

《软件工程》课程作业

－资源回收 O2O 平台项目技术文档

**2020-2021(1)**



项目名称 资源回收 O2O 平台

组长姓名

小组其他成员姓名 沈丹 桂昳然 朱思捷

指导教师 江 颉

所在学院 计算机科学与技术学院

提交日期 2021/1/

**项目评分表**

|  |  |  |  |
| --- | --- | --- | --- |
| **项目组成员** | **任务分工** | **小组自评** | **教师评分** |
| 张三（组长） |  | 90 |  |
| 李四 |  | 85 |  |
| 王五 |  | 85 |  |
|  |  |  |  |
|  |  |  |  |
| **小组项目总体评分** | | **90** |  |

目录

[第1章 绪论 1](#_Toc61098714)

[1.1 项目背景 1](#_Toc61098715)

[1.2 项目基本情况 2](#_Toc61098716)

[1.3 同类产品的分析和比较 3](#_Toc61098717)

[1.4 项目的应用价值和社会效益 4](#_Toc61098718)

[ 项目的高效管理及操作 4](#_Toc61098719)

[ 社会效益 4](#_Toc61098720)

[第2章 项目管理计划和实施过程 6](#_Toc61098721)

[1. 简介 6](#_Toc61098722)

[2.1项目概述 6](#_Toc61098723)

[2.2项目管理计划的演变 7](#_Toc61098724)

[2. 参考资料 7](#_Toc61098725)

[3. 定义和术语 8](#_Toc61098726)

[4. 项目组织 8](#_Toc61098727)

[4.1 外部接口 8](#_Toc61098728)

[4.2 内部结构 8](#_Toc61098729)

[4.3 规则和职责 8](#_Toc61098730)

[5. 管理过程计划 8](#_Toc61098731)

[5.1 启动计划 8](#_Toc61098732)

[5.2 工作计划 9](#_Toc61098733)

[5.3 控制计划 9](#_Toc61098734)

[5.4 风险管理计划 10](#_Toc61098735)

[5.5 项目停止计划 10](#_Toc61098736)

[6. 技术过程计划 10](#_Toc61098737)

[6.1 过程模型。 10](#_Toc61098738)

[6.2 方法、工具和技术。 10](#_Toc61098739)

[6.3 基础设施计划。 10](#_Toc61098740)

[6.4 产品验收计划。 10](#_Toc61098741)

[7. 支持过程计划 11](#_Toc61098742)

[7.1 配置管理计划。 11](#_Toc61098743)

[7.2 测试计划。 11](#_Toc61098744)

[7.3 文档计划。 11](#_Toc61098745)

[7.4~5 质量保证计划和检查和审计计划。 11](#_Toc61098746)

[7.6 问题解决计划。 11](#_Toc61098747)

[7.7 次承包商管理计划。 11](#_Toc61098748)

[8. 附加规划 11](#_Toc61098749)

[第3章 需求说明书 12](#_Toc61098750)

[3.1 可行性分析 12](#_Toc61098751)

[ 经济可行性 12](#_Toc61098752)

[ 技术可行性 12](#_Toc61098753)

[ 社会可行性 13](#_Toc61098754)

[ 风险因素及对策 13](#_Toc61098755)

[ 可行性分析结论 14](#_Toc61098756)

[3.2 成本分析 14](#_Toc61098757)

[3.3 系统需求 15](#_Toc61098758)

[第4章 传统的分析说明书 20](#_Toc61098759)

[4.1 数据流图 20](#_Toc61098760)

[4.2 数据字典 26](#_Toc61098761)

[4.3 ER图 33](#_Toc61098762)

[第5章 面向对象的分析说明书 34](#_Toc61098763)

[5.1 接口类图 34](#_Toc61098764)

[5.2 总体用例类图 35](#_Toc61098765)

[5.3顺序图 35](#_Toc61098766)

[5.3.1 登陆注册模块 35](#_Toc61098767)

[5.3.2回收订单交易模块 38](#_Toc61098768)

[5.3.3 捐赠模块 39](#_Toc61098769)

[5.3.4 取消订单模块 40](#_Toc61098770)

[第6章 面向对象的设计说明书 41](#_Toc61098771)

[6.1 总体类图关系 41](#_Toc61098772)

[6.2 详细类图关系 42](#_Toc61098773)

[6.2.1 具体类图详细设计 42](#_Toc61098774)

[6.2.2具体用例类图及详细设计 43](#_Toc61098775)

[第7章 测试技术说明书 58](#_Toc61098776)

[7.1 注册测试 58](#_Toc61098777)

[7.2 登录测试 58](#_Toc61098778)

[7.3 用户下订单测试 58](#_Toc61098779)

[7.4 回收员确认完成订单测试 59](#_Toc61098780)

[7.5 评价测试 59](#_Toc61098781)

[7.6 管理员审核回收员 59](#_Toc61098782)

[7.7管理员修改、增加测试 59](#_Toc61098783)

[第8章 用户使用说明书 61](#_Toc61098784)

[第9章 做小组项目的体会与心得 61](#_Toc61098785)

[9.1 61](#_Toc61098786)

[参考文献 62](#_Toc61098787)

[附录 63](#_Toc61098788)

[附录A 第4小组资源回收O2O项目的源代码.rar 63](#_Toc61098789)

[附录B 第4小组资源回收O2O项目的安装说明书.doc 63](#_Toc61098790)

# 绪论

## 1.1 项目背景

资源回收的巨大市场空间及交易规模：随着我国人民生活水平的提高，废旧物品产生量也在剧增。根据一份再生资源回收利用市场分析报告中写到，目前的中国再生资源回收企业有5000多家，回收加工处理工厂3000多家，年经营额在6000亿至7000亿元间。将进城收废品的农民工计算在内，从业人员接近1000万。2014年我国十大类别的再生资源回收总量约为2.56亿吨，回收总值为 5902.8亿元。在所有再生资源类别中，C 端消费品可回收资源占总回收总值的27%左右，主要包括一些废纸、废塑料、废弃电子电器产品、废木材（家具）和废纺织品等。

业务背景：

互联网+废品回收的想象空间：从资源回收产业链改造的角度去说，互联网取缔了“回收站点”，将零散的回收从业者个人统一起来，使旧物直接从用户家中到达回收基地（大型回收分拣处理企业），缩短了产业链流程以节约成本，包括单人、单点覆盖面的时间成本，运输成本、层层转手产生的二次成本。如果从 规模上来说，在完全规模化以后，还能实现城市回收基地的取代，直接与再生产 企业建立业务往来，这中间成本几乎为零。另一方面，如果旧物回收工具能够成功进入家庭或社区，成为一道入口，在这个方向上将更加具有想象力。

尽管看起来废品回收的市场空间巨大，事实上这个行业的链条非常长，环节参与者多而且混乱，规则不强。将每个环节的从业者都看做该行业的利益相关诉求者，那将是一件非常累人的事。所以对于互联网创业者来说，首先是理清产业链各层级的不同需求，找准切入点，挖掘行业机会。

（1）城市居民：在N年以前，居民都是在家里攒下好多废品，等着某天小区楼底下一声 吼“收破烂儿嘞！”，然后将其卖掉。但是对于今天的一线城市来说，一方面，很多年轻人已经没有攒废品的习惯，另一方面，废品物价值不高，价格经常波动，叫价不透明，有些时候即便是攒了很久的东西也不能卖到一个好价钱。所以，对于那些有攒废品、卖废品的居民来说，如何让废品价格透明、并且在想卖的时候就能方便的卖出去是攒废品居民最大的诉求。补充说明一 下，此处，我将城市流动拾荒者也归类于有积攒废品行为习惯的居民，此外 绝大多数废品回收商贩也同时是拾荒者。

（2）废品回收商贩：对于目前国内的废品回收现状来看，城市大部分可再生资源是被当成垃圾扔掉，而到了拾荒者手里，仅北京的一线拾荒者就突破十万人，可以说城市废品回收站的可再生资源绝大部分来自拾荒者和废品回收小商贩。对于这些商贩和拾荒者而言，有两方面的需求：“经营”效率最大化。个人商贩的经营特点是，以个人为中心、无牌无证、无经营范围、流动性作业，同行之间用默契和潜规则划定所属的区域，所以他们的经营往往稳定性差，有时候跑了一整天收到的“货”少，或者刚被别人收走了。如何让他们尽可能避免白跑路，提升收“货”效率至关重要。“货物”价值最大化。废品回收商的营收来自于低价收取居民散户手中的废品再高价卖给回收站的差价获利，这种买进卖出的方式缺点之一是积压货品，当市场行情不好的时候，就意味着他们手里的废品不能及时变成现金或者是亏本。综上所述问题，我们可以 为用户与有回收人员搭建一个统一的平台，实现废品的流转。

本项目采用O2O模式(Online to Offline)，将线下商务的机会与互联网结合在了一起，让互联网成为线下交易的前台。线下服务使用线上揽客，消费者可以使用线上来筛选服务，使得每笔交易可查，很快达到规模。

同时，此模式对于项目的线下模块也是一个不小的挑战，很大程度上，线下能力的高低决定了交易的次数和质量。平台只是提供了一个方式，线下的交易和管理才是最重要的方面。

## 1.2 项目基本情况

从资源回收产业链改造的角度去说，互联网取缔了“回收站点”，将零散的回收从业者个人统一起来，使旧物直接从用户家中到达回收基地（大型回收分拣处理企业），缩短了产业链流程以节约成本，包括单人、单点覆盖面的时间成本，运输成本、层层转手产生的二次成本。如果从规模上来说，在完全规模化以后，还能实现城市回收基地的取代，直接与再生产企业建立业务往来，这中间成本几乎为零。另一方面，如果旧物回收工具能够成功进入家庭或社区，成为一道入口，在这个方向上将更加具有想象力。

然而O2O平台在发展过程中依然面临着种种挑战：

* 如何统一社区进行回收资源外包；
* 如何整合回收的资源进行统计；
* 如何增强接单能力，盘活整个回收资源市场；
* 如何根据接单位置判断该由哪家回收站点派人员进行操作。

因此，设计接单范围划分的算法就是重中之重。资源回收O2O平台建成后需要满足如下业务需求：

* 能够促进资源回收利用的分配和使用，令每一有关行业都满意；
* 能够很好的完成平台的运维管理工作；
* 平台能够多增加一下捐赠回收公益项目；
* 社会效益与经济效益并重，优先考虑社会效益。

资源回收O2O平台建成后需要具备如下的功能特性：

**资源回收平台**：能够为实现社区用户下单提交订单，查看废品种类、订单、用户信息等提供有效支撑；

**资源捐赠平台**：在回收资源的前提下开展资源捐赠，一些旧衣物或其他可用但不需要的物品进行捐赠为社会增加有用资源，造福贫困人民。

**统一接单中心**：对于所有的订单进行自动派单，废品回收员主动上门回收，平台进行支付，操作简单，便于推广。

## 1.3 同类产品的分析和比较

“资源回收O2O平台”的创新不在于技术上的创新，而在于组合多种已有的技术来实现其理念上的创新，下面提到的一些功能和技术，又是已有的，但是巧妙的组合能发挥出非一般的功能。

现在的回收平台网站大多只具备提供回收资源的功能，好的页面和多的功能难以全部结合实现，分类太粗糙不知道如何进行选择，而我们的平台分类精细，操作简单，还有辅助功能，还适合高效管理。

本项目实现的系统与同类产品的比较情况如下表所示。

**表1 与同类产品的比较**

|  |  |  |
| --- | --- | --- |
| **系统名称** | **存在的问题** | **主要创新点** |
| 速回收CC | 只与数码产品有关；小众，很多地方没有该平台线下店。 | 回收数码产品，还有二手交易平台，可以以旧换新。 |
| 飞蚂蚁 | 虽然种类丰富，但是细分的几乎无，选择可能性小。 | 回收种类丰富，且价格合理；和各大有名平台有合作，推广高；可以根据回收获得都进行免费领取商品的活动。 |
| 资源回收平台 | 网页打开慢，操作不能很流畅。 | 有推荐商家和供求，可以提供精准的帮助。 |
| 宁波市再生资源回收服务平台 | 页面做的有些简略，操作不易；只有宁波市可用。 | 有每日的指导价格，回收资源表。 |

**（续）表1 与同类产品的比较**

|  |  |  |
| --- | --- | --- |
| 系统名称 | 存在的问题 | 主要创新点 |
| 九管家 | 数量少的不予进行交易操作，大的交易也要操作较长时间。 | 预约下单界面简便，页面舒适，有市价指导。 |
| 本项目的系统 | 页面简略，未实现公众号移植。 | 资源种类介绍繁多，操作便捷，可以查看地图；有统计样式。 |

## 1.4 项目的应用价值和社会效益

“资源回收O2O平台”的应用价值和社会效益主要体现在：

* 项目的高效管理及操作

首先，平台能根据用户的需求，进行订单的实现，方便了用户也保障了废品回收员的基础资金，交互性强，适用于更好的管理。

其次，平台提供了追踪功能，用户可以查看自己的订单信息和回收站之间的位置关系。更方便用户进行回收交易。

管理员可以管理用户信息和废品回收员信息，回收员的人员注册保证有一个确切的依据和管理，保障了用户和回收员各个方面的权利。考虑到回收员可能是一些不太会使用智能产品的高龄人员，配置的专门的回收站点是为了减轻回收员的压力，也更方便管理。

* + - 社会效益

该项目的具体实施可以收整归纳行业，整合某一片地区的回收资源管理，缩短了产业链流程以节约成本，包括单人、单点覆盖面的时间成本，运输成本、层层转手产生的二次成本。甚至直接与再生产企业建立业务往来，使这中间成本几乎为零。另一方面，如果旧物回收工具能够成功进入家庭或社区，成为一道入口，更加方便了人民生活。资源回收的整合本就有利于生态环境的优化，使得生态环境变得越来越好，也更方面了环境资源整改的操作。

本项目还立志于公益项目的实施，在回收资源的同实，帮助一些偏远地区和贫困人民的生活，也对捐赠该物品的用户来说是一个更方便接触公益的很好的平台，可以说是互利互惠的一种关爱社会的形式，体现着市民之间的友好社会氛围。

可以说资源回收O2O平台的建立是社会和经济效益的综合优秀效益的体现，是一个必不可少的项目，更有利于社会的美好生活建设。应用方面广泛，非常有利用价值。

# 项目管理计划和实施过程

## 简介

### 2.1项目概述

#### 2.1.1意图、范畴和目标

资源回收平台改变了传统的资源回收方式，建立了培养了互联网背景下引领的新型废品回收渠道。较比传统的资源回收方式，资源回收O2O平台更是颠覆了传统的回收贩子沿路回收的格局，互联网取缔了“回收站点”，将零散的回收从业者个人统一起来，使旧物直接从用户家中到达回收基地（大型回收分拣处理企业），缩短了产业链流程以节约成本，包括单人、单点覆盖面的时间成本，运输成本、层层转手产生的二次成本。将线上和线下相融合。将回收全过程打造成一个紧密的O2O闭环，最大限度优化用户以及回收商贩的体验，改变传统的物品回收方式，同时整合多种多类的物品回收，最大限度的优化使用使用体验，节约双方的时间和资源。

#### 2.1.2假设和限制

A.包含以下限制：

B.必须满足最后期限。

C.必须满足预算限制。

D.产品必须是可靠的。

E.结构必须是开放的，以便将来增加额外的功能。

F.产品必须是用户友好的。

#### 2.1.3项目可交互使用

整个产品包含用户手册，将在项目开始后7个星期交付使用。

#### 2.1.4时间表和预算概述

每个工作流的周期、人员需求和预算如下所示:

需求流:(1个星期，5个小组成员，12500元)

分析流:(1个星期，5个小组成员，12500元)

设计流:(2个星期，5个小组成员，25000元)

实现流:(2个星期，5个小组成员，25000元)

测试流:(1个星期，5个小组成员，12500元)

总的开发时间是7个星期，总的内部成本为112500元。

### 2.2项目管理计划的演变

版本更新：

12.21-12.27更新

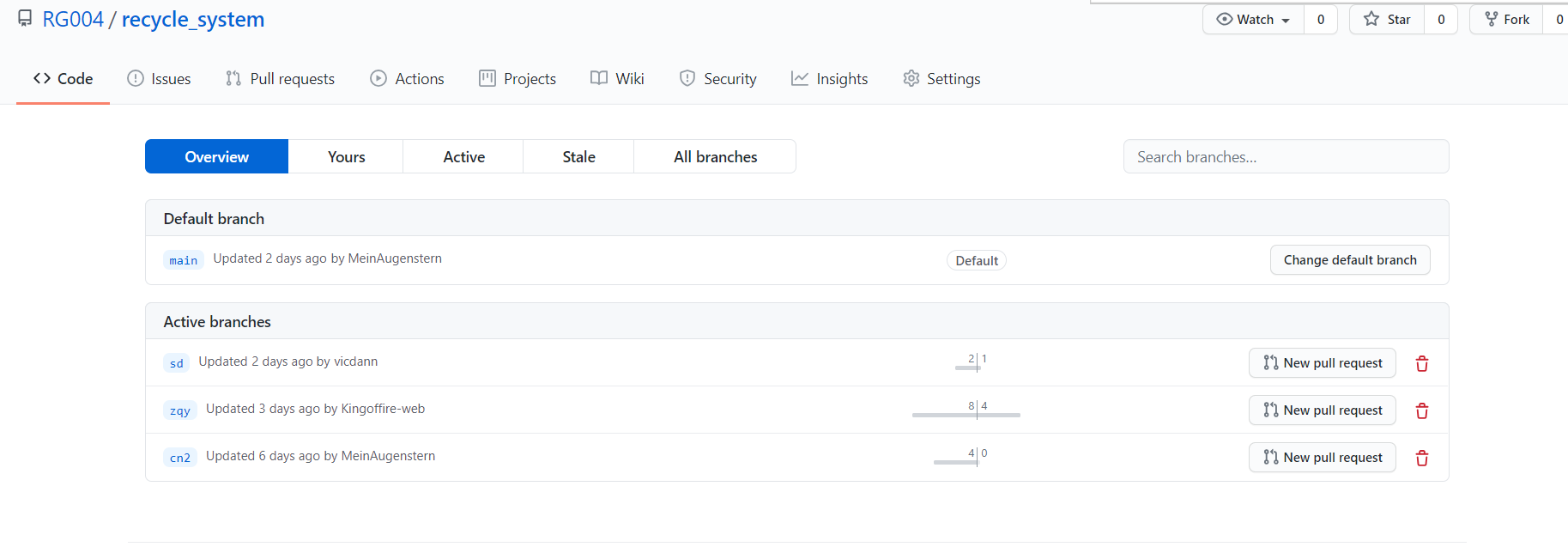


图2-1 第一次版本更新

2020.12.28-2021.1.3更新

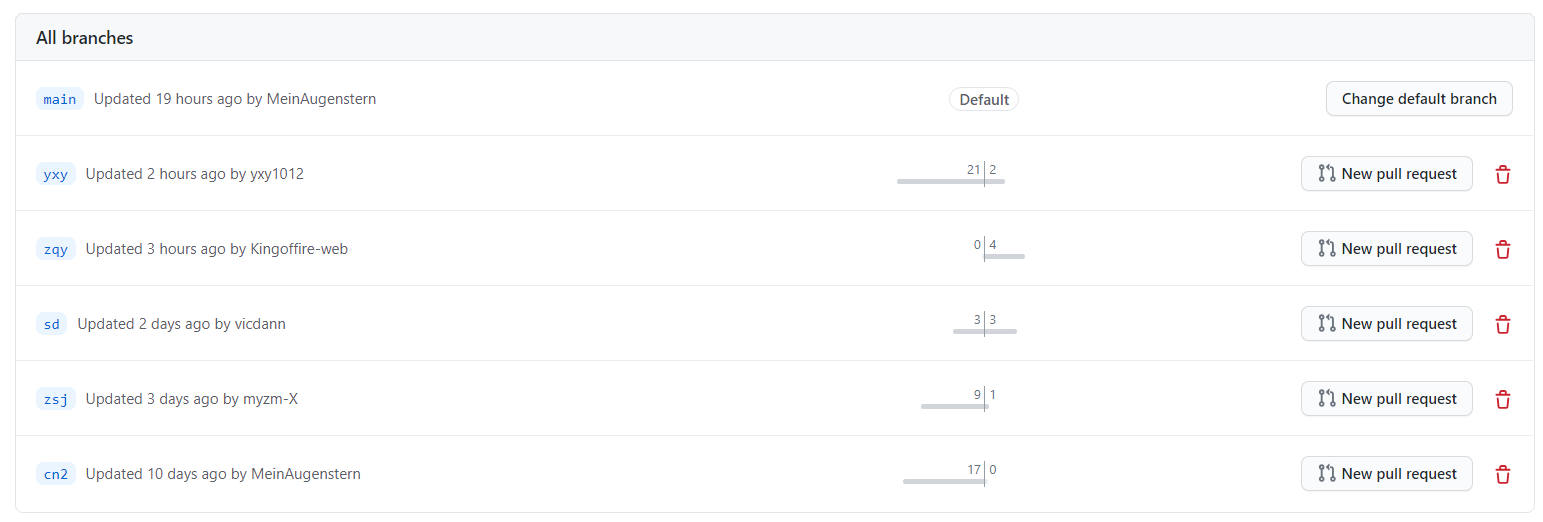


图2-2 第二次版本更新

1.4-1.10更新

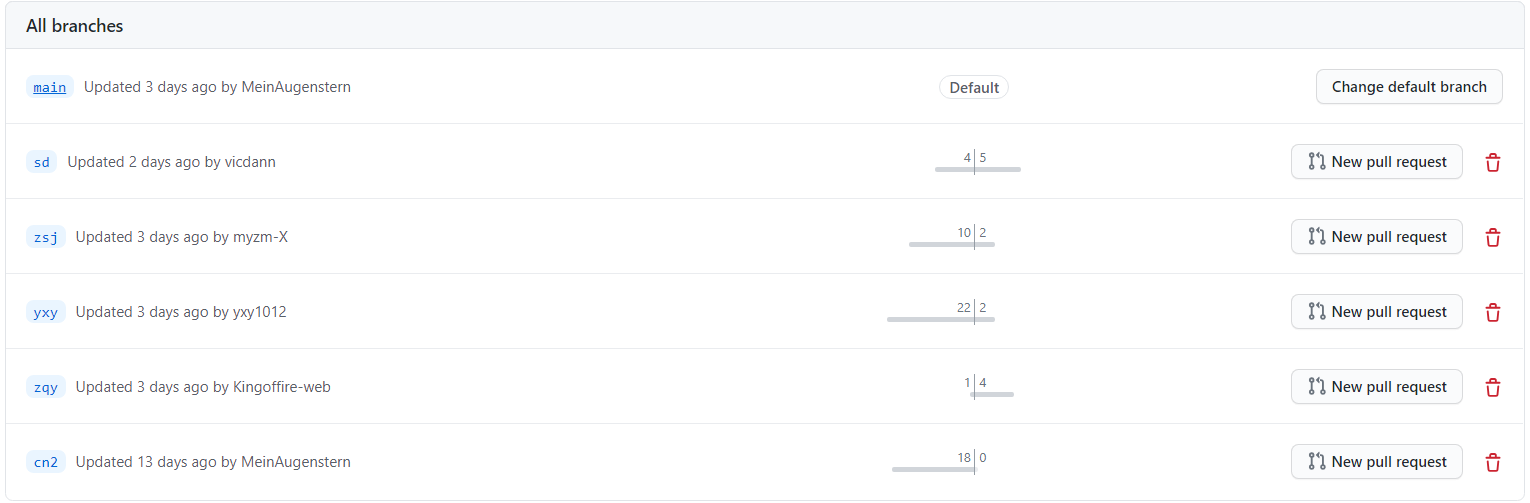


图2-3 第三版本更新

## 参考资料

所有制品都将符合公司的编码、编制文档和测试标准。

## 定义和术语

O2O——Online To Offline，指将线下的商务机会与互联网结合，让互联网成为线下交易的平台。

## 项目组织

### 4.1 外部接口

这个项目的所有工作都有杨昕语、朱琼宇、沈丹、桂昳然、朱思捷完成。每周组织会议，报告进展情况，并讨论可能的修改和调整。

### 4.2 内部结构

开发小组包含杨昕语、朱琼宇、沈丹、桂昳然、朱思捷。

### 4.3 规则和职责

金明瀚和张真瑜将完成设计流，何贤康和何泽伟将实现类定义。张真瑜和何泽伟将构建乘客下单的制品。陈南将完成管理员管理的制品并监督集成和软件产品的整个质量。每个成员都将负责自己所生成的制品的质量。

## 管理过程计划

### 5.1 启动计划

#### 5.1.1 估算计划

如前所述，整个开发时间估计为7周，整个内部成本为112500元。这些数字是通过平均程序员月收入推算得到的。

#### 5.1.2 人员计划

因时间紧张，故整个7周都需要所有成员集体参与讨论和编写。

#### 5.1.3 资源获取计划

该项目所有必须的硬件、软件和CASE工具已经具备。该产品在交付时，只需发布到服务器上。

#### 5.1.4 项目人员培训计划

这个项目不需要额外的人员培训。

### 5.2 工作计划

#### 5.2.1~2 工作活动和时间表分配

第一周：（已完成）小组讨论，确定需求制品

第二、三周：（已完成）对前端进行技术研究，生成分析制品，审查分析制品，审查软件项目管理计划。

第四、五、六周：实现并审查每个类、单元测试和文档。对每个类进行集成，对前端进行集成测试和技术测验。

第七周：集成测试，功能多改少补，进行最终产品测试，审查文档。

#### 5.2.3 资源分配

五个小组成员将划分为三个模块组，在指定给他们的制品上单独工作，分配给王晨、黄占钊的角色是完成司机模块，负责司机系统的实现。分配给的角色是完成乘客模块，负责乘客系统的实现;分配给陈南的角色是完成管理员模块，负责管理员系统的实现。

陈南将确保时间表和预算需求相符合，陈南也是陈南的职责。

使错误最少，用户友好程度最大是陈南的首要任务，陈南负责整个文档的质量，并确保这些文档是最新的。

#### 5.2.4 预算分配

每天工作流的预算如下所示：

需求流:(12500元)

分析流:(12500元)

设计流:(25000元)

实现流:(25000元)

测试流:(12500元)

总计：112500元

### 5.3 控制计划

任何影响里程碑或预算的主要修改必须得到陈南的统一，并形成文档。让每个人测试其他人的工作成果，确保测试的公正性。这通过每周与小组成员的例会实现。在每次会议上，各个小组成员汇报自己的进度和问题。陈南确定他们是否像所期望的那样有进展，以及他们是否按照规格说明文档和项目管理计划行事。小组成员遇到的任何主要问题都将立即报告给陈南。

### 5.4 风险管理计划

风险因素和跟踪机制如下所述:

这个新产品没有已存在的软件可以进行对比。因而，该产品不能与已存在的软件并行运行。因此，该产品应该进行广泛的测试。

假设客户对计算机不熟悉，因此，在分析流和与客户交流时需要予以特别的注意。该产品应尽可能地做到用户友好。

总会可能出现一一个主要的设计错误，因此在设计流应进行广泛的测试。还有，每个小组成员先测试自己的代码，然后测试其他成员的代码。陈南负责集成测试和产品测试。

产品必须符合特定的存储要求和响应时间。由于产品的规模小，这不应是主要问题，但陈南必须在整个开发期间进行监控。

### 5.5 项目停止计划

这里不适用。

## 技术过程计划

### 6.1 过程模型。

使用统一过程

### 6.2 方法、工具和技术。

该工作流将依照统一过程进行。该产品将用JAVA语言，用Springboot和vue-Element框架实现。

### 6.3 基础设施计划。

该产品将使用运行在个人电脑上的Idea进行开发。

### 6.4 产品验收计划。

由客户进行的产品验收按照统一过程的步骤进行。

## 支持过程计划

### 7.1 配置管理计划。

对于所有制品将全程使用码云。

### 7.2 测试计划。

各组员先对各自功能进行测试，再执行统一过程的测试流。

### 7.3 文档计划。

按照统一过程的规定生成文档。

### 7.4~5 质量保证计划和检查和审计计划。

各小组互相测试代码，陈南最后进行集成测试，最后五个人共同进行扩展的产品测试

### 7.6 问题解决计划。

小组面临的任何主要问题都将报告。

### 7.7 次承包商管理计划。

这里不适用。

## 附加规划

无

# 需求说明书

## 3.1 可行性分析

* 经济可行性

废品回收是每家每户都要经历的事，而资源回收O2O平台，以预约上门的服务形式，用户可以获取到派送员的联系方式进行互相的联系，这种线上线下一起结合起来的模式，能够最大程度的优化用户的废品回收体验，改变了传统的用户卖废品的模式，这种方式节约了不必要的沟通成本，还最大化节省了用户和派送员的资源与时间。O2O模式目前在我国发展迅速，具有一定的高效性、广泛性和低成本性，极大程度上满足了用户和派送员双方的需求。

* 技术可行性

本平台采用B/S架构，方便客户的浏览、使用。我们根据J2EE三层结构，分为表现层、业务逻辑层和数据服务层，该分层结构，提高了软件的后期可维护性，确保网站平台的高效、稳定、安全。

本平台采用前后端分离的技术，也是当下社会的主流技术。随着互联网技术的发展以及终端设备的不断增多，前后端分离技术已成为移动互联网领域不可或缺的技术。前后端分离技术的不断完善，让前后端的分工与系统边界划分越来越清晰。正是由于这种日趋严格的边界定义，致使前端工程师很少关注后端技术架构，后端工程师很少关注前端技术框架。

本平台前端使用vue框架，vue是个轻量级的框架，它只关注视图层,是一个构建数据的视图集合,大小只有几十kb。Vue.js通过简洁的API提供高效的数据绑定和灵活的组件系统。Vue最大的优点就是双向绑定数据，也就是所谓的响应式数据绑定。指vue.js会自动对页面中某些数据的变化做出同步的响应。也就是说，vue.js会自动响应数据的变化情况，并且根据用户在代码中预先写好的绑定关系，对所有绑定在一起的数据和视图内容都进行修改。通过MVVM思想实现数据的双向绑定，让开发者不用再操作dom对象，有更多的时间去思考业务逻辑。

本平台后端使用的是springboot.用springboot可以使配置变得更加的简单。并且springboot内嵌了servlet容器，降低了对环境的要求，机器有java运行环境，可以将项目打包成jar包，通过java命令 java -jar \*\*\*.jar 来执行。当然，springboot快速整合第三方框架，无需配置文件，使得代码少了、配置文件少了、不需要对第三方框架烦恼了、项目精简了，对整个团队的开发及维护来说，更大的节约了成本。它减少了大量的开发时间并提高了生产力。它避免了编写大量的样板代码，注释和XML配置。之外，Spring Boot应用程序与其Spring生态系统(如Spring JDBC，Spring ORM，Spring Data，Spring Security等)集成非常容易。

前后端分离的好处:

1.前端只需要独⽴编写客户端代码，后端也只需要独⽴编写服务端代码提供数据接⼝即可。

2.前端通过Ajax请求来访问后端的数据接⼝，将 Model展示到View 中即可。

3.前后端开发者只需要提前约定好接⼝⽂档（URL、参数、数据型…），然后分别独⽴开发即可，前端可以造假数据进⾏测试，完全不需要依赖于后端，最后完成前后端集成即可，真正实现了前后端应⽤的解耦合，极⼤地提升了开发效率。

本平台使用Mysql作为数据库，租用阿里云服务器部署mysql，用navicat作为云端数据库管理工具，便于管理和修改的同时更使实验贴近实践应用，我们小组成员都有过数据库的学习经验，能够熟练使用SQL语言，有过建数据库、建表、触发器编写等经验。

* 社会可行性

资源回收拥有巨大市场空间及交易规模，并且随着我国人民生活水平的提高，废旧 物品产生量也在剧增。根据一份再生资源回收利用市场分析报告中写到，目前的中国再生资源回收企业有 5000 多家，回收加工处理工厂 3000 多家，年经营额在 6000 亿至 7000 亿元间。而在巨大的市场中，其内行业的链条非常长，环节参与者多而且混乱，规则不强。将每个环节的从业者都看做该行业的利益相关诉求者，那将是一件非常累人的事。所以对于互联网创业者来说，首先是理清产业链各层级的不同需求，找准切入点，挖掘行业机会。而相关回收平台的出现，对于相关从业人员的利益以及使用者的需求进行对接。

* 风险因素及对策

本项目采用O2O模式，其核心在于线上与线下的对接，而线上服务基于线下的使用者，若使用者未达到一定的规模，如论是使用者还是商品回收者，其服务都难一正常进行，因此，若此项目正常上线，其初期的使用者从0到1的阶段是最难以渡过的阶段，针对这一问题，我们使用了以下对策：

1.前期投入进行大量的宣传以及平台在初期将一部分收益由于补贴回收者以及使用者，积累初始用户。

2.定点测试，在初期时，可以在某个潜在市场较大，产品较容易推广的城市进行测试运行，测试推广与运行方式的模拟，期间进行调整，在那个区域取得较好的效果后将成熟的模式有点到面的推广。

* 可行性分析结论

本项目的资源回收平台为资源回收者、使用者以及资源回收厂商的平台构建，三者间节约了资源回收者以及使用者的资源和时间，减少了二者不必要的时间损失，精确化的交易以及模式的高效以及广泛性极大程度的满足了三方的需求。同时，O2O模式近些年来的急速发展对于线下传统行业链的冲击在资本的作用下避无可避，如社区买菜的传统行业被电商产业的挤压等等，对于这个空白的市场，资源回收平台的出现拥有者光明的前景。

## 3.2 成本分析

成本分析的步骤：1.软件规模估算 2..软件工作量估算 3.软件开发进度估算 4.软件成本估算 5.成本估算的风险分析 6.项目总报价

软件规模估算 ：采用软件功能点模型

软件工作量估算：采用COCOMO2模型

软件开发进度估算: 采用 COCOMO2模型

软件成本估算：

项目总估算成本（C）=开发费用估算（U）+系统集成费用估算（Z）+维护费用估算（W）+系统解决方案费用估算（F）

（1）开发费用U=开发工作量\*开发费用/人月

=50（人月）\*1.0（万元/人月）=50万元

（2）软件（系统）维护费/年W=开发费用U\*5%=50万\*5%=2.5万元

（3）系统集成费用Z=U 复杂程度系数a 优质系数

=50\*7%\*1.1=3.85万元

（4）解决方案费用F=软件开发费用 复杂程度系数b 优质系数 项目时间

=50 7% 1.1\*1.5=6.3525万元

项目总报价

项目总报价=项目总估算成本\*（1+项目利润+风险基金+税系数）%

=6.3525\*（1+10%+30%+0）=8.8935万元

## 3.3 系统需求

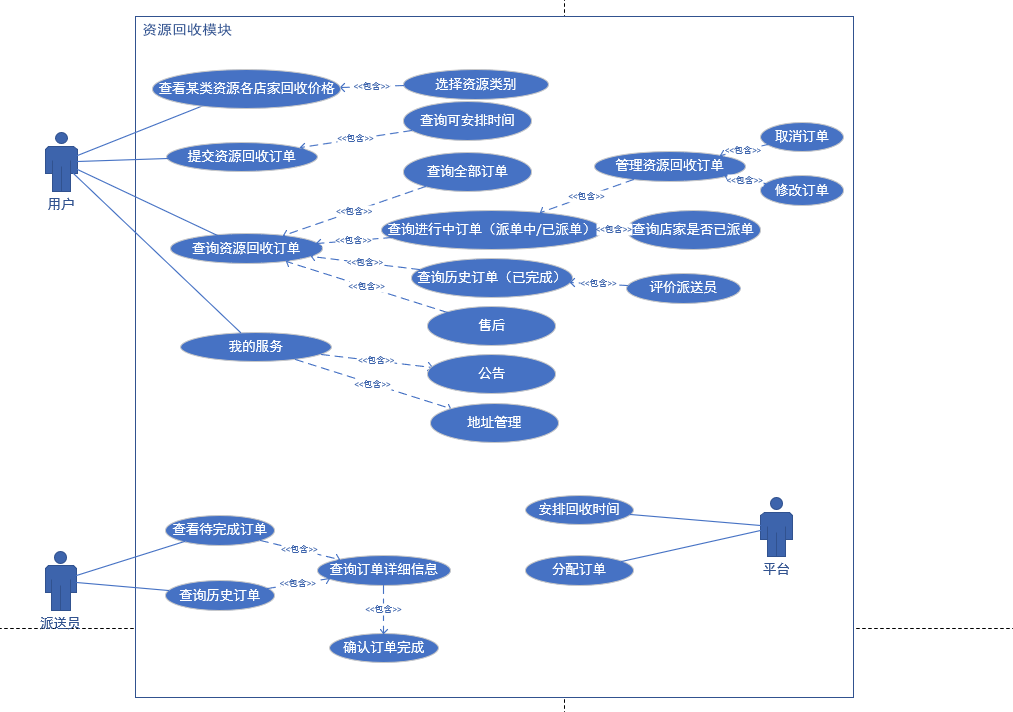


图3-1 资源回收模块用例图

**简要描述：**

资源回收用例使用户、平台、派送员可以对订单进行操作。

**按步骤描述：**

1. 用户查看资源和回收价格。
2. 用户选择资源类别并提交资源回收订单。
3. 用户可查看资源回收订单。
4. 平台根据订单安排回收时间，分配派送员。
5. 派送员接到订单，查看订单。
6. 派送员和用户交互，完成订单。

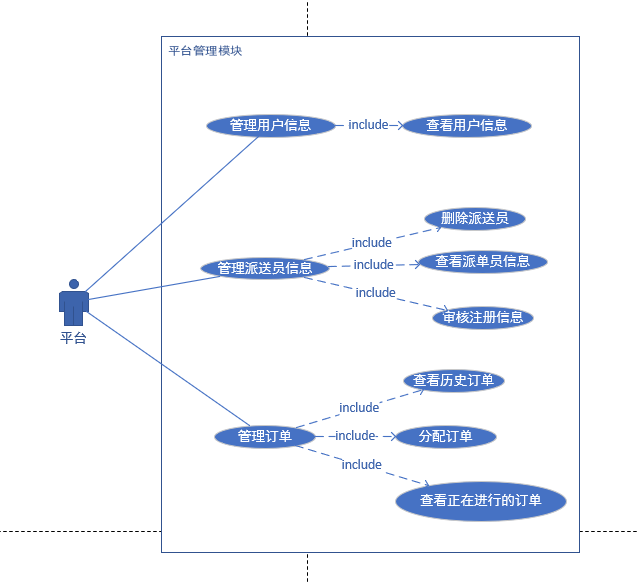


图3-2 平台管理模块用例图

**简要描述：**

平台管理用例使管理员可以管理各类信息。

**按步骤描述：**

1. 管理员查看用户、派送员、订单信息。
2. 管理员可删除、审核派送员信息。
3. 管理员分配订单。

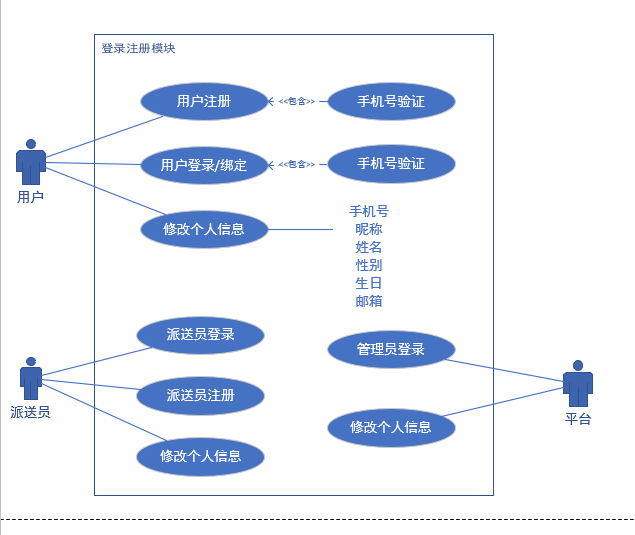


图3-3 登录注册模块用例图

**简要描述：**

登录注册用例使用户、管理员、派送员可以登录注册。

**按步骤描述：**

1. 用户注册并登录，可以修改个人信息。
2. 派送员注册并登录，可以修改个人信息。
3. 管理员登录，可以修改个人信息。

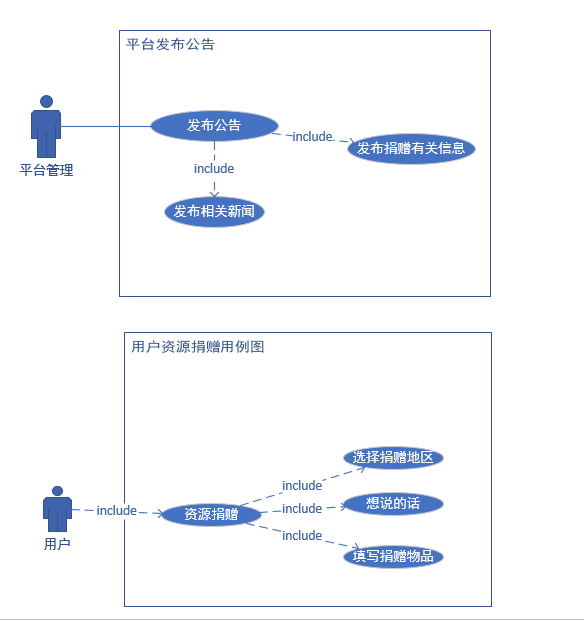


图3-4 平台发布公告用例图

**简要描述：**

平台发布公告用例使管理员发布公告。

**按步骤描述：**

1. 管理员编辑公告。
2. 管理员发布公告。

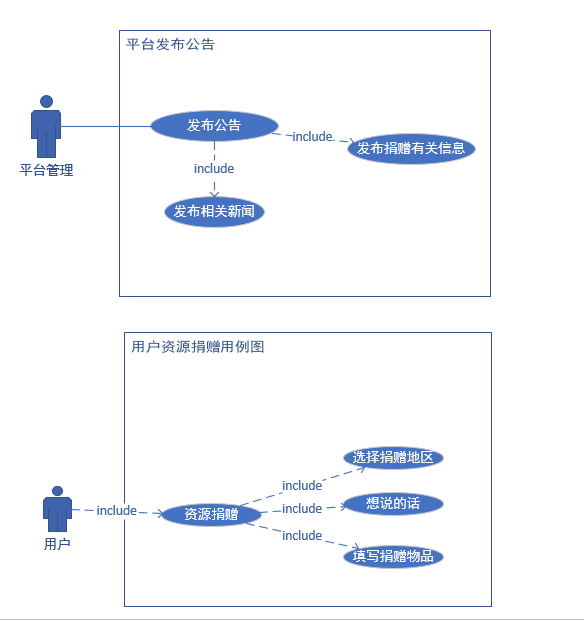


图3-5 用户资源捐赠用例图

**简要描述：**

用户资源捐赠用例使用户可以进行资源捐赠操作。

**按步骤描述：**

1. 用户进入资源捐赠页面。
2. 用户选择捐赠地区，编写想说的话，填写捐赠物品。
3. 用户提交订单，完成操作。

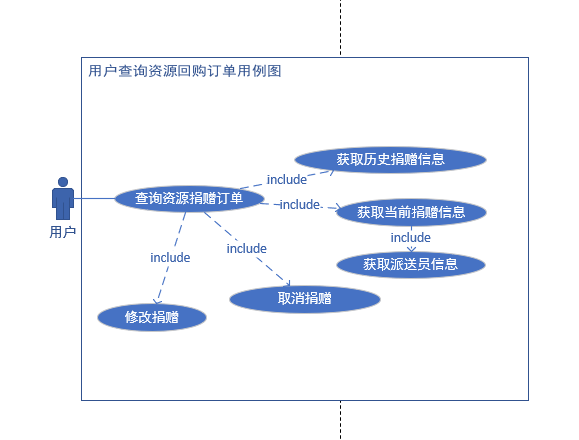


图3-6 用户查询资源捐赠订单用例图

**简要描述：**

用户查询资源捐赠用例使用户可以进行查看操作。

**按步骤描述：**

1. 用户进入查看捐赠订单页面。
2. 用户获得当前信息。
3. 用户可以进行修改取消操作。
4. 用户完成操作。

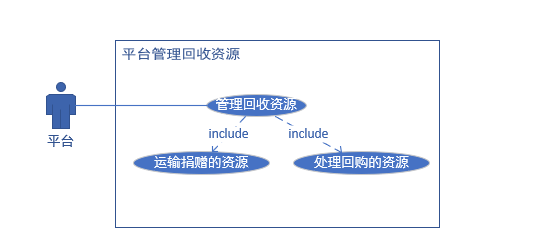


图3-7 平台管理回收资源用例图

**简要描述：**

平台管理回收资源用例使管理员可以管理回收资源。

**按步骤描述：**

1. 管理员管理回收资源。
2. 管理员进行运输捐赠操作。
3. 管理员进行处理回收资源操作。

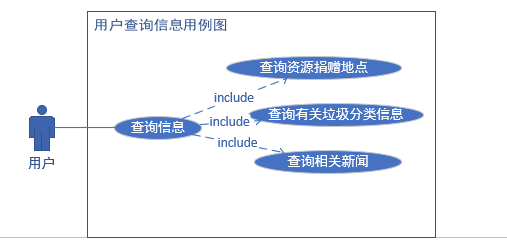


图3-8 用户查询信息用例图

**简要描述：**

用户查看信息用例使用户可以进行查询信息操作。

**按步骤描述：**

1. 用户进入查询信息页面。
2. 用户选择输入查询的内容。
3. 用户查询，完成操作。

# 传统的分析说明书

## 4.1 数据流图



图4-1 订单用例数据流图



图4-2 管理员登陆注册用例数据流图



图4-3 管理员管理用户用例数据流图



图4-4 管理员管理站点用例数据流图



图4-5 管理员管理订单用例数据流图



图4-6 用户注册登陆用例数据流图



图4-7 地址管理用例数据流图



图4-8 用户查询用例数据流图



图4-9 订单生成取消用例数据流图



图4-10 查看订单用例数据流图



图4-11 修改信息用例数据流图



图4-12 订单支付用例数据流图



图4-13 订单评价用例数据流图

## 4.2 数据字典

表 4-2-1 地址表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 地址ID | Address\_ID | INT | auto\_increment | √ | / | not null |
| 用户ID | User\_ID | INT | auto\_increment | / | √ | / |
| 地址细节 | Address\_Details | VARCHAR | 255 | / | / | / |
| 纬度 | Latitude | Double | / | / | / | / |
| 经度 | Longitude | Double | / | / | / | / |

表 4-2-2 管理员表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 管理员ID | Admin\_ID | INT | auto\_increment | √ | / | not null |
| 用户名 | User\_Name | VARCHAR | 20 | / | √ | / |
| 密码 | Password | VARCHAR | 20 | / | / | / |

表 4-2-3 回收员表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 回收员ID | Collector\_Person\_ID | INT | auto\_increment | √ | / | not null |
| 站点ID | Site\_ID | INT | / | / | √ | / |
| 真实姓名 | Collector\_Name | VARCHAR | 20 | / | / | / |
| 用户名 | User\_Name | VARCHAR | 20 | / | √ | / |
| 密码 | Password | VARCHAR | 20 | / | / | / |
| 手机号 | Phone | VARCHAR | 11 | / | / | / |
| ID卡 | IDcard\_Number | VARCHAR | 20 | / | / | / |
| 照片 | Photo | VARCHAR | 255 | / | / | / |
| 状态 | User\_Status | VARCHAR | 10 | / | / | / |

表 4-2-4 捐赠表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 捐赠ID | Donate\_ID | INT | auto\_increment | √ | / | not null |
| 帮助ID | Help\_ID | INT | / | / | √ | / |
| 用户ID | User\_ID | INT | / | / | √ | / |
| 回收员ID | Collector\_Person\_ID | INT | / | / | √ | / |
| 捐赠物品ID | Evaluation\_ID | INT | / | / | √ | / |
| 预定时间 | Scheduled\_Time | datetime | / | / | / | / |
| 完成时间 | Finished\_Time | datetime | / | / | / | / |
| 捐赠细节 | Donate\_Detail | VARCHAR | 255 | / | / | / |

表 4-2-5 评价表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 评价ID | Evaluation\_ID | INT | auto\_increment | √ | / | not null |
| 评价细节 | Evaluation\_Details | VARCHAR | 255 | / | / | / |
| 评价分数 | Evaluation\_Score | INT | / | / | / | / |

表 4-2-6 扶贫表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 帮助ID | Help\_ID | INT | auto\_increment | √ | / | not null |
| 帮助名字 | Help\_Name | VARCHAR | 20 | / | / | / |
| 纬度 | Longitude | Double | / | / | / | / |
| 经度 | Latitude | Double | / | / | / | / |
| 帮助细节 | Help\_Detail | VARCHAR | 255 | / | / | / |
| 帮助状态 | Help\_Status | VARCHAR | 10 | / | / | / |

表 4-2-7 项目表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 项目ID | Item\_ID | INT | auto\_increment | √ | / | not null |
| 项目种类ID | Item\_Type\_ID | INT | / | / | √ | / |
| 项目名字 | Item\_Name | VARCHAR | 20 | / | / | / |
| 项目价格 | Item\_Price | DOUBLE | / | / | / | / |
| 项目照片 | Item\_Pic | VARCHAR | 255 | / | / | / |

表 4-2-8 项目种类表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 项目种类ID | Item\_Type\_ID | INT | auto\_increment | √ | / | not null |
| 项目种类名字 | Item\_Type\_Name | VARCHAR | 20 | / | / | / |

表 4-2-9 回收订单详情表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 回收订单详情ID | Recycle\_Order\_Detail\_ID | INT | auto\_increment | √ | / | not null |
| 项目ID | Item\_ID | INT | / | / | √ | not null |
| 回收订单ID | Recycle\_Order\_ID | INT | / | / | √ | not null |
| 数量 | Quantity | DOUBLE | / | / | / | / |

表 4-2-10 回收订单表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 回收订单ID | Recycle\_Order\_ID | INT | auto\_increment | √ | / | not null |
| 地址ID | Address\_ID | INT | / | / | √ | / |
| 回收员ID | Collector\_Person\_ID | INT | / | / | √ | / |
| 评价ID | Evauation\_ID | INT | / | / | √ | / |
| 用户ID | User\_ID | INT | / | / | √ | / |
| 预定时间 | Scheduled\_Time | Datetime | / | / | / | / |
| 完成时间 | Finished\_Time | Datetime | / | / | / | / |
| 总金额 | Total\_Amount | double | / | / | / | / |

表 4-2-11 回收站点表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 站点ID | Site\_ID | INT | auto\_increment | √ | / | not null |
| 站点名字 | Address\_ID | VARCHAR | 255 | / | / | / |
| 纬度 | Longtitude | Double | / | / | / | / |
| 经度 | Latitude | double | / | / | / | / |

表 4-2-12 用户表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 数据项名 | 英文名 | 数据类型 | 长度 | 实体完整性约束 | 参照完整性约束 | 用户自定义性约束 |
| 用户ID | User\_ID | INT | auto\_increment | √ | / | not null |
| 用户名 | User\_Name | VARCHAR | 20 | / | / | not null |
| 真实姓名 | User\_RealName | VARCHAR | 20 | / | / | / |
| 密码 | Password | VARCHAR | 20 | / | / | / |
| 电话 | Phone | VARCHAR | 11 | / | / | / |

## 4.3 ER图



图4-14 O2O平台ER图

# 面向对象的分析说明书

## 5.1 接口类图

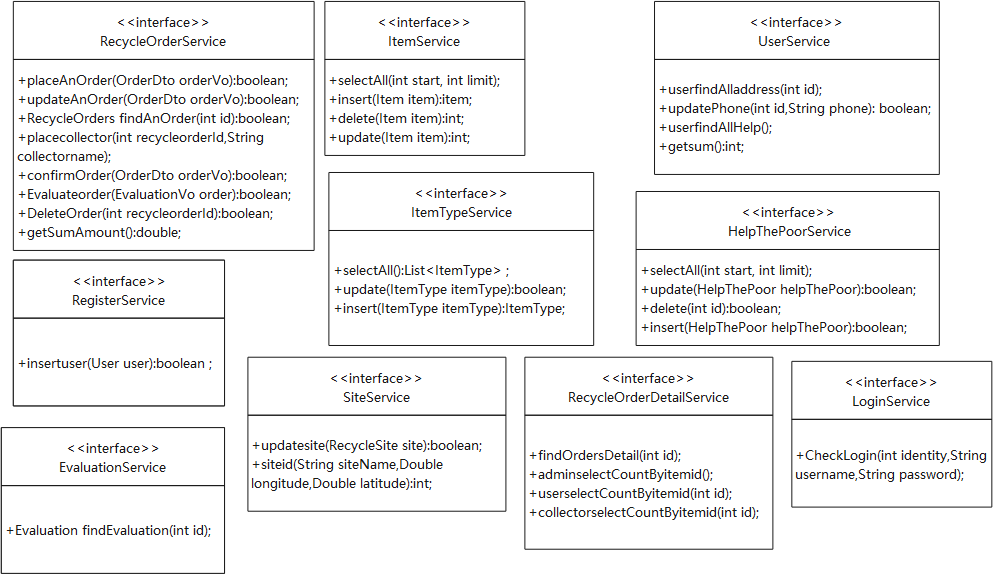


图5-1 接口类图

## 5.2 总体用例类图

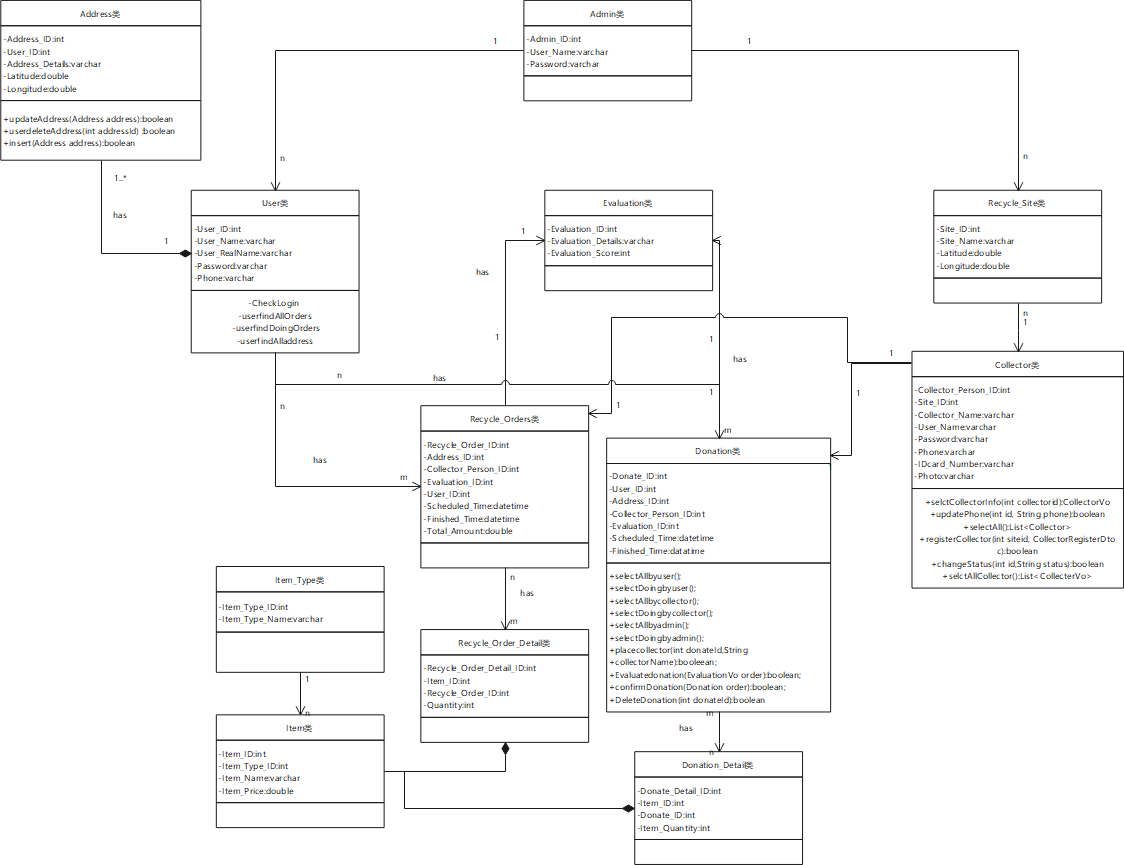


图5-2总体类图关系

## 5.3顺序图

### 5.3.1 登陆注册模块

用户登陆注册：



图5-3 用户注册顺序图



图5-4用户登录顺序图

废品回收员注册：



图5-5 废品回收员注册顺序图



图5-6 废品回收员登陆顺序图

### 5.3.2回收订单交易模块



图5-7 回收订单交易顺序图

### 5.3.3 捐赠模块



图5-8 捐赠订单顺序图

### 5.3.4 取消订单模块



图5-9 取消订单顺序图

# 面向对象的设计说明书

## 6.1 总体类图关系

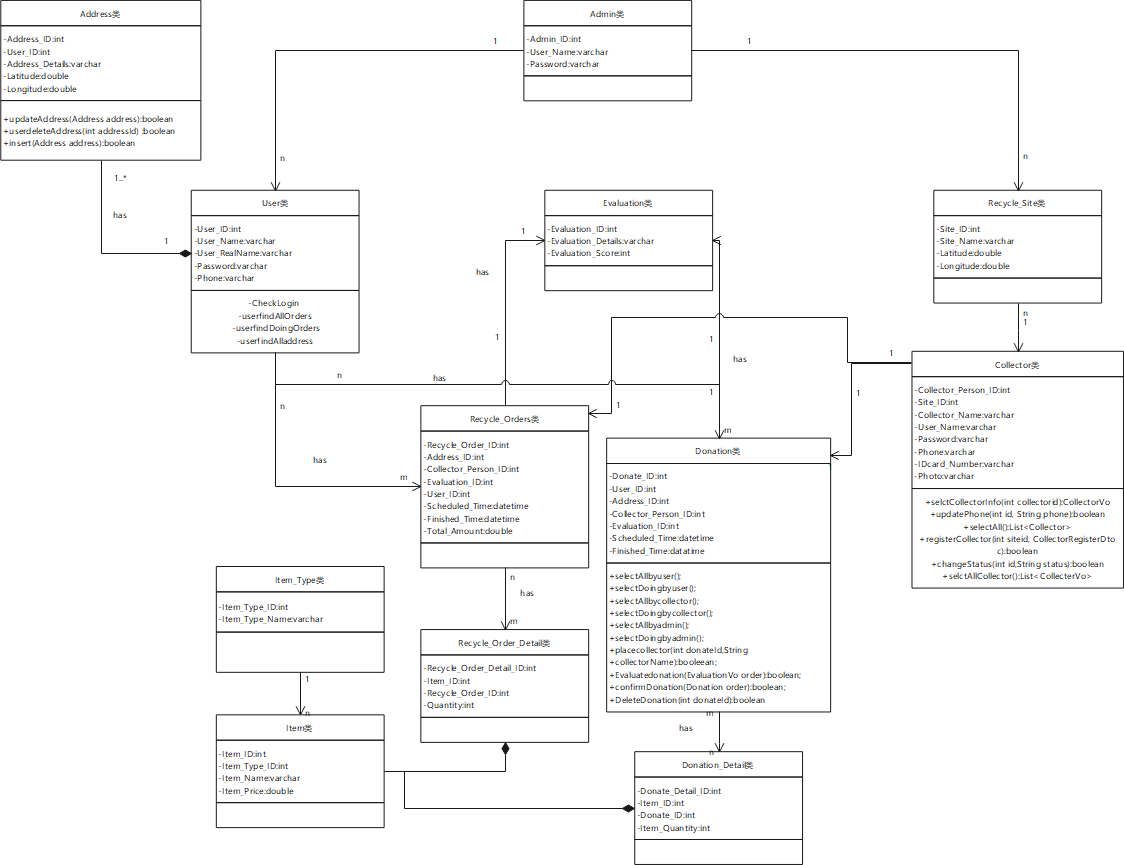


图6-1总体类图具体关系

## 6.2 详细类图关系

### 6.2.1 具体类图详细设计

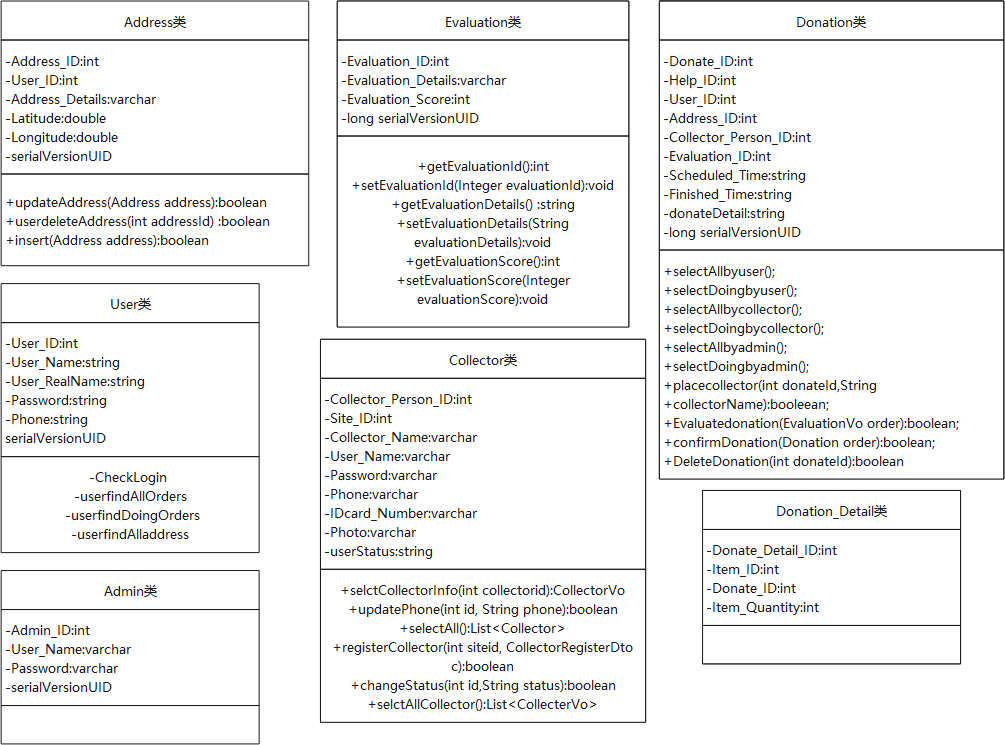


图6-2 用例类图具体定义（部分）

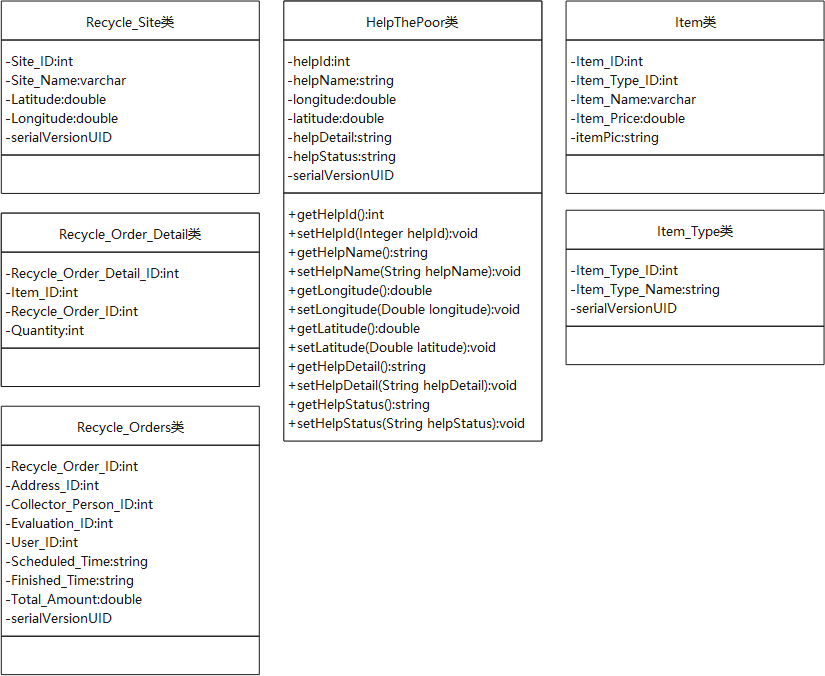


图6-3 用例类图具体定义（续）

### 6.2.2具体用例类图及详细设计

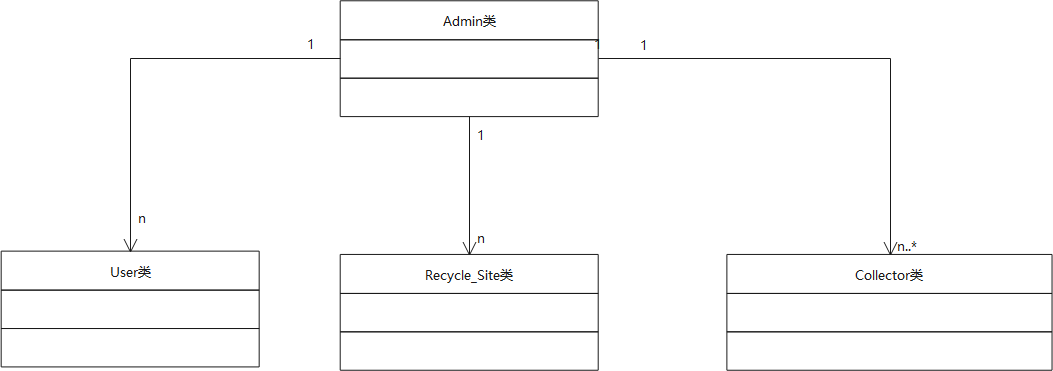


图6-4 管理员管理用例类图

PDL:

//用于管理员对派送员状态的更改(业务逻辑：管理员更改派送员的状态为T或者F，分别为启用和未启用的状态)

@Override

public boolean changeStatus(int id,String status) {

collectorDao.updateStatusByPrimaykey(id,status);

return true;

}

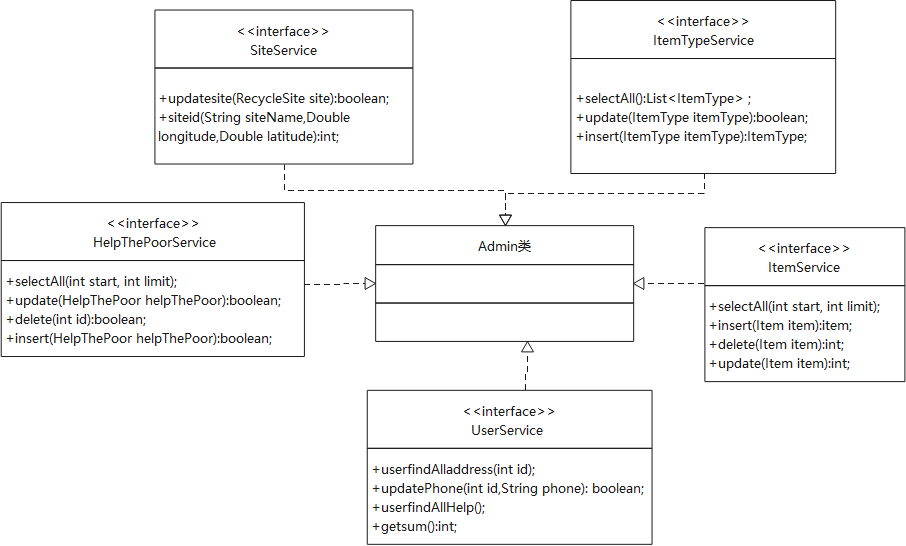


图6-5 管理员管理数据用例关系

PDL:

//用于管理员更新扶贫信息表

@Override

public boolean update(HelpThePoor helpThePoor) {

helpThePoorDao.updateByPrimaryKey(helpThePoor);

return true;

}

//用于管理员删除扶贫信息表

@Override

public boolean delete(int id) {

helpThePoorDao.deleteByPrimaryKey(id);

return true;

}

//用于管理员添加扶贫信息表

@Override

public boolean insert(HelpThePoor helpThePoor) {

helpThePoorDao.insert(helpThePoor);

return true;

}

//用于添加一个item记录

@Override

public Item insert(Item item){

itemDao.insert(item);

Item result = itemDao.selectByPrimaryKey(item.getItemId());

return result;

}

//用于删除一个item记录

@Override

public int delete(Item item){

int result=itemDao.deleteByPrimaryKey(item.getItemId());

return result;

}

//用于更新一个item记录

@Override

public int update(Item item){

int result=itemDao.updateByPrimaryKey(item);

return result;

}

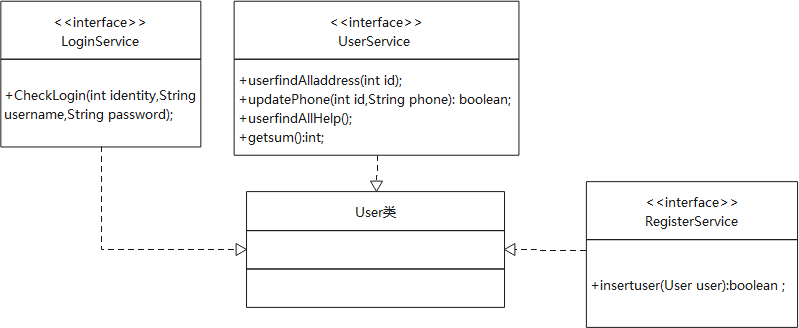


图6-6 用户登陆注册用例关系

PDL：

//用于用户的注册(业务逻辑：用户表中添加一个记录)

@Override

public boolean insertuser(User user) {

User u=UserName());

if(u!=null){

return false;

}

else{

insert(user);

return true;

}

}

//查询出当前用户的所有地址信息(业务逻辑：根据用户的userid查询其相应的地址信息)

@Override

public UserVo userfindAlladdress(int id) {

UserVo userVo = selectAddress(id);

return userVo;

}

//检查登录(业务逻辑：

// 1.用户根据输入的用户名与密码去数据库中查询有无相关记录，若输入正确则返回login为true，并且返回userid和username给前端

// 2.派送员根据输入的用户名与密码去数据库中进行查询，同时查询其启用信息，若是启用为F，意味着管理员还未审核，若输入正确并且启用为T，怎返回login为true

// 3.管理员根据输入的用户名与密码去数据库中进行查询，若输入正确在返回true，返回adminid和adminname给前端)

@Override

public LoginMsg CheckLogin(int identity, String username, String password) {

if(identity==1){

User user= selectByUserName(username);

if(user!=null&&user.getPassword().equals(password)){

用户登录

}

else {

setLogin(false);

}

}

else if(identity==2){

Collector collector = selectByUserName(username);

if (密码正确且审核完成) {

废品回收员登陆

} else if(还未审核完成){

msg.setName("审核中");

msg.setLogin(true);

}else {

msg.setLogin(false);

}

}

else if(identity==3) {

管理员登录

}else {

msg.setLogin(false);

}

}

return 登陆情况;

}

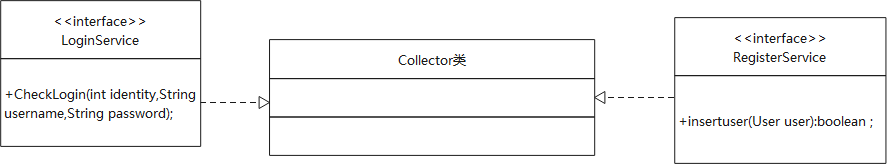


图6-7 废品回收员登陆注册用例关系

PDL:

//查询派送员的所有信息，包括他的站点名字信息

@Override

public CollectorVo selctCollectorInfo(int collectorid) {

CollectorVo collectorVo=collectorDao.selectRecyclesite(collectorid);

return collectorVo;

}

//派送员修改手机号

@Override

public boolean updatePhone(int id, String phone) {

collectorDao.updatePhoneByPrimaykey(id,phone);

return true;

}

//派送员的注册(业务逻辑：注册派送员时，设置派送员的状态为F，意义为未启用的状态，需要管理员审核之后将派送员设置为启用)

@Override

public boolean registerCollector(int siteid, CollectorRegisterDto c) {

collector.setSiteId(siteid);

collector.setUserName(c.getUserName());

collector.setCollectorName(c.getCollectorName());

collector.setPassword(c.getPassword());

collector.setPhone(c.getPhone());

collector.setIdcardNumber(c.getIdcardNumber());

collector.setUserStatus("F");

if(未注册过)

{

insert(collector);

return true;

}

else

return false;

}

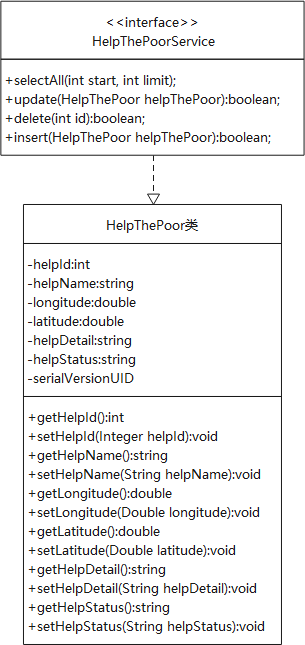


图6-8 扶贫数据管理用例关系

PDL:

//提供所有的扶贫信息给管理员，供管理员进行管理(业务逻辑：查询所有信息)

@Override

public PageInfo<HelpThePoor> selectAll(int start, int limit) {

PageHelper.startPage(start,limit);

Page<HelpThePoor> list= helpThePoorDao.selectAll();

PageInfo<HelpThePoor> result =new PageInfo<>(list);

return result;

}

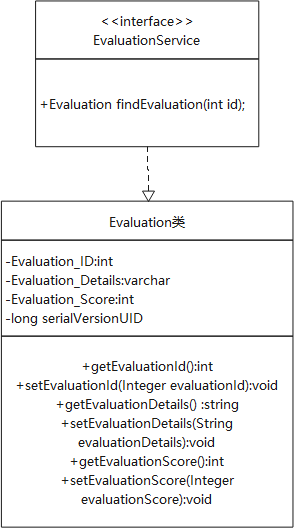


图6-9 评价数据用例关系

PDL:

//根据主键查询出某一个评价记录的信息

@Override

public Evaluation findEvaluation(int id) {

Evaluation result=evaluationDao.selectByPrimaryKey(id);

return result;

}

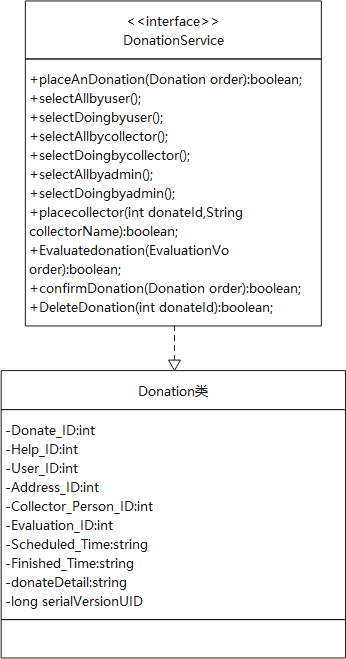


图6-10 捐赠数据用例关系

PDL:

//用户查询所有的捐赠信息(业务逻辑：根据用户提供的相关信息对所有的相关信息进行查询，运用动态sql，动态的根据用户的需求进行查询)

@Override

public PageInfo<DonationVo> selectAllbyuser(int userid, String collectorname, String datebyday, String datebymonth, String datepick, int start, int limit) {

PageHelper.startPage(start,limit);

if(datepick.equals("day")){

Page<DonationVo> list =donationDao.selectAllByuser(userid,collectorname,datebyday,datepick);

PageInfo<DonationVo> result= new PageInfo<>(list);

return result;

}else{

Page<DonationVo> list =donationDao.selectAllByuser(userid,collectorname,datebymonth,datepick);

PageInfo<DonationVo> result= new PageInfo<>(list);

return result;

}

}

//用户查询进行中捐赠信息(业务逻辑：根据用户提供的相关信息对所有的相关信息进行查询，运用动态sql，动态的根据用户的需求进行查询，正在进行中的订单是根据捐赠订单表中的finished\_time为空的记录)

@Override

public PageInfo<DonationVo> selectDoingbyuser(int userid, String collectorname, String datebyday, String datebymonth, String datepick, int start, int limit) {

PageHelper.startPage(start,limit);

if(datepick.equals("day")){

Page<DonationVo> list =donationDao.selectDoingByuser(userid,collectorname,datebyday,datepick);

PageInfo<DonationVo> result= new PageInfo<>(list);

return result;

}else{

Page<DonationVo> list =donationDao.selectDoingByuser(userid,collectorname,datebymonth,datepick);

PageInfo<DonationVo> result= new PageInfo<>(list);

return result;

}

}

//管理员和回收员查询与用户类似

//用于删除一个捐赠订单(业务逻辑：删除一个捐赠订单表中的记录)

@Override

public Boolean DeleteDonation(int donateId) {

donationDao.deleteByPrimaryKey(donateId);

return true;

}

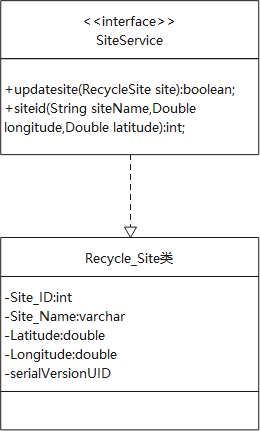


图6-11站点数据用例关系

PDL:

@Override

public boolean updatesite(RecycleSite site) {

updateByPrimaryKey(site);

return true;

}

//用于派送员注册时的相关信息(业务逻辑：根据派送员注册时填写的站点位置信息，先在站点表中新增一条记录，之后返回出自动生成的主键，用于更新派送员记录中的站点id)

@Override

public int siteid(String siteName, Double longitude, Double latitude) {

recycleSite.setSiteName(siteName);

recycleSite.setLatitude(latitude);

recycleSite.setLongitude(longitude);

recycleSiteDao.insert(recycleSite);

return recycleSite.getSiteId();

}

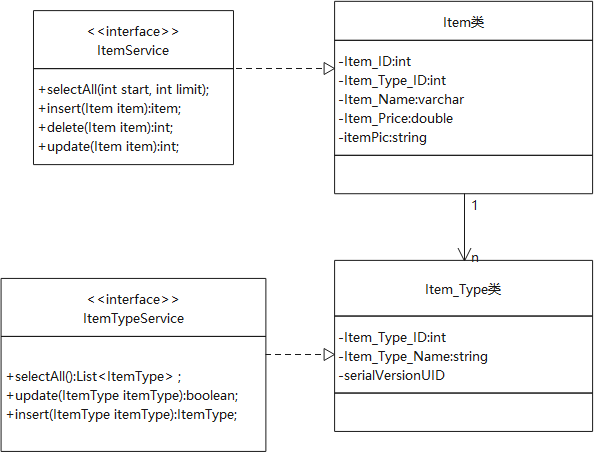


图6-12 物品种类数据用例关系

PDL:

//返回所有的Item信息给管理员(业务逻辑：查询所有的Item信息，同时包括Item属于的Item\_Type中的信息)

@Override

public PageInfo<ItemManageVo> selectAll(int start, int limit) {

PageHelper.startPage(start,limit);

Page<ItemManageVo> list = itemDao.selectAll();

PageInfo<ItemManageVo> result= new PageInfo<>(list);

return result;

}

//查询所有的itemtype信息并返回给管理员（业务逻辑:查询所有）

@Override

public List<ItemType> selectAll() {

List<ItemType> result= itemTypeDao.selectAll();

return result;

}

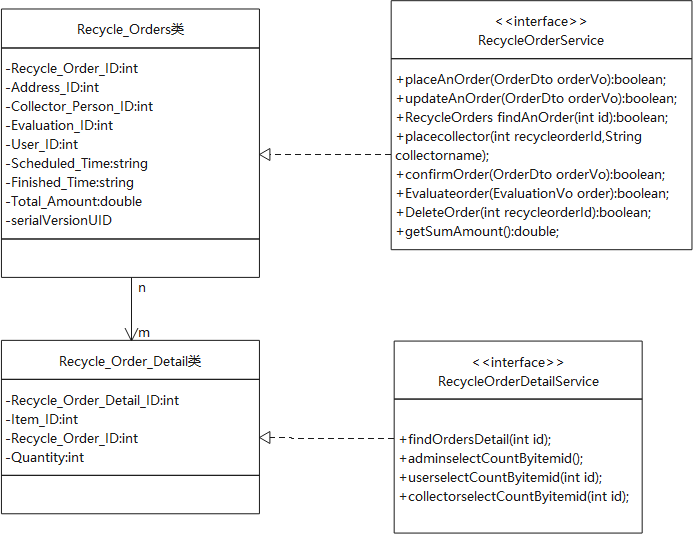


图6-13回收订单数据用例关系

PDL:

//根据主键查询所有的订单详情信息

@Override

public List<RecycleOrdersDetailVo> findOrdersDetail(int id) {

List<RecycleOrdersDetailVo> result = recycleOrderDetailDao.selectDetailByRecycleOrderId(id);

return result;

}

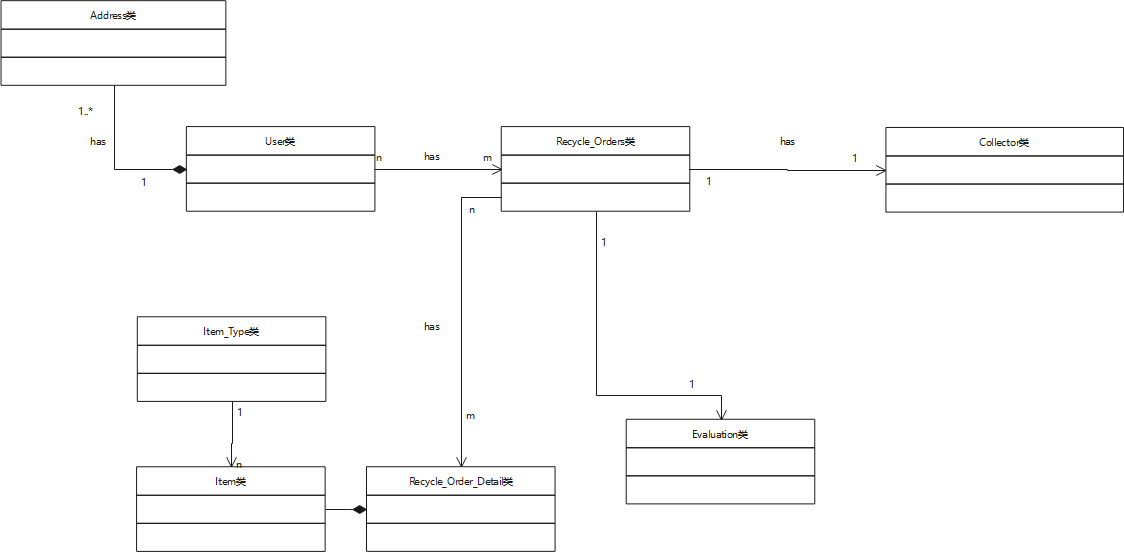


图6-14 用户回收订单用例类图

PDL:

//用户用于更新地址

@Override

public boolean updateAddress(Address address) {

addressDao.updateByPrimaryKey(address);

return true;

}

//用户用于删除地址(业务逻辑：将address表中的userid置空)

@Override

public boolean userdeleteAddress(int addressId) {

addressDao.deleteAddressByUser(addressId);

return true;

}

//用户用于增加地址

@Override

public boolean insert(Address address) {

addressDao.insert(address);

return true;

}

//用户查询所有的订单信息(业务逻辑：根据用户提供的相关信息对所有的订单信息进行查询，运用动态sql，动态的根据用户的需求进行查询)

public PageInfo<OrdersVo> userfindAllOrders(int userid, String collectorname, String datebyday, String datebymonth, String datepick, int start, int limit) {

PageHelper.startPage(start,limit);

if(datepick.equals("day")){

Page<OrdersVo> list =recycleOrdersDao.selectAllByuser(userid,collectorname,datebyday,datepick);

PageInfo<OrdersVo> result= new PageInfo<>(list);

return result;

}else{

Page<OrdersVo> list =recycleOrdersDao.selectAllByuser(userid,collectorname,datebymonth,datepick);

PageInfo<OrdersVo> result= new PageInfo<>(list);

return result;

}

}

//用户查询进行中订单信息(业务逻辑：根据用户提供的相关信息对所有的相关信息进行查询，运用动态sql，动态的根据用户的需求进行查询，正在进行中的订单是根据捐赠订单表中的finished\_time为空的记录)

@Override

public PageInfo<OrdersVo> userfindAllDoingOrders(int userid, String collectorname, String datebyday, String datebymonth, String datepick, int start, int limit) {

PageHelper.startPage(start,limit);

if(datepick.equals("day")){

Page<OrdersVo> list =recycleOrdersDao.selectDoingByuser(userid,collectorname,datebyday,datepick);

PageInfo<OrdersVo> result= new PageInfo<>(list);

return result;

}else{

Page<OrdersVo> list =recycleOrdersDao.selectDoingByuser(userid,collectorname,datebymonth,datepick);

PageInfo<OrdersVo> result= new PageInfo<>(list);

return result;

}

}

//管理员和回收员类似

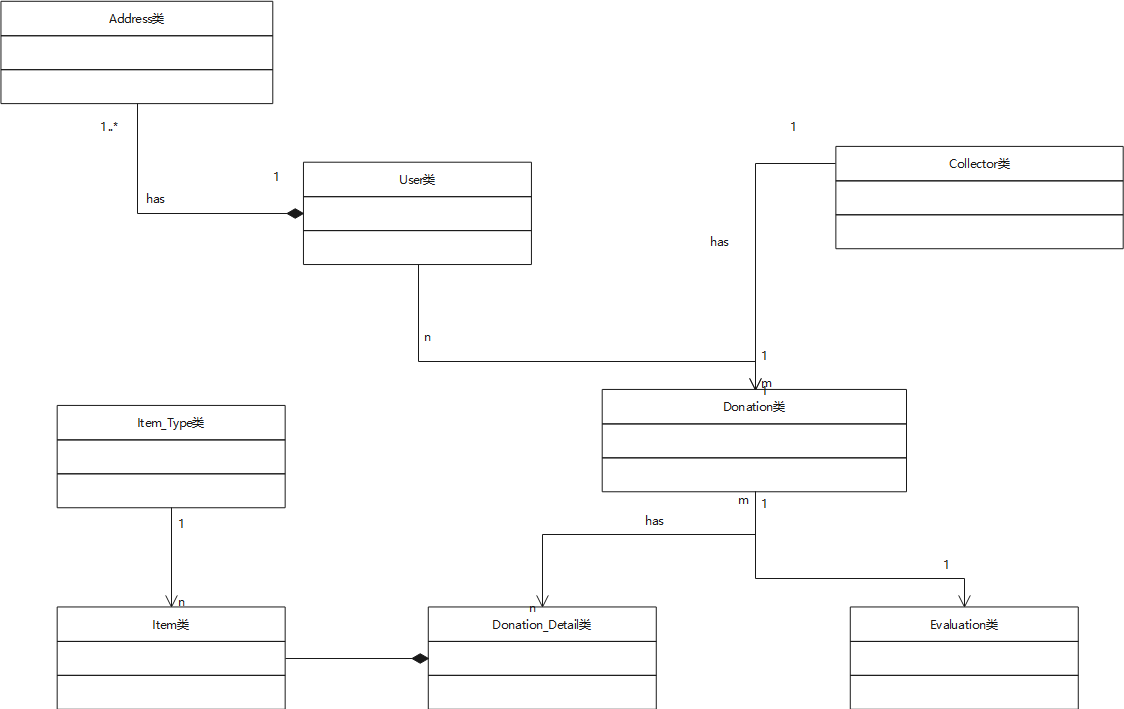


图6-15 用户捐赠订单用例类图

PDL:

//增加一个捐赠订单

public boolean placeAnDonation(Donation order) {

boolean result=false;

int i=donationDao.insert(order);

if(i>0){

result=true;

}

return result;

}

//用于管理员给捐赠订单安排一个派送员(业务逻辑：根据前端返回的派送员名字，查找到派送员的主键，再将它更新到捐赠订单表中)

@Override

public boolean placecollector(int donateId, String collectorName) {

int collectorid= selectBycollectorname(collectorName);

updatecollector(donateId,collectorid);

return true;

}

//用户对捐赠订单进行评价(业务逻辑：先将评价的相关信息添加到数据库表中，根据生成的主键号，更新捐赠订单表中的评价表外键信息)

@Override

public Boolean Evaluatedonation(EvaluationVo order) {

insert(order);

setDonateId(order.getDonateId());

setEvaluationId(order.getEvaluationId());

updateByPrimaryKeySelective(donation);

return true;

}

//派送员对捐赠订单进行确认(业务逻辑：派送员确认捐赠订单时，自动生成当前的时间，并传到后端，更新捐赠订单表中的finished\_time)

@Override

public Boolean confirmDonation(Donation order) {

updateByPrimaryKeySelective(order);

return true;

}

# 测试技术说明书

## 7.1 注册测试

用户/回收员账号的等价类

1.输入的字符串小于20位 可接受

2.少于一个字符 错误（不能为空）

3.大于20个字符 错误（名字太长）

4.最低限度（一个字符）的账号 可接受

用户/回收员密码的等价类

1.输入的字符串小于20位 可接受

2.少于一个字符 错误（不能为空）

3.大于20个字符 错误（密码太长）

4.最低限度（一个字符）的账号 可接受

用户/回收员电话的等价类

1.输入的数字串小于11位 错误（电话为11位）

2.少于一个字符 错误（不能为空）

3.大于11个数字 错误（电话为11位）

4.输入11位数字且符合电话号码规律 可接受

5.输入11位数字但不符合电话号码规律 错误（不符合规律）

## 7.2 登录测试

登录测试的等价类

1. 用户名输入别的内容 错误（不符合用户名）
2. 用户名和密码输入的不是相关联的 错误（密码和用户名不符合）
3. 用户名和密码匹配 可接受

## 7.3 用户下订单测试

用户下订单测试的等价类

1、地址格式输入大于255个字符 错误（地址太长）

2、不选择下单内容直接确认 错误（内容为空）

3、未登录下单 错误（未登录无用户信息为空）

4、不输入地址直接确认 可接受（有默认地址）

5、未确认下单 错误（不进行下单操作）

6、地址输入小于255字符且确认 可接受

7、物品数量小于1 错误（不可为空）

8、物品数量大于1 可接受

9、物品数量大于9999 可接受

## 7.4 回收员确认完成订单测试

确认完成订单测试的等价类

1、回收员未确认订单 错误（不确认订单不算完成）

2、回收员修改细节并确认订单 可接受

3、回收员修改细节未确认订单 错误（修改未传到后台管理）

4、回收员修改订单种类数量为0 可接受

5、回收员修改订单种类数量不变 可接受

## 7.5 评价测试

评价测试的等价类

1、评价内容大于255字符 错误（评价过长）

2、评价内容小于1个字符 可接受

3、评价内容小于255字符 可接受

4、评价打分小于20 错误（小于最低分数设置）

5、评价打分20/40/60/80/100 可接受

6、评价打分大于100 错误（大于最高分数设置）

7、未进行评价测试 可接受

## 7.6 管理员审核回收员

审核回收员的等价类

1、拒绝不合格的回收员 可接受

2、接受合格的回收员 可接受

3、停用不合格的回收员 可接受

## 7.7管理员修改、增加测试

管理员增加等价类

1、增加物品的细节字符大于255 错误（细节过长）

2、增加物品的细节字符小于1 可接受

3、增加物品的细节字符小于255 可接受

4、增加扶贫地区名字小于20 可接受

5、增加扶贫地区名字大于20 错误（名字过长）

6、增加扶贫地区名字小于1 错误（不能为空）

7、增加物品（/种类）名字符大于20 错误（名字过长）

8、增加物品（/种类）名字符小于20 可接受

9、增加物品（/种类）名字符小于1 错误（不能为空）

管理员修改等价类

1、修改物品的细节字符大于255 错误（细节过长）

2、修改物品的细节字符小于1 可接受

3、修改物品的细节字符小于255 可接受

4、修改扶贫地区名字小于20 可接受

5、修改扶贫地区名字大于20 错误（名字过长）

6、修改扶贫地区名字小于1 错误（不能为空）

7、修改物品（/种类）名字符大于20 错误（名字过长）

8、修改物品（/种类）名字符小于20 可接受

9、修改物品（/种类）名字符小于1 错误（不能为空）

10、修改修改1条数据 可接受

11、修改0条数据 可接受（被忽略）

12、修改2333条数据 可接受

# 用户使用说明书

本章分节详细描述系统的使用方法。具体应包含系统功能菜单的各项指令说明。同时附加以图示

# 做小组项目的体会与心得

## 9.1

可以结合企业捐教内容对项目的版本控制，部署，测试等。

# 参考文献

【1】

【2】

# 附录

## 附录A 第4小组资源回收O2O项目的源代码.rar

## 附录B 第4小组资源回收O2O项目的安装说明书.doc