****

**时间交易系统**

分析与设计

项目成员：陈辛夷、李伟彬、黄淼鑫、苏舒婷、许文深

指导老师：衣扬

目 录

[目 录 1](#_Toc500588158)

[1. 时间交易系统概述 3](#_Toc500588159)

[1.1. 项目问题陈述 3](#_Toc500588160)

[1.2. 项目背景 3](#_Toc500588161)

[1.2.1. 双向需求的存在 3](#_Toc500588162)

[1.2.2. 效益差的存在 3](#_Toc500588163)

[1.3. 国内外发展现状及存在问题 4](#_Toc500588164)

[1.3.1. 市场起步 4](#_Toc500588165)

[1.3.2. 现有产品的价值错位 4](#_Toc500588166)

[1.4. 功能 5](#_Toc500588167)

[1.5. 开发环境 6](#_Toc500588168)

[2. 需求分析 7](#_Toc500588169)

[2.1. 时间交易系统用例模型 7](#_Toc500588170)

[2.2. 时间交易系统用例规约 7](#_Toc500588171)

[2.2.1. 用户注册 7](#_Toc500588172)

[2.2.2. 用户登录 9](#_Toc500588173)

[2.2.3. 个人信息管理 11](#_Toc500588174)

[2.2.4. 发布需求 12](#_Toc500588175)

[2.2.5. 接收需求 14](#_Toc500588176)

[2.2.6. 用户管理 16](#_Toc500588177)

[2.2.7. 需求管理 18](#_Toc500588178)

[2.3. 补充规约 19](#_Toc500588179)

[2.3.1. 可靠性 19](#_Toc500588180)

[2.3.2. 安全性 19](#_Toc500588181)

[2.3.3. 性能 20](#_Toc500588182)

[2.3.4. 可支持性 20](#_Toc500588183)

[2.3.5. 可扩展性 20](#_Toc500588184)

[2.4. 术语表 20](#_Toc500588185)

[3. 交易系统的系统设计 21](#_Toc500588186)

[3.1. 时间交易系统框架 21](#_Toc500588187)

[3.2. 系统关键抽象 22](#_Toc500588188)

[4. 用例分析 23](#_Toc500588189)

[4.1. 用例类图分析 23](#_Toc500588190)

[4.1.1. 发布需求用例的类图分析 23](#_Toc500588191)

[4.1.2. 用户管理个人信息的类图分析 24](#_Toc500588192)

[4.2. 用例时序图分析 25](#_Toc500588193)

[4.2.1. 发布需求用例的时序图分析 25](#_Toc500588194)

[4.2.2. 用户管理个人信息的时序图分析 26](#_Toc500588195)

[4.3. 类的分析机制 27](#_Toc500588196)

[4.4. 合并分析类 28](#_Toc500588197)

[5. 附录 30](#_Toc500588198)

# 时间交易系统概述

## 项目问题陈述

本项目开发一个网页服务平台，应用核心为时间交易，所谓时间交易，就是用户可以用金钱来购买服务来节省自己的时间，比如用户通过发布一个帮忙取快递的需求，其他用户可以接单赚钱，而发布者可以节省自己去取快递的时间，这就是时间交易的具体体现。

## 项目背景

随着社会的节奏越来越快，时间已经作为一个稀缺资源越来越为人们所重视，时间资源的不匹配也成为一个重要的问题，以此为起点，“时不时”时间交易平台是我们小组想要结合自身所学，研究、打造的一个时间交易平台，致力于促进时间的合理配置。应用以时间作为突破口，为以实体货物为主的传统电子商务开拓出了一片新的蓝海；交易的同时互助，契合了互帮互助的积极价值观；另外，结合云平台的特点也提供了更好地灵活度和可拓展性。

### 双向需求的存在

生活中总会遇到一些不想去做又不得不去做、必须去做但又没时间去做、想去做但又不想一个人去做的事情，如练球、练琴，不想一个人去图书馆，不想一个人去看电影，不想一个人出去玩没人帮拍照……这样的需求催生出买时间的想法。

而与此同时，由于时间交错性的存在，另一部分人此刻可能有着充足的空闲时间无从消费。抑或是经济困难的人想要通过空闲时间挣取一定的生活费用。这样的交错便产生了时间交易的双向需求。

### 效益差的存在

同一件事情在不同人看来效益可能是不一样，如一个同学可能认为花时间在图书馆看书的效用值10元（当然效益不一定能用金钱来衡量），而去拿快递可能只值3元，如果另一个同学认为他这段时间的效用值5元，而第一个同学愿意用6元的价格购买他的这段时间，这种交易的发生将使每个人的效益都增加。正因为这种效益差的普遍存在，搭建这么一个平台将这种需求和供给联系在一起则很有价值。

## 国内外发展现状及存在问题

### 市场起步

时间买卖的概念在近两年开始出现，以此作为主题的应用更多的处于概念阶段，应用市场广阔。

### 现有产品的价值错位

在应用市场上目前有两款具有一定类似性的应用，“腿儿”以代跑腿为主，时间应用的概念狭隘，同时其基于城市的划分使得用户交易安全性没有保证，市场占有率极低。另一款应用“爱时间”则是以同城社交为主要卖点，将买卖时间侧重在以“约”为主的快餐式荷尔蒙经济上。综上二者均出现了对时间交易理解的价值错位，背离了互助、服务、安全的原则。

## 功能

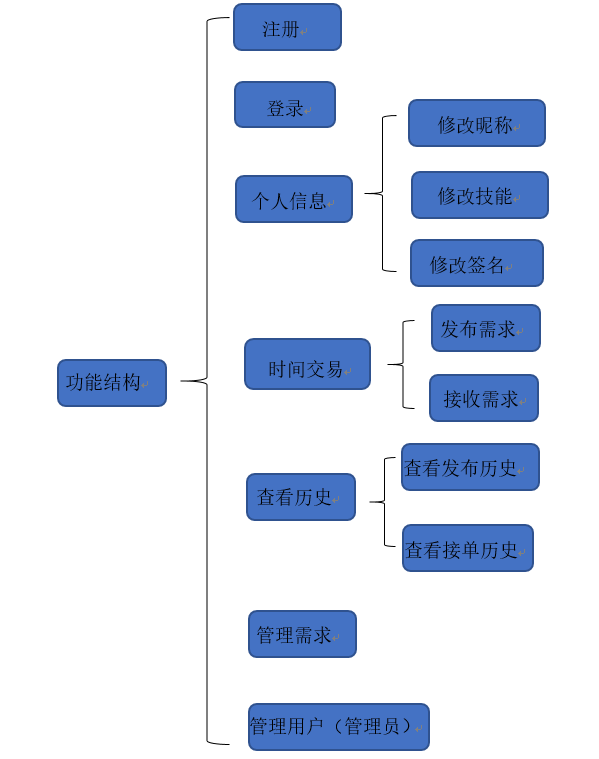


图1-1 功能概述图

* **注册**
* **登录**
* **个人设置**

查看、设置用户技能、用户昵称、用户个性签名等个人信息

* **时间交易**

应用的核心功能，用户在这里进行交易。

1. 发布需求：用户将自己的需求发布到应用上，并注明具体的时间长度和时刻、距离、售卖的金额、和描述关键词（如可辅导功课、陪运动、泡馆等）。
2. 接收需求：用户除了可以发布需求，还可以接收其他用户发布的需求。

* **查看历史**

1. 查看发布历史：用户可以查看自己已发布的所有需求。
2. 查看接单历史：用户可以查看自己已接收的所有需求

* **管理需求**

系统允许删除暂未被接收的需求（已被接收的需求默认已完成交易，不再删除），普通用户可以删除自己发布的需求；管理员可以删除其他用户发布的需求。

* **管理用户**

管理员可以查看并且删除当前系统的用户。

## 开发环境

1. 暂定基于网站的互联网应用
2. 开发语言：Jsp(前端)、Java（后台）
3. JDK版本：JDK1.6
4. 开发工具： Eclipse
5. 本地数据持久化：SQLIte关系型数据库。

# 需求分析

## 时间交易系统用例模型

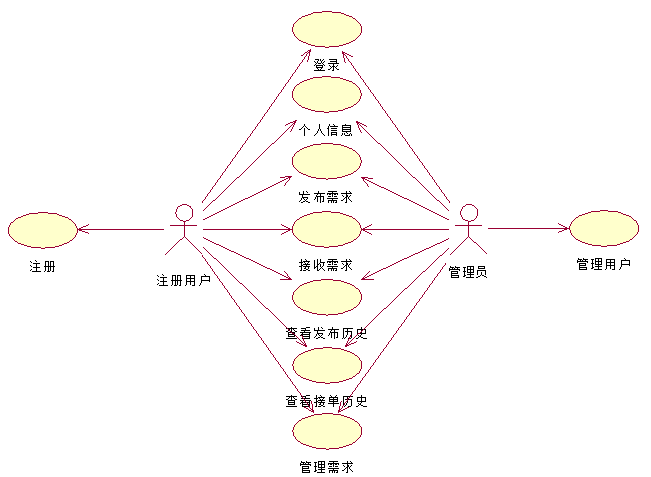


图2-1 时间交易系统用例图

## 时间交易系统用例规约

### 用户注册

1. 简要说明

本用例描述注册用户如何在时间交易系统中注册。

用户名合法性定义：用户名只能包含字母和数字,并且长度小于50。

密码合法性定义：密码长度小于100。

密码一致性定义：注册时输入密码和二次确认密码相同。

1. 参与者

注册用户

1. 事件流
2. 基本事件流

本用例开始于注册用户希望在时间交易系统中注册

1. 系统请求注册用户，输入用户名及新密码与验证密码；  
   A1：用户名已存在  
   A2：用户名不合法。

A3：密码不合法。  
A4：两次密码不一致

1. 系统把当前用户的信息增加到数据库中。
2. 后备事件流

A1：用户名已存在

* 系统显示用户名已存在错误信息
* 返回基本事件流第1步

A2：用户名不合法

* 系统显示用户名不合法错误信息；
* 返回基本事件流第1步

A3：密码不合法

* 系统显示密码长度大于100的消息；
* 返回基本事件流第1步

A4：两次密码不一致

* 系统显示两次密码不一致错误信息；
* 返回基本事件流第1步

1. 特殊需求

密码输入框必须以密文方式呈现。

1. 前置条件

本用例开始前用户打开系统注册界面。

1. 后置条件

如果用例成功， 用户将注册成功， 系统在数据库中增加一条注册用户的相关记录。若失败，系统状态不改变。

1. 活动图

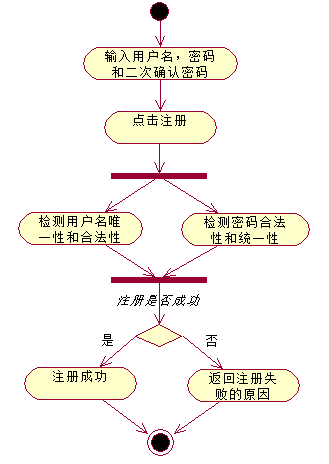


图2-2 用户注册活动图

### 用户登录

1. 简要说明

本用例描述注册用户/管理员如何登录到时间交易系统。

1. 参与者

管理员用户/注册用户

1. 事件流
2. 基本事件流

本用例开始于注册用户/管理员希望登录到时间交易系统。

1. 系统请求注册用户/管理员输入用户名和密码；
2. 注册用户/管理员输入用户名和密码；
3. 系统验证输入的用户名和密码；

C1：用户名不存在  
C2：用户名对应密码不正确

1. 注册用户/管理员成功登录到主界面，进行其他操作
2. 后备事件流

C1：用户名不存在

* 系统显示用户名不存在错误信息；
* 管理员返回基本事件流第1步，而用户将会被系统提示注册

C2：用户名对应密码不正确

* 系统显示用户名密码不正确错误信息；
* 返回基本事件流第1步

1. 特殊需求

密码输入框必须以密文方式呈现。

1. 前置条件

本用例开始前注册用户/管理员已经打开对应的系统登录界面。

1. 后置条件

如果用例成功， 注册用户/管理员将成功登录系统，并赋予相应权限。若失败，系统状态不改变。

1. 活动图

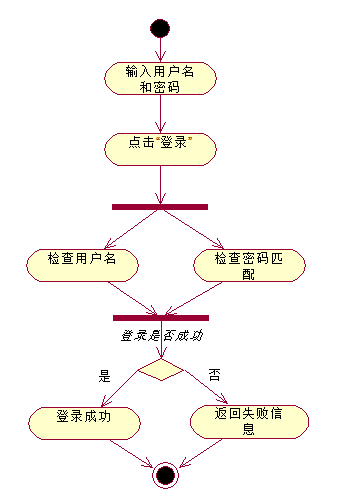


图2-3 用户登录活动图

### 个人信息管理

1. 简要说明

本用例允许用户对自己的个人信息进行查看与修改，在本系统提供技能、昵称、个性签名的修改。

1. 参与者

管理员用户/注册用户

1. 事件流
2. 基本事件流

用例开始于用户已经登录了系统，并点击了菜单栏的“个人信息”选项。

1. 系统显示个人信息页面，包括用户昵称，技能，个性签名。
2. 用户可以在文本框直接编辑自己的个人信息，包括昵称，技能，个性签名。
3. 用户填写个人信息，并点击“提交”：

C1：新技能信息字符长度超过100

C2：昵称字符长度长度超过50

C3：个性签名字符长度超过100

1. 系统成功修改注册用户的个人信息。
2. 后备事件流

C1：新技能信息字符长度超过100

* 系统显示新技能不合法错误信息；
* 返回基本事件流第1步。

C2：昵称字符长度长度超过50

* 系统显示昵称信息不合法错误信息；
* 返回基本事件流第1步。

C3：个性签名字符长度超过100

* 系统显示个性签名字符长度不合法信息；
* 返回基本事件流第1步。

1. 特殊需求

无

1. 前置条件  
    本用例开始前用户已经登录系统
2. 后置条件  
    如果用例成功，用户将成功修改个人信息。若失败，系统状态不改变。
3. 活动图

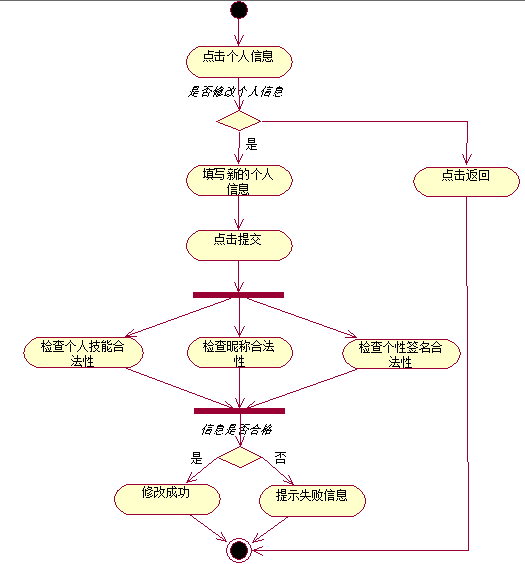


图2-4 用户个人信息管理活动图

### 发布需求

1. 简要说明

本用例允许注册用户发布时间需求。

需求关键词合法性定义：关键词不为空并且长度小于50。

需求时间合法性定义：时间为标准时间格式，结束时间晚于开始时间。

需求地点合法性定义：地点不为空并且长度小于50。

需求金额合法性定义：金额不为空并且长度小于50。

需求备注合法性定义：备注不为空并且长度小于50。

1. 参与者

管理员用户/注册用户

1. 事件流
2. 基本事件流
3. 系统请求用户输入需求的信息，关键词、时间、地点、金额、备注等信息
4. 注册新需求信息
5. 系统验证新需求信息

C1：关键词输入不合法

C2：地点信息不合法

C3：金额信息不合法

C4：时间信息不合法

C5：备注信息不合法

1. 系统修改用户需求发布历史并且同步至服务端
2. 后备事件流

C1：关键词输入不合法

* 系统显示关键词不合法错误信息；
* 返回基本事件流第1步。

C2：地点信息不合法

* 系统显示地点信息不合法错误信息；
* 返回基本事件流第1步。

C3：金额信息不合法

* 系统显示金额信息不合法错误信息；
* 返回基本事件流第1步。

C4：时间信息不合法

* 系统显示时间信息不合法错误信息；
* 返回基本事件流第1步。

C5：备注信息不合法

* 系统显示备注信息不合法错误信息；
* 返回基本事件流第1步。

1. 特殊需求

无

1. 前置条件

本用例开始前注册用户已经登录系统

1. 后置条件

如果用例成功，注册用户将成功发布新需求。若失败，系统状态不改变。

1. 活动图

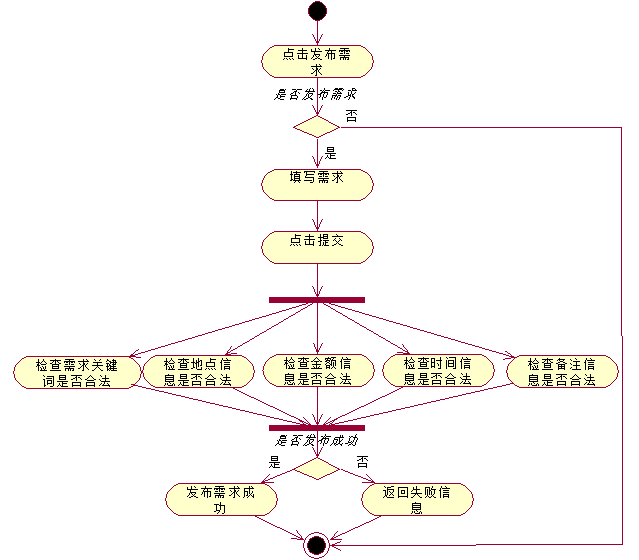


图2-5 发布需求活动图

### 接收需求

1. 简要说明

本用例允许注册用户接收时间需求。

接收需求的用户合法性定义：用户不能接收自己发布的需求。

1. 参与者

管理员用户/注册用户

1. 事件流
2. 基本事件流
3. 系统显示已发布的需求的列表，并显示每个需求的概要信息；
4. 注册用户点击某一个需求的详情查看详细信息；
5. 系统显示该需求的详细信息；
6. 注册用户点击“接单”，系统验证需求信息和用户信息；

D1：接收需求的用户不合法。

1. 显示成功接收别人发布的需求。
2. 后备事件流

D1：接收需求的用户不合法

* 系统显示接收需求的用户不合法错误信息；
* 返回基本事件流第1步

1. 特殊需求

无

1. 前置条件

本用例开始前注册用户已经登录系统

1. 后置条件

如果用例成功，注册用户将成功接收别人发布的需求。若失败，系统状态不改变。

1. 活动图

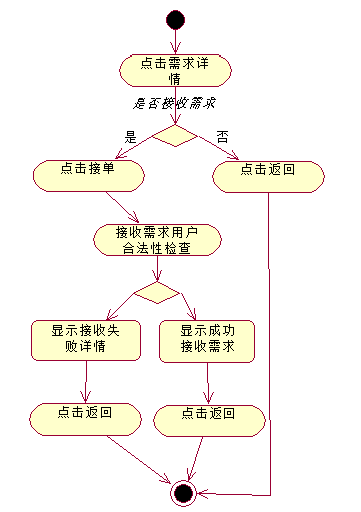


图2-6 接收需求活动图

### 用户管理

1. 简要说明

本用例允许管理员用户查看、删除用户

1. 参与者

管理员用户

1. 事件流

本用例开始于管理员用户查看或删除某个用户

1. 基本事件流
2. 系统提供当前用户列表
3. 管理员选择待删除的用户对象
4. 管理员删除选中的用户对象
5. 系统更新数据库中用户信息
6. 后备事件流

无

1. 特殊需求

无

1. 前置条件

本用例开始前管理员用户已经登录系统

1. 后置条件

如果用例成功，管理员用户将成功删除该注册用户。若失败，系统状态不改变。

1. 活动图

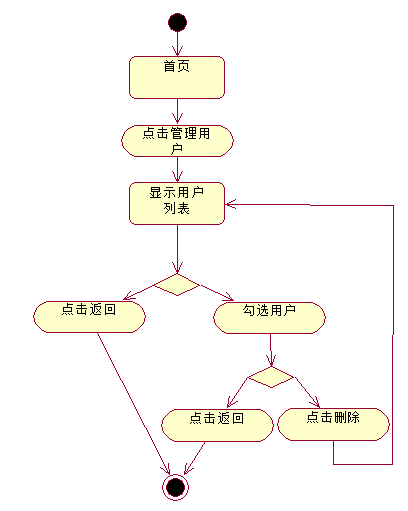


图2-7 管理用户活动图

### 需求管理

1. 简要说明

本用例允许管理员用户删除需求，注册用户删除自己发布的需求。

用户需求管理合法性定义：注册用户不能删除其他用户发布的需求。

1. 参与者

管理员用户/注册用户

1. 事件流

本用例开始于管理员用户希望删除某个需求或者注册用户希望删除自己发布的需求

1. 基本事件流
2. 系统提供当前需求列表；
3. 管理员用户/注册用户选择待删除的需求对象；
4. 管理员用户/注册用户删除选中的需求对象；

C1：注册用户需求管理不合法；

1. 系统更新用户的需求信息；
2. 后备事件流

C1：用户需求管理不合法

* 系统显示用户需求管理不合法错误信息；
* 返回基本事件流第1步；

1. 特殊需求

无

1. 前置条件

本用例开始前管理员用户/注册用户已经登录系统

1. 后置条件

如果用例成功，管理员用户/注册用户将成功删除该需求。若失败，系统状态不改变。

1. 活动图

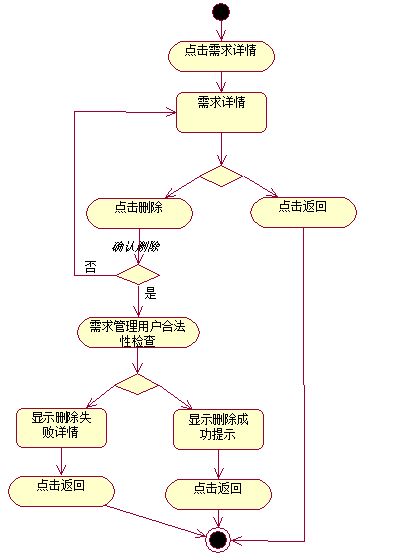


图2-8 需求管理活动图

## 补充规约

### 可靠性

本系统具备良好的可靠性，可以保证7\*24小时连续运行，一周维护时间不会超过2小时。系统要保证用户的信息，用户的历史交易记录无差错保存，并在5年内保持信息的完整性。

### 安全性

密码以密文存储在数据库中。

### 性能

本系统性能允许1000个人同时使用，页面响应时间不能超过3秒，允许大量的并发访问。

### 可支持性

本系统支持主流浏览器，如chrome浏览器，IE浏览器等。

### 可扩展性

本系统能接受对现有功能合理的改进，增加一些新的功能。

## 术语表

|  |  |
| --- | --- |
| **术语** | **含义** |
| 注册用户 | 具备独立民事行为的公民 |
| 管理员用户 | 可查看和管理用户发布的交易需求 |
| 个人信息 | 用户的个人信息页面，用户可以查看和修改个人信息 |
| 需求列表 | 用户可以查看所有用户发布的交易需求 |
| 需求详情 | 用户可以查看一个需求的详细信息，包括需求的发布时间，截止时间，交易内容等 |

# 交易系统的系统设计

## 时间交易系统框架

本系统基于MVC架构用Java开发实现， MVC架构包含界面表示层、业务层、数据访问层这三层，通过spring组装界面层、业务层、数据访问层的Bean对象，从而实现对象管理和事务管理。领域对象的持久化通过hibernate持久化框架完成，界面层运用struts实现界面和界面控制逻辑的分离。采用封装（分层）的思想来降低耦合度，使系统更加灵活，具有更好的扩展性。

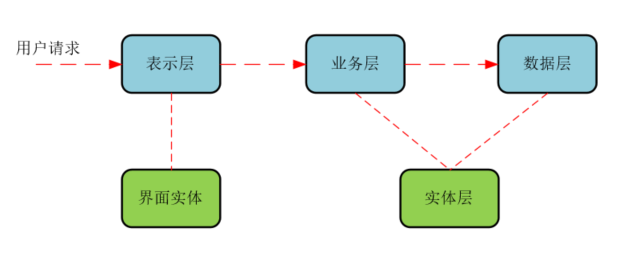


图3-1：时间交易系统框架

表示层：负责获取用户的请求和信息的展现，是用户与系统交互的界面，通过struts的界面和界面控制逻辑分离技术，实现了界面逻辑的重用。本系统表示层的模块包括用户注册，用户登录，个人资料，发布需求等等。

业务层：业务层是实现系统业务逻辑的核心。业务层用于从视图读取数据，并向模型发送数据，管理所有系统功能有关的数据流和控制流，从而实现系统的逻辑功能。向上提供服务接口给表示层调用。比如本系统中的逻辑控制类负责获取到界面传过来的数据并进行逻辑判断。

数据访问层：数据访问层封装了数据库连接，读取，写入等操作，以提供数据库访问的服务，比如本系统中有关需求的数据控制类负责增加或者删除需求，实现数据库具体操作。

实体层包含实体类，独立于上面这三层。

## 系统关键抽象

系统关键抽象即系统实体类图，用于描述系统中的类及其相互之间的各种关系。本系统总共有两个实体类，分别是User用户类和Order需求类，User类用于储存用户信息，包括用户ID、用户名、密码、技能等等，Order类用于储存需求信息，包括需求ID、发布需求的用户ID、接受需求的用户ID、需求金额、开始时间、结束时间、需求描述等等。



图3-2：系统关键抽象

# 用例分析

本章对系统中的用例进行具体分析，内容包括用例时序图分析，用例类图分析，类的分析机制，以及合并分析类。第一部分用时序图描述用户和系统的交互工程，第二部分描述边界类，控制类和实体类，第三部分描述分析类的关联关系，第四部分对所有的类进行合并分析。

用例过多，本章选取用户发布需求和用户管理个人信息两个用例进行分析。

## 用例类图分析

本节分析用例类图，每一个用例都包含相应的界面类，控制类和实体类。界面类主要展示该用例的应用界面，控制类分为逻辑控制类和数据控制类，其中逻辑控制类用于对数据进行逻辑判断，数据控制类包含对所存储的数据进行相应的增删查找功能，控制类就是用于操作数据。

### 发布需求用例的类图分析

发布需求用例的类包括Pulish边界类，OrderController需求逻辑控制器类和OrderDAO需求数据控制类，OrderController和OrderDAO都属于控制类。

边界类Publish类为新需求信息填写页面，获取的信息包括需求ID，开始时间，结束时间，金额等。

OrderController类是发布需求的逻辑控制类，负责进行发布需求和删除需求的逻辑处理。OrderDAO类是对应Order需求实体类的需求数据控制类，用于和数据库交互，发布和删除数据库里的需求。

发布需求用例析取图如图4-1所示：

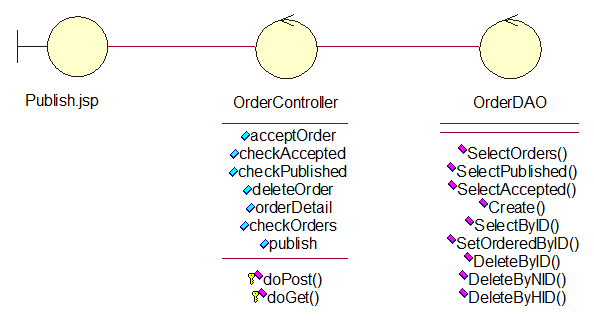


图4-1：发布需求用例析取

### 用户管理个人信息的类图分析

在管理个人信息用例中，用户可以修改个人信息，该用例包含Manage个人信息管理界面类，控制类为UserController和UserDAO ，其中UserController是用户数据的逻辑控制器，UserDAO是用户数据控制类。

边界类UserInfo类为管理个人信息页面，获取的信息包括用户ID，用户昵称，技能，个性签名，这些信息可以由用户进行编辑。

控制类为UserController和UserDAO，其中UserController类是逻辑控制类，负责进行用户信息的一些逻辑处理。UserDAO类是对应User用户实体类的数据控制类，用于和数据库交互，查找增加和删除用户。这里只有管理员有权删除用户，普通用户是不能删除用户的。

用户管理个人信息的用例析取图4-2所示。

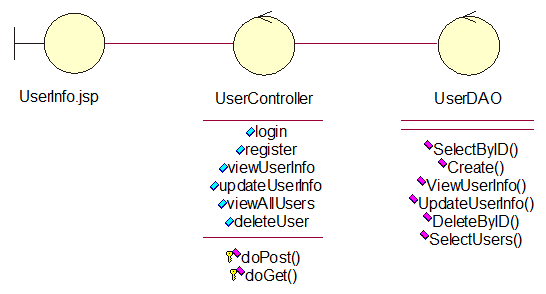


图4-2：管理个人信息用例析取

## 用例时序图分析

本节分析用户发布需求和用户管理个人信息的时序图。

### 发布需求用例的时序图分析

在发布需求用例中，边界类为Publish.jsp，控制类为OrderController逻辑控制类和OrderDAO数据控制类，其中OrderController用于数据逻辑判断，OrderDAO用于连接数据库操作需求数据。用户发布需求的时序图如图4-3所示。

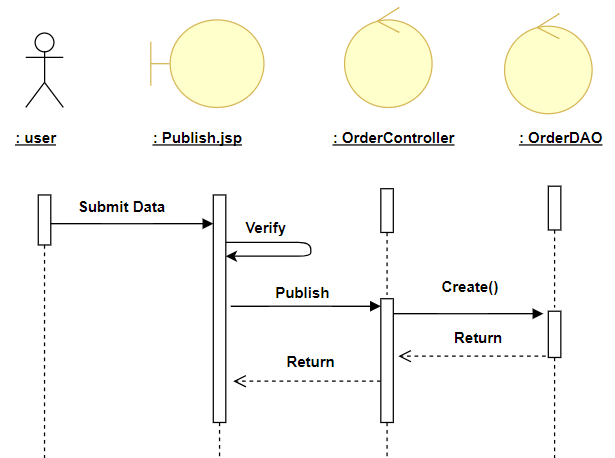


图4-3发布需求时序图

用户在新建需求的Publish界面中，填写好需求信息后，点击保存按钮，把新建需求的数据信息以post的方式传给OrderController控制类。

接着OrderController逻辑控制类先验证需求信息是否合法，判断信息是否完整，判断地点价格关键字和备注是否太长，判断时间格式是否合法以及开始时间是否在结束时间之前，如果这些都合法就可以新建需求，和OrderDAO数据控制类做交互。

OrderDAO数据控制类调用Create()方法创建需求对象，成功创建则返回成功信息，若创建失败则返回错误信息。

### 用户管理个人信息的时序图分析

在用户管理个人信息的用例中，边界类为UserInfo，控制类为UserController和UserDAO，UserController是业务逻辑层的控制类，UserDAO是数据访问层的用户数据控制类。用户管理个人信息的时序图如图4-4所示:

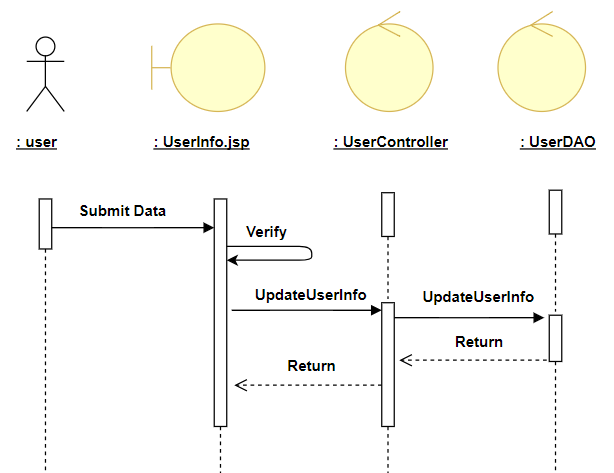


图4-4用户管理个人信息的用例的时序图

在用户管理个人信息的UserInfo界面中，用户填写个人信息，点击修改按钮，把该用户新的个人信息以post的方式传给UserController控制器。

接着UserController逻辑控制类进行数据的逻辑判断，调用UpadteUserInfo对象方法验证用户昵称，技能和个人信息的长度是否合法，如果这些都合法就发送信息给UserDAO数据处理类。

最后OrderDAO数据控制类调用UpdateUserInfo()方法更新用户信息，如果更新成功则返回成功信息，若更新失败则返回错误信息。

这里两个UpdateUserInfo()方法只是刚好重名，UserController逻辑控制类中的UpdateUserInfo用于逻辑判断，是否可以更新用户信息，OrderDAO数据控制类中的UpdateUserInfo用于跟数据库交互，实现更新用户信息。

## 类的分析机制

本节对系统所涉及到的所有类进行机制分析。

界面类：Main.jsp是显示系统所有需求的主页界面，Reg.jsp是用户注册界面，Login.jsp是用户登陆界面，Publish.jsp是发布需求界面，NeedHistory.jsp是查看自己所发布的需求的界面，HelpHistory.jsp是查看自己所接收的需求的界面，OrderDetail.jsp是显示某个需求详情的界面，AcceptOrder.jsp是用户接收需求的界面，DeleteOrder.jsp是用户删除需求的界面，UserInfo.jsp是用户管理个人信息的界面，Manage.jsp是管理员管理用户的界面，这些界面类都拥有通信机制，可以跟用户进行交互。

实体类包括Order需求类和User用户类，拥有持久性和安全性。

控制类分为业务逻辑层的控制类和数据访问层的数据控制类，逻辑控制类负责跟界面交互来获取到用户所提交的信息，数据控制类负责跟数据库交互来实现对数据相关的操作，控制类拥有资源管理的机制。控制类总共有四个，包含OrderController，UserController，OrderDAO和UserDAO类。在逻辑控制类中，OrderController是需求逻辑控制类，UserController是用户逻辑控制类，在数据控制类中， OrderDAO是需求数据控制类，UserDAO是用户数据控制类。

最后可以得到如表4-1的系统分析机制表：

表4-1 分析机制表

|  |  |
| --- | --- |
| **分析类（analysis classes）** | **分析机制(analysis mechanism)** |
| Main | 通信 |
| Reg | 通信 |
| Login | 通信 |
| Pulish | 通信 |
| HelpHistory | 通信 |
| NeedHistory | 通信 |
| OrderDetail | 通信 |
| AcceptOrder | 通信 |
| UserInfo | 通信 |
| Manage | 通信 |
| User | 持久性和安全性 |
| Order | 持久性和安全性 |
| UserController | 资源管理 |
| OrderController | 资源管理 |
| UserDAO | 资源管理 |
| OrderDAO | 资源管理 |

## 合并分析类

在跟用户相关的类中，首先Main，Manage，Reg，Login，UserInfo界面类都和UserController逻辑控制类进行交互，UserController和UserDAO数据控制类进行交互，而UserDAO最后和User实体类交互。

在跟需求相关的类中，Main，Publish，HelpHistory，NeedHistory，OrderDetail， AcceptOrder，DeteleOrder，UserInfo和Manage界面类和OrderController逻辑控制类进行交互，OrderController和OrderDAO数据控制类进行交互，而OrderDAO最后和Order实体类交互

本系统整体类的析取图如图4-5所示：

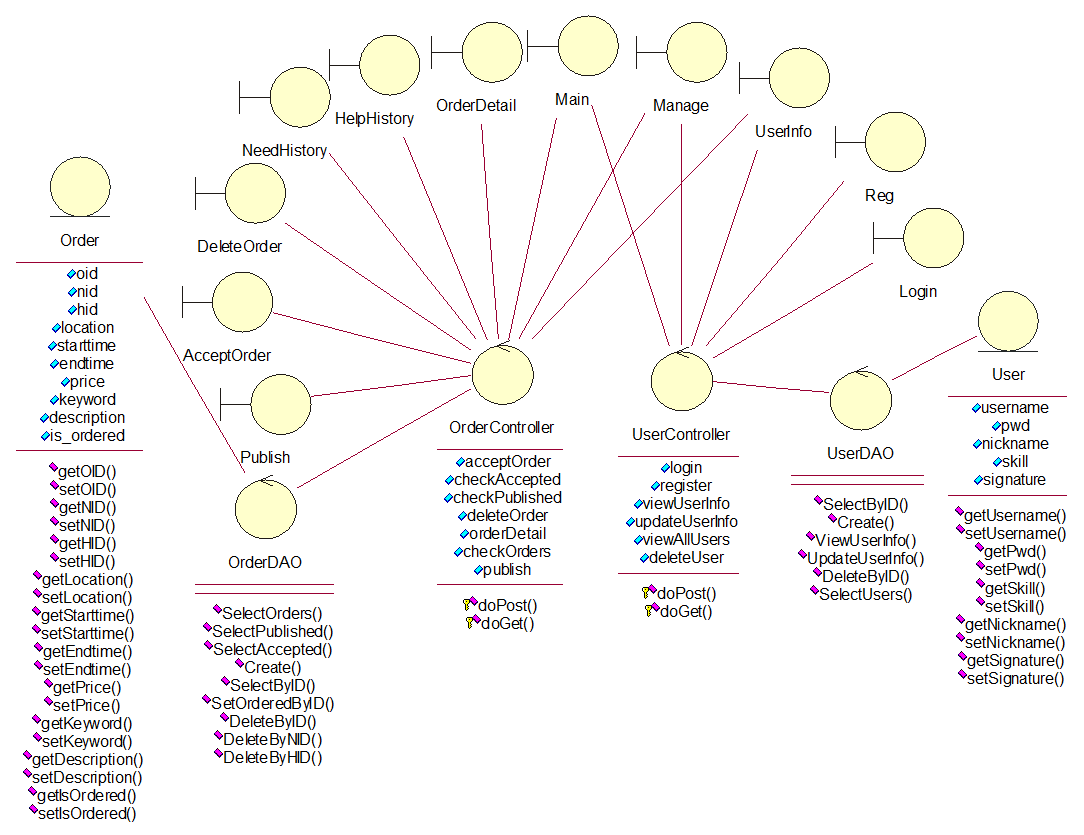


图4-5本系统整体类的析取图

注意这里，Manage界面类同时和UserController控制类以及OrderController控制类交互，Manage界面类用于管理员管理用户，需要与UserController类交互，返回时会跳转到OrderController类查询需求，在Main主页面实时显示当前需求。

同理，UserInfo界面类同时和UserController控制类以及OrderController控制类交互，UserInfo界面类用于用户修改个人信息，需要与UserController类交互，返回时会跳转到OrderController类，在Main主页面实时显示当前需求。

Main界面类也同时和UserController控制类以及OrderController控制类交互，因为Main界面会显示当前需求，所以会与OrderController控制类交互，并且Main界面会跳转到用户管理界面，所以也会与UserController类交互。

# 附录

附录详细说明本系统所有代码及其功能。JSP文件共11个，实现前端界面，Java文件共25个，实现后台操作。

JSP代码说明表5-1所示：

表5-1 Jsp文件说明

|  |  |
| --- | --- |
| **Jsp文件** | **界面描述** |
| Main.jsp | 显示系统未被接受需求的主页界面 |
| Reg.jsp | 用户注册界面 |
| Login.jsp | 用户登陆界面 |
| Publish.jsp | 发布需求界面 |
| HelpHistory.jsp | 查看自己所接收的需求 |
| NeedHistory.jsp | 查看自己所发布的需求 |
| OrderDetail.jsp | 显示某个需求详情 |
| UserInfo.jsp | 用户查看、管理个人信息的界面 |
| Manage.jsp | 管理员查看、管理用户的界面 |
| DeleteDetail.jsp | 删除某个需求详情的界面 |
| AcceptOrder.jsp | 用户删除需求的界面 |

Java文件说明如表5-2所示：

表5-2 Java文件说明

|  |  |
| --- | --- |
| **Java文件** | **功能描述** |
| Order.java | 需求实体类：**属性、setter、getter方法** |
| User.java | 用户实体类：**属性、setter、getter方法** |
| OrderController.java | 需求的逻辑控制类 |
| UserController.java | 用户的逻辑控制类 |
| OrderDAO.java | 需求的数据控制接口Interface |
| UserDAO.java | 用户的数据控制接口Interface |
| DAOFactory.java | 用于获取DAO实例 |
| UserDAOImpl.java | 实现UserDAO接口类的方法，但不负责数据库的打开和关闭 |
| UserDAOProxy.java | **数据库的打开和关闭，并调用UserDAOImpl类对象(真实对象类)的操作** |
| OrderDAOImpl.java | 实现OrderDAO接口类的方法，但不负责数据库的打开和关闭 |
| OrderDAOProxy.java | **数据库的打开和关闭，并调用OrderDAOImpl类对象(真实对象类)的操作** |
| DataBaseConnection.java | 负责连接数据库 |
| AcceptOrder.java | 用户接收需求，并返回成功或失败信息 |
| CheckAccepted.java | 用户查看自己接收的需求 |
| CheckOrders.java | 显示系统中未被接收的需求 |
| CheckPublished.java | 用户查看自己发布的需求 |
| DeleteOrder.java | 用户删除需求 |
| DeleteUser.java | 管理员删除用户 |
| Login.java | 用户登陆，判断用户名与密码，合法就进行登录 |
| OrderDetail.java | 请求需求详情 |
| Publish.java | 发布需求，写入Order新纪录 |
| Register.java | 用户注册，判断用户名与密码，合法就可以注册 |
| UpdateUserInfo.java | 提交用户更改的个人信息 |
| ViewAllUsers.java | 请求系统所有用户 |
| ViewUserInfo.java | 请求用户个人信息 |