



南开大学
Nankai University

南 开 大 学

计 算 机 学 院
高级语言程序设计实验报告

Iwork 公司管理系统搭建

姓 名：姚知言

年 级：2022 级

专 业：计算机科学与技术

指导教师：孙宇

2024 年 5 月 12 日

目录

一、 作业题目	1
二、 开发软件	1
(一) 主要开发软件	1
(二) 使用版本	1
(三) 后台数据库以及连接工具	1
三、 主要流程	1
(一) 整体流程	1
(二) 分过程详解	1
1. 主函数以及 pro 函数中完成 qt 与 mysql 的连接	1
2. 主页面 login	2
3. 用户界面整体设计以及 user 界面	3
4. MySQL 的增删改查样例	4
5. 槽逻辑与页面切换逻辑	5
(三) 单元测试	5
1. 登录页面测试	5
2. user 页面	6
3. admin 页面	6
4. 五个增删改查界面	6
四、 收获	8
(一) 总结与收获	8
(二) 仓库及视频传送门	12

一、 作业题目

Iwork 公司管理系统搭建 by Qt

二、 开发软件

(一) 主要开发软件

Qt Creator 13.0.0

(二) 使用版本

Qt 6.8.0

MSVC2019

(三) 后台数据库以及连接工具

MySQL Server 8.0.35

MySQL Workbench 8.0.36

ODBC 8.0.33

三、 主要流程

(一) 整体流程

建立登录页面，与 MySQL 数据库相连，登陆页面分为普通员工用户登录和管理员用户登录。普通员工用户的信息存储在数据库中。普通员工登录后可以看到自己的信息，管理员登录后可以对各个表（主要是 employee 雇员表和 department 部门表，其中雇员表的部分属性存在子类表 management 和 technician）进行增加，部门表与雇员表的部门 ID 项有外键约束关系，即雇员的部门 ID 必须在部门表中，部门删除后部门下的雇员也会删除。

主要界面有 8 个：登录界面，普通雇员界面，管理员主界面，以及 5 个分别对应每个表操作的操作界面。

(二) 分过程详解

1. 主函数以及 pro 函数中完成 qt 与 mysql 的连接

qt with mysql

```
1 //在pro函数中增加
2 QT+=sql
3 //在main.cpp中增加
4 QSqlDatabase db = QSqlDatabase::addDatabase("QODBC");
5 db.setHostName("127.0.0.1");
6 db.setPort(3306);
7 db.setDatabaseName("myodbc");
8 db.setUserName("root");
9 db.setPassword("1433223aaa");
```

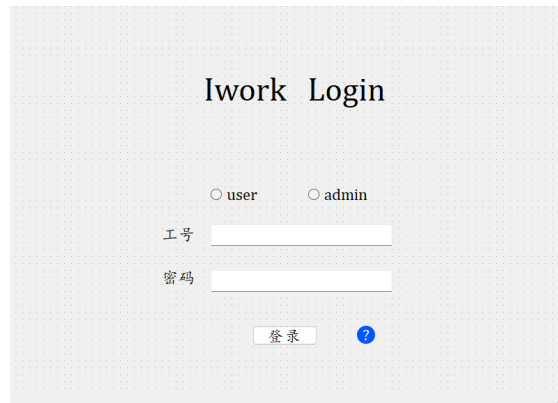


图 1: login 整体设计

```

10  bool ok = db.open();
11  if(ok){
12      w.show();
13  }
14  else{
15      QMessageBox::information(NULL,"失败","数据库连接失败!");
16      qDebug()<<"error_open_database_because"<<db.lastError().text();
17  }

```

通过以上函数，可以完成数据库的连接以及连接验证。

2. 主页面 login

作为整个实验的开始，在该页面中实现了以下功能：

a. 背景图片以及窗口标题

通过在窗格的构造函数中添加以下代码，可以完成对窗格背景图片以及窗口标题的实现。

背景图片以及窗口标题

```

1  this->setWindowTitle("Iwork_Login");
2  this->setStyleSheet("QMainWindow{background-image:url(F:/C/bighomework
    /sunrise.jpg);}");

```

在其他界面中也有类似函数的窗口标题设计，不再一一列出。

b. 页面整体设计

login 页面如图1所示。在 login 页面中，主要实现了单选框 user 和 admin 来选择是以普通雇员登录还是 admin 管理员登录。同时输入框，按钮等元素，是内容清晰易懂。虽然整体图片看起来较为寡淡，但是搭配上背景图片整体颜色较为明丽。

c. 按钮的 css 样式表设计

背景图片以及窗口标题

```

1  QPushButton#help {
2      color:# fffffff;
3      border-radius: 13px;
4      min-width: 26px;
5      min-height: 26px;

```

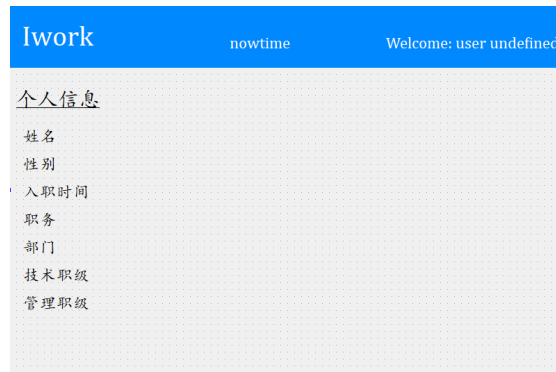


图 2: user 界面

```

6     max-width: 26px;
7     max-height: 26px;
8         background-color:#0055ff;
9 }

```

这样的样式表控制了按钮的大小，使得长和宽都是边框的两倍可以使得按钮呈现圆形。在后续的文本框中也有一些使用了类似的样式表设计，不再列出。

d. 成功登录逻辑

对于普通雇员用户，若其账号和密码可以在数据库中查到，则可以登录。对于 admin 用户，预留了默认账号密码与之匹配，匹配成功则登录。若用户类型选择错误，无法登录。登录失败会有弹窗。整体操作在 button 的槽函数中实现。

e.QMessageBox 的语法实现

QMessageBox 的语法实现如下图所示。

QMessageBox

```

1 QMessageBox::information(NULL,"用户名或密码错误","请检查后重新输入！（您正在尝试管理员用户登录，请确认您的登录类型）");

```

该语法在后续数据库操作检查中频繁应用，以提醒用户数据库操作的成功与否。

3. 用户界面整体设计以及 user 界面

a. 页面整体设计

user 界面如图2所示。在该界面中，蓝色上边栏主要展示了 Iwork 的 logo，中间实现的是当前时间的显示（后面列出），最右边实现的是当前用户 ID 的显示的欢迎语。后续 admin 界面的上边栏设计类似，这一设计将会在此部分的后续内容中说明，不再在后续部分重复列出。

b. 用户 ID 传值：页面之间通信

当用户登录成功后，需要登录界面和用户主界面之间建立通信传递用户 ID，需要建立一个通信槽。

QMessageBox

```

1 //login ,cpp传递
2 connect(this,SIGNAL(sendsignal(QString)),us,SLOT(recsignal(QString)));
3 QString S=ui->getnum->text();
4 emit sendsignal(S);

```

```

5 //user.cpp在此函数下接收，接收后使用受到的结果在数据库进行查询
6 void user::recsignal(QString msg)
7 //user.h
8 private slots:
9 void recsignal(QString msg);
10 //login.h
11 signals:
12 void sendsignal(QString msg);

```

c. 当前时间时钟的实现

时钟在用户界面中被广泛的应用，以下是时钟的实现方法。

QMessageBox

```

1 //头文件
2 public slots:
3 void time(void);
4 //源文件
5 void user::time(){
6 QDateTime dateTime= QDateTime::currentDateTime();
7 QString tt=dateTime.toString("yyyy-MM-dd_hh:mm:ss");
8 ui->tstr->setText(tt);
9 }
10 //以下代码在源文件的构造函数末尾实现
11 QTimer *timer = new QTimer(this);
12 connect(timer,SIGNAL(timeout()),this,SLOT(time()));
13 timer->start(1000);

```

通过这种方法，可以获得每秒刷新的当前时间时钟。

4. MySQL 的增删改查样例

以下给出 MySQL 的增删改查范式，在本次作业中，涉及到四个表的操作，以下给出增删改查的样例，本实验的主要流程为各个组件的槽函数和状态函数的调用与 MySQL in QT 多表嵌套查询逻辑。

MySQL 增删改查样例

```

1 //查找
2 QSqlQuery query;
3 query.exec("select * from employee");
4 while (query.next()) {
5 //在此函数中逐行输出每个元组，通过逻辑比较选择自己需要的元组进行操作
6 }
7 //增加
8 QSqlQuery query;
9 query.prepare("INSERT INTO employee (employeeID,employname) VALUES (:
    value1, :value2)");
10 query.bindValue(":value1",val);
11 query.bindValue(":value2", ui->u2->text());
12 if (!query.exec()) {

```

```

13     QMessageBox::critical(NULL, "错误", "数据库错误, 插入失败, 请检查输入!");
14     return;
15 } // 完成增加的同时检查是否成功, 若失败及时返回错误信息
16 // 更新
17 QSqlQuery query1;
18 query1.prepare("UPDATE employee SET enrollmentdate=:value1 WHERE
    employeeID=:value2");
19 query1.bindValue(":value1", ui->u4->text());
20 query1.bindValue(":value2", val);
21 if (!query1.exec()) {
22     QMessageBox::critical(NULL, "错误", "插入入职时间失败, 但用户已创建, 请检查输入后在用户修改中操作!");
23 }
24 // 删除
25 QSqlQuery query;
26 query.prepare("DELETE FROM department WHERE departmentID=:value");
27 query.bindValue(":value", val);
28 if (!query.exec()) {
29     QMessageBox::critical(NULL, "错误", "移除部门失败, 请检查并更改");
30 }

```

5. 槽逻辑与页面切换逻辑

QT 中调用元件可以通过转到槽的方式自动生成槽函数的框架, 从而当页面受到满足槽函数的操作的时候槽函数会进行调用, 以下是一个简单的切换页面按钮的例子。在本实验中, 通过槽逻辑可以完成绝大部分操作, 也是 QT 的一大特色。

槽样例

```

1 void depop::on_pushButton_clicked()
2 {
3     admin* ad=new admin;
4     this->close();
5     ad->show();
6 }

```

同时, 在此处也显示了 QT 的切换页面逻辑, 即建立了新页面的指针 (需要 include 头文件), 然后将老页面关闭, 新页面打开。在前文提到需要两个页面通信的时候, 也可以不将老页面关闭, 而是仅仅使用 hide 将其隐藏起来不再显示, 但这样就不会释放内存空间。

(三) 单元测试

1. 登录页面测试

在登录界面输入错误的时候, 会产生如图3所示的弹窗。

在登录页面输入正确的情况下, 会根据两种不同登录类型分别进入 user 页面和 admin 页面。

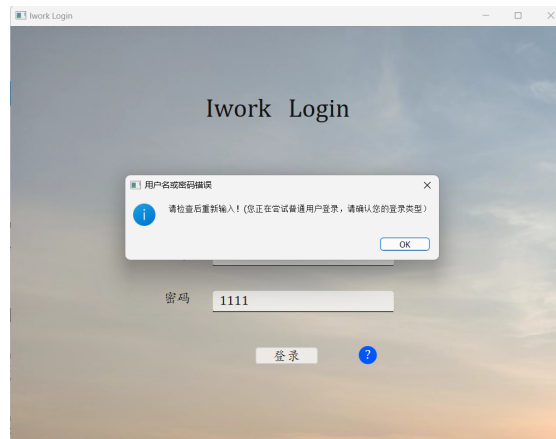


图 3: 登录页面输入错误

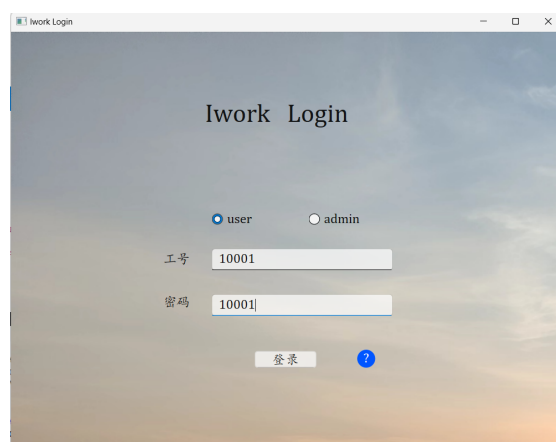


图 4: 登录页面 user 输入正确

2. user 页面

输入如4先前数据库中存在的账号密码，即可进入 user 页面，如图5所示。初始化代码和数据库搭建代码会在 github 和 gitee 仓库中给出。不过图中账号已经在 admin 中进行过一些修改，所以和初始化结果不完全相同。

通过在 admin 创建的雇员也可以完成登录，由于 admin 未设计创建 pwd，新创建的雇员可以以空密码登录，也必须以空密码登录。

可以看到，左边中间正确显示了当前时间，右上角正确显示了 userID，下方正确显示了 user 的个人信息。

3. admin 页面

通过输入预留的 admin 账号密码，正确进入 admin 页面，如图6所示。

可以看到，图中各组件正常显示，下方分别列出了五个常用功能页面及说明。

4. 五个增删改查界面

经过验证，各界面与主界面之间切换正常。

如图7所示，查看部门发现除了默认部门 0 以外还有 333 和 10001 两个部门。因此，通过增加新员工的方式创建如图8员工，没有报错，创建成功。



图 5: user 页面



图 6: admin

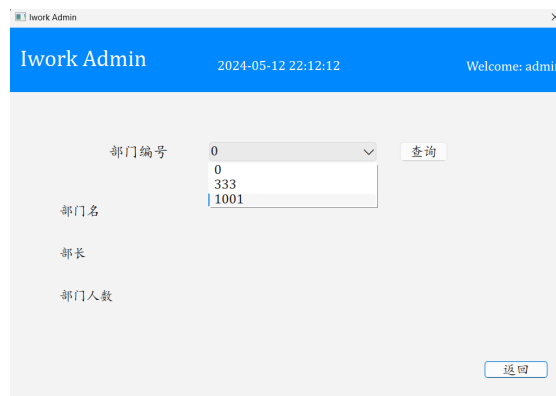


图 7: 查看部门

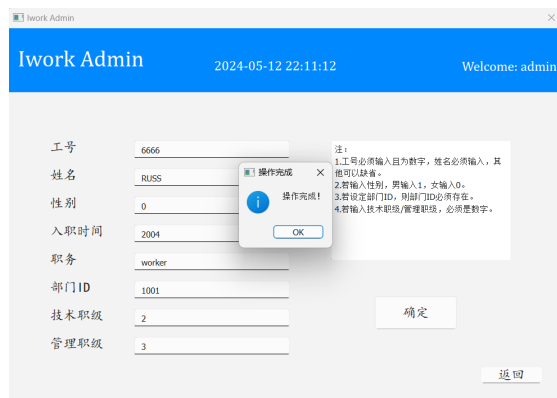


图 8: 增加 employee

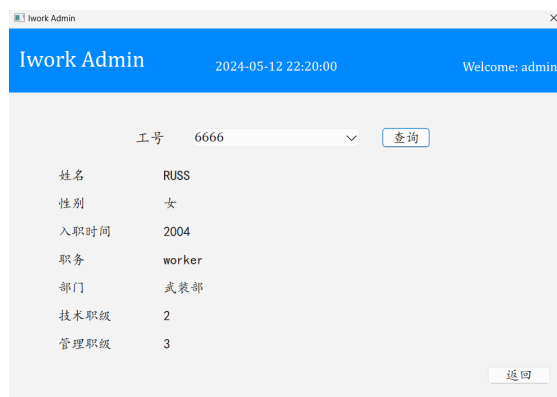


图 9: 查看雇员

完成后，查看该雇员，如图9所示，验证了添加和查询的功能。

表内还有雇员 1 和 11111，如图10,11所示。我们删除雇员 1，并更改一部分雇员 11111 的属性，如图12和图13所示。

图13展示了修改雇员 11111 的流程和第一个弹窗的显示。在修改雇员 1111 弹窗 1 关闭后，还会有修改部门失败的报错，由于该部门 ID 不存在，最后完成修改，结果如图14所示，从图中可以看到未报错的修改已经完成，且 1 已经被删除。

如图15所示，我们添加一个部门，由于部长不在 employee 表中，所以未能成功设置部长，成功报错。部门也成功创建。在此基础上，我们对该部门进行修改，如图16所示，成功执行，没有错误。此时部门 1111 被成功创建，部长 11111 的部门也被修改到部门 1111，如图17所示。

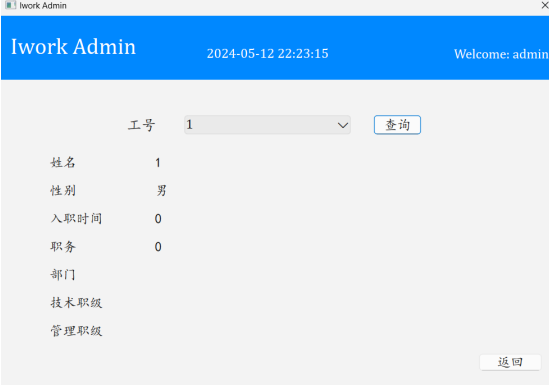
此时，删除部门 1111，可以发现部门 1111 和部门内唯一成员 11111 都被删除了，如图18, 19所示。

至此，程序的每个单元都得到了正确的验证。

四、收获

(一) 总结与收获

在本实验中，我尝试使用 qt 进行图形化编程，并将 qt 与 mysql 数据库完成连接，通过 qt 远程读写数据库。这是一段非常有意义的经历，使得我对 c++ 的类与对象相关内容更加熟悉，也增加了我对于编程学习的乐趣。



The screenshot shows the 'Iwork Admin' window with a blue header bar containing the title 'Iwork Admin', the timestamp '2024-05-12 22:23:15', and the user 'Welcome: admin'. Below the header, there is a search bar with '工号' (Employee ID) and the value '1', followed by a '查询' (Search) button. The main area displays the following information:

姓名	1
性别	男
入职时间	0
职务	0
部门	
技术职级	
管理职级	

At the bottom right, there is a '返回' (Return) button.

图 10: 雇员 1 个人信息




The screenshot shows the 'Iwork Admin' window with a blue header bar containing the title 'Iwork Admin', the timestamp '2024-05-12 22:36:02', and the user 'Welcome: admin'. Below the header, there is a search bar with '工号' (Employee ID) and the value '11111', followed by a '查询' (Search) button. The main area displays the following information:

姓名	cathy
性别	男
入职时间	0
职务	111
部门	
技术职级	5
管理职级	6

At the bottom right, there is a '返回' (Return) button.

图 11: 雇员 11111 个人信息



The screenshot shows the 'Iwork Admin' window with a blue header bar containing the title 'Iwork Admin', the timestamp '2024-05-12 22:26:53', and the user 'Welcome: admin'. Below the header, there is a search bar with '工号' (Employee ID) and the value '1', followed by a '查询' (Search) button. The main area displays the following information:

工号	1
姓名	
性别	
入职时间	
职务	
部门ID	
技术职级	
管理职级	

At the bottom right, there are buttons for '修改' (Modify), '删除' (Delete), and '返回' (Return). A modal dialog box is open in the center, titled '操作完成!' (Operation Completed!), with an 'OK' button. To the right of the modal, there is a note:

注:

1. 工号必须输入且为数字, 其他可以缺省, 以工号来选择删除和删除, 删除只删除工号。
2. 若输入性别, 男输入1, 女输入0。
3. 若设定部门ID, 则部门ID必须存在。
4. 若输入技术职级/管理职级, 必须是数字。
5. 不需填的留空不用输入, 需填的左边是空值或右边的checkbox box, checkbox必须先填数字输入框。

图 12: 删除雇员 1

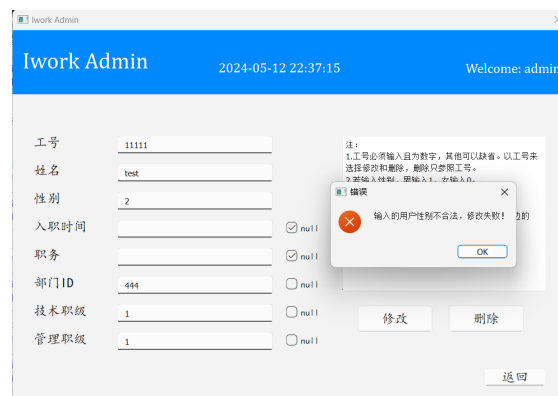


图 13: 修改雇员 11111, 以及第一个弹窗



图 14: 修改删除后结果



图 15: 添加部门



图 16: 修改部门



图 17: 修改部门后 11111 个人界面

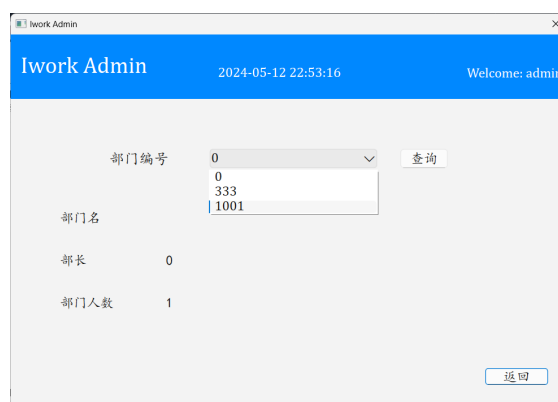


图 18: 删除 1111 后部门查询

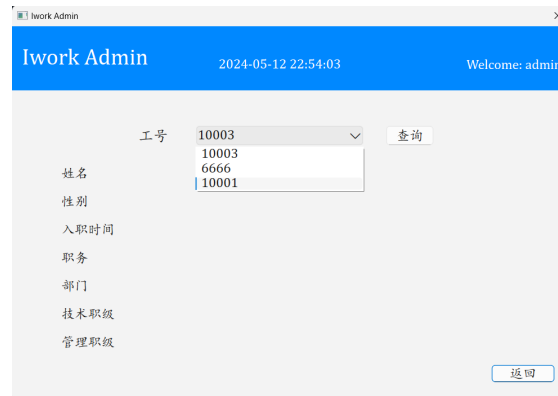


图 19: 删除 1111 后雇员查询

(二) 仓库及视频传送门

[Github 仓库](#)

[Gitee 仓库](#)

[bilibili 个人主页](#)