****

**实验报告**

**（程序设计课程实践类）**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称：** | **软件开发实践** | | |
| **实验时间:** | **2022.2.26** | **~** | **2022.3.17** |
| **实验地点：** | **图书馆-中央2** | | |
| **学 院：** | **软件学院** | | |
| **姓 名：** | **宗艺萱** | | |
| **学 号：** | **211230218** | | |
| **专 业：** | **软件工程** | | |
| **班 级：** | **软件工程2102班** | | |
| **指导教师：** | **孙淑杰** | | |
| **评阅时间：** |  | | |
| **成 绩：** |  | | |

**沈阳工业大学软件学院**

**2022年 3月 17日**

**实验 酒店管理系统**

1. **实验目的**

(1)了解JAVA语言的swing组件的使用。

(2)拥有良好的程序逻辑并能通过基本语句具现化

(3)了解MYSQL的基本语句，如增、删、改、查。

(4)能够利用jdbc连接程序与数据库。

(5)能够利用navicat简化数据库的设计。

(6)能够通过查阅资料正确设置各种驱动、参数。

(7)小组能够完成程序的排查报错和修复。

(8)能够封装出可移植、鲁棒性强的api

1. **实验内容**

使用java编写酒店管理系统，使用mysql存储相关数据，力求软件具有高可用性

1. **分析与设计**

1.开发环境：openjdk17, IntelliJ IDEA IDE,jar打包于jdk1.8。

2. 文件架构：

软件包名：cn.edu.sut.hotel\_management

主类：cn.edu.sut.hotel\_management.Main

模块说明：model --用于存储文件模型

sql --用于存储封装好的数据库操作

ui --用于存储用于存储界面代码

ui.component -- 用于存储ui组件

util --存储比较底层的接口封装

3.文件结构：

图片包含 文本

描述已自动生成 图片包含 日程表

描述已自动生成

1. **代码设计**

|  |
| --- |
| Main.java |
| package cn.edu.sut.hotel\_management;  import cn.edu.sut.hotel\_management.ui.Login;  public class Main {  public static void main(String[] args) {  Login l = new Login();  l.setVisible(true);  } } |

|  |
| --- |
| admin.java |
| package cn.edu.sut.hotel\_management.model;  import java.util.Objects;  public class admin {  private int id;  private String name;  private String account;  private String password;  private String user\_name;  private String img\_url;  private String id\_card;  private String tel;   public admin(String account, String password) {  this.account = account;  this.password = password;  }   public admin() {  }   public admin(int id, String name, String account, String password, String user\_name, String img\_url, String id\_card, String tel) {  this.id = id;  this.name = name;  this.account = account;  this.password = password;  this.user\_name = user\_name;  this.img\_url = img\_url;  this.id\_card = id\_card;  this.tel = tel;  }   @Override  public String toString() {  return "admin{" +  "id=" + id +  ", name='" + name + '\'' +  ", account='" + account + '\'' +  ", password='" + password + '\'' +  ", user\_name='" + user\_name + '\'' +  ", img\_url='" + img\_url + '\'' +  ", id\_card='" + id\_card + '\'' +  ", tel='" + tel + '\'' +  '}';  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (o == null || getClass() != o.getClass()) return false;  admin admin = (admin) o;  return account.equals(admin.account) && password.equals(admin.password);  }   @Override  public int hashCode() {  return Objects.hash(account, password);  } } |

|  |
| --- |
| curuser.java |
| package cn.edu.sut.hotel\_management.model;  import java.util.Objects;  public class curuser {  public static long id;  private static String img\_url;  private static String name;  private static String id\_card;  private static String sex;  private static String tel;  private static String status;  private static String account;  private static String user\_name;  private static String password;   public curuser(long id, String img\_url, String name, String id\_card, String sex, String tel, String status, String account, String user\_name, String password) {  curuser.id = id;  curuser.img\_url = img\_url;  curuser.name = name;  curuser.id\_card = id\_card;  curuser.sex = sex;  curuser.tel = tel;  curuser.status = status;  curuser.account = account;  curuser.user\_name = user\_name;  curuser.password = password;  }   public static long getId() {  return id;  }   public static void setId(long id) {  curuser.id = id;  }   public static String getImg\_url() {  return img\_url;  }   public static void setImg\_url(String img\_url) {  curuser.img\_url = img\_url;  }   public static String getName() {  return name;  }   public static void setName(String name) {  curuser.name = name;  }   public static String getId\_card() {  return id\_card;  }   public static void setId\_card(String id\_card) {  curuser.id\_card = id\_card;  }   public static String getSex() {  return sex;  }   public static void setSex(String sex) {  curuser.sex = sex;  }   public static String getTel() {  return tel;  }   public static void setTel(String tel) {  curuser.tel = tel;  }   public static String getStatus() {  return status;  }   public static void setStatus(String status) {  curuser.status = status;  }   public static String getAccount() {  return account;  }   public static void setAccount(String account) {  curuser.account = account;  }   public static String getUser\_name() {  return user\_name;  }   public static void setUser\_name(String user\_name) {  curuser.user\_name = user\_name;  }   public static String getPassword() {  return password;  }   public static void setPassword(String password) {  curuser.password = password;  }   public static void logout() {  curuser.id = 0;  curuser.img\_url = null;  curuser.name = null;  curuser.id\_card = null;  curuser.sex = null;  curuser.tel = null;  curuser.status = null;  curuser.account = null;  curuser.user\_name = null;  curuser.password = null;  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (o == null || getClass() != o.getClass()) return false;  curuser curuser = (curuser) o;  return account.equals(account) && password.equals(password);  }   @Override  public int hashCode() {  return Objects.hash(account, password);  }   @Override  public String toString() {  return "user{" +  "id=" + id +  ", img\_url='" + img\_url + '\'' +  ", name='" + name + '\'' +  ", id\_card='" + id\_card + '\'' +  ", sex='" + sex + '\'' +  ", tel='" + tel + '\'' +  ", status='" + status + '\'' +  ", account='" + account + '\'' +  ", user\_name='" + user\_name + '\'' +  ", password='" + password + '\'' +  '}';  } } |

|  |
| --- |
| order.java |
| package cn.edu.sut.hotel\_management.model;  import java.util.Objects;  public class order {  private long id;  private String begin;  private int stay;  private float prepay;  private float pay;  private float rental;  private String id\_card;  private String name;  private String phone;   public order(long id) {  this.id = id;  }   public order(long id, String begin, int stay, float prepay, float pay, float rental) {  this.id = id;  this.begin = begin;  this.stay = stay;  this.prepay = prepay;  this.pay = pay;  this.rental = rental;  }   public order(long id, String begin, int stay, float prepay, float pay, float rental, String id\_card, String name, String phone) {  this.id = id;  this.begin = begin;  this.stay = stay;  this.prepay = prepay;  this.pay = pay;  this.rental = rental;  this.id\_card = id\_card;  this.name = name;  this.phone = phone;  }   public order() {  }   public long getId() {  return id;  }   public void setId(long id) {  this.id = id;  }   public String getBegin() {  return begin;  }   public void setBegin(String begin) {  this.begin = begin;  }   public int getStay() {  return stay;  }   public void setStay(int stay) {  this.stay = stay;  }   public float getPrepay() {  return prepay;  }   public void setPrepay(float prepay) {  this.prepay = prepay;  }   public float getPay() {  return pay;  }   public void setPay(float pay) {  this.pay = pay;  }   public float getRental() {  return rental;  }   public void setRental(float rental) {  this.rental = rental;  }   public String getId\_card() {  return id\_card;  }   public void setId\_card(String id\_card) {  this.id\_card = id\_card;  }   public String getName() {  return name;  }   public void setName(String name) {  this.name = name;  }   public String getPhone() {  return phone;  }   public void setPhone(String phone) {  this.phone = phone;  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (o == null || getClass() != o.getClass()) return false;  order order = (order) o;  return id == order.id;  }   @Override  public int hashCode() {  return Objects.hash(id);  }   @Override  public String toString() {  return "order{" +  "id=" + id +  ", begin='" + begin + '\'' +  ", stay='" + stay + '\'' +  ", prepay='" + prepay + '\'' +  ", pay='" + pay + '\'' +  ", rental='" + rental + '\'' +  '}';  }   } |

|  |
| --- |
| price.java |
| package cn.edu.sut.hotel\_management.model;  public class price {  private final int id;  private final float money;  private final float prepay;   public price(int id, float money, float prepay) {  this.id = id;  this.money = money;  this.prepay = prepay;  }  } |

|  |
| --- |
| relation.java |
| package cn.edu.sut.hotel\_management.model;  import java.util.Objects;  public class relation {  private final int id;  private final long user\_id;  private final int room\_id;  private final long order\_id;  private final int admin\_id;  private final String status;   public relation(int id, long user\_id, int room\_id, long order\_id, int admin\_id, String status) {  this.id = id;  this.user\_id = user\_id;  this.room\_id = room\_id;  this.order\_id = order\_id;  this.admin\_id = admin\_id;  this.status = status;  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (o == null || getClass() != o.getClass()) return false;  relation relation = (relation) o;  return id == relation.id && user\_id == relation.user\_id && room\_id == relation.room\_id && order\_id == relation.order\_id && admin\_id == relation.admin\_id;  }   @Override  public String toString() {  return "relation{" +  "id=" + id +  ", user\_id=" + user\_id +  ", room\_id=" + room\_id +  ", order\_id=" + order\_id +  ", admin\_id=" + admin\_id +  ", status='" + status + '\'' +  '}';  }   @Override  public int hashCode() {  return Objects.hash(id, user\_id, room\_id, order\_id, admin\_id);  } } |

|  |
| --- |
| room.java |
| package cn.edu.sut.hotel\_management.model;  import java.util.Objects;  public class room {  private int id;  private String num;   private String type;   private float money;  private float prepay;   private String status;   public room(int id, String num, String type, float prepay, float money, String status) {  this.id = id;  this.num = num;  this.type = type;  this.money = money;  this.prepay = prepay;  this.status = status;  }   public int getId() {  return id;  }   public void setId(int id) {  this.id = id;  }   public String getNum() {  return num;  }   public void setNum(String num) {  this.num = num;  }   public String getType() {  return type;  }   public void setType(String type) {  this.type = type;  }   public float getMoney() {  return money;  }   public void setMoney(float money) {  this.money = money;  }   public float getPrepay() {  return prepay;  }   public void setPrepay(float prepay) {  this.prepay = prepay;  }   public String getStatus() {  return status;  }   public void setStatus(String status) {  this.status = status;  }   @Override  public String toString() {  return "room{" +  "id=" + id +  ", num='" + num + '\'' +  ", type='" + type + '\'' +  ", money=" + money +  ", prepay=" + prepay +  ", status='" + status + '\'' +  '}';  }   @Override  public boolean equals(Object o) {  if (this == o) return true;  if (o == null || getClass() != o.getClass()) return false;  room room = (room) o;  return id == room.id;  }   @Override  public int hashCode() {  return Objects.hash(id);  } } |

|  |
| --- |
| room\_type.java |
| package cn.edu.sut.hotel\_management.model;  public class room\_type {  private final int id;  private final String type;   public room\_type(int id, String type) {  this.id = id;  this.type = type;  }   @Override  public String toString() {  return "room\_type{" +  "id=" + id +  ", type='" + type + '\'' +  '}';  } } |

|  |
| --- |
| user.java |
| package cn.edu.sut.hotel\_management.model;  public class user {  public long id;  private String img\_url;  private String name;  private String id\_card;  private String sex;  private String tel;  private String status;  private String account;  private String user\_name;  private String password;   public user(long id, String img\_url, String name, String id\_card, String sex, String tel, String status, String account, String user\_name, String password) {  this.id = id;  this.img\_url = img\_url;  this.name = name;  this.id\_card = id\_card;  this.sex = sex;  this.tel = tel;  this.status = status;  this.account = account;  this.user\_name = user\_name;  this.password = password;  }   public user(long id, String name, String tel, String account, String user\_name) {  this.id = id;  this.name = name;  this.tel = tel;  this.account = account;  this.user\_name = user\_name;  }   public long getId() {  return id;  }   public void setId(long id) {  this.id = id;  }   public String getImg\_url() {  return img\_url;  }   public void setImg\_url(String img\_url) {  this.img\_url = img\_url;  }   public String getName() {  return name;  }   public void setName(String name) {  this.name = name;  }   public String getId\_card() {  return id\_card;  }   public void setId\_card(String id\_card) {  this.id\_card = id\_card;  }   public String getSex() {  return sex;  }   public void setSex(String sex) {  this.sex = sex;  }   public String getTel() {  return tel;  }   public void setTel(String tel) {  this.tel = tel;  }   public String getStatus() {  return status;  }   public void setStatus(String status) {  this.status = status;  }   public String getAccount() {  return account;  }   public void setAccount(String account) {  this.account = account;  }   public String getUser\_name() {  return user\_name;  }   public void setUser\_name(String user\_name) {  this.user\_name = user\_name;  }   public String getPassword() {  return password;  }   public void setPassword(String password) {  this.password = password;  }   @Override  public String toString() {  return "user{" +  "id=" + id +  ", img\_url='" + img\_url + '\'' +  ", name='" + name + '\'' +  ", id\_card='" + id\_card + '\'' +  ", sex='" + sex + '\'' +  ", tel='" + tel + '\'' +  ", status='" + status + '\'' +  ", account='" + account + '\'' +  ", user\_name='" + user\_name + '\'' +  ", password='" + password + '\'' +  '}';  } } |

|  |
| --- |
| delete.java |
| package cn.edu.sut.hotel\_management.sql;  import cn.edu.sut.hotel\_management.util.Api;  import java.sql.PreparedStatement; import java.sql.SQLException;  public class delete {   public static int deleteOrder(Long id) throws SQLException {  Api api = new Api();  String sql = "DELETE FROM `order` WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, id);  return api.update(pStatement);  }   public static int deleteUser(String id) throws SQLException {  Api api = new Api();  String sql = "DELETE FROM `user` WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  return api.update(pStatement);  } } |

|  |
| --- |
| insert.java |
| package cn.edu.sut.hotel\_management.sql;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.util.Api;  import java.io.FileInputStream; import java.io.IOException; import java.sql.\*;  public class insert {  public static int addUser(String id, String nameString, String phoneString, String roomid, int duration, float deposit, String checkin) throws SQLException {  Api api = new Api();  String sql = "insert into user values(?, ?, ?, ?, ?,?,?)";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  pStatement.setString(2, nameString);  pStatement.setString(3, phoneString);  pStatement.setString(4, roomid);  pStatement.setInt(5, duration);  pStatement.setFloat(6, deposit);  pStatement.setString(7, checkin);  return api.update(pStatement);  }   public static int addOrder(String begin, int stay, float prepay, float pay, float rental, String id\_card, String name, String phone) throws SQLException {  Api api = new Api();  String sql = "INSERT INTO `order` (`begin`, `stay`, `prepay`, `pay`, `rental`, `id\_card`, `name`, `phone`) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";  PreparedStatement pStatement = api.getPreparedStatementWithKey(sql);   pStatement.setString(1, begin);  pStatement.setInt(2, stay);  pStatement.setFloat(3, prepay);  pStatement.setFloat(4, pay);  pStatement.setFloat(5, rental);  pStatement.setString(6, id\_card);  pStatement.setString(7, name);  pStatement.setString(8, phone);   pStatement.executeUpdate();  ResultSet resultSet = pStatement.getGeneratedKeys();  int idx;  if (resultSet.next()) {  idx = resultSet.getInt(1);  return idx;  }  return -1;  }    public static int addRealtion(String user\_id, String room\_id, String order\_id, String status) throws SQLException {  Api api = new Api();  String sql = "INSERT INTO `hotel\_management`.`relation` (`user\_id`, `room\_id`, `order\_id`, `status`) VALUES (?, ?, ?, ?)";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, user\_id);  pStatement.setString(2, room\_id);  pStatement.setString(3, order\_id);  pStatement.setString(4, status);  return api.update(pStatement);  }    public static int register(String account, String password) throws SQLException {  Api api = new Api();  String sql = "INSERT INTO `user` (`account`, `password`) VALUES (?,?)";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, account);  pStatement.setString(2, password);  return api.update(pStatement);  }   public static int addRoom(String num, String type\_id, String price\_id) throws SQLException {  Api api = new Api();  String sql = "INSERT INTO `room` (`num`, `type\_id`, `price\_id`, `status`) VALUES (?, ?, ?, '新增')";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, num);  pStatement.setString(2, type\_id);  pStatement.setString(3, price\_id);  return api.update(pStatement);  }   public static int addPrice(float prepay, float money) throws SQLException {  Api api = new Api();  String sql = "INSERT INTO `price` (`prepay`,`money`) VALUES (?, ?)";  Connection conn = api.getConnection();  PreparedStatement pStatement = conn.prepareStatement(sql, Statement.RETURN\_GENERATED\_KEYS);   pStatement.setFloat(1, prepay);  pStatement.setFloat(2, money);  System.out.println(pStatement);  api.update(pStatement);  ResultSet resultSet = pStatement.getGeneratedKeys();  int idx;  if (resultSet.next()) {  idx = resultSet.getInt(1);  return idx;  }  return -1;  }   public static int addPicture(String path) {  Api api = new Api();  try {  String sql = "UPDATE `user` SET `img` = ? WHERE `id` = ?";  FileInputStream in = new FileInputStream(path);  Connection conn = api.getConnection();  PreparedStatement pStatement = conn.prepareStatement(sql, Statement.RETURN\_GENERATED\_KEYS);  pStatement.setBlob(1, in);  pStatement.setLong(2, curuser.getId());  api.update(pStatement);  ResultSet resultSet = pStatement.getGeneratedKeys();  int idx;  if (resultSet.next()) {  idx = resultSet.getInt(1);  return idx;  }  return -1;  } catch (SQLException | IOException e) {  throw new RuntimeException(e);  }  }  } |

|  |
| --- |
| select.java |
| package cn.edu.sut.hotel\_management.sql;  import cn.edu.sut.hotel\_management.model.\*; import cn.edu.sut.hotel\_management.util.Api;  import java.io.InputStream; import java.sql.\*;  public class select {   public static order getOrder(long id) throws SQLException {  Api api = new Api();  String sql = "select \* from `order` where id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, id);  ResultSet rSet = api.select(pStatement);   if (rSet.next()) {  return new order(rSet.getInt(1), rSet.getString(2), rSet.getInt(3), rSet.getFloat(4), rSet.getFloat(5), rSet.getFloat(6), rSet.getString(7), rSet.getString(8), rSet.getString(9));  } else {  return null;  }  }   public static Object[][] getAllMyOrder(long id) throws SQLException {  Api api = new Api();  String sql = "SELECT `order`.\* FROM `order` INNER JOIN relation ON `order`.id = relation.order\_id INNER JOIN `user` ON relation.user\_id = `user`.id WHERE `user`.id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, id);  ResultSet rSet = api.select(pStatement);   int row = 0;  int col = 0;  while (rSet.next()) {  row++;  }  if (row == 0) {  return null;  }  Object[][] data;  data = new Object[row][6];  rSet.beforeFirst();//这里需要将指针移回到第一行之前  for (; row > 0; row--) {  rSet.next();  data[col][0] = rSet.getLong(1);  data[col][1] = rSet.getString(2);  data[col][2] = rSet.getString(3);  data[col][3] = rSet.getString(4);  data[col][4] = rSet.getString(5);  data[col][5] = rSet.getInt(6);  col++;  }  return data;  }   public static int getDateCur(String start) throws SQLException {  Api api = new Api();  String sql = "SELECT DATEDIFF(?,CURDATE())";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, start);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getInt(1);  }  return -1;  }   public static Object[][] getAllRoom() throws SQLException {  Api api = new Api();  String sql = "SELECT room.id, room.num, room\_type.type, price.prepay, price.money, room.`status` FROM room INNER JOIN room\_type ON room.type\_id = room\_type.id INNER JOIN price ON room.price\_id = price.id";  PreparedStatement pStatement = api.getPreparedStatement(sql);  ResultSet rSet = api.select(pStatement);   int row = 0;  int col = 0;  while (rSet.next()) {  row++;  }  if (row == 0) {  return null;  }  Object[][] data;  data = new Object[row][6];  rSet.beforeFirst();//这里需要将指针移回到第一行之前  for (; row > 0; row--) {  rSet.next();  data[col][0] = rSet.getString(1);  data[col][1] = rSet.getString(2);  data[col][2] = rSet.getString(3);  data[col][3] = rSet.getString(4);  data[col][4] = rSet.getString(5);  data[col][5] = rSet.getString(6);  col++;  }  return data;  }   public static Object[][] getAllUser() throws SQLException {  Api api = new Api();  String sql = "SELECT `user`.id, `user`.account,`user`.user\_name, `user`.`name`, `user`.tel FROM `user`";  PreparedStatement pStatement = api.getPreparedStatement(sql);  ResultSet rSet = api.select(pStatement);   int row = 0;  int col = 0;  while (rSet.next()) {  row++;  }  if (row == 0) {  return null;  }  Object[][] data;  data = new Object[row][6];  rSet.beforeFirst();//这里需要将指针移回到第一行之前  for (; row > 0; row--) {  rSet.next();  data[col][0] = rSet.getString(1);  data[col][1] = rSet.getString(2);  data[col][2] = rSet.getString(3);  data[col][3] = rSet.getString(4);  data[col][4] = rSet.getString(5);  data[col][5] = (select.isAdmin(rSet.getString(1)) ? "是" : "否");  col++;  }  return data;  }   public static relation getRelation(int id) throws SQLException {  Api api = new Api();  String sql = "select \* from `relation` where id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setInt(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return new relation(rSet.getInt(1), rSet.getLong(2), rSet.getInt(3), rSet.getLong(4), rSet.getInt(5), rSet.getString(6));   } else {  return null;  }  }    public static String[] getAllRoomType() throws SQLException {  Api api = new Api();  String sql = "SELECT DISTINCT room\_type.type FROM room\_type";  PreparedStatement pStatement = api.getPreparedStatement(sql);  ResultSet rSet = api.select(pStatement);   int row = 0;  int col = 0;  while (rSet.next()) {  row++;  }  if (row == 0) {  return null;  }  String[] data;  data = new String[row];  rSet.beforeFirst();//这里需要将指针移回到第一行之前  for (; row > 0; row--) {  rSet.next();  data[col] = rSet.getString(1);  col++;  }  return data;  }   public static String[] getAllRoomNum(String type) throws SQLException {  Api api = new Api();  String sql = "SELECT room.num FROM room\_type INNER JOIN room ON room\_type.id = room.type\_id WHERE room\_type.type = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, type);  ResultSet rSet = api.select(pStatement);   int row = 0;  int col = 0;  while (rSet.next()) {  row++;  }  if (row == 0) {  return null;  }  String[] data;  data = new String[row];  rSet.beforeFirst();//这里需要将指针移回到第一行之前  for (; row > 0; row--) {  rSet.next();  data[col] = rSet.getString(1);  col++;  }  return data;  }   public static room getRoom(int id) throws SQLException {  Api api = new Api();  String sql = "SELECT room.id, room.num, room\_type.type, price.prepay, price.money, room.`status` FROM room INNER JOIN room\_type ON room.type\_id = room\_type.id INNER JOIN price ON room.price\_id = price.id WHERE room.id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setInt(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return new room(rSet.getInt(1), rSet.getString(2), rSet.getString(3), rSet.getFloat(4), rSet.getFloat(5), rSet.getString(6));  } else {  return null;  }  }   public static float[] getRoomPrice(String id) throws SQLException {  Api api = new Api();  String sql = "SELECT\n" + " price.prepay,\n" + " price.money\n" + " FROM\n" + " price\n" + " INNER JOIN\n" + " room\n" + " ON\n" + " price.id = room.price\_id\n" + " WHERE\n" + " room.num = ?\n";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return new float[]{rSet.getFloat(1), rSet.getFloat(2)};  } else {  return null;  }  }    public static boolean isAccountExist(String account) throws SQLException {  Api api = new Api();  String sql = "SELECT account FROM `user` WHERE `user`.account = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, account);  ResultSet rSet = api.select(pStatement);  return rSet.next();  }   public static String getPriceId(float prepay, float money) throws SQLException {  Api api = new Api();  String sql = "SELECT price.id FROM price WHERE price.prepay = ? AND price.money = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setFloat(1, prepay);  pStatement.setFloat(2, money);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getString(1);  }  return null;  }   public static String getRoomId(String num) throws SQLException {  Api api = new Api();  String sql = "SELECT room.id FROM room WHERE room.num = ? ";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, num);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getString(1);   }  return null;  }   public static String getTypeId(String num) throws SQLException {  Api api = new Api();  String sql = "SELECT room\_type.id FROM room\_type WHERE room\_type.type = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, num);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getString(1);   }  return null;  }   public static String getOrderNum() throws SQLException {  Api api = new Api();  String sql = "SELECT COUNT(`order`.id) FROM `order` INNER JOIN relation ON `order`.id = relation.order\_id INNER JOIN `user` ON `user`.id = relation.user\_id WHERE `user`.id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, curuser.getId());  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getString(1);  }  return null;  }   public static int getDateDiff(String end,String start) throws SQLException {  Api api = new Api();  String sql = "SELECT DATEDIFF(?,?)";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, start);  pStatement.setString(2, end);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getInt(1);  }  return -1;  }   public static curuser getCurUser(long id) throws SQLException {  Api api = new Api();  String sql = "select \* from `user` where id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return new curuser(rSet.getLong(1), rSet.getString(2), rSet.getString(3), rSet.getString(4), rSet.getString(5), rSet.getString(6), rSet.getString(7), rSet.getString(8), rSet.getString(9), rSet.getString(10));  } else {  return null;  }  }   public static user getUser(long id) throws SQLException {  Api api = new Api();  String sql = "SELECT `user`.`name`, `user`.tel, `user`.account, `user`.user\_name FROM `user` WHERE `user`.id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return new user(id, rSet.getString(1), rSet.getString(2), rSet.getString(3), rSet.getString(4));  } else {  return null;  }  }   public static boolean login(String id, String password) throws SQLException {  Api api = new Api();  String sql = "SELECT `password` FROM `user` WHERE account = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  String pwd = rSet.getString(1);  return password.equals(pwd);  }  return false;  }   public static boolean isAdmin() throws SQLException {  Api api = new Api();  String sql = "SELECT `user`.admin FROM `user` WHERE `user`.id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, curuser.getId());  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  int isad = rSet.getInt(1);  return isad == 1;  }  return false;  }   public static boolean isAdmin(String id) throws SQLException {  Api api = new Api();  String sql = "SELECT `user`.admin FROM `user` WHERE `user`.id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  int isad = rSet.getInt(1);  return isad == 1;  }  return false;  }   public static String getPassword() throws SQLException {  Api api = new Api();  String sql = "SELECT `password` FROM `user` WHERE id = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setLong(1, curuser.getId());  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getString(1);  }  return null;  }   public static long getId(String account) throws SQLException {  Api api = new Api();  String sql = "SELECT `id` FROM `user` WHERE account = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, account);  ResultSet rSet = api.select(pStatement);  if (rSet.next()) {  return rSet.getLong(1);   }  return -1;  }   public static InputStream loadPicture() throws SQLException {   Api api = new Api();  String sql = "SELECT img from `user` WHERE id= ?";  Connection conn = api.getConnection();  PreparedStatement ps = conn.prepareStatement(sql, Statement.RETURN\_GENERATED\_KEYS);  ps.setLong(1, curuser.getId());  ResultSet resultSet = ps.executeQuery();  InputStream img;  if (resultSet.next()) {  img = resultSet.getBinaryStream(1);  return img;  }  return null;  } } |

|  |
| --- |
| update.java |
| package cn.edu.sut.hotel\_management.sql;  import cn.edu.sut.hotel\_management.util.Api;  import java.sql.PreparedStatement; import java.sql.SQLException;  public class update {  public static int updateCurUser(Long id, String name, String id\_card, String tel, String user\_name) throws SQLException {  Api api = new Api();  String sql = "UPDATE `user` SET `name` = ?, `id\_card` = ?, `tel` = ?, `user\_name` = ? WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, name);  pStatement.setString(2, id\_card);  pStatement.setString(3, tel);  pStatement.setString(4, user\_name);  pStatement.setLong(5, id);  return api.update(pStatement);  }   public static int updateUser(Long id, String name, String tel, String user\_name) throws SQLException {  Api api = new Api();  String sql = "UPDATE `user` SET `name` = ?, `tel` = ?, `user\_name` = ? WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, name);  pStatement.setString(2, tel);  pStatement.setString(3, user\_name);  pStatement.setLong(4, id);  return api.update(pStatement);  }   public static int updatePassword(Long id, String password) throws SQLException {  Api api = new Api();  String sql = "UPDATE `user` SET `password` = ? WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, password);  pStatement.setLong(2, id);  return api.update(pStatement);  }   public static int setAdmin(String id) throws SQLException {  Api api = new Api();  String sql = "UPDATE `user` SET `admin` = 1 WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  return api.update(pStatement);  }   public static int unsetAdmin(String id) throws SQLException {  Api api = new Api();  String sql = "UPDATE `user` SET `admin` = 0 WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, id);  return api.update(pStatement);  }   public static int updateRoom(String id, String num, String type\_id, String price\_id) throws SQLException {  Api api = new Api();  String sql = "UPDATE `room` SET `num` = ?, `type\_id` = ?, `price\_id` = ? WHERE `id` = ?";  PreparedStatement pStatement = api.getPreparedStatement(sql);  pStatement.setString(1, num);  pStatement.setString(2, type\_id);  pStatement.setString(3, price\_id);  pStatement.setString(4, id);  return api.update(pStatement);  } } |

|  |
| --- |
| test.java |
| package cn.edu.sut.hotel\_management.test;  import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select;  import java.sql.SQLException;  public class test {  public static void main(String[] args) throws SQLException {  select.getRelation(1);  insert.addUser("", "", "", "", 1, 1, "");  } } |

|  |
| --- |
| AddRoom.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class AddRoom extends JFrame {  public AddRoom() {  this.setSize(350, 280);  this.setLocationRelativeTo(null);   GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel room\_inf = new JPanel();  GridBagLayout room\_inf\_layout = new GridBagLayout();  GridBagConstraints room\_inf\_constraint = new GridBagConstraints();  room\_inf.setLayout(room\_inf\_layout);  // JLabel r\_num\_label = new JLabel("房间号："); // r\_num\_label.setFont(new Font("宋体", Font.BOLD, 16)); // room\_inf\_constraint.insets = new Insets(0, 0, 0, 0); // room\_inf\_constraint.gridx = 0; // room\_inf\_constraint.gridy = 0; // room\_inf\_layout.setConstraints(r\_num\_label, room\_inf\_constraint); // room\_inf.add(r\_num\_label); // // JTextField r\_num = new JTextField(20); // room\_inf\_constraint.insets = new Insets(0, 0, 0, 0); // room\_inf\_constraint.gridx = 1; // room\_inf\_constraint.gridy = 0; // room\_inf\_layout.setConstraints(r\_num, room\_inf\_constraint); // room\_inf.add(r\_num);   JLabel num\_label = new JLabel("房间号：");  num\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 1;  room\_inf\_layout.setConstraints(num\_label, room\_inf\_constraint);  room\_inf.add(num\_label);   JTextField num = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 1;  room\_inf\_layout.setConstraints(num, room\_inf\_constraint);  room\_inf.add(num);   JPanel room\_bar = new JPanel();  GridBagLayout room\_bar\_layout = new GridBagLayout();  GridBagConstraints room\_bar\_constraint = new GridBagConstraints();  room\_bar.setLayout(room\_bar\_layout);   JLabel room\_type\_label1 = new JLabel("房型:");  room\_type\_label1.setFont(new Font("宋体", Font.BOLD, 16));  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridheight = 1;  room\_bar\_constraint.gridwidth = 1;  room\_bar\_constraint.gridx = 0;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type\_label1, room\_bar\_constraint);  room\_bar.add(room\_type\_label1);   JComboBox<String> room\_type = new JComboBox<>();  room\_type.addItem("-请选择-");  try {  String[] datas = select.getAllRoomType();  assert datas != null;  for (String data : datas) {  room\_type.addItem(data);  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridx = 1;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type, room\_bar\_constraint);  room\_bar.add(room\_type);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 2;  layout.setConstraints(room\_bar, constraint);  this.add(room\_bar);   JLabel prepay\_label = new JLabel("预付款：");  prepay\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 3;  room\_inf\_layout.setConstraints(prepay\_label, room\_inf\_constraint);  room\_inf.add(prepay\_label);   JTextField prepay = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 3;  room\_inf\_layout.setConstraints(prepay, room\_inf\_constraint);  room\_inf.add(prepay);   JLabel money\_label = new JLabel("租金：");  money\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 5;  room\_inf\_layout.setConstraints(money\_label, room\_inf\_constraint);  room\_inf.add(money\_label);   JTextField money = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 5;  room\_inf\_layout.setConstraints(money, room\_inf\_constraint);  room\_inf.add(money);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("修改");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("返回");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);   btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String snum = num.getText();  float sprepay = Float.parseFloat(prepay.getText());  float smoney = Float.parseFloat(money.getText());  String sroomtype = null;  String spriceid = null;  try {  sroomtype = select.getTypeId((String) room\_type.getSelectedItem());  spriceid = select.getPriceId(sprepay, smoney);  if (spriceid == null) {  int priceid = insert.addPrice(sprepay, smoney);  spriceid = String.valueOf(priceid);  }  insert.addRoom(snum, sroomtype, spriceid);  JOptionPane.showMessageDialog(null, "添加成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "添加失败！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });   btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(room\_inf, constraint);  this.add(room\_inf);   constraint.insets = new Insets(50, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 3;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);  }   public static void create() {  AddRoom addRoom = new AddRoom();  addRoom.setSize(600, 350);  addRoom.setTitle("添加房间");  addRoom.setLocationRelativeTo(null);  addRoom.setVisible(true);  } } |

|  |
| --- |
| ChangePassword.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.sql.update;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class ChangePassword extends JFrame {  public ChangePassword() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel change\_pwd = new JPanel();  GridBagLayout change\_pwd\_layout = new GridBagLayout();  GridBagConstraints change\_pwd\_constraint = new GridBagConstraints();  change\_pwd.setLayout(change\_pwd\_layout);   JLabel old\_pwd\_label = new JLabel("旧密码：");  old\_pwd\_label.setFont(new Font("宋体", Font.BOLD, 16));  change\_pwd\_constraint.insets = new Insets(0, 0, 0, 0);  change\_pwd\_constraint.gridx = 0;  change\_pwd\_constraint.gridy = 2;  change\_pwd\_layout.setConstraints(old\_pwd\_label, change\_pwd\_constraint);  change\_pwd.add(old\_pwd\_label);   JTextField old\_pwd = new JTextField(20);  change\_pwd\_constraint.insets = new Insets(0, 0, 0, 0);  change\_pwd\_constraint.gridx = 1;  change\_pwd\_constraint.gridy = 2;  change\_pwd\_layout.setConstraints(old\_pwd, change\_pwd\_constraint);  change\_pwd.add(old\_pwd);   JLabel new\_pwd\_label = new JLabel("新密码：");  new\_pwd\_label.setFont(new Font("宋体", Font.BOLD, 16));  change\_pwd\_constraint.insets = new Insets(0, 0, 0, 0);  change\_pwd\_constraint.gridx = 0;  change\_pwd\_constraint.gridy = 3;  change\_pwd\_layout.setConstraints(new\_pwd\_label, change\_pwd\_constraint);  change\_pwd.add(new\_pwd\_label);   JTextField new\_pwd = new JTextField(20);  change\_pwd\_constraint.insets = new Insets(0, 0, 0, 0);  change\_pwd\_constraint.gridx = 1;  change\_pwd\_constraint.gridy = 3;  change\_pwd\_layout.setConstraints(new\_pwd, change\_pwd\_constraint);  change\_pwd.add(new\_pwd);   JLabel confirm\_pwd\_label = new JLabel("确认密码：");  confirm\_pwd\_label.setFont(new Font("宋体", Font.BOLD, 16));  change\_pwd\_constraint.insets = new Insets(0, 0, 0, 0);  change\_pwd\_constraint.gridx = 0;  change\_pwd\_constraint.gridy = 4;  change\_pwd\_layout.setConstraints(confirm\_pwd\_label, change\_pwd\_constraint);  change\_pwd.add(confirm\_pwd\_label);   JTextField confirm\_pwd = new JTextField(20);  change\_pwd\_constraint.insets = new Insets(0, 0, 0, 0);  change\_pwd\_constraint.gridx = 1;  change\_pwd\_constraint.gridy = 4;  change\_pwd\_layout.setConstraints(confirm\_pwd, change\_pwd\_constraint);  change\_pwd.add(confirm\_pwd);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("确认");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("取消");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);   btnA.addActionListener(new ActionListener() {   @Override  public void actionPerformed(ActionEvent e) {  String sold = old\_pwd.getText();  String snew = new\_pwd.getText();  String scon = confirm\_pwd.getText();  try {  if (sold.equals(select.getPassword()) && snew.equals(scon)) {  update.updatePassword(curuser.getId(), confirm\_pwd.getText());  JOptionPane.showMessageDialog(null, "修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);  } else {  JOptionPane.showMessageDialog(null, "修改失败！", "提示", JOptionPane.ERROR\_MESSAGE);  }  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "修改失败！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });   btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(change\_pwd, constraint);  this.add(change\_pwd);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);  }   public static void create() {  ChangePassword changePassword = new ChangePassword();  changePassword.setSize(350, 200);  changePassword.setLocationRelativeTo(null);  changePassword.setTitle("修改密码");  changePassword.setVisible(true);  } } |

|  |
| --- |
| ContainerComponent.java |
| package cn.edu.sut.hotel\_management.ui.component;  import javax.swing.\*;  public class ContainerComponent extends JFrame {  public ContainerComponent() {  super("酒店管理系统");  this.getContentPane().add(new Tab());  this.pack();  this.setSize(800, 500);  this.setLocationRelativeTo(null);  this.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);  } } |

|  |
| --- |
| OrderTable.java |
| package cn.edu.sut.hotel\_management.ui.component;  import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.sql.SQLException;   public class OrderTable extends JScrollPane {  public DefaultTableModel getModel(long id) throws SQLException {  DefaultTableModel tableM = new DefaultTableModel(select.getAllMyOrder(id), new String[]{"订单号", "始住时间", "居住天数", "最低预付款", "实付款", "总租金"});  return tableM;  }   public JTable getTable(long id) throws SQLException {  JTable jTable = new JTable(getModel(id));  return jTable;  } } |

|  |
| --- |
| RoomTable.java |
| package cn.edu.sut.hotel\_management.ui.component;  import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.sql.SQLException;  public class RoomTable extends JScrollPane {  public DefaultTableModel getModel() throws SQLException {  DefaultTableModel tableM = new DefaultTableModel(select.getAllRoom(), new String[]{"房间id", "房间号", "房间类型", "预付款金额", "租金金额", "状态"});  return tableM;  }   public JTable getTable() throws SQLException {  JTable jTable = new JTable(getModel());  return jTable;  } } |

|  |
| --- |
| Tab.java |
| package cn.edu.sut.hotel\_management.ui.component;  import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.ui.\*;  import javax.swing.\*; import javax.swing.event.ChangeEvent; import javax.swing.event.ChangeListener; import java.awt.\*; import java.sql.SQLException;  public class Tab extends JTabbedPane {   public Tab() {  super(JTabbedPane.TOP, JTabbedPane.SCROLL\_TAB\_LAYOUT);  try {  this.setSize(400, 500);  Home home = new Home();  this.add("主页", home);  this.addTab("酒店预订", new Reservation());  this.addTab("订单管理", new OrderManagement());  this.addTab("用户中心", new UserCenter());  if (select.isAdmin()) {  this.addTab("房间管理", new RoomManagent());  this.addTab("用户管理", new UserManagent());  }   this.setFont(new Font("宋体", Font.BOLD, 20));  this.setSelectedIndex(0);   this.getSelectedIndex();  this.addChangeListener(new ChangeListener() {  @Override  public void stateChanged(ChangeEvent e) {  if (Tab.this.getSelectedIndex() == 0) {  home.reloadImage();  }  }  });  } catch (SQLException e) {  throw new RuntimeException(e);  }  } } |

|  |
| --- |
| UserTable.java |
| package cn.edu.sut.hotel\_management.ui.component;  import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.sql.SQLException;  public class UserTable extends JScrollPane {  public DefaultTableModel getModel() throws SQLException {  DefaultTableModel tableM = new DefaultTableModel(select.getAllUser(), new String[]{"用户id", "账号", "用户名", "姓名", "手机号", "是否管理员"});  return tableM;  }   public JTable getTable() throws SQLException {  return new JTable(getModel());  } } |

|  |
| --- |
| Home.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.util.ImageUtil;  import javax.swing.\*; import java.awt.\*; import java.io.IOException; import java.sql.SQLException;  public class Home extends JPanel {  JLabel image;   public Home() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel img\_bar = new JPanel();  GridBagLayout img\_bar\_layout = new GridBagLayout();  GridBagConstraints img\_bar\_constraint = new GridBagConstraints();  img\_bar.setLayout(img\_bar\_layout);   image = new JLabel(" 暂无照片 ");  image.setMinimumSize(new Dimension(150, 200));  image.setMaximumSize(new Dimension(150, 200));  image.setPreferredSize(new Dimension(150, 200));  img\_bar\_constraint.gridheight = 5;  img\_bar\_constraint.gridwidth = 3;  img\_bar\_constraint.gridx = 0;  img\_bar\_constraint.gridy = 1;  img\_bar\_layout.setConstraints(image, img\_bar\_constraint);  img\_bar.add(image);   JPanel user\_inf = new JPanel();  GridBagLayout user\_inf\_layout = new GridBagLayout();  GridBagConstraints user\_inf\_constraint = new GridBagConstraints();  user\_inf.setLayout(user\_inf\_layout);   JLabel user\_id\_label = new JLabel("用户唯一id：");  user\_id\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 0;  user\_inf\_layout.setConstraints(user\_id\_label, user\_inf\_constraint);  user\_inf.add(user\_id\_label);   JLabel user\_id = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 0;  user\_inf\_layout.setConstraints(user\_id, user\_inf\_constraint);  user\_inf.add(user\_id);   JLabel order\_num\_label = new JLabel("订单数量:");  order\_num\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 1;  user\_inf\_layout.setConstraints(order\_num\_label, user\_inf\_constraint);  user\_inf.add(order\_num\_label);   JLabel order\_num = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 1;  user\_inf\_layout.setConstraints(order\_num, user\_inf\_constraint);  user\_inf.add(order\_num);   JLabel id\_card\_label = new JLabel("身份证号：");  id\_card\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 2;  user\_inf\_layout.setConstraints(id\_card\_label, user\_inf\_constraint);  user\_inf.add(id\_card\_label);   JLabel id\_card = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 2;  user\_inf\_layout.setConstraints(id\_card, user\_inf\_constraint);  user\_inf.add(id\_card);   JLabel name\_label = new JLabel("姓名：");  name\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 3;  user\_inf\_layout.setConstraints(name\_label, user\_inf\_constraint);  user\_inf.add(name\_label);   JLabel name = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 3;  user\_inf\_layout.setConstraints(name, user\_inf\_constraint);  user\_inf.add(name);   JLabel phone\_label = new JLabel("手机号：");  phone\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 4;  user\_inf\_layout.setConstraints(phone\_label, user\_inf\_constraint);  user\_inf.add(phone\_label);   JLabel phone = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 4;  user\_inf\_layout.setConstraints(phone, user\_inf\_constraint);  user\_inf.add(phone);   JLabel welcome = new JLabel("欢迎使用 SUT酒店管理系统");  welcome.setFont(new Font("宋体", Font.BOLD, 30));  constraint.insets = new Insets(0, 0, 100, 0);  constraint.gridwidth = 2;  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(welcome, constraint);  this.add(welcome);   constraint.gridwidth = 1;  constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 2;  layout.setConstraints(img\_bar, constraint);  this.add(img\_bar);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 1;  constraint.gridy = 2;  layout.setConstraints(user\_inf, constraint);  this.add(user\_inf);  try {  order\_num.setText(select.getOrderNum());  } catch (SQLException e) {  throw new RuntimeException(e);  }  user\_id.setText(String.valueOf(curuser.getId())); // username.setText(user.getUser\_name());  id\_card.setText(curuser.getId\_card());  name.setText(curuser.getName());  phone.setText(curuser.getTel());  try {  ImageUtil.loadPicture(image);  } catch (IOException e) {  e.printStackTrace();  }  }   public void reloadImage() {  try {  ImageUtil.loadPicture(image);  } catch (IOException e) {  throw new RuntimeException(e);  }  } } |

|  |
| --- |
| Login.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.ui.component.ContainerComponent;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class Login extends JFrame {  public Login() {  setBounds(200, 200, 400, 250);  this.setLocationRelativeTo(null);  this.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);  setDefaultCloseOperation(EXIT\_ON\_CLOSE);  Container c = getContentPane();  c.setLayout(null);  setTitle("登录");  JLabel j1 = new JLabel("账号");  j1.setFont(new Font("宋体", Font.BOLD, 16));  j1.setBounds(50, 50, 100, 20);  JLabel j2 = new JLabel("密码");  j2.setFont(new Font("宋体", Font.BOLD, 16));  j2.setBounds(50, 100, 100, 20);  JTextField username = new JTextField();  username.setBounds(150, 50, 200, 20);  JPasswordField password = new JPasswordField();  password.setBounds(150, 100, 200, 20);  JButton login = new JButton("登录");  login.setFont(new Font("宋体", Font.BOLD, 16));  login.setBounds(80, 150, 100, 30);  JButton register = new JButton("注册");  register.setFont(new Font("宋体", Font.BOLD, 16));  register.setBounds(220, 150, 100, 30);  c.add(j1);  c.add(username);  c.add(j2);  c.add(password);  c.add(login);  c.add(register);  login.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String uname = username.getText();  String pwd = new String(password.getPassword());  try {  if (select.login(uname, pwd)) {  curuser.setId(select.getId(uname));  select.getCurUser(curuser.getId());  ContainerComponent container = new ContainerComponent();  container.setVisible(true);  System.out.println(select.getId(uname));  JOptionPane.showMessageDialog(null, "登录成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);  dispose();  } else {  JOptionPane.showMessageDialog(null, "登录失败！", "提示", JOptionPane.ERROR\_MESSAGE);  }  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  }  });  register.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  Register.create();  dispose();  }  });  } } |

|  |
| --- |
| OrderDetil.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.order; import cn.edu.sut.hotel\_management.sql.delete; import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class OrderDetil extends JFrame {  order o;   public OrderDetil(long idx) {  this.setSize(350, 280);  this.setLocationRelativeTo(null);   GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel order\_inf = new JPanel();  GridBagLayout order\_inf\_layout = new GridBagLayout();  GridBagConstraints order\_inf\_constraint = new GridBagConstraints();  order\_inf.setLayout(order\_inf\_layout);   JLabel o\_id\_label = new JLabel("订单号：");  o\_id\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 0;  order\_inf\_layout.setConstraints(o\_id\_label, order\_inf\_constraint);  order\_inf.add(o\_id\_label);   JLabel o\_id = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 0;  order\_inf\_layout.setConstraints(o\_id, order\_inf\_constraint);  order\_inf.add(o\_id);   JLabel id\_label = new JLabel("身份证号：");  id\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 1;  order\_inf\_layout.setConstraints(id\_label, order\_inf\_constraint);  order\_inf.add(id\_label);   JLabel id = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 1;  order\_inf\_layout.setConstraints(id, order\_inf\_constraint);  order\_inf.add(id);   JLabel name\_label = new JLabel("姓名：");  name\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 2;  order\_inf\_layout.setConstraints(name\_label, order\_inf\_constraint);  order\_inf.add(name\_label);   JLabel name = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 2;  order\_inf\_layout.setConstraints(name, order\_inf\_constraint);  order\_inf.add(name);   JLabel phone\_label = new JLabel("手机号：");  phone\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 3;  order\_inf\_layout.setConstraints(phone\_label, order\_inf\_constraint);  order\_inf.add(phone\_label);   JLabel phone = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 3;  order\_inf\_layout.setConstraints(phone, order\_inf\_constraint);  order\_inf.add(phone);   JLabel start\_label = new JLabel("起租时间：");  start\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 5;  order\_inf\_layout.setConstraints(start\_label, order\_inf\_constraint);  order\_inf.add(start\_label);   JLabel start = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 5;  order\_inf\_layout.setConstraints(start, order\_inf\_constraint);  order\_inf.add(start);   JLabel stay\_label = new JLabel("居住天数：");  stay\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 6;  order\_inf\_layout.setConstraints(stay\_label, order\_inf\_constraint);  order\_inf.add(stay\_label);    JLabel stay = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 6;  order\_inf\_layout.setConstraints(stay, order\_inf\_constraint);  order\_inf.add(stay);   JLabel pay\_label = new JLabel("实付款：");  pay\_label.setFont(new Font("宋体", Font.BOLD, 16));  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 0;  order\_inf\_constraint.gridy = 7;  order\_inf\_layout.setConstraints(pay\_label, order\_inf\_constraint);  order\_inf.add(pay\_label);   JLabel pay = new JLabel();  order\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  order\_inf\_constraint.gridx = 1;  order\_inf\_constraint.gridy = 7;  order\_inf\_layout.setConstraints(pay, order\_inf\_constraint);  order\_inf.add(pay);    JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("取消订单");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("返回");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);    constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(order\_inf, constraint);  this.add(order\_inf);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 3;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar); // order o = new order(idx);  try {  o = select.getOrder(idx);  } catch (SQLException e) {  throw new RuntimeException(e);  }  o\_id.setText(String.valueOf(o.getId()));  id.setText(o.getId\_card());  name.setText(o.getName());  phone.setText(o.getPhone());  start.setText(o.getBegin());  stay.setText(String.valueOf(o.getStay()));  pay.setText(String.valueOf(o.getPay()));   btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  try {  int datesur = select.getDateCur(o.getBegin());  if (o.getStay() + datesur < 0) {  int opt = JOptionPane.showConfirmDialog(null, "该订单已结束,您要将其删除吗？", "询问", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);  if (opt == JOptionPane.YES\_OPTION) {  delete.deleteOrder(idx);  dispose();  JOptionPane.showConfirmDialog(null, "删除成功", "提示", JOptionPane.DEFAULT\_OPTION);  }  return;  }  if (datesur >= 0) {  int opt = JOptionPane.showConfirmDialog(null, "将给您退款" + String.format("%.2f", o.getPay()) + "元", "退款", JOptionPane.OK\_CANCEL\_OPTION);  if (opt == JOptionPane.YES\_OPTION) {  delete.deleteOrder(idx);  JOptionPane.showConfirmDialog(null, "退款成功", "提示", JOptionPane.DEFAULT\_OPTION);   }  return;  }  float money = o.getPay() \*(o.getStay()+datesur)/(float)o.getStay();  if (money <= 0) {  JOptionPane.showMessageDialog(null, "当前订单生效，但无款可退！", "提示", JOptionPane.ERROR\_MESSAGE);  dispose();  return;  }  int opt = JOptionPane.showConfirmDialog(null, "将给您退款" + String.format("%.2f", money) + "元", "退款", JOptionPane.OK\_CANCEL\_OPTION);  if (opt == JOptionPane.YES\_OPTION) {  delete.deleteOrder(idx);  JOptionPane.showConfirmDialog(null, "退款成功", "提示", JOptionPane.DEFAULT\_OPTION);   }  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "订单出错！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });   btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });  } } |

|  |
| --- |
| OrderManagement.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.ui.component.OrderTable;  import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;   public class OrderManagement extends JPanel {   JTable myTable;   public OrderManagement() throws SQLException {  System.out.println(curuser.getId());  OrderTable orderTableComponent = new OrderTable();   setLayout(new BorderLayout());   JPanel jPanel = new JPanel();   JLabel jl1 = new JLabel("按住CTRL可多选");  JButton look = new JButton("查看/管理");  JButton reload = new JButton("刷新");    myTable = orderTableComponent.getTable(curuser.getId());  DefaultTableModel model = (DefaultTableModel) myTable.getModel();  orderTableComponent.setViewportView(myTable);  String[] hrader = new String[]{"订单号", "始住时间", "居住天数", "最低预付款", "实付款", "总租金"};  model.setDataVector(select.getAllMyOrder(curuser.getId()), hrader);    look.addActionListener(new ActionListener() {//查看事件  public void actionPerformed(ActionEvent e) {  int selectedRow = myTable.getSelectedRow();//获得选中行的索引  int selectedRows = myTable.getSelectedRows().length;   if (selectedRows > 1) {  JOptionPane.showMessageDialog(null, "每次只能查看一个！", "提示",  JOptionPane.INFORMATION\_MESSAGE);  return;  }  long id = (long) myTable.getValueAt(selectedRow, 0);  OrderDetil orderDetil = new OrderDetil(id);  orderDetil.setVisible(true);  }  });   reload.addActionListener(new ActionListener() {//刷新事件  public void actionPerformed(ActionEvent e) {  try {  model.setDataVector(select.getAllMyOrder(curuser.getId()), hrader);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  myTable.updateUI();  }  });  jPanel.add(jl1);  jPanel.add(look);  jPanel.add(reload);  add(jPanel, BorderLayout.NORTH);  add(orderTableComponent, BorderLayout.CENTER);  } } |

|  |
| --- |
| Register.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class Register extends JFrame {   public Register() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel register = new JPanel();  GridBagLayout register\_layout = new GridBagLayout();  GridBagConstraints register\_constraint = new GridBagConstraints();  register.setLayout(register\_layout);   JLabel account\_label = new JLabel("账号");  account\_label.setFont(new Font("宋体", Font.BOLD, 16));  register\_constraint.insets = new Insets(0, 0, 0, 0);  register\_constraint.gridx = 0;  register\_constraint.gridy = 0;  register\_layout.setConstraints(account\_label, register\_constraint);  register.add(account\_label);   JTextField account = new JTextField(20);  register\_constraint.insets = new Insets(0, 20, 0, 0);  register\_constraint.gridx = 1;  register\_constraint.gridy = 0;  register\_layout.setConstraints(account, register\_constraint);  register.add(account);   JLabel password\_label = new JLabel("密码");  password\_label.setFont(new Font("宋体", Font.BOLD, 16));  register\_constraint.insets = new Insets(0, 0, 0, 0);  register\_constraint.gridx = 0;  register\_constraint.gridy = 1;  register\_layout.setConstraints(password\_label, register\_constraint);  register.add(password\_label);   JTextField password = new JTextField(20);  register\_constraint.insets = new Insets(0, 20, 0, 0);  register\_constraint.gridx = 1;  register\_constraint.gridy = 1;  register\_layout.setConstraints(password, register\_constraint);  register.add(password);   JLabel confirm\_label = new JLabel("确认密码");  confirm\_label.setFont(new Font("宋体", Font.BOLD, 16));  register\_constraint.insets = new Insets(0, 0, 0, 0);  register\_constraint.gridx = 0;  register\_constraint.gridy = 2;  register\_layout.setConstraints(confirm\_label, register\_constraint);  register.add(confirm\_label);   JTextField confirm = new JTextField(20);  register\_constraint.insets = new Insets(0, 20, 0, 0);  register\_constraint.gridx = 1;  register\_constraint.gridy = 2;  register\_layout.setConstraints(confirm, register\_constraint);  register.add(confirm);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("注册");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("返回");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);   constraint.insets = new Insets(50, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(register, constraint);  this.add(register);   constraint.insets = new Insets(50, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);   btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String saccount = account.getText();  String spwd = password.getText();  String scon = confirm.getText();  try {  if (!select.isAccountExist(saccount)) {  insert.register(saccount, spwd);  JOptionPane.showMessageDialog(null, "注册成功，请完善信息", "提示", JOptionPane.INFORMATION\_MESSAGE);  } else {  JOptionPane.showMessageDialog(null, "注册失败,账号已存在", "提示", JOptionPane.ERROR\_MESSAGE);  }  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "注册失败，出现sql错误", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });  btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  Login l = new Login();  l.setVisible(true);  }  });  }    public static void create() {  Register register = new Register();  register.setSize(400, 300);  register.setTitle("注册");  register.setLocationRelativeTo(null);  register.setVisible(true);  }  } |

|  |
| --- |
| Reservation.java |
| package cn.edu.sut.hotel\_management.ui;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener;  public class Reservation extends JPanel {   public Reservation() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("为自己办理");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("为他人办理");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);   btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  ReservationForMe.create();  }  });   btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  ReservationForOther.create();  }  });   constraint.insets = new Insets(0, 50, 0, 0);  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);  } } |

|  |
| --- |
| ReservationForMe.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.awt.event.ItemEvent; import java.awt.event.ItemListener; import java.sql.SQLException;  public class ReservationForMe extends JFrame {  public ReservationForMe() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);  // constraint.insets = new Insets(0, 0, 0, 0); // JLabel picture\_label = new JLabel("照片"); // picture\_label.setFont(new Font("宋体", Font.BOLD, 16)); // constraint.gridx = 0; // constraint.gridy = 0; // layout.setConstraints(picture\_label, constraint); // this.add(picture\_label);    JPanel date\_select = new JPanel();  GridBagLayout date\_select\_layout = new GridBagLayout();  GridBagConstraints date\_select\_constraint = new GridBagConstraints();  date\_select.setLayout(date\_select\_layout);   JLabel date\_label1 = new JLabel("入住日期:");  date\_label1.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridheight = 1;  date\_select\_constraint.gridwidth = 1;  date\_select\_constraint.gridx = 0;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label1, date\_select\_constraint);  date\_select.add(date\_label1);    JComboBox<String> year = new JComboBox<>();  year.addItem("-请选择-");  for (int i = 2024; i >= 2023; i--) {  year.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 1;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(year, date\_select\_constraint);  date\_select.add(year);   JLabel date\_label2 = new JLabel("年");  date\_label2.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 2;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label2, date\_select\_constraint);  date\_select.add(date\_label2);   JComboBox<String> mon = new JComboBox<>();  mon.addItem("-请选择-");  for (int i = 12; i >= 1; i--) {  mon.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 3;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(mon, date\_select\_constraint);  date\_select.add(mon);   JLabel date\_label3 = new JLabel("月");  date\_label3.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 4;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label3, date\_select\_constraint);  date\_select.add(date\_label3);   JComboBox<String> day = new JComboBox<>();  day.addItem("-请选择-");  for (int i = 31; i >= 1; i--) {  day.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 5;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(day, date\_select\_constraint);  date\_select.add(day);   JLabel date\_label4 = new JLabel("日");  date\_label4.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 6;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label4, date\_select\_constraint);  date\_select.add(date\_label4);    JLabel date1\_label1 = new JLabel("离开日期:");  date1\_label1.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridheight = 1;  date\_select\_constraint.gridwidth = 1;  date\_select\_constraint.gridx = 0;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label1, date\_select\_constraint);  date\_select.add(date1\_label1);   JComboBox<String> year1 = new JComboBox<>();  year1.addItem("-请选择-");  for (int i = 2024; i >= 2023; i--) {  year1.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 1;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(year1, date\_select\_constraint);  date\_select.add(year1);   JLabel date1\_label2 = new JLabel("年");  date1\_label2.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 2;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label2, date\_select\_constraint);  date\_select.add(date1\_label2);   JComboBox<String> mon1 = new JComboBox<>();  mon1.addItem("-请选择-");  for (int i = 12; i >= 1; i--) {  mon1.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 3;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(mon1, date\_select\_constraint);  date\_select.add(mon1);   JLabel date1\_label3 = new JLabel("月");  date1\_label3.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 4;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label3, date\_select\_constraint);  date\_select.add(date1\_label3);   JComboBox<String> day1 = new JComboBox<>();  day1.addItem("-请选择-");  for (int i = 31; i >= 1; i--) {  day1.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 5;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(day1, date\_select\_constraint);  date\_select.add(day1);   JLabel date1\_label4 = new JLabel("日");  date1\_label4.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 6;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label4, date\_select\_constraint);  date\_select.add(date1\_label4);    JPanel room\_bar = new JPanel();  GridBagLayout room\_bar\_layout = new GridBagLayout();  GridBagConstraints room\_bar\_constraint = new GridBagConstraints();  room\_bar.setLayout(room\_bar\_layout);    JLabel room\_type\_label1 = new JLabel("房型:");  room\_type\_label1.setFont(new Font("宋体", Font.BOLD, 16));  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridheight = 1;  room\_bar\_constraint.gridwidth = 1;  room\_bar\_constraint.gridx = 0;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type\_label1, room\_bar\_constraint);  room\_bar.add(room\_type\_label1);   JComboBox<String> room\_type = new JComboBox<>();  room\_type.addItem("-请选择-");  try {  String[] datas = select.getAllRoomType();  assert datas != null;  for (String data : datas) {  room\_type.addItem(data);  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridx = 1;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type, room\_bar\_constraint);  room\_bar.add(room\_type);    JLabel room\_id\_label1 = new JLabel("房号:");  room\_id\_label1.setFont(new Font("宋体", Font.BOLD, 16));  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridheight = 1;  room\_bar\_constraint.gridwidth = 1;  room\_bar\_constraint.gridx = 0;  room\_bar\_constraint.gridy = 1;  room\_bar\_layout.setConstraints(room\_id\_label1, room\_bar\_constraint);  room\_bar.add(room\_id\_label1);   JComboBox<String> room\_id = new JComboBox<>();  room\_id.addItem("-请选择-");   room\_type.addItemListener(new ItemListener() {  @Override  public void itemStateChanged(ItemEvent e) {  try {  if (room\_type.getSelectedItem() == null) {  itemStateChanged(e);  }  String[] datas1 = select.getAllRoomNum((String) room\_type.getSelectedItem());  room\_id.removeAllItems();  for (String data : datas1) {  room\_id.addItem(data);  }  } catch (SQLException e1) {  throw new RuntimeException(e1);  }  }  });  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridx = 1;  room\_bar\_constraint.gridy = 1;  room\_bar\_layout.setConstraints(room\_id, room\_bar\_constraint);  room\_bar.add(room\_id);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnY = new JButton("确认");  btnY.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnY, btn\_bar\_constraint);  btn\_bar.add(btnY);  JButton btnN = new JButton("取消");  btnN.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnN, btn\_bar\_constraint);  btn\_bar.add(btnN);   btnY.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {   int stay = -1;  String syear = null;  String smon = null;  String sday = null;  String syear1 = null;  String smon1 = null;  String sday1 = null;  try {  syear = (String) year.getSelectedItem();  smon = (String) mon.getSelectedItem();  sday = (String) day.getSelectedItem();  syear1 = (String) year1.getSelectedItem();  smon1 = (String) mon1.getSelectedItem();  sday1 = (String) day1.getSelectedItem();  stay = select.getDateDiff(syear + "-" + smon + "-" + sday, syear1 + "-" + smon1 + "-" + sday1);  if (stay <= 0) {  JOptionPane.showMessageDialog(null, "时间填写有误！", "提示", JOptionPane.ERROR\_MESSAGE);  return;  }  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "时间填写有误！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  String sid = (String) room\_id.getSelectedItem();  float[] fprice;  try {  fprice = select.getRoomPrice(sid);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  if (fprice[0] <= 0 || fprice[1] <= 0) {  JOptionPane.showMessageDialog(null, "房间价格有误！", "提示", JOptionPane.ERROR\_MESSAGE);  return;  }   int opt = JOptionPane.showConfirmDialog(null, "您需要支付" + String.format("%.2f", fprice[0] \* stay) + "元", "支付确认", JOptionPane.OK\_CANCEL\_OPTION);  if (opt == JOptionPane.YES\_OPTION) {  try {  int order\_id = insert.addOrder(syear + "-" + smon + "-" + sday, stay, fprice[0] \* stay, fprice[0] \* stay, fprice[1] \* stay, curuser.getId\_card(), curuser.getName(), curuser.getTel());  String room\_id = select.getRoomId(sid);  insert.addRealtion(String.valueOf(curuser.getId()), room\_id, String.valueOf(order\_id), "有效");  JOptionPane.showConfirmDialog(null, "支付成功", "提示", JOptionPane.DEFAULT\_OPTION);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "信息填写有误！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }    }  });   btnN.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });    constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(date\_select, constraint);  this.add(date\_select);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 3;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);    constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 2;  layout.setConstraints(room\_bar, constraint);  this.add(room\_bar);   }   public static void create() {  ReservationForMe reservationForMe = new ReservationForMe();  reservationForMe.setSize(600, 350);  reservationForMe.setTitle("预定办理");  reservationForMe.setLocationRelativeTo(null);  reservationForMe.setVisible(true);  } } |

|  |
| --- |
| ReservationForOther.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.awt.event.ItemEvent; import java.awt.event.ItemListener; import java.sql.SQLException;  public class ReservationForOther extends JFrame {  public ReservationForOther() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel date\_select = new JPanel();  GridBagLayout date\_select\_layout = new GridBagLayout();  GridBagConstraints date\_select\_constraint = new GridBagConstraints();  date\_select.setLayout(date\_select\_layout);   JLabel date\_label1 = new JLabel("入住日期:");  date\_label1.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridheight = 1;  date\_select\_constraint.gridwidth = 1;  date\_select\_constraint.gridx = 0;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label1, date\_select\_constraint);  date\_select.add(date\_label1);   JComboBox<String> year = new JComboBox<>();  year.addItem("-请选择-");  for (int i = 2024; i >= 2023; i--) {  year.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 1;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(year, date\_select\_constraint);  date\_select.add(year);   JLabel date\_label2 = new JLabel("年");  date\_label2.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 2;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label2, date\_select\_constraint);  date\_select.add(date\_label2);   JComboBox<String> mon = new JComboBox<>();  mon.addItem("-请选择-");  for (int i = 12; i >= 1; i--) {  mon.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 3;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(mon, date\_select\_constraint);  date\_select.add(mon);   JLabel date\_label3 = new JLabel("月");  date\_label3.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 4;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label3, date\_select\_constraint);  date\_select.add(date\_label3);   JComboBox<String> day = new JComboBox<>();  day.addItem("-请选择-");  for (int i = 31; i >= 1; i--) {  day.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 5;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(day, date\_select\_constraint);  date\_select.add(day);   JLabel date\_label4 = new JLabel("日");  date\_label4.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 6;  date\_select\_constraint.gridy = 0;  date\_select\_layout.setConstraints(date\_label4, date\_select\_constraint);  date\_select.add(date\_label4);    JLabel date1\_label1 = new JLabel("离开日期:");  date1\_label1.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridheight = 1;  date\_select\_constraint.gridwidth = 1;  date\_select\_constraint.gridx = 0;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label1, date\_select\_constraint);  date\_select.add(date1\_label1);   JComboBox<String> year1 = new JComboBox<>();  year1.addItem("-请选择-");  for (int i = 2024; i >= 2023; i--) {  year1.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 1;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(year1, date\_select\_constraint);  date\_select.add(year1);   JLabel date1\_label2 = new JLabel("年");  date1\_label2.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 2;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label2, date\_select\_constraint);  date\_select.add(date1\_label2);   JComboBox<String> mon1 = new JComboBox<>();  mon1.addItem("-请选择-");  for (int i = 12; i >= 1; i--) {  mon1.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 3;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(mon1, date\_select\_constraint);  date\_select.add(mon1);   JLabel date1\_label3 = new JLabel("月");  date1\_label3.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 4;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label3, date\_select\_constraint);  date\_select.add(date1\_label3);   JComboBox<String> day1 = new JComboBox<>();  day1.addItem("-请选择-");  for (int i = 31; i >= 1; i--) {  day1.addItem(String.valueOf(i));  }  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 5;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(day1, date\_select\_constraint);  date\_select.add(day1);   JLabel date1\_label4 = new JLabel("日");  date1\_label4.setFont(new Font("宋体", Font.BOLD, 16));  date\_select\_constraint.insets = new Insets(0, 0, 0, 0);  date\_select\_constraint.gridx = 6;  date\_select\_constraint.gridy = 1;  date\_select\_layout.setConstraints(date1\_label4, date\_select\_constraint);  date\_select.add(date1\_label4);    JPanel check\_in\_information = new JPanel();  GridBagLayout check\_in\_information\_layout = new GridBagLayout();  GridBagConstraints check\_in\_information\_constraint = new GridBagConstraints();  check\_in\_information.setLayout(check\_in\_information\_layout);   JLabel id\_label = new JLabel("身份证号：");  id\_label.setFont(new Font("宋体", Font.BOLD, 16));  check\_in\_information\_constraint.insets = new Insets(0, 0, 0, 0);  check\_in\_information\_constraint.gridx = 0;  check\_in\_information\_constraint.gridy = 0;  check\_in\_information\_layout.setConstraints(id\_label, check\_in\_information\_constraint);  check\_in\_information.add(id\_label);   JTextField id = new JTextField(30);  check\_in\_information\_constraint.insets = new Insets(0, 0, 0, 0);  check\_in\_information\_constraint.gridx = 1;  check\_in\_information\_constraint.gridy = 0;  check\_in\_information\_layout.setConstraints(id, check\_in\_information\_constraint);  check\_in\_information.add(id);   JLabel name\_label = new JLabel("姓名：");  name\_label.setFont(new Font("宋体", Font.BOLD, 16));  check\_in\_information\_constraint.insets = new Insets(0, 0, 0, 0);  check\_in\_information\_constraint.gridx = 0;  check\_in\_information\_constraint.gridy = 1;  check\_in\_information\_layout.setConstraints(name\_label, check\_in\_information\_constraint);  check\_in\_information.add(name\_label);   JTextField name = new JTextField(30);  check\_in\_information\_constraint.insets = new Insets(0, 0, 0, 0);  check\_in\_information\_constraint.gridx = 1;  check\_in\_information\_constraint.gridy = 1;  check\_in\_information\_layout.setConstraints(name, check\_in\_information\_constraint);  check\_in\_information.add(name);   JLabel phone\_label = new JLabel("手机号：");  phone\_label.setFont(new Font("宋体", Font.BOLD, 16));  check\_in\_information\_constraint.insets = new Insets(0, 0, 0, 0);  check\_in\_information\_constraint.gridx = 0;  check\_in\_information\_constraint.gridy = 2;  check\_in\_information\_layout.setConstraints(phone\_label, check\_in\_information\_constraint);  check\_in\_information.add(phone\_label);   JTextField phone = new JTextField(30);  check\_in\_information\_constraint.insets = new Insets(0, 0, 0, 0);  check\_in\_information\_constraint.gridx = 1;  check\_in\_information\_constraint.gridy = 2;  check\_in\_information\_layout.setConstraints(phone, check\_in\_information\_constraint);  check\_in\_information.add(phone);    JPanel room\_bar = new JPanel();  GridBagLayout room\_bar\_layout = new GridBagLayout();  GridBagConstraints room\_bar\_constraint = new GridBagConstraints();  room\_bar.setLayout(room\_bar\_layout);    JLabel room\_type\_label1 = new JLabel("房型:");  room\_type\_label1.setFont(new Font("宋体", Font.BOLD, 16));  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridheight = 1;  room\_bar\_constraint.gridwidth = 1;  room\_bar\_constraint.gridx = 0;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type\_label1, room\_bar\_constraint);  room\_bar.add(room\_type\_label1);   JComboBox<String> room\_type = new JComboBox<>();  room\_type.addItem("-请选择-");   try {  String[] datas = select.getAllRoomType();  assert datas != null;  for (String data : datas) {  room\_type.addItem(data);  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridx = 1;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type, room\_bar\_constraint);  room\_bar.add(room\_type);    JLabel room\_id\_label1 = new JLabel("房号:");  room\_id\_label1.setFont(new Font("宋体", Font.BOLD, 16));  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridheight = 1;  room\_bar\_constraint.gridwidth = 1;  room\_bar\_constraint.gridx = 0;  room\_bar\_constraint.gridy = 1;  room\_bar\_layout.setConstraints(room\_id\_label1, room\_bar\_constraint);  room\_bar.add(room\_id\_label1);   JComboBox<String> room\_id = new JComboBox<>();  room\_id.addItem("-请选择-");   room\_type.addItemListener(new ItemListener() {  @Override  public void itemStateChanged(ItemEvent e) {  try {  if (room\_type.getSelectedItem() == null) {  itemStateChanged(e);  }  String[] datas1 = select.getAllRoomNum((String) room\_type.getSelectedItem());  room\_id.removeAllItems();   for (String data : datas1) {  room\_id.addItem(data);  }  } catch (SQLException e1) {  throw new RuntimeException(e1);  }  }  });  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridx = 1;  room\_bar\_constraint.gridy = 1;  room\_bar\_layout.setConstraints(room\_id, room\_bar\_constraint);  room\_bar.add(room\_id);    JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnY = new JButton("确认");  btnY.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnY, btn\_bar\_constraint);  btn\_bar.add(btnY);  JButton btnN = new JButton("取消");  btnN.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnN, btn\_bar\_constraint);  btn\_bar.add(btnN);    btnY.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String sid\_card = id.getText();  String sname = name.getText();  String sphone = phone.getText();  int stay = -1;  String syear = null;  String smon = null;  String sday = null;  String syear1 = null;  String smon1 = null;  String sday1 = null;  try {  syear = (String) year.getSelectedItem();  smon = (String) mon.getSelectedItem();  sday = (String) day.getSelectedItem();  syear1 = (String) year1.getSelectedItem();  smon1 = (String) mon1.getSelectedItem();  sday1 = (String) day1.getSelectedItem();  stay = select.getDateDiff(syear + "-" + smon + "-" + sday, syear1 + "-" + smon1 + "-" + sday1);  if (stay <= 0) {  JOptionPane.showMessageDialog(null, "时间填写有误！", "提示", JOptionPane.ERROR\_MESSAGE);  return;  }  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "时间填写有误！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  String sid = (String) room\_id.getSelectedItem();  float[] fprice;   try {  fprice = select.getRoomPrice(sid);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  if (fprice[0] <= 0 || fprice[1] <= 0) {  JOptionPane.showMessageDialog(null, "房间价格有误！", "提示", JOptionPane.ERROR\_MESSAGE);  return;  }   int opt = JOptionPane.showConfirmDialog(null, "您需要支付" + String.format("%.2f", fprice[0] \* stay) + "元", "支付确认", JOptionPane.OK\_CANCEL\_OPTION);  if (opt == JOptionPane.YES\_OPTION) {  try {  int order\_id = insert.addOrder(syear + "-" + smon + "-" + sday, stay, fprice[0] \* stay, fprice[0] \* stay, fprice[1] \* stay, sid\_card, sname, sphone);  String room\_id = select.getRoomId(sid);  insert.addRealtion(String.valueOf(curuser.getId()), room\_id, String.valueOf(order\_id), "有效");  JOptionPane.showConfirmDialog(null, "支付成功", "提示", JOptionPane.DEFAULT\_OPTION);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "信息填写有误！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  }  });    btnN.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });  constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(date\_select, constraint);  this.add(date\_select);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(check\_in\_information, constraint);  this.add(check\_in\_information);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 3;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 2;  layout.setConstraints(room\_bar, constraint);  this.add(room\_bar);  }   public static void create() {  ReservationForOther reservationForOther = new ReservationForOther();  reservationForOther.setSize(600, 350);  reservationForOther.setTitle("预定办理");  reservationForOther.setLocationRelativeTo(null);  reservationForOther.setVisible(true);  } } |

|  |
| --- |
| RoomDetil.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.room; import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.sql.update;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class RoomDetil extends JFrame {  public RoomDetil(int idx) {  this.setSize(350, 280);  this.setLocationRelativeTo(null);   GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel room\_inf = new JPanel();  GridBagLayout room\_inf\_layout = new GridBagLayout();  GridBagConstraints room\_inf\_constraint = new GridBagConstraints();  room\_inf.setLayout(room\_inf\_layout);   JLabel r\_id\_label = new JLabel("房间id：");  r\_id\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 0;  room\_inf\_layout.setConstraints(r\_id\_label, room\_inf\_constraint);  room\_inf.add(r\_id\_label);   JLabel r\_id = new JLabel();  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 0;  room\_inf\_layout.setConstraints(r\_id, room\_inf\_constraint);  room\_inf.add(r\_id);   JLabel num\_label = new JLabel("房间号：");  num\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 1;  room\_inf\_layout.setConstraints(num\_label, room\_inf\_constraint);  room\_inf.add(num\_label);   JTextField num = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 1;  room\_inf\_layout.setConstraints(num, room\_inf\_constraint);  room\_inf.add(num);  // JLabel type\_label = new JLabel("房间类型"); // type\_label.setFont(new Font("宋体", Font.BOLD, 16)); // room\_inf\_constraint.insets = new Insets(0, 0, 0, 0); // room\_inf\_constraint.gridx = 0; // room\_inf\_constraint.gridy = 2; // room\_inf\_layout.setConstraints(type\_label, room\_inf\_constraint); // room\_inf.add(type\_label); // // JTextField type = new JTextField(20); // room\_inf\_constraint.insets = new Insets(0, 0, 0, 0); // room\_inf\_constraint.gridx = 1; // room\_inf\_constraint.gridy = 2; // room\_inf\_layout.setConstraints(room\_bar, room\_inf\_constraint); // room\_inf.add(room\_bar);  JPanel room\_bar = new JPanel();  GridBagLayout room\_bar\_layout = new GridBagLayout();  GridBagConstraints room\_bar\_constraint = new GridBagConstraints();  room\_bar.setLayout(room\_bar\_layout);   JLabel room\_type\_label1 = new JLabel("房型:");  room\_type\_label1.setFont(new Font("宋体", Font.BOLD, 16));  room\_bar\_constraint.insets = new Insets(0, 0, 0, 0);  room\_bar\_constraint.gridheight = 1;  room\_bar\_constraint.gridwidth = 1;  room\_bar\_constraint.gridx = 0;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type\_label1, room\_bar\_constraint);  room\_bar.add(room\_type\_label1);   JComboBox<String> room\_type = new JComboBox<>();  room\_type.addItem("-请选择-");  try {  String[] datas = select.getAllRoomType();  assert datas != null;  for (String data : datas) {  room\_type.addItem(data);  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  room\_bar\_constraint.insets = new Insets(0, 30, 0, 0);  room\_bar\_constraint.gridx = 1;  room\_bar\_constraint.gridy = 0;  room\_bar\_layout.setConstraints(room\_type, room\_bar\_constraint);  room\_bar.add(room\_type);  // JTextField type = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, -175, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 2;  room\_inf\_layout.setConstraints(room\_bar, room\_inf\_constraint);  room\_inf.add(room\_bar);   JLabel prepay\_label = new JLabel("预付款：");  prepay\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 3;  room\_inf\_layout.setConstraints(prepay\_label, room\_inf\_constraint);  room\_inf.add(prepay\_label);   JTextField prepay = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 3;  room\_inf\_layout.setConstraints(prepay, room\_inf\_constraint);  room\_inf.add(prepay);   JLabel money\_label = new JLabel("租金：");  money\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 5;  room\_inf\_layout.setConstraints(money\_label, room\_inf\_constraint);  room\_inf.add(money\_label);   JTextField money = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 5;  room\_inf\_layout.setConstraints(money, room\_inf\_constraint);  room\_inf.add(money);   JLabel status\_label = new JLabel("状态：");  status\_label.setFont(new Font("宋体", Font.BOLD, 16));  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 0;  room\_inf\_constraint.gridy = 6;  room\_inf\_layout.setConstraints(status\_label, room\_inf\_constraint);  room\_inf.add(status\_label);    JTextField status = new JTextField(20);  room\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  room\_inf\_constraint.gridx = 1;  room\_inf\_constraint.gridy = 6;  room\_inf\_layout.setConstraints(status, room\_inf\_constraint);  room\_inf.add(status);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("修改");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("返回");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);    constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(room\_inf, constraint);  this.add(room\_inf);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 3;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);  room ro;  try {  ro = select.getRoom(idx);  } catch (SQLException e) {  throw new RuntimeException(e);  }   r\_id.setText(String.valueOf(ro.getId()));  num.setText(ro.getNum());  room\_type.setSelectedItem(ro.getType());  prepay.setText(String.valueOf(ro.getPrepay()));  money.setText(String.valueOf(ro.getMoney()));  status.setText(ro.getStatus());    btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String snum = num.getText();  float sprepay = Float.parseFloat(prepay.getText());  float smoney = Float.parseFloat(money.getText());  String sroomtype = (String) room\_type.getSelectedItem();  String spriceid = null;  try {  sroomtype = select.getTypeId((String) room\_type.getSelectedItem());  spriceid = select.getPriceId(sprepay, smoney);  if (spriceid == null) {  int priceid = insert.addPrice(sprepay, smoney);  spriceid = String.valueOf(priceid);  }  update.updateRoom(String.valueOf(ro.getId()), snum, sroomtype, spriceid);  JOptionPane.showMessageDialog(null, "修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "修改失败！", "提示", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });   btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });   }  } |

|  |
| --- |
| RoomManagent.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.ui.component.RoomTable;  import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class RoomManagent extends JPanel {  public RoomManagent() throws SQLException {  System.out.println(curuser.getId());  RoomTable roomTable = new RoomTable();   setLayout(new BorderLayout());   JPanel jPanel = new JPanel();   JLabel jl1 = new JLabel("按住CTRL可多选");  JButton add = new JButton("添加");  JButton look = new JButton("管理");  JButton reload = new JButton("刷新");  JButton btnAddType = new JButton("新增房型");   JTable myTable = roomTable.getTable();  DefaultTableModel model = (DefaultTableModel) myTable.getModel();  roomTable.setViewportView(myTable);  String[] hrader = new String[]{"房间id", "房间号", "房间类型", "预付款金额", "租金金额", "状态"};  model.setDataVector(select.getAllRoom(), hrader);   look.addActionListener(new ActionListener() {//查看事件  public void actionPerformed(ActionEvent e) {  int selectedRow = myTable.getSelectedRow();//获得选中行的索引  int selectedRows = myTable.getSelectedRows().length;   if (selectedRows > 1) {  JOptionPane.showMessageDialog(null, "每次只能查看一个！", "提示",  JOptionPane.INFORMATION\_MESSAGE);  return;  }  int id = Integer.parseInt((String) myTable.getValueAt(selectedRow, 0));  RoomDetil r = new RoomDetil(id);  r.setVisible(true);  }  });   reload.addActionListener(new ActionListener() {//刷新事件  public void actionPerformed(ActionEvent e) {  try {  model.setDataVector(select.getAllRoom(), hrader);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  myTable.updateUI();  }  });   add.addActionListener(new ActionListener() {//刷新事件  public void actionPerformed(ActionEvent e) {  AddRoom.create();  }  });   jPanel.add(jl1);  jPanel.add(add);  jPanel.add(look);  jPanel.add(reload);  add(jPanel, BorderLayout.NORTH);  add(roomTable, BorderLayout.CENTER);  } } |

|  |
| --- |
| UserCenter.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.sql.update; import cn.edu.sut.hotel\_management.util.ImageUtil;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.io.IOException; import java.sql.SQLException;  public class UserCenter extends JPanel {  public UserCenter() {  GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel img\_bar = new JPanel();  GridBagLayout img\_bar\_layout = new GridBagLayout();  GridBagConstraints img\_bar\_constraint = new GridBagConstraints();  img\_bar.setLayout(img\_bar\_layout);   JLabel image = new JLabel(" 暂无照片 ");  image.setMinimumSize(new Dimension(150, 200));  image.setMaximumSize(new Dimension(150, 200));  image.setPreferredSize(new Dimension(150, 200));  img\_bar\_constraint.gridheight = 5;  img\_bar\_constraint.gridwidth = 3;  img\_bar\_constraint.gridx = 0;  img\_bar\_constraint.gridy = 1;  img\_bar\_layout.setConstraints(image, img\_bar\_constraint);  img\_bar.add(image);   JPanel user\_inf = new JPanel();  GridBagLayout user\_inf\_layout = new GridBagLayout();  GridBagConstraints user\_inf\_constraint = new GridBagConstraints();  user\_inf.setLayout(user\_inf\_layout);   JLabel user\_id\_label = new JLabel("用户唯一id：");  user\_id\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 0;  user\_inf\_layout.setConstraints(user\_id\_label, user\_inf\_constraint);  user\_inf.add(user\_id\_label);   JLabel user\_id = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 0;  user\_inf\_layout.setConstraints(user\_id, user\_inf\_constraint);  user\_inf.add(user\_id);   JLabel order\_num\_label = new JLabel("订单数量:");  order\_num\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 1;  user\_inf\_layout.setConstraints(order\_num\_label, user\_inf\_constraint);  user\_inf.add(order\_num\_label);   JLabel order\_num = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 1;  user\_inf\_layout.setConstraints(order\_num, user\_inf\_constraint);  user\_inf.add(order\_num);   JLabel username\_label = new JLabel("用户名：");  username\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 2;  user\_inf\_layout.setConstraints(username\_label, user\_inf\_constraint);  user\_inf.add(username\_label);   JTextField username = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 2;  user\_inf\_layout.setConstraints(username, user\_inf\_constraint);  user\_inf.add(username);   JLabel id\_card\_label = new JLabel("身份证号：");  id\_card\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 3;  user\_inf\_layout.setConstraints(id\_card\_label, user\_inf\_constraint);  user\_inf.add(id\_card\_label);   JTextField id\_card = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 3;  user\_inf\_layout.setConstraints(id\_card, user\_inf\_constraint);  user\_inf.add(id\_card);   JLabel name\_label = new JLabel("姓名：");  name\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 4;  user\_inf\_layout.setConstraints(name\_label, user\_inf\_constraint);  user\_inf.add(name\_label);   JTextField name = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 4;  user\_inf\_layout.setConstraints(name, user\_inf\_constraint);  user\_inf.add(name);   JLabel phone\_label = new JLabel("手机号：");  phone\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 5;  user\_inf\_layout.setConstraints(phone\_label, user\_inf\_constraint);  user\_inf.add(phone\_label);   JTextField phone = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 5;  user\_inf\_layout.setConstraints(phone, user\_inf\_constraint);  user\_inf.add(phone);    JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton upload = new JButton("上传照片");  upload.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(upload, btn\_bar\_constraint);  btn\_bar.add(upload);   JButton btnA = new JButton("修改密码");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);   JButton btnB = new JButton("确认修改");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 2;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);   JButton btnC = new JButton("重置输入");  btnC.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 3;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnC, btn\_bar\_constraint);  btn\_bar.add(btnC);   JButton btnD = new JButton("退出登录");  btnD.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 4;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnD, btn\_bar\_constraint);  btn\_bar.add(btnD);   btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  ChangePassword.create();  }  });  btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String sid\_card = id\_card.getText();  String sname = name.getText();  String stel = phone.getText();  String suname = username.getText();  try {  update.updateCurUser(curuser.getId(), sname, sid\_card, stel, suname);  JOptionPane.showConfirmDialog(null, "修改成功", "提示", JOptionPane.DEFAULT\_OPTION);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "修改失败！", "错误", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });   btnC.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  username.setText(curuser.getUser\_name());  id\_card.setText(curuser.getId\_card());  name.setText(curuser.getName());  phone.setText(curuser.getTel());  }  });   btnD.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  curuser.logout();  SwingUtilities.getWindowAncestor((Component) e.getSource()).dispose();  Login l = new Login();  l.setVisible(true);  }  });   upload.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  ImageUtil.addPicture(image, upload);   }  });   user\_id.setText(String.valueOf(curuser.getId()));  username.setText(curuser.getUser\_name());  id\_card.setText(curuser.getId\_card());  name.setText(curuser.getName());  phone.setText(curuser.getTel());  try {  order\_num.setText(select.getOrderNum());  } catch (SQLException e) {  throw new RuntimeException(e);  }  try {  ImageUtil.loadPicture(image);  } catch (IOException e) {  e.printStackTrace();  }   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 0;  layout.setConstraints(user\_inf, constraint);  this.add(img\_bar);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 1;  constraint.gridy = 0;  layout.setConstraints(user\_inf, constraint);  this.add(user\_inf);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridwidth = 2;  constraint.gridy = 1;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);  }   public static void create() {  Register register = new Register();  register.setSize(400, 300);  register.setTitle("注册");  register.setLocationRelativeTo(null);  register.setVisible(true);  } } |

|  |
| --- |
| UserDetil.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.user; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.sql.update;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class UserDetil extends JFrame {  public UserDetil(String idx) {  this.setSize(350, 280);  this.setLocationRelativeTo(null);   GridBagLayout layout = new GridBagLayout();  GridBagConstraints constraint = new GridBagConstraints();  this.setLayout(layout);   JPanel user\_inf = new JPanel();  GridBagLayout user\_inf\_layout = new GridBagLayout();  GridBagConstraints user\_inf\_constraint = new GridBagConstraints();  user\_inf.setLayout(user\_inf\_layout);   JLabel u\_id\_label = new JLabel("用户id：");  u\_id\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 0;  user\_inf\_layout.setConstraints(u\_id\_label, user\_inf\_constraint);  user\_inf.add(u\_id\_label);   JLabel u\_id = new JLabel();  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 0;  user\_inf\_layout.setConstraints(u\_id, user\_inf\_constraint);  user\_inf.add(u\_id);   JLabel account\_label = new JLabel("账号：");  account\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 1;  user\_inf\_layout.setConstraints(account\_label, user\_inf\_constraint);  user\_inf.add(account\_label);   JLabel account = new JLabel("");  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 1;  user\_inf\_layout.setConstraints(account, user\_inf\_constraint);  user\_inf.add(account);   JLabel u\_name\_label = new JLabel("用户名：");  u\_name\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 2;  user\_inf\_layout.setConstraints(u\_name\_label, user\_inf\_constraint);  user\_inf.add(u\_name\_label);   JTextField u\_name = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 2;  user\_inf\_layout.setConstraints(u\_name, user\_inf\_constraint);  user\_inf.add(u\_name);   JLabel name\_label1 = new JLabel("姓名:");  name\_label1.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 3;  user\_inf\_layout.setConstraints(name\_label1, user\_inf\_constraint);  user\_inf.add(name\_label1);   JTextField name = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 3;  user\_inf\_layout.setConstraints(name, user\_inf\_constraint);  user\_inf.add(name);   JLabel phone\_label = new JLabel("手机号：");  phone\_label.setFont(new Font("宋体", Font.BOLD, 16));  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 0;  user\_inf\_constraint.gridy = 4;  user\_inf\_layout.setConstraints(phone\_label, user\_inf\_constraint);  user\_inf.add(phone\_label);   JTextField phone = new JTextField(20);  user\_inf\_constraint.insets = new Insets(0, 0, 0, 0);  user\_inf\_constraint.gridx = 1;  user\_inf\_constraint.gridy = 4;  user\_inf\_layout.setConstraints(phone, user\_inf\_constraint);  user\_inf.add(phone);   JPanel btn\_bar = new JPanel();  GridBagLayout btn\_bar\_layout = new GridBagLayout();  GridBagConstraints btn\_bar\_constraint = new GridBagConstraints();  btn\_bar.setLayout(btn\_bar\_layout);   JButton btnA = new JButton("修改");  btnA.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 0;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnA, btn\_bar\_constraint);  btn\_bar.add(btnA);  JButton btnB = new JButton("返回");  btnB.setFont(new Font("宋体", Font.BOLD, 16));  btn\_bar\_constraint.gridx = 1;  btn\_bar\_constraint.gridy = 0;  btn\_bar\_layout.setConstraints(btnB, btn\_bar\_constraint);  btn\_bar.add(btnB);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 1;  layout.setConstraints(user\_inf, constraint);  this.add(user\_inf);   constraint.insets = new Insets(0, 0, 0, 0);  constraint.gridx = 0;  constraint.gridy = 3;  layout.setConstraints(btn\_bar, constraint);  this.add(btn\_bar);  user u;  try {  u = select.getUser(Long.parseLong(idx));  } catch (SQLException e) {  throw new RuntimeException(e);  }   u\_id.setText(String.valueOf(u.getId()));  u\_name.setText(u.getUser\_name());  name.setText(u.getName());  phone.setText(String.valueOf(u.getTel()));  account.setText(String.valueOf(u.getAccount()));   btnA.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  String sname = name.getText();  String su\_name = u\_name.getText();  String sphone = (phone.getText());  try {  update.updateUser(Long.parseLong(idx), sname, sphone, su\_name);  JOptionPane.showMessageDialog(null, "修改成功！", "提示", JOptionPane.INFORMATION\_MESSAGE);  } catch (SQLException ex) {  JOptionPane.showMessageDialog(null, "修改失败！", "错误", JOptionPane.ERROR\_MESSAGE);  throw new RuntimeException(ex);  }  }  });   btnB.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  dispose();  }  });  } } |

|  |
| --- |
| UserManagent.java |
| package cn.edu.sut.hotel\_management.ui;  import cn.edu.sut.hotel\_management.model.curuser; import cn.edu.sut.hotel\_management.sql.delete; import cn.edu.sut.hotel\_management.sql.select; import cn.edu.sut.hotel\_management.sql.update; import cn.edu.sut.hotel\_management.ui.component.UserTable;  import javax.swing.\*; import javax.swing.table.DefaultTableModel; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.SQLException;  public class UserManagent extends JPanel {  public UserManagent() throws SQLException {  System.out.println(curuser.getId());  UserTable userTable = new UserTable();   setLayout(new BorderLayout());   JPanel jPanel = new JPanel();   JLabel jl1 = new JLabel("按住CTRL可多选");  JButton look = new JButton("管理");  JButton reload = new JButton("刷新");  JButton setadmin = new JButton("设为/取消管理员");  JButton deleteU = new JButton("删除");   JTable myTable = userTable.getTable();  DefaultTableModel model = (DefaultTableModel) myTable.getModel();  userTable.setViewportView(myTable);  String[] hrader = new String[]{"用户id", "账号", "用户名", "姓名", "手机号", "是否管理员"};  model.setDataVector(select.getAllUser(), hrader);   look.addActionListener(new ActionListener() {//查看事件  public void actionPerformed(ActionEvent e) {   int selectedRow = myTable.getSelectedRow();//获得选中行的索引  int selectedRows = myTable.getSelectedRows().length;   if (selectedRows > 1) {  JOptionPane.showMessageDialog(null, "每次只能管理一个！", "提示",  JOptionPane.INFORMATION\_MESSAGE);  return;  }  String id = ((String) myTable.getValueAt(selectedRow, 0));  UserDetil r = new UserDetil(id);  r.setVisible(true);  }  });   reload.addActionListener(new ActionListener() {//刷新事件  public void actionPerformed(ActionEvent e) {  try {  model.setDataVector(select.getAllUser(), hrader);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  myTable.updateUI();  }  });   setadmin.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  int[] selectedRows = myTable.getSelectedRows();  for (int i : selectedRows) {  if (i != -1) //存在选中行  {  String id = (String) myTable.getValueAt(i, 0);  try {  boolean status = select.isAdmin(id);  if (status) {  update.unsetAdmin(id);  } else {  update.setAdmin(id);  }  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  }  }  try {  model.setDataVector(select.getAllUser(), hrader);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  myTable.updateUI();   }  });  deleteU.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  int[] selectedRows = myTable.getSelectedRows();  for (int i : selectedRows) {  if (i != -1) //存在选中行  {  String id = (String) myTable.getValueAt(i, 0);  try {  delete.deleteUser(id);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  }  }  try {  model.setDataVector(select.getAllUser(), hrader);  } catch (SQLException ex) {  throw new RuntimeException(ex);  }  myTable.updateUI();  }  });   jPanel.add(jl1);  jPanel.add(look);  jPanel.add(reload);  jPanel.add(setadmin);  jPanel.add(deleteU);  add(jPanel, BorderLayout.NORTH);  add(userTable, BorderLayout.CENTER);  } } |

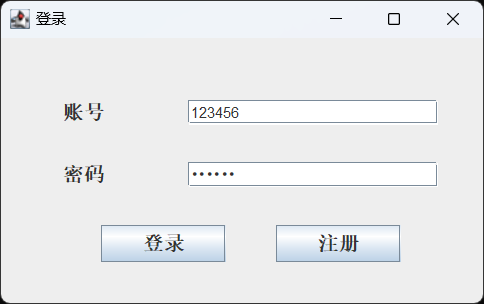
|  |
| --- |
| Api.java |
| package cn.edu.sut.hotel\_management.util;  import javax.swing.\*; import java.sql.\*;  public class Api {   public ResultSet select(PreparedStatement sql) {  try {  return sql.executeQuery();  } catch (SQLException e) {  throw new RuntimeException(e);  }  }    public int insert(PreparedStatement sql) {  try {  return sql.executeUpdate();  } catch (SQLException e) {  throw new RuntimeException(e);  }  }    public int update(PreparedStatement sql) {  try {  return sql.executeUpdate();  } catch (SQLException e) {  throw new RuntimeException(e);  }  }    public int delete(PreparedStatement sql) {  try {  return sql.executeUpdate();  } catch (SQLException e) {  throw new RuntimeException(e);  }  }   public Connection getConnection() {  return JDBC.getConnection();  }   public PreparedStatement getPreparedStatement(String sql) {  try {  Connection connection = JDBC.getConnection();  return connection.prepareStatement(sql, ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);  } catch (SQLException e) {  e.printStackTrace();  }  return null;  }   public PreparedStatement getPreparedStatementWithKey(String sql) {  try {  Connection connection = JDBC.getConnection();  return connection.prepareStatement(sql, Statement.RETURN\_GENERATED\_KEYS);  } catch (SQLException e) {  throw new RuntimeException(e);  }  } } |

|  |
| --- |
| dbconfig.java |
| package cn.edu.sut.hotel\_management.util;  public interface dbconfig {  String JDBC\_DRIVER = "com.mysql.cj.jdbc.Driver";  String JDBC\_URL = "jdbc:mysql://localhost:3306/hotel\_management?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC";  String JDBC\_USER = "root";  String JDBC\_PASSWORD = "root"; } |

|  |
| --- |
| ImageUtil.java |
| package cn.edu.sut.hotel\_management.util;  import cn.edu.sut.hotel\_management.sql.insert; import cn.edu.sut.hotel\_management.sql.select;  import javax.swing.\*; import javax.swing.filechooser.FileNameExtensionFilter; import java.awt.\*; import java.io.\*; import java.sql.SQLException;  public class ImageUtil {  public static ImageIcon createAutoAdjustIcon(java.awt.Image image, boolean constrained) {  ImageIcon icon = new ImageIcon(image) {  @Override  public synchronized void paintIcon(java.awt.Component cmp, Graphics g, int x, int y) {  //初始化参数  Point startPoint = new Point(0, 0);//默认绘制起点  Dimension cmpSize = cmp.getSize();//获取组件大小  Dimension imgSize = new Dimension(getIconWidth(), getIconHeight());//获取图像大小  //计算绘制起点和区域  if (constrained) {//等比例缩放  //计算图像宽高比例  double ratio = 1.0 \* imgSize.width / imgSize.height;  //计算等比例缩放后的区域大小  imgSize.width = (int) Math.min(cmpSize.width, ratio \* cmpSize.height);  imgSize.height = (int) (imgSize.width / ratio);  //计算绘制起点  startPoint.x = (int)  (cmp.getAlignmentX() \* (cmpSize.width - imgSize.width));  startPoint.y = (int)  (cmp.getAlignmentY() \* (cmpSize.height - imgSize.height));  } else {//完全填充  imgSize = cmpSize;  }  //根据起点和区域大小进行绘制  if (getImageObserver() == null) {  g.drawImage(getImage(), startPoint.x, startPoint.y,  imgSize.width, imgSize.height, cmp);  } else {  g.drawImage(getImage(), startPoint.x, startPoint.y,  imgSize.width, imgSize.height, getImageObserver());  }  }  };  return icon;  }   public static ImageIcon createAutoAdjustIcon(String filename, boolean constrained) {  return createAutoAdjustIcon(new ImageIcon(filename).getImage(), constrained);  }   public static ImageIcon createAutoAdjustIcon(byte[] img\_aray, boolean constrained) {  return createAutoAdjustIcon(new ImageIcon(img\_aray).getImage(), constrained);  }   public static int addPicture(JLabel lblNewLabel, JButton button) {  JFileChooser chooser = new JFileChooser();  chooser.setMultiSelectionEnabled(true);  FileNameExtensionFilter filter = new FileNameExtensionFilter("图像文件(jpg/png/gif)", "jpg", "jpeg", "png", "gif");  chooser.setFileFilter(filter);  int returnVal = chooser.showOpenDialog(button);  String absolutePath = null;  int idx = 0;  if (returnVal == JFileChooser.APPROVE\_OPTION) {  File[] arrfiles = chooser.getSelectedFiles();  if (arrfiles == null || arrfiles.length == 0) {  return -1;  }  File ff = chooser.getSelectedFile();  String fileName = ff.getName();  String prefix = fileName.substring(fileName.lastIndexOf(".") + 1);  if (!(prefix.equals("jpg") || prefix.equals("png") || prefix.equals("gif"))) {  JOptionPane.showMessageDialog(new JDialog(), ":请选择.jpg 或 .png 或 .gif 格式的图片");  return -1;  }  FileInputStream input = null;  try {  for (File f : arrfiles) {  absolutePath = chooser.getSelectedFile().getAbsolutePath();  ImageIcon imageIcon = ImageUtil.createAutoAdjustIcon(absolutePath, true);  lblNewLabel.setIcon(imageIcon);  idx = insert.addPicture(absolutePath);  input = new FileInputStream(f);  input.close();  }  JOptionPane.showMessageDialog(null, "上传成功！", "提示",  JOptionPane.INFORMATION\_MESSAGE);  } catch (FileNotFoundException e1) {  JOptionPane.showMessageDialog(null, "上传失败！", "提示",  JOptionPane.ERROR\_MESSAGE);  e1.printStackTrace();  } catch (IOException e1) {  JOptionPane.showMessageDialog(null, "上传失败！", "提示",  JOptionPane.ERROR\_MESSAGE);  e1.printStackTrace();  }  }  return idx;  }   public static boolean loadPicture(JLabel lblNewLabel) throws IOException {  try {  InputStream img = select.loadPicture();  if (img == null) {  return false;  }  byte[] img\_array = new byte[img.available()];  img.read(img\_array);  if (img\_array.length == 0) {  return false;  }  ImageIcon imageIcon = ImageUtil.createAutoAdjustIcon(img\_array, true);  lblNewLabel.setIcon(imageIcon);  return true;  } catch (SQLException e) {  throw new RuntimeException(e);  }  } } |

|  |
| --- |
| JDBC.java |
| package cn.edu.sut.hotel\_management.util;  import javax.swing.\*; import java.sql.Connection; import java.sql.DriverManager; import java.sql.SQLException;  public class JDBC {   static {  try {  Class.forName(dbconfig.JDBC\_DRIVER);  } catch (ClassNotFoundException e) {  e.printStackTrace();  }  }   public static Connection getConnection() {  try {  return DriverManager.getConnection(dbconfig.JDBC\_URL, dbconfig.JDBC\_USER, dbconfig.JDBC\_PASSWORD);  } catch (  SQLException e) {  JOptionPane.showMessageDialog(null, "数据库连接失败！", "提示", JOptionPane.ERROR\_MESSAGE); // throw new RuntimeException(e);  }  return null;  } } |

1. **运行结果**



图形用户界面

描述已自动生成

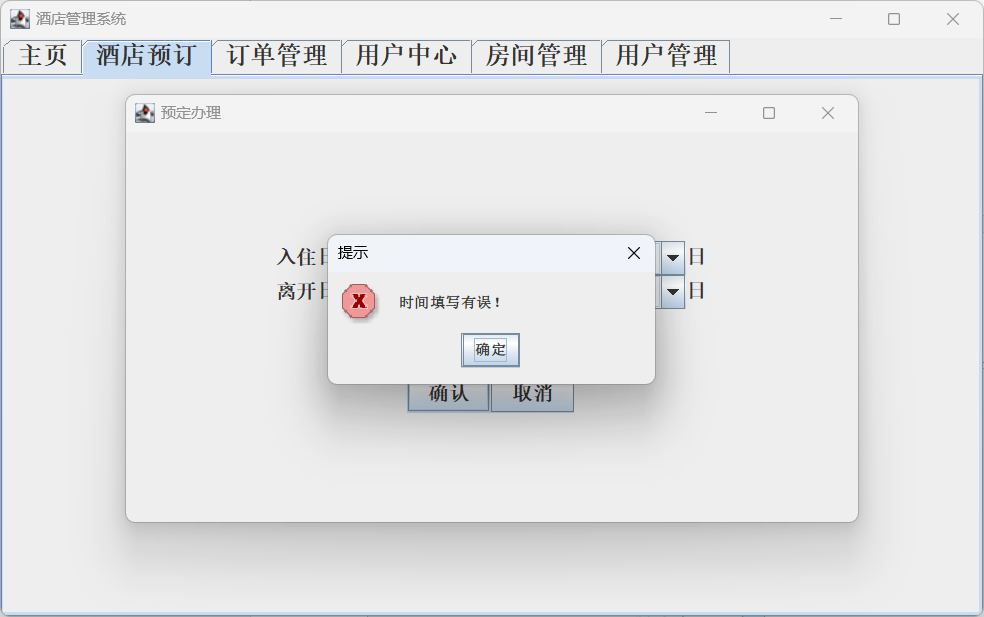
图形用户界面, 应用程序

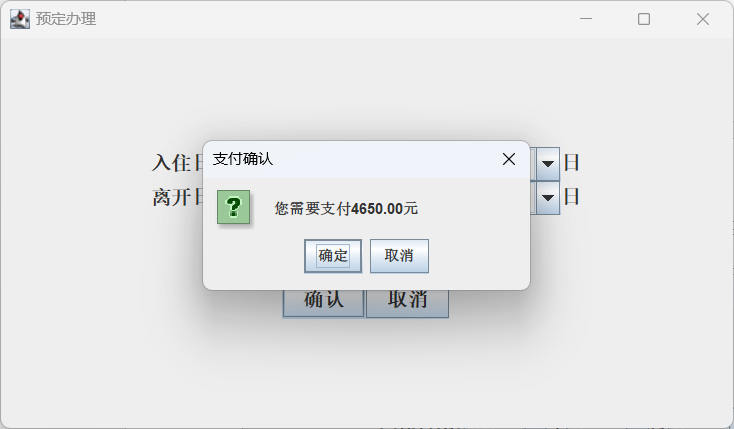
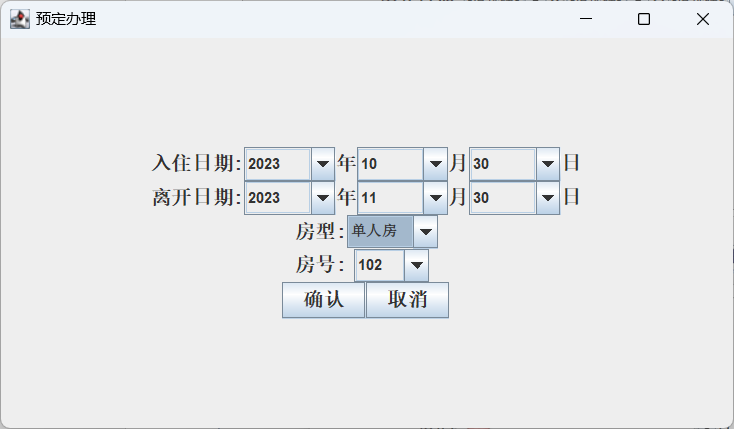
描述已自动生成

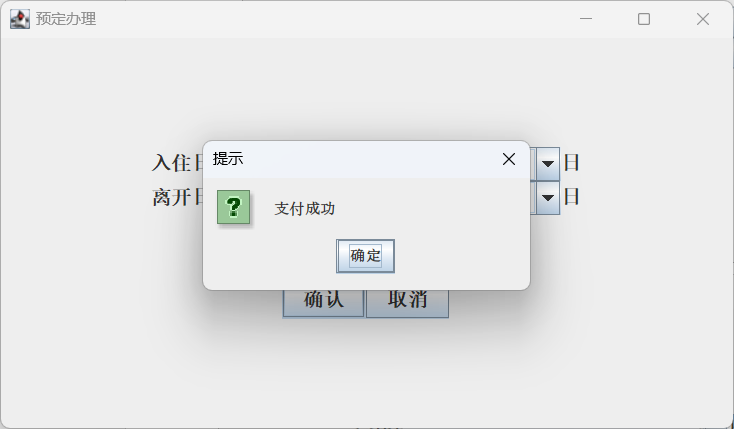
图形用户界面, 文本, 应用程序, 电子邮件

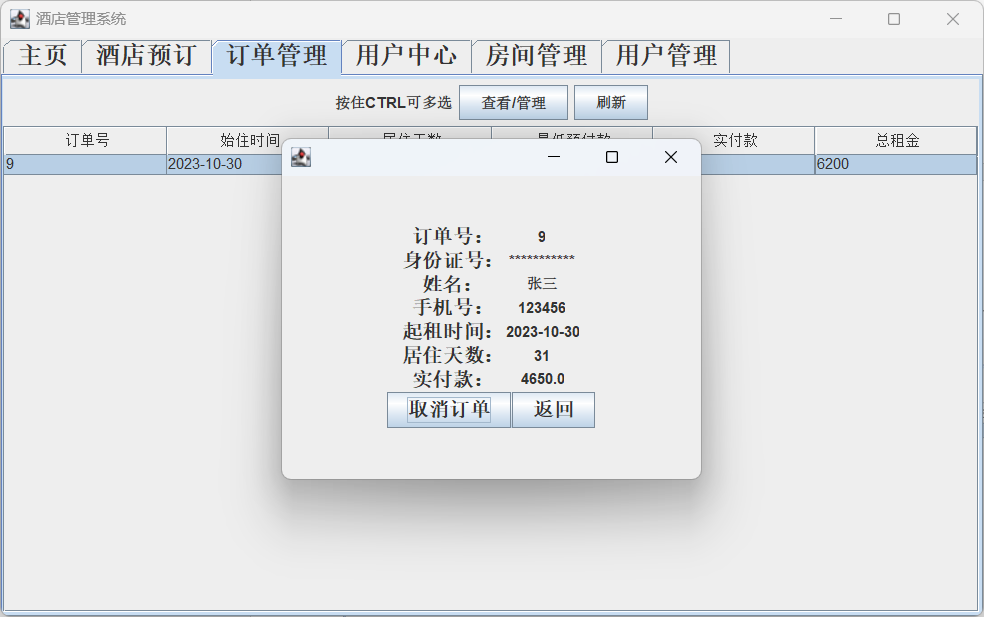
描述已自动生成

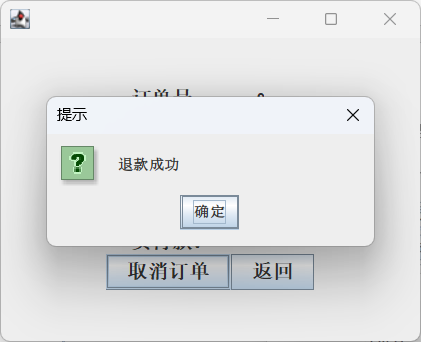
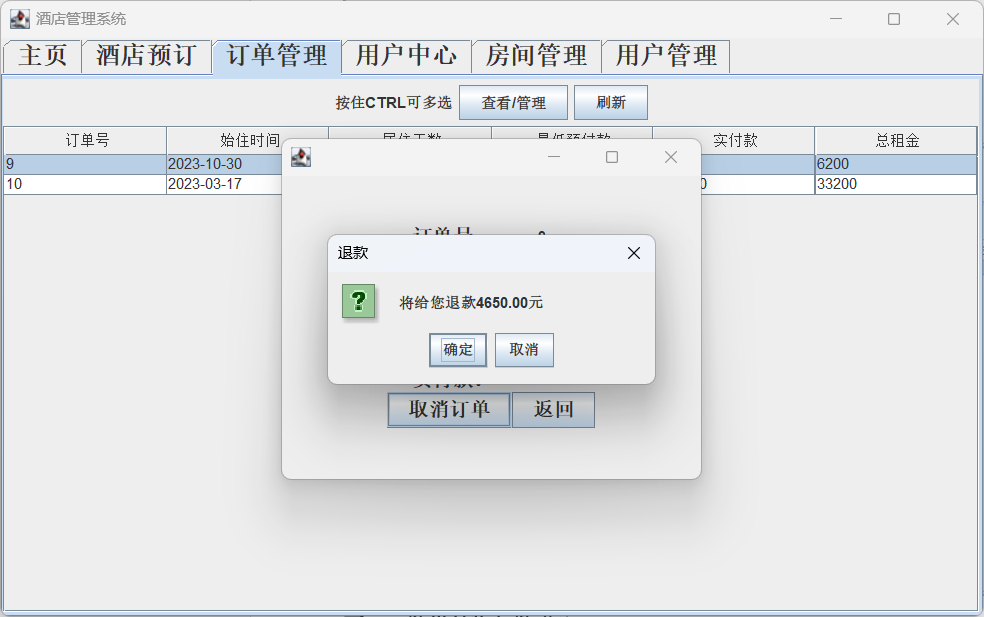
图形用户界面, 应用程序

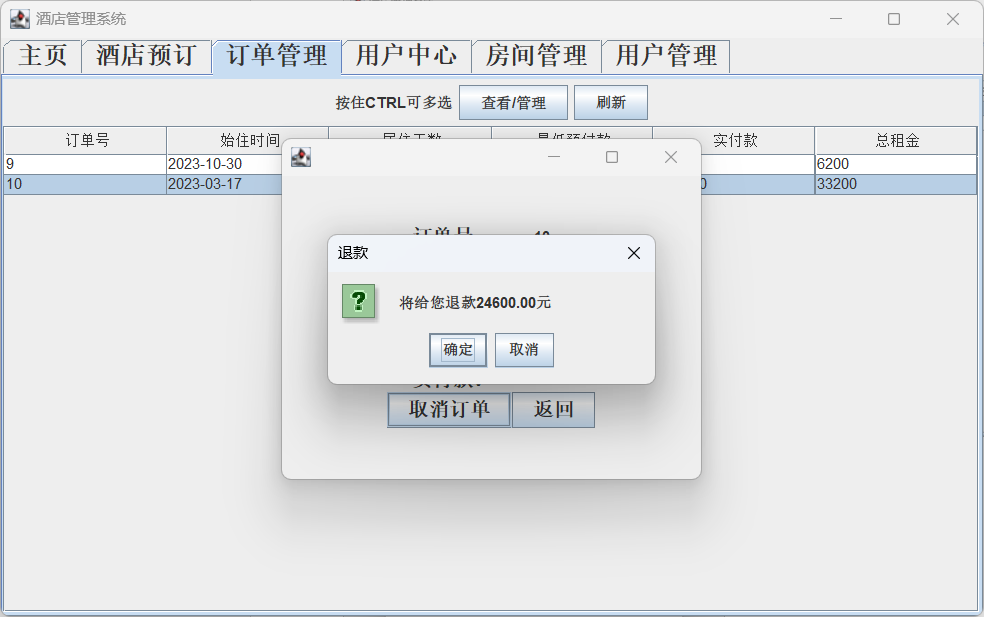
描述已自动生成









图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本

描述已自动生成图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本

描述已自动生成表格

描述已自动生成

图形用户界面

描述已自动生成图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序, Word

描述已自动生成

1. **结果分析与结论**

在本次实验中学到了许多关于java的知识，体验了工程化软件开发，学到了很多有用的知识