

# **“C++程序设计与训练”课程大作业**

## **项目报告**

**项目名称：超市商品管理系统**

姓名： 张若弛

学号： 2013011551

班级： 自 36 班

日期： 2014/9/20

目录

- 1 系统功能设计 .....5
  - 1.0 概述 ..... 5
  - 1.1 商品信息管理模块 ..... 5
    - 1.1.1 商品信息录入.....5
    - 1.1.2 商品信息修改/删除 .....6
    - 1.1.3 商品信息条件查找.....6
    - 1.1.4 商品信息统计分析 .....6
  - 1.2 多用户模块 ..... 6
    - 1.2.1 多用户的注册和登录.....6
    - 1.2.2 用户信息补充/修改 .....7
    - 1.2.3 多用户的权限管理.....7
  - 1.3 附加功能模块 ..... 8
    - 1.3.1 换肤功能.....8
    - 1.3.2 备忘录功能.....8
    - 1.3.3 软件更新建议收集功能.....8
    - 1.3.4 即时时间更新功能.....8
- 2 系统总体结构 .....9
  - 2.1 概要设计 ..... 9
    - 2.1.1 数据库构造.....9
    - 2.1.2 模块分工与关系.....9

2.1.3 商品信息的类与排序所用接口 .....	10
2.2 小组组员分工 .....	12
3.本人工作内容 .....	13
3.0 功能实现表 .....	13
3.1 UI 的设计与界面的美化 .....	14
3.2 主体框架的设计、搭建 .....	17
3.3 各类函数的编写与类的设计 .....	17
3.4 各数据库的设计、构造与系统的连接.....	20
3.4.1 商品信息数据库.....	20
3.4.2 用户信息数据库.....	21
3.4.3 绘制图表专用数据库.....	21
3.4.4 建议数据库.....	21
3.4.5 内部代码数据库.....	22
3.4.6 数据库与系统的连接.....	22
3.5 主要功能模块的编写 .....	23
3.5.1 商品信息的新建.....	23
3.5.2 商品信息的修改与删除.....	24
3.5.3 商品信息的查询（条件性查询） .....	25
3.5.4 商品信息的统计与水晶报表的制作.....	27
3.6 用户管理模块的编写 .....	29
3.6.1 用户的注册、登录、信息完善与密码的修改 .....	29

3.6.2 用户权限管理.....	30
3.6.3 用户信息管理.....	31
3.6.4 其他附加功能的实现.....	32
4.项目总结 .....	33
4.1 版本管理 .....	33
4.2 个人感想 .....	35

# 1 系统功能设计

## 1.0 概述

该软件用于对超市商品信息进行管理，实现信息的记录、修改、阅览、查找、统计等基础功能。通过被授权的用户进行安全化管理。同时附加了对用户权限的管理等功能，增加了软件的多功能性。软件力求功能实用性和操作易行性，界面的简洁化与美观性，提升用户体验。同时由于其与数据库交互频繁，要求其有着良好的容错性与健壮性，以避免错误数据的输入毁掉整个数据库。

## 1.1 商品信息管理模块

商品信息管理模块是软件主体。该模块能够让用户记录商品各项基本信息；还能够让用户修改信息、删除商品记录；提供精确与较模糊的查找功能为使用者提供查找商品信息的最为便捷的方式；提供统计功能，是用户按需统计商品信息，并根据记录输出图形化分析数据。

### 1.1.1 商品信息录入

实现商品信息的录入，录入信息包括：商品一级分类、二级分类、商品代码、商品名称、商品状态、商品数量、商品进价、商品售价、操作时间（对商品完成操作的时间）、商品补充信息（退回原因、售出季度等）。

录入窗口提供可供用户选择的一级分类、二级分类的下拉框，并根据分类自动生成商品代码的前四位（前两位代表一级分类，后两位代表二级分类）。另外，根据商品的状态不同提示用户输入退回理由或者根据生成日期自动生成售出季度，根据商品的分类不同提示输入不同的一级分类特点与两个二级分类特点。同时具备商品代码查重功能等与容错相关的函

数，通过检索数据库对重复的商品代码或其他不合理输入给出错误提示信息。

### **1.1.2 商品信息修改/删除**

根据商品某项或多项特征信息查找出需要修改的商品，对选中商品完成修改或删除。被修改的信息除相应项发生改变外，其记录者一项信息应自动更改为正在进行操作的用户。

### **1.1.3 商品信息条件查找**

通过用户设定的信息对商品进行查找，查找出的结果会根据用户设定的排序方式（商品销量、商品利润等）出现，排序顺序最多可设两项，执行排序功能后输出到表格。同时应该提供表格数据的输出为文件功能或打印功能（可以通过输出 Excel 表格再打印）方便超市的实际应用管理。

### **1.1.4 商品信息统计分析**

可统计出用户所需要的商品信息（如：销售量最大的商品、利润最大的商品等）；根据用户需求生成图表（商品季度销售曲线、直方图、环状图等），支持对报表进行存档、打印、放大等功能。

## **1.2 多用户模块**

### **1.2.1 多用户的注册和登录**

系统需要支持多用户，则首先需要让用户注册和登录。因为该软件针对的是超市内部的员工，为保证该超市商品信息安全性，注册用户除了输入用户名、密码还应该输入一个指定验证码证明其合法身份。用户密码的储存应该加密而不应该直接保存，以防止被居心不良的

人轻易盗取账号。用户使用已有账号完成登录进入管理界面。注册与登录都需要与数据库的交互

1.2.2 用户信息补充/修改

新注册的用户第一次登陆后，应该被提醒完善用户信息（包括性别、部门、职位）以便管理者管理员工信息（图 1.2.2-1）。同时允许用户更改密码（图 1.2.2-2）。



图 1.2.2-1

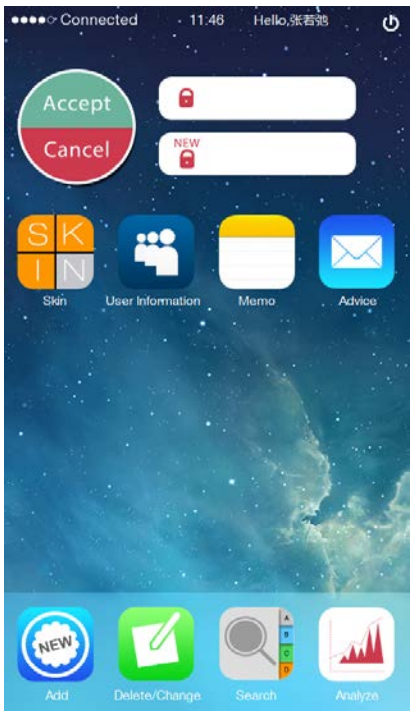


图 1.2.2-2

1.2.3 多用户的权限管理

为了更加符合现实中超市的架构，我们设计采取了 5 级权限制度，其中 0 为最高权限，支持所有功能并有着管理用户权限、薪水以及注册时内部代码的特殊功能。1 支持所有的基  
本功能，2 则无法查看统计数据与销售报表，3 与 2 相比无法使用条件搜索功能，4 与 3 相比无法查找修改删除商品信息。

## 1.3 附加功能模块

### 1.3.1 换肤功能

为实现界面的美观化与个性化，支持用户对软件界面背景进行切换。

### 1.3.2 备忘录功能

备忘录组件中，用户可以记录备忘信息，也可选择将记录信息删除。

### 1.3.3 软件更新建议收集功能

该功能收集存储用户在使用软件后的反馈意见与建议，方便程序设计人员日后对软件的更新与完善设计，便利用户使用（图 1.3.3-1）。



Advice

标题：唔~~你们为什么要开发换肤功能啊

你们开发换肤功能是寂寞了吗~~不过还挺好看的说... (•'◡'•)|

Submit Cancel

图 1.3.3-1

### 1.3.4 即时时间更新功能

主界面正中上方显示当前时间，该时间与计算机时间一致，并每分钟自动更新一次。



## 2 系统总体结构

### 2.1 概要设计

软件按照功能模块划分构建窗体，用数据库存储信息，根据用途和信息类别的不同存在在不同的数据表中，通过 SQL 语言对其中数据进行存取、查询、更新、管理。通过 OLE DB 对公开的数据进行访问。通过库中各种控件完成用户可视化操作。根据商品信息的分类不同使用了不同的类，另外由于排序功能的需要，根据排序因素的不同编写了不同的 IComparer 接口类。

#### 2.1.1 数据库构造

根据用途和信息类别的不同，一共构造了五个数据表，分别为 productinfo（用于存储商品信息），userinfo（用于存储用户个人信息），advice（用于存储用户反馈的建议），temp（作为临时数据表储存用于水晶报表显示所需的数据），innercode（用于存储注册用户时所需的内部代码）

#### 2.1.2 模块分工与关系

本系统主要实现对商品信息的管理。管理的对象主要是商品条目数量等的商品清单，和商品库存信息。管理功能分为身份验证、商品信息管理等部门。功能模块列表如图 2.1.2-1，图 2.2.2-2 各模块的关系见图 2.1.2-3。

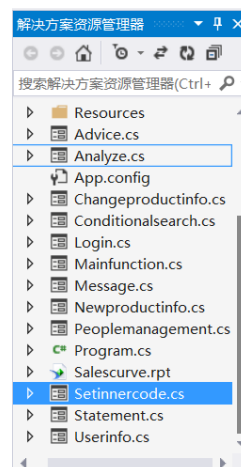


图 2.1.2-1 功能模块



图 2.1.2-2 功能模块

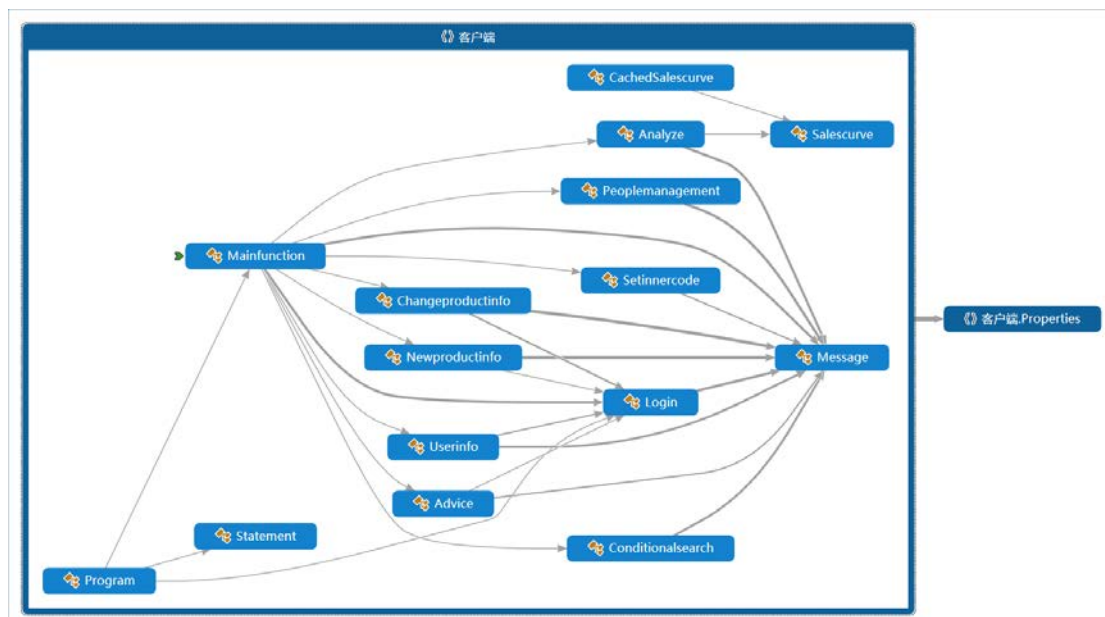


图 2.1.2-2 功能模块

### 2.1.3 商品信息的类与排序所用接口

本程序中储存及处理商品信息的类的设计是一个重点。我设计了一个叫做 ProductInfo 的基类，包括 set 与 update 的虚方法。尤其派生出四个派生类，分别表示四个一级分类，增加了一级分类特点这一属性，再由四个派生类分别派生出共计 13 个最远派生类，均增加

了不同的 2 个二级分类特点并覆盖了基类中的 set 与 update 方法。另外包括一个独立的 GeneralProductInfo 的通用商品信息类，用于商品信息的快速读取与排序（不用于商品信息的上传、修改与删除）。（见图 2.1.3-1）每个类具体所含字段与函数由于类图放不下，会在下文阐释，此处不再赘述。另外由于排序功能的需要，我还编写了许多的 IComparer 接口类，与 Sort()函数结合，实现了不同字段的排序功能。（见图 2.1.3-2）

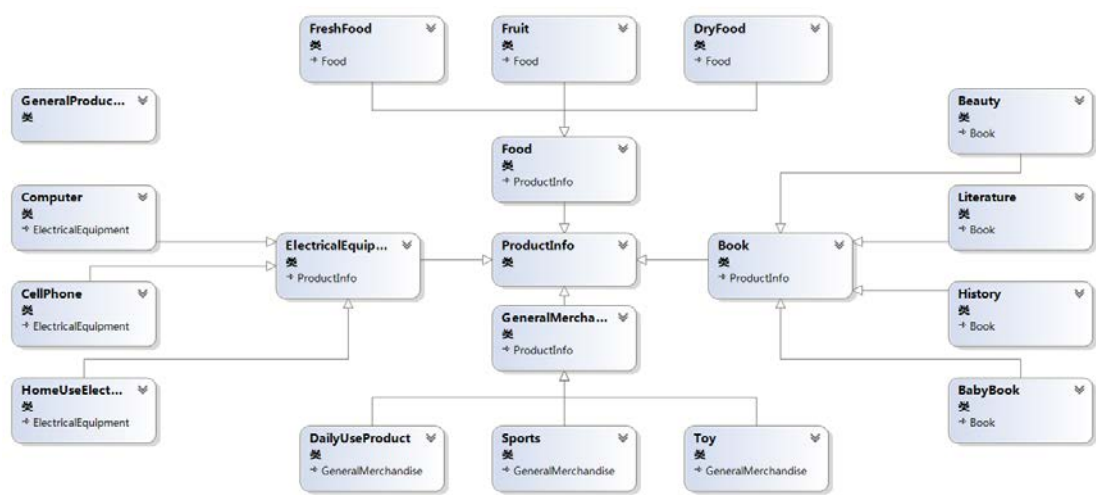


图 2.1.3-1

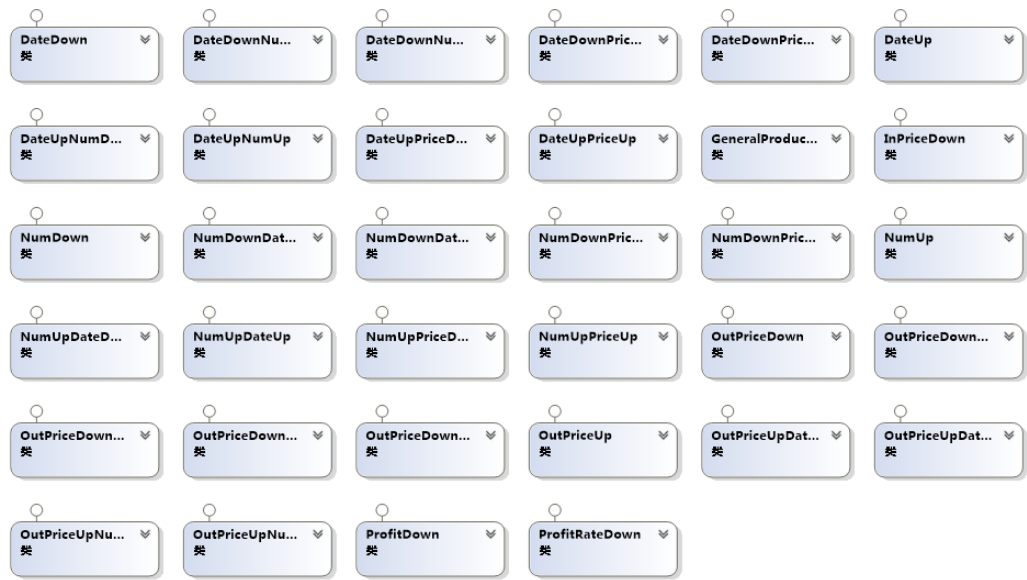


图 2.1.3-2

## 2.2 小组成员分工

小组分工如下表：

姓名	班级	学号	分工
张若弛	自 36	2013011551	UI 的设计与界面的美化 主框架的搭建，主框架、应用程序间的数据交互 各类函数的编写与类的设计 绘制图表专用数据库、产品数据库的构造 系统和数据库的连接 用户登录、注册、管理、权限模块的编写 商品信息新建、修改、条件搜索功能的编写 商品统计功能的完善 参与水晶报表的制作 换肤功能、备忘录功能的编写 其他拓展功能的编写
周婧雯	自 36	2013011552	意见数据库与用户数据库的构造 数据库商品信息的扩充填写 商品信息统计功能 1.0 的编写 商品信息修改功能的编写 查询结果导出文件功能的编写 意见反馈功能的编写 参与水晶报表的制作

			商品信息条件搜索功能的完善
			代码风格的维护
			产品的测试

### 3.本人工作内容

#### 3.0 功能实现表

功能类型	实现的功能点	实现方式	备注
		1) 自己编写 C#代码 2) 使用 C#标准库 3) 使用第三方库 4) 使用 SQL 语句	
初级功能	信息的读写	(1),(4)	将使用 SQL 语句编写的函数封装在自己编写的类中
	信息增加	(1),(4)	同上
	信息删除	(4)	仅使用简单的 SQL 的 DELETE 语句
基本功能	状态变更	(4)	仅使用简单的 SQL 的 UPDATE 语句
	记录退回原因	(1),(4)	将使用 SQL 语句编写的函数封装在自己编写的类中
	简单查询	(4)	仅使用简单的 SQL 的 SELECT 语句
	数据可存储至文件或数据库	(1),(4)	将使用 SQL 语句编写的函数封装在自己编写的类中
	容错特性	(1),(2)	自己编写了许多函数也使用了 C#自带的正则表达式
数据库支持	数据库读写	(4)	
复杂查询功能	排序	(1),(2)	利用 C#标准库中的 Sort()函数结合自己编写的 IComparer 类

统计功能	数值类的计算输出	(1),(2),(4)	与复杂查询中的排序功能类似，用 SQL 读出数据后使用 Sort()函数结合 IComparer 接口实现排序并输出最大值
	图表的生成	(1),(3),(4)	将所选范围的数据先保存在内存中，再利用 SQL 语句转存至临时数据库，最后通过水晶报表调用该数据库实现图表的产生
其它功能	用户的登录	(1),(4)	通过 SQL 语句与数据库进行校验实现用户的登录
	用户信息权限管理	(1),(4)	与商品信息管理类似
	换肤功能	(1)	
	备忘录功能	(1)	

### 3.1 UI 的设计与界面的美化

用户体验是决定一个软件生命力的所在，对软件的用户体验的考量一直贯穿在我们编写软件的全过程。界面的设计整体遵从时下流行的扁平化元素，通过一些动图或控件摆脱界面死板静止的原样。我在界面设计的时候考虑了以下几大原则：

首先，通过自身使用经验和询问其他同学，我认为界面上存在的按钮或其他控件应该尽可能的少，所给出的都是当前用得到的。那些使用频率不高或者在当前功能不会使用的控件应该隐藏以避免用户误点，同时也能使得界面更加简洁，美观。而实现手段就是控件复用与控件隐藏。而这两个手段在这个软件的各个角落都可以找到。图 3.1.1-1 与图 3.1.1-2 为本人在课余开发的一款聊天软件的登陆界面与本大作业的登陆界面。

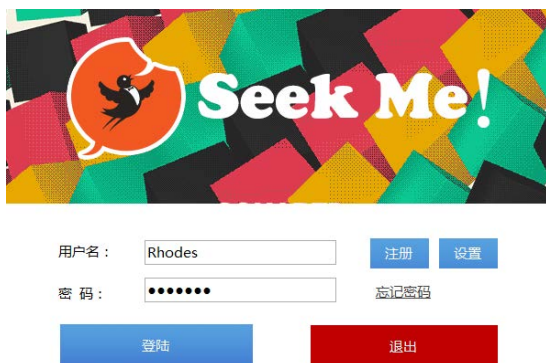


图 3.1.1-1

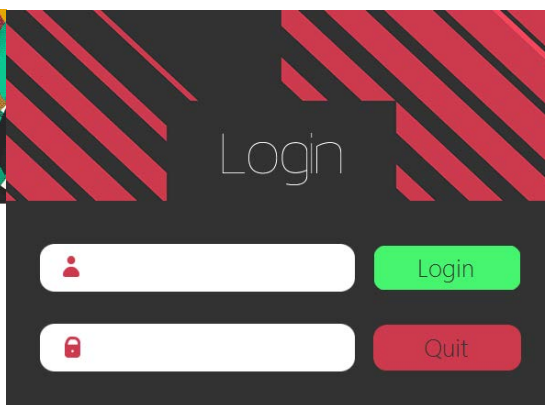


图 3.1.1-2

两相比较,是否后者的界面显得更为简洁呢?因为我将实际使用中并不常用的注册功能给隐藏了,但需要使用时也可通过单击 LOGO 快速召出(如图 3.1.1-3)

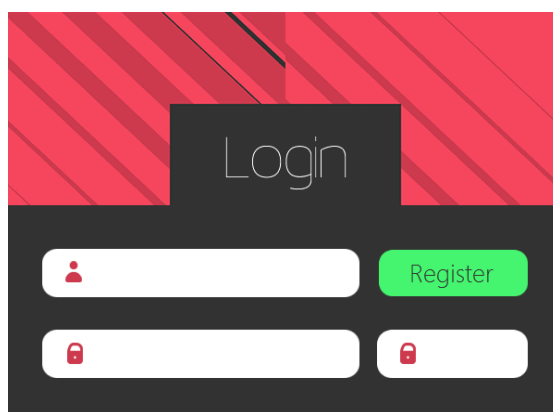


图 3.1.1-3

其次,我通过 PS、AI、FLASH 等软件的帮助制作了动图并通过 picturebox 控件添加在软件的各处。有统计显示,由于人脑的响应机制,动态因素的添加能使人对某款软件产生更大的好感,同时一些动图也有着在软件加载数据的等待阶段分散注意力的作用(比如腾讯 QQ 就是一个很好的例子)。

第三,我通过时间控件和窗体的 Opacity 属性实现了所有窗口的渐变加载效果,使得各个功能模块的过渡更加自然。同时通过时间控件和其他控件的 Location 属性实现了个别控件的运动加载效果,有着提醒用户控件加载的作用。

第四,追求界面风格的统一。界面风格的统一不仅是单纯为了好看,更是基于对人性化的考量。如果在使用类似 QQ 这样精心设计过的软件时弹出了 Windows 自带的信息弹窗,

大部分人的第一反应是系统出错了。而对于超市商品管理系统这样一个与系统交互频繁，需要大量信息弹窗的软件，如果还使用自带弹窗可能不太合适。我因此重绘了系统自带的 MessageBox 类，使得其更符合整体风格（如图 3.1.1-4）



图 3.1.1-4

最后，从实际用户的角度出发，优化界面设计。我在制作完 UI 后让一些同学使用后提出意见，并据此不断调整 UI。比如修改商品信息功能：修改栏本来是隐藏在数据表的背面的，直到单击 Change 后才出现（可以发现此时 Change 变为了 Cancel，这也体现了我上述的控件复用与界面简化的想法），本来上方的选择框是空的，但在同学的建议下修改为显示所修改的商品的信息。（图 3.1.1-5，图 3.1.1-6）



图 3.1.1-5



图 3.1.1-6



总而言之,我始终将用户体验放在仅次于逻辑层的重要位置,为此我们做了许多测试与优化。这从我们基本上每个窗体 UI 层代码占到三分之一左右的分类就可见一斑。

### 3.2 主体框架的设计、搭建

软件的各个功能通过多个窗体实现,主要窗体为“Mainfunction”。用户首先通过 Login 窗体登录,如登陆成功,Login 窗体中的静态变量 id,permission 将记录当前用户的用户名与权限。Mainfunction 根据不同的权限显示不同的界面来实现对用户所能使用功能的区分。同时 Mainfunction 是用户选择使用何种功能的所在窗体,是软件的主窗口。

### 3.3 各类函数的编写与类的设计

为了使用更加简便与体现面向对象的思想,我们将许多常用功能封装为函数或封装到类。比如条件搜索与修改删除功能中 SQL 搜索语句的生成就是通过封装为 searchcommand() 函数来进行动态生成的。再比如软件中大量存在使用的查错函数,包括检验是否只含数字的 isnumber()函数,是否为实数的 isfloat()函数等等,这些功能频繁使用,因此将其封装为函数大大简化了后期编程的过程。

而在类的设计上,我建立了一个类库 Class1.cs,其中包括了基类 ProductInfo 与其派生子类。其派生关系如图 3.3-1

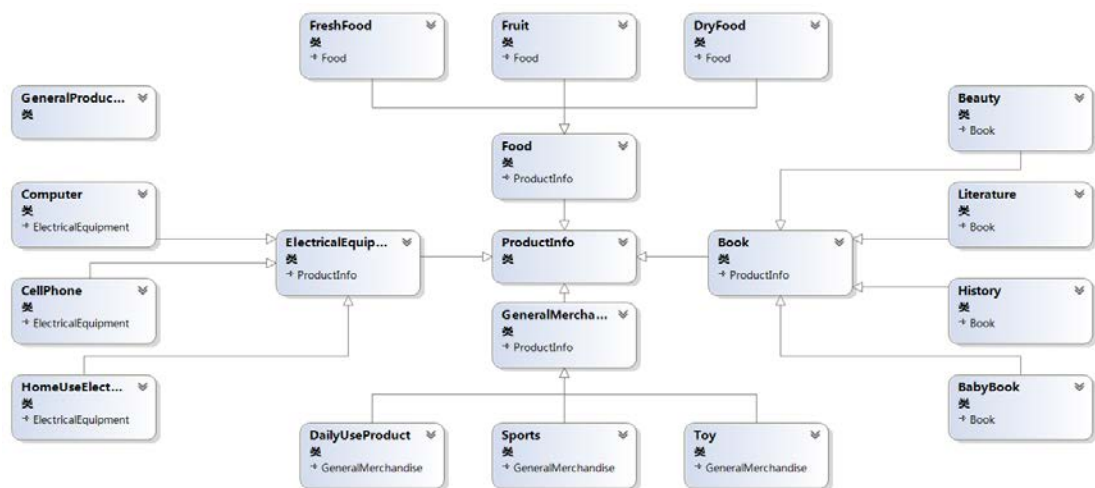


图 3.3-1

下具体阐释各个类所包含的数据成员与函数成员（如图 3.3-2）

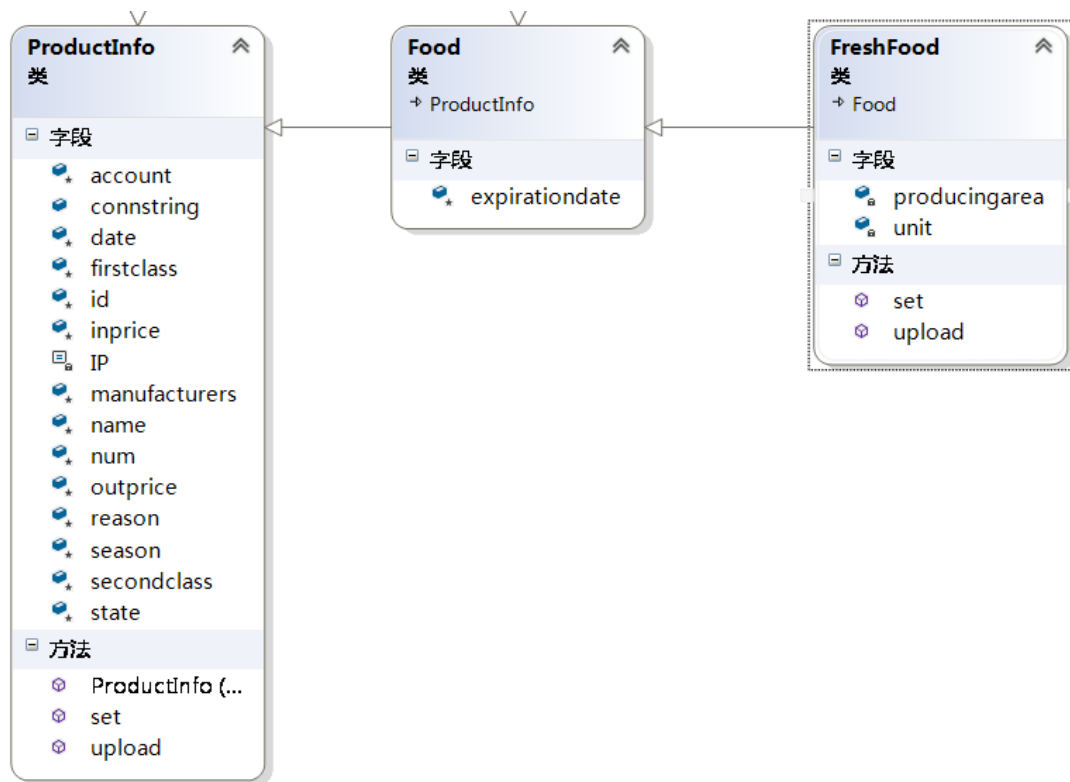


图 3.3-2

以众多派生关系中的一条为例，基类 ProductInfo 中包含有包括商品名称，代码，生产日期，厂商在内的 13 条信息与两条与数据库连接相关的数据成员，同时含有除构造函数外的 set(),upload()两个数据库操作函数。派生类 Food 继承了 ProductInfo 并增添了保质期这一一级分类特点，然后最远派生类 FreshFood 继承 Food 并增添了生产地点与售卖单位

这两个二级分类特点，同时覆盖了基类的两个函数。

同时还有一个 GeneralProductInfo 的商品通用信息类（如图 3.3-3），用于在搜索功能时排序用到。为了防止类变得过于庞大以及使用的方便，各成员均声明为公共类。此类只用于快速读取与排序操作，与前面的类相比增加了利润与利润率两个变量，方便排序的使用。同时多了一个 string 类数组 data 成员，方便输出语句的编写。以上多出的三个变量在 set 函数初始化对象的同时由数据成员经过操作赋值。

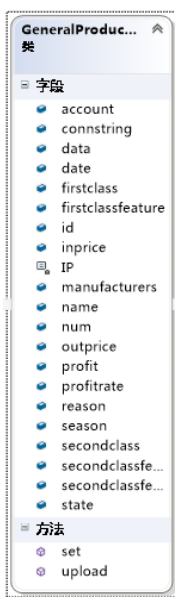


图 3.3-3

另外同时由于条件性搜索以及统计功能所需，类库中还含有大量的 IComparer 接口类（如图 3.3-4），均不含数据成员只含有一个 Compare 方法，这些类与 Sort()函数结合，完成了大部分的排序功能。各个窗口通过对命名空间“类库”的引用，使用这些类，方便了数据的存储与上传，同时利于将来更多属性的添加与扩展，通过类的继承与多态体现了面向对象的思想。

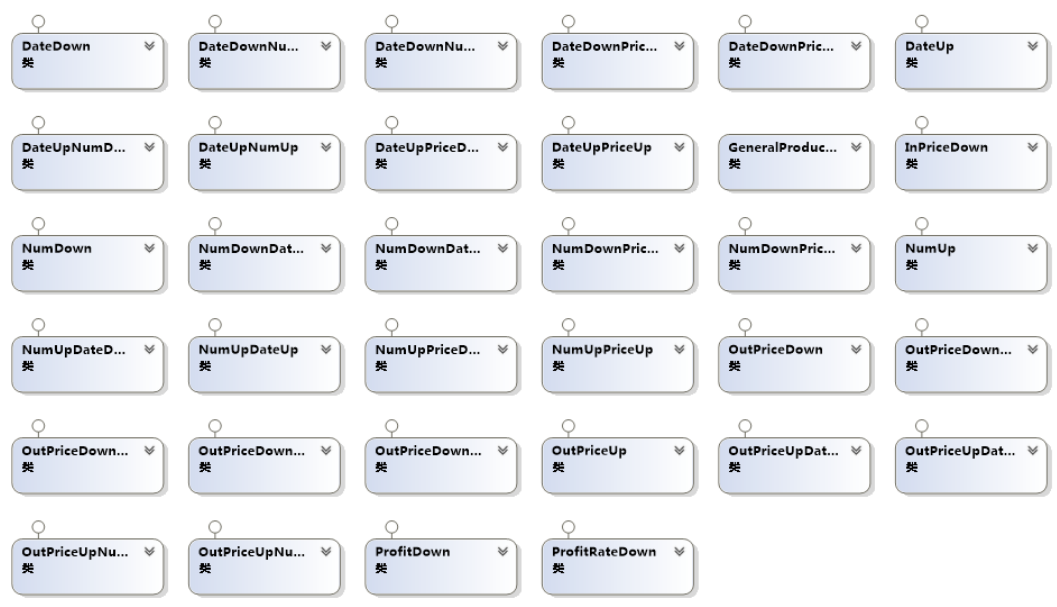


图 3.3-4

3.4 各数据库的设计、构造与系统的连接

3.4.1 商品信息数据库

	列名	数据类型	允许 Null 值
▶	firstclass	varchar(50)	<input checked="" type="checkbox"/>
	secondclass	varchar(50)	<input checked="" type="checkbox"/>
	id	varchar(50)	<input checked="" type="checkbox"/>
	name	varchar(50)	<input checked="" type="checkbox"/>
	state	varchar(50)	<input checked="" type="checkbox"/>
	inprice	float	<input checked="" type="checkbox"/>
	outprice	float	<input checked="" type="checkbox"/>
	season	varchar(50)	<input checked="" type="checkbox"/>
	reason	varchar(100)	<input checked="" type="checkbox"/>
	num	int	<input checked="" type="checkbox"/>
	account	varchar(50)	<input checked="" type="checkbox"/>
	date	datetime	<input checked="" type="checkbox"/>
	manufacturers	varchar(50)	<input checked="" type="checkbox"/>
	firstclassfeature	varchar(50)	<input checked="" type="checkbox"/>
	secondclassfeature1	varchar(50)	<input checked="" type="checkbox"/>
	secondclassfeature2	varchar(50)	<input checked="" type="checkbox"/>

图 3.4.1-1

此数据库用于储存商品的信息，不同类的商品信息都上传到这个数据库中。

3.4.2 用户信息数据库

	列名	数据类型	允许 Null 值
▶	name	varchar(50)	<input checked="" type="checkbox"/>
	password	varchar(50)	<input checked="" type="checkbox"/>
	sex	varchar(50)	<input checked="" type="checkbox"/>
	department	varchar(50)	<input checked="" type="checkbox"/>
	position	varchar(50)	<input checked="" type="checkbox"/>
	salary	int	<input checked="" type="checkbox"/>
	permission	int	<input checked="" type="checkbox"/>

图 3.4.2-1

	name	password	sex	departme...	position	salary	permission
▶	admin	D033E22A...	男		总经理	100000	0
	test1	B444AC06...	女	门市部	店员	50000	1
	zjw	5BBEE5EB...	女	销售	职员	50000	1
	张若弛	7C4A8D09...	NULL	NULL	NULL	3000	4

图 3.4.2-2

此数据库储存了用户的账号密码、工作信息与访问权限。如图 3.4.2-2 所示，数据库内的密码都通过哈希算法加密了以防止用户的账号密码被轻易盗取。

3.4.3 绘制图表专用数据库

	列名	数据类型	允许 Null 值
▶	season	varchar(50)	<input checked="" type="checkbox"/>
	num	int	<input checked="" type="checkbox"/>

图 3.4.3-1

仅存在 season 与 num 两列，是水晶报表生成折线图时所用的数据库（具体后文会提到）。

3.4.4 建议数据库

	name	subject	content
*	NULL	NULL	NULL

图 3.4.4-1

用于保存用户所提的建议，分别保存用户名、建议标题与建议内容。

### 3.4.5 内部代码数据库

	code	id
▶	2333	1
*	NULL	NULL

图 3.4.5-1

用于保存注册时所用的内部代码。其实单独为内部代码建立一个数据表实在有些浪费资源，但是内部代码又必须保存在服务端而不能集成在本地文件。实在想不到更好的办法只能出此下策。

### 3.4.6 数据库与系统的连接

```
public static string IP = "59.66.135.77";//默认的服务器IP
```

```
public static string connstring = String.Format("Data
```

```
Source={0};InitialCatalog=ProductManagement;User
```

```
ID=Rhodes;Password=123456", IP);//默认的连接字符串
```

以上即为 SQL 连接所用的字符串，通过 sqlconnection 对象来实现对数据库的连接，同时通过一些封装为函数或封装在类内的函数实现了与数据库的交互。由于一些主观及客观的原因（电脑安装 SQL LITE 错误），我们采用了 SQL SERVER 而不是 SQL LITE 作为我们的数据库来源。其优点在于，将数据库储存在一台电脑（服务器）上，而由另外安装有管理软件的电脑读取数据的架构更为科学。实际应用中，超市不可能选择只使用一台电脑来管理商品信息，更不可能通过 U 盘或网络来将数据库信息来进行传递。但选择 SQL SERVER 也加重了我们的工作负担，光是配置静态 IP 与网络连接就耗费了一定时间。我们将数据库的连接与数据上传修改的操作封装成函数或封装在类内。使得一些常用操作的调用变得更加简单。

而 SQL 语句较为灵活的特点也使得我们可以编写出类似 searchcommand()之类的 SQL 语句生成函数。

### 3.5 主要功能模块的编写

#### 3.5.1 商品信息的新建



图 3.5.1-1

新建商品窗口的核心功能如下：

一是检测用户输入的内容是否合理：我们建立了一个函数库，里面包括了检验输入中是否只含数字、是否只含数字和小数点以及其他格式检测函数。通过这些函数以及对输入框是否为空的判断即可基本确保用户输入的信息是合理的，减少了数据库报错的可能性

二是根据用户的选择实时调整各个输入框所要求输入的内容。比如，根据用户所选择的第一分类，自动更改二级分类的内容，根据一级二级分类，显示具体的一级分类特点与两个二级分类特点。再比如根据商品的状态不同，“选填”变为“售出季度”或是“退回理由”

这样的特点还体现再很多地方，此处不再赘述。

三是将用户所输入的数据在内存处理后上传到数据库。根据软件的设计，在判断用户输入为合理的之后，根据一二级分类的不同新建不同类的对象，并将用户输入的数据通过对象的设定函数初始化对象，然后通过对象的上传函数将数据上传至数据库，并返回是否成功的信号，最终调用 Message 窗口显示是否上传成功的信息。

### 3.5.2 商品信息的修改与删除

Search / Change / Delete Product Information

一级分类

商品名称

当前状态

Change

Delete

二级分类

商品条码

操作时间

Search

Quit

	一级分类	二级分类	商品条码	商品名称	状态	商品进价	商品售价	售出季度	退回理由	商品数量	录入员
	百货	日用品	020101	金可儿...	售出	4800	6599			65	Rhodes 2
	百货	日用品	020102	金可儿...	上架	4800	6599			43	Rhodes 2
	百货	日用品	020103	金可儿...	售出	4800	6599	2		54	Rhodes 2
	百货	日用品	020104	金可儿...	售出	4800	6599	1		432	Rhodes 2
	百货	日用品	020105	乐享瓷...	上架	300	369			5	Rhodes 2
	百货	日用品	020106	珍珠生...	上架	700	919			32	Rhodes 2

图 3.5.2-1

Search / Change / Delete Product Information

一级分类 百货

商品名称 020101

当前状态 售出

Cancel

Delete

二级分类 日用品

商品条码 金可儿弹簧床垫

操作时间 2014/9/11 0:00

Search

Quit

新状态 售出

售出季度 1

Submit

图 3.5.2-2



商品信息的修改与删除功能其实暗含了商品信息的查询功能,但我认为此处商品信息查询的结果应该为商品信息的修改删除功能服务,所以要求其给出的结果尽量精确尽量少。所以我给出了包括一二级分类、商品名称等六个选项,用户输入这些信息并通过合理性检测函数后通过 searchcommand()函数生成了 SQL 的搜索语句字符串,然后通过 sqldatareader 返回搜索到的数据并输出到 datagridview。用户通过选择 datagridview 上的数据再通过“Delete”、“Change”按钮进入进一步的操作。删除功能是通过 SQL 的 DELETE 语句,以商品条码(id)作为唯一标识,操作后返回是否删除成功的信息,通过调用 Message 窗口显示删除情况。修改功能类似,通过 SQL 的 UPDATE 语句完成修改功能。细节上继续体现控件复用的想法,比如 Change 按钮在单击后变为 Cancel,上方原本的查询输入框变为了显示原来信息的显示框(文本框调整为不可输入)。

3.5.3 商品信息的查询（条件性查询）



图 3.5.3-1

商品信息的查询功能更偏重商品信息的范围查询,用户可以操作的内容包括数据的范围以及数据显示的排序方式。数据范围的限定与上一功能类似,都是通过 searchcommand()函数动态生成 SQL 语句然后读到 datareader 中的。区别在于,此处 datareader 并不直接

将数据输出到表格中，而是先保存在一个 GeneralProductInfo 类的对象数组中，然后对该数组按照用户选择的第一第二顺序调用类库中的 IComparer 接口类与 Sort()函数实现排序。最后将这个对象数组中的数据都输出到表格中。同时本窗口提供了让用户将数据导出保存为 Excel 文件的功能，在表格区域右键并单击另存为数据，即会弹出相应的对话框，保存后，即可通过 Excel 打开相应的数据文件。(如图 3.5.3-2 与图 3.5.3-3)

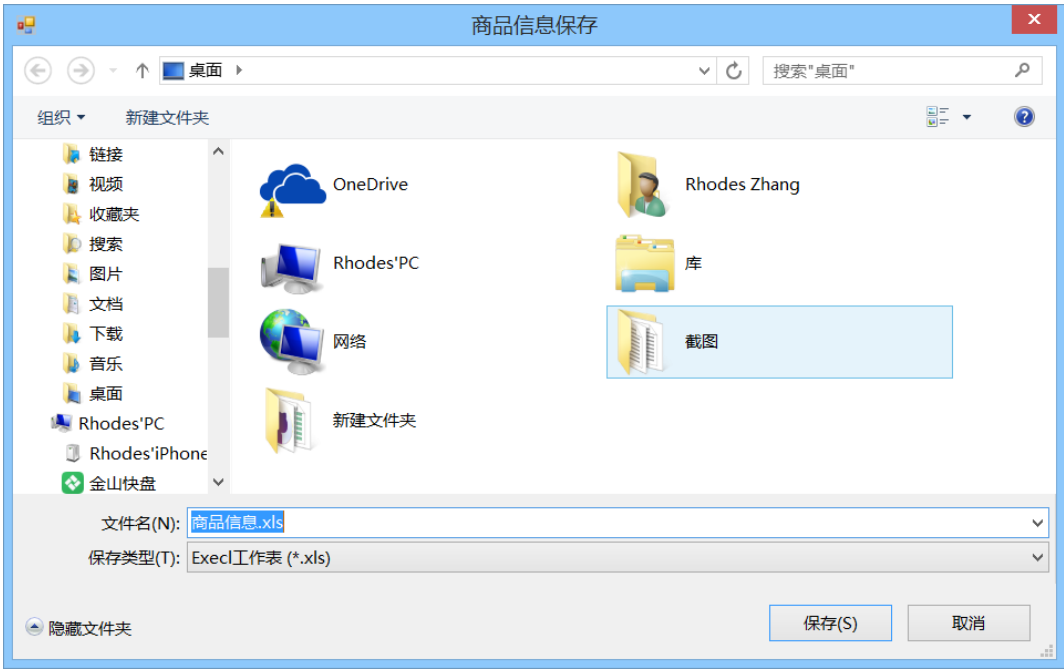


图 3.5.3-2

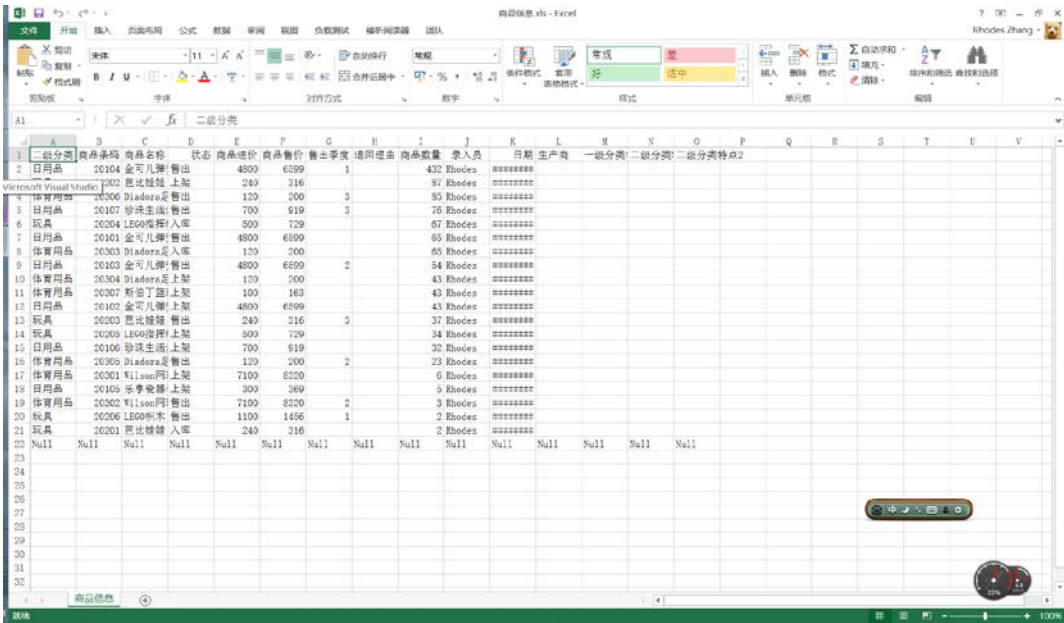


图 3.5.3-3

3.5.4 商品信息的统计与水晶报表的制作



图 3.5.4-1

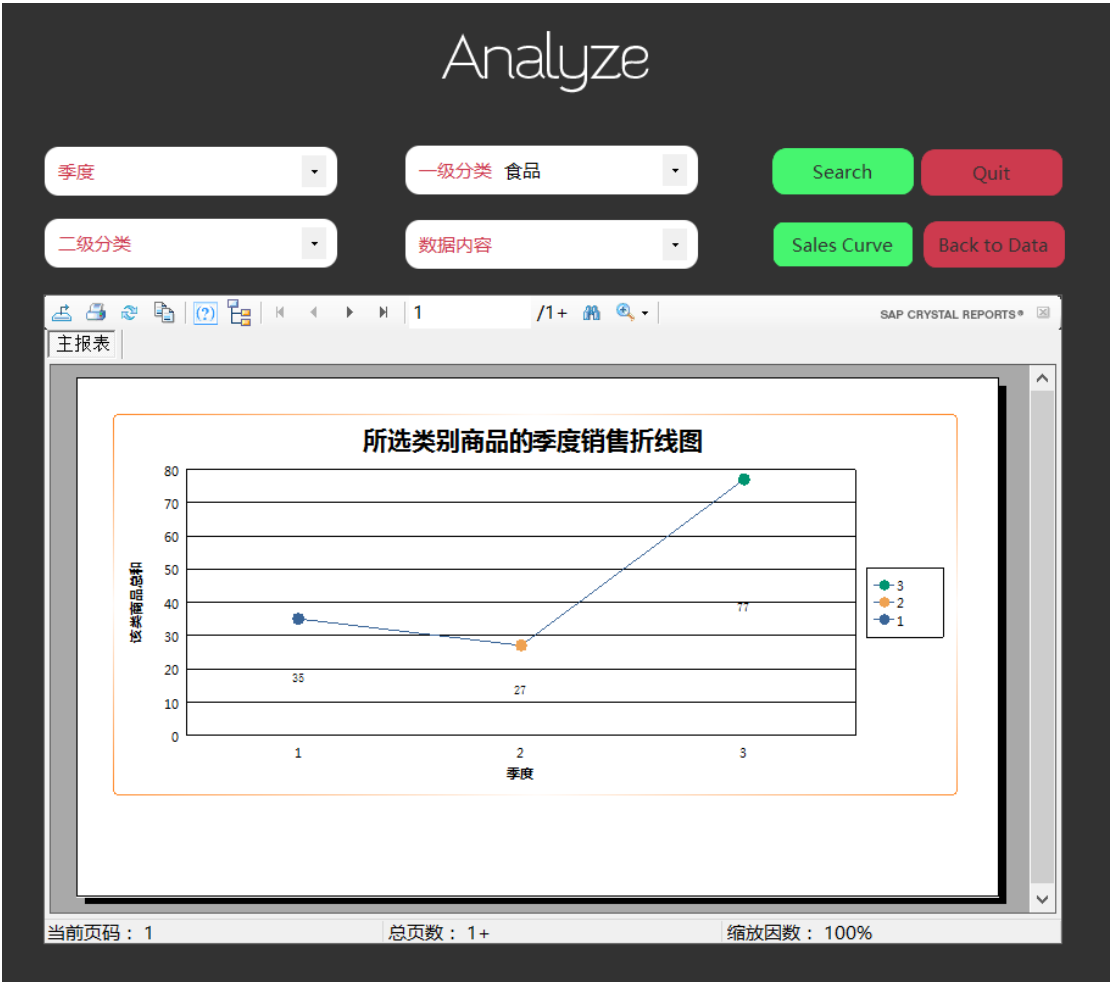


图 3.5.4-2

商品信息的统计功能其实分为两部分，一部分是数据类的统计，另一部分则是图表类的生成。

数据类的统计指的是通过选择季度与一二级分类选择了统计范围，然后再根据数据内容选项选定了要显示的数据（比如利润率最大的商品）。用户先输入数据的查询范围与数据内容然后单击 Search 即可得到想要的统计数据。实现手段其实与条件性查询中的排序方法类似，也是先通过 SQL 语句中的 SELECT 先读出需要统计的数据并保存在一个对象的 List 中，然后用自己编写的 IComparer 接口实现自己需要的排序方式，通过 Sort()实现对 List 的排序。最后读出数组中的第一条数据即为所需统计的最值。

图表类的生成指的是通过选择一二级分类生成销售总量与季度的折线图。用户选择一二级分类后单击“Sales Curve”生成销售曲线，然后单击“Back to Data”回到数据查询模式。实现上使用了 Visual Studio 所集成的水晶报表。但是在开发的过程中发现水晶报表虽然可能很方便的选择需要生成图表的数据库来源并且在确定横纵轴坐标后自动生成各种图表。但是无法做到图表数据来源的动态改变。然而我们必须要实现通过用户选择的数据范围不同生成不同的折线图。一种解决办法是将所有可能性的图都画一遍，然后根据用户的选择显示不同的图表。但这样重复机械的劳动显然是不科学的，也背离了大作业本身的意义。而我后来选择的方式是将水晶报表的数据来源选定为一个全新的数据库 temp，用户选定范围后通过 SQL 的 SELECT 与 INSERT 语句动态更新 temp 数据库内的数据，然后通过对报表控件的刷新实现了图表的动态更改。在水晶报表使用过程中还有另一个问题，因为其动态访问数据库，每次连接时都会弹出登陆框要求用户输入数据库的登陆账号与密码。这一问题如果不修正显然是一个巨大的 BUG，因为实际用户不可能也不应该知道数据库的登陆账号与密码。因此数据库的登陆操作必须集成在软件本身。所以我编写了认证水晶报表的函数，并且在窗口加载这一事件中调用这一函数，实现了水晶报表的后台认证，使得软件更加科学，

使用更加简便。软件中的水晶报表直接包含了折线图、饼状图与柱状图，方便用户查看自己想要的图表类型。但是由于一些原因，我选择了将三个图表直接做在一个页面，而不是通过用户的选择来显示相应的图表。原因在于一旦动态改变图表类型，登录数据库的窗口就再次弹出，我试过许多办法试图解决却未能成功。此处瑕疵为此次软件开发的一个不完美之处，希望通过以后的学习我能找到相应的解决办法。

### 3.6 用户管理模块的编写

#### 3.6.1 用户的注册、登录、信息完善与密码的修改

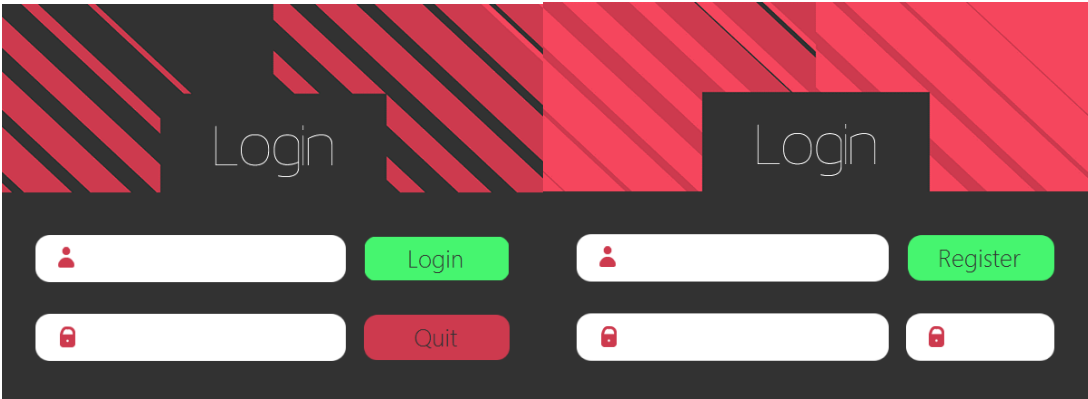


图 3.6.1-1

图 3.6.1-2

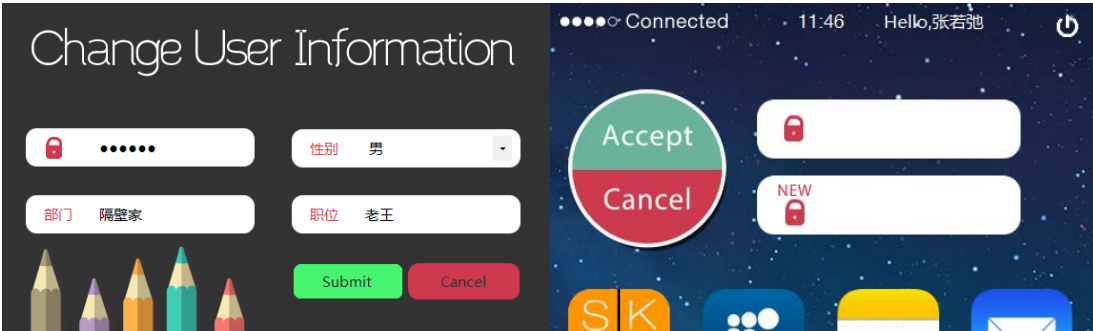


图 3.6.1-3

图 3.6.1-4

登录与注册集成在一个窗口。用户单击动图中央的“Login”字样激活隐藏的注册控件，输入正确的内部代码后，完成注册并登录。首次登录主窗口会提示完善个人信息。同时主窗口集成了修改用户密码的功能，单击用户头像后隐藏的密码修改控件出现即可修改密码。

这几个过程中密码的传输都经过哈希算法加密。考虑到密码的确认在修改密码与修改信

息的时候会多次使用，确认密码的功能被封装为一个独立的函数。同时此处也有很多独立封装的查错函数，包括用户名是否重复，密码长度是否合理等等。不论是用户的信息完善还是密码修改，在本质上都是通过了密码确认函数后，使用 SQL 的 UPDATE 语句修改数据库相应的数据条目。

### 3.6.2 用户权限管理

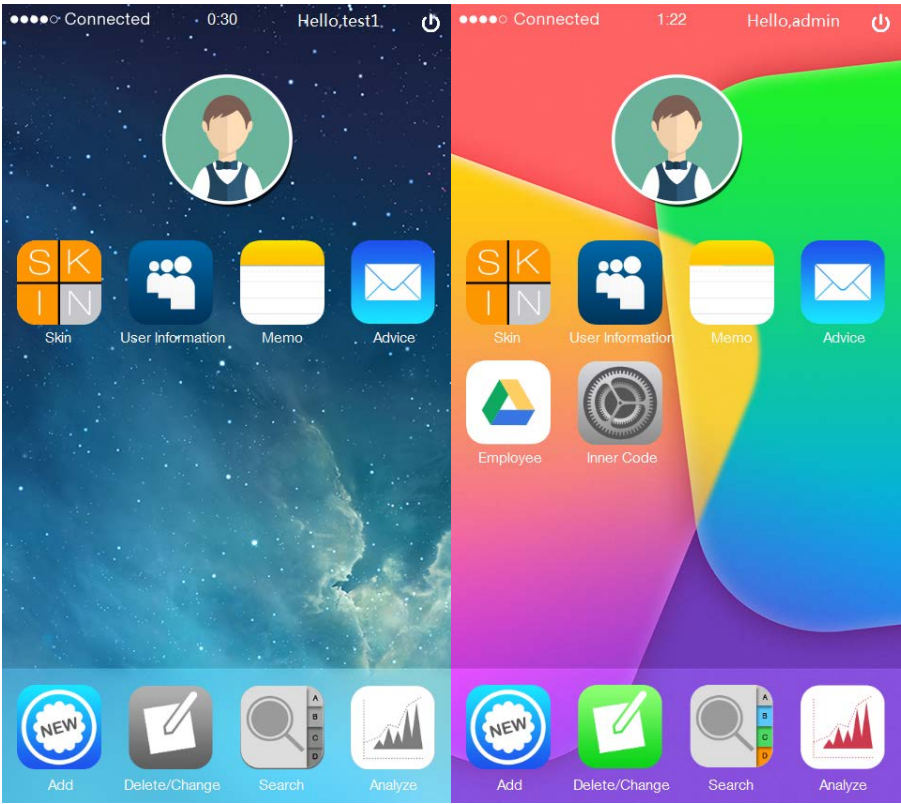


图 3.6.2-1

图 3.6.2-2

不同的用户拥有不同的权限，这在上文已经提及。用户刚注册时权限默认为 4，薪水为底薪 3000 元。这可以通过账号 admin 或任何权限为 0 的账号修改。

不同账号不同权限的实现的方法就是在 Mainfunction 窗体加载时根据用户权限的不同将一些原有功能的图表隐藏，或隐藏后显示一个黑白的图片框以表示功能不可用。如图 3.6.2-1 为一个权限为 4 的用户的登录界面，发现其基本功能中有三个是无法操作的，只能使用新建商品信息的功能。而图 3.6.2-2 则是权限为 0 的用户，除了能实现所有的基本功能

外，还附加了人员管理功能与内部代码修改功能（各应用程序图标从上至下从左至右分别为:Skin——换肤功能 ,User Information——用户个人信息修改 ,Memo——备忘录功能，Advice——建议反馈功能，Employee——员工权限与信息的修改删除，Inner Code——内部代码修改功能，Add——商品信息新建，Delete/Change——商品信息的修改，Search——商品信息的复杂查询，Analyze——商品信息的统计功能）。

3.6.3 用户信息管理

人员管理系统

姓名

性别

Change

Delete

部门

职位

Search

Quit

	姓名	性别	部门	职位	薪水	权限
▶	test1	女	门市部	店员	50000	4
	zjw	女	销售	职员	50000	1
*						

图 3.6.3-1

人员管理系统

姓名 test1 性别 女 Cancel Delete

部门 门市部 职位 店员 Search Quit

权限 4

薪水 50000

Submit

图 3.6.3-2

当登陆的用户权限为 0 时，拥有用户信息管理功能。可以实现对员工权限与薪水的更改，也可以实现对员工的删除操作。具体实现手法与商品信息的修改与删除类似，此处不再赘述。

### 3.6.4 其他附加功能的实现

本软件考虑到超市实际应用的需要还添加了类似换肤之类的个性化美化功能与类似备忘录之类的方便员工使用的功能。同时还有实时时间与现实数据库连接状况的小图标的附加功能。（如图 3.6.4-1 与图 3.6.4-2）。实现方法，换肤功能众多图片框都在一个 `flowlayyoutpanel` 容器中，双击图片框就会使窗口的 `BackgroundImage` 属性发生改变，从而实现换肤功能。备忘录则使用了 `checkedlistbox` 这一控件 实现了备忘录的简单功能。



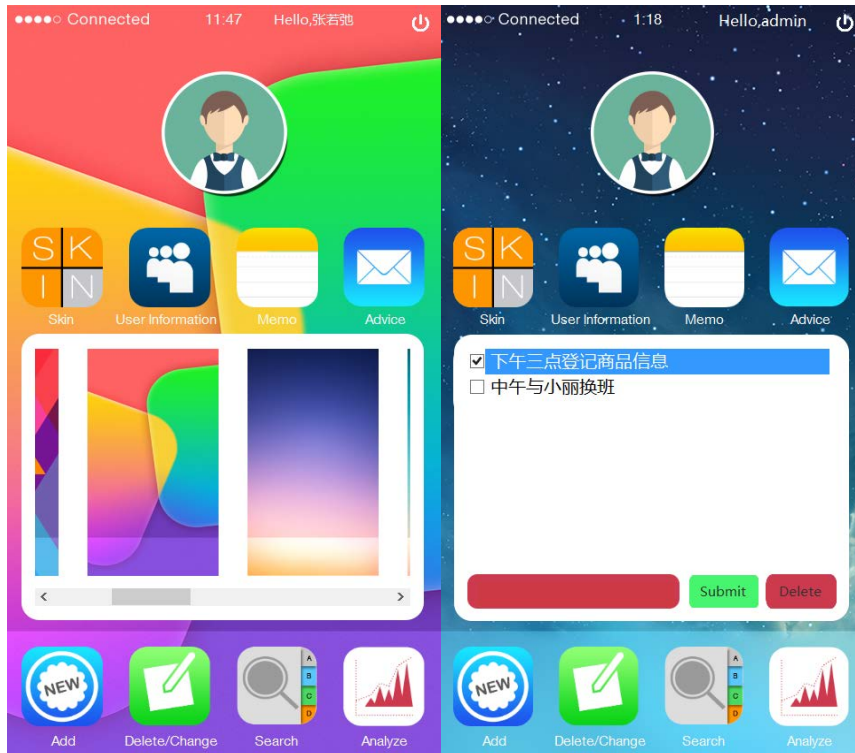


图 3.6.4-1

图 3.6.4-2

## 4.项目总结

### 4.1 版本管理

为了方便版本管理与在多台电脑上修改编写程序，我们使用了 Github 作为工程的管理软件，实现了历史版本的保存（方便错误修改后找回历史版本）与多人开发代码的交互，为我们这次的大作业提供了很多的便利。

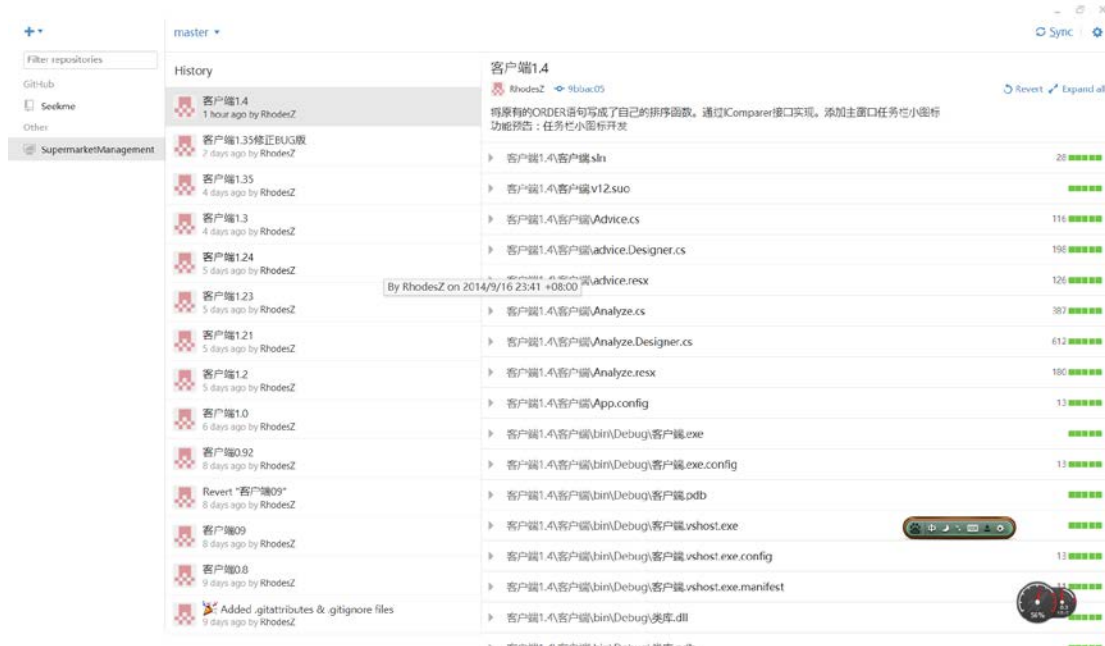


图 4.1-1

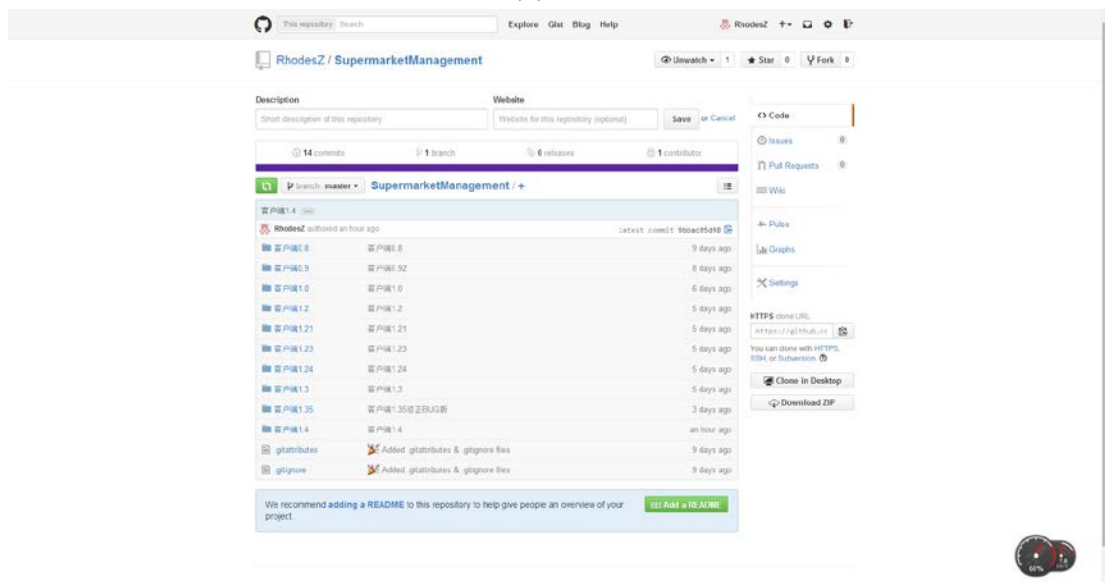


图 4.1-2

在开发的两周内我们从 0.8( 基本实现除了图表之外的所有功能 )到最新的 1.4 版本( 包含目前所有功能并更换了一些不符合大作业规定的语句 )我一共上传了 10 次左右的代码。

Github 作为我们组的一个版本管理软件，见证了这个小程序的发展与成长。

本软件的 Github 网址为：<https://github.com/RhodesZ/SupermarketManagement>

## 4.2 个人感想

虽然我平常对程序应用的开发比较感兴趣，也尝试编写过一些简单的应用程序，但是这次 C++ 大作业的完成经历还是带给了我许多很深的感触。

首先，老师要求我们只能使用简单的数据库读写语句，至于排序之类的操作应该对读到内存中的数据进行操作。一开始，我对这点十分苦恼，认为这是无意义的对工作量的加大。但在自己编写完相关代码后，我发现这一过程使得我能够更深入地思考程序对数据原有的操作过程，加深了我对面向对象思想的理解。一味地使用别人封装好的函数或控件并不能使自己真正学到东西，反而会产生依赖心理，最终连“码农”都不是，只能成为“代码的搬运工”。

其次，是对 UI 的美化与不断完善的用户体验。因为从事社工的种类，我平常对 PS、AI 之类的软件接触较多，也对 UI 设计比较感兴趣。所以这次的开发过程中，我在本子上画下基本控件布局后就使用许多其他软件对全套 UI 进行了美化，后期还结合了一些控件实现了一些小动画。但是在与一些同学交流让他们试用的过程中，他们反馈，有些动画意义不大，并没有起到类似吸引用户注意力或让用户在加载数据过程中耐心等待的作用，同时一开始的控件布局过于冗杂，窗口上显示的控件过多，用户无法准确点击。于是我不断修改 UI，同时在一些细节上（例如窗口的 Acceptbutton 的设置，光标放在按钮上变成手型，各输入框按 TAB 键的切换顺序等）不断完善。最终形成了现在的界面。

最后，组员的合作让我对当前 IT 产业的开发模式有了初步的窥探。如何保证不同人代码风格的统一，如何版本管理，如何与人合作。这点我非常感谢我的队友，她在代码风格的统一上做出了很大的贡献。我们一开始就制订了窗体、类、变量、函数的命名规则以及注释规则，在编写过程中，我有时不太注意会忘记这些规则，队友就会提醒我及时修正不符合命名规则的变量名。合作开发是未来软件开发的必由之路，这次的合作经历让我学会

了如何与人合作，让我不禁期待下一次与同学一起做大作业的机会。