C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\admindlg.h:28

#ifndef ADMINDLG\_H

#define ADMINDLG\_H

#include <QDialog>

namespace Ui {

class AdminDlg;

}

class AdminDlg : public QDialog

{

Q\_OBJECT

public:

explicit AdminDlg(QWidget \*parent = 0);

~***AdminDlg***();

public slots:

void **onQuery**();

void **onLogout**();

void **onClear**();

signals:

void **startMainWnd**();

private:

Ui::AdminDlg \*ui;

};

#endif // ADMINDLG\_H

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\database.h:18

#ifndef DATABASE\_H

#define DATABASE\_H

#include "macro.h"

#include <Qtsql/QSqlDatabase>

#include <Qtsql/QSqlQuery>

#include <QtSql/QSql>

#include <QDebug>

//根据数据库定义宏,连接数据库

extern int **connDataBase**();

//根据传入QSqlQuery和sql语句查询(游标,sql语句)

extern void **sqlExec**(QSqlQuery& query, QString sql, int col=0);

#endif // DATABASE\_H

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\macro.h:12

#ifndef MACRO\_H

#define MACRO\_H

//数据库信息相关宏

#define DB\_DRIVERS "QMYSQL"

#define HOST "127.0.0.1"

#define DB\_NAME "ticketmaster"

#define DB\_USERNAME "root"

#define DB\_PASSWD "lyh2o16"

#endif // MACRO\_H

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\mainwindow.h:31

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include "userdlg.h"

#include "admindlg.h"

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = 0);

~***MainWindow***();

private slots:

void **onLogin**();

private:

Ui::MainWindow \*ui;

AdminDlg adminWnd; //管理窗口

UserDlg userWnd; //购票窗口

};

#endif // MAINWINDOW\_H

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\userdlg.h:34

#ifndef USERDLG\_H

#define USERDLG\_H

#include <QDialog>

#include <Qtsql/QSqlQuery>

namespace Ui {

class UserDlg;

}

class UserDlg : public QDialog

{

Q\_OBJECT

public:

explicit UserDlg(QWidget \*parent = 0);

~***UserDlg***();

public:

QString userName;//保存用户名

public:

void **updateList**();

signals:

void **startMainWnd**();

public slots:

void **onBuy**();

void **onRefund**();

void **onLogout**();

void **onChooseChange**();

private:

Ui::UserDlg \*ui;

};

#endif // USERDLG\_H

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\admindlg.cpp:68

#include "admindlg.h"

#include "ui\_admindlg.h"

#include <QSqlQuery>

#include "database.h"

AdminDlg::**AdminDlg**(QWidget \*parent) :

QDialog(parent),

ui(new Ui::AdminDlg)

{

ui->setupUi(this);

//设置初始窗口特性

ui->date\_query->setDate(QDate::currentDate());//默认为当前日期

ui->date\_query->setCalendarPopup(true);//设置日期弹出式选择

ui->date\_query->setDisplayFormat("yyyy-MM-dd");//设置日期显示格式

ui->edit\_reslut->setReadOnly(true);//设置文本框只读

ui->edit\_reslut->setFont(QFont("Timers" , 28 , QFont::Bold));//设置文本框字体格式

//信号与槽关联

connect(ui->btn\_query,SIGNAL(clicked(bool)),this,SLOT(onQuery()));//点击查询,响应onQuery()函数

connect(ui->btn\_clear,SIGNAL(clicked(bool)),this,SLOT(onClear()));//点击清除响应onClear()函数

connect(ui->btn\_logout,SIGNAL(clicked(bool)),this,SLOT(onLogout()));//点击退出登录响应onLogout()函数

}

AdminDlg::~***AdminDlg***()

{

delete ui;

}

//槽:响应查询按钮

//根据控件的日期和查询类型和优惠选项统计各种类别的销售情况

void AdminDlg::**onQuery**()

{

QSqlQuery query;

QString sql;

QString queryDate=ui->date\_query->date().toString("yyyy-MM-dd");//设置日期格式

QString dateType=ui->cobox\_queryType->currentText();//查询的日期类型,按日或按月

QString choose=ui->cobox\_choose->currentText();//查询的优惠选项5个之一

QString sales=0;//查询的总销售额

QString number=0;//查询的总销售量

if(dateType=="按日查询")

sql="call procCntDaySold('"+queryDate+"','"+choose+"',"+"@outval1, @outval2);";

else if(dateType=="按月查询")

sql="call procCntMonthSold('"+queryDate.mid(5,2)+"','"+choose+"',@outval1, @outval2);";

qDebug()<<sql;

query.exec(sql);//执行sql

query.exec("select @outval1, @outval2;");

query.next();

qDebug()<<query.value(0).toString()<<query.value(1).toString();

sales="销售额:"+query.value(0).toString()+"\n";//取查询结果的第一行0列

number="销售量:"+query.value(1).toString();//1列

ui->edit\_reslut->setText(sales+number);//设置文本框文本

}

//槽:点击退出登录,关闭本窗口,显示登录主窗口

void AdminDlg::**onLogout**()

{

this->close();//关闭当前adminWnd窗口

emit startMainWnd();//发送startMainWnd信号

}

//槽:点击清空, 可清空文本框

void AdminDlg::**onClear**()

{

ui->edit\_reslut->clear();//清空文本框内容

}

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\database.cpp:38

#include "database.h"

#include "macro.h"

#include <Qtsql/QSqlDatabase>

#include <Qtsql/QSqlQuery>

#include <QtSql/QSql>

#include <QDebug>

//根据数据库定义宏,连接数据库

int **connDataBase**()

{

int ret=-1;

//打印可用数据库驱动

QStringList drivers = QSqlDatabase::drivers();

foreach(QString driver, drivers)

qDebug() << "\t驱动可用:"<< driver;

qDebug() << "End";

QSqlDatabase db = QSqlDatabase::addDatabase(DB\_DRIVERS);

db.setHostName(HOST);

db.setDatabaseName(DB\_NAME);

db.setUserName(DB\_USERNAME);

db.setPassword(DB\_PASSWD);

if(db.open())

{

qDebug() << "数据库打开成功!";

ret=-1;

}

else

{

qDebug() << "数据库打开错误!";

ret=0;

}

return ret;

}

//根据传入QSqlQuery和sql语句查询(游标,sql语句)

void **sqlExec**(QSqlQuery& query, QString sql, int col)

{

qDebug() << "当前执行:"<<sql;

query.prepare(sql);//指定sql语句

if(!query.exec())//执行查询

{

qDebug()<<"执行错误";

// qDebug()<<sqlError;

}

else

{

qDebug()<<"执行结果:";

while(query.next())

{

for(int i=0;i<col;++i)

qDebug()<<query.value(i).toString();

qDebug()<<"\n";

}

}

}

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\main.cpp:15

#include <QApplication>

#include <QDebug>

#include "mainwindow.h"

#include "database.h"

#include <QSqlRecord>

int **main**(int argc, char \*argv[])

{

QApplication a(argc, argv);

connDataBase();

MainWindow mainWnd;

mainWnd.show();

return a.exec();

}

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\mainwindow.cpp:48

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QMessageBox>

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

//设置MainWindow初始特性

//设置信号和槽

connect(ui->btn\_login,SIGNAL(clicked()),this,SLOT(onLogin())); //点击login,执行onLogin

connect(ui->btn\_cancel,SIGNAL(clicked(bool)),this,SLOT(close())); //点击取消,关闭窗口

connect(&userWnd,SIGNAL(startMainWnd()),this,SLOT(show())); //来自userWnd窗口的信号,显示主窗口

connect(&adminWnd,SIGNAL(startMainWnd()),this,SLOT(show())); //来自adminWnd窗口的信号,显示主窗口

}

MainWindow::~***MainWindow***()

{

delete ui;

}

//槽:主窗口,点击登录按钮,响应槽

void MainWindow::**onLogin**()

{

QString userName=this->ui->edit\_user->text();

if(userName=="admin") //只有一个管理员/营销员=admin,否则为用户

{

this->adminWnd.setWindowTitle(QString("营销员\_admin"));

this->adminWnd.show(); //跳转管理窗口

this->close(); //关闭登录主窗口

}

else if(userName=="user")

{

this->userWnd.userName=userName; //将用户名初始化

this->userWnd.setWindowTitle(QString("用户购票\_")+userName);

this->userWnd.show(); //跳转用户购票窗口

this->close(); //关闭登录主窗口

}

else

{ //弹出消息, 用户名不存在

QMessageBox::information(this,"user does not exist","User does not exist, please re-enter!");

}

}

C:\Users\ATWN\_Yhlee\Desktop\TicketMaster\code\TicketMaster\userdlg.cpp:126

#include "userdlg.h"

#include "ui\_userdlg.h"

#include <QDebug>

#include "database.h"

#include <QSqlQueryModel>

#include <QMessageBox>

UserDlg::**UserDlg**(QWidget \*parent) :

QDialog(parent),

ui(new Ui::UserDlg)

{

ui->setupUi(this);

//设置初始窗口特性

ui->date\_useDate->setDate(QDate::currentDate());//默认为当前日期

ui->date\_useDate->setCalendarPopup(true);//设置日期弹出式选择

ui->date\_useDate->setDisplayFormat("yyyy-MM-dd");//设置日期显示格式

ui->tableView->setSelectionBehavior(QAbstractItemView::SelectRows);//设置整行选中

ui->lab\_money->setStyleSheet("background-color: rgb(250, 0, 0);font-size:60px;color:blue");

//设置信号和槽

connect(ui->btn\_buy,SIGNAL(clicked(bool)),this,SLOT(onBuy())); //点击buy,执行onBuy

connect(ui->btn\_refund,SIGNAL(clicked(bool)),this,SLOT(onRefund())); //点击refund,执行onRefund

connect(ui->btn\_logout,SIGNAL(clicked(bool)),this,SLOT(onLogout())); //点击退出登录,执行关闭本窗口, 启动MainWnd

connect(ui->cobox\_Discount,SIGNAL(currentIndexChanged(int)),this,SLOT(onChooseChange())); //当优惠选项改变时候改变下面金额

//设置初始化操作

this->updateList();//从数据库更新列表

this->onChooseChange();//刷新金额

}

UserDlg::~***UserDlg***()

{

delete ui;

}

//槽:userdlg窗口,点击购票,实现购票逻辑

void UserDlg::**onBuy**()

{

QDate tmpDate=ui->date\_useDate->date();

QString useDate=tmpDate.toString("yyyy-MM-dd");//根据日期

QString choose = ui->cobox\_Discount->currentText();//根据优惠选项

if(tmpDate>=QDate::currentDate())

{

//将执行SQL语句: call procBuyTicket(useDate,choose,user)

QString sql= "call procBuyTicket('"+useDate+"','"+choose+"','user');";//调用购票存储过程

QSqlQuery query;

sqlExec(query,sql,3);

}

else

{

qDebug()<<"超过游览时间, 不能购票!";

//弹出提示框

QMessageBox::information(this, QString::fromLocal8Bit("Refund failed"),QString::fromLocal8Bit("No tickets can be purchased beyond the tour time."));

}

this->updateList();

}

//槽:userdlg窗口,点击退票,实现退票逻辑

void UserDlg::**onRefund**()

{

//这两行只能获取选中单个的单元格数据

// QString id=ui->tableView->currentIndex().data().toString();

// qDebug()<<id;

int curRow= ui->tableView->currentIndex().row();//获取选中行号

QAbstractItemModel \*modessl = ui->tableView->model();

QModelIndex tmpIndex0 = modessl->*index*(curRow,0);//第0列(票号列)

QModelIndex tmpIndex1 = modessl->*index*(curRow,1);//第1列(游览时间)

QString ticketId=modessl->*data*(tmpIndex0).toString();//保存票号为string类型将加入sql查询

QDate useDate=modessl->*data*(tmpIndex1).toDate();//保存使用日期为date类型将于当前日期比较

//如果当前日期在使用日期之前,则退票

if(QDate::currentDate() <= useDate)

{

QString sql="call procRefund("+ticketId+");";

QSqlQuery query;

qDebug()<<sql;

sqlExec(query,sql);//执行查询

this->updateList();

}

else

{

qDebug()<<"超过游览时间, 不能退票!";

//弹出提示框不在允许的退票时间内

QMessageBox::information(this, QString::fromLocal8Bit("Refund failed"),QString::fromLocal8Bit("No refund is allowed after the tour time."));

}

}

//槽:点击退出登录,关闭userWnd,启动mainWnd

void UserDlg::**onLogout**()

{

this->close();

emit startMainWnd();//发送启动MainWnd的信号

}

//槽:当优惠选项改变时候, 查库改变相应的金额

void UserDlg::**onChooseChange**()

{

QSqlQuery query;

QString choose=ui->cobox\_Discount->currentText();

QString sql="select dMoney from Discount where dChoose='"+choose+"';";

sqlExec(query,sql,1);

query.first();

ui->lab\_money->setText("$ "+query.value(0).toString());

}

//从数据库更新列表控件

void UserDlg::**updateList**()

{

//添加表项到listView

QString sql="select \* from UserTicket;";

QSqlQueryModel \*model = new QSqlQueryModel;

model->setQuery(sql);

qDebug()<<"更新list查询:"<<sql;

model->*setHeaderData*(0, Qt::Horizontal, tr("票号"));

model->*setHeaderData*(1, Qt::Horizontal, tr("游览时间"));

model->*setHeaderData*(2, Qt::Horizontal, tr("优惠选项"));

model->*setHeaderData*(2, Qt::Horizontal, tr("票xx"));

ui->tableView->*setModel*(model);

}