

在线拍卖项目要求

目标

这个项目的目的是建立一个 "在线拍卖管理系统",让买家和卖家聚集在一起,几乎可以交易任何东西。

该系统由一个门户网站组成,注册用户可以在该网站上提出新的拍卖建议,出价购买拍卖物品,向其他用户发送信息,并通过电子邮件自动接收新闻、报价等信息。

用户注册前要进行 "预注册":为了检查用户是否输入了真实的电子邮件地址,用户会收到一封带有自动生成的暗码的电子邮件,用户需要在第二时间输入暗码,以确认所输入的数据(姓名、地址、电话号码等)。未经确认,用户无法使用门户网站的功能。

拍卖有名称、描述、用户可能上传的(相关物品)照片和结束时间:拍卖间隔(开始-结束时间)结束时,用户不能出价,但如果没有人出价,可以延长拍卖间隔。

此外,管理员还可以接受或拒绝用户提议的拍卖,查看用户和物品的相关信息,创建、修改和删除拍卖类别(汽车、书籍、音乐物品等拍卖)。

该系统采用三层架构实现:关系数据库,用于存储有关物品、用户、拍卖和拍卖类别的信息;应用服务器,负责系统的业务逻辑;表现层,包括用户可以与系统交互的网络浏览器。

在这种架构下,数据库永远不会被直接访问:例如,管理员可以更改数据库中存储的数据,而无需直接连接数据库,只需使用自己的浏览器即可。

技术现状

在项目开始之前,我们考虑了一些已有的在线拍卖系统及其功能。然后,我们决定采用何种系统结构和软件技术。本项目成功的关键因素是采用最先进的技术,既具有创新性,又能提供用户友好的解决方案。

现有在线拍卖系统

项目的第一部分是对网络上已有的在线拍卖系统进行调查。我们考虑了三个最著名的拍卖网站:eBay.com、steinrete.com 和 onsale.com。

下表介绍了三大拍卖系统为用户提供的功能:

用户故事	eBay.com	asteinrete.com	onsale.com
主页	Х	Х	Х
注册	Х	Х	Х
登录	Х	Х	Х
个人页面	Х	Х	Х
搜索	Х	Х	Х
• 排除单词	Х		
• 在某一类别中	Х	Х	Х
• 在某个城市	Х		
• 价格为欧元至欧元	Х		Х
浏览	Х	Х	Х
项目 页次	Х	Х	х
投标	Х	Х	Х
发布拍卖	Х	Х	Х
帮助	Х	Х	Х
更改语言			
聊天	Х		
行政部门	?	?	?

如表所示,所有这三个系统都提供了注册和登录网站的可能性,并且都有一个主页,对门户网站进行总体介绍。它们还提供了一个个人页面,每个用户都可以在这里查看自己的拍卖情况或报价情况。这些门户网站的另一个特点是包含一个物品页面,该页面描述了拍卖的每一件物品(文字描述、照片等)。

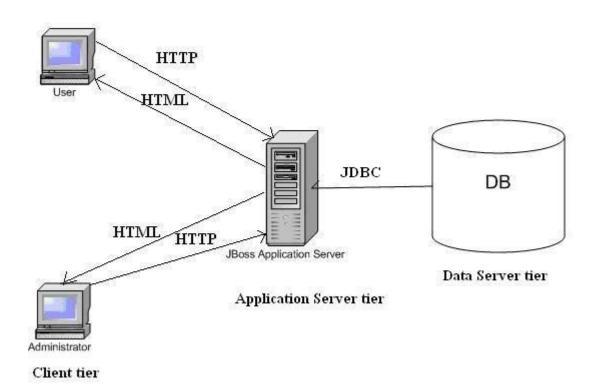
搜索功能也非常重要:除了普通的关键字搜索外,eBay 还提供排除特定词搜索、特定类别搜索、特定城市拍卖搜索以及价格范围搜索等功能。

所有这三个系统都可以出价、发布拍卖信息,并有一些帮助页面解释门户网站的目的和功能。 其他门户网站都不能更改语言,只有 eBay.com 为用户提供聊天室。

三层架构

为了实现在线拍卖系统,我们采用了3层系统结构,如图所示。

3-tier architecture



在这种架构中,有3个主要元素:

- 客户层,负责展示数据、接收用户元素和控制用户界面。
- 应用服务器层,负责系统的业务逻辑。事实上,实现业务规则的业务对象就 "生活 "在这里,并可供客户层使用。这一层保护数据不被客户直接访问。在该项目中,我们使用 JBoss 作为应用服务器。
- 数据服务器层,负责数据存储。我们使用开源关系数据库 PostgreSQL 作为数据服务器。

JBoss 和 EJB

JBoss 是一款免费、开源的应用服务器,实现了完整的 Java 2 企业版(J2EE)堆栈,包括 Java 服务器页面(JSP)、servlets 和企业 JavaBeans(EJB)。

为了实现该项目,我们使用了 EJB 技术,该技术为构建分布式、基于组件的企业级 J2EE 应用程序 提供了一种架构。 EJB 体系结构可处理较低级服务的复杂性,如组件生命周期、状态管理、持久性、多线程、连接池、事务管理和安全性。

EJB 组件被称为企业 Bean,具有可扩展性、事务性和安全性。EJB 由四部分组成:

- 一个主接口,提供控制 EJB 生命周期操作的方法。
- 组件接口,用于定义向 EJB 客户端公开的业务方法。
- bean 实现,包含执行业务逻辑的方法。
- 部署描述符指定了描述 EJB 及其所需服务的声明性语义。

此外,EJB 有两种类型:一种是会话 bean,用于表示对持久数据的操作,但不表示数据;另一种是实体 bean,用于表示持久数据(通常是数据库中的记录)。

实体 bean 的持久性有两种管理方式:

- CMP(容器管理持久化),其中容器使用部署描述符中的信息自动管理 EJB 到存储系统的映射,从而减少 bean 中的代码量。
- BMP(Bean 管理持久性),Bean 直接管理映射,通常使用一些 JDBC 代码。

为了实现这个项目,我们采用了 CMP 管理技术。

数据库和 JDBC

我们计划使用 PostgreSQL 作为项目的数据库,这是一个开源关系数据库。关系数据库是一种数据库管理系统(DBMS),以相关表格的形式存储数据。关系数据库功能强大,因为它们几乎不需要假设数据是如何关联的,也不需要假设如何从数据库中提取数据。因此,可以用多种不同的方式查看同一个数据库。

选择 DBMS 时最常见的选项是 Oracle、MySQL 和 PostgreSQL。下表总结了这些数据库系统的优缺点:

	甲骨文	MySQL	PostreSQL
优点	- 最快的商业	- 开放源代码	- 开放源代码
	数据库管理系统	低机器	- 易于

	-作家从不阻挡读者 - 交易、 回滚和子选择 支持。	要求 易于设置	管理 交易和 支持回滚
缺点	- 非开源(甲骨文许可证价	- 无事务、回滚和子选择。	- 不支持容错安装。
	格昂贵)		

这三个系统的主要区别在于,MySQL和 PostgreSQL是开源的,而 Oracle 不是。与其他两个系统相比,Oracle 提供了更先进的功能:速度最快,支持事务(被视为单一操作的基本操作集),并具有企业级数据保护和分发功能,如大规模集群复制。另一方面,在所有可用的开源数据库解决方案中,PostgreSQL是最完善的,它拥有事务支持、ANSI SQL92 合规性、查询优化器和基本的数据完整性控制等。作为在线拍卖门户网站的数据库系统,我们选择了 PostreSQL:它不支持容错安装,但在我们的环境和需求中,使用它没有任何缺点。

JDBC应用程序接口是Java编程语言与各种数据库之间独立于数据库的连接的行业标准,JBoss可以使用JDBC应用程序接口与数据服务器层 "通信"。事实上,JDBC API 为基于 SQL 的数据库访问提供了一个调用级 API,可以实现以下三点:

- 与数据库建立连接或访问任何表格数据源。
- 发送 SQL 语句
- 处理结果。

JSP

为了实现在线拍卖门户网站的网页,我们使用了 JSP(Java 服务器页面)技术,JBoss 应用服务器也采用了该技术。JSP 允许开发人员使用识别 Java 代码的特殊标记,在同一文档中结合 HTML 和 Java 代码。

JSP 对 Java 语言没有任何限制,其开发也非常简单。事实上,JSP 页面只是带有特殊标记的 HTML 页面,这些标记包括执行详细说明和提供 HTML 代码的 Java 代码。

JSP 的首次执行包括三个步骤:

- 将 JSP 代码转换为 JavaServlet 编译 Servlet
- 执行 Servlet。

接下来的执行只包括最后一步。此外,还可以对所有 JSP 进行预编译,以减少第一次执行所需的时间。

Eclipse 和 MyEclipse

对于项目的编程部分,我们计划使用 Eclipse,这是一个非常强大的开源集成开发环境(IDE)。该集成开发环境可为开发人员提供多种服务:它具有编译器感知编辑功能;语法错误和简单的语义错误(如缺少声明)会在出现时被高亮显示。Eclipse 支持方法补全,以图形符号简洁地显示类接口,并通过飞越名称解析等功能支持程序的交互式探索。此外,Eclipse 还支持打包、调试、测试、重构和版本管理等软件工程原则。使用 Eclipse 的另一个重要优势是可以在其上安装多个插件,如MyEclipse,这在使用 Enterprise JavaBeans 时尤其有用。

事实上,MyEclipse 为 Java 和 J2EE(Java 2 企业版)开发人员提供了经济实惠的工具。其功能包括

- 网络开发工具,如 JSP、HTML、XML 和 CSS 带有代码自动补全和语法着色功能的编辑器。它们还包括 JSP 语法验证和本地 JSP 调试。
- 生产率向导,如 EJB 向导、基于归档的部署(EAR 和 WAR)以及按需同步将应用程序自动部署到集成服务器。
- 应用服务器集成,包括启动和停止应用服务器的集成控制,以及对已部署应用的全面热插拔调试支持。

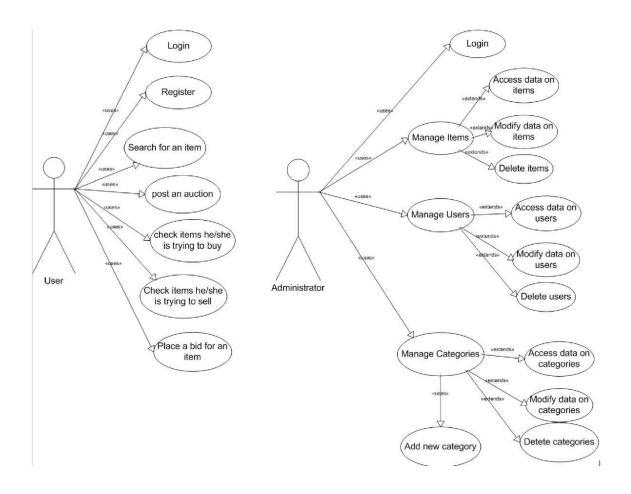
功能要求和系统设计

业务案例推动了在线拍卖门户网站的功能需求。在此过程中,我们利用了 "使用案例 "和 "用户故事 "。功能需求一旦达成一致,我们就利用 UML 类图、数据库设计的 EER 图和页面流程图对其进行增强和验证。

使用案例

功能需求收集的第一步是用例。用例是 "对系统执行的一系列动作序列(包括变体)的描述,这些动作序列会产生对行为者有价值的可观察结果"。使用用例的目的是: 从用户角度设计系统,用用户术语交流系统行为,并列举所有外部可见的行为。

以下是在线拍卖系统项目的用例(系统有两个角色: 普通用户和管理员)。



如图所示,普通用户可以注册系统、浏览可用物品、出价并发布新的拍卖信息。另一方面,管理

员可以插入和修改有关物品、用户和物品类别的可用数据。

用户故事

收集完所有用例后,就可以编写用户故事了。用户故事 "是允许客户定义(和引导)系统路径所必需的最小信息量(一个步骤)"。

用户故事分为两大类: 用户方面(针对普通用户的故事)和管理方面(针对系统管理员的故事)。

用户侧故事:

主页:

主页是访问主要服务的入口:

- 注册
- 我的个人主页
- 搜索
- 帮助

主页还显示分类列表,以简化物品搜索和最新拍卖。

注册:

注册页面允许用户提供其个人资料(姓名、地址、电子邮件地址、电话号码、用户 ID、密码)并获得一个用户 ID 和一个密码。用户 ID 和密码允许用户访问其个人页面、参加拍卖和发布新拍卖。

系统对输入的数据进行基本检查,如果用户 ID 和/或用户 ABN 已存在于系统中,则提供用户注册或错误信息。

登录

每次用户尝试访问非公开区域(个人页面、投标、发布拍卖信息......)时,都会被要求提供个人 ID 和密码。这些信息通过表格输入。如果用户名和密码正确,用户就登录了,在整个会话期间不 会再被要求登录。

否则会出现错误信息。

个人页面:

要访问个人页面,用户需要登录或注册。个人页面记录了用户目前试图购买和近期购买的所有物品,以及用户试图出售的所有物品。在此页面还可以发布新的拍卖信息。

浏览:

用户可以从多个类别的物品(如汽车、书籍等)中进行选择,浏览拍卖结果。结果将以表格形式 显示,用户可按价格、拍卖间隔(拍卖持续时间)进行排序。

搜索:

用户可提供关键词,按不同标准搜索拍卖物品:

- 排除单词
- 在某一类别中
- 在某个城市

• 价格从给定欧元值到另一数值的拍卖

注册用户和未注册用户均可使用这项服务。

项目页面:

项目页面显示项目特征。用户可以在该页面按下 "PLACE A BID(出价)"按钮出价,并查看出价的时间顺序。

投标:

出价的用户如果尚未登录,会被要求登录。如果系统接受出价,物品就会出现在用户的个人页面上。出价只能在拍卖时间段内进行,出价必须比当前价格至少高出一个最低增价。

发布拍卖信息:

用户可在其个人页面上通过特定表格发布拍卖信息,提供他/她愿意出售的物品的特征。如果拍 卖被系统接受,该物品就会出现在用户的个人页面上,其他用户可以出价购买。

帮助:

系统必须提供帮助页面,说明如何执行所有可能的操作,如注册、搜索物品、发布拍卖信息等。

更改语言:

用户可以在每个页面上通过组合框更改阅读页面的语言。

聊天

用户可以用自己的用户 ID 作为昵称,用自己的个人密码作为密码进入聊天室。在聊天室中,用户可以向所有用户发送信息,也可以只向一个用户发送私人信息。

留言

用户可以用文字和主题向其他用户发送信息,信息将通过电子邮件发送给接收者。用户还可以在 网站上查看收到的信息并进行回复。

行政长官身边的故事

管理员页面:

在登录页面,提供管理员 ID 和密码后,他/她可以访问管理员页面,该页面显示管理员菜单,可访问所有管理活动(管理项目、管理用户、管理类别)。

管理项目:

管理员可以访问数据库中存储的所有物品数据,也可以删除这些数据,但不能修改物品的特征(初始价格、描述等)。

管理用户:

管理员可以访问和修改数据库中存储的所有用户数据,也可以删除这些数据。

管理类别:

管理员可以访问和修改数据库中存储的所有类别数据,添加新类别,也可以删除类别。管理员可以删除一个类别,前提是该类别没有相关联的项目,否则会出现错误信息。

接受/拒绝拍卖:

管理员可以控制用户发布的新拍卖,并决定接受或拒绝这些拍卖。

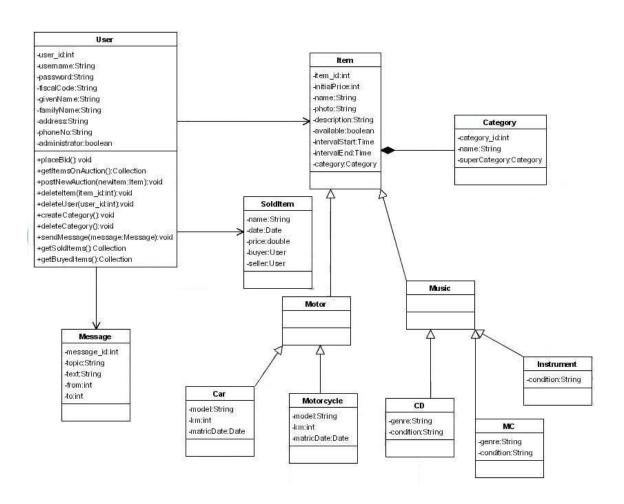
然后,我们编写一个表格,为每个用户故事分配优先级和风险级别:

用户故事	优先权	风险
用户侧故事		
主页	必须	低
注册	必须	中型
登录	必须	中型
个人页面	必须	高
搜索	必须	中型
排除单词	应该	中型
在某一类别中	必须	中型
价格从欧元到欧元不等	应该	中型

在某个城市	应该	中型
浏览	必须	中型
项目页面	必须	中型
投标	必须	高
发布拍卖	必须	中型
帮助	应该	低
更改语言	能否	中型
聊天(秒杀)	能否	高
信息	应该	中型
行政方面的故事		
管理员页面	必须	低
管理项目	必须	中型
管理用户	必须	中型
管理类别	必须	中型
接受/拒绝拍卖	必须	中型

UML 类图

设计阶段的下一步是绘制系统的 UML 类图。由于系统的编程语言是面向对象的语言,因此 UML 类图特别适用于显示系统的类、它们之间的相互关系以及类的操作和属性。下面是项目的类图。



User 类包含几个参数,这些参数是浏览系统的普通用户的信息(用户名、密码、姓名……); administrator 参数是一个标志,表示该用户是否是管理员。该类执行普通用户的操作(搜索物品、获取拍卖物品、发布新拍卖、向其他用户发送消息)和系统管理员的操作(创建新类别、删除类别、删除物品、删除用户)。消息类包含一个文本、一个主题以及发送者和接收者的用户名。物品 "类包含拍卖物品的若干信息(物品名称、照片、文字描述……)和所属类别的信息。

一个类别有一个名称,最后还有它所属的超类别:一个类别可以属于另一个类别(例如,运动型汽车类别属于汽车类别)。

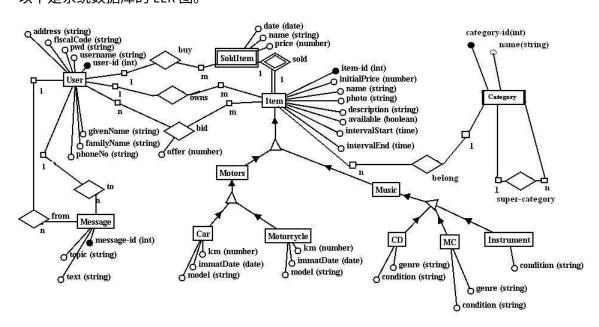
类 Motor 和 Music 是 Item 的子类。汽车和摩托车是 "汽车 "的子类:它们继承了类 "项 "的属性,并有自己的特定属性(型号、公里数、注册日期)。CD、MC 和 Instrument 也继承了 Item 的属性,并有自己的特定属性(流派、状态)。

SoldItem 类代表已售出的拍卖物品,其属性包括物品名称、最终价格、售出日期以及买家和卖家

数据库设计的 EER 图

在绘制了在线拍卖系统的 UML 类图之后,数据库中应存储哪些数据就很清楚了。由于 PostgreSQL 是关系数据库,EER 建模方法对设计数据库模式非常有用,因为它能很好地映射到关系模型,而且 ER 模型中使用的构造可以很容易地转换成关系表。

以下是系统数据库的 EER 图。



如模式所示,有多个实体(用户、项目、消息……),它们有各自的属性和关系。汽车和音乐是一种特殊的物品(它们扩展了物品实体)。每个用户都拥有一个或多个项目,并可以为其中一个或多个项目出价。另一方面,每个物品由一个用户拥有,属于一个类别。一个类别可以有一个超级类别。一条信息从一个用户发送给另一个用户。

汽车、摩托车和 CD、MC、乐器分别是汽车和音乐的子实体。除了从 item 继承的属性外,它们还有各自的特定属性(公里数、入学日期、流派……)。

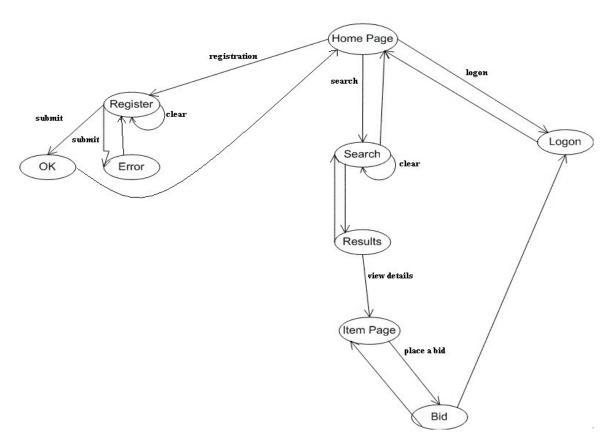
出售项目是一种特殊的实体,被称为 "弱实体"。弱实体与普通实体不同,自身没有关键属性。 当一个物品被出售时,它就变成了一个已售物品。

从 EER 图中,使用 "Elmasri "算法很容易得到数据库的表,以下是在线拍卖系统数据库的表。按照 EMJ 惯例,每个实体都有一个整数 ID 作为主键。

USER(user_id(PK), username, pwd, administrator, givenName, familyName, phoneNo, address, civicNo, city, state, ABNCode)
MESSAGE(message_id(PK), from(FK USER), to(FK USER), topic, text)
ITEM(item_id(PK), initialPrice, name, photo, description, available, intervalStart, intervalEnd, owner(FK USER))BID(bid_id(PK), item(FK ITEM), offerer(FK USER), offer)
MOTORS(motors_id(PK, FK ITEM))
CAR(car_id(PK, FK MOTORS),公里数, immatDate, 型号)
MOTORCYCLE(motorcycle_id(PK, FK MOTORS),公里数, immatDate, 型号)
CD(cd_id(PK, FK MUSIC),类型, 条件)
MC(mc_id(PK, FK MUSIC),类型, 条件) INSTRUMENT(instrument_id(PK, FK MUSIC),条件) SOLDITEM(item_id(PK), 名称, 日期, 价格, 买方(FK USER))CATEGORY(category_id(PK, FK ITEM), 名称, super-category(FK CATEGORY))

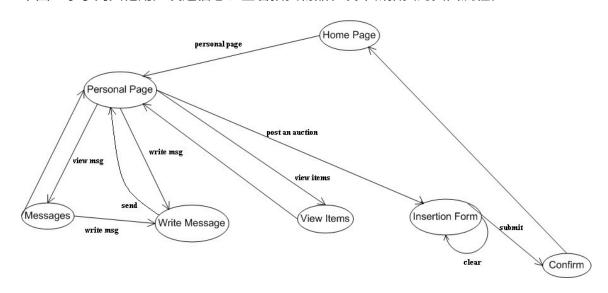
页面流程图

由于在线拍卖门户网站由网页组成,因此绘制页面流程图非常有用。页面流程图是一种直观组织网页流程和操作的图表。下面的页面流程图代表了普通用户注册系统、搜索物品、出价和登录的路径。



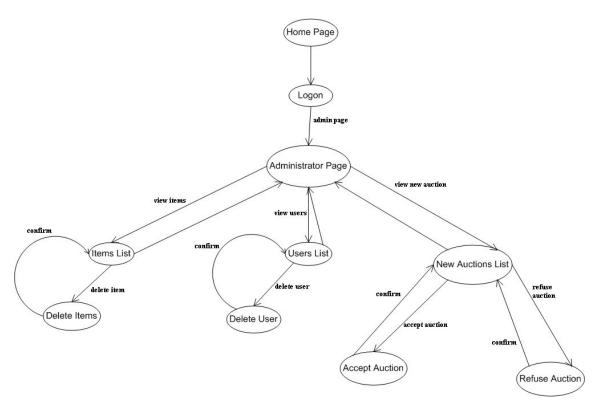
如图所示,从主页可以进入注册页面,输入用户的个人数据。如果数据正确,用户可以进入确认 页面,否则(例如,用户选择的用户名已被使用)将进入错误页面。用户也可以在主页上输入用 户名和密码登录,然后进入搜索页面进行搜索。搜索结果将显示物品链接,以便查看物品详情(描述、照片……)并为该物品出价。

下图显示了向其他用户发送信息、查看拍卖物品和发布新拍卖的页面流程。



登录用户可从主页进入个人页面。在个人页面上,用户可以给其他用户留言,阅读收到的留言,查看自己的拍卖物品,并通过插入表格发布新的拍卖信息。

最后是管理活动的页面流程图。



如果用户是系统管理员,在输入用户名和密码后,就可以进入管理页面。在管理页面,可以查看数据库中关于用户和物品的所有数据,并对其进行管理。管理员还可以查看用户提出的新拍卖清单,并接受或拒绝这些拍卖。

系统实施

实施工具

XDoclet

XDoclet 是一个开源代码生成引擎,可实现面向属性的 Java 编程。这意味着,通过在 java 源代码中插入元数据(属性),可以为代码添加更多意义。这可以通过特殊的 JavaDoc 标签来实现。
XDoclet 可解析源文件并生成许多人工制品,如 XML 描述符和源代码。这些文件由使用源代码及其 JavaDoc 标记中提供的信息的 模板生成。

下面的示例说明了我们如何使用 XDoclet 工具创建 "类别 "实体 Bean。

软件包it.nrkauction.ejb.PI

```
import javax.ejb.CreateException; import javax.ejb.EJBException; import javax.ejb.EntityBean; import javax.ejb.EntityContext;
```

import javax.ejb.RemoveException;

```
/**
                          name = "类别"
* @ejb.bean
                          类型= "CMP"
                          cmp-version = "2.x"
                          display-name = "类别"
                          description = "类别EJB"
                          视图类型 = "本地"
                          jndi-name = "ejb/CategoryHome"
                          local-jndi-name = "ejb/CategoryLocalHome"
                          TBL CATEGORY
* @jboss:table-name
                          签名 = "Collection findAll()"
* @ejb:finder
                          未选中 = "true"
                          query = "SELECT DISTINCT OBJECT(category) FROM
类别"
                                  结果类型映射 = "本地"
                          生成 = "true"
                          生成="物理"
* @ejb:util
*/
```

公共抽象类 Category 实现 EntityBean {

在类声明之前,有一个特殊的 JavaDoc 文件,其中包含一些 XDoclet 属性。第一个属性(@ejb.bean)描述了实体的名称、类型(可以是 CMP 或 CMR)以及相关主页接口的名称(CategoryLocalHome)。第二个属性(@jboss:table-name)表示

与该实体相关的数据库表的名称(事实上,每个实体 bean 都与一个数据库表相关)。

属性 @ejb:finder 代表一种查询,它是用 EJBQL(一种与 SQL 非常相似的语言)编写的。在本例中,它告诉 XDoclet 引擎创建一个名为 "findAll() "的方法,返回一个集合类型的对象,其中包含表中所有类别的完整列表。

启动 XDoclet 代码生成引擎后,它会自动创建三个人工制品:

1. 带有 findAll() 方法的主界面:

```
public interface CategoryLocalHome extends javax.ejb.EJBLocalHome { public static final String COMP_NAME="java:comp/env/ejb/CategoryLocal"; public static final String JNDI NAME="ejb/CategoryLocalHome";
```

public java.util.Collection findAll() throws javax.ejb.FinderException;

2. 业务方法的组件接口:

}

```
public interface CategoryLocal extends javax.ejb.EJBLocalObject{

public int getCategory_ID();

public java.lang.String getName();

public void setName(java.lang.String value);

public it.nrkauction.interfaces.CategoryLocal getSupercategory();

public void setSupercategory(it.nrkauction.interfaces.CategoryLocal owner);
}
```

3. XML 部署描述符(参见附录--部署描述符)。

使用 XDoclet 有几个好处:

• 只要开发人员接触代码,就不必担心部署元数据,因为这些元数据会被持续集成进来

- EJB 通常由更多的文件组成,因此很容易丢失。有了 XDoclet,开发人员只需维护其中一个文件,其他文件都会自动生成。
- 开发人员可以大大缩短开发时间,在 XDoclet 生成其余代码的同时,专注于业务逻辑。

系统手册

在此,我们将简要介绍一下该系统。在线拍卖门户网站的首页如下所示:



主页包含有关门户网站如何运作的若干信息以及通往其他页面的若干链接。页面左侧有一个菜单,输入用户名和密码即可登录,非注册用户也可在此注册拍卖门户网站。页面右侧是可用拍卖类别列表。用户可以通过点击其中一个类别浏览属于该类别的各种物品。主页上部有一个导航菜单,用户可以进入搜索页面(搜索物品)、个人页面、新拍卖方案页面(发布新拍卖方案)和网站帮助页面。只有注册用户才能访问个人页面和新拍卖方案页面。

为了使用门户网站的全部功能,用户必须注册,并在注册页面上填写个人资料,如下所示:

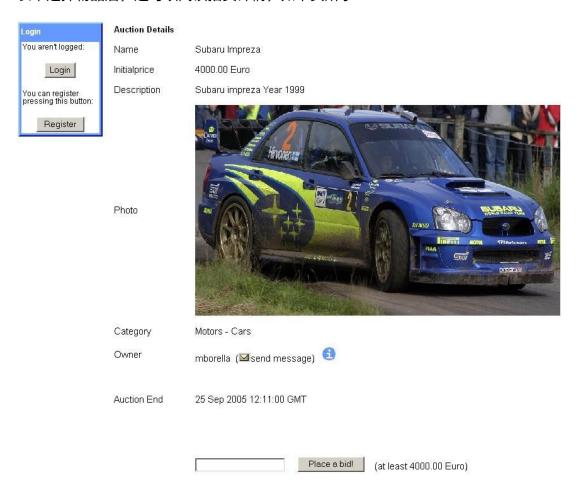


在此表格中,用户必须输入个人数据(名、姓、地址等),并选择用户名和密码。输入数据后,用户会收到一封带有密匙字符串的电子邮件,用户必须输入密匙字符串才能确认注册。这样做是为了验证用户输入的电子邮件地址确实是他/她的电子邮件地址,而不是假的。

要搜索拍卖物品,用户可在搜索页面指定关键字和物品类别,如下图所示:



输入关键词并选择类别后,用户会得到一个结果列表(如果有符合搜索条件的物品)。此外,从列表中选择物品后,还可以阅读拍卖详情,如下页所示:



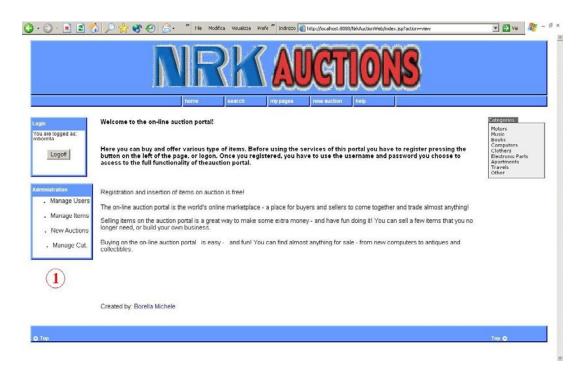
在此页面上有关于物品的信息(描述、价格、照片、拍卖结束日期),注册用户可以向物品所有者发送信息,查看所有者的信息,并最终出价。物品所有者每次出价后都会收到电子邮件通知。注册用户可在个人页面查看其收到的信息和拍卖状态:



注册用户还可以从横向菜单中选择 "新拍卖",提出新的拍卖。选择拍卖类别后,用户可在表格中填写有关其拍卖的数据:



用户可在此输入拍卖信息(名称、描述、初始价格、可能的照片......)并提交其建议书。 管理员登录后,除了主页上的常规菜单外,还会出现一个管理菜单,如下图所示:



页面左侧有一个菜单,可在此访问数据库中存储的有关用户、物品和类别的数据(管理用户、管理物品、管理类别),并删除或修改这些数据。还可以查看用户发布的新拍卖提案(新拍卖)列表,并接受或拒绝这些提案(事实上,拍卖必须得到管理员的接受才能在门户网站上发布)。例如,这里是管理用户的管理页面:



从该页面可以看到系统用户的完整列表。管理员可以通过选择按钮查看用户的所有数据,也可以通过删除按钮从门户中删除用户。还可以通过页面上部的搜索表单搜索用户名。此外,管理员还可以插入新的拍卖类别,或修改或删除其中一个类别,如页面所示:



在此页面可以查看系统中的所有拍卖类别。管理员可通过编辑按钮查看和修改类别数据,或通过删除按钮删除类别。管理员还可以使用页面上部的按钮插入新类别,或使用搜索表单搜索现有类别。