

**Online Film ticket purchase system**



Profession Class : B4

Team Name : SnowWhite

Team Members :

ChenYang（Leader）:R8J9400014

CaiSiyuan:R8J9400020

WangJingru:R8J9400033

SunRuitong:R8J9400032

计算机与网络空间安全学院

软件工程（NIIT）四班

dialog

project description..........................- 3 -

1.1 Project background....................- 3 -

1.2 Project characteristics...............- 4 -

1. Team Profile...............................- 4 -

2.Team spirit............................- 4 -

2.2 division of labor.....................- 4 -

1. Project development and Prospects...........- 5 -

3.1Project R&D and related concepts.......- 5 -

3.2 Project prospect................ ......- 7 -

3.3 Project advantage......................- 7 -

1. Market analysis..............................- 8 -

4.1 Market situation........................- 8 -

4.2 market prospect.........................- 9 -

4.3 SWOT analyze...........................- 10 -

4.3.1 advantage....................... ..- 10 -

4.3.2 Inferiority.......................- 10 -

4.3.3Opportunities and Threats..........- 10 -

**Introduction**

with the rapid development of social economy, people's economic level has improved significantly, and the quality of life has also undergone qualitative changes. However, people's work pressure has also increased in the invisible, in such a fast-paced work environment, people are eager to slow down in order to better adjust their pace. People are adjusting themselves in a variety of ways to reduce stress, but movies are undoubtedly the choice of most people. A good movie can really make people physically and mentally happy, shocking sound effects are also obsessed with, as long as the lower price can be happy within 2 hours. Film can now be said to be the best choice for leisure and entertainment, the age span is very large. Looking back on the old cinema, I believe everyone has some feelings. It was troublesome for people to want to see a movie at that time. It is not clear when the film will be broadcast, and it is not clear whether there are any tickets left for a certain film, and I have to go to the cinema to queue up to buy tickets. Maybe I have been filming for a long time, but when it comes to myself, I can't buy the one I want. This will undoubtedly waste customers a lot of valuable time, in this era of time is money, this is unacceptable to many people. On the other hand, the demand for movies is so great that traditional cinemas are unable to meet the needs of traditional cinemas. With the help of the rapid development of Internet technology, online cinemas emerge as the times require ~ the so-called online cinemas, as the name suggests, is the online cinema on the Internet, handheld cinema. It integrates most of the functions of the cinema, people as long as sliding fingertip film information will appear in front of us, with a quick view of time, quickly get the information they want. Through the online cinema, people can obtain film information, movie ticket information, online seat selection and other functions, which greatly save consumers valuable time. The online film ticketing system is to centrally manage some of the information on the web page that needs to be changed frequently, such as film information, film ticketing information, and so on, and classify them by some of the commonalities of the information. Finally, a systematic and standardized website application published on a website. The website information is added to the database through a simple interface, and then published to the website through the existing web template format and audit process. With the "Cinema online Film ticketing system", it is very convenient for customers to lie at home on the sofa and view cinema movie information on the Internet, and buy the movie tickets they want to watch online according to their own wishes. This is undoubtedly very popular with modern customers, and it obviously provides customers with a convenient, comfortable and time-saving way to buy tickets.

1. **项目描述**

**1.1项目背景**

Most families in China now like to buy their own home theaters at home. However, there are still many people who like to go to the cinema to see movies. Because after all, the atmosphere at home is not as good as the cinema. Therefore, the emergence of modern home cinema will not make it impossible for cinemas to survive. In order to improve the efficiency of labor, save costs and improve the quality of service, our team developed this project. It is used to facilitate the sale of movie tickets and the purchase of customers. Through this software, you can quickly achieve some common services, and ensure that there are no mistakes, efficient. Each social service system has its own set of management mechanisms. Of course, the cinema is not excluded. In fact, the management system of the cinema should be a little simpler than the management of other social service systems. Cinemas are nothing more than introducing movies, setting a timetable for showing them, buying tickets, checking in, and some data management. Such as staff management, ticketing management and so on. In order to make the working mechanism simple and orderly, it is necessary to introduce a set of tubes.At a time when computers are popular in the world, computers can do more for us. In the past, if we wanted to see the latest film in the cinema, we had to spend most of our time and energy waiting in line to buy tickets, which was time-consuming and laborious, and that kind of waiting was annoying. In order to solve the problem of buying tickets, we can complete it through computer and Internet. This system is developed and designed according to this requirement. We develop a secure and stable online movie ticket booking system to meet the needs of users and provide users with convenient services.

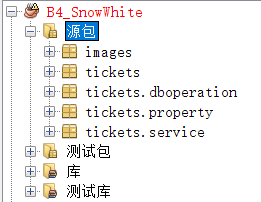
**1.2 项目特征**

The online film ticket purchase system is to improve the efficiency of labor, save costs, improve the quality of service, and facilitate film ticket sales and customer purchase. The online movie ticketing system involves the functions of unregistered user registration, registered user login, movie query, order creation, payment and so on, as well as the management and operation of the database to realize the task of the project as a whole. System roles are divided into unregistered users, registered users and administrators, each corresponding to the corresponding tasks to achieve functions.

1. **团队简介**
   1. **团队精神**

Team spirit: unite and help each other, dare to be the first.

**2.2团队成员具体分工**



陈阳：do the dboperation package and service package

蔡思远：design the UI and project report

王静如：design the UI 、flow chart and PPT

孙瑞佟：do the property package and project report

1. **项目研发**

In the final analysis, what cinemas really need is the volume of business, that is, to maximize the number of viewers. Therefore, the business management system of the cinema is particularly important, and it can not be ignored in the whole management process of the cinema. Whether there is a strong and comprehensive business management system is already a trend and an indicator of the success of a cinema. Java SE development technology has been quite mature today, the online film ticketing system officially uses the B / S mode architecture, and uses the combination of Sqlserver database to achieve.

**研究方法**

1.literature research method: consult the relevant literature.

2.Experience summary method: self-analysis and summary of the experimental results. 3.Comparative analysis: the situation at home and abroad is compared.

The online film ticketing system mainly realizes the cinema staff to the movie ticket management, the inquiry and the statistics and so on function. The system adopts B / S mode as the architecture, Sqlserver as the background database and Java language as the development language of user interface, so as to realize the related functions of film ticketing information.

**设计思路：**

For the design of the online film ticketing system, first of all, we have a charging understanding of the relevant process information of cinema movie tickets, and then adopt the current mainstream information processing means. In order to be able to better team movie tickets for a systematic management, so that the majority of users can get the most convenient and comfortable experience, and finally be able to truly integrate film tickets and Internet technology. The system also strictly adheres to the relevant regulations of cinema movie tickets, has the fixed mode programming, the administrator background can carry on the operation to it, such as adding and deleting and so on. It is similar to today's e-commerce online shopping platform, after entering the home page, if necessary, visitors can register for the operation of users of this website, after becoming users of the website, in addition to viewing the relevant cinema information and movie tickets, You can also buy multiple movie tickets online.

**流程图:**



**主要代码:**

**java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**jButton1ActionPerformed(evt);**

**}**

**});**

**jButton2.setBackground(new java.awt.Color(255, 0, 51));**

**jButton2.setFont(new java.awt.Font("幼圆", 3, 24)); // NOI18N**

**jButton2.setText("电影");**

**jsearch\_nametext.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**jsearch\_jbutton.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**jsearch\_jbutton.setText("搜索");**

**jsearch\_jbutton.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jsearch\_jbuttonMouseClicked(evt);**

**}**

**});**

**enterMUI\_btn.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**enterMUI\_btn.setText("进入");**

**enterMUI\_btn.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**moviecard1\_btnMouseClicked(evt);**

**}**

**});**

**movietb=new JTable(){**

**public boolean isCellEditable( int row, int column) {**

**return false;**

**}**

**};**

**movietb.setModel(new DefaultTableModel(**

**new Object[][] {**

**},**

**new String[] {**

**"电影编号", "电影名字", "语言", " 地区","时长","类别","分数"**

**}**

**) {**

**boolean[] columnEditables = new boolean[] {**

**false, false, false, false**

**};**

**public boolean isCellEditable(int row, int column) {**

**return columnEditables[column];**

**}**

**});**

**fillTable( movielist);**

**movietb.setRowHeight(70);**

**movietb.setFont(font);**

**movietb.getTableHeader().setFont(new Font("楷体", 1, 20));**

**movietb.getTableHeader().setBackground(Color.orange);**

**movietb.getTableHeader().setReorderingAllowed(false); // 不可交换顺序**

**movietb.getTableHeader().setResizingAllowed(false); // 不可拉动表格**

**movietb.addMouseListener(new MouseAdapter() {**

**@Override**

**public void mousePressed(MouseEvent me) {**

**movietbmousePressed(me);**

**}**

**});**

**JScrollPane moviesPane = new JScrollPane(movietb);**

**moviesPane.setBounds(0, 510, 1800, 450);**

**this.add(moviesPane);**

**javax.swing.GroupLayout jp1Layout = new javax.swing.GroupLayout(jp1);**

**jp1.setLayout(jp1Layout);**

**jp1Layout.setHorizontalGroup(**

**jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addGroup(jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addComponent(jLabel1)**

**.addGroup(jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addGap(142, 142, 142)**

**.addComponent(jButton1)**

**.addGap(91, 91, 91)**

**.addComponent(jButton2))**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addGap(27, 27, 27)**

**.addComponent(jLabel2))))**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addGap(95, 95, 95)**

**.addComponent(jsearch\_nametext, javax.swing.GroupLayout.PREFERRED\_SIZE, 216, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(64, 64, 64)**

**.addComponent(jsearch\_jbutton, javax.swing.GroupLayout.PREFERRED\_SIZE, 97, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(18, 18, 18)**

**.addComponent(enterMUI\_btn, javax.swing.GroupLayout.PREFERRED\_SIZE, 97, javax.swing.GroupLayout.PREFERRED\_SIZE)))**

**.addGap(222, 413, Short.MAX\_VALUE))**

**);**

**jp1Layout.setVerticalGroup(**

**jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addGroup(jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jLabel1)**

**.addGroup(jp1Layout.createSequentialGroup()**

**.addGap(53, 53, 53)**

**.addComponent(jLabel2)**

**.addGap(57, 57, 57)**

**.addGroup(jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)**

**.addComponent(jButton1)**

**.addComponent(jButton2))))**

**.addGap(74, 74, 74)**

**.addGroup(jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)**

**.addComponent(jsearch\_nametext)**

**.addGroup(jp1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)**

**.addComponent(jsearch\_jbutton, javax.swing.GroupLayout.PREFERRED\_SIZE, 51, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(enterMUI\_btn, javax.swing.GroupLayout.PREFERRED\_SIZE, 51, javax.swing.GroupLayout.PREFERRED\_SIZE)))**

**.addContainerGap(1262, Short.MAX\_VALUE))**

**);**

**jScrollPane1.setViewportView(jp1);**

**javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());**

**getContentPane().setLayout(layout);**

**layout.setHorizontalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 1019, Short.MAX\_VALUE)**

**);**

**layout.setVerticalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 1096, Short.MAX\_VALUE)**

**);**

**pack();**

**}// </editor-fold>**

**private void jsearch\_jbuttonMouseClicked(java.awt.event.MouseEvent evt) {**

**// TODO add your handling code here:**

**movielist=mo.queryMovieBymovieName(jsearch\_nametext.getText());**

**if(movielist!=null) {**

**fillTable(movielist);**

**}**

**}**

**protected void movietbmousePressed(MouseEvent me) {**

**// TODO Auto-generated method stub**

**row1=movietb.getSelectedRow();**

**}**

**private void fillTable(List<Movie> movielist2) {**

**// TODO Auto-generated method stub**

**DefaultTableModel dtm=(DefaultTableModel) movietb.getModel();**

**dtm.setRowCount(0); // 设置成0行**

**int movieRow= movielist2.size();**

**for(int i=0;i<movieRow;i++) {**

**Vector v=new Vector();**

**v.add(movielist2.get(i).getmId());**

**v.add(movielist2.get(i).getmName());**

**v.add(movielist2.get(i).getMlanguage());**

**v.add(movielist2.get(i).getMlocality());**

**v.add(movielist2.get(i).getmDuration());**

**v.add(movielist2.get(i).getmType());**

**v.add(movielist2.get(i).getmGrade());**

**dtm.addRow(v);**

**}**

**DefaultTableCellRenderer r = new DefaultTableCellRenderer();**

**r.setHorizontalAlignment(JLabel.CENTER);**

**}**

**private void moviecard1\_btnMouseClicked(java.awt.event.MouseEvent evt) {**

**// TODO add your handling code here:**

**if(row1!=-1) {**

**new MovieUi(movielist.get(row1),this.user);**

**dispose();**

**}else {**

**JOptionPane.showMessageDialog(null, "请选择电影");**

**}**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**HomeUi1 hp=new HomeUi1(user);**

**hp.setVisible(true);**

**dispose();**

**}**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(HomeUi2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(HomeUi2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(HomeUi2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(HomeUi2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**private void textPsActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**}**

**public static void winMessage(String str) {// 提示窗口，有多个地方调用**

**JOptionPane.showMessageDialog(null, str, "提示",**

**JOptionPane.INFORMATION\_MESSAGE);**

**}**

**public void cleanUserInfo() {**

**this.textName.setText("");**

**this.textPs.setText("");**

**this.textCode.setText("");**

**}**

**private String code() {**

**int m = 0;**

**for (int i = 0; i < 4; i++) {**

**m \*= 10;**

**m += (int) (Math.random() \* 9 + 1);**

**}**

**return ((Integer) m).toString();**

**}**

**private void Button1ActionPerformed(java.awt.event.ActionEvent evt) {**

**String id = textName.getText();**

**String pswd = new String(textPs.getPassword());**

**if (id.equals("") || pswd.equals("")) {**

**LoginUi.winMessage("账号、密码不能为空！");**

**cleanUserInfo();**

**this.code2.setText(code());**

**} else {**

**String code1 = textCode.getText();**

**String code = code2.getText();**

**if (code1.equals(code)) {**

**int choice=0;**

**if (choice1.isSelected())**

**choice = 0;**

**else**

**choice = 1;**

**UserService userService = new UserService();**

**User user = userService.login(new User(id, pswd, choice));**

**if (user == null) {**

**LoginUi.winMessage("用户名或密码不正确！登录失败！");**

**cleanUserInfo();**

**this.code2.setText(code());**

**} else {**

**user=userService.queryUser(user);**

**if (user.getuType()==0) {**

**HomeUi1 h=new HomeUi1(user);**

**h.setVisible(true);**

**} else if (user.getuType()==1) {**

**// new AdminUi();**

**}**

**this.dispose();**

**LoginUi.winMessage("登录成功！");**

**}**

**} else {**

**LoginUi.winMessage("验证码不正确！");**

**cleanUserInfo();// 可以考虑不清空信息**

**this.code2.setText(code());**

**}**

**}**

**}**

**private void Button2MouseClicked(java.awt.event.MouseEvent evt) {**

**RegisterUi r=new RegisterUi();**

**r.setVisible(true);**

**}**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(LoginUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(LoginUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(LoginUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(LoginUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new LoginUi().setVisible(true);**

**}**

**});**

**}**

**public class MovieUi {**

**jmDetail.setBackground(Color.white);**

**jreturn.setBounds(530, 700, 150, 50);**

**jreturn.setFont(font1);**

**jreturn.setBackground(Color.GRAY);**

**jreturn.setForeground(Color.black);**

**jreturn.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**returnActionPerformed(e);**

**}**

**});**

**jBuy.setBounds(1000, 700, 150, 50);**

**jBuy.setFont(font1);**

**jBuy.setBackground(Color.GRAY);**

**jBuy.setForeground(Color.white);**

**jBuy.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**buyActionPerformed(e);**

**}**

**});**

**User user=new User();**

**/\***

**\* maxLable.add(jPic); maxLable.add(jmName); maxLable.add(jmType);**

**\* maxLable.add(jmLoca); maxLable.add(jmDuration); maxLable.add(jmLanguage);**

**\* maxLable.add(jmDetail); maxLable.add(jreturn); maxLable.add(jBuy);**

**\* maxLable.setBounds(0, 0, 1500, 1500);**

**\*/**

**p.setBackground(Color.white);**

**p.add(jPic);**

**p.add(jmName);**

**p.add(jmType);**

**p.add(jmLoca);**

**p.add(jmDuration);**

**p.add(jmLanguage);**

**p.add(jmDetail);**

**p.add(jmGrade);**

**p.add(jreturn);**

**p.add(jBuy);**

**p.add(jcomment);**

**JScrollPane scrollPane\_1 = new JScrollPane();**

**scrollPane\_1.setBounds(830, 340, 600, 310);**

**p.add(scrollPane\_1);**

**scrollPane\_1.setViewportView(jmDetail);**

**jmDetail.setEditable(false);**

**p.setLayout(null);**

**/\*JScrollPane a = new JScrollPane(p);\*/**

**p.setPreferredSize(new Dimension(1500,1200));**

**JScrollPane a = new JScrollPane(p);**

**con.add(a);**

**jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**a.setVisible(true);**

**jf.setVisible(true);**

**a.setVisible(true);**

**}**

**protected void buyActionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**new SessionUi(movie, u);**

**jf.dispose();**

**}**

**protected void returnActionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**HomeUi1 uu=new HomeUi1(u);**

**uu.setVisible(true);**

**jf.dispose();**

**}**

**}**

**七.**

**MyTickets.java**

**public class MyTickets extends JFrame{**

**private JTable ticktb;**

**private int row = -1;**

**private List<Ticket> ticketlist;**

**private UserService uo=new UserService();**

**private MovieService mo = new MovieService();**

**private TicketService ticketservice=new TicketService();**

**private CinemaService cinemaservice=new CinemaService();**

**private SessionService sessionservice=new SessionService();**

**private HallService hallservice=new HallService();**

**private Font font = new Font("楷体", 0, 20);**

**private User user;**

**private JButton delTicket;**

**/\*\***

**\* Creates new form MyTickets**

**\*/**

**public MyTickets(User u) {**

**user=u;**

**initComponents();**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">**

**private void initComponents() {**

**delTicket=new JButton("退订");**

**delTicket.setFont(font);**

**delTicket.setBounds(1400, 40, 100, 50);**

**delTicket.addActionListener(new ActionListener(){**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**Ticket ticket;**

**if(row==-1) {**

**JOptionPane.showMessageDialog(null, "未选择影票");**

**}else {**

**ticket=ticketlist.get(row);**

**if(ticketservice.delTicket(ticket.gettId())) {**

**Session desession=sessionservice.querySessionById(ticket.getsId());**

**desession.setRemain(desession.getRemain()+1);**

**if(sessionservice.updateSession(desession)) {**

**JOptionPane.showMessageDialog(null, "退票成功");**

**user.setBalance(user.getBalance()+desession.getPrice());**

**uo.updateUser(user);**

**ticketlist.remove(row);**

**fillticktable(ticketlist);**

**}else {**

**JOptionPane.showMessageDialog(null, "退票失败");**

**}**

**}**

**else {**

**JOptionPane.showMessageDialog(null, "退票失败");**

**}**

**}**

**}**

**});**

**ticketlist=ticketservice.queryAllTicketuId(user.getUid());**

**setDefaultCloseOperation(javax.swing.WindowConstants.HIDE\_ON\_CLOSE);**

**javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());**

**getContentPane().setLayout(layout);**

**layout.setHorizontalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGap(0, 2000, Short.MAX\_VALUE)**

**);**

**layout.setVerticalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGap(0, 500, Short.MAX\_VALUE)**

**);**

**// ticketlist=ticketservice.queryAllTicketuId(user.getUid());**

**ticktb = new JTable() {**

**public boolean isCellEditable( int row, int column) {**

**return false;**

**}**

**};**

**ticktb.setModel(new DefaultTableModel(**

**new Object[][] {**

**},**

**new String[] {**

**"影票编号", "用户名", "影院名称","影院地址","场厅名字","电影名称","开始时间","价格","座位号"**

**}**

**) {**

**boolean[] columnEditables = new boolean[] {**

**false, false, false, false,false,false,false,false,false**

**};**

**public boolean isCellEditable(int row, int column) {**

**return columnEditables[column];**

**}**

**});**

**DefaultTableCellRenderer r = new DefaultTableCellRenderer();**

**fillticktable(ticketlist);**

**r.setHorizontalAlignment(JLabel.CENTER);**

**ticktb.setDefaultRenderer(Object.class, r);**

**ticktb.setRowHeight(90);**

**ticktb.setFont(font);**

**ticktb.getTableHeader().setFont(new Font("楷体", 1, 20));**

**ticktb.getTableHeader().setBackground(Color.orange);**

**ticktb.getTableHeader().setReorderingAllowed(false); // 不可交换顺序**

**ticktb.getTableHeader().setResizingAllowed(false); // 不可拉动表格**

**ticktb.addMouseListener(new MouseAdapter() {**

**@Override**

**public void mousePressed(MouseEvent me) {**

**ticktablemousePressed(me);**

**}**

**});**

**JScrollPane ticsPane = new JScrollPane(ticktb);**

**this.add(delTicket);**

**ticsPane.setBounds(0, 100, 2000, 500);**

**this.add(ticsPane);**

**pack();**

**}**

**public void ticktablemousePressed(MouseEvent me) {**

**// TODO Auto-generated method stub**

**Session csession;**

**row = ticktb.getSelectedRow();**

**if(row!=-1) {**

**csession=sessionservice.querySessionById(ticketlist.get(row).getsId());**

**}**

**}**

**public void fillticktable(List<Ticket> ticketlist2) {**

**// TODO Auto-generated method stub**

**DefaultTableModel dtm=(DefaultTableModel) ticktb.getModel();**

**dtm.setRowCount(0); // 设置成0行**

**int ticketRow = ticketlist.size();**

**Session session;**

**Hall hall;**

**Cinema cinema;**

**Movie tmovie;**

**for (int i = 0; i < ticketRow; i++) {**

**session=sessionservice.querySessionById(ticketlist.get(i).getsId());**

**hall=hallservice.queryHallById(session.gethId());**

**cinema=cinemaservice.queryCinemaById(session.getcId());**

**tmovie=mo.queryMovieById(session.getmId());**

**Vector v=new Vector();**

**v.add(ticketlist.get(i).gettId());**

**v.add(user.getuName());**

**v.add( cinema.getCname());**

**v.add(cinema.getCaddress());**

**v.add(hall.gethName());**

**v.add(tmovie.getmName());**

**v.add(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(session.getBeginTime()));**

**v.add(session.getPrice());**

**v.add(ticketlist.get(i).getSeat());**

**dtm.addRow(v);**

**}**

**}**

**}**

**八.**

**RegisterUi.java**

**private void textPs1ActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**}**

**private void Button1ActionPerformed(java.awt.event.ActionEvent evt) {**

**String pswd = new String(textPs1.getPassword());**

**String pswd2 = new String(textPs2.getPassword());**

**String name = textName.getText();**

**if ("".equals(name) || "".equals(pswd) || "".equals(pswd2)) {**

**LoginUi.winMessage("账号、密码不能为空！");**

**cleanUserInfo();**

**} else {**

**if (pswd.equals(pswd2)) {**

**int rs = userService.register(new User(name, pswd,0));**

**if (rs == 3) {**

**LoginUi.winMessage("注册失败，该用户名已存在！");**

**cleanUserInfo();**

**} else if(rs == 1) {**

**new LoginUi();**

**dispose();**

**LoginUi.winMessage("注册成功！");**

**} else {**

**LoginUi.winMessage("注册失败，原因。。。");**

**cleanUserInfo();**

**}**

**} else{**

**LoginUi.winMessage("两次输入的密码不一致！");**

**cleanUserInfo();**

**}**

**}**

**}**

**public void cleanUserInfo() {**

**this.textName.setText("");**

**this.textPs1.setText("");**

**this.textPs2.setText("");**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(RegisterUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(RegisterUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(RegisterUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(RegisterUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new RegisterUi().setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JButton Button1;**

**private javax.swing.JLabel jLabel1;**

**private javax.swing.JLabel jLabel2;**

**private javax.swing.JLabel jLabel3;**

**private javax.swing.JLabel jLabel4;**

**private javax.swing.JTextField textName;**

**private javax.swing.JPasswordField textPs1;**

**private javax.swing.JPasswordField textPs2;**

**// End of variables declaration**

**private UserService userService=new UserService();**

**}**

**九.**

**SessionUi.java**

**package tickets;**

**import java.awt.Color;**

**import java.awt.Container;**

**import java.awt.Font;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.awt.event.MouseAdapter;**

**import java.awt.event.MouseEvent;**

**import java.text.SimpleDateFormat;**

**import java.util.\*;**

**import javax.swing.\*;**

**import javax.swing.table.DefaultTableCellRenderer;**

**import javax.swing.table.DefaultTableModel;**

**import tickets.property.\*;**

**import tickets.service.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class SessionUi {**

**private Movie mo;**

**private User ue;**

**private SessionService seser = new SessionService();**

**private CinemaService cinema = new CinemaService();**

**private HallService hall = new HallService();**

**private Font font = new Font("楷体", 0, 20);**

**private Font font1 = new Font("楷体", 0, 28);**

**private JFrame jf = new JFrame("电影购票系统");**

**private JLabel jb = new JLabel();**

**private JTable tb;// 表**

**private int row = -1;// 行**

**private Container con = jf.getContentPane();**

**private JPanel p = new JPanel();**

**private JButton jBuyticket = new JButton("选座购票");**

**private JButton jreturn=new JButton("返回上一层");**

**private List<Session> sessionlist;**

**private JLabel search=new JLabel("影院名");**

**private JTextField serchText=new JTextField();**

**private JButton serbtn=new JButton("搜索场次");**

**public SessionUi(Movie mo, User ue) {**

**this.mo = mo;**

**this.ue = ue;**

**p.setLayout(null);**

**sessionlist = seser.querySessionBymId(mo.getmId());**

**int sessionRow = sessionlist.size();**

**Hall h = new Hall();**

**Cinema ci = new Cinema();**

**/\* String[][] rinfo = new String[sessionRow][8];**

**for (int i = 0; i < sessionRow; i++) {**

**h = hall.queryHallById(sessionlist.get(i).gethId());**

**ci = cinema.queryCinemaById(sessionlist.get(i).getcId());**

**rinfo[i][0] = String.valueOf(sessionlist.get(i).getsId());**

**rinfo[i][1] = h.gethName();**

**rinfo[i][2] = ci.getCname();**

**rinfo[i][3] = ci.getCaddress();**

**rinfo[i][4] = mo.getmName();**

**rinfo[i][5] = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(sessionlist.get(i).getBeginTime());**

**rinfo[i][6] = String.valueOf(sessionlist.get(i).getPrice());**

**rinfo[i][7] = String.valueOf(sessionlist.get(i).getRemain());**

**}**

**// 加上影院地址**

**String[] tbheadnames = { "场次编号", "场厅名称", "影院名称", "影院地址", "电影名称", "开始时间", "价格", "剩余座位数" };\*/**

**tb = new JTable() {**

**public boolean isCellEditable(int row, int column) {**

**return false;**

**}**

**};**

**tb.setModel(new DefaultTableModel(**

**new Object[][] {**

**},**

**new String[] {**

**"场次编号", "场厅名称", "影院名称","影院地址","电影名称","开始时间","价格","剩余座位数"**

**}**

**) {**

**boolean[] columnEditables = new boolean[] {**

**false, false, false, false,false,false,false,false**

**};**

**public boolean isCellEditable(int row, int column) {**

**return columnEditables[column];**

**}**

**});**

**DefaultTableCellRenderer r = new DefaultTableCellRenderer();**

**fillsessiontable(sessionlist);**

**r.setHorizontalAlignment(JLabel.CENTER);**

**tb.setDefaultRenderer(Object.class, r);**

**tb.setRowHeight(90);**

**tb.setFont(font);**

**/\* tb.setEnabled(true); \*/**

**tb.getTableHeader().setFont(new Font("楷体", 1, 20));**

**tb.getTableHeader().setBackground(Color.orange);**

**tb.getTableHeader().setReorderingAllowed(false); // 不可交换顺序**

**tb.getTableHeader().setResizingAllowed(false); // 不可拉动表格**

**tb.addMouseListener(new MouseAdapter() {**

**@Override**

**public void mousePressed(MouseEvent me) {**

**tablemousePressed(me);**

**}**

**});**

**JScrollPane sPane = new JScrollPane(tb);**

**jb.setBounds(0, 100, 1788, 795);**

**sPane.setBounds(0, 100, 1800, 800);**

**sPane.setVisible(true);**

**search.setBounds(200, 60, 150, 60);**

**search.setFont(font1);**

**serchText.setBounds(400, 60, 250, 60);**

**serchText.setFont(font1);**

**serbtn.setBounds(700, 60, 200, 60);**

**serbtn.setFont(font1);**

**serbtn.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**serachactionPerformed(e);**

**}**

**});**

**jreturn.setBounds(980, 60, 250, 60);**

**jreturn.setFont(font1);**

**jreturn.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**returnActionPerformed(e);**

**}**

**});**

**jBuyticket.setBounds(1400, 60, 150, 60);**

**jBuyticket.setFont(font1);**

**jBuyticket.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**buyActionPerformed(e);**

**}**

**});**

**jb.add(sPane);**

**p.add(serbtn);**

**p.add(serchText);**

**p.add(search);**

**p.add(jreturn);**

**p.add(jBuyticket);**

**p.add(jb);**

**con.add(p);**

**con.setVisible(true);**

**jf.setVisible(true);**

**jf.setExtendedState(JFrame.MAXIMIZED\_BOTH);**

**jf.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**}**

**protected void serachactionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**if("".equals(serchText.getText())) {**

**JOptionPane.showMessageDialog(null, "请输入影院名字");**

**}else {**

**String cName=serchText.getText();**

**List<Session> sesslist =new ArrayList<Session>();**

**List<Cinema> cinemalist = cinema.queryCinemaByCinemaName(cName);**

**if(cinemalist!=null) {**

**for (Cinema cinema : cinemalist) {**

**List<Session> sesslist1;**

**sesslist1=seser.querySessionBycIdandmId(cinema.getCid(), this.mo.getmId());**

**sesslist.addAll(sesslist1);**

**}**

**}**

**sessionlist=sesslist;**

**fillsessiontable(sessionlist);**

**}**

**}**

**private void fillsessiontable(List<Session> sessionlist2) {**

**// TODO Auto-generated method stub**

**DefaultTableModel dtm=(DefaultTableModel) tb.getModel();**

**dtm.setRowCount(0); // 设置成0行**

**Hall h = new Hall();**

**Cinema ci = new Cinema();**

**int sessionRow=sessionlist2.size();**

**for(int i=0;i<sessionRow;i++) {**

**h = hall.queryHallById(sessionlist2.get(i).gethId());**

**ci = cinema.queryCinemaById(sessionlist2.get(i).getcId());**

**Vector v=new Vector();**

**v.add(sessionlist2.get(i).getsId());**

**v.add(h.gethName());**

**v.add(ci.getCname());**

**v.add(ci.getCaddress());**

**v.add(mo.getmName());**

**v.add(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(sessionlist2.get(i).getBeginTime()));**

**v.add(sessionlist2.get(i).getPrice());**

**v.add(sessionlist2.get(i).getRemain());**

**dtm.addRow(v);**

**}**

**}**

**protected void returnActionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**new MovieUi(mo,ue);**

**jf.dispose();**

**}**

**protected void buyActionPerformed(ActionEvent e) {**

**// TODO Auto-generated method stub**

**System.out.println(ue.toString());**

**if (row == -1) {**

**JOptionPane.showMessageDialog(null, "请选择电影场次");**

**} else {**

**Hall h1 = hall.queryHallById(sessionlist.get(row).gethId());**

**new BuyTicketUi(ue, sessionlist.get(row), mo, h1);**

**jf.dispose();**

**}**

**}**

**protected void tablemousePressed(MouseEvent me) {**

**// TODO Auto-generated method stub**

**row = tb.getSelectedRow();**

**}**

**public static void main(String[] args) {**

**}**

**}**

**十.**

**private void rechargeActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**if("".equals(charge.getText())) {**

**LoginUi.winMessage("您还未输金额！");**

**}else {**

**try {**

**double reprice=Double.parseDouble(charge.getText());**

**user.setBalance(user.getBalance()+reprice);**

**if(uo.updateUser(user)) {**

**LoginUi.winMessage("充值成功");**

**ballb.setText(user.getBalance()+"元");**

**c.setVisible(false);**

**}else {**

**LoginUi.winMessage("充值失败");**

**}**

**}**

**catch (NumberFormatException e2) {**

**JOptionPane.showMessageDialog(null, "输入格式错误");**

**}**

**}**

**}**

**}**

**pack();**

**}// </editor-fold>**

**private void confirmUpActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**String oldpassw = new String(this.oldpass.getPassword());**

**String newpassw = new String(this.newpass.getPassword());**

**if("".equals(oldpassw) || "".equals(newpassw)) {**

**LoginUi.winMessage("信息不完整！");**

**} else if(!oldpassw.equals(user.getuPass())) {**

**LoginUi.winMessage("原密码错误，无法更改！");**

**}**

**else {**

**user.setuPass(newpassw);**

**if(uo.updateUser(user)) {**

**LoginUi.winMessage("密码修改成功！");**

**this.setVisible(false);**

**} else {**

**LoginUi.winMessage("密码修改失败！");**

**}**

**}**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**// Variables declaration - do not modify**

**private javax.swing.JButton confirmUp;**

**private javax.swing.JLabel newjl;**

**private javax.swing.JPasswordField newpass;**

**private javax.swing.JLabel oldjl;**

**private javax.swing.JPasswordField oldpass;**

**// End of variables declaration**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">**

**private void initComponents(User u) {**

**c=new Chargef();**

**p=new Pswd();**

**// mt=new MyTickets();**

**user=u;**

**ulb = new javax.swing.JLabel();**

**jLabel1 = new javax.swing.JLabel();**

**jb1 = new javax.swing.JButton();**

**jLabel2 = new javax.swing.JLabel();**

**ballb = new javax.swing.JLabel();**

**jb2 = new javax.swing.JButton();**

**jLabel3 = new javax.swing.JLabel();**

**setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);**

**ulb.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**ulb.setText(user.getuName());**

**jLabel1.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**jLabel1.setText("Hello!");**

**jb1.setText("充值");**

**jb1.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jb1MouseClicked(evt);**

**}**

**});**

**jLabel2.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**jLabel2.setText("账户余额：");**

**ballb.setFont(new java.awt.Font("微软雅黑", 0, 18)); // NOI18N**

**ballb.setText(user.getBalance()+"元");**

**jb2.setText("修改密码");**

**jb2.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jb2MouseClicked(evt);**

**}**

**});**

**jLabel3.setFont(new java.awt.Font("微软雅黑", 0, 14)); // NOI18N**

**jLabel3.setText("我的影票");**

**jLabel3.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jLabel3MouseClicked(evt);**

**}**

**});**

**javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());**

**getContentPane().setLayout(layout);**

**layout.setHorizontalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(layout.createSequentialGroup()**

**.addGap(53, 53, 53)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(layout.createSequentialGroup()**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(layout.createSequentialGroup()**

**.addComponent(jb1, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(30, 30, 30)**

**.addComponent(jb2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addGroup(layout.createSequentialGroup()**

**.addComponent(jLabel2)**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)**

**.addComponent(ballb)))**

**.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))**

**.addGroup(layout.createSequentialGroup()**

**.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)**

**.addComponent(ulb, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 90, Short.MAX\_VALUE)**

**.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 66, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addGap(23, 23, 23))))**

**);**

**layout.setVerticalGroup(**

**layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)**

**.addGroup(layout.createSequentialGroup()**

**.addGap(62, 62, 62)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)**

**.addComponent(ulb, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jLabel3))**

**.addGap(26, 26, 26)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)**

**.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(ballb, javax.swing.GroupLayout.PREFERRED\_SIZE, 35, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 36, Short.MAX\_VALUE)**

**.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)**

**.addComponent(jb1, javax.swing.GroupLayout.PREFERRED\_SIZE, 44, javax.swing.GroupLayout.PREFERRED\_SIZE)**

**.addComponent(jb2, javax.swing.GroupLayout.PREFERRED\_SIZE, 44, javax.swing.GroupLayout.PREFERRED\_SIZE))**

**.addGap(62, 62, 62))**

**);**

**pack();**

**}// </editor-fold>**

**private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {**

**MyTickets mtc=new MyTickets(user);**

**mtc.setVisible(true);**

**}**

**private void jb2MouseClicked(java.awt.event.MouseEvent evt) {**

**p.setVisible(true);**

**}**

**private void jb1MouseClicked(java.awt.event.MouseEvent evt) {**

**c.setVisible(true);**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(UserUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(UserUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(UserUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(UserUi.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**//</editor-fold>**

**//</editor-fold>**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**User u=new User();**

**new UserUi(u).setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify**

**private javax.swing.JLabel ballb;**

**private javax.swing.JLabel jLabel1;**

**private javax.swing.JLabel jLabel2;**

**private javax.swing.JLabel jLabel3;**

**private javax.swing.JButton jb1;**

**private javax.swing.JButton jb2;**

**private javax.swing.JLabel ulb;**

**private Font font = new Font("楷体", 0, 20);**

**// End of variables declaration**

**}**

**十一.**

**public class BaseDao {**

**public static final String DRIVER = "com.mysql.jdbc.Driver";**

**public static final String URL = "jdbc:mysql://localhost:3306/tickets";**

**static {**

**try {**

**Class.forName(DRIVER);**

**} catch (ClassNotFoundException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**}**

**// 获得连接**

**public Connection getConn() {**

**Connection conn = null;**

**try {**

**conn = DriverManager.getConnection(URL, "root", "1234");**

**} catch (SQLException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return conn;**

**}**

**// 关闭所有，防止内存泄露**

**public void releaseAll(ResultSet rs, PreparedStatement pstmt, Connection conn) {**

**try {**

**if (rs != null) {**

**rs.close();**

**}**

**if (pstmt != null) {**

**pstmt.close();**

**}**

**if (conn != null) {**

**conn.close();**

**}**

**} catch (SQLException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**}**

**// 增删改 封装**

**public boolean operUpdate(String sql, List<Object> params) {**

**Connection conn = null;**

**PreparedStatement pstmt = null;**

**int res = 0;**

**// 获得与数据库的连接对象**

**conn = getConn();**

**try {**

**pstmt = conn.prepareStatement(sql);**

**if (params != null) {**

**for (int i = 0; i < params.size(); i++) {**

**pstmt.setObject(i + 1, params.get(i));**

**}**

**}**

**// 增刪改的統一方法**

**res = pstmt.executeUpdate();**

**//返回的是sql在数据库中影响的行数**

**} catch (SQLException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} finally {**

**releaseAll(null, pstmt, conn);**

**}**

**return res > 0 ? true : false;**

**}**

**public <T> List<T> operQuery(String sql, List<Object> params, Class<T> cls) throws Exception {**

**Connection conn = null;**

**PreparedStatement pstmt = null;**

**ResultSet rs = null;**

**List<T> list = new ArrayList<T>();**

**conn = getConn();**

**try {**

**pstmt = conn.prepareStatement(sql);**

**if (params != null) {**

**for (int i = 0; i < params.size(); i++) {**

**pstmt.setObject(i + 1, params.get(i));**

**}**

**}**

**// 增刪改的統一方法**

**rs = pstmt.executeQuery();**

**ResultSetMetaData rsmd = rs.getMetaData();**

**while (rs.next()) {**

**T m = cls.newInstance();**

**for (int i = 0; i < rsmd.getColumnCount(); i++) {**

**String col\_name = rsmd.getColumnName(i + 1);**

**Object value = rs.getObject(col\_name);**

**Field field;**

**field = cls.getDeclaredField(col\_name);**

**field.setAccessible(true);**

**field.set(m, value);**

**}**

**list.add(m);**

**}**

**}**

**catch (SQLException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**} finally {**

**releaseAll(rs, pstmt, conn);**

**}**

**return list;**

**}**

**}**

**十二.**

**CinemaDao.java**

**package tickets.dboperation;**

**import java.util.\*;**

**import tickets.property.\*;**

**public class CinemaDao extends BaseDao{**

**public boolean addCinema(Cinema cinema) {**

**// TODO Auto-generated method stub**

**if (cinema == null) {**

**System.out.println("are you kiding me?");**

**return false;**

**}**

**String sql="insert into cinema (cname,caddress) values(?,?)";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(cinema.getCname());**

**params1.add(cinema.getCaddress());**

**return operUpdate(sql, params1);**

**}**

**public boolean delCinema(int cId) {**

**// TODO Auto-generated method stub**

**String sql = "delete from cinema where cId=" + cId;**

**return operUpdate(sql, null);**

**}**

**//根据id更新**

**public boolean updateCinema(Cinema cinema) {**

**// TODO Auto-generated method stub**

**String sql = "update cinema set cname=?,caddress=? where cId=?";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(cinema.getCname());**

**params1.add(cinema.getCaddress());**

**params1.add(cinema.getCid());**

**return operUpdate(sql, params1);**

**}**

**public List<Cinema> queryAllCinema() {**

**// TODO Auto-generated method stub**

**String sql = "select \* from cinema";**

**List<Cinema> list=null;**

**try {**

**list = operQuery(sql, null, Cinema.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Cinema queryCinemaById(int cId) {**

**// TODO Auto-generated method stub**

**String sql ="select \* from cinema where cId="+cId;**

**List<Cinema> list=null;**

**try {**

**list = operQuery(sql, null,Cinema.class);**

**if (list.size() == 0) {**

**return null;**

**} else {**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**public List<Cinema> queryCinemaByCinemaName(String cName) {**

**// TODO Auto-generated method stub**

**String name="'%";**

**for(int i = 0; i < cName.length(); i++) {**

**name += cName.charAt(i)+"%";**

**}**

**name+="'";**

**String sql="select \* from cinema where cName like"+name;**

**List<Cinema> list=null;**

**try {**

**list = operQuery(sql,null,Cinema.class);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public List<Cinema> queryCinemaByCinemaAddress(String cAddress) {**

**// TODO Auto-generated method stub**

**String address="'%";**

**for(int i = 0; i < cAddress.length(); i++) {**

**address += cAddress.charAt(i)+"%";**

**}**

**address+="'";**

**String sql="select \* from cinema where cAddress like"+address;**

**List<Cinema> list=null;**

**try {**

**list = operQuery(sql,null,Cinema.class);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public List<Cinema> queryCinemaByCinemaNameAndCinemaAddress(String cName, String cAddress) {**

**// TODO Auto-generated method stub**

**String address="'%";**

**for(int i = 0; i < cAddress.length(); i++) {**

**address += cAddress.charAt(i)+"%";**

**}**

**address+="'";**

**String name="'%";**

**for(int i = 0; i < cName.length(); i++) {**

**name += cName.charAt(i)+"%";**

**}**

**name+="'";**

**String sql="select \* from cinema where cName like"+name+"and cAddress like"+address;**

**List<Cinema> list=null;**

**try {**

**list = operQuery(sql,null,Cinema.class);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Cinema queryCinema(Cinema cinema) {**

**// TODO Auto-generated method stub**

**String sql = "select \* from cinema where cName=? and cAddress=? ";**

**List<Object> params = new ArrayList<Object>();**

**params.add(cinema.getCname());**

**params.add(cinema.getCaddress());**

**List<Cinema> list=null;**

**try {**

**list = operQuery(sql,params,Cinema.class);**

**if(list.size() == 0){**

**return null;**

**}else{**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**}**

**十三.**

**HallDao.java**

**package tickets.dboperation;**

**import java.util.\*;**

**import tickets.property.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class HallDao extends BaseDao{**

**public boolean addHall(Hall hall) {**

**// TODO Auto-generated method stub**

**if(hall==null) {**

**System.out.println("are you kiding me?");**

**return false;**

**}**

**String sql="insert into hall (hName,cId,capacity) values (?,?,?)";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(hall.gethName());**

**params1.add(hall.getcId());**

**params1.add(hall.getCapacity());**

**return operUpdate(sql, params1);**

**}**

**public boolean delHall(int hId) {**

**// TODO Auto-generated method stub**

**String sql = "delete from hall where hId=" + hId;**

**return operUpdate(sql, null);**

**}**

**public boolean updateHall(Hall hall) {**

**// TODO Auto-generated method stub**

**String sql = "update hall set hName=?,cId=?,capacity=? where hId=?";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(hall.gethName());**

**params1.add(hall.getcId());**

**params1.add(hall.getCapacity());**

**params1.add(hall.gethId());**

**return operUpdate(sql, params1);**

**}**

**public List<Hall> queryAllHall() {**

**// TODO Auto-generated method stub**

**String sql = "select \* from hall";**

**List<Hall> list=null;**

**try {**

**list = operQuery(sql, null, Hall.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Hall queryHallById(int hId) {**

**// TODO Auto-generated method stub**

**String sql ="select \* from hall where hId="+hId;**

**List<Object> params = new ArrayList<Object>();**

**List<Hall> list=null;**

**try {**

**list = operQuery(sql, null,Hall.class);**

**if (list.size() == 0) {**

**return null;**

**} else {**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**public List<Hall> queryAllHallcId(int cId) {**

**// TODO Auto-generated method stub**

**String sql="select \* from hall where cId="+cId;**

**List<Hall> list=null;**

**try {**

**list = operQuery(sql, null,Hall.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Hall queryHall(Hall hall) {**

**// TODO Auto-generated method stub**

**String sql="select \* from hall where hName=? and cId=? and capacity=?";**

**List<Object> params = new ArrayList<Object>();**

**params.add(hall.gethName());**

**params.add(hall.getcId());**

**params.add(hall.getCapacity());**

**List<Hall> list=null;**

**try {**

**list = operQuery(sql,params,Hall.class);**

**if(list.size() == 0){**

**return null;**

**}else{**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**}**

**十四．**

**MovieDao.java**

**package tickets.dboperation;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class MovieDao extends BaseDao{**

**public boolean addMovie(Movie movie) {**

**// TODO Auto-generated method stub**

**if(movie==null) {**

**System.out.println("are you kiding me?");**

**return false;**

**}**

**String sql="insert into movie (mName,mlanguage,mlocality,mPicaddress,mDetail,mDuration,mType,mGrade) values(?,?,?,?,?,?,?,?)";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(movie.getmName());**

**params1.add(movie.getMlanguage());**

**params1.add(movie.getMlocality());**

**params1.add(movie.getmPicaddress());**

**params1.add(movie.getmDetail());**

**params1.add(movie.getmDuration());**

**params1.add(movie.getmType());**

**params1.add(movie.getmGrade());**

**return operUpdate(sql, params1);**

**}**

**public boolean delMovie(int mid) {**

**// TODO Auto-generated method stub**

**String sql="delete from movie where mId="+mid;**

**return operUpdate(sql, null);**

**}**

**public boolean updateMovie(Movie movie) {**

**// TODO Auto-generated method stub**

**String sql="update movie set mName=?,mlanguage=?,mlocality=?,mPicaddress=?,mDetail=?,"**

**+ "mDuration=?,mType=?,mGrade=? where mId=?";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(movie.getmName());**

**params1.add(movie.getMlanguage());**

**params1.add(movie.getMlocality());**

**params1.add(movie.getmPicaddress());**

**params1.add(movie.getmDetail());**

**params1.add(movie.getmDuration());**

**params1.add(movie.getmType());**

**params1.add(movie.getmGrade());**

**params1.add(movie.getmId());**

**return operUpdate(sql, params1);**

**}**

**public List<Movie> queryAllMovie() {**

**// TODO Auto-generated method stub**

**String sql="select \* from movie";**

**List<Movie> list=null;**

**try {**

**list = operQuery(sql, null,Movie.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Movie queryMovieById(int mId) {**

**// TODO Auto-generated method stub**

**String sql ="select \*from movie where mId="+mId;**

**List<Movie> list = null;**

**try {**

**list = operQuery(sql,null,Movie.class);**

**if(list.size() == 0){**

**return null;**

**}else{**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**public List<Movie> OrderMovieTen() {**

**// TODO Auto-generated method stub**

**String sql = "select \* from movie order by mGrade desc limit 10 ";**

**List <Movie> list=null;**

**try {**

**list = operQuery(sql, null,Movie.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public List<Movie> queryMovieByType(String mType) {**

**// TODO Auto-generated method stub**

**String type="'";**

**for(int i=0;i<mType.length();i++) {**

**type+=mType.charAt(i);**

**}**

**type+="'";**

**String sql ="select \* from movie where mType="+type;**

**List <Movie> list=null;**

**try {**

**list = operQuery(sql, null,Movie.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public List<Movie> queryMovieBymovieName(String mName) {**

**// TODO Auto-generated method stub**

**String name="'%";**

**for(int i = 0; i < mName.length(); i++) {**

**name += mName.charAt(i)+"%";**

**}**

**name += "'";**

**String sql="select \* from movie where mName like"+name;**

**List<Movie> list=null;**

**try {**

**list = operQuery(sql,null,Movie.class);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Movie queryMovie(Movie movie) {**

**// TODO Auto-generated method stub**

**String sql = "select \* from movie where mName=? and mlanguage=? and mlocality=? and mPicaddress=? and mDetail=? "**

**+ "and mDuration=? and mType=? and mGrade=?";**

**List<Object> params = new ArrayList<Object>();**

**params.add(movie.getmName());**

**params.add(movie.getMlanguage());**

**params.add(movie.getMlocality());**

**params.add(movie.getmPicaddress());**

**params.add(movie.getmDetail());**

**params.add(movie.getmDuration());**

**params.add(movie.getmType());**

**params.add(movie.getmGrade());**

**List<Movie> list = null;**

**try {**

**list = operQuery(sql,params,Movie.class);**

**if(list.size() == 0){**

**return null;**

**}else{**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**public List<Movie> queryMovieByTypeandName(String mType, String mName) {**

**// TODO Auto-generated method stub**

**String name="'%";**

**String type="'";**

**for(int i=0;i<mType.length();i++) {**

**type+=mType.charAt(i);**

**}**

**for(int i = 0; i < mName.length(); i++) {**

**name += mName.charAt(i)+"%";**

**}**

**name += "'";**

**type +="'";**

**String sql="select \* from movie where mType ="+type+"and mName like"+name;**

**List<Movie> list=null;**

**try {**

**list = operQuery(sql,null,Movie.class);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Movie queryMovieByName(String mName) {**

**// TODO Auto-generated method stub**

**String name="'";**

**for(int i=0;i<mName.length();i++) {**

**name+=mName.charAt(i);**

**}**

**name+="'";**

**String sql ="select \* from movie where mName="+name;**

**List <Movie> list=null;**

**try {**

**list = operQuery(sql,null,Movie.class);**

**if(list.size() == 0){**

**return null;**

**}else{**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**}**

**十五．**

**SessionDao.java**

**package tickets.dboperation;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class SessionDao extends BaseDao{**

**public boolean addSession(Session session) {**

**// TODO Auto-generated method stub**

**if(session==null) {**

**System.out.println("cannot be blank");**

**}**

**String sql="insert into session(hId,cId,mId,beginTime,price,remain) values (?,?,?,?,?,?)";**

**List<Object> params1 = new ArrayList<Object>();**

**params1.add(session.gethId());**

**params1.add(session.getcId());**

**params1.add(session.getmId());**

**params1.add(session.getBeginTime());**

**params1.add(session.getPrice());**

**params1.add(session.getRemain());**

**return operUpdate(sql, params1);**

**}**

**public boolean delSession(int sId) {**

**// TODO Auto-generated method stub**

**String sql="delete from session where sId="+sId;**

**return operUpdate(sql, null);**

**}**

**}**

**public List<Session> queryAllSession() {**

**// TODO Auto-generated method stub**

**String sql="select \* from session";**

**List<Session> list=null;**

**try {**

**list = operQuery(sql, null,Session.class);**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return list;**

**}**

**public Session querySessionById(int sId) {**

**// TODO Auto-generated method stub**

**String sql="select \*from session where sId="+sId;**

**List<Session> list=null;**

**try {**

**list = operQuery(sql,null,Session.class);**

**if(list.size() == 0){**

**return null;**

**}else{**

**return list.get(0);**

**}**

**} catch (Exception e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**return null;**

**}**

**@Override**

**public String toString() {**

**return "Ticket [tId=" + tId + ", uId=" + uId + ", sId=" + sId + ", seat=" + seat + "]";**

**}**

**}**

**二十三.**

**User.java**

**package tickets.property;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class User {**

**private int uId;**

**private String uName;**

**private String uPass;**

**private double balance;**

**private int uType;**

**public User() {**

**super();**

**// TODO Auto-generated constructor stub**

**}**

**public User(int uId, String uName, String uPass, double balance, int uType) {**

**super();**

**this.uId = uId;**

**this.uName = uName;**

**this.uPass = uPass;**

**this.balance = balance;**

**this.uType = uType;**

**}**

**public User(String uName, String uPass, int uType) {**

**super();**

**this.uName = uName;**

**this.uPass = uPass;**

**this.uType = uType;**

**}**

**public User(String uName, String uPass, double balance) {**

**super();**

**this.uName = uName;**

**this.uPass = uPass;**

**this.balance = balance;**

**}**

**public User(String uName, String uPass, double balance, int uType) {**

**super();**

**this.uName = uName;**

**this.uPass = uPass;**

**this.balance = balance;**

**this.uType = uType;**

**}**

**public int getUid() {**

**return uId;**

**}**

**public void setUid(int uid) {**

**this.uId = uid;**

**}**

**public String getuName() {**

**return uName;**

**}**

**public void setuName(String uName) {**

**this.uName = uName;**

**}**

**public String getuPass() {**

**return uPass;**

**}**

**public void setuPass(String uPass) {**

**this.uPass = uPass;**

**}**

**public double getBalance() {**

**return balance;**

**}**

**public void setBalance(double balance) {**

**this.balance = balance;**

**}**

**public int getuType() {**

**return uType;**

**}**

**public void setuType(int uType) {**

**this.uType = uType;**

**}**

**@Override**

**public String toString() {**

**return "User [uid=" + uId + ", uName=" + uName + ", uPass=" + uPass + ", balance=" + balance + ", uType="**

**+ uType + "]";**

**}**

**}**

**二十四.**

**CinemaService.java**

**package tickets.service;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**public class CinemaService {**

**private CinemaDao cinemaDao;**

**public CinemaService() {**

**// TODO Auto-generated constructor stub**

**cinemaDao=new CinemaDao();**

**}**

**public int addCinema(Cinema cinema) {**

**// TODO Auto-generated method stub**

**Cinema c=cinemaDao.queryCinema(cinema);**

**if(c==null) {**

**boolean res\_add =cinemaDao.addCinema(cinema);**

**if (res\_add) {**

**return 1;//添加成功**

**}else {**

**return 2;//添加失败**

**}**

**}**

**return 3;//添加失败-电影院已存在**

**}**

**public boolean delCinema(int cid) {**

**// TODO Auto-generated method stub**

**return cinemaDao.delCinema(cid);**

**}**

**public boolean updateCinema(Cinema cinema) {**

**// TODO Auto-generated method stub**

**return cinemaDao.updateCinema(cinema);**

**}**

**public List<Cinema> queryAllCinema() {**

**// TODO Auto-generated method stub**

**return cinemaDao.queryAllCinema();**

**}**

**public Cinema queryCinemaById(int cId) {**

**// TODO Auto-generated method stub**

**return cinemaDao.queryCinemaById(cId);**

**}**

**public List<Cinema> queryCinemaByCinemaName(String cName) {**

**// TODO Auto-generated method stub**

**return cinemaDao.queryCinemaByCinemaName(cName);**

**}**

**public List<Cinema> queryCinemaByCinemaAddress(String cAddress) {**

**// TODO Auto-generated method stub**

**return cinemaDao.queryCinemaByCinemaAddress(cAddress);**

**}**

**public List<Cinema> queryCinemaByCinemaNameAndCinemaAddress(String cName, String cAddress) {**

**// TODO Auto-generated method stub**

**return cinemaDao.queryCinemaByCinemaNameAndCinemaAddress(cName, cAddress);**

**}**

**}**

**二十五.**

**HallService.java**

**package tickets.service;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class HallService {**

**private HallDao hallDao=new HallDao();**

**public int addHall(Hall hall) {**

**// TODO Auto-generated method stub**

**Hall h=hallDao.queryHall(hall);**

**if(h==null) {**

**boolean res\_add=hallDao.addHall(hall);**

**if (res\_add) {**

**return 1;//添加成功**

**}else {**

**return 2;//添加失败**

**}**

**}**

**return 3;//添加失败**

**}**

**public boolean delHall(int hId) {**

**// TODO Auto-generated method stub**

**return hallDao.delHall(hId);**

**}**

**public boolean updateHall(Hall hall) {**

**// TODO Auto-generated method stub**

**return hallDao.updateHall(hall);**

**}**

**public List<Hall> queryAllHall() {**

**// TODO Auto-generated method stub**

**return hallDao.queryAllHall();**

**}**

**public Hall queryHallById(int hId) {**

**// TODO Auto-generated method stub**

**return hallDao.queryHallById(hId);**

**}**

**public List<Hall> queryAllHallcId(int cId) {**

**// TODO Auto-generated method stub**

**return hallDao.queryAllHallcId(cId);**

**}**

**}**

**二十六.**

**MovieService.java**

**package tickets.service;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class MovieService {**

**private MovieDao movieDao;**

**public MovieService() {**

**movieDao = new MovieDao();**

**}**

**public int addMovie(Movie movie) {**

**// TODO Auto-generated method stub**

**Movie m=movieDao.queryMovie(movie);**

**if(m==null) {**

**boolean res\_add =movieDao.addMovie(movie);**

**if (res\_add) {**

**return 1;//添加成功**

**}else {**

**return 2;//添加失败**

**}**

**}**

**return 3;//添加失败-电影已存在**

**}**

**public boolean delMovie(int mid) {**

**// TODO Auto-generated method stub**

**return movieDao.delMovie(mid);**

**}**

**public boolean updateMovie(Movie movie) {**

**// TODO Auto-generated method stub**

**return movieDao.updateMovie(movie);**

**}**

**public List<Movie> queryAllMovie() {**

**// TODO Auto-generated method stub**

**return movieDao.queryAllMovie();**

**}**

**public List<Movie> OrderMovieTen() {**

**// TODO Auto-generated method stub**

**return movieDao.OrderMovieTen();**

**}**

**public Movie queryMovieById(int mId) {**

**// TODO Auto-generated method stub**

**return movieDao.queryMovieById(mId);**

**}**

**public List<Movie> queryMovieByType(String mType) {**

**// TODO Auto-generated method stub**

**return movieDao.queryMovieByType(mType);**

**}**

**public List<Movie> queryMovieBymovieName(String mName) {**

**// TODO Auto-generated method stub**

**return movieDao.queryMovieBymovieName(mName);**

**}**

**public Movie queryMovie(Movie movie) {**

**// TODO Auto-generated method stub**

**return null;**

**}**

**public List<Movie> queryMovieByTypeandName(String mType, String mName) {**

**// TODO Auto-generated method stub**

**return movieDao.queryMovieByTypeandName(mType, mName);**

**}**

**public Movie queryMovieByName(String mName) {**

**// TODO Auto-generated method stub**

**return movieDao.queryMovieByName(mName);**

**}**

**}**

**二十七.**

**SessionService.java**

**package tickets.service;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class SessionService {**

**private SessionDao sessionDao;**

**public SessionService() {**

**sessionDao=new SessionDao() ;**

**}**

**public int addSession(Session session) {**

**// TODO Auto-generated method stub**

**Session s=sessionDao.querySession(session);**

**if(s==null) {**

**boolean res\_add=sessionDao.addSession(session);**

**if(res\_add) {**

**return 1;//添加成功**

**}else {**

**return 2;//添加失败**

**}**

**}**

**return 3;//添加失败**

**}**

**public boolean delSession(int sId) {**

**// TODO Auto-generated method stub**

**return sessionDao.delSession(sId);**

**}**

**public boolean updateSession(Session session) {**

**// TODO Auto-generated method stub**

**return sessionDao.updateSession(session);**

**}**

**public List<Session> queryAllSession() {**

**// TODO Auto-generated method stub**

**return sessionDao.queryAllSession();**

**}**

**public List<Session> querySessionBymId(int mId) {**

**// TODO Auto-generated method stub**

**return sessionDao.querySessionByMid(mId);**

**}**

**public Session querySessionById(int sId) {**

**// TODO Auto-generated method stub**

**return sessionDao.querySessionById(sId);**

**}**

**public List<Session> querySessionByHid(int hId) {**

**// TODO Auto-generated method stub**

**return sessionDao.querySessionByHid(hId);**

**}**

**public List<Session> querySessionBycIdandmId(int cId, int mId) {**

**// TODO Auto-generated method stub**

**return sessionDao.querySessionBycIdandmId(cId, mId);**

**}**

**public List<Session> querySessionBycId(int cId) {**

**// TODO Auto-generated method stub**

**return sessionDao.querySessionByCid(cId);**

**}**

**}**

**二十八.**

**TicketService.java**

**package tickets.service;**

**import java.util.\*;**

**import tickets.property.\*;**

**import tickets.dboperation.\*;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class TicketService {**

**private TicketDao ticketDao;**

**public TicketService(){**

**ticketDao=new TicketDao();**

**}**

**public boolean addTicket(Ticket ticket) {**

**// TODO Auto-generated method stub**

**return ticketDao.addTicket(ticket);**

**}**

**public boolean delTicket(int tId) {**

**// TODO Auto-generated method stub**

**return ticketDao.delTicket(tId);**

**}**

**public List<Ticket> queryAllTicket() {**

**// TODO Auto-generated method stub**

**return ticketDao.queryAllTicket();**

**}**

**public List<Ticket> queryAllTicketsId(int sId) {**

**// TODO Auto-generated method stub**

**return ticketDao.queryAllTicketsId(sId);**

**}**

**public List<Ticket> queryAllTicketuId(int uId) {**

**// TODO Auto-generated method stub**

**return ticketDao.queryAllTicketuId(uId);**

**}**

**}**

**二十九.**

**UserService.java**

**package tickets.service;**

**import tickets.property.User;**

**import tickets.dboperation.UserDao;**

**/\*\***

**\***

**\* @author 94540**

**\*/**

**public class UserService {**

**private UserDao mUserDao;**

**public UserService() {**

**mUserDao=new UserDao();**

**}**

**public User login(User user) {**

**// TODO Auto-generated method stub**

**return mUserDao.queryUser(user);**

**}**

**public int register(User user) {**

**// TODO Auto-generated method stub**

**User u = mUserDao.queryUserByName(user.getuName());**

**if (u == null) {**

**boolean res\_add = mUserDao.addUser(user);**

**if (res\_add) {**

**return 1;//注册成功**

**} else {**

**return 2;//注册失败**

**}**

**}**

**return 3;//注册失败-用户名已存在**

**}**

**public User queryUserByName(String uName) {**

**// TODO Auto-generated method stub**

**return mUserDao.queryUserByName(uName);**

**}**

**public boolean updateUserPass(User user,String pass) {**

**return mUserDao.updateUserPass(user,pass);**

**}**

**public User queryUser(User user) {**

**return mUserDao.queryUser(user);**

**}**

**public boolean updateUser(User user) {**

**return mUserDao.updateUser(user);**

**}**

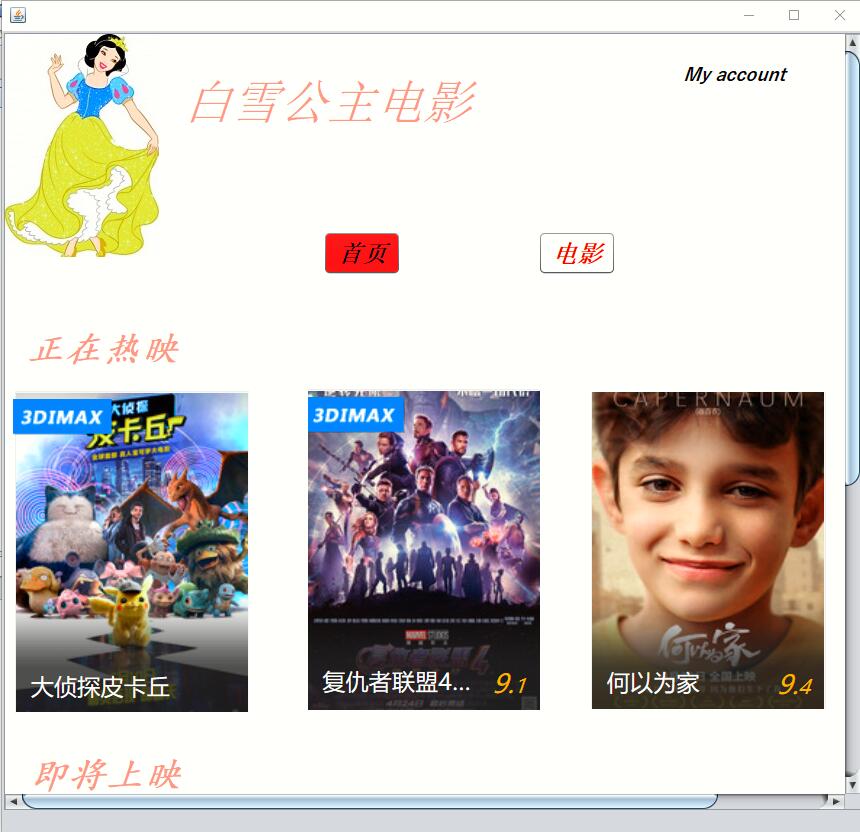
**public User queryUserByid(int uid) {**

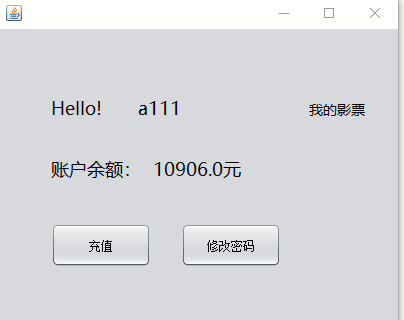
**return mUserDao.queryUserByid(uid);**

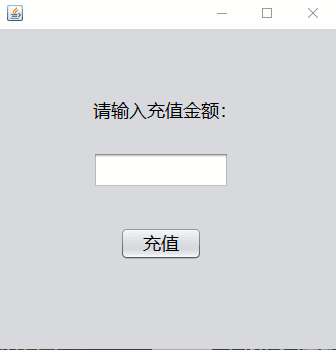
**}**

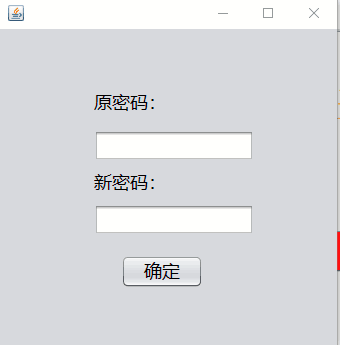


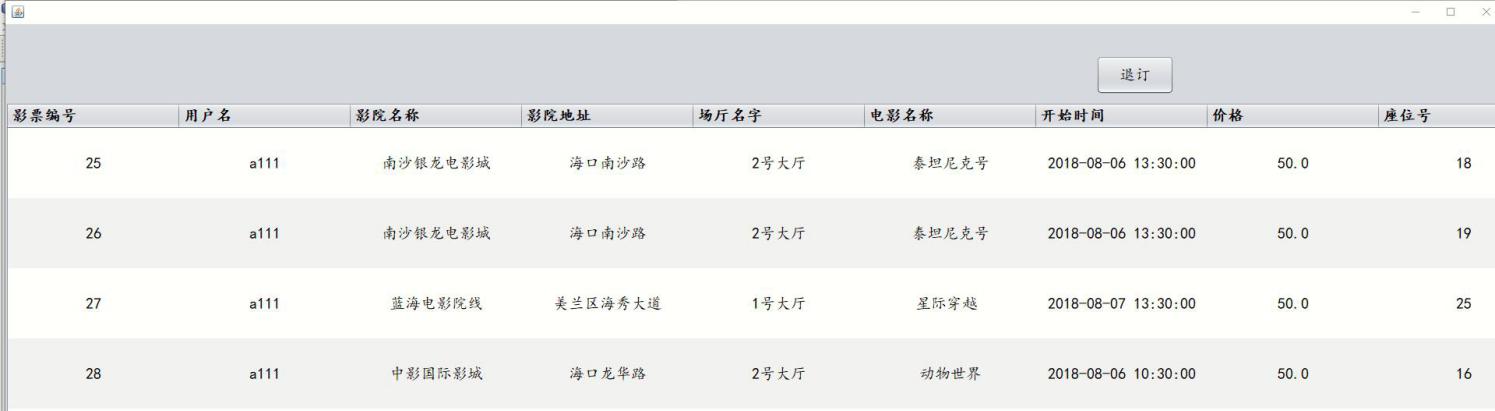


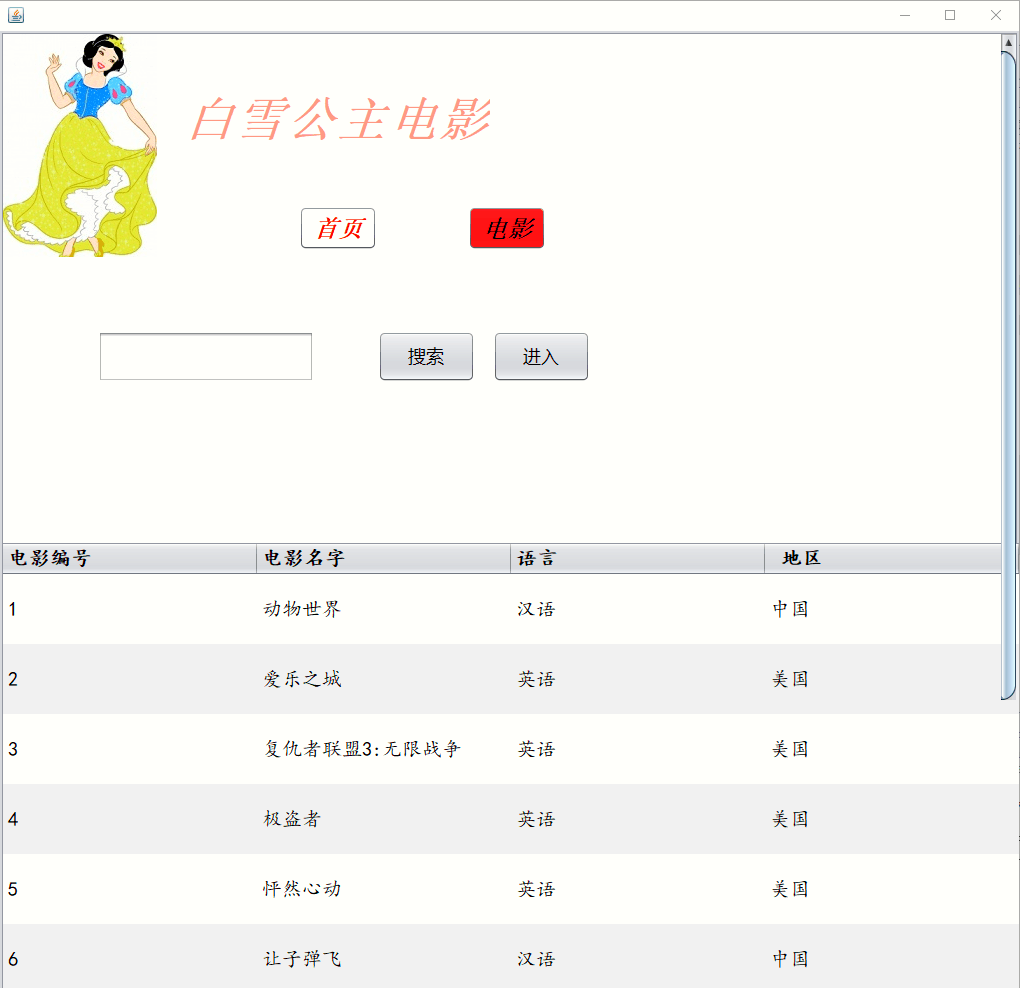




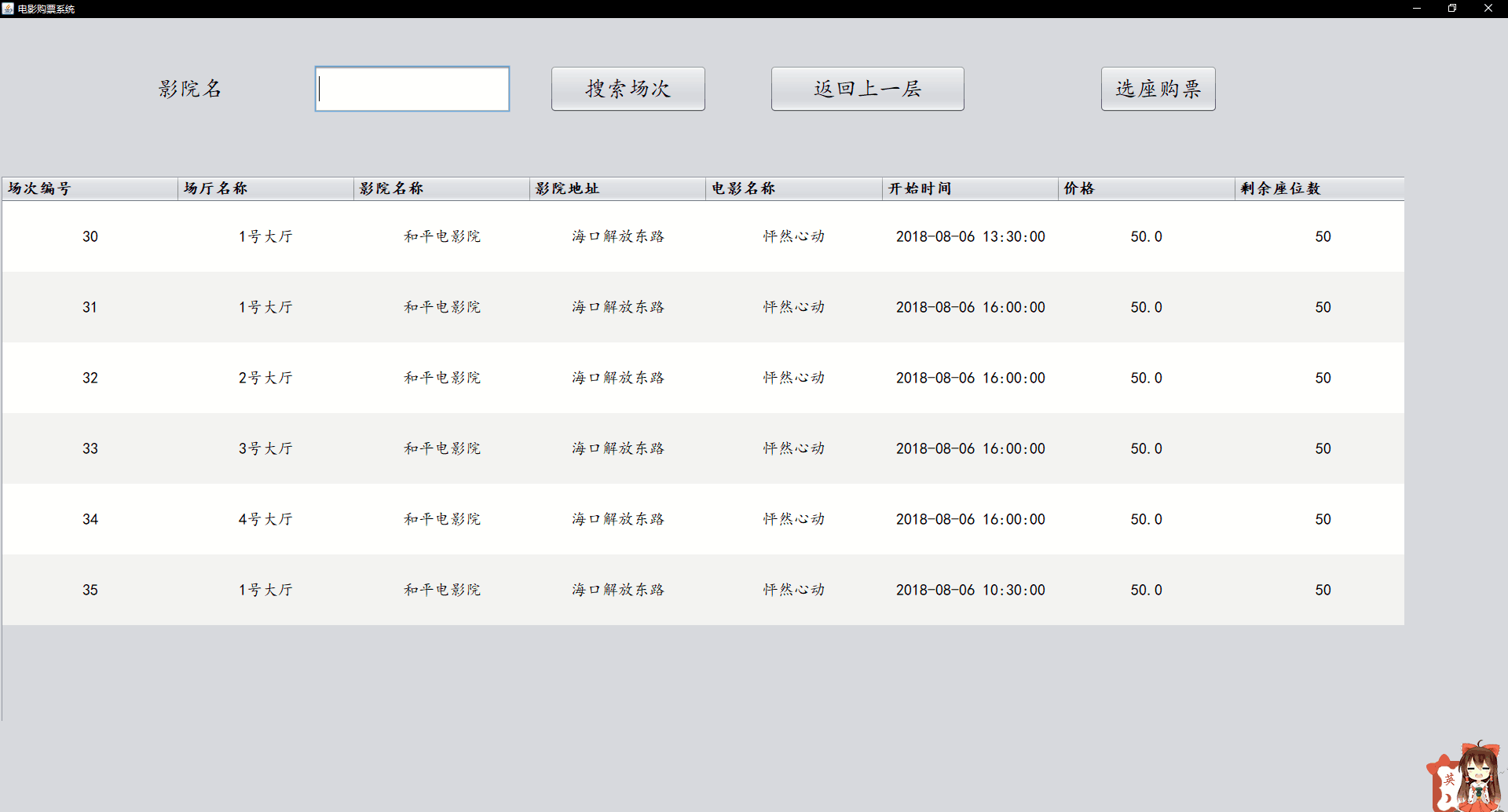


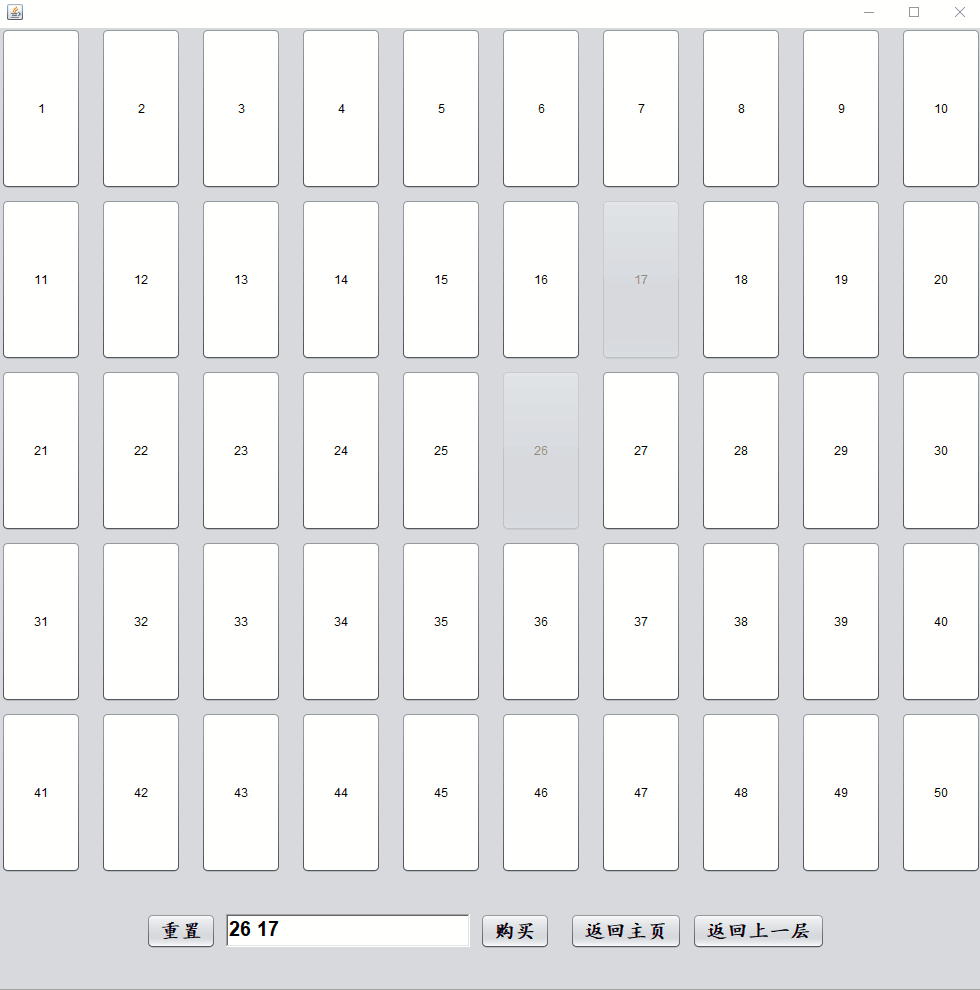


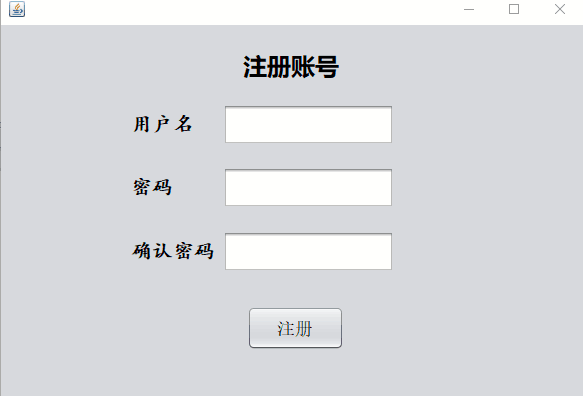




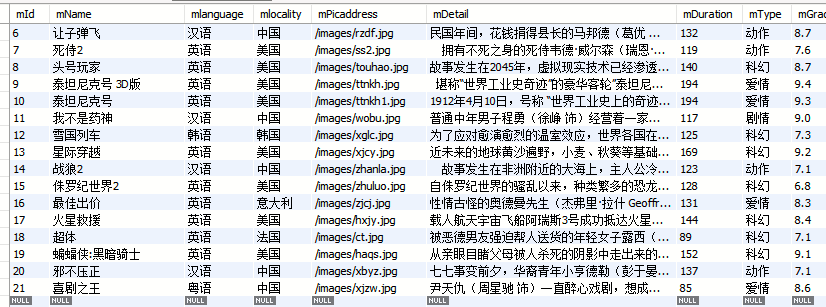


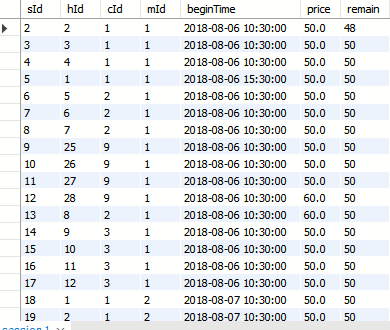


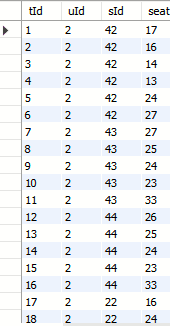


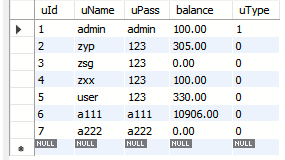


**Database**









**相关技术简介:**

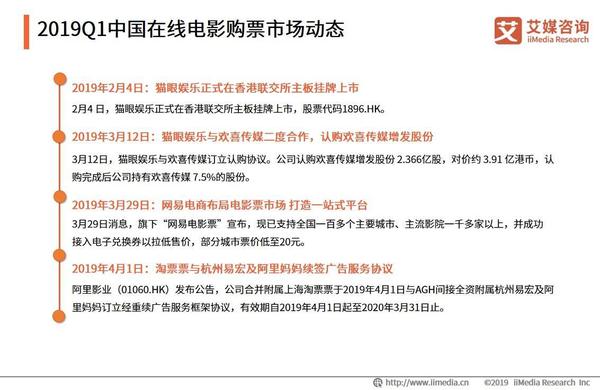
1.Java-NetBeans:NetBeans is the home of the open source developer and customer community founded by Sun (acquired by Oracle in 2009) in 2000 to build a world-class Java IDE. NetBeans can currently be developed on Solaris, Windows, Linux, and Macintosh OS X platforms and used within the scope of the SPL (Sun public license. NetBeans includes an open source development environment and application platform, NetBeans IDE that enables developers to use the Java platform to quickly create Web, enterprise, desktop, and mobile applications. NetBeans IDE already supports development languages such as PHP, Ruby, JavaScript, Groovy, Grails, and C +.

2.SQL-mysql:MySQL is a relational database management system developed by MySQL AB of Sweden and currently owned by Oracle. MySQL is one of the most popular relational database management systems. In WEB applications, MySQL is the best RDBMS (Relational Database Management System, relational database management system) application software. MySQL is a relational database management system that stores data in different tables instead of all data in a large warehouse, which increases speed and flexibility. The SQL language used by MySQL is the most commonly used standardized language for accessing databases. MySQL software adopts dual authorization policy, which is divided into community version and commercial version. Because of its small size, high speed and low total cost of ownership, especially open source, the development of small and medium-sized websites choose MySQL as the website database. Because of the excellent performance of its community version, it can be combined with PHP and Apache to form a good development environment.

**3.市场分析**

**3.1市场现状**

According to data from iiMedia Research (Ai Media Consulting, in the first quarter of 2019, the box office of films in mainland China reached 18.61 billion yuan, and the box office of films in China during the Spring Festival totaled 5.84 billion yuan, an increase of 2.04 percent over the same period last year. After the rapid popularity of mobile payments, Online has become the main channel to buy tickets, offline channels account for only 14.3% of the film ticket market share. According to data from iiMedia Research (Ai Media Consulting, Cat's Eye and Taobao tickets account for 42.6% and 31.5% of the Chinese film ticket market respectively, with the largest number of monthly active users and 59.54% of female users. There were more users aged 24 and below, accounting for 53.77%. The two platforms are backed by Alibaba and Tencent, each with development characteristics, and the competition between the two platforms will exist for a long time in the future. After the explosion and growth of the film market, the film ticket consumption of the audience began to return to rationality.



**3.2市场前景**

The future development trend of China's online film ticket market. Online has become the main channel for buying tickets for movies, and Shuangxiong will develop in the competition. Analysts believe that since the rapid popularity of mobile payment, online has become the main channel for viewers to buy tickets for movies. In recent years, with the outbreak of growth in the film market, Online movie ticket shopping platforms are also becoming more competitive. At present, there is a situation in the market that Taobao and Cat's Eye compete for hegemony. The two platforms rely on Alibaba and Tencent respectively, and each has its own characteristics of development. In the future, the competition between the two will still exist for a long time, and the two platforms will develop in the competition.

**3.3 SWOT分析**

**优势**：Fast, cheap, convenient, you can quickly buy movie tickets anytime, anywhere, after all, through the computer or mobile phone, do not have to go to the cinema to queue up.

**劣势**：After buying, more people in line to change tickets to the cinema have to go early, and it is possible that the cinema is far away from you, and sometimes the location is not very good.

**机遇与威胁**：

The rapid development of online film ticket market also brings high cost burden to enterprises, the head platform is in a state of loss, and each platform will excavate more sustainable profit models in the future. Ai media consulting analysts believe that in the future, the business model of the online film ticket purchase industry will become more diversified, such as commission extraction, advertising revenue, big data and so on will become the main source of income of the platform.