项目名称： 拼拼——校园拼团代购

1. 系统概要

在疫情和学业的双重压力之下，学生对于探索北京美食的需求与难以长时间出校的现实相矛盾，基于此诞生了我们的创业项目——校园拼团代购。我们致力于帮助同学们足不出校尝遍北京美食。

基于这一校园创业项目我们自主开发系统，拥有面向用户和面向管理者两大主要模块，可以满足用户下单、信息维护、管理员查看报表等多种需求。通过在登陆界面进行区分为其提供服务。

1.1功能性需求

即系统必须完成哪些事、实现哪些功能，以及为了向其用户提供有用的功能所需执行的动作

本平台核心业务即为美食拼团，面向客户和采购配送人员两大主要用户。我们设计了针对两类人群的功能及业务。

下表为面向消费者的系统主要功能分析：从登录、注册到购物、支付。我们模拟电商消费场景下全流程业务需求：

|  |  |  |  |
| --- | --- | --- | --- |
| 面向消费者 | 主要功能 | 衍生启示 | 功能细节 |
|  | 注册 | 信息定义 | insert into |
|  |  |  | 昵称 |
|  |  |  | 电话号码（123） |
|  |  |  | 学院 |
|  |  |  | 所在地区（主校区（南门、东门）、东校区（。。。）、学苑） |
|  | 登录 |  | 遍历表进行匹配（电话号码、密码（123456）） |
|  | 区分 | 下拉列表 | （bs架构研究） |
|  | 查询自己的信息 | 个人中心 | 昵称 |
|  |  |  | 电话号码 |
|  |  |  | 学院（等信息） |
|  |  |  | 余额 |
|  | 编辑自己的信息 | 地址等 | 只允许更改所在地区 |
|  | 申请余额增加 |  | 增加多少钱，insert into number |
|  | 商品的选取 | 加入购物车 | 商品编号、商品表 |
|  |  | 商品价格制定 | 商品名称、商品价格 |
|  |  | 商品图片收集 | 基于商品采购价格，增加20% |
|  |  |  |  |
|  | 购物车的结算 | 支付 | 联系方式的填写（默认电话号码） |
|  |  |  | 为用户建订单表 |
|  |  |  | 订单表，订单编号、购买量、总价 |
|  | 支付按钮 |  | 弹窗花费多少元，剩余多少元 |
|  | 消费者余额（钱包） | 扣除余额 |  |
|  | 确认收货 | 订单状态 | 启动积分结算 |
|  | 积分机制 | 设计积分规则 | 积分1：1 |
|  | 历史订单查询 | 历史订单属性 | 订单编号 |
|  |  |  | 订单商品编号—>商品名称 |
|  |  |  | 商品数量 |
|  |  |  | 商品单价、订单总价格 |

下表为面向管理人员的系统主要功能分析：实现基本的注册和登录功能后。他们还需要满足对于订单的查询功能、获悉总支出和利润等财务数据的功能。

|  |  |  |
| --- | --- | --- |
| 面向管理人员（采购配送） | 主要功能 | 功能细节 |
|  | 注册 | 姓名 |
|  |  | 电话号码 |
|  |  | 员工号（001） |
|  | 登录 | 密码（123456） |
|  | 区分 | 下拉列表 |
|  | 查询未完成的订单 | 采购表（根据订单表，加总商品数量） |
|  |  | 商品单价->商品总价 |
|  |  | 需要去供应商的地方 |
|  | 查询总支出 | 总支出+运费10元每去一个供应商 |
|  | 总利润 | 面向消费者价格表 |
|  |  | 面向采购的价格表 |
|  |  | 总支出表 |
|  |  | 得到利润表 |
|  | 供应商信息表 | 供应商联系方式、地址、供应商号 |
|  | 采购商品信息表 | 商品信息、商品编号 |
|  | 员工表 | 工资 |
|  |  | 相关信息、员工号 |
|  | 为消费者充值 | 增加余额（余额表） |
|  |  | 浏览余额增加申请表 |
|  | 商品推荐 | 根据昨天的总销量进行展示图片 |
|  |  | max(商品量)，商品编号 |
|  |  | 建表（商品编号，对应图片编号） |

1.2非功能性需求

作为对功能性需求的补充，系统需求分析还包括非功能需求。主要包括系统使用时对性能方面的要求、运行环境要求。软件设计必须遵循的相关标准、规范、用户界面设计的具体细节、未来可能的扩充方案等。

本系统的本质仍然为电子商务平台，离不开客户购物订单数据的流通与存储。因此，系统对于数据存储便捷性、安全性，以及数据交流的灵活性要求较高。我们选择安全有保障、数据传输有效率的Oracle数据库作为本项目的数据库系统，选用的Oracle数据库版本为11g，虽然不是最新版，但是其基本功能已经能满足系统的日常运维数据需求。

除此之外，我们还使用Navicat数据库管理工具，帮助我们更方便快捷地调整、维护数据库。更方便了相关数据记录的插入与其他处理。灵活地使用工具和SQL语句，让我们的工作更轻松。

由于业务性质的特殊性，我们的数据库系统需要能够支持频繁地被消费者访问。并且由于业务流程，数据库系统在面向消费者的业务场景对于数据库的性能要求更高。频繁读取数据，并且经常写入数据库订单记录。

数据库稳定性、安全性都有要求。持续为用户提供服务，在出现bug后尽快修复；保障客户的订单数据及隐私，防止外部攻击致使数据库泄漏。

用户界面设计细节：在Winform窗体、Html界面作为用户访问的窗口，对用户友好程度极其重要。因此我们的界面要求美观、功能齐全，能够被用户轻松地识别相关功能。

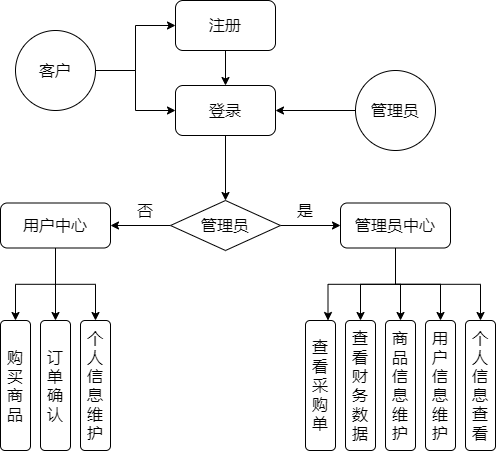
1.2设计约束

一般也称做设计限制条件，通常是对一些设计或实现方案的约束说明。

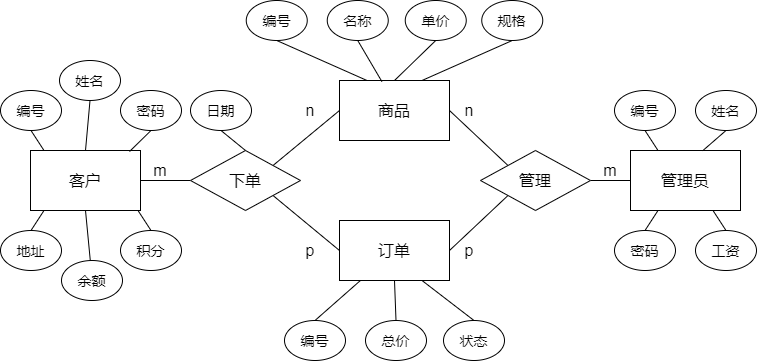
本系统要求待开发软件必须使用Oracle数据库系统完成数据管理功能；运行时必须要求连接网络，以在局域网内连接数据库，实现数据的共享。

2. 系统分析

2.1功能总览

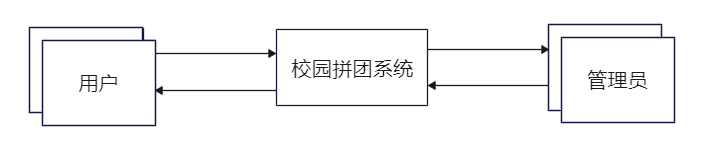


2.2E-R图

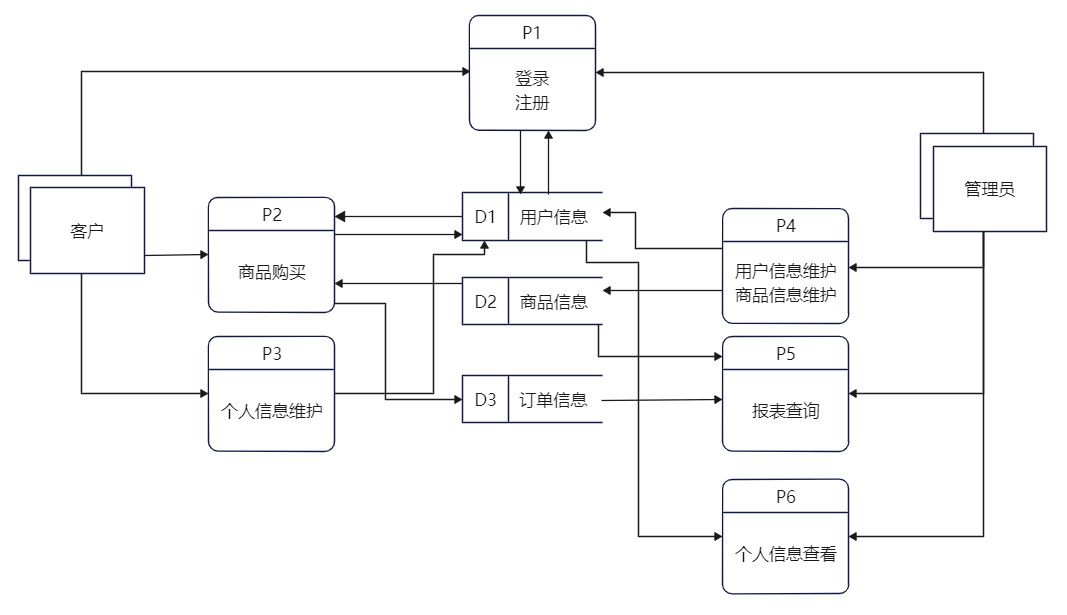


2.3数据流图（结构化分析）

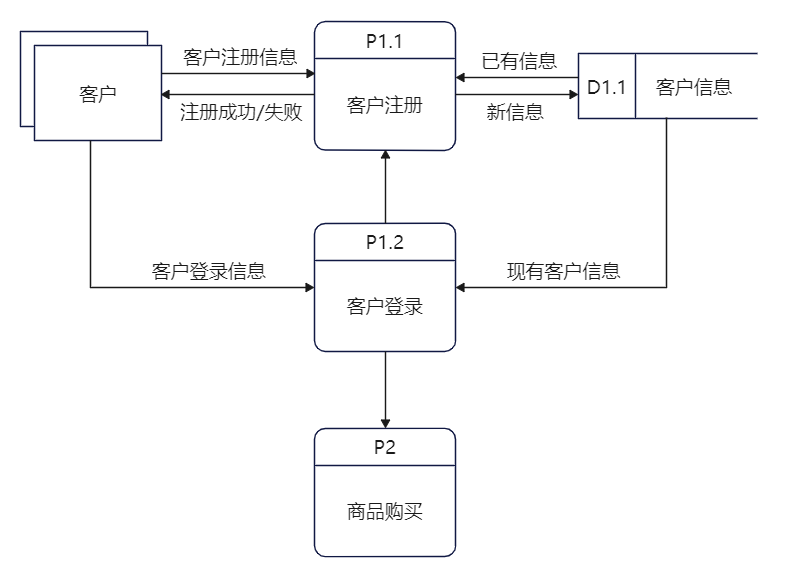
顶层

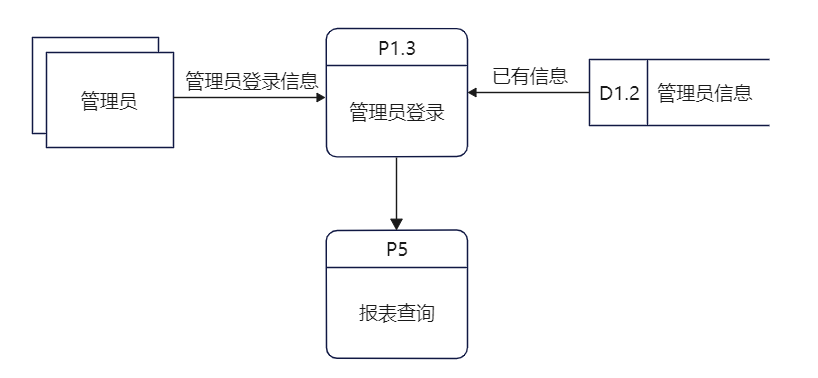


第一层

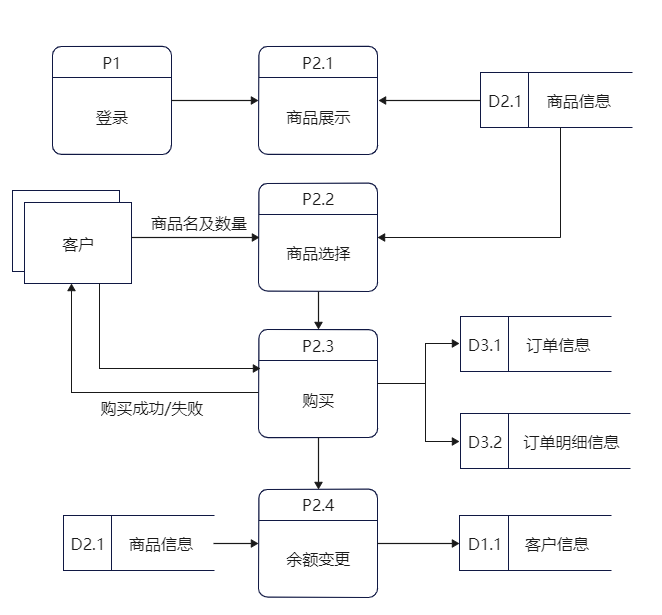


第二层-1

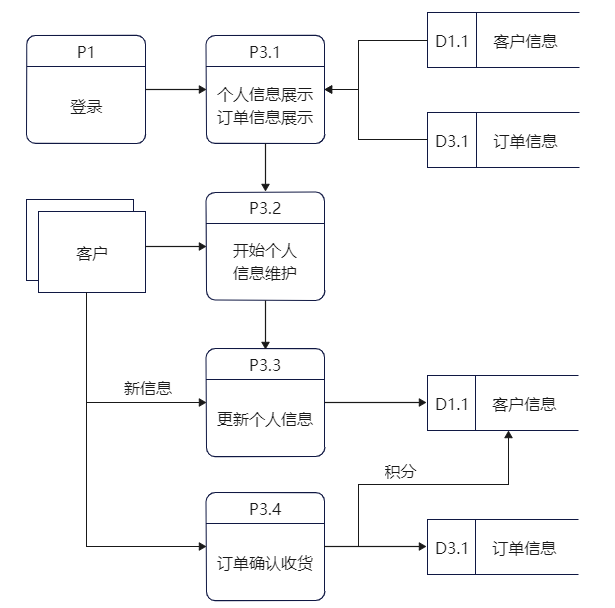




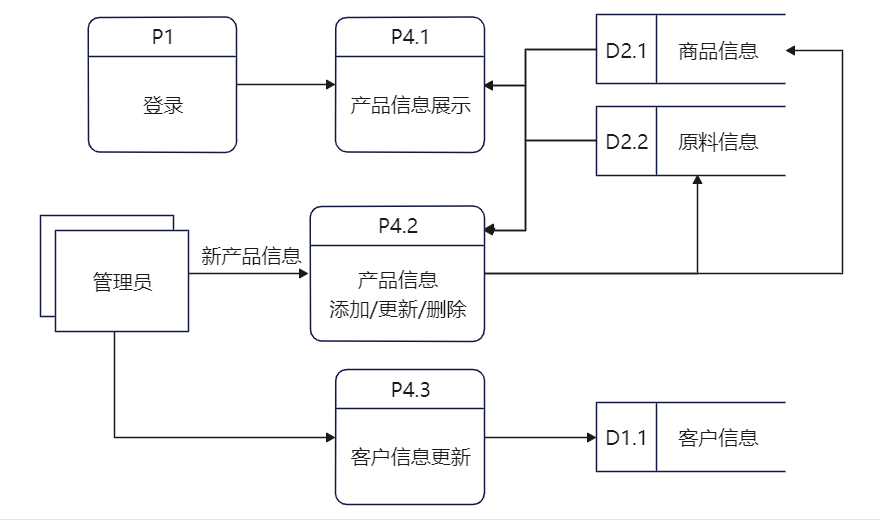
第二层-2



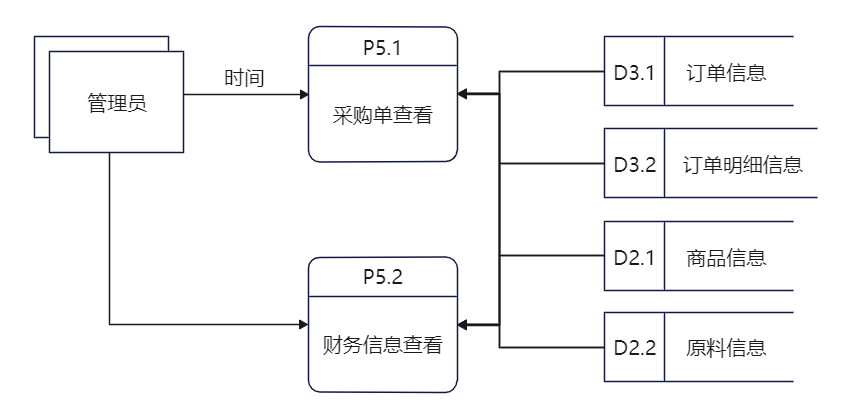
第三层-3



第二层-4



第二层-5



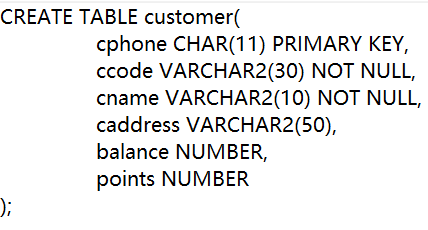
3 数据库设计

3.1表结构设计与表创建

根据系统需求设计客户表、产品表、订单表、订单明细表、原料表、采购表、职工表以及供应商表，共计8张表。

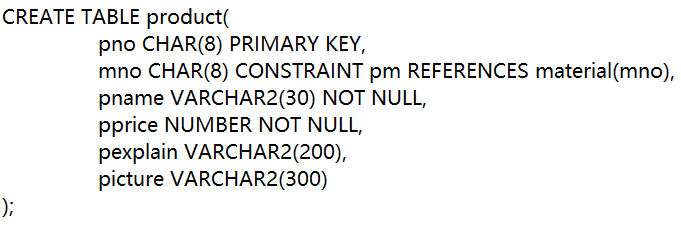
客户表包括手机号、密码、姓名、地址、余额及积分，共6个字段。主要用途为记录用户个人信息：

|  |  |  |  |
| --- | --- | --- | --- |
| 客户表-customer | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 手机号 | cphone | CHAR(11) | PRIMARY KEY |
| 密码 | ccode | VARCHAR2(30) | NOT NULL |
| 姓名/昵称 | cname | VARCHAR2(10) | NOT NULL |
| 地址 | caddress | VARCHAR2(50) |  |
| 余额 | balance | NUMBER |  |
| 积分 | points | NUMBER |  |



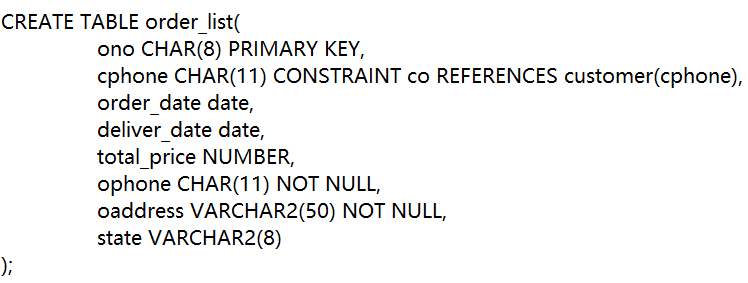
产品表包括产品号、原料号、产品名称、单价、规格说明及图片地址，共6个字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 产品表-product | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 产品号 | pno | CHAR(8) | PRIMARY KEY |
| 原料号 | mno | CHAR(8) | REFERENCES material(mno) |
| 产品名称 | pname | VARCHAR2(30) | NOT NULL |
| 单价 | pprice | NUMBER | NOT NULL |
| 规格说明 | pexplain | VARCHAR2(200) |  |
| 图片地址 | picture | VARCHAR2(300) |  |



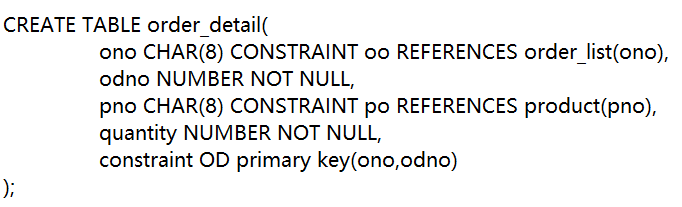
订单表包括订单号、客户号、下单日期、交付日期、总价、联系方式、送货地址及订单状态，共8个字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 订单表-order\_list | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 订单号 | ono | CHAR(8) | PRIMARY KEY |
| 客户号 | cphone | CHAR(11) | REFERENCES customer(cphone) |
| 下单日期 | order\_date | date |  |
| 交付日期 | deliver\_date | date |  |
| 总价 | total\_price | NUMBER |  |
| 联系方式 | ophone | CHAR(11) | NOT NULL |
| 送货地址 | oaddress | VARCHAR2(50) | NOT NULL |
| 状态 | state | VARCHAR2(8) |  |



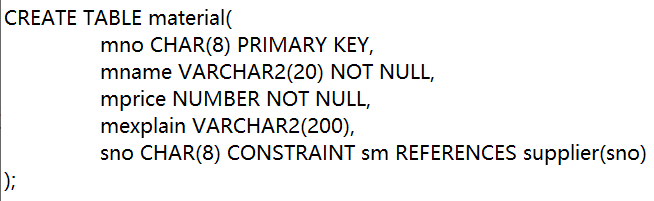
订单明细表包括订单号、序号、产品号及数量，共4个字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 订单明细表-order\_details | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 订单号 | ono | CHAR(8) | REFERENCES order\_list(ono)，PRIMARY KEY |
| 序号 | odno | NUMBER | PRIMARY KEY |
| 产品号 | pno | CHAR(8) | REFERENCES product(pno) |
| 数量 | quantity | NUMBER | NOT NULL |



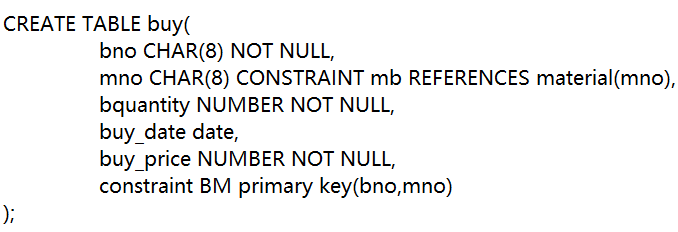
原料表包括原料号、原料名称、单价、规格说明及供应商号，共5个字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 原料表-material | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 原料号 | mno | CHAR(8) | PRIMARY KEY |
| 原料名称 | mname | VARCHAR2(20) | NOT NULL |
| 单价 | mprice | NUMBER | NOT NULL |
| 规格说明 | mexplain | VARCHAR2(200) |  |
| 供应商号 | sno | CHAR(8) | REFERENCES supplier(sno) |



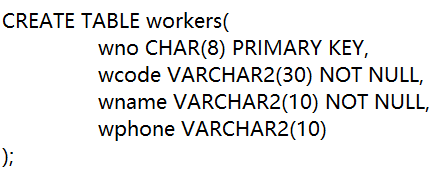
采购表包括采购编号、原料号、数量、采购日期及总计，共5个字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 采购表-buy | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 采购编号 | bno | CHAR(8) | PRIMARY KEY |
| 原料号 | mno | CHAR(8) | REFERENCES material(mno) |
| 数量 | bquantity | NUMBER | NOT NULL |
| 采购日期 | buy\_date | date |  |
| 总价 | buy\_price | NUMBER | NOT NULL |



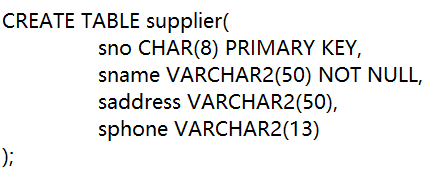
职工表包括职工号、密码、姓名及电话，共4个字段：

|  |  |  |  |
| --- | --- | --- | --- |
| 职工表-workers | | | |
|  | 字段名称 | 字段类型 | 说明 |
| 职工号 | wno | CHAR(8) | PRIMARY KEY |
| 密码 | wcode | VARCHAR2(30) | NOT NULL |
| 姓名/昵称 | wname | VARCHAR2(10) | NOT NULL |
| 电话 | wphone | VARCHAR2(10) |  |



供应商表包括供应商号、供应商名称、地址及联系方式，共4个字段：

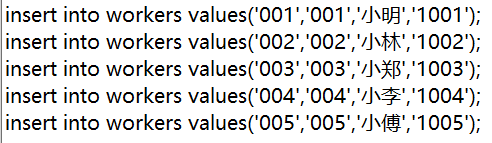
|  |  |  |  |
| --- | --- | --- | --- |
| 供应商表-supplier | | | |
|  | 字段名称 | 字段类型 | 其他 |
| 供应商号 | sno | CHAR(8) | PRIMARY KEY |
| 供应商名称 | sname | VARCHAR2(50) | NOT NULL |
| 地址 | saddress | VARCHAR2(50) |  |
| 联系方式 | sphone | CHAR(11) |  |



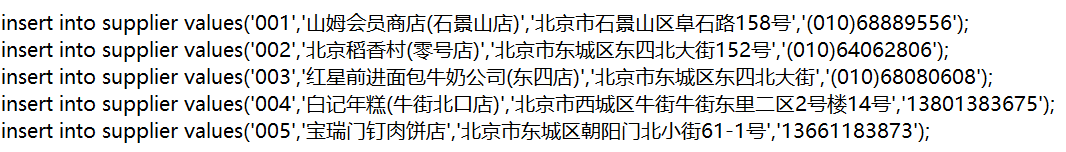
3.2部分表信息插入

由于系统设计中不具备职工信息与供应商信息维护功能，因此在数据库设计中为其插入初始数据：

插入职工信息：



插入供应商信息：

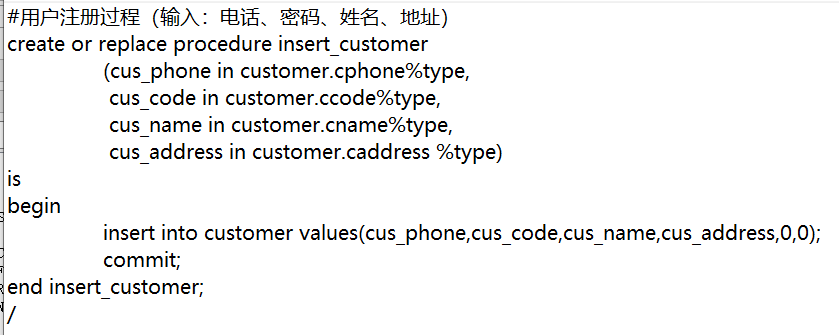


3.3过程设计

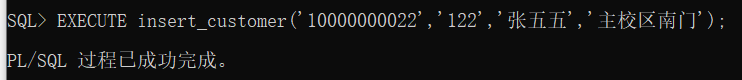
（1）用户注册过程 insert\_customer：

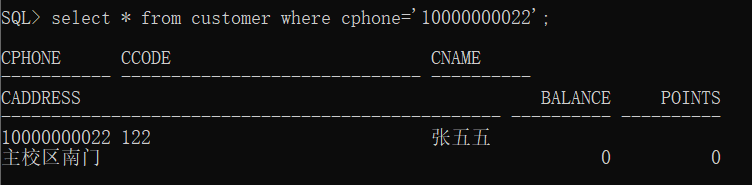
输入：电话、密码、姓名、地址

功能：将用户注册信息插入客户表，余额与积分默认为0



测试执行：

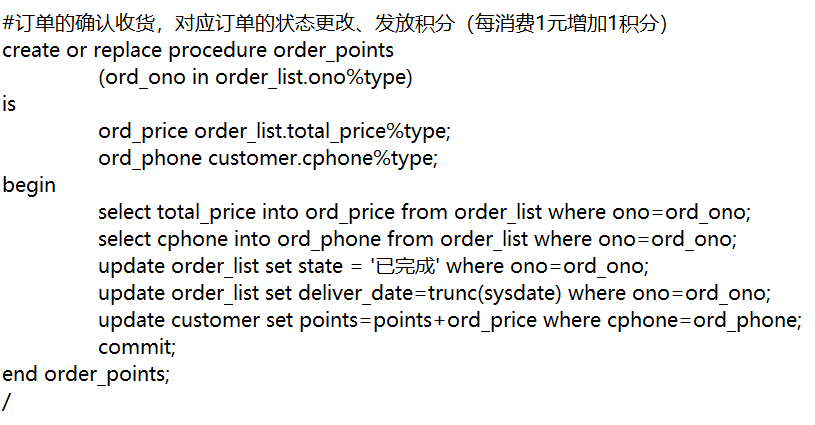




（2）订单收货 order\_points

输入：订单号

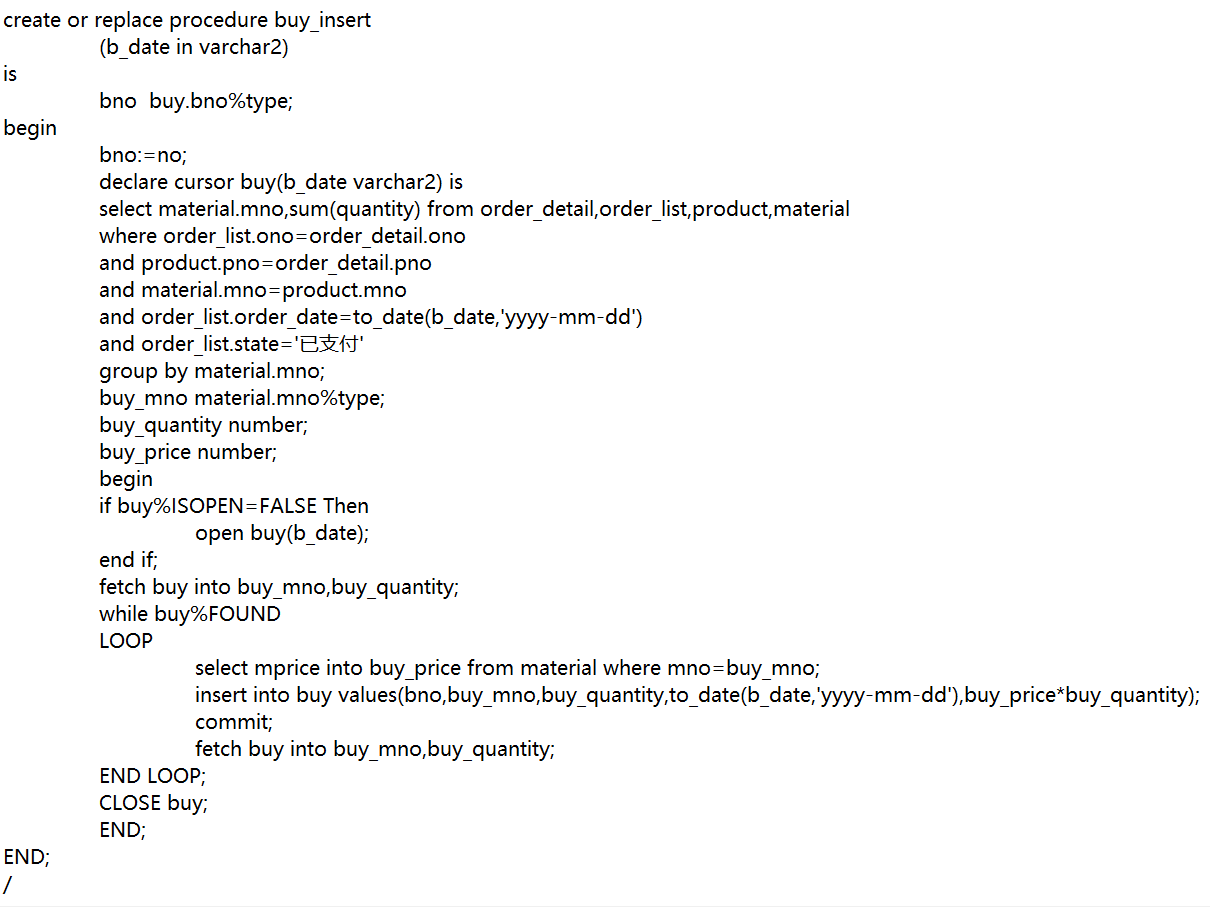
功能：将订单状态更改为“已完成”；收货时间更新为当前系统时间；为用户增加积分



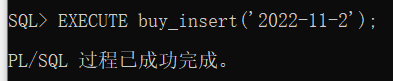
（3）过程——生成某日采购单 buy\_insert

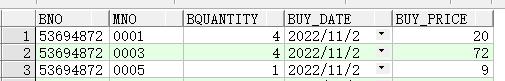
输入：日期

功能：根据日期筛选出当日已支付的订单，计算每个原料应该采购的数量、采购总价等插入采购表



测试执行：



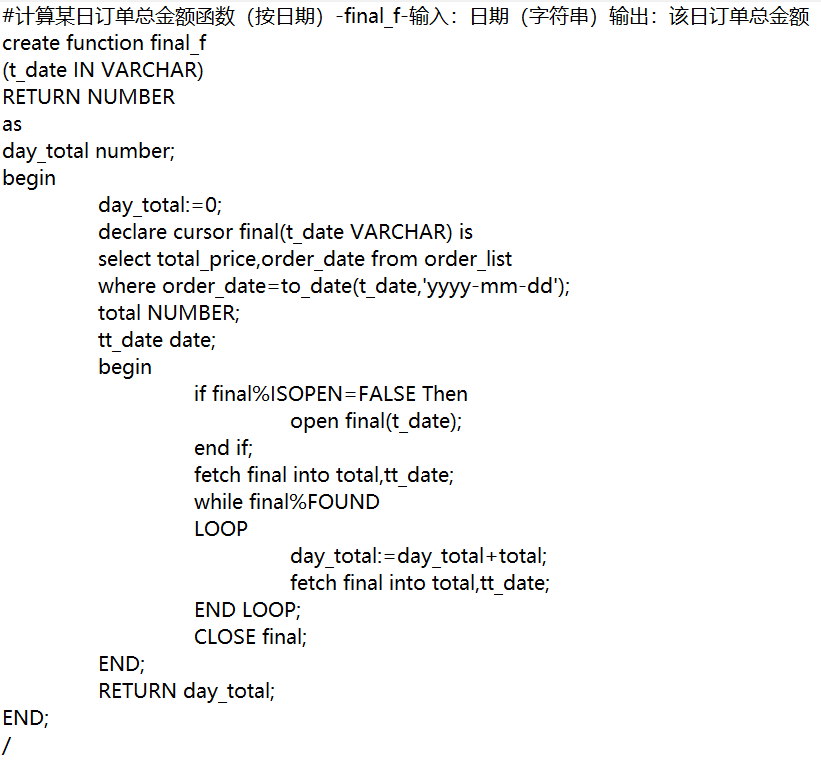


3.4函数设计

（1）计算某日订单总金额 final\_f

输入：日期（字符串）

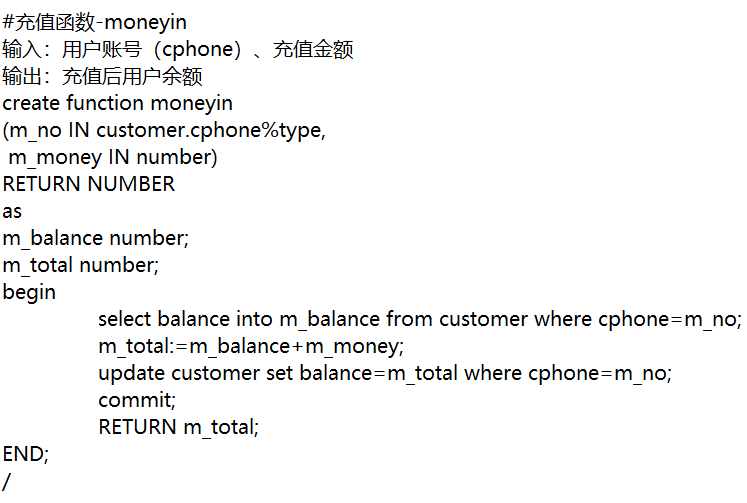
输出：该日总收入



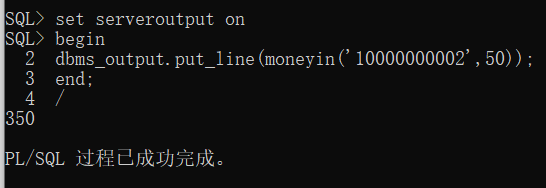
（2）充值 moneyin

输入：用户账号、充值金额

输出：充值后总金额



测试执行：

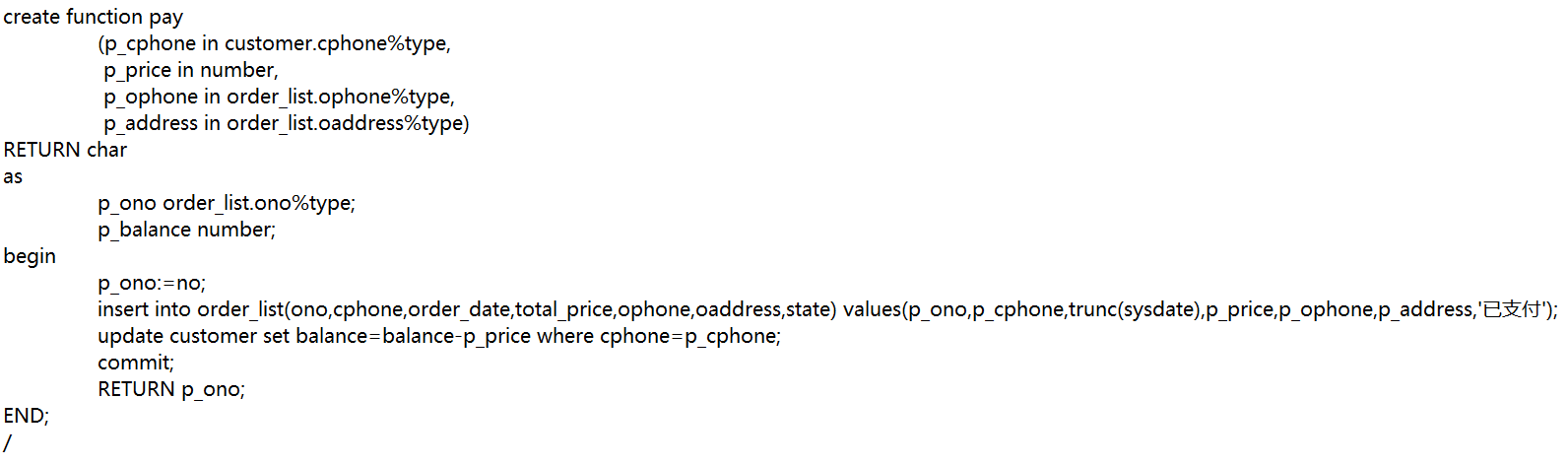


（3）支付 pay

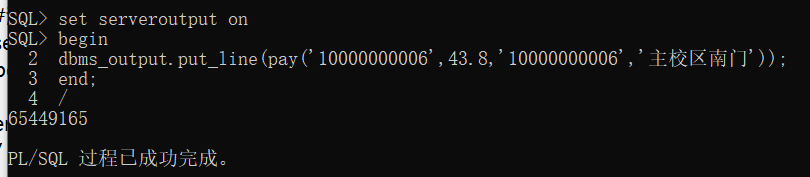
输入：用户账号、订单总价、电话、地址

输出：订单号

功能：插入订单表记录，生产订单号，下单时间为当前系统时间，订单状态为已支付



测试执行：

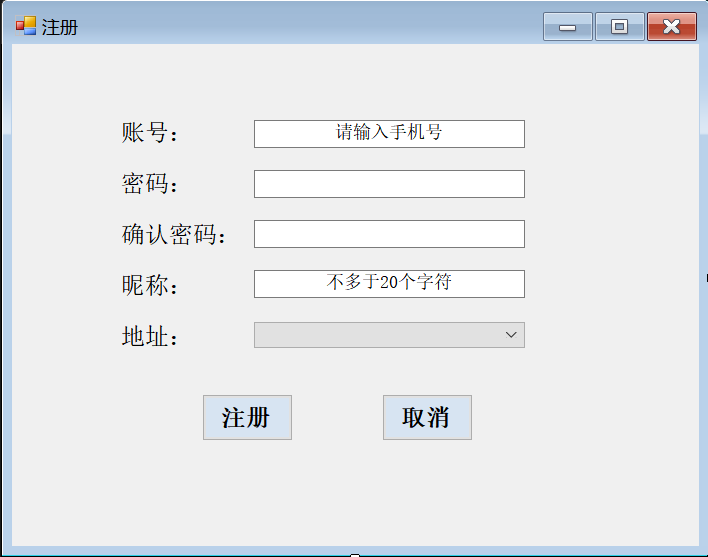


4. C/S架构拼拼系统窗体及功能展示

（1）首先显示登录主界面，有用户登录、取消、注册、和管理员登录的选项按钮。该窗体的主要功能为用户登录，需要匹配数据库当中的用户表，当账号和密码都正确时方可登录。



（2）没有账号的用户，可以点击注册按钮，进入注册界面。和许多注册界面一致，设置了电话的检验，必须为正确的11位手机号，以及密码的两次确认，地址设置了下拉列表，选择仅有的校内送货地址，确认无误后将新注册的用户信息更新至数据库的用户表中。



（3）管理员登录界面。管理员登录界面比较简单，仅需要检验账号和密码是否与数据库中的管理员表匹配即可登录。没有设置管理员注册界面因为管理员通常是直接添加的账号信息。



（4）管理员登录后出现管理员个人中心界面，在这里可以显示管理员的账号、密码、昵称和工资信息。均为从数据库中调取管理员个人数据进行显示。在上方是管理员可以进行的一些功能按钮。



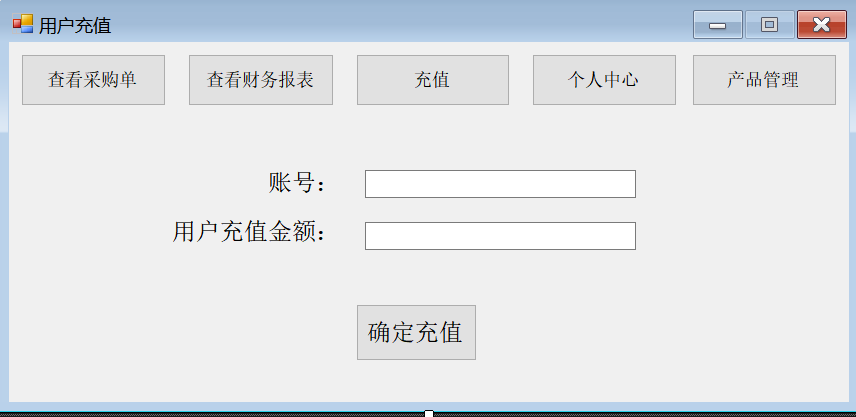
（5）查看采购单页面窗体。在这个窗体的下方灰色区域中可以显示总体的采购信息，均为从数据库中的采购单中调取信息显示。



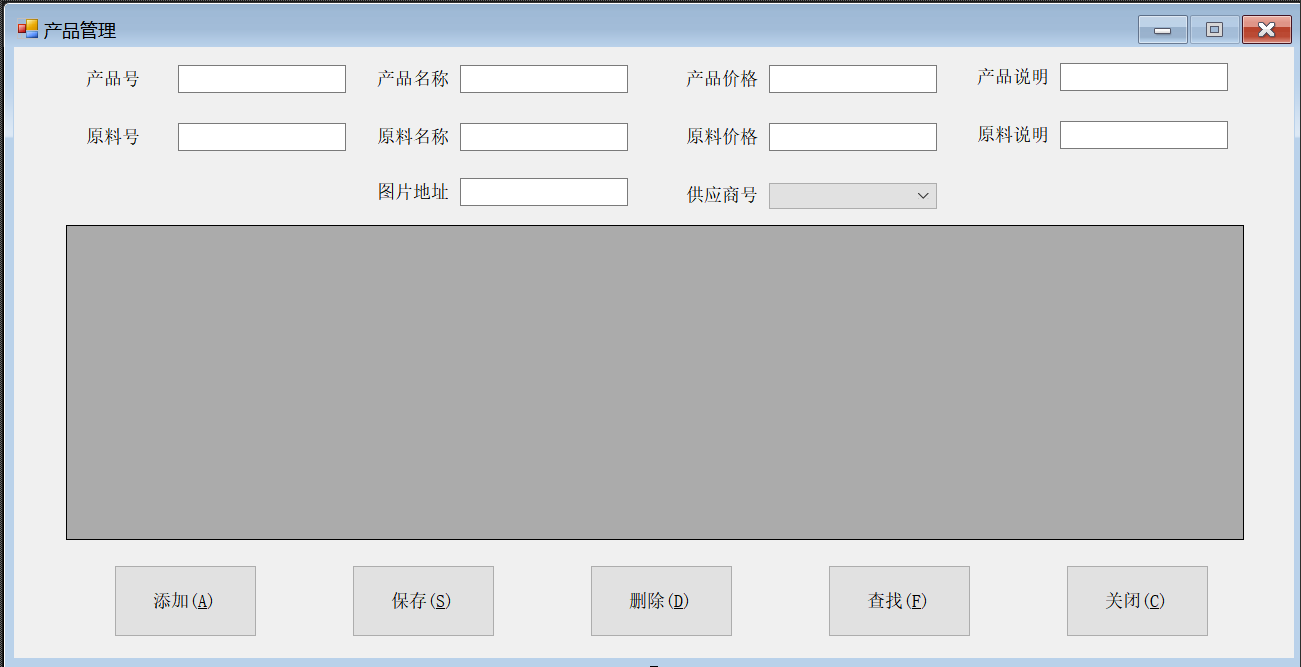
（6）查看财务报表窗体。这里设置了一个简单的日利润查看，即显示成本、销售额、净利润的数值，便于判断当日盈利。

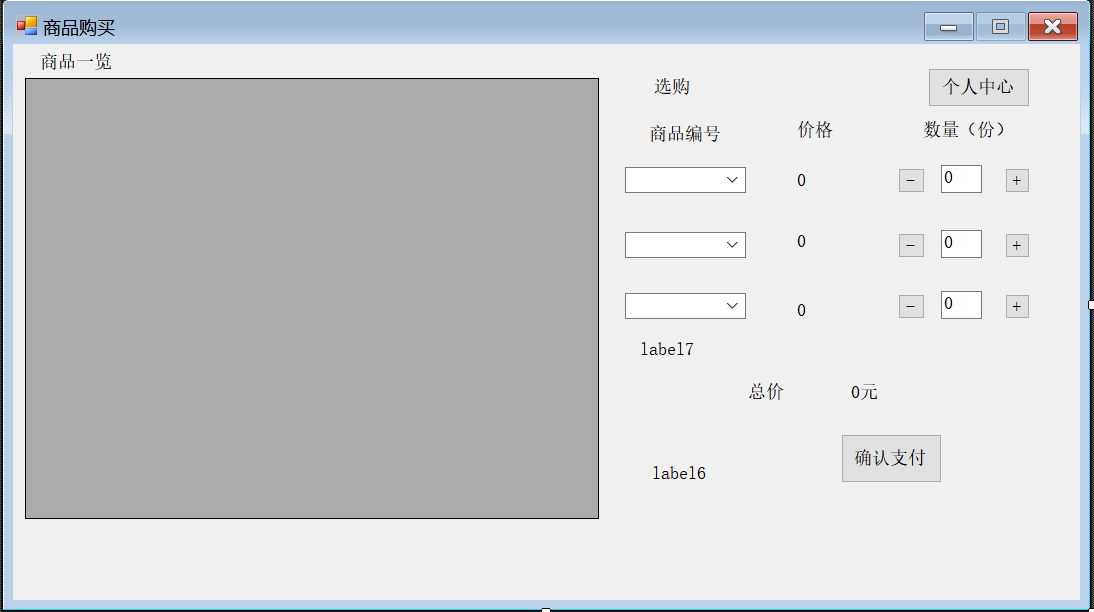


（7）充值窗体。这里假定用户在充完值后，我们为其手动增加账户金额。在确认充值后用户账户的余额应该增加，公式更改数据库中的用户余额。



（8）产品管理窗体。在产品管理窗体需要输入产品号和原料号，点击查找就可以在上方显示该产品的信息；输入其他要更改的信息点击保存即可更新产品信息；输入新的产品号和原料号以及其他信息点击添加即可添加新的产品信息；点击删除可以产出该产品信息。下方灰色区域显示从数据库中调取的所有产品信息。

  
 （9）用户登录后加购购买界面。左侧灰色区域显示所有的产品，包括商品标号和其他信息，在右侧可以从下拉列表中选择购买的商品，添加数量，下方会自动显示总价，点击确认支付就会在订单表中生成一项订单，同时在用户的个人中心可以看到自己的订购记录。



（10）用户个人中心界面。在用户的个人中心界面可以查看自己的账号、密码、余额、昵称、地址、积分等信息。在我的订单页面可以查看当前的订单信息以及订单状态。

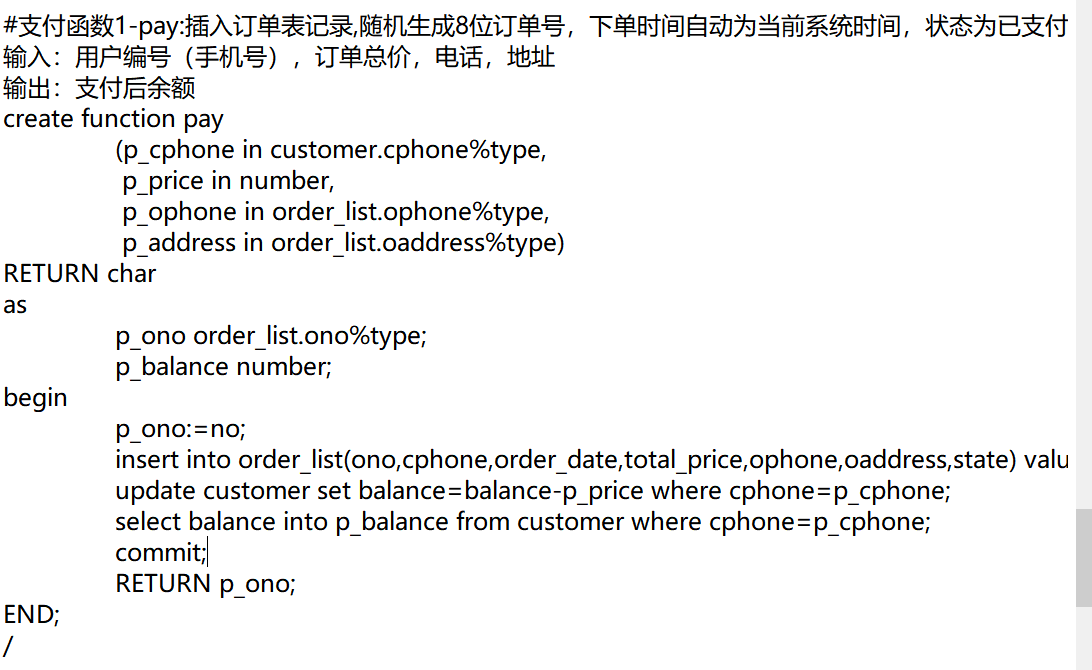


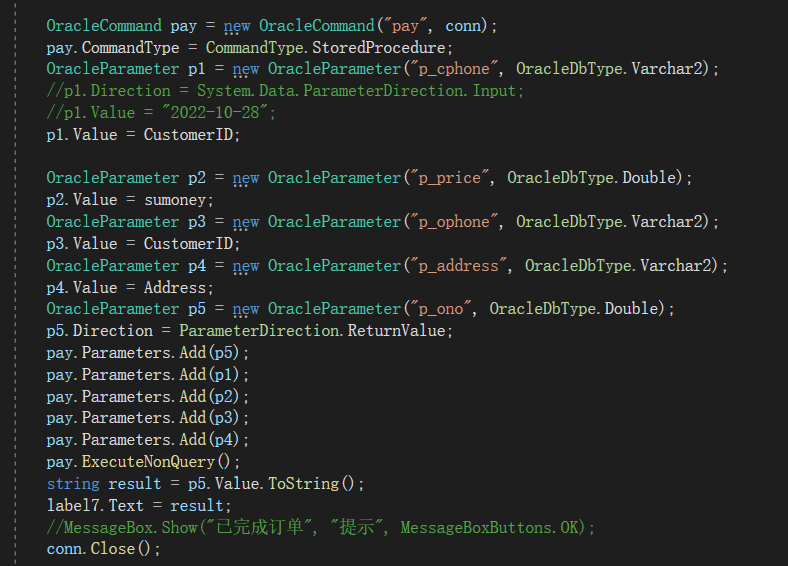
1. 用户修改个人信息窗体。在本窗体用户可以选择修改个人密码、昵称和地址。更新完的信息会同步修改至数据库中对应表中。



4.2 与数据库有关的部分代码展示

（1）Oracle函数调用

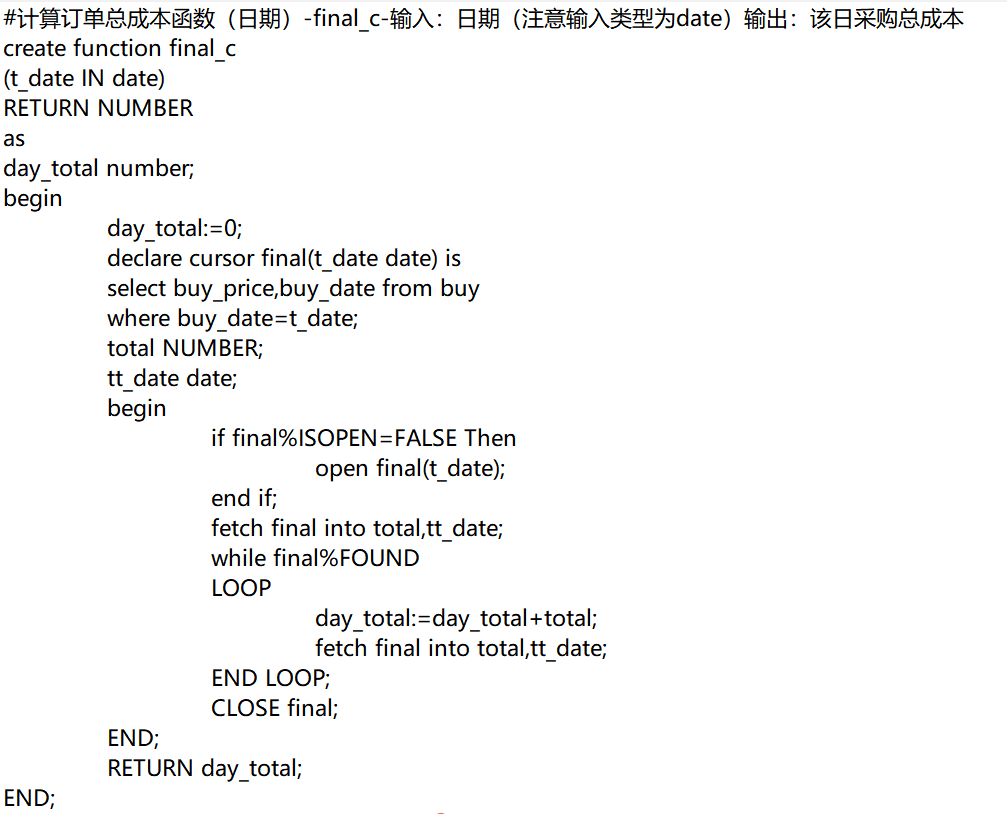


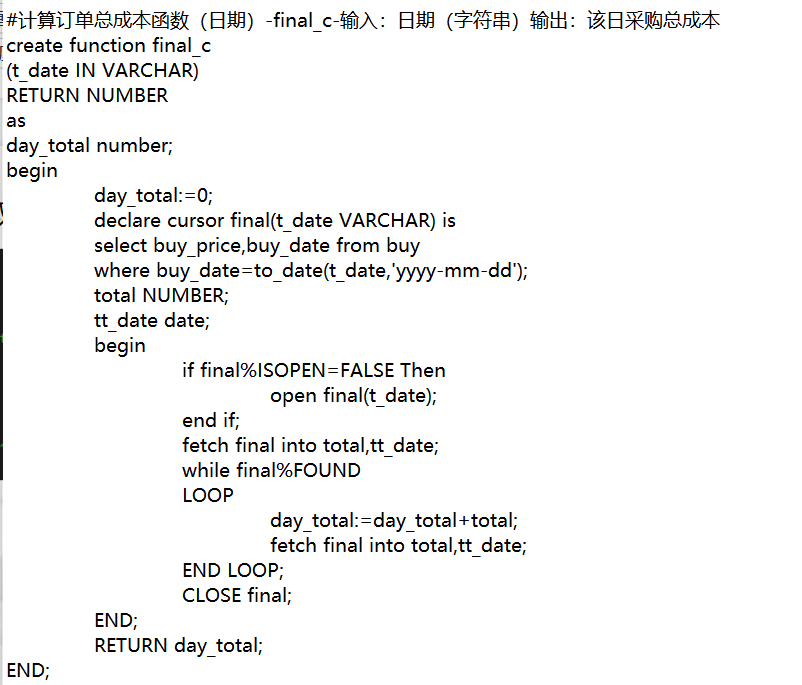


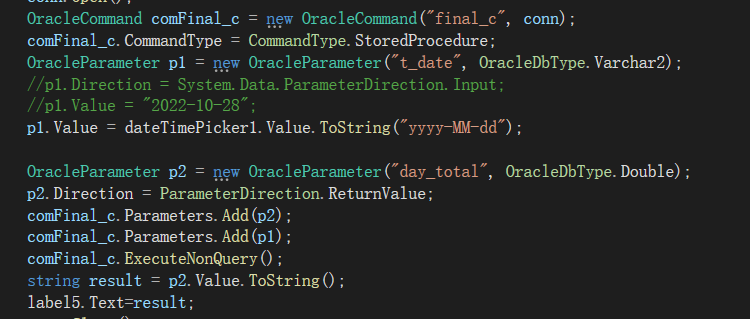
（2）时间类型的转换，时分秒的截取

在编写查看当日成本的过程中，需要调用相关数据库内函数，但是却发现始终无法显示，最终和同学讨论发现，在数据库函数中输入的是不带有时分秒的date型数据，但是在VS设计当中的timepicker控件是带有时分秒的date型数据。查阅各种资料后也没有找到如

何将VS中的时间变为只有年月日和date型数据。最后我们的解决办法是在数据库的该函数中，将输入的时间类型改为varchar型，并在函数中进行数据类型转换







4.3可优化的不足

（1）系统健壮性不足

（2）系统美观性不足

（3）其他细节问题

5. BS架构开发成果与说明

5.1 “拼拼”项目

我们的项目使用Python开发，需要使用Pycharm等IDE打开该项目文件夹



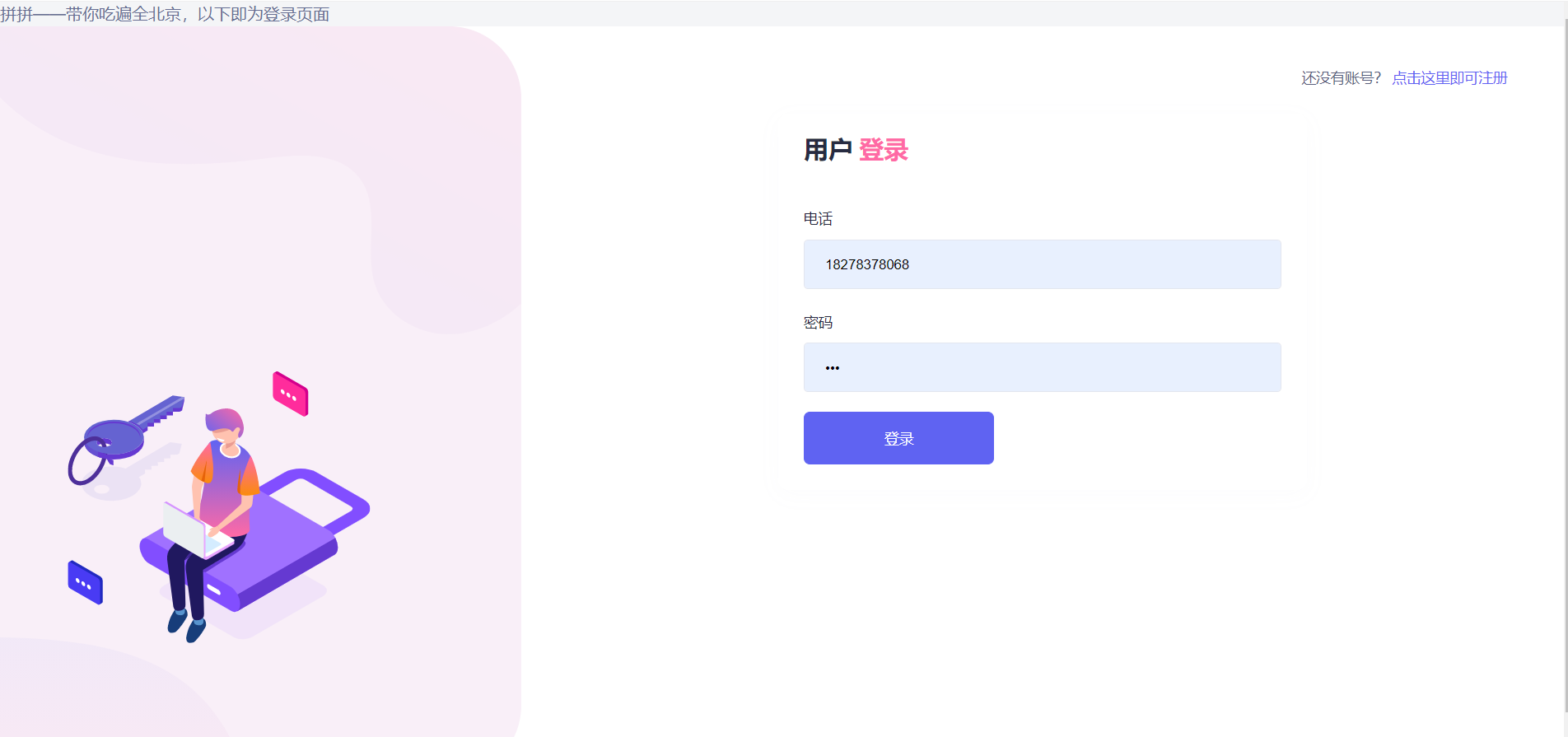
之后就是我们的相关代码。

5.2 项目的运行

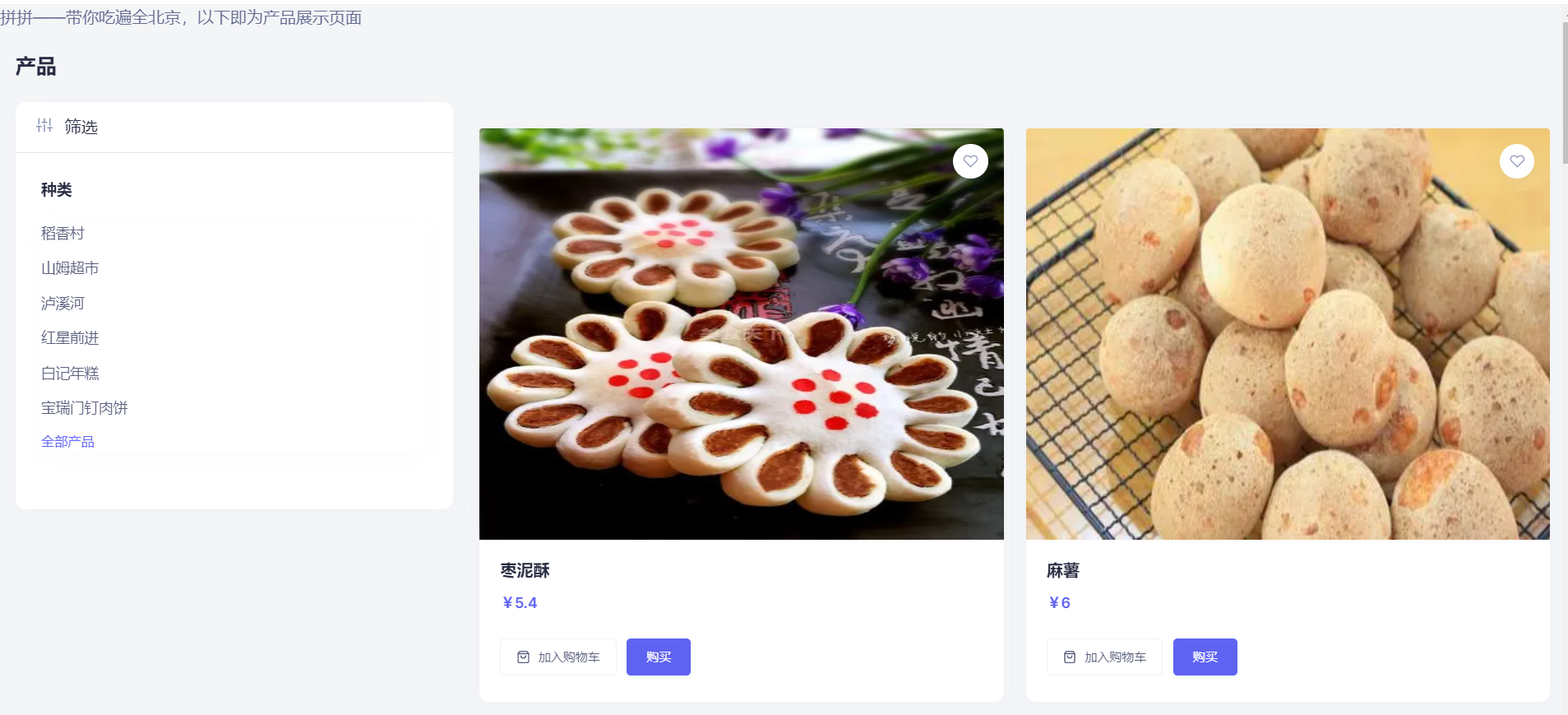
在venv目录下，我们运行app.py文件，之后得到以下结果：



我们单击加入即可看到登陆界面



由于数据库需要在局域网内才可访问的限制，用户登录等行为受到影响。通过html文件也可以访问到相关页面。



由于数据文件及网页相关内容全部存储在数据库中，前端仅作为一个展示窗口，所以在直接访问html文件会出现一些无法查询到结果的问题。这也是为了数据安全需求而做出的设计考量。