



UML 系统分析设计案例——电子商务

目 录

前言	1
第一章 面向对象的 UML 建模	2
1.1 UML 建模语言简介	2
1.2 RUP 过程指导与本系统分析设计过程	3
第二章 仓储系统业务用例建模	5
2.1 仓储系统业务流程分析	5
2.1.1 入库流程分析	5
2.2 业务需求用例建模阶段	7
2.2.1 业务角色的查找及建立	7
2.2.2 业务用例查找与分析	7
2.2.3 业务用例图	7
2.2.3 业务活动图	8
第三章 仓储系统系统需求用例建模	9
3.1 入库管理需求用例分析	9
3.1.1 确定系统角色:	9
3.1.2 确定系统顶层用例	9
3.1.3 入库管理功能性分析:	9
3.2 系统整体功能描述	10
第四章 业务领域分析与设计	10
4.1 系统顺序图, 状态图	10
4.2 定义基本对象与类	12
4.3 入库系统类图:	13
4.4 系统设计顺序图, 入库类图	13
4.5 系统构架设计	14
第五章 系统实现测试与配置	15
5.1 系统实现的工具与技术	15
5.2 系统实现方式图	15
第六章 系统开发的思考	16
6.1 数据库设计问题	16
6.2 数据库访问设计问题	17
结束	17
参考文献	18



基于 UML 的仓储管理系统的设计与实现

面向对象技术是软件工程和过程工程领域中的重要技术，统一建模语言 UML 则是近几年国际上比较流行的面向对象的标准建模语言。本文阐述了基于 UML 建模的仓储管理信息系统的开发。

建模是开发优秀软件所有活动的核心部分，也是本文工作的重点所在。在开发中我们考虑了 RUP 迭代式开发的优点并与本仓储系统开发的特色相结合，提出了自己的一套 UML 的建模过程。基于这个过程我们从系统分析，设计，实现与测试，运用 UML 建模思想与各种模型对仓储系统进行了详细的描述。

上岸吧教育微信公众号：shang-an-ba



第一章 面向对象的 UML 建模

1.1 UML 建模语言简介

面向对象的分析与设计方法的发展在 80 年代末至 90 年代中出现了一个高潮, UML 是这个高潮的产物。它不仅统一了 Booch, Rumbaugh 和 Jacobson 的表示方法, 而且对其作了进一步的发展, 并最终统一为大众所接受的标准建模语言。

UML 语言具有以下特点:

1 UML 已经成为面向对象建模语言的标准

从理论角度讲, UML 是三位很有影响的面向对象方法大师以及许多大公司共同努力的成果, Booch, Rumbaugh 和 Jacobson 各自提出的方法及有关著作拥有不少用户和读者, 而 UML 是他们以往研究的继续。在定义模型元素语义的基础上定义建模语言, 使 UML 在理论上比较健全, 具有表达力强、清晰和一致的特点, UML 定义的各种图和各种建模元素是来自多种 OO 方法的研究成果汇集。

2 UML 缺乏过程指导

UML 并不是十全十美的东西, 最重要的一点是, 它只是一种建模语言而非建模方法。

3 UML 过于复杂

UML 在具有丰富的表达力的同时, 其复杂性也是惊人的。UML 的开发者称: 只要 UML 中 20% 的内容就可以表达实际应用中 80% 的系统。但 UML 中并没有给出哪些内容是必要的, 哪些是可选的。这给用户的使用带来了不小的麻烦。

UML 的模型主要是通过一系列相关的图来表达, 在 UML 中共定义了 9 种图, 它们分别从不同的侧面来描述系统, 下图为在 UML 中定义的视图与图:

主要的域	视图	图	主要概念
结构	静态视图	类图	类, 关联, 泛化, 依赖关系, 实现, 接口
	用例视图	用例图	用例, 参与者, 关联, 扩展, 包含, 泛化
	实现视图	构件图	构件, 接口, 依赖关系, 实现
	培植视图	配置图	节点, 构件, 依赖关系, 位置
动态	状态机视图	状态视图	状态, 事件, 转换, 动作
	活动视图	活动图	状态, 活动, 完成转换, 分叉, 结合
	交互视图	顺序图	交互, 对象, 消息, 激活
		协作图	协作, 交互, 协作角色, 消息
模型管理	模型管理视图	类图	包, 子系统, 模型



1.2 RUP 过程指导与本系统分析设计过程

UML 是一种建模语言而不是一种方法, UML 的表示法和规则能够用来为系统进行面向对象建模, 但并没有指定应用 UML 的过程和方法。1998 年正式颁布的 RUP (Rational Unified Process) 是 UML 设计者们共同定义的一个软件开发的公共过程框架。

统一过程主要包括四个阶段: 开始阶段、细化阶段、构建阶段和移交阶段。

1 进行业务流程建模

通过使用 UML 的活动图与用例图描述企业的业务流程来理解领域知识, 建立业务用例图; 在建模之前我们对仓储系统的业务过程辅助业务流程图来分析, 这也是我们在本次系统开发中的一创新点。

2 进行系统功能建模

寻找用例及其之间的关系(用例图), 通过详细描述用例来捕获系统的功能需求, 同时辅助活动图, 并建立系统用例图, 这是整个开发过程的基础;

3 进行领域建模

寻找对象与类, 主要是使用类图表现领域中各业务类之间的静态关系, 并用交互图、顺序图等具体描述类之间的交互以及对象的状态变化。主要涉及以下活动(并不一定是顺序的):

- 1> 分析用例以及业务领域
- 2> 发现对象, 为对象分类, 确定对象之间的交互
- 3> 确定类之间的关系
- 4> 定义类的属性和操作
- 5> 分析对象的状态变化

4 进行系统设计, 在系统分析的基础上进行:

系统架构的设计

进行高层的系统决策、确定基本的软件结构, 并对应用系统进行划分;

对象进一步设计

对领域模型中的业务类进行精化、调整, 增添界面类、控制类等用于实现的类。

数据模型设计

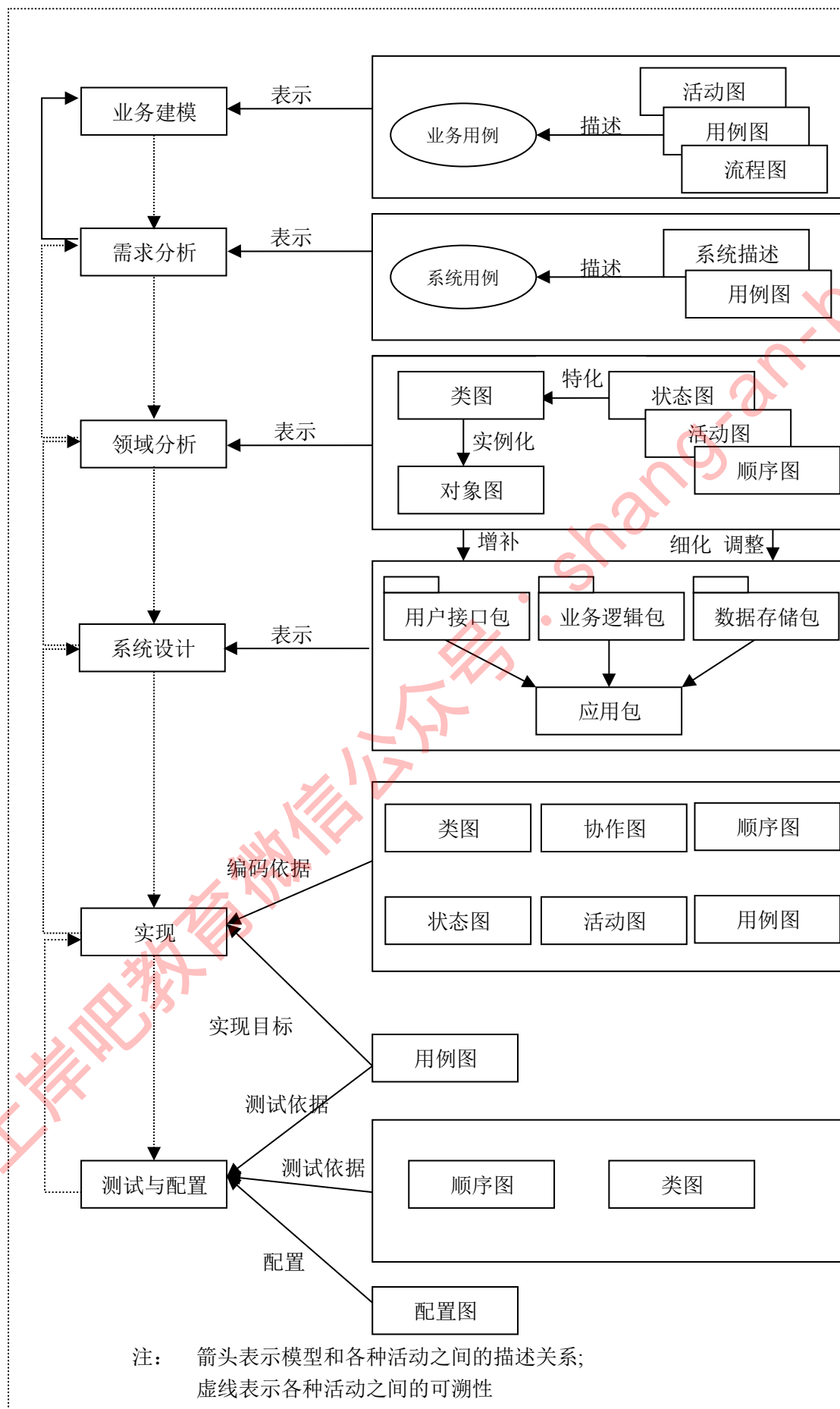
考虑解决持久对象到数据库的映射问题, 设计数据模型以及数据存取访问部分设计。

系统设计

得到的模型是精化和调整后的类图、交互图等, 同时还有对象持久设计得到的数据模型。



基于 UML 的仓储管理系统的设计与实现





5 实现

实现的依据是设计过程中得到的静态视图(类图、对象图)、动态视图(顺序图、状态图、协作图、活动图);同时可以将类映射为组件,进而使用 CASE 工具的框架代码自动生成的功能;同时通过实现图(组件图、配置图)来描述系统的物理视图。

6 单元测试,集成与系统测试

测试实现的部分是否满足用例的功能要求;另外,类图、组件图、协作图等也可以用来进行测试。

第二章 仓储系统业务用例建模

2.1 仓储系统业务流程分析

2.1.1 入库流程分析

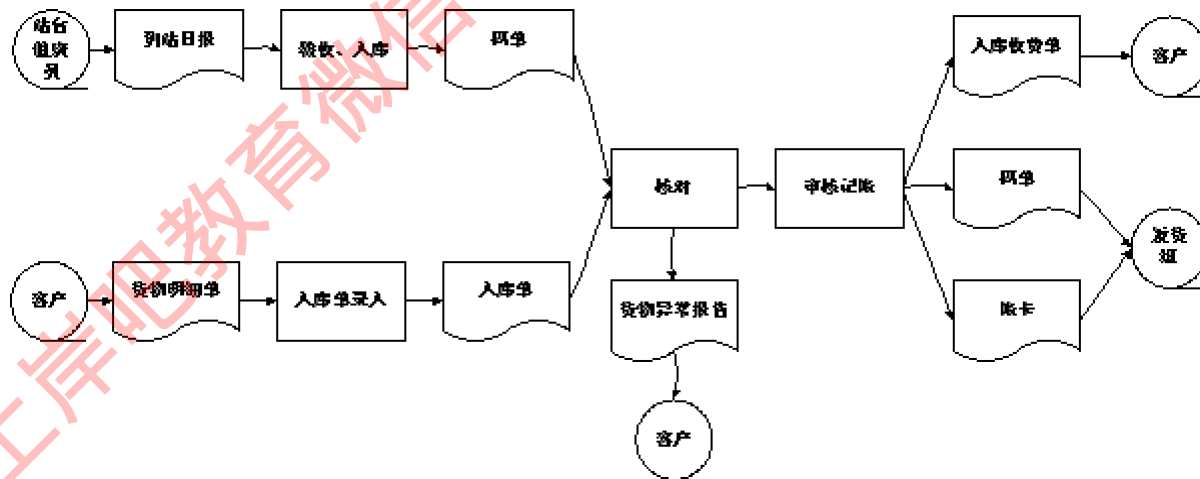
1、货物到达后,站台值班员组织卸货,大致清点品种、件数,编写《物资到站日报》,送至收货组;

2、收货组根据《物资到站日报表》核对验收货物,分配库位,填写《码单》。

3、客户《货物明细单》到达后,填写《入库单》。核对《码单》、《货物明细单》,《入库单》,如出现差错,返客户《货物异常报告》。

4、总经办审核记帐后,生成《帐卡》、《入库收费单》,数据不再允许改动。

入库业务流程图如下:



2.1.2 出库流程分析

1 客户在入库时提供提货样单,用户持提货单至发货组,要求提货。调度员核实提货单的合法性,填写《出库单》,显示相关业务号信息,如果有层次,则显示层次信息,并修改层次信息;如果是一个业务号的尾货则给出尾货提示。然后向用户出具《出库收费单》,同时生成《派车单》,并派保管员准备发货。

2 保管员持《派车单》组织货物装运,记录实出货物信息,填写《码单出库信息

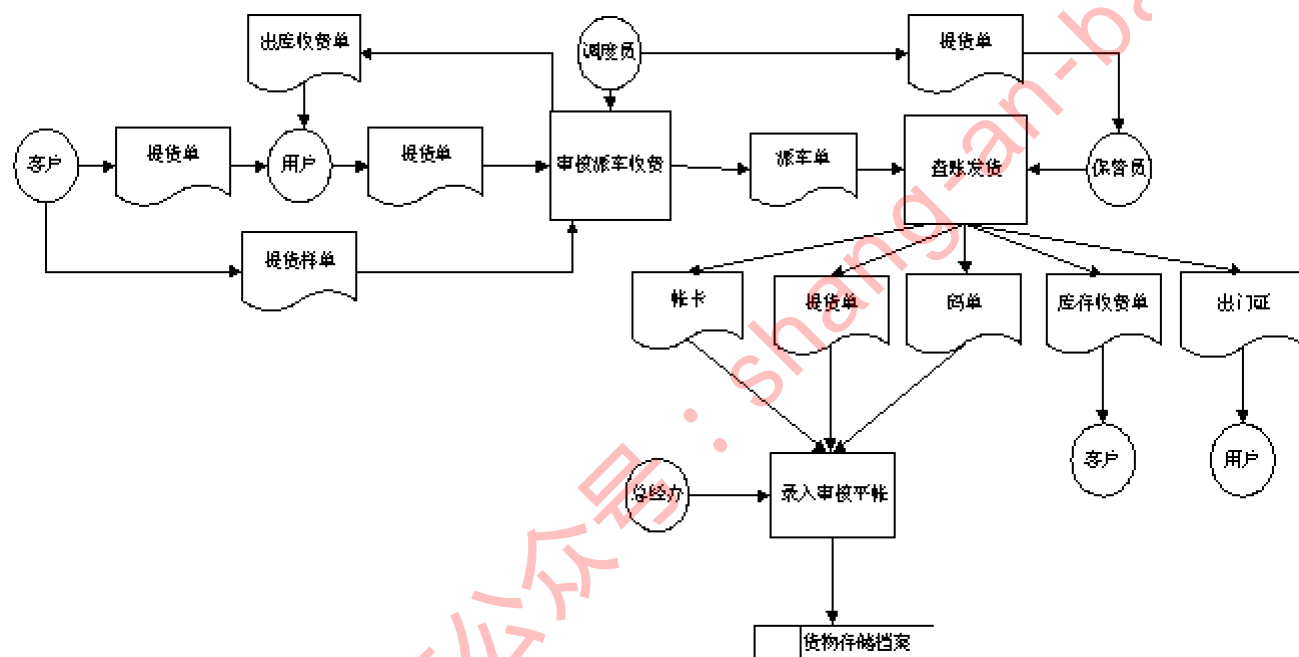


基于 UML 的仓储管理系统的设计与实现

表》。

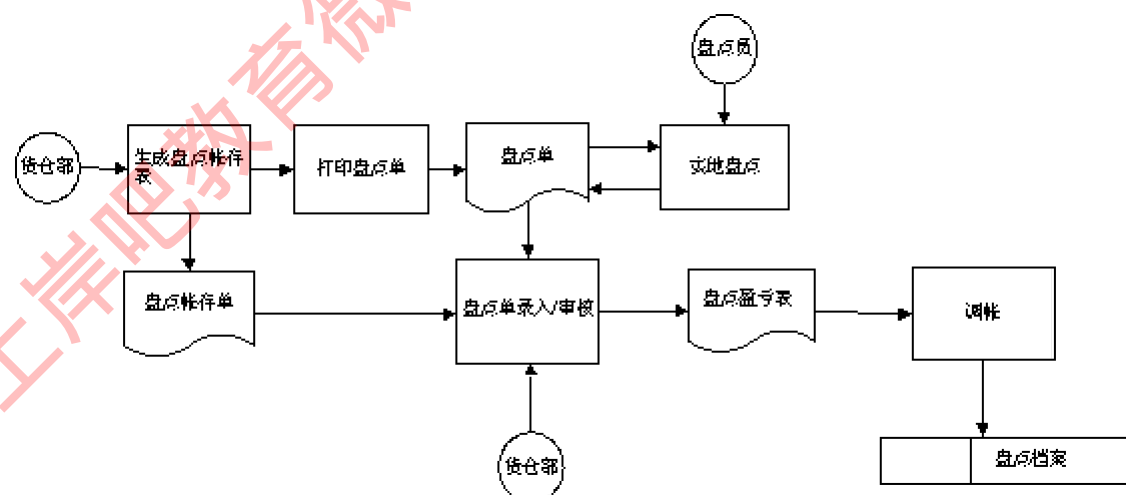
- 3 用户缴费后，保管员开具出门证给用户。
- 4 总经办审核记帐后，生成保管收费单。
- 5 一批货物全部出库后，保管员将《帐卡》、《提货单》、《码单》送至总经办核实平帐后，存档。

出库业务流程图如下：



2.1.3 库存管理业务流程分析

盘点业务流程图如下：





2.2 业务需求用例建模阶段

2.2.1 业务角色的查找及建立

根据此仓储管理的分析知，仓储管理说涉及的业务角色有：站台值班员，收货组，总经办，客户，保管员，货舱部，盘点员，调度员

2.2.2 业务用例查找与分析

根据业务流程可以找到基本的业务用例有很多，简要列举如下：

1 站台值班员卸货审核清点货物

货物到达后，站台值班员组织卸货，大致清点品种、件数，编写《物资到站日报》，送至收货组。

2 收货组验收入库货物分配库位

收货组根据《物资到站日报表》核对验收货物，分配库位，填写《码单》。

3 收货组审核入库单据处理异常

客户《货物明细单》到达后，填写《入库单》。核对《码单》、《货物明细单》，《入库单》，如出现差错，返客户《货物异常报告》。

4 总经办进行入库审核记帐

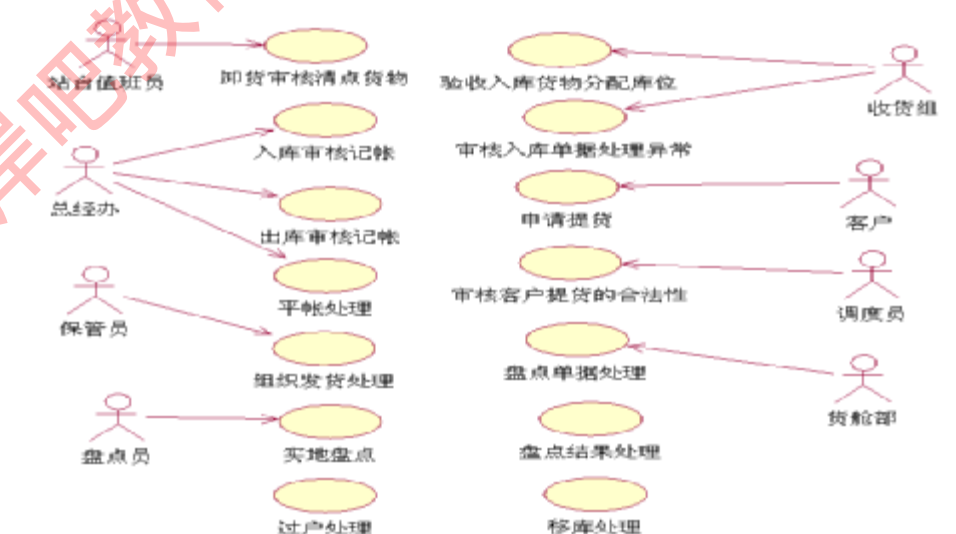
总经办审核记帐后，生成《帐卡》、《入库收费单》，数据不再允许改动。

注：客户提供提单的同时，提供一份提货样单。

5 客户申请提货

客户在入库时提供提货样单，用户持提货单至发货组，要求提货。

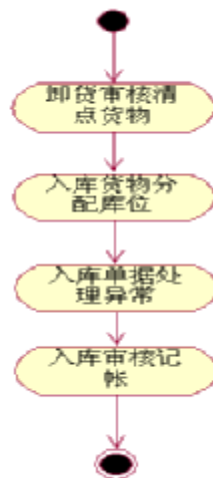
2.2.3 业务用例图



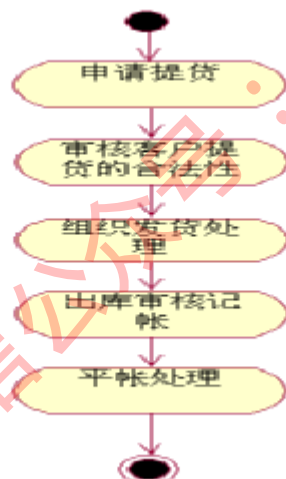


2.2.3 业务活动图

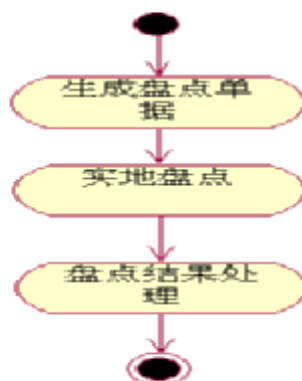
入库过程:



出库过程:



盘点过程:





第三章 仓储系统系统需求用例建模

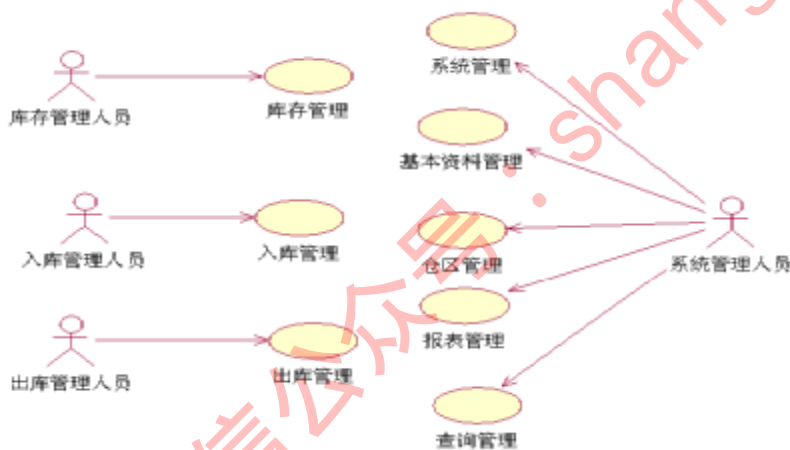
3.1 入库管理需求用例分析

3.1.1 确定系统角色：

一个角色可以是人、硬件设备、甚至另外一个系统。对于一个系统而言，每种类型的用户表示为一个或几个角色，每个与该系统交互的外部系统也表示为一个或几个角色。仓储管理系统中，我们通过分析，可以找出这样一些角色：客户，货物，入库管理人员，仓库，出库管理人员，库存管理人员，系统管理人员。

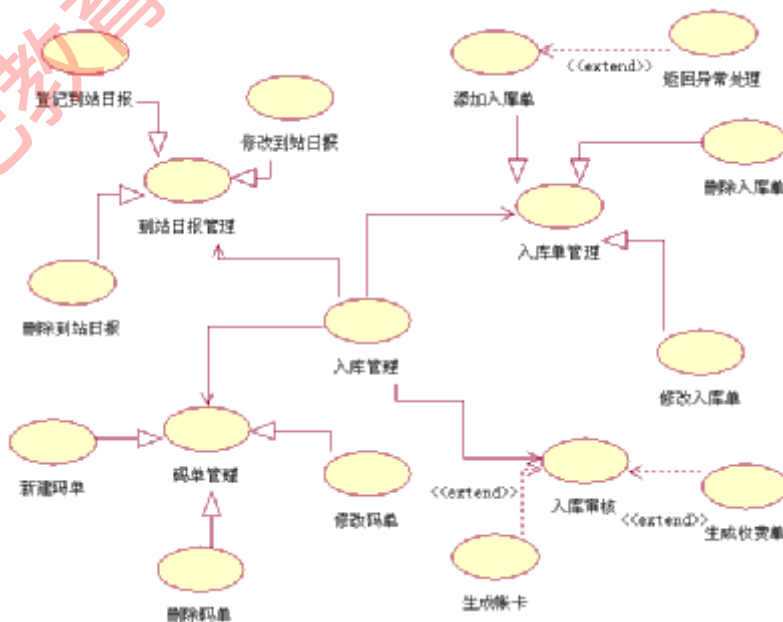
3.1.2 确定系统顶层用例

系统得顶层用例图如下：



3.1.3 入库管理功能性分析：

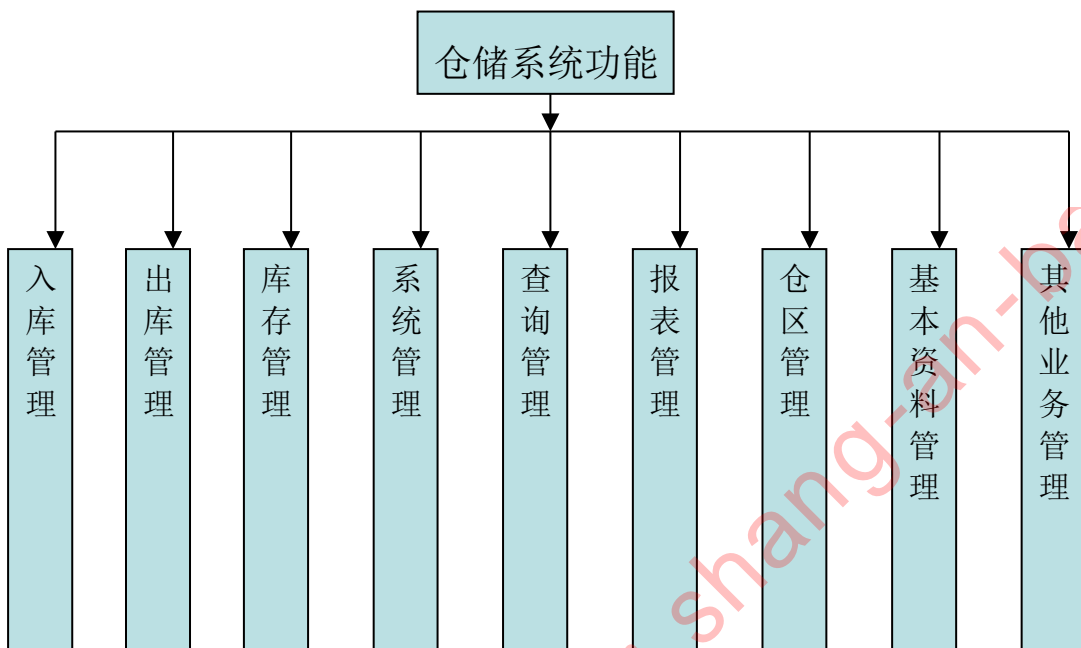
根据业务描述，我们可以确定入库管理的分层用例图如下：





3.2 系统整体功能描述

系统整体功能图如下：

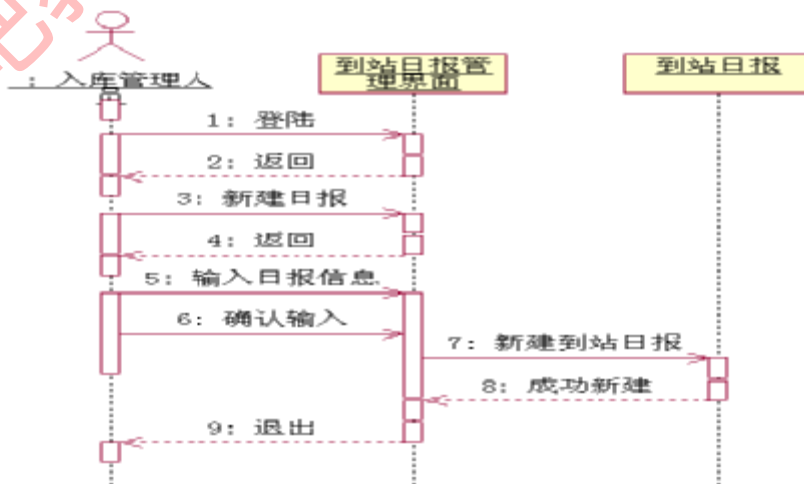


第四章 业务领域分析与设计

4.1 系统顺序图，状态图

权限管理对于系统来说十分重要，因为它关系到系统得安全性，因此在系统得开发与设计中，我们始终把系统的安全性放在十分重要的位置，因此我们在用顺序图分析的时候始终从系统用户未登陆状态来分析的，在后边系统详细设计时，我们会给出用户登录的顺序图；部分顺序图如下所示：

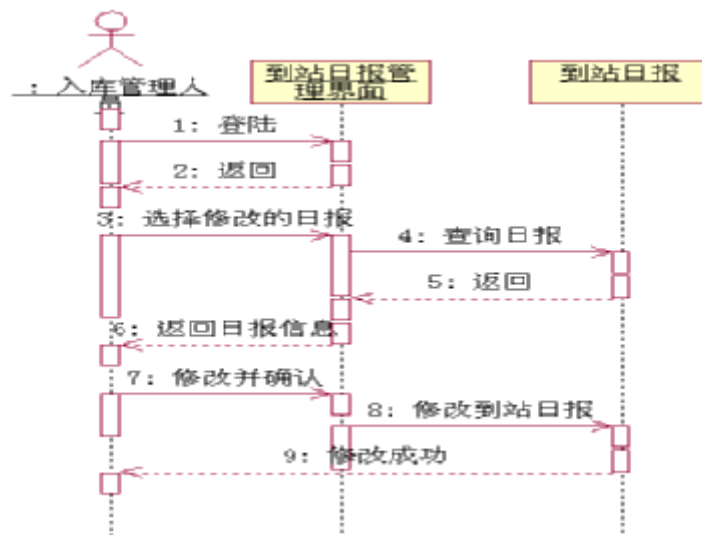
1 新建到站日报：



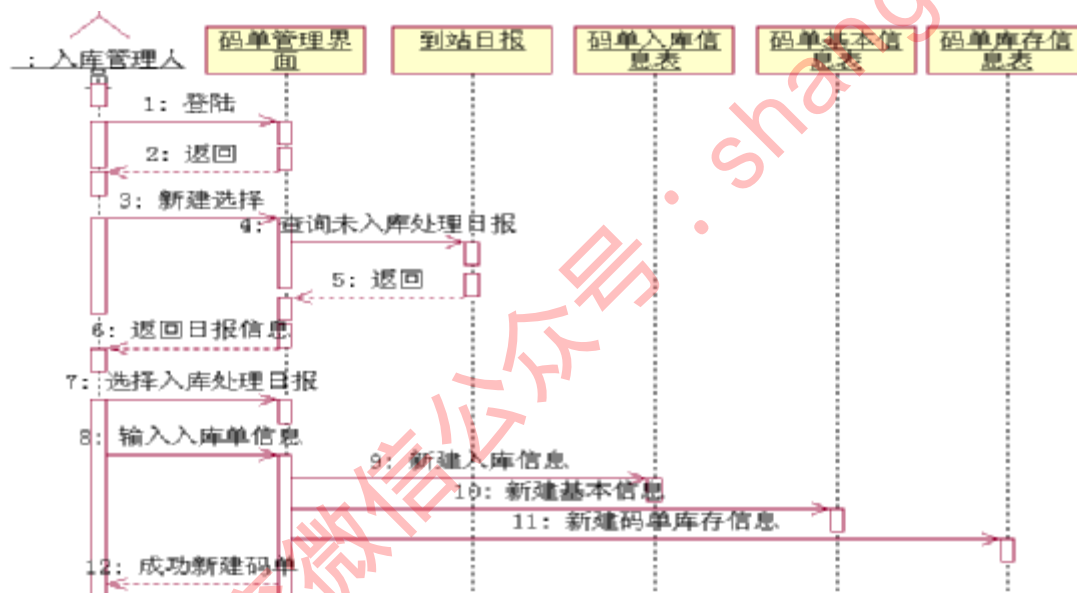
2 修改到站日报：



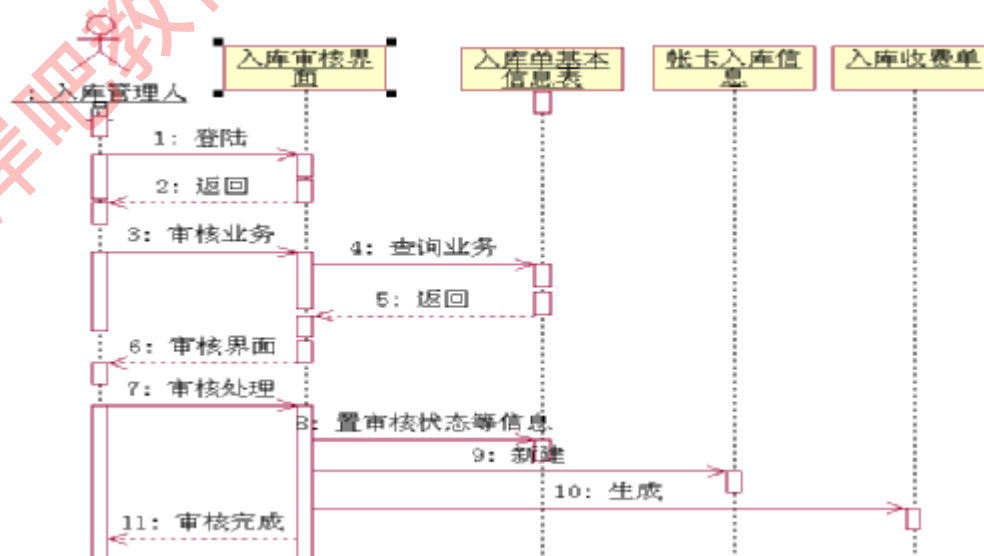
基于 UML 的仓储管理系统的设计与实现



3 新建码单:



4 审核处理





4.2 定义基本对象与类

入库管理子系统的对象分析：

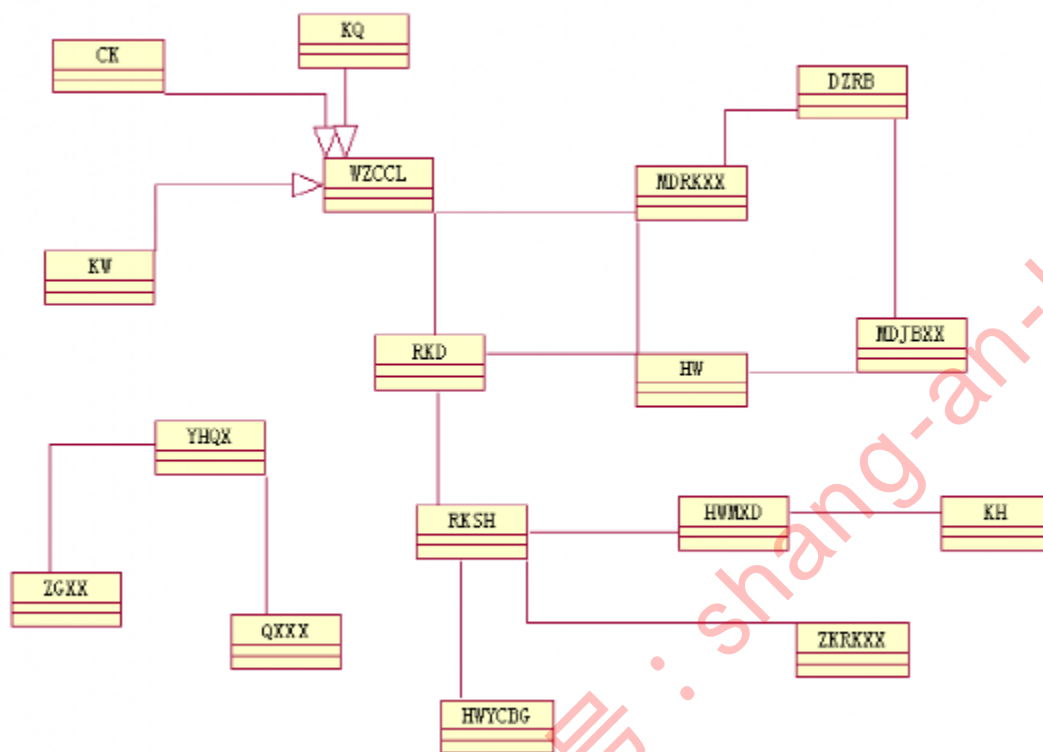
实体对象	货物，物资到站日报，码单入库信息表，仓库，货物明细单，入库单，货物异常报告，帐卡入库信息，入库收费单，客户信息，库区，库位，码单基本信息表，职工信息表，用户权限表，权限信息表，验收工具表，设备表，站台表，计量单位表，部门表，物资明晰分类表，业务类别表
边界对象	入库到站日报管理界面，入库码单管理界面，入库单管理界面，入库审核界面
控制对象	入库审核

根据分析级的顺序图与系统的对象分析定义系统中涉及的类：

- (1) 类 客户 KH
- (2) 类 货物 HW
- (3) 类 仓库 CK
- (4) 类 库区 KQ
- (5) 类 库位 KW
- (6) 类 物资到站日报 DZRB
- (7) 类 码单基本信息 MDJBXX
- (8) 类 码单入库信息：MDRKXX
- (9) 类 保管员入库验收信息 (BGYYSSXX)
- (10) 类 码单货物存放明细 (MDHWCF)
- (11) 类 入库单 RKD
- (12) 类 货物异常报告 HWYCBG
- (13) 类 帐卡入库信息 ZKRKXX
- (14) 类 入库收费单 RKSFD
- (15) 类 用户权限 YHQX
- (16) 类 权限信息 QXXX
- (17) 类 入库审核 RKSH
- (18) 类 职工信息 ZGXX
- (19) 类 物资存储类 WZCL
- (20) 类 业务类别 YWLB
- (21) 类 物资明晰分类 WZMXFL
- (22) 类 部门 BM
- (23) 类 计量单位 JLDW
- (24) 类 站点 ZD
- (25) 类 验收工具 YSGJ
- (26) 类 设备 SB

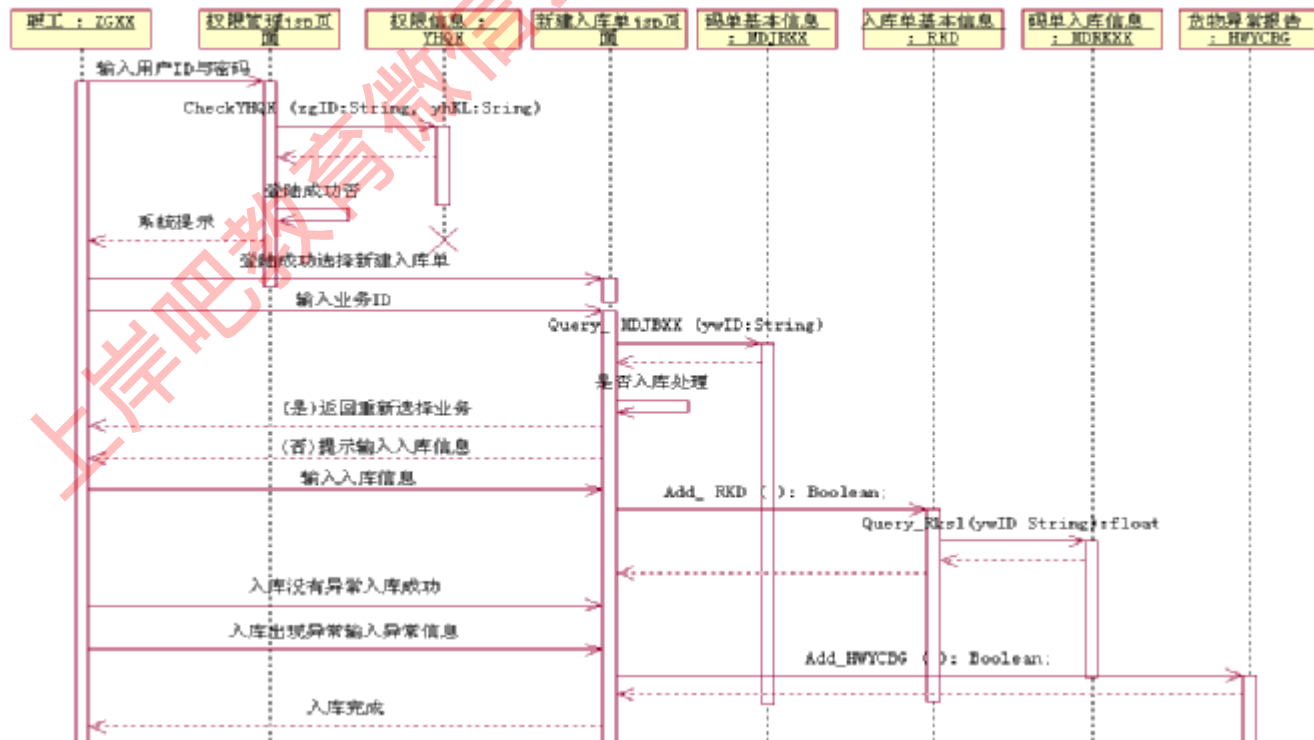


4.3 入库系统类图：



4.4 系统设计顺序图，入库类图

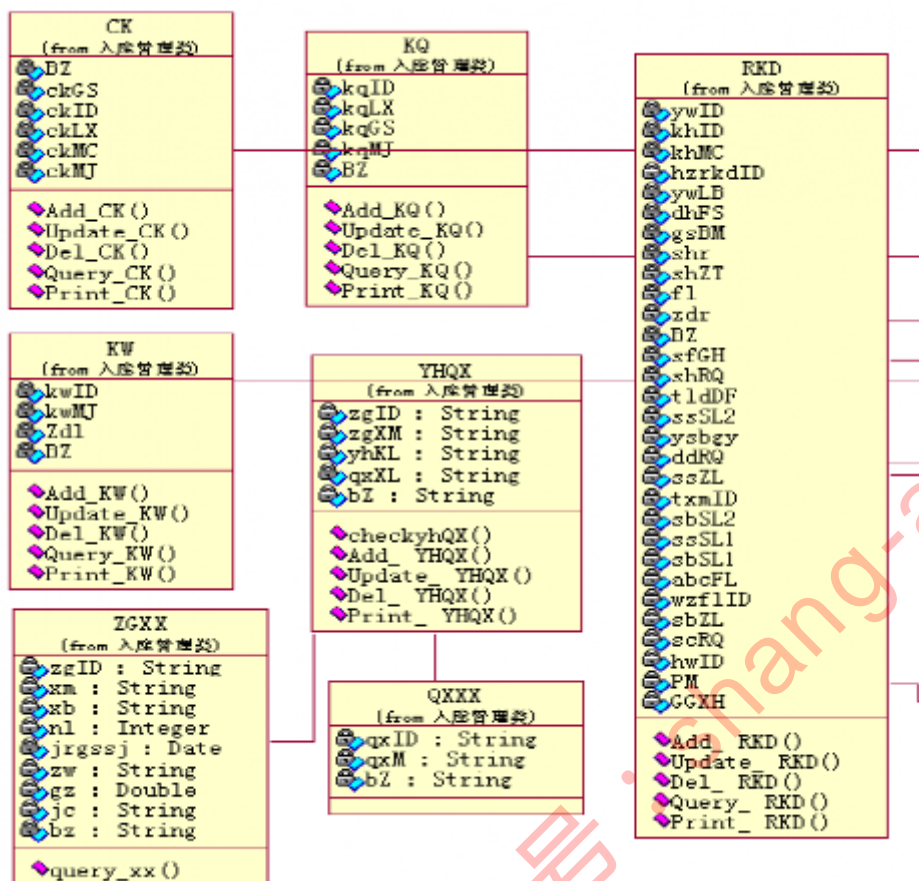
新建入库单顺序图如下：



入库类：



基于 UML 的仓储管理系统的设计与实现

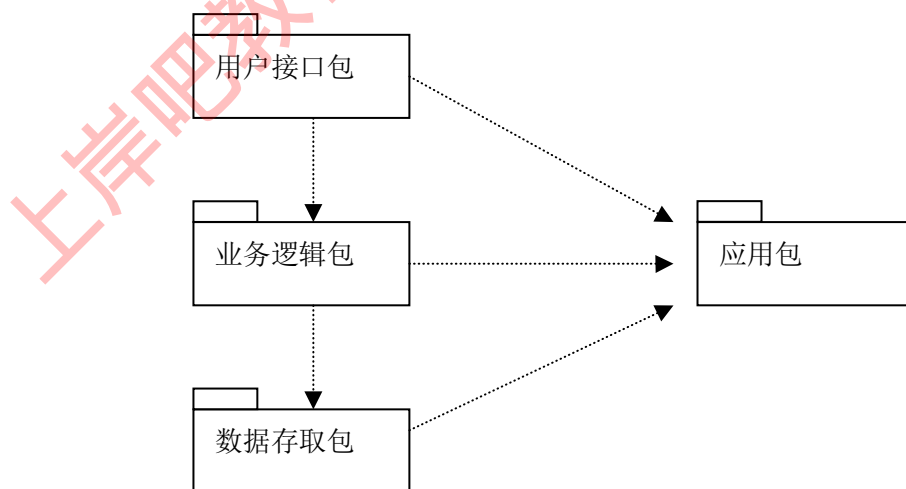


4.5 系统构架设计

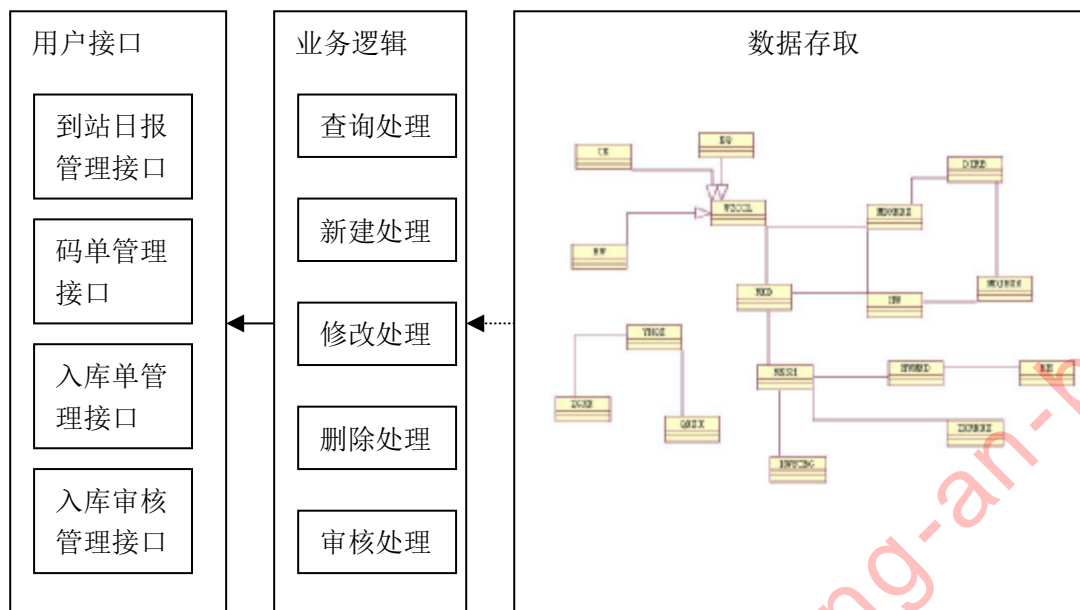
包是类或者文件的组合，在入库系统的开发中，我们定义了用户接口包，入库管理类包，数据存储包，以及在开发中涉及到的 java 包。

用户接口：通过用户接口类，用户可以浏览系统中的数据、输入新的数据。其操作是通过调用业务逻辑中的操作来完成的。

数据存取：提供服务给业务逻辑层中的类，以及类的定义。



入库子系统的设计架构：



第五章 系统实现测试与配置

5.1 系统实现的工具与技术

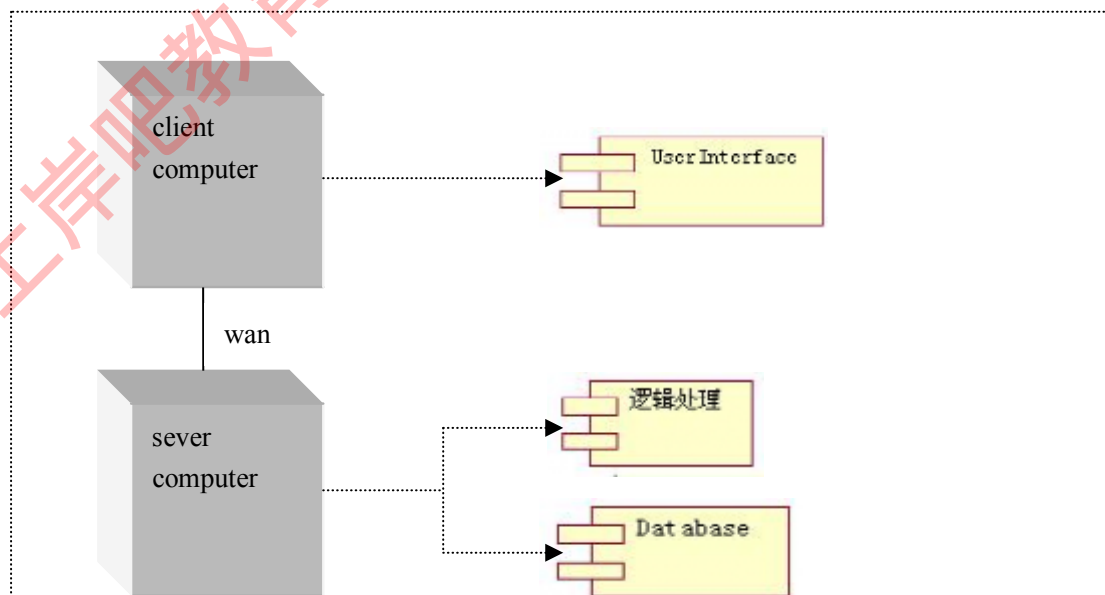
本次系统地开发工具有 Jbuilder9, Rose, Sql Sever, 以及 Tomcat;

Tomcat 是一个开放源代码、运行 servlet 和 JSP Web 应用程序的基于 Java 的 Web 应用软件容器。

系统开发在 Jbuilder9 中开发 Java Servlet,jsp 页面, 以 Tomcat 作为应用程序服务器, SQL Server 作为数据库服务器。

5.2 系统实现方式图

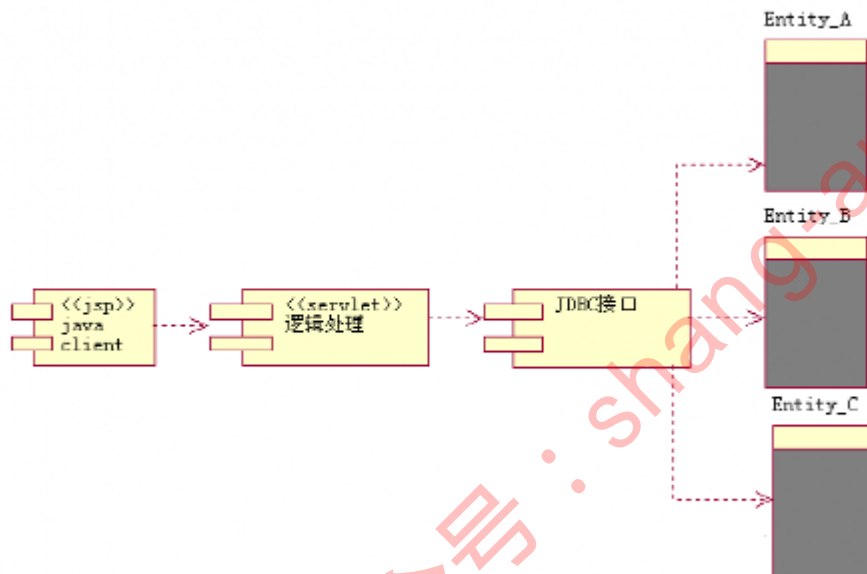
系统配置图:





系统组件图：

UML 是用组件来表示代码物理模块的。组件可以包括代码库和运行文件。在生成代码之前，将每个文件映射相应组件。在本系统中，使用的是 Java 开发。每个类映射一个组件，表示这个类的 .java 文件。生成代码时，Rose 用组件信息创建相应的代码库文件。本系统构造的组件图框架如下：



第六章 系统开发的思考

6.1 数据库设计问题

现在的开发环境越来越多的是面向对象的，而存储机制却是不同于此的关系型数据库，这两者之间存在着很大差异。这种差异使系统的开发活动不能统一。典型的情况是，越来越多的应用系统是三层甚至多层体系结构，在此情况下，用户接口层和业务逻辑层是用面向对象技术开发的，而数据库多数仍然是关系型的。

因此，在采用面向对象建模技术分析获得对象模型后，怎样得到关系型的数据库呢？这也是我们本次系统开发中遇到的问题。

(1) 属性类型映射成域

UML 中的属性类型 (Attribute Type) 映射成数据库中的域 (Domain)。域的使用提高了设计的一致性，且优化了应用的移植性。简单的域是非常容易实现的，仅仅需要替换相对应的数据类型和数据的尺寸。同时，对于使用域的属性，可能要求为域的约束加入 SQL 的 Check 串。例如，限定域的取值范围等。

枚举域 (Enumeration Domain) 限制了域允许取值的集合。其实现通常有几种方法：定义



SQL 约束来限定取值;为每个枚举值定义标志;枚举表;对枚举值进行编码等。

在本系统地开发中,所有涉及到枚举型的数据象,我们都单独设置一表来表达,例如计量单位表中就是入库单等表中计量单位数据项的一枚举约束。

(2)类的属性映射至关系数据库表中的列

属性可以直接映射为表中的零到多列。通常,一个属性映射为表中的一列,但也有例外:

①对于非持久的属性可以不进行映射,有些属性置只做为中间值用于计算而不需保存在数据库中。

②某些对象属性本身就是对象,客户中的地址属性(如果较复杂)可以映射为数据库表或多列。此时,属性映射成多个字段。反之,也可以将多个相似的简单属性映射为一列。

(3)类映射成表

类到表的映射通常不是直接的。只有非常简单的应用,类与表之间才会存在一一对应的关系。

6. 2 数据库访问设计问题

上面的实体对象映射到数据库表,主要是考虑 UML 的静态模型到数据库的映射,我们继续讨论动态模型在数据库设计中的实现,主要是数据库存取部分设计。

一般来说,数据库存取部分设计主要有以下几种策略:

(1)每个对象或类自己负责数据库的访问工作。

这种方法不需要专门的服务类,但是业务逻辑和数据操作逻辑都集中在业务类中,任务划分不清晰,业务类既要处理业务逻辑,又要处理数据操作,其中每一个的改动,都会影响到这个业务类。

(2)利用开发环境提供的机制

许多开发环境提供了初级的访问关系数据库的机制,它们是很好的一个开始。常用的方法包括 Microsoft 的 ODBC 机制(开放数据库连接—ODBC)和 Java 数据库连接(JDBC),绝大多数面向对象的开发环境提供了封装这些标准方法之一的类库。

在本系统地开发中,我们用到了前者。

结束语

本文从面向对象的思想出发,讨论了 UML 建模的语法结构和 UML 的建模思路与过程,并基于此对仓储管理系统进行了分析,设计,以及最后的实现与测试,由于篇幅有限,本文重点介绍了系统的设计部分,对系统的实现与测试只是选择重点分析。

从整个论文的撰写和用例的开发中,我对基于 UML 和统一过程的面向对象建模方法有了相当的认识。在统一过程开发中,把一个较大的项目的功能分配到一个个用例上去实现,让开发过程变得比较轻松,更适合多个人共同开发;同时,通过开发阶段的限制和迭代的划分,使开发过程的透明度更高,目标更明确,整个项目开发的风险也降低了。