);

orderbean.setSender(Sender);

orderbean.setStelephone(STelephone);

orderbean.setSaddress(Saddress);

orderbean.setRecipients(Recipients);

orderbean.setRtelephone(RTelephone);

orderbean.setRaddress(Raddress);

orderbean.setCargo(Cargo);

orderbean.setQuantity(Quantity);

orderbean.setWeight(Weight);

orderbean.setVolume(Volume);

orderbean.setProValue(ProValue);

orderbean.setOrderDate(OrderDate);

orderbean.getOrderDate();

orderbean.setCharge(Charge);

**if**(receptionistdao.insertorder(orderbean)){

System.***out***.println("success");

response.sendRedirect("success.jsp");

}

**else** {

response.sendRedirect("failure.jsp");

}

}

**elseif**("update".equals(str)){

receptionistdao=**new** ReceptionistDao();

String OrderID2=request.getParameter("OrderID2");

String first=request.getParameter("first");

String next=request.getParameter("next");

String before=request.getParameter("before");

String Location=request.getParameter("Location");

String date=request.getParameter("date");

String time=request.getParameter("time");

String received=request.getParameter("received");

**if**("已签收".equals(received)){

received="已签收";

}**else**{

received="未签收";

}

LogisticsBean logisticsbean=**new** LogisticsBean();

logisticsbean.setOrderID(OrderID2);

logisticsbean.setFirst(first);

logisticsbean.setNext(next);

logisticsbean.setBefore(before);

logisticsbean.setLocation(Location);

logisticsbean.setDate(date);

logisticsbean.setTime(time);

logisticsbean.setReceived(received);

**if**(receptionistdao.updatelog(logisticsbean)){

response.sendRedirect("success.jsp");

}

**else**{

response.sendRedirect("failure.jsp");

}

}

**elseif**("select".equals(str)){

receptionistdao=**new** ReceptionistDao();

String area=request.getParameter("area");

List<LogisticsBean>list=receptionistdao.selectlog(area.toString());

request.setAttribute("ulist", list);

request.getRequestDispatcher("Rsellog.jsp").forward(request, response);

}

**elseif**("select1".equals(str)){

receptionistdao=**new** ReceptionistDao();

String OrderID=request.getParameter("OrderID");

String next1=request.getParameter("next");

String next=**new** String(next1.getBytes("ISO-8859-1"),"UTF-8");

System.***out***.println(OrderID);

System.***out***.println(next1);

LogisticsBean logisticsbean=receptionistdao.selectlog1(OrderID,next);

request.setAttribute("logisticsbean", logisticsbean);

request.getRequestDispatcher("Rupdate.jsp").forward(request, response);

}

}

}

**监管人员servlet：**

package Exp.servlet;

import java.io.IOException;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import Exp.bean.LogisticsBean;

import Exp.bean.OrderBean;

import Exp.dao.SupervisorDao;

/\*\*

\* Servlet implementation class SupervisorServlet

\*/

@WebServlet("/SupervisorServlet")

public class SupervisorServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

private SupervisorDao supervisordao;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

doPost(request,response);

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

response.setContentType("text/html;charset=utf-8");

supervisordao=new SupervisorDao();

String str=request.getParameter("str");

if("insert".equals(str)){

String OrderID1=request.getParameter("OrderID");

String Location=request.getParameter("Location");

String first=request.getParameter("first");

String before=request.getParameter("before");

String next=request.getParameter("next");

String date=request.getParameter("date");

String time=request.getParameter("time");

String received=request.getParameter("received");

LogisticsBean logisticsbean=new LogisticsBean();

logisticsbean.setOrderID(OrderID1);

logisticsbean.setLocation(Location);

logisticsbean.setFirst(first);

logisticsbean.setBefore(before);

logisticsbean.setNext(next);

logisticsbean.setDate(date);

logisticsbean.setTime(time);

if("未签收".equals(received)){

received="未签收";

}else{

received="已签收";

}

logisticsbean.setReceived(received);

if(supervisordao.insertlog(logisticsbean)){

response.sendRedirect("success.jsp");

}

else {

response.sendRedirect("failure.jsp");

}

}

else if("selectall".equals(str)){

List<OrderBean> list=supervisordao.SelectAllOrder();

request.setAttribute("ulist", list);

request.getRequestDispatcher("selectall.jsp").forward(request, response);

}

else if("selectlog".equals(str)){

String OrderID=request.getParameter("OrderID");

request.getSession().setAttribute("OrderID", OrderID);

List<LogisticsBean> list=supervisordao.SelectLog(OrderID);

request.setAttribute("ulist", list);

request.getRequestDispatcher("sellog.jsp").forward(request, response);

}

}

}

**登录方法：**

package Exp.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import Exp.util.\*;

public class UserDao {

private DButil dbutil;

public boolean VerifyLogin(String ID,String password,String position)

{

dbutil=new DButil();

boolean flag = false;

String sql="select position from user where ID='"+ID+"'and password='"+password+"'";

ResultSet rs=dbutil.excuteSql(sql);

try{

if(rs.next())

{

flag=true;

}

}catch (SQLException e) {

e.printStackTrace();

}

return flag;

}

public String Area(String ID){

dbutil=new DButil();

String sql="select area from user where ID='"+ID+"'";

ResultSet rs=dbutil.excuteSql(sql);

String area=null;

try{

rs.next();

area=rs.getString(1);

}catch (SQLException e) {

e.printStackTrace();

}

return area;

}

}

**业务员方法：**

package Exp.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import Exp.bean.OrderBean;

import Exp.bean.LogisticsBean;

import Exp.util.\*;

public class ReceptionistDao {

private DButil dbUtil;

public boolean insertorder(OrderBean orderbean){

dbUtil=new DButil();

String sql="insert into order1(OrderID,Sender,STelephone,Saddress,Recipients,RTelephone,Raddress,Cargo,Quantity,Weight,Volume,ProValue,OrderDate,Charge)values(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

System.out.println(sql);

int row=dbUtil.excute(sql,new Object[]{orderbean.getOrderID(),orderbean.getSender(),orderbean.getStelephone(),orderbean.getSaddress(),orderbean.getRecipients(),orderbean.getRtelephone(),orderbean.getRaddress(),orderbean.getCargo(),orderbean.getQuantity(),orderbean.getWeight(),orderbean.getVolume(),orderbean.getProValue(),orderbean.getOrderDate(),orderbean.getCharge()});

if(row>0)

{

return true;

}

else

{

return false;

}

}

public boolean updatelog(LogisticsBean logisticsbean){

dbUtil=new DButil();

String sql="insert into logistics(OrderID,Location,first,next,before1,date,time,received)values(?,?,?,?,?,?,?,?)";

int row=dbUtil.excute(sql,new Object[]{logisticsbean.getOrderID(),logisticsbean.getLocation(),logisticsbean.getFirst(),logisticsbean.getNext(),logisticsbean.getBefore(),logisticsbean.getDate(),logisticsbean.getTime(),logisticsbean.getReceived()});

if(row>0)

{

return true;

}

else

{

return false;

}

}

public List<LogisticsBean> selectlog(String area){

dbUtil=new DButil();

List<LogisticsBean> list=new ArrayList<LogisticsBean>();

String sql="select \*from logistics where next='"+area+"'";

ResultSet rs=dbUtil.excuteSql(sql);

try {

while (rs.next()) {

int i=2;

LogisticsBean logisticsbean=new LogisticsBean();

logisticsbean.setOrderID(rs.getString(i++));

logisticsbean.setLocation(rs.getString(i++));

logisticsbean.setFirst(rs.getString(i++));

logisticsbean.setBefore(rs.getString(i++));

logisticsbean.setNext(rs.getString(i++));

logisticsbean.setDate(rs.getString(i++));

logisticsbean.setTime(rs.getString(i++));

logisticsbean.setReceived(rs.getString(i++));

list.add(logisticsbean);

}

}catch (SQLException e) {

e.printStackTrace();

}

finally{

dbUtil.close();

}

return list;

}

public LogisticsBean selectlog1(String OrderID,String next){

dbUtil=new DButil();

LogisticsBean logisticsbean=new LogisticsBean();

String sql="select \*from logistics where OrderID='"+OrderID+"'and next='"+next+"'";

ResultSet rs=dbUtil.excuteSql(sql);

try {

while (rs.next()) {

int i=2;

logisticsbean.setOrderID(rs.getString(i++));

logisticsbean.setLocation(rs.getString(i++));

logisticsbean.setFirst(rs.getString(i++));

logisticsbean.setBefore(rs.getString(i++));

logisticsbean.setNext(rs.getString(i++));

logisticsbean.setDate(rs.getString(i++));

logisticsbean.setTime(rs.getString(i++));

logisticsbean.setReceived(rs.getString(i++));

}

}catch (SQLException e) {

e.printStackTrace();

}

return logisticsbean;

}

}

**监管人员方法：**

**package** Exp.dao;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** Exp.bean.OrderBean;

**import** Exp.bean.LogisticsBean;

**import** Exp.util.\*;

**publicclass** SupervisorDao {

**private** DButil dbUtil;

**publicboolean** insertlog(LogisticsBean logisticsbean){

dbUtil=**new** DButil();

String sql="insert into logistics(OrderID,Location,first,before1,next,date,time,received)values(?,?,?,?,?,?,?,?)";

**int**row=dbUtil.excute(sql,**new** Object[]{logisticsbean.getOrderID(),logisticsbean.getLocation(),logisticsbean.getFirst(),logisticsbean.getBefore(),logisticsbean.getNext(),logisticsbean.getDate(),logisticsbean.getTime(),logisticsbean.getReceived()});

**if**(row>0)

{

**returntrue**;

}

**else**

{

**returnfalse**;

}

}

**public** List<OrderBean> SelectAllOrder(){

dbUtil=**new** DButil();

List<OrderBean>list=**new** ArrayList<OrderBean>();

String sql="select \*from order1";

ResultSet rs=dbUtil.excuteSql(sql);

**try** {

**while** (rs.next()) {

**int**i=1;

OrderBean orderbean=**new** OrderBean();

orderbean.setOrderID(rs.getString(i++));

orderbean.setSender(rs.getString(i++));

orderbean.setStelephone(rs.getString(i++));

orderbean.setSaddress(rs.getString(i++));

orderbean.setRecipients(rs.getString(i++));

orderbean.setRtelephone(rs.getString(i++));

orderbean.setRaddress(rs.getString(i++));

orderbean.setCargo(rs.getString(i++));

orderbean.setQuantity(rs.getInt(i++));

orderbean.setWeight(rs.getDouble(i++));

orderbean.setVolume(rs.getDouble(i++));

orderbean.setProValue(rs.getDouble(i++));

orderbean.setOrderDate(rs.getString(i++));

orderbean.setCharge(rs.getDouble(i++));

list.add(orderbean);

}

}**catch** (SQLException e) {

e.printStackTrace();

}

**finally**{

dbUtil.close();

}

**return**list;

}

**public** List<LogisticsBean> SelectLog(String OrderID){

dbUtil=**new** DButil();

List<LogisticsBean>list=**new** ArrayList<LogisticsBean>();

String sql="select Location,date,time from logistics where OrderID='"+OrderID+"' ORDER BY date ASC ";

ResultSet rs=dbUtil.excuteSql(sql);

**try**{

**while** (rs.next()) {

**int**i=1;

LogisticsBean logisticsbean=**new** LogisticsBean();

logisticsbean.setLocation(rs.getString(i++));

logisticsbean.setDate(rs.getString(i++));

logisticsbean.setTime(rs.getString(i++));

list.add(logisticsbean);

}

}**catch** (SQLException e) {

e.printStackTrace();

}

**finally**{

dbUtil.close();

}

**return**list;

}

}

**小组分工：**

项 英：负责电子商务平台的总体分析设计、三种可选设计策略、对三种可选设计策略的定性与定量分析、更新后的BPP

郭 雪：负责网络图、客户服务系统的总体分析设计、物理表的设计和反规范化、对象关系特征的设计、系统体系结构、构件、框架

陈雅楠：负责通过PERT技术估算的项目持续时间、核心业务系统的总体分析与设计、域的设计、文件组织、设计类图

曾子轩：负责进一步细化的项目进度安排、核心业务系统的总体分析与设计、代码实现

许子吟：负责WBS和所需资源、客户服务系统的总体分析设计、表单和报表设计、对话设计

李和嫒：负责领导决策支持系统的总体分析设计、系统需求汇总