



**数据库基础课程设计**

长大电影售票系统

**学院：理学院**

**专业：信息与计算科学**

**姓名： 刘曼琪 陈赞华**

**学号： 2019901318 2019904367**

**指导教师：邱瑞**

**完成时间：二〇二二年六月**

目录

[第一章 概述 3](#_Toc105447911)

[1.1选题目的和项目背景 3](#_Toc105447912)

[1.2课程设计主要内容 3](#_Toc105447913)

[1.3应用程序预期达到的效果 3](#_Toc105447914)

[1.4开发环境 3](#_Toc105447915)

[第二章 需求分析 3](#_Toc105447916)

[2.1用户的功能 3](#_Toc105447917)

[2.2商家的功能 4](#_Toc105447918)

[2.3售票系统的要求 4](#_Toc105447919)

[第三章 数据库概念结构设计 4](#_Toc105447920)

[3.1实体设计（黑体表示为主键，波浪线表示为外键） 4](#_Toc105447921)

[3.2E-R图 4](#_Toc105447922)

[3.2.1实体属性图 5](#_Toc105447923)

[3.2.2实体联系图 6](#_Toc105447924)

[第四章 数据库逻辑结构设计 6](#_Toc105447925)

[4.1表的设计 6](#_Toc105447926)

[4.1.1用户表（User） 6](#_Toc105447927)

[4.1.2放映厅表（ScreenHall） 7](#_Toc105447928)

[4.1.3电影表（Movie） 7](#_Toc105447929)

[4.1.4评论表（Comments） 7](#_Toc105447930)

[4.1.5电影安排表（MovieArrangement） 7](#_Toc105447931)

[4.1.6电影票表（Ticket） 8](#_Toc105447932)

[第五章 物理结构设计 8](#_Toc105447933)

[5.1创建各对象表 8](#_Toc105447934)

[5.1.1用户表 8](#_Toc105447935)

[5.1.2放映厅表 9](#_Toc105447936)

[5.1.3电影表 9](#_Toc105447937)

[5.1.4评论表 9](#_Toc105447938)

[5.1.5电影安排表 10](#_Toc105447939)

[5.1.6电影票 10](#_Toc105447940)

[第六章 数据库运行和维护 10](#_Toc105447941)

[6.1 查询、删除、修改 10](#_Toc105447942)

[6.1.1用户登录时，查询输入的账号密码是否和数据库用户表中的一致 10](#_Toc105447943)

[6.1.2用户登录后，可以查询电影菜单，使用隐式内连接查询结果 10](#_Toc105447944)

[6.1.3用户购买电影票之后，应该更新某放映厅的剩余票数 11](#_Toc105447945)

[6.1.4用户购票后，如果需要退票，删除该订单 11](#_Toc105447946)

[6.1.5用户查看某一部电影的评论 11](#_Toc105447947)

[6.1.6查看自己的购票记录（未退票的）,使用隐式内连接和子查询 11](#_Toc105447948)

[第七章 应用程序设计 11](#_Toc105447949)

[7.1用户登录注册入口 12](#_Toc105447950)

[7.2主窗口 12](#_Toc105447951)

[7.3买票窗口 12](#_Toc105447952)

[7.4退票窗口 13](#_Toc105447953)

[7.5查看电影票信息窗口 13](#_Toc105447954)

[7.6评论窗口 13](#_Toc105447955)

[总结 14](#_Toc105447956)

[参考文献 14](#_Toc105447957)

[附录 14](#_Toc105447958)

# 概述

## 1.1选题目的和项目背景

影院作为线下重要内容的传播媒介，其高光时刻是全国分账票房从2010年的101.7亿元增加至2015年的441亿元（全国分账票房复合增率达34.1%）；观影人次从2012年的4.57亿人增加至2015年12.23亿人（全国观影人次高复合增速达38.8%），在票房与观影人次的高复合增速下，我们又看到年度人均票价在2012年及2015年分别为37.4元、36.1元（2012年-2015年的人均票价复合增率为-1.2%）。

虽然近三年收到疫情的影响，电影院的发展举步维艰，有部分中小影院面临倒闭的现状，但是在影院看电影仍是人们会选择的娱乐方式。根据初步的调研，大学生平常的出游计划中通常会有看电影这一项目，并且在大学中有足够的场地可以播放电影，但却没有相对于的电影售票机制。这一选题正好弥补了市场的空缺，具有极大的可行性。

## 1.2课程设计主要内容

课程设计主要围绕着长安大学电影售票系统展开，具体从需求分析、数据库的概念结构设、数据库的逻辑结构设计、数据库的物理结构设计、以及数据库后续的运行与维护，再通过高级程序设计语言进行实现，让用户们可以通过使用相应的软件，进行电影的查看、购买、评论等操作。

## 1.3应用程序预期达到的效果

用户可以登录和注册账号，查看最近上映的电影以及其他用户对电影的评论，可以购票退票，看完电影后可以评论该电影。

电影院管理员可以上架最近上映的电影，并对电影的放映做出安排,可以给用户设定会员等级，可以修改任何信息。

## 1.4开发环境

数据库管理系统为sqlite3，高级程序设计语言为python。

前端设计工具Qt designer

# 需求分析

电影售票系统面对观影人群，虽然该系统用户仅为观影人群，但系统设计是采用模块，面向对象的思想，在时间充分的情况下可以添加电影院管理员

## 2.1用户的功能

（1）查询个人订单和电影片具体信息.

（2）查看最近上映的电影及评论。

（3）自主订票和退票。

## 2.2商家的功能

（1）上架电影。

（2）安排电影放映场次和时间。

（3）可以查看和修改数据库所有信息。

## 2.3售票系统的要求

（1）一个电影可以在多个放映厅放映，一个放映厅可以放映多个电影。

（2）一个用户可以观看多个电影，一个电影可以被多个用户观看。

（3）一个用户可以对不同电影进行评价，一个电影拥有多条评论。

（4）一个电影可以对应多张电影票，一张电影票只能对应一部电影。

（5）一个用户只有一个会员等级。

# 数据库概念结构设计

3.1实体设计（黑体表示为主键，波浪线表示为外键）

用户（**用户名**，密码，会员等级）

放映厅（**放映厅编号**，放映厅名字，座位数）

电影（**电影号**，电影名，类型，导演，演员，时长，价格）

评论（**评论序号**，电影号，用户号，评论）

电影安排（**放映序号**，电影号，放映厅编号，开始放映时间，结束放映时间）

电影票（**电影票号**，用户号，放映序号）

3.2E-R图

### 3.2.1实体属性图

用户

密码

会员等级

用户名

放映厅

电影

评论

电影安排

电影票

放映厅名字

座位数

放映厅编号

电影号

电影名

类型

导演

时长

演员

价格

电影号

用户号

评论

电影号

放映厅编号

开始放映时间

放映序号

结束放映时间

放映序号

用户号

电影票号

评论序号

用户号

### 3.2.2实体联系图

用户

 放映厅

电影

 评论

 电影安排

电影票

观看

放映

安排

评论

显示

n

m

n

m

n

m

n

m

p

n

m

p

1

1

# 数据库逻辑结构设计

4.1表的设计

### 4.1.1用户表（User）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项说明 | 字段名 | 数据类型 | 宽度 | 约束 |
| 用户名 | UserName | VARCHAR | 100 | PRIMARY KEY,  NOT NULL,  UNIQUE |
| 密码 | PassWord | VARCHAR | 100 | NOT NULL |
| 会员等级 | VipLevel | INTEGER |  | NOT NULL,  DEFAULT (1) |

### 4.1.2放映厅表（ScreenHall）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项说明 | 字段名 | 数据类型 | 宽度 | 约束 |
| 放映厅编号 | ID | VARCHAR | 100 | PRIMARY KEY,  NOT NULL |
| 放映厅名字 | Name | VARCHAR | 100 | NOT NULL |
| 座位数 | Seats | INT | 10 | DEFAULT (50) |

### 4.1.3电影表（Movie）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项说明 | 字段名 | 数据类型 | 宽度 | 约束 |
| 电影号 | MovieID | VARCHAR | 100 | PRIMARY KEY, |
| 电影名 | MovieName | VARCHAR | 100 | NOT NULL |
| 类型 | Category | VARCHAR | 100 | NOT NULL |
| 导演 | Director | VARCHAR | 100 | NOT NULL |
| 演员 | Actors | VARCHAR | 100 | NOT NULL |
| 时长 | Time | VARCHAR |  | DEFAULT (120),  NOT NULL |
| 价格 | Price | DOUBLE |  | NOT NULL |

### 4.1.4评论表（Comments）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项说明 | 字段名 | 数据类型 | 宽度 | 约束 |
| 评论序号 | ID | INTEGER |  | PRIMARY KEY AUTOINCREMENT |
| 电影号 | MovieID | VARCHAR | 100 | REFERENCES Movie (MovieID),  NOT NULL |
| 用户号 | UserID | VARCHAR | 100 | PRIMARY KEY,  REFERENCES User (UserName),  NOT NULL |
| 评论 | Content | VARCHAR | 200 | NOT NULL |

### 4.1.5电影安排表（MovieArrangement）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项说明 | 字段名 | 数据类型 | 宽度 | 约束 |
| 放映序号 | ID |  | 100 | PRIMARY KEY,  NOT NULL |
| 电影号 | MovieID | VARCHAR | 100 | REFERENCES Movie (MovieID) |
| 放映厅号 | HallID | INT |  | REFERENCES ScreenHall (ID),  NOT NULL |
| 开始放映时间 | StartTime | DATETIME |  | NOT NULL |
| 结束放映时间 | EndTime | DATETIME |  | NOT NULL |
| 用户号 | UserID |  |  | DEFAULT NULL |

### 4.1.6电影票表（Ticket）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项说明 | 字段名 | 数据类型 | 宽度 | 约束 |
| 电影票号 | TicketID |  |  | PRIMARY KEY AUTOINCREMENT，  NOT NULL |
| 用户号 | UserID | VARCHAR | 100 | REFERENCES User (UserName) |
| 放映序号 | ID | VARCHAR | 100 | FERENCES MovieArrangement (ID) |

# 物理结构设计

## 5.1创建各对象表

### 5.1.1用户表

CREATE TABLE User (

UserName VARCHAR (100) PRIMARY KEY

NOT NULL

UNIQUE,

PassWord VARCHAR (100) NOT NULL,

VipLevel INTEGER NOT NULL

DEFAULT (1)

);

### 5.1.2放映厅表

CREATE TABLE ScreenHall (

ID VARCHAR (100) PRIMARY KEY

NOT NULL,

Name VARCHAR (100) NOT NULL,

Seats INT (10) DEFAULT (50)

);

### 5.1.3电影表

CREATE TABLE Movie (

MovieID VARCHAR (100) PRIMARY KEY,

MovieName VARCHAR (100) NOT NULL,

Category VARCHAR (100) NOT NULL,

Director VARCHAR (100) NOT NULL,

Actors VARCHAR (100) NOT NULL,

Time VARCHAR NOT NULL

DEFAULT (120),

Price DOUBLE NOT NULL

);

### 5.1.4评论表

CREATE TABLE Comments (  
    ID      INTEGER       PRIMARY KEY AUTOINCREMENT,  
    MovieID VARCHAR (100) REFERENCES Movie (MovieID)   
                          NOT NULL,  
    UserID  VARCHAR (100) NOT NULL  
                          REFERENCES User (UserName),  
    Content VARCHAR (200) NOT NULL  
);

### 5.1.5电影安排表

CREATE TABLE MovieArrangement (  
    ID        VARCHAR (100) PRIMARY KEY  
                            NOT NULL,  
    MovieID   VARCHAR (100) REFERENCES Movie (MovieID),  
    HallID    INT           REFERENCES ScreenHall (ID)   
                            NOT NULL,  
    StartTime DATETIME      NOT NULL,  
    EndTime   DATETIME      NOT NULL,  
    UserID                  DEFAULT NULL  
);

### 5.1.6电影票

CREATE TABLE Ticket (  
    TicketID INTEGER       PRIMARY KEY AUTOINCREMENT  
                           NOT NULL,  
    ID       VARCHAR (100) REFERENCES MovieArrangement (ID),  
    UserID   VARCHAR (100) REFERENCES User (UserName)   
);

# 数据库运行和维护

6.1 查询、删除、修改

### 6.1.1用户登录时，查询输入的账号密码是否和数据库用户表中的一致

sql = '''SELECT PassWord FROM User WHERE UserName= {}'''.format(UserName)

### 6.1.2用户登录后，可以查询电影菜单，使用隐式内连接查询结果

sql =

"""SELECT MA.ID,M.MovieName,M.Director,M.Actors,MA.StartTime,MA.EndTime,M.Price

FROM

Movie M ,MovieArrangement MA

WHERE

1. MovieID = MA.MovieID"""

### 6.1.3用户购买电影票之后，应该更新某放映厅的剩余票数

sql4 = "UPDATE ScreenHall SET Seats = {} WHERE ID = {} ".format(seatnumber,HallID)

### 6.1.4用户购票后，如果需要退票，删除该订单

sql3 = ''' DELETE FROM Ticket WHERE TicketID = {}'''.format(result)

### 6.1.5用户查看某一部电影的评论

sql2 =

"SELECT

c.UserID,m.MovieName,c.Content

FROM

(SELECT \* FROM Comments WHERE MovieID ={}) c ,Movie m

WHERE

m.MovieID = c.MovieID".format(MovieID)

### 6.1.6查看自己的购票记录（未退票的）,使用隐式内连接和子查询

sql = '''

SELECT

TB.ID,m.MovieName,TB.StartTime,TB.EndTime,m.Price

FROM

Movie m,

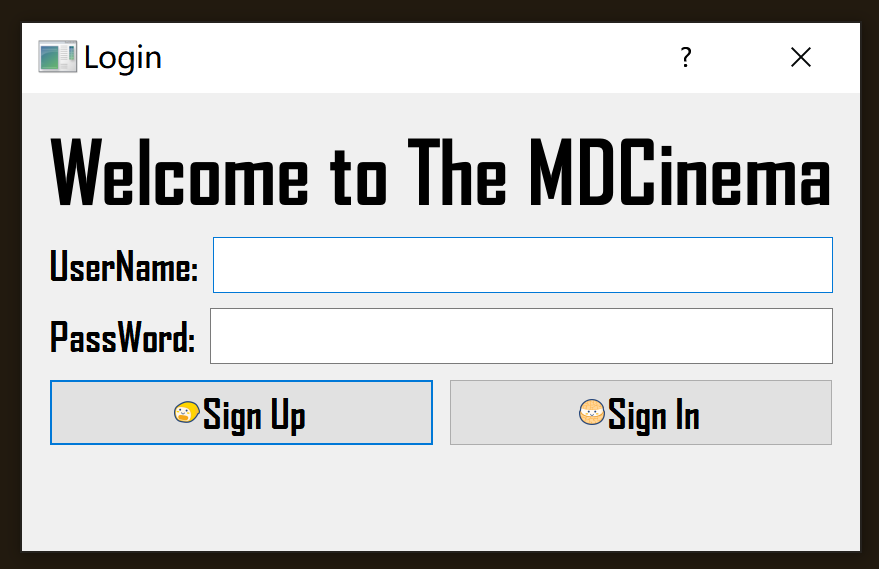
(SELECT MA.ID,MA.MovieID,MA.StartTime,MA.EndTime FROM Ticket T,MovieArrangement MA WHERE T.ID = MA.ID and T.UserID = {} ) TB

WHERE

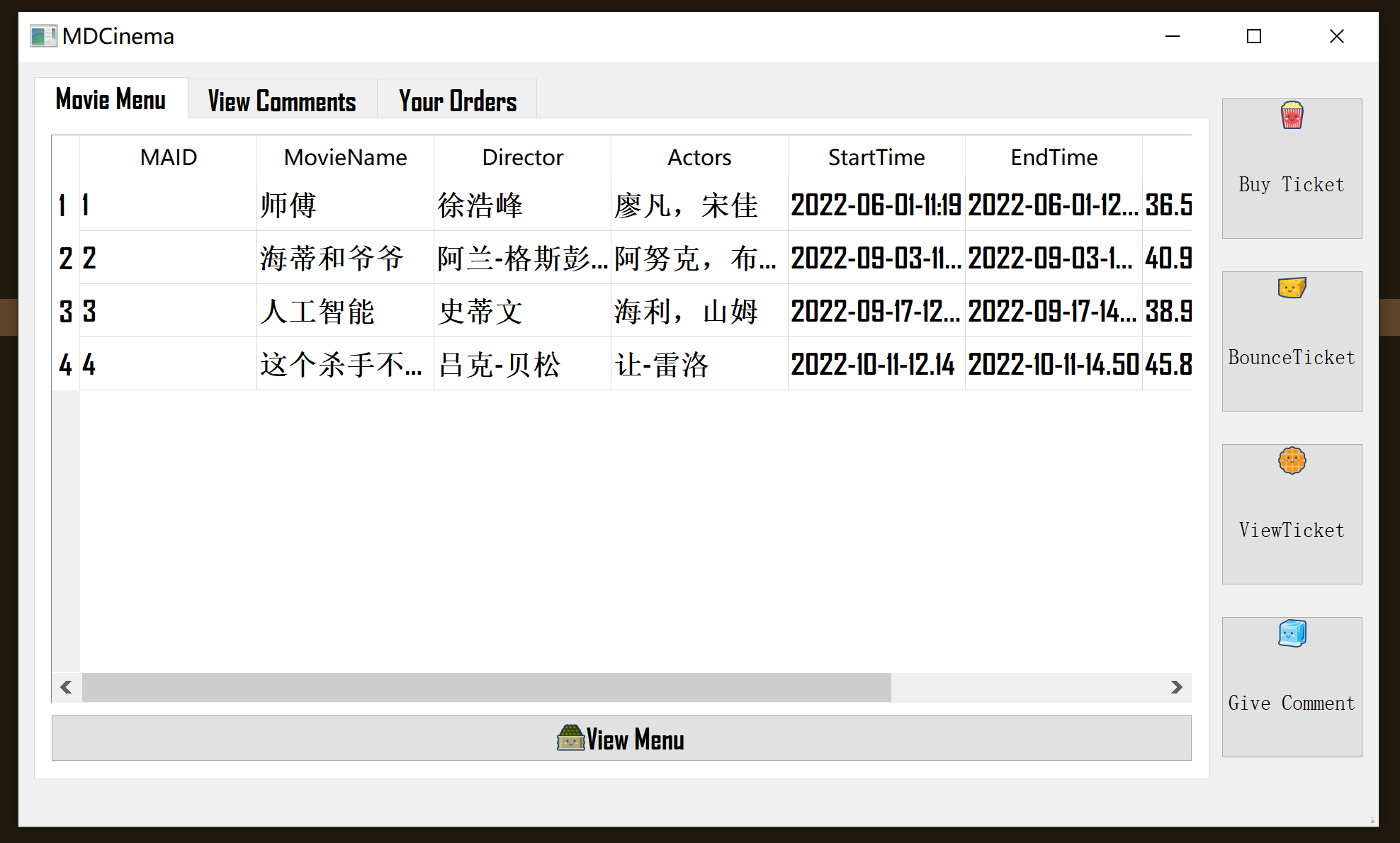
TB.MovieID = m.MovieID'''.format(UserID)

# 应用程序设计

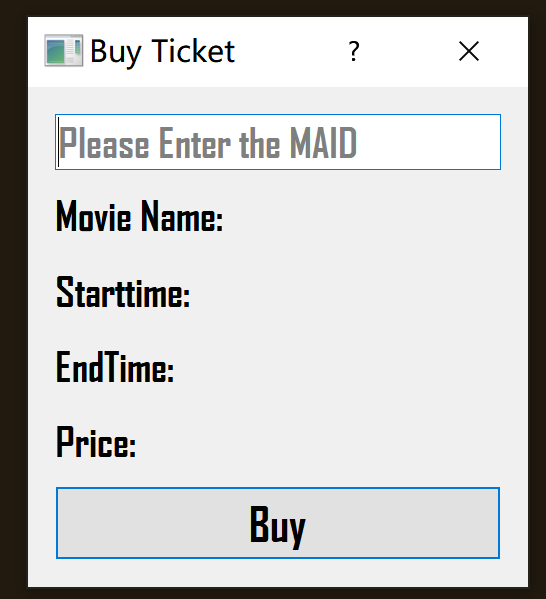
## 7.1用户登录注册入口



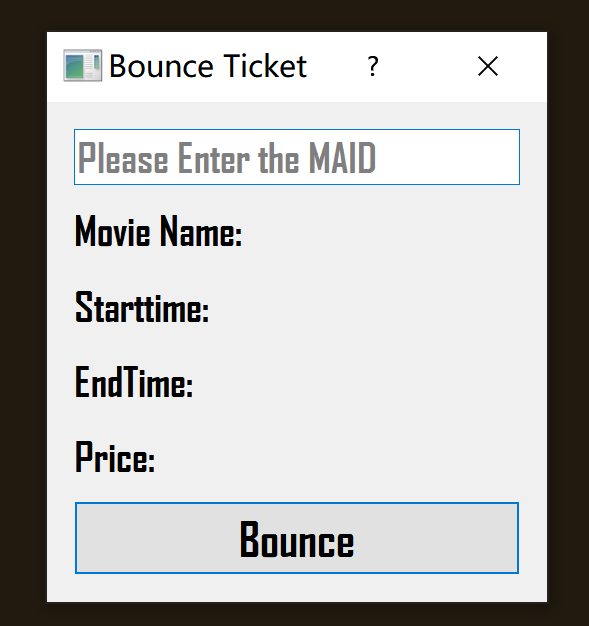
## 7.2主窗口



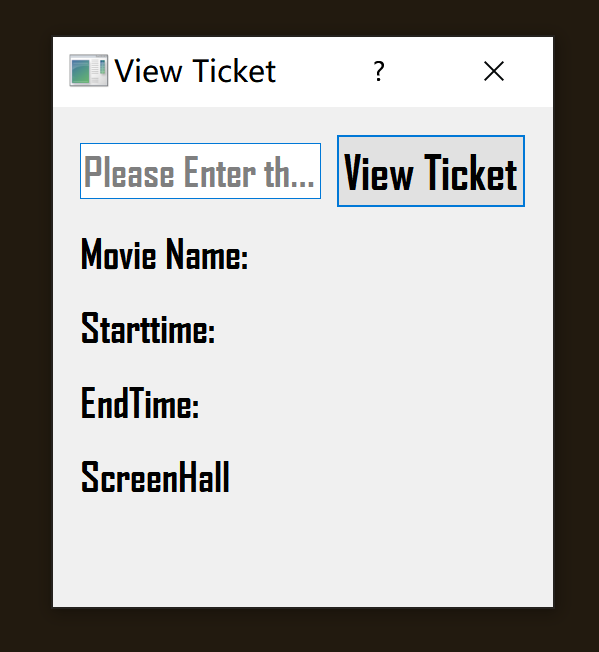
## 7.3买票窗口



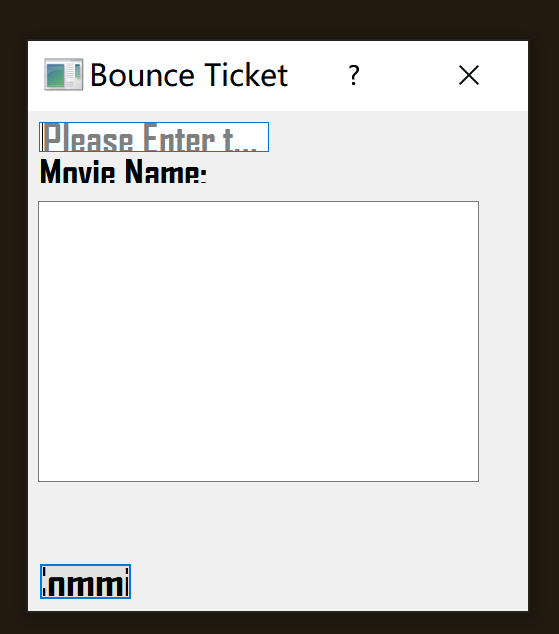
## 7.4退票窗口



## 7.5查看电影票信息窗口



## 7.6评论窗口



# 总结

1、做课设的目的：

做该课设的目的是想挑战自己，然后发挥团队合作的作用。

2、完成课设过程中遇到的挑战

(1)在选择项目之后，如何指定明确项目所需要储备的知识，以及制定项目的规划。

(2)探索在知识储备不够完成一个整个项目的前提下，明确自己需要学习哪些知识（这个是我在 这个项目中学到的最重要的知识）。（我的解决该问题的灵感主要是来自编程中的模块化思想， 以及面向对象的思想）

(3)在团队合作中，如何发挥团队中个人的长处，以及根据团队中个人的长处分配不同的任务， 团队成员如何合作完成重叠的任务，并完成整个项目。

3、 项目的优点和缺陷

优点： 该项目采用模块化以及面向对象的思想，便于后期的修改和更新。该程序有一定的健壮性。

缺点： 数据库设计部分为很好的体现面向对象的思想，数据库的设计还不够好。希望我们在时间充裕的情况在重新设计数据库部分。增强数据的逻辑独立性。是得以后的开发和维护更方便。

1. 回顾与建议

在本次课设中，我花费了90% 的时间和精力来完成其它的知识储备（除数据库外的）、程序的细节处理，仅仅只能花费了10% 的时间在数据库的设计，以及对数据库中数据的操作，因为完成的课设的时间过短，以及本学期其他课程任务中的情况下，很难在花时间数据库的设计和打磨上，所以完成该项目之后，让我产生该课设的中心没有落在设计更好的数据库、以及深入理解数据独立性，范式，模式等核心知识上。如果老师在我们设计该项目前给我们提供该项目的一些框架，或者给我们讲解一些如何设计复杂项目的知识，我们可以更专注该项目中对数据库设计、使用，甚至加深我们对数据库这门课程的理解。

# 参考文献

# 附录

[[1]](#endnote-1)

1. **启动程序**

   import sys

   from PyQt5.QtWidgets import QApplication

   from MDCinemaSystem import MDCSystem

   app = QApplication(sys.argv)

   cinema  = MDCSystem()

   sys.exit(app.exec())

   **电影院系统管理中心**

   from PyQt5.QtWidgets import QMainWindow,QDialog,QMessageBox

   from MDCinema import Ui\_MainWindow

   from buyTicket import BUY\_Dialog

   from bounceTicket import Bounce\_Dialog

   from viewTicket import view\_Dialog

   from comment import comment\_Dialog

   from loginUi import Ui\_Login\_Dialog

   class MDCSystem(QMainWindow,Ui\_MainWindow):

       def \_\_init\_\_(self):

           super().\_\_init\_\_()

           self.loginState = False

           self.login()

           self.setupUi(self)

           # buy tickets

           self.toolButton\_BuyTicket.clicked.connect(self.buyTicket)

           # bounce tickets

           self.toolButton\_bounceTickets.clicked.connect(self.bounceTicket)

           # view tickets

           self.toolButton\_viewTicket.clicked.connect(self.viewTicket)

           # give comments

           self.toolButton\_giveComments.clicked.connect(self.comments)

           self.show()

       def login(self):

           dialog = QDialog()

           ui = Ui\_Login\_Dialog()

           ui.setupUi(dialog)

           dialog.exec()

           self.loginState = ui.state

           dialog.close()

       def buyTicket(self):

           if self.loginState:

               dialog = QDialog()

               ui = BUY\_Dialog()

               ui.setupUi(dialog)

               dialog.exec()

           else:

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

       def bounceTicket(self):

           if self.loginState:

               dialog = QDialog()

               ui = Bounce\_Dialog()

               ui.setupUi(dialog)

               dialog.exec()

           else:

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

       def viewTicket(self):

           if self.loginState:

               dialog = QDialog()

               ui = view\_Dialog()

               ui.setupUi(dialog)

               dialog.exec()

           else:

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

       def comments(self):

           if self.loginState:

               dialog = QDialog()

               ui = comment\_Dialog()

               ui.setupUi(dialog)

               dialog.exec()

           else:

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

   **主窗口界面：**

   # -\*- coding: utf-8 -\*-

   # Form implementation generated from reading ui file 'MDCinema.ui'

   #

   # Created by: PyQt5 UI code generator 5.15.4

   #

   # WARNING: Any manual changes made to this file will be lost when pyuic5 is

   # run again.  Do not edit this file unless you know what you are doing.

   from PyQt5 import QtCore, QtGui, QtWidgets

   from PyQt5.QtWidgets  import QMessageBox,QTableWidgetItem

   import sqlite3

   class Ui\_MainWindow(object):

       def setupUi(self, MainWindow):

           MainWindow.setObjectName("MainWindow")

           MainWindow.resize(1920, 1080)

           self.centralwidget = QtWidgets.QWidget(MainWindow)

           self.centralwidget.setObjectName("centralwidget")

           self.horizontalLayout\_5 = QtWidgets.QHBoxLayout(self.centralwidget)

           self.horizontalLayout\_5.setObjectName("horizontalLayout\_5")

           self.tabWidget = QtWidgets.QTabWidget(self.centralwidget)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.tabWidget.setFont(font)

           self.tabWidget.setObjectName("tabWidget")

           self.tab\_2 = QtWidgets.QWidget()

           self.tab\_2.setObjectName("tab\_2")

           self.verticalLayout\_2 = QtWidgets.QVBoxLayout(self.tab\_2)

           self.verticalLayout\_2.setObjectName("verticalLayout\_2")

           self.tableWidgetMovieMenu = QtWidgets.QTableWidget(self.tab\_2)

           self.tableWidgetMovieMenu.setObjectName("tableWidgetMovieMenu")

           self.tableWidgetMovieMenu.setColumnCount(8)

           self.tableWidgetMovieMenu.setRowCount(0)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(0, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(1, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(2, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(3, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(4, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(5, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(6, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidgetMovieMenu.setHorizontalHeaderItem(7, item)

           self.verticalLayout\_2.addWidget(self.tableWidgetMovieMenu)

           self.pushButton\_ViewMenu = QtWidgets.QPushButton(self.tab\_2)

           icon = QtGui.QIcon()

           icon.addPixmap(QtGui.QPixmap("Image/鱼子酱.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.pushButton\_ViewMenu.setIcon(icon)

           self.pushButton\_ViewMenu.setIconSize(QtCore.QSize(40, 40))

           self.pushButton\_ViewMenu.setObjectName("pushButton\_ViewMenu")

           # view movie Menu

           self.pushButton\_ViewMenu.clicked.connect(self.movieMenu)

           self.verticalLayout\_2.addWidget(self.pushButton\_ViewMenu)

           self.tabWidget.addTab(self.tab\_2, "")

           self.tab\_4 = QtWidgets.QWidget()

           self.tab\_4.setObjectName("tab\_4")

           self.verticalLayout\_5 = QtWidgets.QVBoxLayout(self.tab\_4)

           self.verticalLayout\_5.setObjectName("verticalLayout\_5")

           self.lineEdit\_viewcommentMAID = QtWidgets.QLineEdit(self.tab\_4)

           self.lineEdit\_viewcommentMAID.setObjectName("lineEdit\_viewcommentMAID")

           self.verticalLayout\_5.addWidget(self.lineEdit\_viewcommentMAID)

           self.tableWidget\_viewComments = QtWidgets.QTableWidget(self.tab\_4)

           self.tableWidget\_viewComments.setObjectName("tableWidget\_viewComments")

           self.tableWidget\_viewComments.setColumnCount(3)

           self.tableWidget\_viewComments.setRowCount(0)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_viewComments.setHorizontalHeaderItem(0, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_viewComments.setHorizontalHeaderItem(1, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_viewComments.setHorizontalHeaderItem(2, item)

           self.tableWidget\_viewComments.horizontalHeader().setDefaultSectionSize(230)

           self.verticalLayout\_5.addWidget(self.tableWidget\_viewComments)

           self.pushButton\_viewComments = QtWidgets.QPushButton(self.tab\_4)

           icon1 = QtGui.QIcon()

           icon1.addPixmap(QtGui.QPixmap("Image/口香糖.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.pushButton\_viewComments.setIcon(icon1)

           self.pushButton\_viewComments.setIconSize(QtCore.QSize(40, 40))

           self.pushButton\_viewComments.setObjectName("pushButton\_viewComments")

           # view comments

           self.pushButton\_viewComments.clicked.connect(self.viewComments)

           self.verticalLayout\_5.addWidget(self.pushButton\_viewComments)

           self.tabWidget.addTab(self.tab\_4, "")

           self.tab\_3 = QtWidgets.QWidget()

           self.tab\_3.setObjectName("tab\_3")

           self.verticalLayout\_3 = QtWidgets.QVBoxLayout(self.tab\_3)

           self.verticalLayout\_3.setObjectName("verticalLayout\_3")

           self.tableWidget\_Tickets = QtWidgets.QTableWidget(self.tab\_3)

           self.tableWidget\_Tickets.setObjectName("tableWidget\_Tickets")

           self.tableWidget\_Tickets.setColumnCount(5)

           self.tableWidget\_Tickets.setRowCount(0)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_Tickets.setHorizontalHeaderItem(0, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_Tickets.setHorizontalHeaderItem(1, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_Tickets.setHorizontalHeaderItem(2, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_Tickets.setHorizontalHeaderItem(3, item)

           item = QtWidgets.QTableWidgetItem()

           self.tableWidget\_Tickets.setHorizontalHeaderItem(4, item)

           self.tableWidget\_Tickets.horizontalHeader().setDefaultSectionSize(142)

           self.verticalLayout\_3.addWidget(self.tableWidget\_Tickets)

           self.pushButton\_viewTickets = QtWidgets.QPushButton(self.tab\_3)

           icon2 = QtGui.QIcon()

           icon2.addPixmap(QtGui.QPixmap("Image/果冻.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.pushButton\_viewTickets.setIcon(icon2)

           self.pushButton\_viewTickets.setIconSize(QtCore.QSize(40, 40))

           self.pushButton\_viewTickets.setObjectName("pushButton\_viewTickets")

           # view order

           self.pushButton\_viewTickets.clicked.connect(self.viewOrder)

           self.verticalLayout\_3.addWidget(self.pushButton\_viewTickets)

           self.tabWidget.addTab(self.tab\_3, "")

           self.horizontalLayout\_5.addWidget(self.tabWidget)

           self.verticalLayout\_4 = QtWidgets.QVBoxLayout()

           self.verticalLayout\_4.setObjectName("verticalLayout\_4")

           self.toolButton\_BuyTicket = QtWidgets.QToolButton(self.centralwidget)

           self.toolButton\_BuyTicket.setMaximumSize(QtCore.QSize(200, 200))

           icon3 = QtGui.QIcon()

           icon3.addPixmap(QtGui.QPixmap("Image/爆米花.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.toolButton\_BuyTicket.setIcon(icon3)

           self.toolButton\_BuyTicket.setIconSize(QtCore.QSize(40, 40))

           self.toolButton\_BuyTicket.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)

           self.toolButton\_BuyTicket.setObjectName("toolButton\_BuyTicket")

           self.verticalLayout\_4.addWidget(self.toolButton\_BuyTicket)

           self.toolButton\_bounceTickets = QtWidgets.QToolButton(self.centralwidget)

           self.toolButton\_bounceTickets.setMaximumSize(QtCore.QSize(200, 200))

           icon4 = QtGui.QIcon()

           icon4.addPixmap(QtGui.QPixmap("Image/奶酪.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.toolButton\_bounceTickets.setIcon(icon4)

           self.toolButton\_bounceTickets.setIconSize(QtCore.QSize(40, 40))

           self.toolButton\_bounceTickets.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)

           self.toolButton\_bounceTickets.setObjectName("toolButton\_bounceTickets")

           self.verticalLayout\_4.addWidget(self.toolButton\_bounceTickets)

           self.toolButton\_viewTicket = QtWidgets.QToolButton(self.centralwidget)

           self.toolButton\_viewTicket.setMaximumSize(QtCore.QSize(200, 200))

           icon5 = QtGui.QIcon()

           icon5.addPixmap(QtGui.QPixmap("Image/华夫饼.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.toolButton\_viewTicket.setIcon(icon5)

           self.toolButton\_viewTicket.setIconSize(QtCore.QSize(40, 40))

           self.toolButton\_viewTicket.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)

           self.toolButton\_viewTicket.setObjectName("toolButton\_viewTicket")

           self.verticalLayout\_4.addWidget(self.toolButton\_viewTicket)

           self.toolButton\_giveComments = QtWidgets.QToolButton(self.centralwidget)

           self.toolButton\_giveComments.setMaximumSize(QtCore.QSize(200, 200))

           icon6 = QtGui.QIcon()

           icon6.addPixmap(QtGui.QPixmap("Image/冰块.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.toolButton\_giveComments.setIcon(icon6)

           self.toolButton\_giveComments.setIconSize(QtCore.QSize(40, 40))

           self.toolButton\_giveComments.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)

           self.toolButton\_giveComments.setObjectName("toolButton\_giveComments")

           #MainWindow.setGeometry(QtCore.QRect(800,500,1000,800))

           self.verticalLayout\_4.addWidget(self.toolButton\_giveComments)

           self.horizontalLayout\_5.addLayout(self.verticalLayout\_4)

           MainWindow.setCentralWidget(self.centralwidget)

           self.menubar = QtWidgets.QMenuBar(MainWindow)

           self.menubar.setGeometry(QtCore.QRect(0, 0, 900, 22))

           self.menubar.setObjectName("menubar")

           self.menuwenjian = QtWidgets.QMenu(self.menubar)

           self.menuwenjian.setTitle("")

           self.menuwenjian.setObjectName("menuwenjian")

           MainWindow.setMenuBar(self.menubar)

           self.statusbar = QtWidgets.QStatusBar(MainWindow)

           self.statusbar.setObjectName("statusbar")

           MainWindow.setStatusBar(self.statusbar)

           self.menubar.addAction(self.menuwenjian.menuAction())

           self.retranslateUi(MainWindow)

           self.tabWidget.setCurrentIndex(0)

           QtCore.QMetaObject.connectSlotsByName(MainWindow)

       def movieMenu(self):

           if self.loginState:

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   sql = """SELECT

                   MA.ID,M.MovieName,M.Director,M.Actors,MA.StartTime,MA.EndTime,M.Price

                   FROM

                   Movie M ,MovieArrangement MA

                   WHERE

                   M.MovieID = MA.MovieID"""

                   cur.execute(sql)

                   result = cur.fetchall()

                   self.tableWidgetMovieMenu.setRowCount(0)

                   for row\_number,row\_data in enumerate(result):

                       self.tableWidgetMovieMenu.insertRow(row\_number)

                       for column\_nember,data in enumerate(row\_data):

                           self.tableWidgetMovieMenu.setItem(row\_number,column\_nember,QTableWidgetItem(str(data)))

               except sqlite3.Error as e:

                   print(e)

           else:

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

       def viewComments(self):

           if  self.loginState:

               MAID = self.lineEdit\_viewcommentMAID.text()

               if MAID!= "":

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   sql ="SELECT MovieID,UserID FROM MovieArrangement WHERE ID = {}".format(MAID)

                   cur.execute(sql)

                   result = cur.fetchone()

                   MovieID = result[0]

                   sql2 = "SELECT c.UserID,m.MovieName,c.Content FROM (SELECT \* FROM Comments WHERE MovieID ={}) c ,Movie m WHERE m.MovieID = c.MovieID".format(MovieID)

                   cur.execute(sql2)

                   result = cur.fetchall()

                   self.tableWidget\_viewComments.setRowCount(0)

                   for row\_number,row\_data in enumerate(result):

                       self.tableWidget\_viewComments.insertRow(row\_number)

                       for column\_nember,data in enumerate(row\_data):

                           self.tableWidget\_viewComments.setItem(row\_number,column\_nember,QTableWidgetItem(str(data)))

               else:

                   QMessageBox.about(self,"Warming ","Please Enter MAID")

           else :

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

       def viewOrder(self):

           if self.loginState:

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   sql0 = '''SELECT UserID from MovieArrangement'''

                   cur.execute(sql0)

                   result  = cur.fetchone()

                   UserID = 0

                   for User in result:

                       UserID = User

                   #print(UserID)

                   sql = '''

                   SELECT

                   TB.ID,m.MovieName,TB.StartTime,TB.EndTime,m.Price

                   FROM

   Movie m,(SELECT MA.ID,MA.MovieID,MA.StartTime,MA.EndTime FROM Ticket T,MovieArrangement MA WHERE T.ID = MA.ID and T.UserID = {} ) TB

                   WHERE

                   TB.MovieID = m.MovieID'''.format(UserID)

                   cur.execute(sql)

                   result = cur.fetchall()

                   self.tableWidget\_Tickets.setRowCount(0)

                   for row\_number,row\_data in enumerate(result):

                       self.tableWidget\_Tickets.insertRow(row\_number)

                       for column\_nember,data in enumerate(row\_data):

                           self.tableWidget\_Tickets.setItem(row\_number,column\_nember,QTableWidgetItem(str(data)))

               except sqlite3.Error as e:

                   print(e)

           else:

               QMessageBox.about(self,"Warming ","Please Login Firstly!")

       def retranslateUi(self, MainWindow):

           \_translate = QtCore.QCoreApplication.translate

           MainWindow.setWindowTitle(\_translate("MainWindow", "MDCinema"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(0)

           item.setText(\_translate("MainWindow", "MAID"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(1)

           item.setText(\_translate("MainWindow", "MovieName"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(2)

           item.setText(\_translate("MainWindow", "Director"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(3)

           item.setText(\_translate("MainWindow", "Actors"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(4)

           item.setText(\_translate("MainWindow", "StartTime"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(5)

           item.setText(\_translate("MainWindow", "EndTime"))

           item = self.tableWidgetMovieMenu.horizontalHeaderItem(6)

           item.setText(\_translate("MainWindow", "Price"))

           self.pushButton\_ViewMenu.setText(\_translate("MainWindow", "View Menu"))

           self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_2), \_translate("MainWindow", "Movie Menu"))

           self.lineEdit\_viewcommentMAID.setPlaceholderText(\_translate("MainWindow", "Please Enter the MAID "))

           item = self.tableWidget\_viewComments.horizontalHeaderItem(0)

           item.setText(\_translate("MainWindow", "Users"))

           item = self.tableWidget\_viewComments.horizontalHeaderItem(1)

           item.setText(\_translate("MainWindow", "MovieName"))

           item = self.tableWidget\_viewComments.horizontalHeaderItem(2)

           item.setText(\_translate("MainWindow", "Comments"))

           self.pushButton\_viewComments.setText(\_translate("MainWindow", "View Comments"))

           self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_4), \_translate("MainWindow", "View Comments"))

           item = self.tableWidget\_Tickets.horizontalHeaderItem(0)

           item.setText(\_translate("MainWindow", "MAID"))

           item = self.tableWidget\_Tickets.horizontalHeaderItem(1)

           item.setText(\_translate("MainWindow", "MovieName"))

           item = self.tableWidget\_Tickets.horizontalHeaderItem(2)

           item.setText(\_translate("MainWindow", "StartTime"))

           item = self.tableWidget\_Tickets.horizontalHeaderItem(3)

           item.setText(\_translate("MainWindow", "EndTime"))

           item = self.tableWidget\_Tickets.horizontalHeaderItem(4)

           item.setText(\_translate("MainWindow", "Price"))

           self.pushButton\_viewTickets.setText(\_translate("MainWindow", "View Tickets"))

           self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_3), \_translate("MainWindow", "Your Orders"))

           self.toolButton\_BuyTicket.setText(\_translate("MainWindow", "Buy Ticket"))

           self.toolButton\_bounceTickets.setText(\_translate("MainWindow", "BounceTicket"))

           self.toolButton\_viewTicket.setText(\_translate("MainWindow", "ViewTicket"))

           self.toolButton\_giveComments.setText(\_translate("MainWindow", "Give Comment"))

   if \_\_name\_\_ == "\_\_main\_\_":

       import sys

       app = QtWidgets.QApplication(sys.argv)

       MainWindow = QtWidgets.QMainWindow()

       ui = Ui\_MainWindow()

       ui.setupUi(MainWindow)

       MainWindow.show()

       sys.exit(app.exec\_())

   **登陆界面：**

   # -\*- coding: utf-8 -\*-

   # Form implementation generated from reading ui file 'Login.ui'

   #

   # Created by: PyQt5 UI code generator 5.15.4

   #

   # WARNING: Any manual changes made to this file will be lost when pyuic5 is

   # run again.  Do not edit this file unless you know what you are doing.

   from PyQt5 import QtCore, QtGui, QtWidgets

   import sqlite3

   class Ui\_Login\_Dialog(object):

       def setupUi(self, Login\_Dialog):

           Login\_Dialog.setObjectName("Login\_Dialog")

           Login\_Dialog.resize(400, 300)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(15)

           font.setBold(True)

           font.setWeight(75)

           Login\_Dialog.setFont(font)

           self.verticalLayout = QtWidgets.QVBoxLayout(Login\_Dialog)

           self.verticalLayout.setObjectName("verticalLayout")

           self.label\_welcome = QtWidgets.QLabel(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(26)

           font.setBold(True)

           font.setWeight(75)

           self.label\_welcome.setFont(font)

           self.label\_welcome.setObjectName("label\_welcome")

           self.verticalLayout.addWidget(self.label\_welcome)

           self.horizontalLayout = QtWidgets.QHBoxLayout()

           self.horizontalLayout.setObjectName("horizontalLayout")

           self.label\_UserName = QtWidgets.QLabel(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_UserName.setFont(font)

           self.label\_UserName.setObjectName("label\_UserName")

           self.horizontalLayout.addWidget(self.label\_UserName)

           self.lineEdit\_UserName = QtWidgets.QLineEdit(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.lineEdit\_UserName.setFont(font)

           self.lineEdit\_UserName.setObjectName("lineEdit\_UserName")

           self.horizontalLayout.addWidget(self.lineEdit\_UserName)

           self.verticalLayout.addLayout(self.horizontalLayout)

           self.horizontalLayout\_2 = QtWidgets.QHBoxLayout()

           self.horizontalLayout\_2.setObjectName("horizontalLayout\_2")

           self.label\_PassWord = QtWidgets.QLabel(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_PassWord.setFont(font)

           self.label\_PassWord.setObjectName("label\_PassWord")

           self.horizontalLayout\_2.addWidget(self.label\_PassWord)

           self.lineEdit\_PassWord = QtWidgets.QLineEdit(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.lineEdit\_PassWord.setFont(font)

           self.lineEdit\_PassWord.setObjectName("lineEdit\_PassWord")

           self.horizontalLayout\_2.addWidget(self.lineEdit\_PassWord)

           self.verticalLayout.addLayout(self.horizontalLayout\_2)

           self.horizontalLayout\_3 = QtWidgets.QHBoxLayout()

           self.horizontalLayout\_3.setObjectName("horizontalLayout\_3")

           self.pushButton\_signup = QtWidgets.QPushButton(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.pushButton\_signup.setFont(font)

           icon = QtGui.QIcon()

           icon.addPixmap(QtGui.QPixmap("Image/班戟.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.pushButton\_signup.setIcon(icon)

           self.pushButton\_signup.setIconSize(QtCore.QSize(26, 26))

           self.pushButton\_signup.setObjectName("pushButton\_signup")

           self.pushButton\_signup.clicked.connect(self.signup)

           self.horizontalLayout\_3.addWidget(self.pushButton\_signup)

           self.pushButton\_signIn = QtWidgets.QPushButton(Login\_Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.pushButton\_signIn.setFont(font)

           icon1 = QtGui.QIcon()

           icon1.addPixmap(QtGui.QPixmap("Image/泡芙.png"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

           self.pushButton\_signIn.setIcon(icon1)

           self.pushButton\_signIn.setIconSize(QtCore.QSize(26, 26))

           self.pushButton\_signIn.setObjectName("pushButton\_signIn")

           self.pushButton\_signIn.clicked.connect(self.signin)

           self.horizontalLayout\_3.addWidget(self.pushButton\_signIn)

           self.verticalLayout.addLayout(self.horizontalLayout\_3)

           self.label\_loginstate = QtWidgets.QLabel(Login\_Dialog)

           self.label\_loginstate.setText("")

           self.label\_loginstate.setObjectName("label\_loginstate")

           self.verticalLayout.addWidget(self.label\_loginstate)

           self.retranslateUi(Login\_Dialog)

           QtCore.QMetaObject.connectSlotsByName(Login\_Dialog)

           self.state =False

       def retranslateUi(self, Login\_Dialog):

           \_translate = QtCore.QCoreApplication.translate

           Login\_Dialog.setWindowTitle(\_translate("Login\_Dialog", "Login"))

           self.label\_welcome.setText(\_translate("Login\_Dialog", "Welcome to The MDCinema"))

           self.label\_UserName.setText(\_translate("Login\_Dialog", "UserName:"))

           self.label\_PassWord.setText(\_translate("Login\_Dialog", "PassWord:"))

           self.pushButton\_signup.setText(\_translate("Login\_Dialog", "Sign Up"))

           self.pushButton\_signIn.setText(\_translate("Login\_Dialog", "Sign In"))

       def signup(self):

           UserName = self.lineEdit\_UserName.text()

           PassWord = self.lineEdit\_PassWord.text()

           if UserName!="" and PassWord!="":

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   sql = "INSERT INTO User (UserName,PassWord) VALUES (?,?)"

                   value = (UserName,PassWord)

                   cur.execute(sql,value)

                   conn.commit()

                   self.label\_loginstate.setText("Sign Up Successfully! Your can Sign In now")

                   self.state =False

               except sqlite3.Error as e:

                   print(e)

                   self.label\_loginstate.setText("Sign Up Failed! Please change another UserName!")

                   self.state = False

           else:

               self.label\_loginstate.setText("Error !Please Enter UserName and PassWord")

               self.state = False

       def signin(self):

           UserName = self.lineEdit\_UserName.text()

           PassWord = self.lineEdit\_PassWord.text()

           if UserName!="" and PassWord!="":

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   sql = '''SELECT PassWord FROM User WHERE UserName= {}'''.format(UserName)

                   cur.execute(sql)

                   result = cur.fetchone()

                   #print(result)

                   if result!= None and result[0]==PassWord:

                       sql2 = "UPDATE MovieArrangement SET UserID ={}".format(UserName)

                       cur.execute(sql2)

                       conn.commit()

                       self.label\_loginstate.setText("Sign In Successfully")

                       self.state = True

                   else :

                       self.state = False

                       self.label\_loginstate.setText("Sign In Failed!")

               except sqlite3.Error as e:

                   self.state = False

                   print(e)

           else:

               self.label\_loginstate.setText("Error !Please Enter UserName and PassWord")

               self.state = False

   if \_\_name\_\_ == "\_\_main\_\_":

       import sys

       app = QtWidgets.QApplication(sys.argv)

       Login\_Dialog = QtWidgets.QDialog()

       ui = Ui\_Login\_Dialog()

       ui.setupUi(Login\_Dialog)

       Login\_Dialog.show()

       sys.exit(app.exec\_())

   **买票界面：**

   # -\*- coding: utf-8 -\*-

   # Form implementation generated from reading ui file 'buyTicket.ui'

   #

   # Created by: PyQt5 UI code generator 5.15.4

   #

   # WARNING: Any manual changes made to this file will be lost when pyuic5 is

   # run again.  Do not edit this file unless you know what you are doing.

   from multiprocessing import connection

   from PyQt5 import QtCore, QtGui, QtWidgets

   import sqlite3

   from sqlite3 import Error

   class BUY\_Dialog(object):

       def setupUi(self, Dialog):

           Dialog.setObjectName("Dialog")

           Dialog.resize(500, 500)

           self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)

           self.verticalLayout.setObjectName("verticalLayout")

           self.lineEdit\_MAID = QtWidgets.QLineEdit(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.lineEdit\_MAID.setFont(font)

           self.lineEdit\_MAID.setObjectName("lineEdit\_MAID")

           self.verticalLayout.addWidget(self.lineEdit\_MAID)

           self.label\_moviename = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_moviename.setFont(font)

           self.label\_moviename.setObjectName("label\_moviename")

           self.verticalLayout.addWidget(self.label\_moviename)

           self.label\_starttime = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_starttime.setFont(font)

           self.label\_starttime.setObjectName("label\_starttime")

           self.verticalLayout.addWidget(self.label\_starttime)

           self.label\_endtime = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_endtime.setFont(font)

           self.label\_endtime.setObjectName("label\_endtime")

           self.verticalLayout.addWidget(self.label\_endtime)

           self.label\_Price = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_Price.setFont(font)

           self.label\_Price.setObjectName("label\_Price")

           self.verticalLayout.addWidget(self.label\_Price)

           self.pushButton\_buy = QtWidgets.QPushButton(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(14)

           font.setBold(True)

           font.setWeight(75)

           self.pushButton\_buy.setFont(font)

           self.pushButton\_buy.setObjectName("pushButton\_buy")

           self.pushButton\_buy.clicked.connect(self.buy)

           self.verticalLayout.addWidget(self.pushButton\_buy)

           self.retranslateUi(Dialog)

           QtCore.QMetaObject.connectSlotsByName(Dialog)

       def buy(self):

           MAID = self.lineEdit\_MAID.text()

           flag = self.buyIsMAID(MAID)

           if flag != None:

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   # 查找放映厅

                   sql0 = '''SELECT HallID FROM MovieArrangement WHERE ID={} '''.format(MAID)

                   cur.execute(sql0)

                   HallID = cur.fetchone()

                   HallID = HallID[0]

                   #print(HallID)

                   # 查看该放映厅的票数

                   sql = '''SELECT Seats FROM ScreenHall  WHERE ID ={} '''.format(HallID)

                   cur.execute(sql)

                   seatnumber = cur.fetchone()

                   seatnumber=seatnumber[0]

                   #print(seatnumber)

                   if int(seatnumber)>=1:

                       # 查看UserName

                       sql2 = '''SELECT UserID FROM MovieArrangement WHERE ID = {}'''.format(MAID)

                       cur.execute(sql2)

                       UserName = cur.fetchone()

                       UserName=UserName[0]

                       #print(UserName)

                       # 购票操作（一个人可以购买多张电影票）

                       sql3 = """INSERT INTO Ticket (ID,UserID) VALUES (?,?)"""

                       value = (MAID,UserName)

                       cur.execute(sql3,value)

                       # 将该放映厅中的余票数减一

                       seatnumber-=1

                       sql4 = "UPDATE ScreenHall SET Seats = {} WHERE ID   = {} ".format(seatnumber,HallID)

                       cur.execute(sql4)

                       #展示电影票的一些信息

                       sql5 = '''

                       SELECT

                       M.MovieName,MA.StartTime,MA.EndTime,M.Price

                       FROM

                       Movie M,(SELECT \* FROM MovieArrangement WHERE ID = {})MA

                       WHERE

                       M.MovieID = MA.MovieID'''.format(MAID)

                       cur.execute(sql5)

                       result1 = cur.fetchone()

                       self.label\_moviename.setText("MovieName: "+result1[0])

                       self.label\_starttime.setText("StartTime: "+result1[1])

                       self.label\_endtime.setText("EndTime: "+result1[2])

                       self.label\_Price.setText("Price: "+str(result1[3]) )

                       conn.commit()

                   else:

                       self.label\_moviename.setText("All tickets are sold out.Please choose another movie from Movie Menu")

                       self.label\_starttime.setText("")

                       self.label\_endtime.setText("")

                       self.label\_Price.setText("")

               except Error as e:

                   print(e)

           else:

               self.label\_moviename.setText("Please Enter the Right MAID")

               self.label\_starttime.setText("")

               self.label\_endtime.setText("")

               self.label\_Price.setText("")

       # 检测输入的MAID 是否在MocieMenu 中。

       def buyIsMAID(self,MAID):

           flag  = None

           try :

               conn  = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = "SELECT ID FROM MovieArrangement WHERE ID = {}".format(MAID)

               cur.execute(sql)

               flag = cur.fetchone()

               conn.commit()

           except sqlite3.Error as e:

               print(e)

           return flag

       def retranslateUi(self, Dialog):

           \_translate = QtCore.QCoreApplication.translate

           Dialog.setWindowTitle(\_translate("Dialog", "Buy Ticket"))

           self.lineEdit\_MAID.setPlaceholderText(\_translate("Dialog", "Please Enter the MAID"))

           self.label\_moviename.setText(\_translate("Dialog", "Movie Name:"))

           self.label\_starttime.setText(\_translate("Dialog", "Starttime:"))

           self.label\_endtime.setText(\_translate("Dialog", "EndTime:"))

           self.label\_Price.setText(\_translate("Dialog", "Price:"))

           self.pushButton\_buy.setText(\_translate("Dialog", "Buy"))

   if \_\_name\_\_ == "\_\_main\_\_":

       import sys

       app = QtWidgets.QApplication(sys.argv)

       Dialog = QtWidgets.QDialog()

       ui = BUY\_Dialog()

       ui.setupUi(Dialog)

       Dialog.show()

       sys.exit(app.exec\_())

   **退票界面：**

   # -\*- coding: utf-8 -\*-

   # Form implementation generated from reading ui file 'bounceTicket.ui'

   #

   # Created by: PyQt5 UI code generator 5.15.4

   #

   # WARNING: Any manual changes made to this file will be lost when pyuic5 is

   # run again.  Do not edit this file unless you know what you are doing.

   import sqlite3

   from PyQt5 import QtCore, QtGui, QtWidgets

   class Bounce\_Dialog(object):

       def setupUi(self, Dialog):

           Dialog.setObjectName("Dialog")

           Dialog.resize(500, 500)

           self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)

           self.verticalLayout.setObjectName("verticalLayout")

           self.lineEdit\_MAID = QtWidgets.QLineEdit(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.lineEdit\_MAID.setFont(font)

           self.lineEdit\_MAID.setObjectName("lineEdit\_MAID")

           #self.lineEdit\_MAID.returnPressed.connect(self.bounceinfo)

           self.verticalLayout.addWidget(self.lineEdit\_MAID)

           self.label\_moviename = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_moviename.setFont(font)

           self.label\_moviename.setObjectName("label\_moviename")

           self.verticalLayout.addWidget(self.label\_moviename)

           self.label\_starttime = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_starttime.setFont(font)

           self.label\_starttime.setObjectName("label\_starttime")

           self.verticalLayout.addWidget(self.label\_starttime)

           self.label\_endtime = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_endtime.setFont(font)

           self.label\_endtime.setObjectName("label\_endtime")

           self.verticalLayout.addWidget(self.label\_endtime)

           self.label\_Price = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_Price.setFont(font)

           self.label\_Price.setObjectName("label\_Price")

           self.verticalLayout.addWidget(self.label\_Price)

           self.pushButton\_bounce = QtWidgets.QPushButton(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(14)

           font.setBold(True)

           font.setWeight(75)

           self.pushButton\_bounce.setFont(font)

           self.pushButton\_bounce.setObjectName("pushButton\_bounce")

           self.pushButton\_bounce.clicked.connect(self.bounceTicket)

           self.verticalLayout.addWidget(self.pushButton\_bounce)

           self.retranslateUi(Dialog)

           QtCore.QMetaObject.connectSlotsByName(Dialog)

       # 退票前检测是否买票

       def checkifbuy(self,MAID):

           flag  = None

           try :

               conn  = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = "SELECT ID FROM Ticket  WHERE ID = {}".format(MAID)

               cur.execute(sql)

               flag = cur.fetchone()

               conn.commit()

           except sqlite3.Error as e:

               print(e)

           return flag

       # 检测输入的MAID 是否在MocieMenu 中。

       def bounceIsMAID(self,MAID):

           flag  = None

           try :

               conn  = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = "SELECT ID FROM MovieArrangement WHERE ID = {}".format(MAID)

               cur.execute(sql)

               flag = cur.fetchone()

               conn.commit()

           except sqlite3.Error as e:

               print(e)

           return flag

       def bounceTicket(self):

           MAID = self.lineEdit\_MAID.text()

           # 判断输入是否买了该票

           flag = self.checkifbuy(MAID)

           if flag!= None:

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   # 从电影票表中查找是否有订单

                   sql2 = '''

                   SELECT TicketID FROM Ticket WHERE ID = {} '''.format(MAID)

                   cur.execute(sql2)

                   result1 = cur.fetchone()

                   if result1!=None:

                       for res in result1:

                           result = res

                       sql3 = '''

                       DELETE FROM Ticket WHERE TicketID = {}'''.format(result)

                       cur.execute(sql3)

                       #根据MAID 查找放映厅，将放映厅中余票数加一（即退票操作）

                       sql1 = '''

                       UPDATE

                       ScreenHall

                       SET

                       Seats=Seats+1

                       WHERE

                       ID = (SELECT HallID FROM MovieArrangement WHERE ID ={} )'''.format(MAID)

                       cur.execute(sql1)

                       # 展示电影的信息，并告知删除成功

                       self.bounceinfo()

                       self.label\_Price.setText(self.label\_Price.text()+"Bounce Successfully!")

                       conn.commit()

                   else:

                       self.label\_moviename.setText("You did't buy this Ticket!!!!!")

                       self.label\_starttime.setText("")

                       self.label\_endtime.setText("")

                       self.label\_Price.setText("")

               except sqlite3.Error as e:

                   print(e)

           else:

               self.label\_moviename.setText("You didn't buy this Tickets! Please buy ticket firstly")

               self.label\_starttime.setText("")

               self.label\_endtime.setText("")

               self.label\_Price.setText("")

       def bounceinfo(self):

           MAID = self.lineEdit\_MAID.text()

           try:

               conn = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = '''

                       SELECT

                       M.MovieName,MA.StartTime,MA.EndTime,M.Price

                       FROM

                       Movie M,(SELECT \* FROM MovieArrangement WHERE ID = {})MA

                       WHERE

                       M.MovieID = MA.MovieID'''.format(MAID)

               cur.execute(sql)

               result = cur.fetchone()

               self.label\_moviename.setText("Movie Name: " +result[0])

               self.label\_starttime.setText("Starttime: "+ result[1])

               self.label\_endtime.setText("EndTime: "+ result[2])

               self.label\_Price.setText("Price: "+ str(result[3])+"        ")

               conn.commit()

           except sqlite3.Error as e:

               print(e)

       def retranslateUi(self, Dialog):

           \_translate = QtCore.QCoreApplication.translate

           Dialog.setWindowTitle(\_translate("Dialog", "Bounce Ticket"))

           self.lineEdit\_MAID.setPlaceholderText(\_translate("Dialog", "Please Enter the MAID"))

           self.label\_moviename.setText(\_translate("Dialog", "Movie Name:"))

           self.label\_starttime.setText(\_translate("Dialog", "Starttime:"))

           self.label\_endtime.setText(\_translate("Dialog", "EndTime:"))

           self.label\_Price.setText(\_translate("Dialog", "Price:"))

           self.pushButton\_bounce.setText(\_translate("Dialog", "Bounce"))

   if \_\_name\_\_ == "\_\_main\_\_":

       import sys

       app = QtWidgets.QApplication(sys.argv)

       Dialog = QtWidgets.QDialog()

       ui = Bounce\_Dialog()

       ui.setupUi(Dialog)

       Dialog.show()

       sys.exit(app.exec\_())

   **查看电影票界面：**

   # -\*- coding: utf-8 -\*-

   # Form implementation generated from reading ui file 'ViewTicket.ui'

   #

   # Created by: PyQt5 UI code generator 5.15.4

   #

   # WARNING: Any manual changes made to this file will be lost when pyuic5 is

   # run again.  Do not edit this file unless you know what you are doing.

   from PyQt5 import QtCore, QtGui, QtWidgets

   import sqlite3

   class view\_Dialog(object):

       def setupUi(self, Dialog):

           Dialog.setObjectName("Dialog")

           Dialog.resize(500, 500)

           self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)

           self.verticalLayout.setObjectName("verticalLayout")

           self.horizontalLayout = QtWidgets.QHBoxLayout()

           self.horizontalLayout.setObjectName("horizontalLayout")

           self.lineEdit\_MAID = QtWidgets.QLineEdit(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.lineEdit\_MAID.setFont(font)

           self.lineEdit\_MAID.setObjectName("lineEdit\_MAID")

           self.horizontalLayout.addWidget(self.lineEdit\_MAID)

           self.pushButton\_viewTicket = QtWidgets.QPushButton(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(14)

           font.setBold(True)

           font.setWeight(75)

           self.pushButton\_viewTicket.setFont(font)

           self.pushButton\_viewTicket.setObjectName("pushButton\_viewTicket")

           self.pushButton\_viewTicket.clicked.connect(self.viewTicket)

           self.horizontalLayout.addWidget(self.pushButton\_viewTicket)

           self.verticalLayout.addLayout(self.horizontalLayout)

           self.label\_moviename = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_moviename.setFont(font)

           self.label\_moviename.setObjectName("label\_moviename")

           self.verticalLayout.addWidget(self.label\_moviename)

           self.label\_startTime = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_startTime.setFont(font)

           self.label\_startTime.setObjectName("label\_startTime")

           self.verticalLayout.addWidget(self.label\_startTime)

           self.label\_Endtime = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_Endtime.setFont(font)

           self.label\_Endtime.setObjectName("label\_Endtime")

           self.verticalLayout.addWidget(self.label\_Endtime)

           self.label\_ScreenHall = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_ScreenHall.setFont(font)

           self.label\_ScreenHall.setObjectName("label\_ScreenHall")

           self.verticalLayout.addWidget(self.label\_ScreenHall)

           self.label\_warning = QtWidgets.QLabel(Dialog)

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_warning.setFont(font)

           self.label\_warning.setObjectName("label\_warning")

           self.verticalLayout.addWidget(self.label\_warning)

           self.retranslateUi(Dialog)

           QtCore.QMetaObject.connectSlotsByName(Dialog)

       # 检测输入的MAID 是否在MocieMenu 中。

       def ticketIsMAID(self,MAID):

           flag  = None

           try :

               conn  = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = "SELECT ID FROM MovieArrangement WHERE ID = {}".format(MAID)

               cur.execute(sql)

               flag = cur.fetchone()

               conn.commit()

           except sqlite3.Error as e:

               print(e)

           return flag

       def viewTicket(self):

           MAID = self.lineEdit\_MAID.text()

           flag = self.ticketIsMAID(MAID)

           if flag!= None:

               try:

                   conn = sqlite3.connect("cinema.db")

                   cur = conn.cursor()

                   sql ='''

                   SELECT

                   M.MovieName,MA.StartTime,MA.EndTime,MA.HallID

                   FROM

                   Movie M,MovieArrangement MA

                   WHERE

                   M.MovieID=MA.MovieID and MA.ID = {}'''.format(MAID)

                   cur.execute(sql)

                   result= cur.fetchone()

                   self.label\_moviename.setText("Movie Name: "+result[0])

                   self.label\_startTime.setText("Starttime: "+ result[1])

                   self.label\_Endtime.setText("EndTime: "+ result[2])

                   self.label\_ScreenHall.setText("ScreenHall No. "+ str(result[3]))

                   self.label\_warning.setText("Enjoy your movie!!!!")

                   conn.commit()

               except sqlite3.Error as e :

                   print(e)

           else:

               self.label\_warning.setText("Warning : Please Enter Right MAID!")

               self.label\_moviename.setText("")

               self.label\_startTime.setText("")

               self.label\_Endtime.setText("")

               self.label\_ScreenHall.setText("")

       def retranslateUi(self, Dialog):

           \_translate = QtCore.QCoreApplication.translate

           Dialog.setWindowTitle(\_translate("Dialog", "View Ticket"))

           self.lineEdit\_MAID.setPlaceholderText(\_translate("Dialog", "Please Enter the MAID"))

           self.pushButton\_viewTicket.setText(\_translate("Dialog", "View Ticket"))

           self.label\_moviename.setText(\_translate("Dialog", "Movie Name:"))

           self.label\_startTime.setText(\_translate("Dialog", "Starttime:"))

           self.label\_Endtime.setText(\_translate("Dialog", "EndTime:"))

           self.label\_ScreenHall.setText(\_translate("Dialog", "ScreenHall"))

           self.label\_warning.setText(\_translate("Dialog", ""))

   if \_\_name\_\_ == "\_\_main\_\_":

       import sys

       app = QtWidgets.QApplication(sys.argv)

       Dialog = QtWidgets.QDialog()

       ui = view\_Dialog()

       ui.setupUi(Dialog)

       Dialog.show()

       sys.exit(app.exec\_())

   **评论界面：**

   # -\*- coding: utf-8 -\*-

   # Form implementation generated from reading ui file 'comment.ui'

   #

   # Created by: PyQt5 UI code generator 5.15.4

   #

   # WARNING: Any manual changes made to this file will be lost when pyuic5 is

   # run again.  Do not edit this file unless you know what you are doing.

   from PyQt5 import QtCore, QtGui, QtWidgets

   import sqlite3

   class comment\_Dialog(object):

       def setupUi(self, Dialog):

           Dialog.setObjectName("Dialog")

           Dialog.resize(500, 500)

           self.lineEdit\_MAID = QtWidgets.QLineEdit(Dialog)

           self.lineEdit\_MAID.setGeometry(QtCore.QRect(11, 11, 230, 30))

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.lineEdit\_MAID.setFont(font)

           self.lineEdit\_MAID.setObjectName("lineEdit\_MAID")

           self.label\_moviename = QtWidgets.QLabel(Dialog)

           self.label\_moviename.setGeometry(QtCore.QRect(11, 48, 400, 24))

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(12)

           font.setBold(True)

           font.setWeight(75)

           self.label\_moviename.setFont(font)

           self.label\_moviename.setObjectName("label\_moviename")

           self.pushButton\_commit = QtWidgets.QPushButton(Dialog)

           self.pushButton\_commit.setGeometry(QtCore.QRect(11, 452, 93, 37))

           font = QtGui.QFont()

           font.setFamily("Agency FB")

           font.setPointSize(14)

           font.setBold(True)

           font.setWeight(75)

           self.pushButton\_commit.setFont(font)

           self.pushButton\_commit.setObjectName("pushButton\_commit")

           # give comments

           self.pushButton\_commit.clicked.connect(self.givecomments)

           self.textEdit\_comments = QtWidgets.QTextEdit(Dialog)

           self.textEdit\_comments.setGeometry(QtCore.QRect(10, 90, 441, 281))

           self.textEdit\_comments.setObjectName("textEdit\_comments")

           self.retranslateUi(Dialog)

           QtCore.QMetaObject.connectSlotsByName(Dialog)

       def checkifbuy(self,MAID):

           flag  = None

           try :

               conn  = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = "SELECT ID FROM Ticket  WHERE ID = {}".format(MAID)

               cur.execute(sql)

               flag = cur.fetchone()

               conn.commit()

           except sqlite3.Error as e:

               print(e)

           return flag

       def givecomments(self):

           MAID = self.lineEdit\_MAID.text()

           # # 先查看自己有没有买电影票，否则不能给评价

           flag = self.checkifbuy(MAID)

           if flag!= None:

               comment = self.textEdit\_comments.toPlainText()

               if comment!= "":

                   try:

                       conn = sqlite3.connect("cinema.db")

                       cur = conn.cursor()

                       sql0 = '''SELECT MovieID,UserID FROM  MovieArrangement WHERE ID = {}'''.format(MAID)

                       cur.execute(sql0)

                       result = cur.fetchone()

                       MovieID,UserID = result[0],result[1]

                       sql00 = '''SELECT MovieName FROM Movie WHERE MovieID = {}'''.format(MovieID)

                       cur.execute(sql00)

                       MovieName = cur.fetchone()

                       for mn in MovieName:

                           res = mn

                       MovieName = res

                       sql = '''

                       INSERT INTO Comments (MovieID,UserID,Content) VALUES (?,?,?)'''

                       value = (MovieID,UserID,comment)

                       cur.execute(sql,value)

                       conn.commit()

                       self.label\_moviename.setText("{}       Give Comments successfully!!!!".format(MovieName))

                   except sqlite3.Error as e:

                       print(e)

               else :

                   self.label\_moviename.setText("Please give Comments!!!!")

           else:

               self.label\_moviename.setText("You didn't buy this Tickets! Please buy ticket firstly")

       def commentsIsMAID(self,MAID):

           flag  = None

           try :

               conn  = sqlite3.connect("cinema.db")

               cur = conn.cursor()

               sql = "SELECT ID FROM MovieArrangement WHERE ID = {}".format(MAID)

               cur.execute(sql)

               flag = cur.fetchone()

               conn.commit()

           except sqlite3.Error as e:

               print(e)

           return flag

       def retranslateUi(self, Dialog):

           \_translate = QtCore.QCoreApplication.translate

           Dialog.setWindowTitle(\_translate("Dialog", "Bounce Ticket"))

           self.lineEdit\_MAID.setPlaceholderText(\_translate("Dialog", "Please Enter the MAID"))

           self.label\_moviename.setText(\_translate("Dialog", "Movie Name:"))

           self.pushButton\_commit.setText(\_translate("Dialog", "Commit"))

   if \_\_name\_\_ == "\_\_main\_\_":

       import sys

       app = QtWidgets.QApplication(sys.argv)

       Dialog = QtWidgets.QDialog()

       ui = comment\_Dialog()

       ui.setupUi(Dialog)

       Dialog.show()

       sys.exit(app.exec\_()) [↑](#endnote-ref-1)