

基于 QT 的汽车销售管理系统的设计与实现

张丞豪 白俊鸽

成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】 21 世纪是计算机的时代。很多手工纸质记录数据的方式已经被淘汰，现在基本全部采用电脑系统来管理和记录数据，销售管理系统是目前很多主流公司管理商品常用的一种方式，本文主要设计的是一款基于 QT 的简易便利的汽车销售管理系统，适合小型企业使用，方便快捷地管理商品数据，对数据进行增删改查等一系列操作。前台使用 QT 做 GUI 界面，后台使用关系型数据库 MySQL 来存储商品数据，并且连接本地服务器，这样做可以最大程度降低成本，并且安全性也可以得到一定程度的保证，因为不需要通过网络进行数据的传输，数据库和服务器都在本地搭建。

【关键词】 管理系统；QT；C++；MySQL

1 引言

目前市场上有很多功能强大的系统平台，但是大多数都价格不菲，这对很多中小型企业很不友好，而且其实大多数的中小型企业对于系统功能并非要求必须非常强大，他们看重的还是成本的高低，成本高了他们无力承担。所以本文设计的这一款汽车销售管理系统主要针对的是中小型企业，对于功能的多样性并没有过多的要求，但对数据的安全性有一定要求。

2 开发环境介绍

Qt 是一个 C++ 图形用户界面应用程序框架，因为它的跨平台性所以很多公司都用它来开发应用程序界面，它提供建立艺术级图像界面所需的所有功能，它也是面向对象的，容易扩展。现在的很多软件都采用 Qt 的技术，如 WPS 等等。Qt 软件提供给图形界面开发者所需要的各种控件和库，以便于更好的实现图形化界面的开发，Qt 软件和 windows 下面的 MFC 是同类型的软件。^[1,2]

MySQL 是很多公司经常使用的关系型数据库，因为它的比较小的体积、速度也很快、也很可靠、方便使用并且它的源代码是开放的，而且支持多用户，很多小型和中型公司都使用它来存储数据。它使用的是标准化的数据库访问语言 SQL。^[3]

本系统主要使用 Qt Creator 作为开发环境，该开发环境支持跨平台开发，遇到不会的问题可以查看帮助文档，帮助文档中有 Qt 所有接口的使用说明，方便学习，初学者也能够快速上手。

3 主要功能描述

本文设计的汽车销售管理系统主要包含六个模块，一共有六个界面。分别是登录、注册、商品管理、订单管理、日志管理以及密码管理。运行程序进入登录界面，在登录界面上可以选择注册账号，点击注册后跳转到注册账号页面，在该页面中输入姓名、年龄、性别以及密码，点击确定弹出对话框注册成功，否则弹出注册失败，之后点击返回退回到登录界面，输入账号密码后，登录成功进入商品管理页面，该页面左边显示商品信息数据：商品的编号、商品的名称、商品的类型、商品的型号、商品的价格以及商品的数量，右边提供了对商品的刷新、增加商品、修改商品、删除商品的操作。

然后点击订单管理，进入订单管理页面，该页面上边显示订单信息数据，操作和商品管理一致，在此就不

做详细说明了，需要说的一点是订单上信息改变会同步更新在商品表中。

之后点击日志管理，进入日志管理界面，该页面显示一张日志表，这张表主要显示订单号、订单日期、订单客户名、订单商品名、订单商品数量、订单商品单价以及订单商品总价。这张日志表的数据随着订单表的更新而更新，每当订单表中增加、修改、删除数据时，日志表中相对应的与订单表同步更新数据。

最后点击密码管理，进入密码管理界面，在该页面中你可以修改该账号的密码，修改成功后下次登录就不能使用旧密码登录了，必须使用新密码才能登录成功。

4 界面设计

登录界面主要显示账号输入框和密码输入框，以及两个按钮登录按钮和退出按钮。并且进行一系列的布局，登录成功后进入销售管理界面，该页面是该系统的核心页面，该界面的菜单栏中有以下 4 个管理块：销售管理、日志管理、密码管理、退出管理。其中销售管理中运用了 Qt 中的 QStackWidget 控件用来进行商品界面和订单界面的切换。商品界面中用了 QTableWidgetItem、QGroupBox、QLineEdit、QComboBox、QLabel、QPushButton 等 Qt 控件。其他的界面中也主要用了以上的某几个控件在此就不一一列举了，还有就是在每个界面的控件中我都加了一些 qss 的样式，每当鼠标经过控件时，都会有相应的样式的改变，主要是通过 Qt Creator 中的 ui 布局界面中的添加样式表来给 ui 界面中的控件增添样式，使得界面看起来更加的美观。如图 1。



图 1：商品管理界面

5 数据库配置

5.1 表

在本地 MySQL 数据库中创建了一个名为 db_qt 的数据库, 在该数据库中创建了四张表: 用户表 (users)、商品表 (commodity)、订单表 (order_form) 以及订单日志表 (sale_log), 具体的字段已在主要功能描述中说明。

5.1 存储过程

为了实现该销售管理系统的增删查改的功能, 在本地 MySQL 中编写了 12 个存储过程, 分别是: sp_Login、sp_ShowCommodity、sp_ShowOrder、sp_ShowSaleLog、

sp_RegisterUser、

sp_ModifyData、sp_ModifyOrder、sp_ModifyPassword、sp_InsertCommodity、sp_InsertOrder、sp_DeleteData 以及 sp_DeleteOrder。

sp_ShowCommodity() 用于显示商品数据, 通过编写 Wampserver 软件包中的 server 文件夹下的 sql.json 文件, 调用 MySQL 数据库中的存储过程, 返回相对应的 json 格式的数据, 前台再通过 jsoncpp 库解析后台返回

的数据, 显示在 QTableWidgetItem 控件中, 代码如图 2。

```

1 DELIMITER $$
2
3 USE `db_qt` $$
4
5 DROP PROCEDURE IF EXISTS `sp_ShowCommodity` $$
6
7 CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_ShowCommodity`()
8 BEGIN
9     SELECT DISTINCT 0 AS result, commodity.c_id AS id, commodity.c_name AS NAME, commodity.c_kind AS kind,
10    commodity.c_type AS TYPE, commodity.c_price AS price, commodity.c_amount AS amount FROM commodity;
11 END $$
12
13 DELIMITER ;

```

图 2: sp_ShowCommodity() 存储过程

6 服务器配置

本服务器因为是部署在本地的, 所以使用的地址是本地地址 127.0.0.1, 端口号配置的是 5240. 因为要连接本地 MySQL 数据库, 所以需要修改 server 文件夹下的 config.json 文件, 在里面配置连接本地的 MySQL 数据库, host 修改成 127.0.0.1, name 修改成 db_qt, password 修改为你连接本地数据库的密码, user 可以是 root 也可以是你创建的用户, 还有就是端口号修改成 MySQL 的端口号 3306。

7 功能实现

7.1 登录

通过 ui->lineEdit->text() 获取用户输入的数据信息, 再通过 toUtf8().data() 接口函数将 QString 类型的数据转换成 char* 类型的数据, 在获取了用户输入的账号和密码之后, 将它们组装成 json 格式的数据, 再通过运用 libcurl 库中的 http 协议将该数据传输至后台数据库, 数据库通过调用相应的存储过程, 返回相应的 json 格式返回值, 前台再通过 jsoncpp 库解析返回的数据, 判断该返回值是否为 0, 如果为 0 则登录成功跳转页面, 否则登录失败重新登录。

7.2 注册管理

依旧通过使用 ui->lineEdit->text() 接口函数, 接收用户输入的数据, 依次输入姓名、年龄、性别以及密码, 然后点击注册, 调用槽函数 on_register_btn_2_clicked(), 通过将 QString 类型的数据转换为 char* 类型的数据, 再将数据打包成 json 格式的数据, 通过 server 文件夹中的 sql.json 文件调用 MySQL 数据库中的 sp_RegisterUser 存储过程, 返回相应的数据, 判断返回值, 如果返回值是 0 的话, 则注册成功。之后点击返回按钮到登录界面。

7.3 商品管理

商品管理中的主要操作是刷新商品、增加商品、修改商品以及删除商品。刷新商品是在构造函数中调用的, 所以系统运行起来后自动调用刷新商品的槽函数, 使得在登录进去之后已经在 TableWidget 控件中显示了数据。增加、修改、删除商品都是使用 ui->lineEdit->text() 以及 ui->comboBox->currentText() 来获取用户输入的信息, 再通过一系列操作将数据打包成 json 字符串的格式, 通过 http 协议传输到后端数据中, 数据库调用相应的存储过程: sp_InsertCommodity、sp_ModifyData 以及 sp_DeleteData, 再通过判断它们相应的返回值, 弹出对话框与用户交互判断是否增加、修改、删除商品成功。

7.4 订单管理

订单管理大致上与商品管理的过程一致, 也是分为刷新订单, 增加订单、修改订单以及删除订单, 都是通过获取用户输入的数据, 将数据打包传输到后端, 在后端调用相应的存储过程, 将结果返回给前台, 前台相应的通过解析返回的数据判断该操作是否成功执行。与商品管理不同的是, 每当订单表中增加、删除或修改一条数据时, 商品表中相应商品的数据也会相应的更新, 这样才符合逻辑, 保证数据的一致性。

7.5 日志管理

日志表的变化, 主要是随着订单表的变化而变化, 每当订单表中插入一个订单信息时, 日志表中便会自动添加一条记录, 该记录中包含了订单 id、添加订单的日期、订单客户名、订单商品名、订单商品数量、订单商品的单价以及订单商品的总价。订单表修改时也会修改日志表中对应的数据。

7.6 密码管理

密码管理中主要是接收用户输入的账号、旧密码、新密码三条数据, 然后调用存储过程, 给前台返回相应

的数据，前台再进行判断看是否修改成功。

8 测试及改进

项目完成之后，我对该系统进行了功能的测试，发现以下问题并进行了总结。

我们在新用户注册时，因为只输入了姓名，年龄，性别，密码，但是用户表中还有个用户 id 字段。所以在创建新用户时，用户表中用户 id 就是空值。还有就是注册时如果注册人的信息在表里已经存在相同的数据，系统会直接报错。

对于以上问题，解决方案如下：在注册界面上再增加一个编辑框，用户输入用户 id，并且修改相应的存储过程。对于用户重复这个问题，首先在注册的存储过程中先判断是否有与将要注册的信息相同的数据，如果有直接返回 -1，否则再进行数据的插入，新增用户。

9 结语

本文主要介绍了如和利用 QT 和 MySQL 数据库来开发

一款汽车销售管理系统，本系统适合中小型企业使用，成本较低，功能相应简洁单一，在使用过程中发现有些地方操作起来比较繁杂，比如我们要新增一个订单时，因为需要填写很多信息，可能记不住商品的价格或者是类型，这时我们如果可以提供一个查询服务的话，便能够使我们增加商品的时候方便记录。

【参考文献】

[1]Jasmin Blanchette MS.C++ GUI Qt3 编程 [M]. (齐亮译). 北京：北京航空航天大学出版社，2006,125-150.

[2]Jasmin Blanchette MS.C++ GUI Qt 4 编程 [M]. (齐亮译). 北京：电子工业出版社，2008,133-150.

[3] 林慧. MySQL 在 PHP 动态网页设计中的应用 [J]. 电子技术与软件工程，2021(06):141-143.