

高级语言程序设计大作业实验报告

王硕 2312456

2024 年 5 月 13 日

目录

1 作业题目	1
2 开发软件	1
3 课题要求	2
4 主要内容	2
4.1 数据库设计	2
4.2 登录 ui 设计	3
4.3 管理 ui 设计	6
5 收获、思考以及我的创作历程	8
6 后面的设想	9

1 作业题目

基于 qt 和 sqlite 数据库的超市管理系统

2 开发软件

Qt6.7.0 + Qt Creator 13.0.0

Sqlite3 + Navicat Premium 16

3 课题要求

- 面向对象
- 数据库设计

4 主要内容

4.1 数据库设计

这里数据库相关信息的设计是单独放在一个.h 和.cpp 文件中的，在设计的时候，我们首先把员工和用户的信息放入相应的结构体里面：

```
1      void Page_Login::on_btn_exit_clicked()
2  struct EmployInfo
3  {
4      int id;
5      QString name;
6      int age;
7      QString address;
8      QString position;
9      int salary;
10 };
11
12 struct UserInfo
13 {
14     QString uername;
15     QString password;
16     QString auth;
17 };
```

Listing 1: Qt 代码示例

之后是数据的增加，删除，更新等操作功能，相应的函数如下：（每个函数的大致功能如备注所示，具体代码见相应的 sql_manage.cpp 文件）

```
1      void init();    //初始化数据库连接
2      explicit sql_manage(QObject *parent = nullptr);
3
4      //针对员工信息
5      quint32 getCount();    //获取员工表中的员工数量
```

```

6      QList<EmployInfo> getPage(quint32 page,quint32 uiCnt);
      //分页查询员工信息并返回指定页码的员工信息列表
7      bool add(EmployInfo);    //添加新的员工信息
8      bool delEmp(int id);    //根据员工的id删除员工的信息
9      void clearStuTable();    //清空员工表的数据
10     void Update(EmployInfo); //更新员工表中的数据
11     QList<EmployInfo> select(QString name); //根据员工姓名查询员工信息
12
13     //针对用户账号
14     QList<UserInfo> getAllUser(); //获取所有用户的信息列表
15     bool isExit(QString strUser); //检查用户的信息是否存在
16     bool UpdateUser(UserInfo info); //更新用户信息
17     bool AddUser(UserInfo info); //增加用户信息
18     bool delUser(QString strUserName); //删除用户信息
19     UserInfo selUser(QString name); //查询用户信息

```

Listing 2: Qt 代码示例

同时通过如下代码实现了一个单例模式，确保程序中只有一个 sql_manage 类的实例存在，并提供了静态方法 getInstance() 来获取该实例。

```

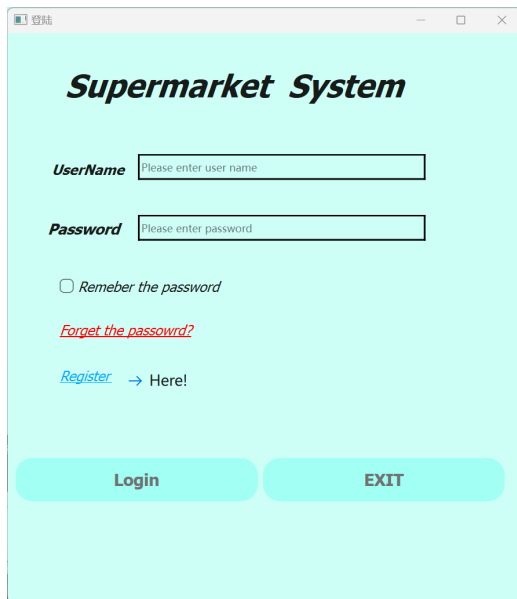
1      static sql_manage *ptrsql;
2      static sql_manage *getInstance()
3      {
4          if(ptrsql==nullptr)
5          {
6              ptrsql= new sql_manage;
7          }
8          return ptrsql;
9      }

```

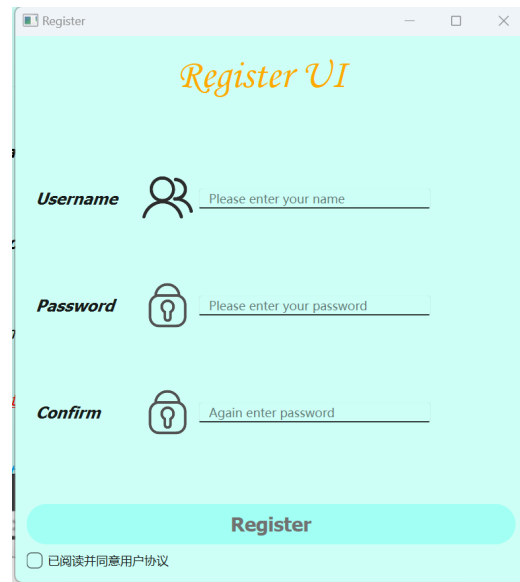
Listing 3: Qt 代码示例

4.2 登录 ui 设计

这里的 ui 界面主要包含两部分，分别是注册界面和登录界面。图示如下：



(a) 登录界面



(b) 注册界面

首先是界面设计的介绍，其中的字体、界面颜色等等的设计主要基于 qss 设计，具体代码参考相应的 ui 设计的代码文件，这里的排版我主要是自己手动设计的，个人感觉 qt 自带的界面排版设计有一点的抽象，总是设计的很习惯，后来索性我就自己一点一点的调整了（强迫症劝退），下面是一些按钮功能的介绍：

登录界面：

关闭操作：

```
1 void Page_Login::on_btn_exit_clicked()
2 {
3     exit(0);
4 }
```

Listing 4: Qt 代码示例

登录操作：

```
1
2 void Page_Login::on_btn_login_clicked()
3 {
4     //数据库查找用户名和密码
5     QString username=ui->_user->text();
6     QString password=ui->password->text();
```

```

7      UserInfo info= ptrUser->selUser(username);
8
9
10
11     //失败就提示 成功进入主界面
12     if(info.username==username&&info.password==password)
13     {
14
15         QMessageBox::information(this, tr("Sucess!"), tr("登陆成功! "),
16                                     QMessageBox::Ok);
17
18         this->hide();
19         emit sendLoginSucess();
20     }
21     else
22     {
23         QMessageBox::warning(this, tr("危险弹窗"),
24                                 tr("警告!!!\n密码或用户名错误! \n请重新输入! "),
25                                 QMessageBox::Ok);
26
27         ui->_user->clear();
28         ui->password->clear();
29     }
30 }

```

Listing 5: Qt 代码示例

进入注册界面的操作:

```

1      void Page_Login::on_commandLinkButton_clicked()
2      {
3          reg.show();
4          this->hide();
5
6
7          auto f= [&]() {
8              this->show();
9          };
10
11          connect(&reg,&page_register::sendRegisterSucess,this,f);
12

```

```
13 }
```

Listing 6: Qt 代码示例

注册界面：

注册操作：

```
1  void page_register::on_registerButton_clicked()
2  {
3      UserInfo info=getinfo();
4      if(info.username.isEmpty())
5      {
6          QMessageBox::warning(this, tr("危险弹窗"),
7                                tr("警告!!!\n两次输入密码不同!"),
8                                QMessageBox::Ok);
9          ui->passwordLineEdit->clear();
10         ui->confirmLineEdit->clear();
11     }
12     else
13     {
14         ptrsql->AddUser(info);
15
16         this->hide();
17         emit sendRegisterSucess();
18     }
19 }
```

Listing 7: Qt 代码示例

剩下的关于 label 和 text 的设计就不在过多描述。

4.3 管理 ui 设计

首先是主界面的设计，相应的展示图如下：

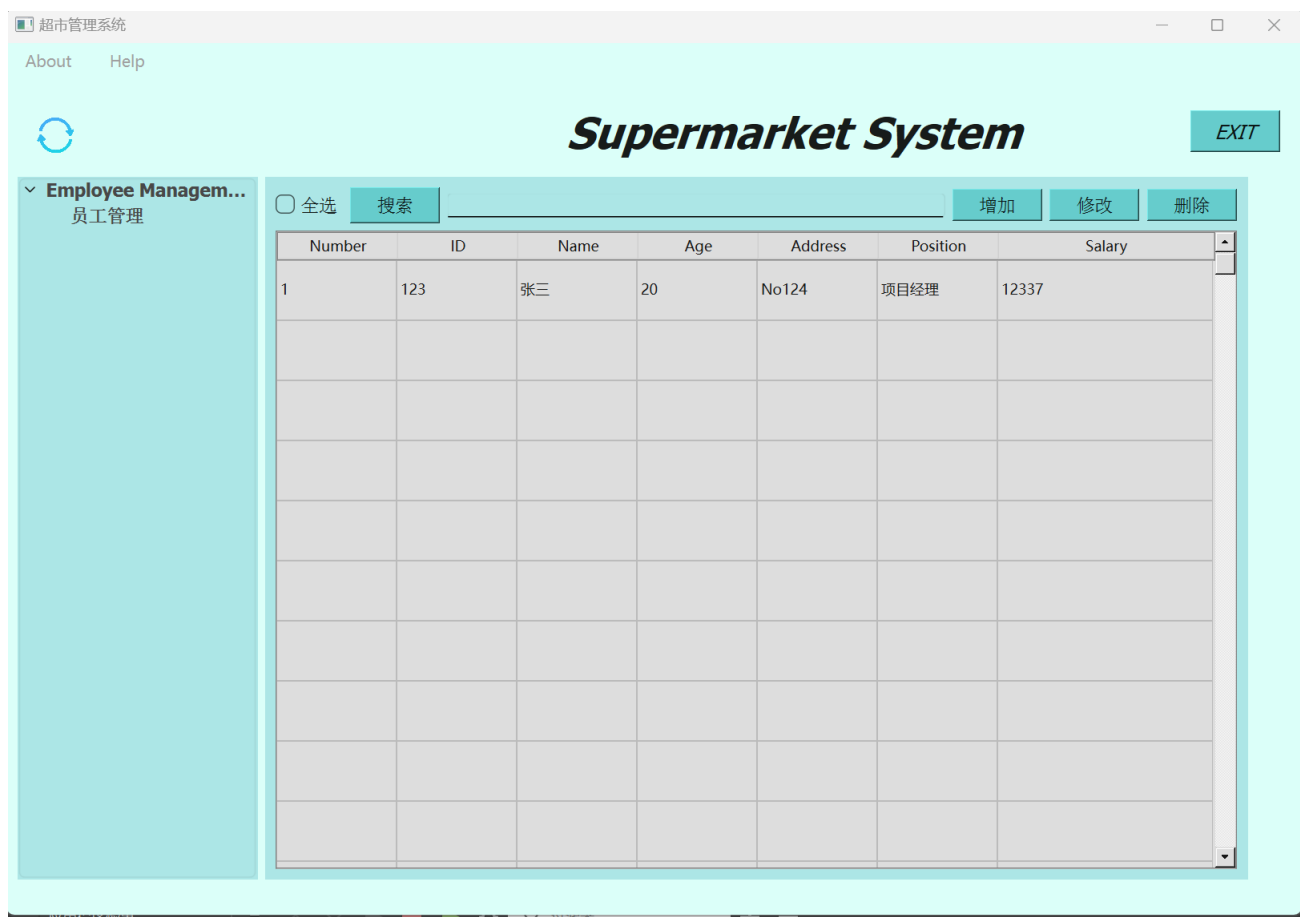


图 1: 主界面示意图

整体来说，这个主界面首先使用了一个 QGridLayout 的布局管理系，将主窗口分成左右两个部分，之后在左侧放置了一个 QTreeWidget 控件用于显示树状结构的数据，右侧放置了一个 QStackedWidget 控件，用与显示多个页面，可以根据需要切换显示不同的页面。之后就是一些常规的按钮，比如增加，修改，删除等等的按钮，还有相应的 itemform 插入在界面中，用于显示用户的信息。按钮的功能就不在这里一一描述，参考相应的 mainwindows.cpp 和 itemform.cpp 文件。

下面是一些按钮功能的介绍：

```

1 void on_deleteButton_clicked(); //删除员工信息
2
3 void on_updateButton_clicked(); //更新员工信息

```

```

4
5 void on_addButton_clicked(); //添加员工信息
6
7 void on_exitButton_clicked(); //退出界面
8
9 void showList(); //从数据库中获取员工信息并在表格中显示出来
10
11 void on_searchButton_clicked(); //查询员工信息
12
13 void on_showButton_clicked(); //重新加载并显示员工信息

```

Listing 8: Qt 代码示例

除此之外就是一个小小的一个增添信息的界面，如下图所示：

图 2: 增添信息界面示意图

其具体的功能参考相应的 `addempdialog.cpp` 文件。

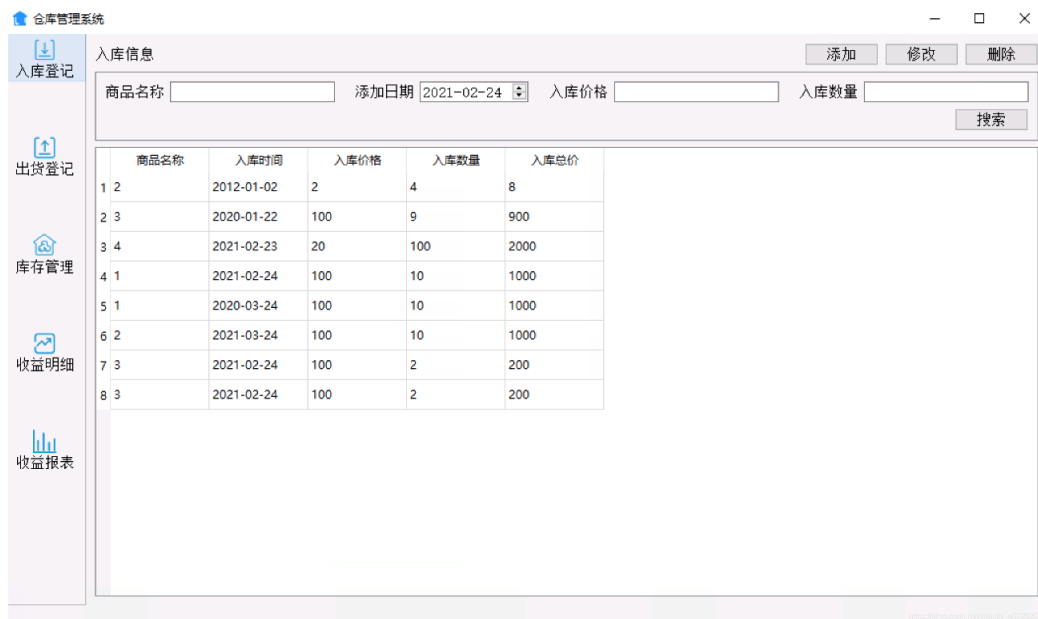
5 收获、思考以及我的创作历程

总体来说，其实这次做 c++ 大作业收获是多的，我也是第一次接触到数据库，当时想做这个的初衷也是想要学习新的知识，可能表现能力比较差吧，没有什么特效，不过总体来说还可

以，能收获到很多也很不错。近几年我看很多人都在做游戏，我就在想能不能换一个思路，做一个比较有实际的东西（主要是游戏的话网上也有很多源代码，而且游戏的题材大多雷同，所以就不太想做），之后在跟高中同学聊天的过程中他跟我提到了这个，于是我就想这个能不能尝试做出来这个，但很显然最终的结果算是失败的吧，只能说做出来一个残次品出来，最终我也没能把数据库和 qt 连接起来（不知道是不是我的电脑的问题，等到我想要换个题材做的时候也已经没有时间了）。开始的时候我用的是 mysql，但是我的 qt 总是显示没有相关 mysql 驱动器（刚开始我连问题都没发现，大概找了一下午才知道自己的问题大致在哪里），之后我就开始去着手解决这个问题，按照网上的教程把相应的 mysql.lib 和另外一个（名字忘了）文件放在 qt 的 bin 目录中，但还是失败，之后又是一段痛苦的搜查问题的过程，然后就知道了 qt 的 6.0 以上的版本是不自带 mysql 的 sqldriver 文件的，而且要自己写.....，可以想象一下我此时的精神状态，好在我没有一股脑写下去，去网上搜了一下发现有的 github 的大佬已经写好了，于是就是下载，给 qt 增添文件的过程，结果一顿操作下来还是不行.... 后面好像是意识到版本不匹配的问题，于是我就把 mysql 的给删了，打算重新下载，结果密码总是不对.... 当时我就崩了，想了想就算了，于是把数据库换成了 sqlite 了，但是后面的结果也是类似的，qt 自带 sqlite 的 qsqldriver 文件的，但到最后我还是没有成功的连接，只能是展示成这样了，不过后边我打算自学一段时间数据库，然后再去完善一下这个超市管理系统吧，感觉这个都快变成我心中的噩梦了。

6 后面的设想

当时由于数据库的连接问题，把整个员工的信息全部记录之后就没再往下写了，其实还有相应的货物管理，参考了一下网上的一些博主做的，找了一个不错的（如下图）：



其实当时就想做这个... 数据库我都在 navicate 里面设计好了，发现不能用（苦笑）（我当时甚至还画出了相应的思维导图...）。之后再尝试着实现吧。

