目录

[引言 2](#_Toc281138104)

[编写目的 2](#_Toc281138105)

[背景 2](#_Toc281138106)

[定义 2](#_Toc281138107)

[参考资料 2](#_Toc281138108)

[各模块详细设计 3](#_Toc281138109)

[用户登录、注册模块 3](#_Toc281138110)

[碳计算器模块 4](#_Toc281138111)

[环保公益组织填写反馈模块 6](#_Toc281138112)

[个人主页模块 6](#_Toc281138113)

[查看购物历史模块 9](#_Toc281138114)

[反馈查看模块 12](#_Toc281138115)

[查看用户排名模块 14](#_Toc281138116)

# 引言

## 编写目的

此文档用于定义、设计本项目的详细设计，提供开发时的标准，及后期维护时的参考。

## 背景

软件名称：低碳交易平台

本项目的任务提出者：软件学院

开发者：温传博、罗伟锋、李明威、许强、张浩然

用户：大众、网上商铺卖家、环保公益组织

计算中心：172.29.132.248

## 定义

买家：在本平台中已经注册的用户，充当本平台中商品购买者的角色。

卖家：在本平台中已经注册的用户，充当本平台中商品销售者的角色。

游客：未在本平台注册，并游览 、使用本平台的人。

环保公益组织：在平台中已经注册的用户，在买卖交易过程中产生的捐赠款原则上捐赠给该环保公益组织，该组织亦有义务对捐赠作出反馈。

爱心树：游客或者买家在本平台中以购买此项作为捐赠环保公益组织的一种方式。

## 参考资料

需求规格说明书、概要设计说明书

# 各模块详细设计

## 用户登录、注册模块

* + 1. 程序结构

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 标识符 | 层次结构关系 | 说明 |
| 用户登录 | Login | 与其他程序平行 | 调用LINQ模块，根据用户名和密码判断是否登录成功 |
| 用户注册 | Register | 与其他程序平行 | 调用LINQ模块，注册新用户 |

* + 1. 程序描述

1.1.2.1用户登录

概述：

本程序从网页控件中获取登录信息，并判断是否登录成功，若成功，则转到个人主页，若不成功，提示用火狐从新输入。

性能：

1. 本程序的响应时间主要被数据库查询时间、网页读取时间占

据，预计响应时间不超过1s。

输入项：

Consumer 的ID， 密码

输出项：

用户的个人主页。

流程逻辑：

1. 用户在网页中输入用户名及密码；
2. 程序判断用户名和密码是否匹配；
3. 若匹配，则登陆成功，若不成功，则提醒用户。

接口：

1. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
2. 数据库接口：利用Linq的 context获取consumer表进行数据查询

测试计划：

输入数据：不同用户的ID；

预期结果：显示登录结果。

1.1.2.2 用户注册

概述：

本程序用于注册新用户。

性能：

本程序的响应时间主要被数据库查询时间、网页读取时间占

据，预计响应时间不超过1s。

输入项：

Consumer 的ID， 密码， 邮箱

输出项：

个人主页。

流程逻辑：

1. 用户在网页中输入用户名及密码、邮箱；
2. 判断输入项是否合法；
3. 若匹配，则注册成功，若不成功，则提醒用户，重新输入。

接口：

数据库接口：利用Linq的 context获取consumer表进行数据查询

测试计划：

输入数据：新的用户名、密码、邮箱

预期结果：显示注册结果。

## 碳计算器模块

* + 1. 程序结构

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 标识符 | 层次结构关系 | 说明 |
| 计算碳足迹 | CalculateCarbon | 与其他程序平行 | 在后台中根据现有的碳足迹算法，计算碳足迹含量。 |

* + 1. 程序描述

2.1.2.1计算碳足迹

概述：

本程序从网页控件中获取用户的输入信息，根据现有的成型碳足迹计算方法，计算用户的碳足迹。

性能：

1. 本程序的响应时间主要被数据库查询时间、网页读取时间占

据，预计响应时间不超过1s。

输入项：

|  |
| --- |
| 输入项 |
| 家庭人数 |
| 消耗电能 |
| 天然气 |
| 热力用燃油 |
| 煤炭 |
| 液化气 |
| 木屑 |
| 航班类型 |
| 出发地 |
| 目的地 |
| 舱位 |
| 私家车里程数 |
| 功率 |
| 摩托车里程数 |
| 功率 |
| 巴士里程 |
| 长途客车 |
| 国际铁路 |
| 国内 铁路 |
| 有轨电车 |
| 地铁 |
| Taxi |
| 食物偏好 |
| 包装 |
| 轿车拥有量 |

输出项：

计算结果。

流程逻辑：

1. 用户在网页中输入各种日常生活信息；
2. 程序判断用户名和密码是否匹配；
3. 若匹配，则登陆成功，若不成功，则提醒用户。

接口：

无

测试计划：

输入数据：各种参数

预期结果：相应的计算结果。

## 环保公益组织填写反馈模块

## 个人主页模块

* + 1. 程序结构

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 标识符 | 层次结构关系 | 说明 |
| 显示基本用户信息 | ShowConsumerInfo | 与其他程序平行 | 调用LINQ模块，根据consumerID获取单个消费者的信息，并进行显示。 |
| 修改用户密码 | modifyPassword | 与其他程序平行 | 调用LINQ模块，根据consumerID获取单个消费者的信息，进行密码修改。 |
| 修改用户邮箱 | modifyEmail | 与其他程序平行 | 调用LINQ模块，根据consumerID获取单个消费者的信息，进行邮箱修改。 |
| 提交订单编号 | SubmitTid | 与其他程序平行 | 用户填写tid后利用ajax将tid写入到session中，再跳转到淘宝用户验证页面 |

* + 1. 程序描述

2.3.2.1显示基本用户信息

概述：

本程序用于从数据库中读取买家详细信息，然后直接输出到相应的Label控件中，头像在数据库若为空则显示默认头像。

功能IPO图：



性能：

1. 本程序的响应时间主要被数据库查询时间、网页读取时间占

据，预计响应时间不超过1s。

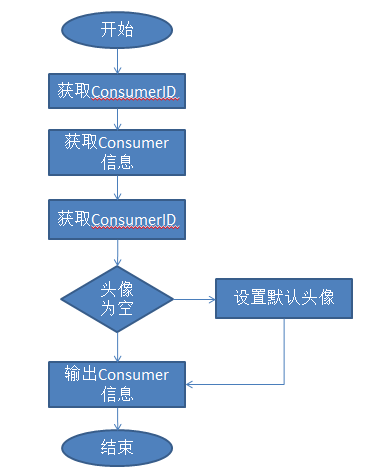
输入项：

Consumer 的ID。

输出项：

用户的详细信息。

流程逻辑：



接口：

1. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
2. 数据库接口：利用Linq的 context获取consumer表进行数据查询

测试计划：

输入数据：不同用户的ID；

预期结果：显示详细用户信息。

2.3.2.2 修改用户的密码和邮箱

概述：

本程序用于从数据库中读取买家详细信息，根据用户的需求进行

邮箱和密码的修改 。该过程较为简单故不进行详细叙述。

2.3.2.3 提交用户的交易号

概述：

本程序用于将用户的提交的Tid值写入到session之中，并且

打开新的页面用于用户权限验证。

功能IPO图：



性能：

本程序的响应时间主要被数据库查询时间、网页读取时间占

据，预计响应时间不超过1s。

输入项：

Tid值。

输出项：

用户权限验证页面。

测试计划：

输入数据：不同的Tid值；

预期结果：用户权限验证页面以及session的Tid值。

## 查看购物历史模块

* + 1. 程序结构

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 标识符 | 层次结构关系 | 说明 |
| 根据tid值添加购物信息并显示 | TopTidConfirm | 与其他程序平行 | 调用TopAPI验证用户的Tid值并获取交易信息，通过与数据库中的商品号对比判断其购买的是否该平台的产品，添加购物历史，并根据判断是否添加低碳树。 |
| 刷新购物历史状态并显示 | HistoryRefresh | 与其他程序平行 | 读取用户未付款的交易，通过TopAPI获取交易状态进行交易更新。并根据判断是否添加低碳树。 |

* + 1. 程序描述

2.4.2.1根据Tid值添加购物信息并显示

概述：

本程序调用TopAPI验证用户的Tid值并获取交易信息，通过与数据库中的商品号对比判断其购买的是否该平台的产品，以添加购物历史。若购买的为低碳树，交易状态为已付款，则低碳树条目。最终动态生成购物历史页面。

功能IPO图：



性能：

本程序的响应时间主要访问TopAPI获取交易信息、被数据库查询时间、网页读取时间占据，预计响应时间不超过2s。

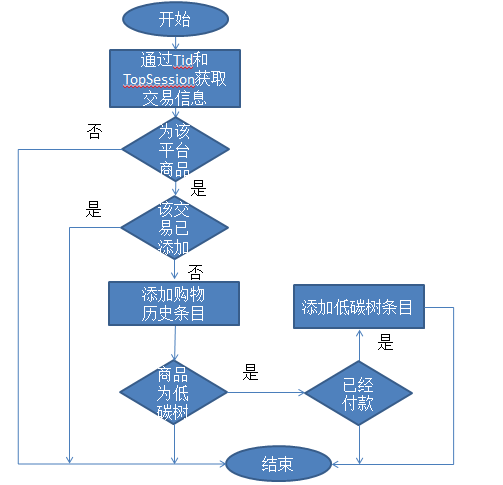
输入项：

交易编号Tid，用于用户授权的TopSession值。

输出项：

用户的历史购物详细购物信息。

流程逻辑：



接口：

1. 通过TopAPI获取具体的交易信息。
2. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
3. 通过Linq进行数据库查询。

测试计划：

输入数据：处于不同交易状态的Tid值以及TopSession；

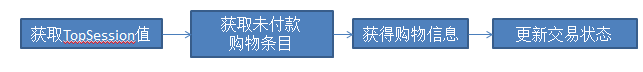
预期结果：显示详细用户购物信息。

2.4.2.2刷新购物历史并显示

概述：

读取用户未付款的交易，通过TopAPI获取交易状态进行交易更新。若更新内容为低碳树且更新状态为已付款状态则添加相关数量的低碳树条目。

功能IPO图：



性能：

本程序的响应时间主要被访问TopAPI获取交易信息、数据库查询时间、网页读取时间占据，预计响应时间不超过2s。

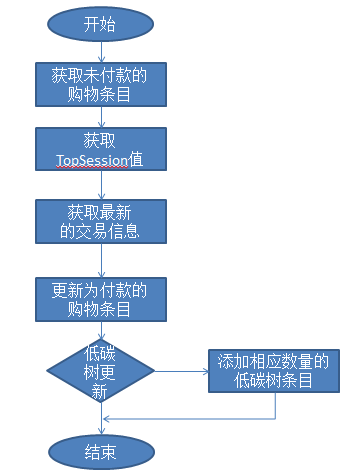
输入项：

用于用户授权的TopSession值。

输出项：

更新后用户的历史购物详细购物信息。

流程逻辑：



接口：

1. 通过TopAPI获取具体的交易信息。
2. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
3. 通过Linq进行数据库查询。

测试计划：

输入数据： 在不同的购物状态下输入TopSession；

预期结果：显示详细的刷新后的用户购物信息。

## 反馈查看模块

* + 1. 程序结构

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 标识符 | 层次结构关系 | 说明 |
| 显示指定用户低碳树列表 | showLowCarbonTrees | 与其他程序平行 | 调用LINQ模块，获得该用户的状态为1的低碳树列表,并生成动态网页进行显示。 |

* + 1. 程序描述

2.5.2.1显示指定用户程序描述

概述：

本程序用于从数据库中读取指定用户的状态为1的低碳树列表，，并在提取相关信息在Web网页中生成div列表显示。

功能IPO图：



性能：

本程序的响应时间主要被数据库查询时间、网页生成时间占据，预计响应时间不超过1s。

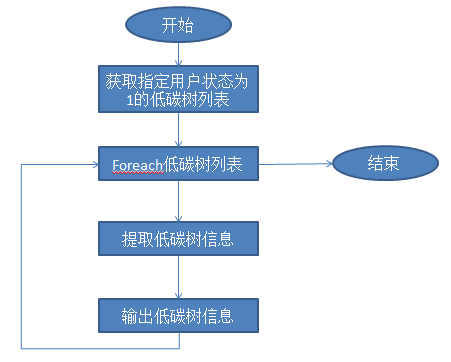
输入项：

Consumer的ID值

输出项：

Web网页：div列表，包括低碳树的各种详细信息。

流程逻辑：



接口：

1. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
2. 数据库接口：使用Linq中的context获取低碳树表，并进一步的提取状态为1的低碳树

测试计划：

输入数据：不同的ConsumerID；

预期结果：不同的用户对应的低碳树条目信息的Web网页显示。

## 查看用户排名模块

* + 1. 程序结构

|  |  |  |  |
| --- | --- | --- | --- |
| 程序名 | 标识符 | 层次结构关系 | 说明 |
| 生成消费活跃度排名 | createListByTrade | 与其他程序平行 | 调用LINQ模块，获得用户列表，并按照UserTradeTimes（消费次数）排名 |
| 生成环保积分排名 | createListByEnv | 与其他程序平行 | 调用LINQ模块，获得用户列表，并按照UserScore（环保积分）排名 |

* + 1. 程序描述

2.6.2.1生成消费活跃度排名程序描述

概述：

本程序用于从数据库中读取买家列表，并按照数据库的排序方式将买家列表按照消费活跃度进行排名，并在Web网页中生成div列表显示。

功能IPO图：



性能：

1. 本程序的响应时间主要被数据库查询时间、网页生成时间占据，预计响应时间不超过1s。
2. 本程序排序算法采用数据库的默认算法，递减排序，依据为消费活跃度，精度为int32。

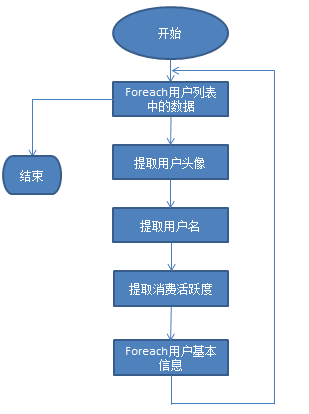
输入项：

数据库中数据：DataTable类型的用户列表及其消费活跃度。

输出项：

Web网页：div列表，包括：用户头像、用户名、性别、消费活跃度排名、环保积分。

流程逻辑：



接口：

1. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
2. 数据库接口：调用RankDBOperation.GetUserScoreRange()函数，获取DataTable类型的用户列表。

测试计划：

输入数据：虚拟的用户列表；

预期结果：已排序的用户列表，Web网页的显示。

2.6.2.2生成环保积分排名程序描述

概述：

本程序用于从数据库中读取买家列表，并按照.NET数据集的默认排序方式将买家列表按照环保积分进行排名，并在Web网页中生成div列表显示。

功能IPO图：



性能：

1. 本程序的响应时间主要被数据库查询时间、网页生成时间占据，预计响应时间不超过1s。
2. 本程序排序算法采用.NET的默认算法，递减排序，依据为环保积分，精度为int32。

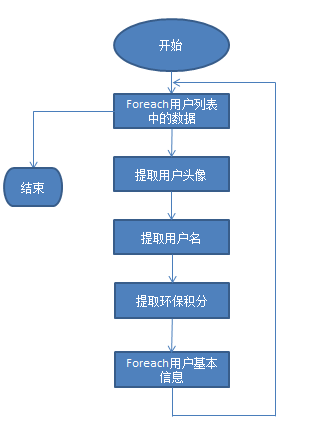
输入项：

数据库中数据：DataTable类型的用户列表及其消费活跃度。

输出项：

Web网页：div列表，包括：用户头像、用户名、性别、环保积分排名、消费活跃度。

流程逻辑：



接口：

1. 读取头像接口：ImageHandler.ashx文件，传入：from、id参数；
2. 排序算法，设置.NET数据集中的排序依据，v.Sort = "UserScore desc";

测试计划：

输入数据：虚拟的用户列表；

预期结果：已排序的用户列表，Web网页的显示。