|  |
| --- |
| 北京工业大学 经管学院 |
| 火车票预订系统 |
| 颜雨祺18110218 王慧18110219 自来乐.白克力18110260 |

|  |
| --- |
| 颜雨祺 王慧 自来乐.白克力  2020/5/10 |

**小组信息和软件信息**

**小组成员：**颜雨祺18110218

王慧 18110219

自来乐.白克力 18110260

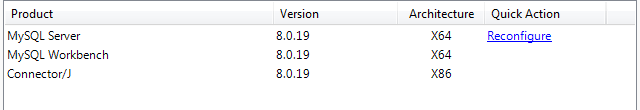
**小组分工：**

颜雨祺：报告——系统分析部分；整合报告；数据库设计和数据编写；主页和查询界面制作、程序编写；火车车次详情界面制作、程序编写；高铁动车车次详情界面制作、程序编写；订单详情界面制作、程序编写；订单详情1界面制作、程序编写；火车车次添加界面制作、程序编写；高铁动车车次添加界面制作、程序编写；修改密码界面制作、程序编写；系统错误修改与完善。

王慧：报告——需求分析部分；答辩视频；数据库设计和数据编写；登录界面制作、程序编写；乘客个人中心界面制作、程序编写；管理员界面制作、程序编写；退票界面制作、程序编写；火车车次添加程序编写；高铁动车车次添加程序编写；火车车次取消程序编写；高铁动车车次取消程序编写；系统错误修改与完善。

自来乐.白克力：报告——数据库设计部分；答辩PPT；数据库设计和数据编写；留言界面制作、程序编写；火车车次取消界面制作、程序编写；高铁动车车次取消界面制作、程序编写；查看乘客信息界面制作、程序编写；管理乘客密码界面制作、程序编写；删除乘客信息界面制作、程序编写；发布公告界面制作、程序编写；留言显示界面制作、程序编写；美化界面。

**开发工具及其版本说明：**



NetBeans IDE 8.1

目录

[一、系统需求分析 4](#_Toc40034859)

[1. 背景 4](#_Toc40034860)

[2. 功能 4](#_Toc40034861)

[3. 功能结构图 4](#_Toc40034862)

[二、数据库分析 5](#_Toc40034863)

[1. admin\_ 6](#_Toc40034864)

[2. hsorder 6](#_Toc40034865)

[3. hsrailway 6](#_Toc40034866)

[4. hsrailway\_price 7](#_Toc40034867)

[5. message 7](#_Toc40034868)

[6. notice 7](#_Toc40034869)

[7. passenger 8](#_Toc40034870)

[8. torder 8](#_Toc40034871)

[9. train 9](#_Toc40034872)

[10. train\_price 9](#_Toc40034873)

[三、系统实现 10](#_Toc40034874)

[1. 实体类的实现 10](#_Toc40034875)

[1.1 database类的实现 10](#_Toc40034876)

[1.2 admin\_类的实现 11](#_Toc40034877)

[1.3 hsorder类的实现 11](#_Toc40034878)

[1.4 hsrailway类的实现 14](#_Toc40034879)

[1.5 hsrailway\_price类的实现 16](#_Toc40034880)

[1.6 message类的实现 18](#_Toc40034881)

[1.7 notice类的实现 18](#_Toc40034882)

[1.8 passenger类的实现 19](#_Toc40034883)

[1.9 torder类的实现 21](#_Toc40034884)

[1.10 train类的实现 24](#_Toc40034885)

[1.11 train\_price类的实现 26](#_Toc40034886)

[2. control类的实现 28](#_Toc40034887)

[3. 表单对象类的实现 46](#_Toc40034888)

[4. 判断类的实现 46](#_Toc40034889)

[5. 边界类的实现 46](#_Toc40034890)

[5.1 登录界面 46](#_Toc40034891)

[5.2 主页和查询界面 49](#_Toc40034892)

[5.3 火车车次详情界面 50](#_Toc40034893)

[5.4 高铁动车车次详情界面 56](#_Toc40034894)

[5.5 乘客个人中心界面 60](#_Toc40034895)

[5.6 修改密码 62](#_Toc40034896)

[5.7 订单详情1（<60岁的乘客出现该界面） 64](#_Toc40034897)

[5.8 订单详情（>=60岁的乘客出现该界面） 66](#_Toc40034898)

[5.9 退票 66](#_Toc40034899)

[5.10 留言界面 71](#_Toc40034900)

[5.11 管理员界面 72](#_Toc40034901)

[5.12 火车车次添加界面 74](#_Toc40034902)

[5.13 高铁车次添加界面 77](#_Toc40034903)

[5.14 火车车次取消界面 77](#_Toc40034904)

[5.15 高铁车次取消界面 79](#_Toc40034905)

[5.16 查看乘客信息界面 80](#_Toc40034906)

[5.17 乘客密码管理（管理乘客密码界面） 82](#_Toc40034907)

[5.18 删除乘客信息界面 84](#_Toc40034908)

[5.19 发布公告 87](#_Toc40034909)

[5.20 留言管理（留言显示界面） 88](#_Toc40034910)

[四、课程设计体会 90](#_Toc40034911)

[颜雨祺 18110218： 90](#_Toc40034912)

[王慧 18110219： 90](#_Toc40034913)

[自来乐·白克力 18110260： 90](#_Toc40034914)

**火车票预订系统**

18110218颜雨祺、18110219王慧、18110260 自来乐.白克力

一、系统需求分析

1. 背景

在互联网高速发展的时代，网络预订火车票已经越来越常见。通过手机、电脑就可以轻松预定火车票，极大地方便了人们的生活。因此，我们小组决定开发设计一套火车票预订系统。

1. 功能

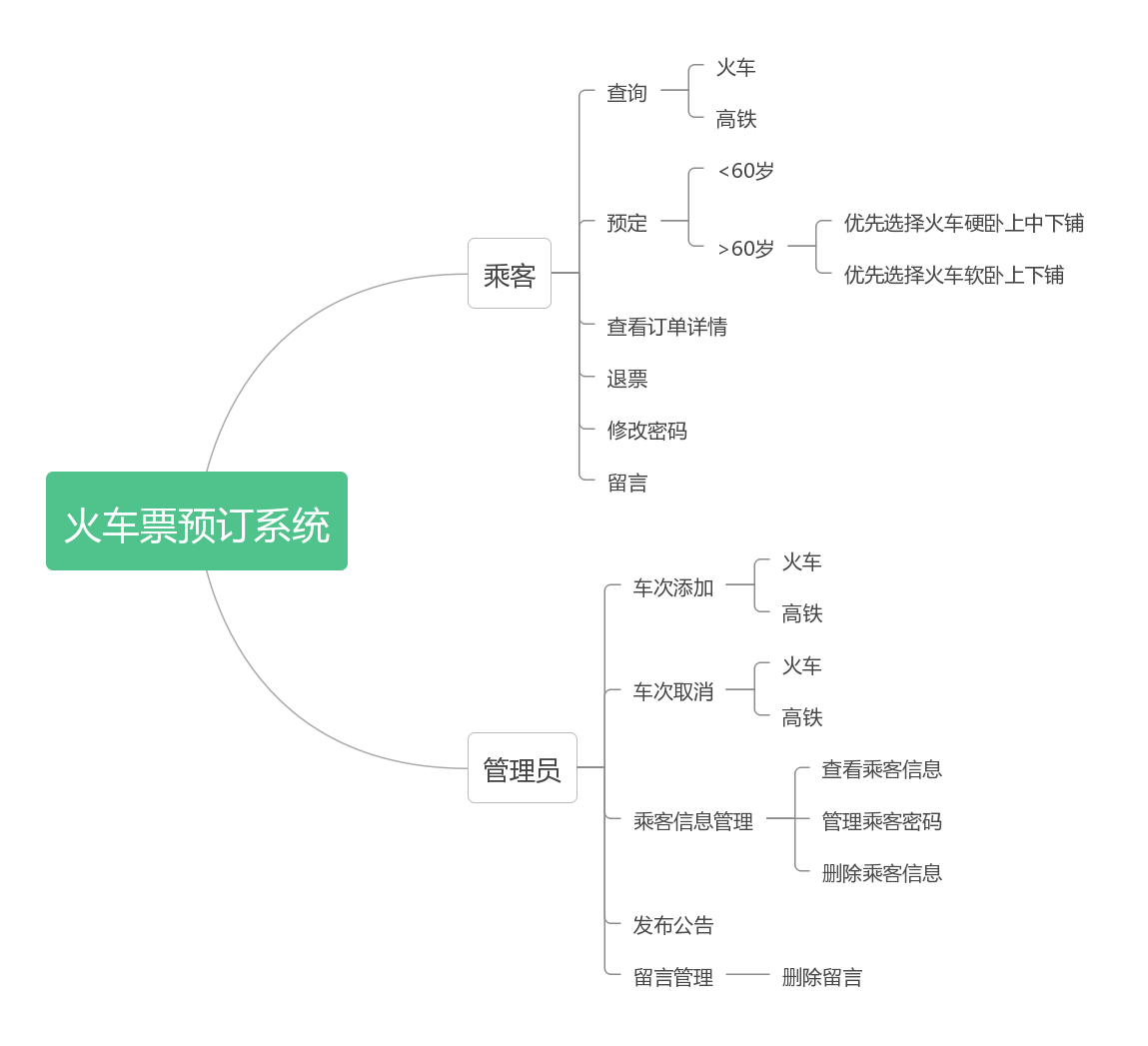
本系统主要功能为为乘客提供查询、预定、退票及老年人优先选择上下铺的服务。

从乘客角度看：

1. 注册：首次使用本系统需先进行注册
2. 登录：已有账号或已注册过的用户，可以进行登录
3. 查询：乘客可以根据出发日期、出发地点、到达地点、是否乘坐高铁动车进行查询
4. 预定：查询出符合要求的列车信息，点击方可预定，若年龄大于60岁的乘客，可以优先选择上下铺
5. 订单详情及退票：乘客预定后，可以进行订单查询及退票
6. 修改密码：乘客可在个人中心页面修改密码
7. 留言：在个人中心页面可以点击留言，进行反馈

从管理员角度看：

1. 登录：管理员可以通过登录权限进入管理员模式
2. 车次添加及取消：管理员可以对火车车次和高铁车次进行添加或取消
3. 乘客信息管理：管理员可以查看乘客信息，管理乘客密码及删除乘客信息
4. 发布公告：管理员可以进行公告的发布，公告将在查询界面显示
5. 留言管理：管理员可以对留言进行删除
6. 功能结构图

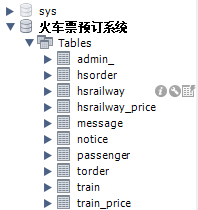


通过上述功能，可以实现网上火车票查询、预定、退票功能，还可以针对老年人进行优选选座，足不出户便可预定车票，方便快捷，减少资源浪费。

二、**数据库分析**

本火车票预订系统中设有十张表，表设定如下：

1. admin\_表中含有A\_id、A\_password。A\_id是主键。
2. hsorder表中含有H\_name、P\_id、H\_seattype、nowtime、H\_date。H\_name是主键。
3. hsrailway表中有H\_id、H\_name、H\_leavetime、H\_arrivetime、H\_destination、H\_departure、H\_date。H\_id是主键。
4. hsrailway\_price表中有H\_id、H\_name、H\_seattype、H\_number、H\_price、H\_date。H\_id是主键。
5. message表中有M\_message。
6. passenger表中有P\_name、P\_phone、P\_address、P\_id、P\_password、P\_email。P\_id是主键。
7. notice表中有content。
8. torder表中有T\_name、P\_id、T\_seattype、nowtime、T\_date、softtype、hardtype。T\_name是主键。
9. train表中有T\_id、T\_name、T\_leavetime、T\_arrivetime、T\_departure、T\_destination、T\_date。T\_id是主键
10. train\_price表中有T\_id、T\_name、T\_seattype、T\_price、T\_number、T\_date。T\_id是主键。



十张表具体设计：

1. admin\_



|  |  |
| --- | --- |
| 字段名 | 含义 |
| A\_id | 管理员id（主键） |
| A\_password | 管理员密码 |

1. hsorder



|  |  |
| --- | --- |
| 字段名 | 含义 |
| H\_name | 高铁动车车次（主键） |
| P\_id | 乘客身份证号（外键） |
| H\_seattype | 高铁动车座位类型 |
| nowtime | 现在的时间 |
| H\_date | 高铁动车出发日期 |

1. hsrailway



|  |  |
| --- | --- |
| 字段名 | 含义 |
| H\_id | 高铁动车序号 |
| H\_name | 高铁动车车次 |
| H\_leavetime | 高铁动车出发时间 |
| H\_arrivetime | 高铁动车到达时间 |
| H\_destination | 高铁动车出发地 |
| H\_departure | 高铁动车到达地 |
| H\_date | 高铁动车出发日期 |

1. hsrailway\_price



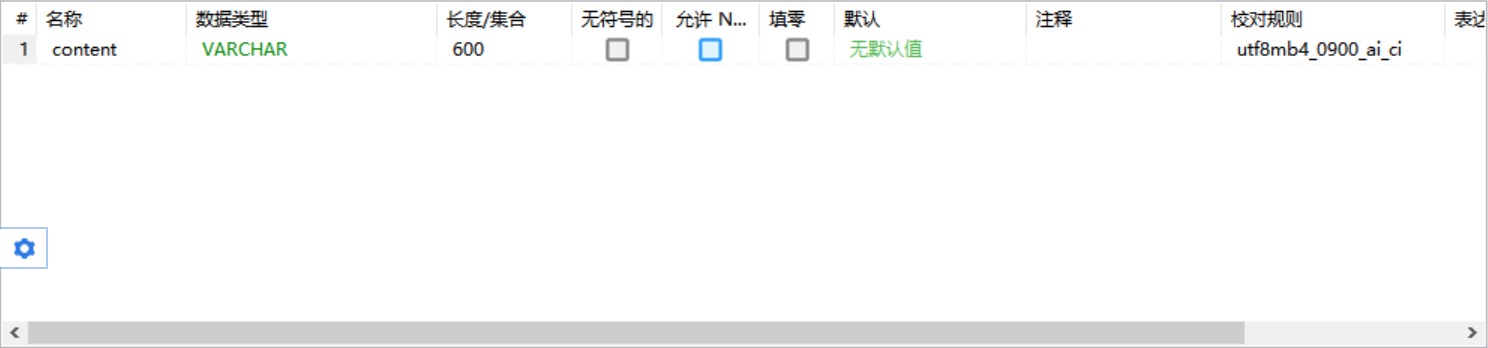
|  |  |
| --- | --- |
| 字段名 | 含义 |
| H\_id | 高铁动车序号（主键） |
| H\_name | 高铁车次 |
| H\_seattype | 高铁座位类型 |
| H\_number | 每个座位余量 |
| H\_price | 每个座位对应价格 |
| H\_date | 高铁出发日期 |

1. message



|  |  |
| --- | --- |
| 字段名 | 含义 |
| M\_message | 乘客留言内容 |

1. notice



|  |  |
| --- | --- |
| 字段名 | 含义 |
| content | 管理员发布公告内容 |

1. passenger



|  |  |
| --- | --- |
| 字段名 | 含义 |
| P\_name | 乘客姓名 |
| P\_phone | 乘客手机号码 |
| P\_address | 乘客地址 |
| P\_id | 乘客身份证号码（主键） |
| P\_password | 乘客密码 |
| P\_email | 乘客电子邮箱 |

1. torder



|  |  |
| --- | --- |
| 字段名 | 含义 |
| T\_name | 火车车次名（主键） |
| P\_id | 乘客身份证号码（外键） |
| T\_seattype | 火车座位类型 |
| nowtime | 现在的时间 |
| T\_date | 火车出发日期 |
| softtype | 火车软卧类型（上、下铺） |
| hardtype | 火车硬卧类型（上、中、下铺） |

1. train



|  |  |
| --- | --- |
| 字段名 | 含义 |
| T\_id | 火车序号（主键） |
| T\_name | 火车车次名字 |
| T\_leavetime | 火车出发时间 |
| T\_arrivetime | 火车到达时间 |
| T\_departure | 火车出发地 |
| T\_destination | 火车到达地 |
| T\_date | 火车出发日期 |

1. train\_price



|  |  |
| --- | --- |
| 字段名 | 含义 |
| T\_id | 火车车次序号（主键） |
| T\_name | 火车车次名字 |
| T\_seattype | 火车座位类型 |
| T\_price | 各座位类型对应的价格 |
| T\_number | 各座位类型余量 |
| T\_date | 火车出发时间 |

三、**系统实现**

1. 实体类的实现

在系统中共设计了11个实体类，包括database类，admin\_类，hsorder类，hsrailway类，hsrailway\_price类，message类，notice类，passenger类，torder类，train类，train\_price类。database类实现了数据的访问，admin\_类，hsorder类，hsrailway类，hsrailway\_price类，message类，notice类，passenger类，torder类，train类，train\_price类描述了管理员，高铁订单，高铁，高铁价格，留言，公告，乘客，火车订单，火车和火车价格的实体，分别与数据库中的10个表和字段一一对应。

1. **database类的实现**

通过database类实现数据库访问。代码如下：

package newpackage;

import java.sql.Connection;

import java.sql.DriverManager;

public class database {

private String dbpath="jdbc:mysql://localhost:3306/火车票预订系统?serverTimezone=GMT%2B8&useSSL=false";

private String dbUserName="root";

private String dbPassword="Yanyuqi0623";//s输入你的数据库密码

private String jdbcName="com.mysql.cj.jdbc.Driver";

public database(){

}

public String GetDBPath()

{

return dbpath;

}

public void SetDBPath(String dbpath)

{

dbpath=dbpath;

}

public Connection getCon() throws Exception{

Class.forName(jdbcName);

//Connection con=DriverManager.getConnection(dbpath,dbUserName,dbPassword);

Connection con=DriverManager.getConnection(dbpath,dbUserName,dbPassword);

return con;

}

public void closeCon(Connection con) throws Exception{

if(con!=null){

con.close();

}

}

public void OpenDB()

{database db=new database();

try {

db.getCon();

System.out.println("数据库连接成功!");

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public void main(String[] args) {

OpenDB();

}

}

1. **admin\_类的实现**

package newpackage;

public class admin\_{

private String A\_id;

private String A\_password;

public admin\_()

{

super();

}

public String GetA\_id(){

return A\_id;

}

public void setA\_id(String A\_id)

{

this.A\_id=A\_id;

}

public String GetA\_password(){

return A\_password;

}

public void setA\_password(String A\_password)

{

this.A\_password=A\_password;

}

}

1. **hsorder类的实现**

package newpackage;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

public class hsorder extends hsrailway{

private String H\_name;

private String H\_seattype;

private String P\_id;

private Timestamp Nowtime;

private String H\_date;

public hsorder(){

super();

}

public hsorder(String H\_id,String H\_name){

}

public String GetP\_id(){

return P\_id;

}

public void setP\_id(String P\_id)

{

this.P\_id=P\_id;

}

public String GetH\_seattype(){

return H\_seattype;

}

public void setH\_seattype(String H\_seattype)

{

this.H\_seattype=H\_seattype;

}

public String GetH\_name(){

return H\_name;

}

public void setH\_name(String H\_name)

{

this.H\_name=H\_name;

}

public Timestamp GetNowtime(){

return Nowtime;

}

public void setNowtime(Timestamp Nowtime)

{

this.Nowtime=Nowtime;

}

public String GetH\_date(){

return H\_date;

}

public void setH\_date(String H\_date)

{

this.H\_date=H\_date;

}

public ResultSet PassengerList(Connection con, passenger passenger) throws SQLException {

StringBuffer sb = new StringBuffer("select hsrailway.H\_name,hsrailway.H\_departure,hsrailway.H\_destination,hsrailway.H\_leavetime,hsrailway.H\_arrivetime,hsorder.H\_seattype,hsrailway\_price.H\_price,hsorder.H\_date,hsrailway\_price.H\_date from hsrailway\_price,hsrailway,hsorder where hsrailway.H\_name=hsorder.H\_name and hsrailway.H\_id=hsrailway\_price.H\_id and hsrailway\_price.H\_seattype=hsorder.H\_seattype and hsrailway\_price.H\_date=hsorder.H\_date");

if (StringUtil.isNotEmpty(passenger.GetP\_id())) {

sb.append(" and P\_id = '" + passenger.GetP\_id() + "'");

}

System.out.println(sb.toString());

PreparedStatement pstmt = con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}

public int SelectionCancel(Connection con)throws Exception{

String sql="delete from hsorder where H\_name=? and H\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, H\_name);

pstmt.setString(2, H\_date);

return pstmt.executeUpdate();

}

public int NumSelectedMinus(Connection con)throws Exception{

String sql="update hsorder set numSelected = numSelected - 1 where H\_name=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, H\_name);

return pstmt.executeUpdate();

}

public int SelectionAdd(Connection con)throws Exception{

String sql="insert into hsorder (H\_name,P\_id,H\_seattype,Nowtime,H\_date) values(?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,H\_name);

pstmt.setString(2,P\_id);

pstmt.setString(3,H\_seattype);

pstmt.setTimestamp(4,Nowtime);

pstmt.setString(5,H\_date);

return pstmt.executeUpdate();

}

public int H\_numberMinus(Connection con,String H\_name,String H\_seattype,String H\_date)throws Exception{

String sql="update hsrailway\_price set H\_number = H\_number - 1 where H\_name=? and H\_seattype=? and H\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, H\_name);

pstmt.setString(2, H\_seattype);

pstmt.setString(3, H\_date);

return pstmt.executeUpdate();

}

public ResultSet isExisted(Connection con) throws SQLException{

String sql = "select \* from hsorder where P\_id=? and H\_name=? ";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, P\_id);

pstmt.setString(2, H\_name);

return pstmt.executeQuery();

}

public int H\_numberAdd(Connection con,String H\_name,String H\_seattype,String H\_date)throws Exception{

String sql="update hsrailway\_price set H\_number = H\_number + 1 where H\_name=? and H\_seattype=? and H\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, H\_name);

pstmt.setString(2, H\_seattype);

pstmt.setString(3, H\_date);

return pstmt.executeUpdate();

}

}

1. **hsrailway类的实现**

package newpackage;

import java.sql.Time;

public class hsrailway extends hsrailway\_price{

static void update(String sql, String time) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

private String H\_id;

private String H\_name;

private Time H\_leavetime;

private Time H\_arrivetime;

private String H\_departure;

private String H\_destination;

private String H\_date;

public hsrailway()

{

super();

}

public hsrailway(String H\_id)

{

super();

H\_id=H\_id;

}

public hsrailway(String H\_id,String H\_name) {

super();

H\_id=H\_id;

H\_name=H\_name;

}

public String GetH\_id(){

return H\_id;

}

public void setH\_id(String H\_id)

{

this.H\_id=H\_id;

}

public String GetH\_name(){

return H\_name;

}

public void setH\_name(String H\_name)

{

this.H\_name=H\_name;

}

public Time GetH\_leavetime(){

return H\_leavetime;

}

public void setH\_leavetime(Time H\_leavetime)

{

this.H\_leavetime=H\_leavetime;

}

public Time GetH\_arrivetime(){

return H\_arrivetime;

}

public void setH\_arrivetime(Time H\_arrivetime)

{

this.H\_arrivetime=H\_arrivetime;

}

public String GetH\_departure(){

return H\_departure;

}

public void setH\_departure(String H\_departure)

{

this.H\_departure=H\_departure;

}

public String GetH\_destination(){

return H\_destination;

}

public void setH\_destination(String H\_destination)

{

this.H\_destination=H\_destination;

}

public String GetH\_date(){

return H\_date;

}

public void setH\_date(String H\_date)

{

this.H\_date=H\_date;

}

}

1. **hsrailway\_price类的实现**

package newpackage;

public class hsrailway\_price {

private String H\_id;

private String H\_name;

private String H\_seattype;

private String H\_price;

private String H\_number;

private String H\_date;

public hsrailway\_price()

{

}

public hsrailway\_price(String H\_id,String H\_name,String H\_seattype)

{ H\_id=H\_id;

H\_name=H\_name;

H\_seattype=H\_seattype;

}

public String GetH\_name()

{

return H\_name;

}

public void setH\_name(String H\_name)

{

this.H\_name=H\_name;

}

public String GetH\_seattype()

{

return H\_seattype;

}

public void setH\_seattype(String H\_seattype)

{

this.H\_seattype=H\_seattype;

}

public String GetH\_id()

{

return H\_id;

}

public void setH\_id(String H\_id)

{

this.H\_id=H\_id;

}

public String GetH\_price()

{

return H\_price;

}

public void setH\_price(String H\_price)

{

this.H\_price=H\_price;

}

public String GetH\_number()

{

return H\_number;

}

public void setH\_number(String H\_number)

{

this.H\_number=H\_number;

}

public String GetH\_date(){

return H\_date;

}

public void setH\_date(String H\_date)

{

this.H\_date=H\_date;

}

}

1. **message类的实现**

package newpackage;

public class message {

private String M\_number;

private String M\_message;

public String GetM\_number(){

return M\_number;

}

public void setM\_number(String M\_number)

{

this.M\_number=M\_number;

}

public String GetM\_message(){

return M\_message;

}

public void setM\_message(String M\_message)

{

this.M\_message=M\_message;

}

}

1. **notice类的实现**

package newpackage;

public class message {

private String M\_number;

private String M\_message;

public String GetM\_number(){

return M\_number;

}

public void setM\_number(String M\_number)

{

this.M\_number=M\_number;

}

public String GetM\_message(){

return M\_message;

}

public void setM\_message(String M\_message)

{

this.M\_message=M\_message;

}

}

1. **passenger类的实现**

package newpackage;

public class passenger{

private String P\_phone;

private String P\_address;

private String P\_id;

private String P\_password;

private String P\_email;

private String P\_name;

private String P\_age;

private String P\_sex;

public passenger()

{

}

public passenger(String P\_name)

{

P\_name=P\_name;

}

public passenger(String P\_name,String P\_password) {

super();

P\_name=P\_name;

P\_password=P\_password;

}

public String GetP\_name(){

return P\_name;

}

public void setP\_name(String P\_name)

{

this.P\_name=P\_name;

}

public String GetP\_password(){

return P\_password;

}

public void setP\_password(String P\_password)

{

this.P\_password=P\_password;

}

public String GetP\_phone(){

return P\_phone;

}

public void setP\_phone(String P\_phone)

{

this.P\_phone=P\_phone;

}

public String GetP\_address(){

return P\_address;

}

public void setP\_address(String P\_address)

{

this.P\_address=P\_address;

}

public String GetP\_id(){

return P\_id;

}

public void setP\_id(String P\_id)

{

this.P\_id=P\_id;

}

public String GetP\_email(){

return P\_email;

}

public void setP\_email(String P\_email){

this.P\_email=P\_email;

}

public String GetP\_age(){

return P\_age;

}

public void setP\_age(String P\_age)

{

this.P\_age=P\_age;

}

public String GetP\_sex(){

return P\_sex;

}

public void setP\_sex(String P\_sex)

{

this.P\_sex=P\_sex;

}

void P\_age(String string) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

1. **torder类的实现**

package newpackage;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Timestamp;

public class torder extends train {

private String T\_name;

private String P\_id;

private String T\_seattype;

private Timestamp Nowtime;

private String T\_date;

private String softype;

private String hardtype;

public torder(){

super();

}

public String GetP\_id(){

return P\_id;

}

public void setP\_id(String P\_id)

{

this.P\_id=P\_id;

}

public String GetT\_seattype(){

return T\_seattype;

}

public void setT\_seattype(String T\_seattype)

{

this.T\_seattype=T\_seattype;

}

public String GetT\_name(){

return T\_name;

}

public void setT\_name(String T\_name)

{

this.T\_name=T\_name;

}

public Timestamp GetNowtime(){

return Nowtime;

}

public void setNowtime(Timestamp Nowtime)

{

this.Nowtime=Nowtime;

}

public String GetT\_date(){

return T\_date;

}

public void setT\_date(String T\_date)

{

this.T\_date=T\_date;

}

public String Getsoftype(){

return softype;

}

public void setsoftype(String softype)

{

this.softype=softype;

}

public String Gethardtype(){

return hardtype;

}

public void sethardtype(String hardtype)

{

this.hardtype=hardtype;

}

public ResultSet PassengerList(Connection con, passenger passenger) throws SQLException {

StringBuffer sb = new StringBuffer("select train.T\_name,train.T\_departure,train.T\_destination,train.T\_leavetime,train.T\_arrivetime,torder.T\_seattype,train\_price.T\_price,torder.T\_date,train\_price.T\_date from train\_price,train,torder where train.T\_name=torder.T\_name and train.T\_id=train\_price.T\_id and train\_price.T\_seattype=torder.T\_seattype and train\_price.T\_date=torder.T\_date");

if (StringUtil.isNotEmpty(passenger.GetP\_id())) {

sb.append(" and P\_id = '" + passenger.GetP\_id() + "'");

}

System.out.println(sb.toString());

PreparedStatement pstmt = con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}

public int SelectionCancel(Connection con)throws Exception{

String sql="delete from torder where T\_name=? and T\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, T\_name);

pstmt.setString(2, T\_date);

return pstmt.executeUpdate();

}

public int NumSelectedMinus(Connection con)throws Exception{

String sql="update torder set numSelected = numSelected - 1 where T\_name=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, T\_name);

return pstmt.executeUpdate();

}

public int T\_numberMinus(Connection con,String T\_name,String T\_seattype,String T\_date)throws Exception{

String sql="update train\_price set T\_number = T\_number - 1 where T\_name=? and T\_seattype=? and T\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, T\_name);

pstmt.setString(2, T\_seattype);

pstmt.setString(3, T\_date);

return pstmt.executeUpdate();

}

public ResultSet isExisted(Connection con) throws SQLException{

String sql = "select \* from torder where P\_id=? and T\_name=? ";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, P\_id);

pstmt.setString(2, T\_name);

return pstmt.executeQuery();

}

public int SelectionAdd(Connection con)throws Exception{

String sql="insert into torder (T\_name,P\_id,T\_seattype,Nowtime,T\_date,softype,hardtype) values(?,?,?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,T\_name);

pstmt.setString(2,P\_id);

pstmt.setString(3,T\_seattype);

pstmt.setTimestamp(4,Nowtime);

pstmt.setString(5,T\_date);

pstmt.setString(6,softype);

pstmt.setString(7,hardtype);

return pstmt.executeUpdate();

}

public int T\_numberAdd(Connection con,String T\_name,String T\_seattype,String T\_date)throws Exception{

String sql="update train\_price set T\_number = T\_number + 1 where T\_name=? and T\_seattype=? and T\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1, T\_name);

pstmt.setString(2, T\_seattype);

pstmt.setString(3, T\_date);

return pstmt.executeUpdate();

}

}

1. **train类的实现**

package newpackage;

import java.sql.Time;

public class train extends train\_price{

static void update(String sql, String time) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

private String T\_id;

private String T\_name;

private Time T\_leavetime;

private Time T\_arrivetime;

private String T\_departure;

private String T\_destination;

private String T\_date;

public train()

{

super();

}

public train(String T\_id)

{

super();

T\_id=T\_id;

}

public train(String T\_id,String T\_name) {

super();

T\_id=T\_id;

T\_name=T\_name;

}

public String GetT\_id(){

return T\_id;

}

public void setT\_id(String T\_id)

{

this.T\_id=T\_id;

}

public String GetT\_name(){

return T\_name;

}

public void setT\_name(String T\_name)

{

this.T\_name=T\_name;

}

public Time GetT\_leavetime(){

return T\_leavetime;

}

public void setT\_leavetime(Time T\_leavetime)

{

this.T\_leavetime=T\_leavetime;

}

public Time GetT\_arrivetime(){

return T\_arrivetime;

}

public void setT\_arrivetime(Time T\_arrivetime)

{

this.T\_arrivetime=T\_arrivetime;

}

public String GetT\_departure(){

return T\_departure;

}

public void setT\_departure(String T\_departure)

{

this.T\_departure=T\_departure;

}

public String GetT\_destination(){

return T\_destination;

}

public void setT\_destination(String T\_destination)

{

this.T\_destination=T\_destination;

}

public String GetT\_date(){

return T\_date;

}

public void setT\_date(String T\_date)

{

this.T\_date=T\_date;

}

}

1. **train\_price类的实现**

package newpackage;

public class train\_price {

private String T\_id;

private String T\_name;

private String T\_seattype;

private String T\_price;

private String T\_number;

private String T\_date;

public train\_price()

{

}

public train\_price(String T\_id,String T\_name,String T\_seattype)

{ T\_id=T\_id;

T\_name=T\_name;

T\_seattype=T\_seattype;

}

public String GetT\_name()

{

return T\_name;

}

public void setT\_name(String T\_name)

{

this.T\_name=T\_name;

}

public String GetT\_seattype()

{

return T\_seattype;

}

public void setT\_seattype(String T\_seattype)

{

this.T\_seattype=T\_seattype;

}

public String GetT\_id()

{

return T\_id;

}

public void setT\_id(String T\_id)

{

this.T\_id=T\_id;

}

public String GetT\_price()

{

return T\_price;

}

public void setT\_price(String T\_price)

{

this.T\_price=T\_price;

}

public String GetT\_number()

{

return T\_number;

}

public void setT\_number(String T\_number)

{

this.T\_number=T\_number;

}

public String GetT\_date(){

return T\_date;

}

public void setT\_date(String T\_date)

{

this.T\_date=T\_date;

}

}

1. control类的实现

public int passengerAdd(Connection con,passenger passenger)throws Exception {

String sql="insert into sinfo values(?,?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_name());

pstmt.setString(2,passenger.GetP\_phone());

pstmt.setString(3,passenger.GetP\_address());

pstmt.setString(4,passenger.GetP\_id());

pstmt.setString(5,passenger.GetP\_password());

pstmt.setString(6,passenger.GetP\_email());

return pstmt.executeUpdate();

}

public int psssengerModify(Connection con, passenger passenger)throws Exception{

String sql="update sinfo set P\_name = ? ,P\_phone = ?, P\_address = ?, P\_password = ?,P\_email = ? where P\_id = ?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_name());

pstmt.setString(2,passenger.GetP\_phone());

pstmt.setString(3,passenger.GetP\_address());

pstmt.setString(4,passenger.GetP\_id());

pstmt.setString(5,passenger.GetP\_password());

pstmt.setString(6,passenger.GetP\_email());

return pstmt.executeUpdate();

}

public int passengerDelete(Connection con, String P\_id) throws SQLException {

//To change body of generated methods, choose Tools | Templates.

String sql = "delete from passenger where P\_id=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, P\_id);

return pstmt.executeUpdate();

}

public int trainAdd(Connection con,train train)throws Exception {

String sql="insert into train values(?,?,?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,train.GetT\_id());

pstmt.setString(2,train.GetT\_name());

pstmt.setTime(3,train.GetT\_leavetime());

pstmt.setTime(4,train.GetT\_arrivetime());

pstmt.setString(5,train.GetT\_departure());

pstmt.setString(6,train.GetT\_destination());

pstmt.setString(7,train.GetT\_date());

return pstmt.executeUpdate();

}

public int trainModify(Connection con, train train)throws Exception{

String sql="update train set T\_name = ? ,T\_leavetime = ?, T\_arrivetime = ?, T\_departure = ?,T\_destination = ?，T\_seattype = ?,T\_price = ?,T\_number = ? where T\_id = ?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(2,train.GetT\_name());

pstmt.setTime(3,train.GetT\_leavetime());

pstmt.setTime(4,train.GetT\_arrivetime());

pstmt.setString(5,train.GetT\_departure());

pstmt.setString(6,train.GetT\_destination());

pstmt.setString(7,train.GetT\_seattype());

pstmt.setString(8,train.GetT\_price());

pstmt.setString(9,train.GetT\_number());

return pstmt.executeUpdate();

}

public int trainDelete(Connection con,String T\_id, String T\_date,String T\_name) throws SQLException {

//To change body of generated methods, choose Tools | Templates.

String sql = "delete from train where T\_id=? and T\_date=? and T\_name=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, T\_id);

pstmt.setString(2, T\_date);

pstmt.setString(3, T\_name);

return pstmt.executeUpdate();

}

public int hsrailwayAdd(Connection con,hsrailway hsrailway)throws Exception {

String sql="insert into hsrailway values(?,?,?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,hsrailway.GetH\_id());

pstmt.setString(2,hsrailway.GetH\_name());

pstmt.setTime(3,hsrailway.GetH\_leavetime());

pstmt.setTime(4,hsrailway.GetH\_arrivetime());

pstmt.setString(5,hsrailway.GetH\_departure());

pstmt.setString(6,hsrailway.GetH\_destination());

pstmt.setString(7,hsrailway.GetH\_date());

return pstmt.executeUpdate();

}

public int hsrailwayModify(Connection con, hsrailway hsrailway)throws Exception{

String sql="update train set H\_name = ? ,H\_leavetime = ?, H\_arrivetime = ?, H\_departure = ?,H\_destination = ?,H\_seattype = ?,H\_price = ?,H\_number = ? where H\_id = ?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(2,hsrailway.GetH\_name());

pstmt.setTime(3,hsrailway.GetH\_leavetime());

pstmt.setTime(4,hsrailway.GetH\_arrivetime());

pstmt.setString(5,hsrailway.GetH\_departure());

pstmt.setString(6,hsrailway.GetH\_destination());

pstmt.setString(7,hsrailway.GetH\_seattype());

pstmt.setString(8,hsrailway.GetH\_price());

pstmt.setString(9,hsrailway.GetH\_number());

return pstmt.executeUpdate();

}

public int hsrailwayDelete(Connection con, String H\_id,String H\_date,String H\_name) throws SQLException {

//To change body of generated methods, choose Tools | Templates.

String sql = "delete from hsrailway where H\_id=? and H\_date=? and H\_name=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, H\_id);

pstmt.setString(2, H\_date);

pstmt.setString(3, H\_name);

return pstmt.executeUpdate();

}

public int messageAdd(Connection con,message message)throws Exception {

String sql="insert into message values(?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,message.GetM\_message());

return pstmt.executeUpdate();

}

public int messageDelete(Connection con, String M\_message) throws SQLException {

//To change body of generated methods, choose Tools | Templates.

String sql = "delete from message where M\_message=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, M\_message);

return pstmt.executeUpdate();

}

public passenger Verifypassenger(Connection con,passenger passenger) throws Exception{

passenger resultpas=null;

String sql="select \* from passenger where P\_id=? and P\_password=?"; PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_id());

pstmt.setString(2, passenger.GetP\_password());

ResultSet rs=pstmt.executeQuery();

if(rs.next()){

resultpas=new passenger();

resultpas.setP\_id(rs.getString("P\_id"));

resultpas.setP\_password(rs.getString("P\_password"));

}

return resultpas;

}

public admin\_ Verifyadmin\_(Connection con, admin\_ admin\_) throws Exception{

admin\_ resultad=null;

String sql="select \* from admin\_ where A\_id=? and A\_password=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,admin\_.GetA\_id());

pstmt.setString(2, admin\_.GetA\_password());

ResultSet rs=pstmt.executeQuery();

if(rs.next()){

resultad=new admin\_();

resultad.setA\_id(rs.getString("A\_id"));

resultad.setA\_password(rs.getString("A\_password"));

}

return resultad;

}

public String QueryPassword(Connection con,passenger passenger) throws SQLException

{

String sql ="select P\_password from passenger where P\_id = ? ";

System.out.println(passenger.GetP\_id());

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_id());

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return rs.getString("P\_password");

}

else

{

return "";

}

}

public int ModifyPassword(Connection con,passenger passenger)throws Exception{

String sql="update passenger set P\_password = ? where P\_id = ?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_password());

pstmt.setString(2,passenger.GetP\_id());

return pstmt.executeUpdate();

}

public String QueryPassword\_(Connection con,admin\_ admin\_) throws SQLException

{

String sql ="select A\_password from admin\_ where A\_id = ? ";

System.out.println(admin\_.GetA\_id());

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,admin\_.GetA\_id());

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return rs.getString("A\_password");

}

else

{

return "";

}

}

public int ModifyPassword\_(Connection con,admin\_ admin\_)throws Exception{

String sql="update admin\_ set A\_password = ? where A\_id = ?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,admin\_.GetA\_password());

pstmt.setString(2,admin\_.GetA\_password());

return pstmt.executeUpdate();

}

public Boolean CanpassengerAdd(Connection con,String tablename,String columnname,String P\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public Boolean CanpassengerModify(Connection con,String tablename,String columnname,String P\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public Boolean CantrainAdd(Connection con,String tablename,String columnname,String T\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,T\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public Boolean CantrainModify(Connection con,String tablename,String columnname,String T\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,T\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public Boolean CanhsrailwayAdd(Connection con,String tablename,String columnname,String H\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,H\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public Boolean CanhsrailwayModify(Connection con,String tablename,String columnname,String H\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,H\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public Boolean CanmessageAdd(Connection con,String tablename,String columnname,String M\_number) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,M\_number);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public Boolean CanmessageModify(Connection con,String tablename,String columnname,String M\_number) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,M\_number);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public ResultSet QueryList(Connection con,String tablename,String columnname,String pno) throws SQLException

{

String sql="";

if (pno.isEmpty()){

sql="select \* from " + tablename;

PreparedStatement pstmt=con.prepareStatement(sql);

return pstmt.executeQuery();

}

else

{

sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,pno);

return pstmt.executeQuery();

}

}

public ResultSet Querytrain(Connection con,String T\_departure,String T\_destination,String T\_date) throws SQLException

{

String sql = "select train.T\_name,T\_leavetime,T\_arrivetime,T\_departure,T\_destination,train.T\_date,train\_price.T\_seattype,T\_price,T\_number from train,train\_price where train.T\_id=train\_price.T\_id and train.T\_departure=? and train.T\_destination=? and train.T\_date=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,T\_departure);

pstmt.setString(2,T\_destination);

pstmt.setString(3,T\_date);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public ResultSet Queryhsrailway(Connection con,String H\_departure,String H\_destination,String H\_date) throws SQLException

{

String sql="";

sql = "select hsrailway.H\_name,H\_leavetime,H\_arrivetime,H\_departure,H\_destination,hsrailway.H\_date,hsrailway\_price.H\_seattype,H\_price,H\_number from hsrailway,hsrailway\_price where hsrailway.H\_id=hsrailway\_price.H\_id and hsrailway.H\_departure=? and hsrailway.H\_destination=? and hsrailway.H\_date=?" ;

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,H\_departure);

pstmt.setString(2,H\_destination);

pstmt.setString(3,H\_date);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public ResultSet Queryallhsrailway(Connection con) throws SQLException

{

String sql="";

sql = "select hsrailway.H\_name,H\_leavetime,H\_arrivetime,H\_departure,H\_destination,hsrailway.H\_date,hsrailway\_price.H\_seattype,H\_price,H\_number from hsrailway,hsrailway\_price where hsrailway.H\_id=hsrailway\_price.H\_id";

PreparedStatement pstmt=con.prepareStatement(sql);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public ResultSet Queryalltrain(Connection con) throws SQLException

{

String sql="";

sql = "select train.T\_name,T\_leavetime,T\_arrivetime,T\_departure,T\_destination,train.T\_date,train\_price.T\_seattype,T\_price,T\_number from train,train\_price where train.T\_id=train\_price.T\_id" ;

PreparedStatement pstmt=con.prepareStatement(sql);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public int noticeAdd(Connection con,notice notice)throws Exception {

String sql="insert into notice values (?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,notice.Getcontent());

return pstmt.executeUpdate();

}

boolean CannoticeAdd(Connection con, String notice, String content) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

public ResultSet Querytrainorder(Connection con,passenger passenger) throws SQLException

{

StringBuffer sb = new StringBuffer("select train.T\_name,torder.T\_date,train.T\_departure,train.T\_destination,train.T\_leavetime,train.T\_arrivetime,torder.T\_seattype,torder.hardtype,torder.softype,train\_price.T\_price,torder.Nowtime from train\_price,train,torder where train.T\_name=torder.T\_name and train.T\_id=train\_price.T\_id and train\_price.T\_seattype=torder.T\_seattype and train\_price.T\_date=torder.T\_date");

if (StringUtil.isNotEmpty(passenger.GetP\_id())) {

sb.append(" and P\_id = '" + passenger.GetP\_id() + "'");

}

System.out.println(sb.toString());

PreparedStatement pstmt = con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}

public ResultSet Querytrainorder1(Connection con,passenger passenger) throws SQLException

{

StringBuffer sb = new StringBuffer("select train.T\_name,torder.T\_date,train.T\_departure,train.T\_destination,train.T\_leavetime,train.T\_arrivetime,torder.T\_seattype,train\_price.T\_price,torder.Nowtime from train\_price,train,torder where train.T\_name=torder.T\_name and train.T\_id=train\_price.T\_id and train\_price.T\_seattype=torder.T\_seattype and train\_price.T\_date=torder.T\_date");

if (StringUtil.isNotEmpty(passenger.GetP\_id())) {

sb.append(" and P\_id = '" + passenger.GetP\_id() + "'");

}

System.out.println(sb.toString());

PreparedStatement pstmt = con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}

public ResultSet Queryhsrailwayorder(Connection con,passenger passenger) throws SQLException

{

StringBuffer sb = new StringBuffer("select hsrailway.H\_name,hsorder.H\_date,hsrailway.H\_departure,hsrailway.H\_destination,hsrailway.H\_leavetime,hsrailway.H\_arrivetime,hsorder.H\_seattype,hsrailway\_price.H\_price,hsorder.Nowtime from hsrailway\_price,hsrailway,hsorder where hsrailway.H\_name=hsorder.H\_name and hsrailway.H\_id=hsrailway\_price.H\_id and hsrailway\_price.H\_seattype=hsorder.H\_seattype and hsrailway\_price.H\_date=hsorder.H\_date");

if (StringUtil.isNotEmpty(passenger.GetP\_id())) {

sb.append(" and P\_id = '" + passenger.GetP\_id() + "'");

}

System.out.println(sb.toString());

PreparedStatement pstmt = con.prepareStatement(sb.toString());

return pstmt.executeQuery();

}

public ResultSet Querynotice(Connection con) throws SQLException

{

String sql="";

sql = "select\* from notice";

PreparedStatement pstmt=con.prepareStatement(sql);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public ResultSet Queryallpassenger(Connection con) throws SQLException

{

String sql="";

sql = "select P\_name,P\_phone,P\_address,P\_id,P\_password,P\_email,\n" +

"timestampdiff(year,str\_to\_date(substring(P\_id,7,8),\"%Y%m%d\"),str\_to\_date(now(),\"%Y-%m-%d\"))as P\_age,\n" +

"case if(length(P\_id)=18, cast(substring(P\_id,17,1) as UNSIGNED)%2, \n" +

"if(length(P\_id)=15,cast(substring(P\_id,15,1) as UNSIGNED)%2,3))\n" +

"when 1 then '男' \n" +

"when 0 then '女'\n" +

"else '未知'\n" +

"end as P\_sex\n" +

"from passenger";

PreparedStatement pstmt=con.prepareStatement(sql);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public ResultSet Querypassenger(Connection con,String P\_id) throws SQLException

{

String sql="";

sql = "select P\_name,P\_phone,P\_address,P\_id,P\_password,P\_email,\n" +

"timestampdiff(year,str\_to\_date(substring(P\_id,7,8),\"%Y%m%d\"),str\_to\_date(now(),\"%Y-%m-%d\"))as P\_age,\n" +

"case if(length(P\_id)=18, cast(substring(P\_id,17,1) as UNSIGNED)%2, \n" +

"if(length(P\_id)=15,cast(substring(P\_id,15,1) as UNSIGNED)%2,3))\n" +

"when 1 then '男' \n" +

"when 0 then '女'\n" +

"else '未知'\n" +

"end as P\_sex\n" +

"from passenger\n" +

"where P\_id =?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

System.out.println(pstmt.toString());

return pstmt.executeQuery();

}

public String QueryoldPassword(Connection con,passenger passenger) throws SQLException

{

String sql ="select P\_password from passenger where P\_id = ? ";

System.out.println(passenger.GetP\_id());

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_id());

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return rs.getString("P\_password");

}

else

{

return "";

}

}

public int train\_priceAdd(Connection con,train\_price train\_price)throws Exception {

String sql="insert into train\_price values(?,?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,train\_price.GetT\_id());

pstmt.setString(2,train\_price.GetT\_name());

pstmt.setString(3,train\_price.GetT\_seattype());

pstmt.setString(4,train\_price.GetT\_price());

pstmt.setString(5,train\_price.GetT\_number());

pstmt.setString(6,train\_price.GetT\_date());

return pstmt.executeUpdate();

}

public int hsrailway\_priceAdd(Connection con,hsrailway\_price hsrailway\_price)throws Exception {

String sql="insert into hsrailway\_price values(?,?,?,?,?,?)";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,hsrailway\_price.GetH\_id());

pstmt.setString(2,hsrailway\_price.GetH\_name());

pstmt.setString(3,hsrailway\_price.GetH\_seattype());

pstmt.setString(4,hsrailway\_price.GetH\_price());

pstmt.setString(5,hsrailway\_price.GetH\_number());

pstmt.setString(6,hsrailway\_price.GetH\_date());

return pstmt.executeUpdate();

}

public Boolean Cantrain\_priceAdd(Connection con,String tablename,String columnname,String T\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,T\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public Boolean Canhsrailway\_priceAdd(Connection con,String tablename,String columnname,String H\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,H\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public Boolean CannoticeAdd(Connection con,String tablename,String columnname,String content) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,content);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public ResultSet passengerList(Connection con, passenger passenger) throws SQLException {

/\*\*StringBuffer sb = new StringBuffer("select \* from slogin ");

if (student.getSno() != "") {

sb.append("where Sno='" + student.getSno() + "'");

}

PreparedStatement pstmt = con.prepareStatement(sb.toString());\*/

StringBuilder sb = new StringBuilder("select \* from passenger");

/\*\*if (student.getSno() != "") {

sb.append(" and Sno=" + student.getSno());

}\*/

if (StringUtil.isNotEmpty(passenger.GetP\_id())) {

sb.append(" and jTextField1 ='").append(passenger.GetP\_id()).append("'");

}

PreparedStatement pstmt = con.prepareStatement(sb.toString().replaceFirst("and", "where"));

return pstmt.executeQuery();

}

public int ModifyP\_password(Connection con,passenger passenger)throws Exception{

String sql="update passenger set spassword = ? where P\_id = ?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,passenger.GetP\_password());

pstmt.setString(2,passenger.GetP\_id());

return pstmt.executeUpdate();

}

public ResultSet messageList(Connection con, message message) throws SQLException {

/\*\*StringBuffer sb = new StringBuffer("select \* from slogin ");

if (student.getSno() != "") {

sb.append("where Sno='" + student.getSno() + "'");

}

PreparedStatement pstmt = con.prepareStatement(sb.toString());\*/

StringBuilder sb = new StringBuilder("select \* from message");

/\*\*if (student.getSno() != "") {

sb.append(" and Sno=" + student.getSno());

}\*/

if (StringUtil.isNotEmpty(message.GetM\_message())) {

sb.append(" and message ='").append(message.GetM\_message()).append("'");

}

PreparedStatement pstmt = con.prepareStatement(sb.toString().replaceFirst("and", "where"));

return pstmt.executeQuery();

}

public int DeleteM\_message(Connection con, String M\_message) throws SQLException

{

String sql = "delete from message where M\_message=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1,M\_message);

return pstmt.executeUpdate();

}

public Boolean Candelete(Connection con,String tablename,String columnname,String M\_message) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,M\_message);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return false;

}

else

{

return true;

}

}

public int ConfirmDelete(Component com,String str,String Sno)

{

int n= JOptionPane.showConfirmDialog(com, "确定要删除" + str + Sno +"吗？");

return n;

}

public Boolean CanModify(Connection con,String tablename,String columnname,String P\_id) throws SQLException

{

String sql="select \* from " + tablename + " where " + columnname + " = ? ";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public Boolean Modifyseattype1(Connection con,String P\_id) throws SQLException

{

String sql="select\* from torder where T\_seattype = '硬卧'and P\_id=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public Boolean Modifyseattype2(Connection con,String P\_id) throws SQLException

{

String sql="select\* from torder where T\_seattype = '软卧'and P\_id=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

public int train\_priceDelete(Connection con,String T\_id, String T\_date,String T\_name) throws SQLException {

//To change body of generated methods, choose Tools | Templates.

String sql = "delete from train\_price where T\_id=? and T\_date=? and T\_name=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, T\_id);

pstmt.setString(2, T\_date);

pstmt.setString(3, T\_name);

return pstmt.executeUpdate();

}

public int hsrailway\_priceDelete(Connection con, String H\_id,String H\_date,String H\_name) throws SQLException {

//To change body of generated methods, choose Tools | Templates.

String sql = "delete from hsrailway\_price where H\_id=? and H\_date=? and H\_name=?";

PreparedStatement pstmt = con.prepareStatement(sql);

pstmt.setString(1, H\_id);

pstmt.setString(2, H\_date);

pstmt.setString(3, H\_name);

return pstmt.executeUpdate();

}

}

1. 表单对象类的实现

import java.awt.Component;

import javax.swing.JOptionPane;

public class FormObject {

public control thecontrol;

public train thetrain[];

public hsrailway thehsrailway[];

public FormObject()

{

}

public void ShowInfo(Component com,String strinfo)

{

JOptionPane.showMessageDialog(com, strinfo);

}

public void ShowSuccessInfo(Component com,String strinfo)

{

JOptionPane.showMessageDialog(com, strinfo);

}

}

1. 判断类的实现

public class StringUtil {

public static boolean isEmpty(String str){

if("".equals(str)||str==null){

return true;

}else{

return false;

}

}

public static boolean isNotEmpty(String str){

if(!"".equals(str)&&str!=null){

return true;

}else{

return false;

}

}

}

1. 边界类的实现
2. 登录界面



该界面主要有两个功能：管理员登录，乘客登录。管理员通过输入账号（A\_id）和密码（A\_password）与Mysql表中的admin\_进行比对，如果信息正确则进入管理员中心界面，如果所填内容为空或者错误则出现提示，“用户名不能为空”，“密码不能为空”，“登录失败”，无法登录。乘客通过输入账号（P\_id）和密码（P\_password）与Mysql表中的passenger进行比对，如果信息正确则进入主页和查询界面，如果所填内容为空或者错误则出现提示“用户名不能为空”，“密码不能为空”，“登录失败”。

该界面有三个按钮：登录，注册，重置

登录按钮下的程序如下所示：

private void jb\_logOnActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String userName = userNameTxt.getText();

String password = new String(passwordTxt.getPassword());

if (StringUtil.isEmpty(userName)) {

fo.ShowInfo(this, "用户名不能为空!");

return;

}

if (StringUtil.isEmpty(password)) {

fo.ShowInfo(this, "密码不能为空!");

return;

}

Connection con=null;

if (this.jrb\_passenger.isSelected()) {

passenger passenger = new passenger();

passenger.setP\_id(userName);

passenger.setP\_password(password);

try {

con=database.getCon();

currentpassenger = co.Verifypassenger(con,passenger);

if (currentpassenger != null) {

this.dispose();

new 主页和查询JFrame().setVisible(true);

} else {

JOptionPane.showMessageDialog(this, "登录失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "登录失败!");

}finally{

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

} else if (this.jrb\_admin\_.isSelected()) {

admin\_ admin\_ = new admin\_();

admin\_.setA\_id(userName);

admin\_.setA\_password(password);

try {

con=database.getCon();

currentadmin\_ = co.Verifyadmin\_(con, admin\_);

if (currentadmin\_!= null) {

this.dispose();

new 管理员JFrame().setVisible(true);

} else {

JOptionPane.showMessageDialog(this, "登录失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "登录失败!");

}finally{

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

P\_id = this.userNameTxt.getText();

}

点击注册按钮，进入注册界面。

点击重置按钮，清空所输入的账号和密码。

1. 主页和查询界面



该界面（主页和查询JFrame）主要实现了两个功能：乘客对火车的查询，乘客对高铁动车的查询。乘客通过输入出发日期（T\_date/H\_date），出发地点（T\_departure/H\_daparture），到达地点（T\_destination/H\_destination），点击查询按钮查找Mysql表中train/hsrailway中的信息，进入火车车次详情进行火车的查询。在基础上点击是否乘坐高铁动车，可以进入高铁动车车次详情，实现对高铁动车的查询。

该界面有两个按钮：个人中心，查询。

查询按钮下的程序如下：

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

if (this.jCheckBox1.isSelected()) {

try {

H\_date= this.jTextField1.getText();

H\_departure = this.jTextField6.getText();

H\_destination = this.jTextField7.getText();

System.out.println(主页和查询JFrame.H\_date);

System.out.println(主页和查询JFrame.H\_departure);

System.out.println(主页和查询JFrame.H\_destination);

this.dispose();

高铁动车车次详情JFrame1 b = new 高铁动车车次详情JFrame1();

b.setVisible(true);

} catch (Exception ex) {

Logger.getLogger(主页和查询JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

else{

try {

T\_date = this.jTextField1.getText();

T\_departure = this.jTextField6.getText();

T\_destination = this.jTextField7.getText();

System.out.println(主页和查询JFrame.T\_date);

System.out.println(主页和查询JFrame.T\_departure);

System.out.println(主页和查询JFrame.T\_destination);

this.dispose();

火车车次详情JFrame1 a = new 火车车次详情JFrame1();

a.setVisible(true);

} catch (Exception ex) {

Logger.getLogger(主页和查询JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

点击个人中心按钮，进入乘客个人中心界面。

该界面中的公告，会显示管理员发布的公告信息。从Mysql的notice表中获得数据。

1. 火车车次详情界面



该界面主要实现一个功能：火车票预订。乘客在表中点击想要预定的车次，若预订成功会出现“预订成功”的提示。若预订火车重复、同时选择多于一个车次或火车票没有剩余则会出现相应的提示。

当乘客年龄>=60岁，系统自动识别，跳出软卧硬卧的铺位选择。





该界面有四个按钮：预订，全部火车信息，个人中心，返回。

预订按钮下的程序如下：

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Timestamp Nowtime =new Timestamp(System.currentTimeMillis());

if (T\_name== "" && T\_seattype=="") {

JOptionPane.showMessageDialog(this, "请选择一个车次！");

return;

}

if (T\_number== 0) {

JOptionPane.showMessageDialog(this, "该车次没有票了！");

return;

}

int n = JOptionPane.showConfirmDialog(this, "确定要选择这个车次吗?");

if (n == 0) {

Connection con = null;

String currentP\_id = 登录.currentpassenger.GetP\_id();

torder sc=new torder();

sc.setT\_name(T\_name);

sc.setT\_seattype(T\_seattype);

sc.setNowtime(Nowtime);

sc.setT\_date(T\_date);

sc.setP\_id(currentP\_id);

try {

con = database.getCon();

ResultSet rs = sc.isExisted(con);

while (rs.next()) {

JOptionPane.showMessageDialog(this, "你已经选过这个车次，不能重复选择！");

return;

}

int selectionNum = sc.SelectionAdd(con);

int selectedNum = sc.T\_numberMinus(con,T\_name,T\_seattype,T\_date);

System.out.println(selectionNum);

System.out.println(selectedNum);

if (selectionNum == 1 && selectedNum >= 1) {

JOptionPane.showMessageDialog(this, "预订成功!");

} else {

JOptionPane.showMessageDialog(this, "预订失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "预订失败!");

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

Connection con = null;

try {

con=database.getCon();

} catch (Exception ex) {

Logger.getLogger(火车车次详情JFrame1.class.getName()).log(Level.SEVERE, null, ex);

}

Boolean rs = null;

try {

rs = Queryage(con,登录.P\_id);

System.out.println(rs);

} catch (Exception ex) {

Logger.getLogger(火车车次详情JFrame1.class.getName()).log(Level.SEVERE, null, ex);

}

if(rs != false){

Boolean modifyseattype1 = null;

Boolean modifyseattype2 = null;

try {

modifyseattype1 = co.Modifyseattype1(con,登录.P\_id);

modifyseattype2 = co.Modifyseattype2(con,登录.P\_id);

} catch (SQLException ex) {

Logger.getLogger(火车车次详情JFrame1.class.getName()).log(Level.SEVERE, null, ex);

}

System.out.println(T\_seattype);

System.out.println("车次为："+getname);

if(gettype.equals("硬卧")){

if(modifyseattype1 == true){

老年人硬卧 a = new 老年人硬卧();

a.setVisible(true);

}

}

if(gettype.equals("软卧")){

if(modifyseattype2 == true){

老年人软卧 b = new 老年人软卧();

b.setVisible(true);

}

}

}

}

全部火车信息按钮下的程序如下：

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

fillTable1(train);

}

点击个人中心按钮，进入乘客个人中心界面。

点击返回按钮，返回主页和查询界面界面。

鼠标功能的实现：

private void jTable1MousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = jTable1.getSelectedRow();

T\_name = (String) this.jTable1.getValueAt(row, 0);

T\_seattype = (String) this.jTable1.getValueAt(row, 6);

T\_date = (String) jTable1.getValueAt(row, 5);

T\_number = Integer.parseInt((String) this.jTable1.getValueAt(row, 8));

gettype=(String) this.jTable1.getValueAt(row, 6);

System.out.println(gettype);

getname=(String) this.jTable1.getValueAt(row, 0);

}

filltable功能的实现：

private void fillTable(train train, String T\_departure, String T\_destination,String T\_date) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

try {

} catch (Exception ex){

Logger.getLogger(主页和查询JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

ResultSet rs = co.Querytrain(con,T\_departure,T\_destination,T\_date);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

v.add(rs.getString(9));

dtm.addRow(v);

}

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

private void fillTable1(train train){

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.Queryalltrain(con);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

v.add(rs.getString(9));

dtm.addRow(v);

}

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

识别年龄功能的实现：

public Boolean Queryage(Connection con,String P\_id) throws Exception{

String sql="";

sql="select timestampdiff(year,str\_to\_date(substring(P\_id,7,8),\"%Y%m%d\"),str\_to\_date(now(),\"%Y-%m-%d\"))>= 60 as P\_age from passenger where timestampdiff(year,str\_to\_date(substring(P\_id,7,8),\"%Y%m%d\"),str\_to\_date(now(),\"%Y-%m-%d\"))>= 60 and P\_id=?";

PreparedStatement pstmt=con.prepareStatement(sql);

pstmt.setString(1,P\_id);

System.out.println(pstmt.toString());

ResultSet rs=pstmt.executeQuery();

if (rs.next()){

return true;

}

else

{

return false;

}

}

1. 高铁动车车次详情界面



该界面主要实现一个功能：高铁动车车票预订。乘客在表中点击想要预定的车次，若预订成功会出现“预订成功”的提示。若预订高铁动车重复、同时选择多于一个车次或高铁动车票没有剩余则会出现相应的提示。

该界面有四个按钮：预订，全部高铁动车信息，个人中心，返回。

预订按钮下的程序如下：

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

Timestamp Nowtime =new Timestamp(System.currentTimeMillis());

if (H\_name== "" && H\_seattype=="") {

JOptionPane.showMessageDialog(this, "请选择一个车次！");

return;

}

if (H\_number== 0) {

JOptionPane.showMessageDialog(this, "该车次没有票了！");

return;

}

int n = JOptionPane.showConfirmDialog(this, "确定要选择这个车次吗?");

if (n == 0) {

Connection con = null;

String currentP\_id = 登录.currentpassenger.GetP\_id();

hsorder sc=new hsorder();

sc.setH\_name(H\_name);

sc.setH\_seattype(H\_seattype);

sc.setNowtime(Nowtime);

sc.setH\_date(H\_date);

sc.setP\_id(currentP\_id);

try {

con = database.getCon();

ResultSet rs = sc.isExisted(con);

while (rs.next()) {

JOptionPane.showMessageDialog(this, "你已经选过这个车次，不能重复选择！");

return;

}

int selectionNum = sc.SelectionAdd(con);

int selectedNum = sc.H\_numberMinus(con,H\_name,H\_seattype,H\_date);

System.out.println(selectionNum);

System.out.println(selectedNum);

if (selectionNum == 1 && selectedNum >= 1) {

JOptionPane.showMessageDialog(this, "预订成功!");

} else {

JOptionPane.showMessageDialog(this, "预订失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "预订失败!");

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

全部高铁动车信息程序如下：

fillTable1(hsrailway);

点击个人中心按钮，进入乘客个人中心界面。

点击返回按钮，返回主页和查询界面界面。

鼠标功能的实现：

private void jTable1MousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = jTable1.getSelectedRow();

H\_name = (String) jTable1.getValueAt(row, 0);

H\_seattype = (String) jTable1.getValueAt(row, 6);

H\_date = (String) jTable1.getValueAt(row, 5);

H\_number = Integer.parseInt((String) jTable1.getValueAt(row, 8));

}

Filltable功能的实现：

private void fillTable(hsrailway hsrailway, String H\_departure, String H\_destination,String H\_date) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

try {

} catch (Exception ex){

Logger.getLogger(主页和查询JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

ResultSet rs = co.Queryhsrailway(con,H\_departure,H\_destination,H\_date);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

v.add(rs.getString(9));

dtm.addRow(v);

}

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

private void fillTable1(hsrailway hsrailway){

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.Queryallhsrailway(con);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

v.add(rs.getString(9));

dtm.addRow(v);

}

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

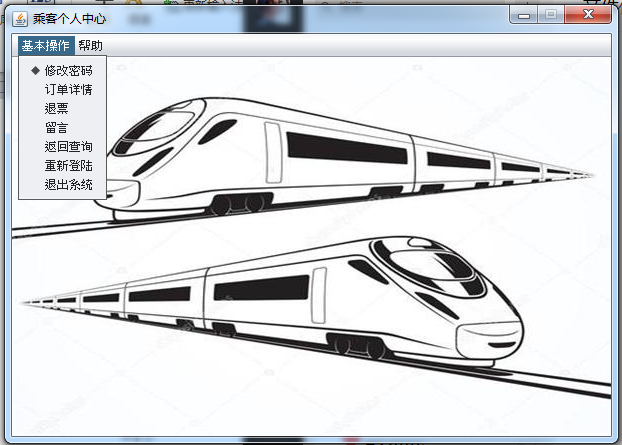
e.printStackTrace();

}

}

}

1. 乘客个人中心界面



该界面中的基本操作下，有七个选项：修改密码，订单详情，退票，留言，返回系统，重新登录，退出系统。分别连接了修改密码，订单详情，退票，留言，返回系统，登录界面。点击退出系统，会出现提示。

其中，若>=60岁的乘客点击订单详情，进入订单详情界面；若<60岁的乘客点击订单详情，则进入订单详情1。

订单详情按钮下的程序如下：

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String P\_id= 登录.currentpassenger.GetP\_id();

Boolean rs = null;

Connection con = null;

try {

con=database.getCon();

} catch (Exception ex) {

Logger.getLogger(乘客个人中心JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

try {

火车车次详情JFrame1 a = new 火车车次详情JFrame1();

rs = a.Queryage(con,登录.P\_id);

System.out.println(rs);

} catch (Exception ex) {

Logger.getLogger(火车车次详情JFrame1.class.getName()).log(Level.SEVERE, null, ex);

}

if(rs == true){

this.dispose();

new 订单详情JFrame().setVisible(true);

}

else{

this.dispose();

new 订单详情JFrame1().setVisible(true);

}

}

1. 修改密码



该界面实现了一个功能：密码的修改。乘客输入原密码，新密码，再输入一遍新密码。

该界面有三个按钮：返回，修改，重置

修改按钮下的程序如下：

private void jb\_modifyActionPerformed(java.awt.event.ActionEvent evt) {

try {

// TODO add your handling code here:

String oldPassword = new String(this.oldPasswordTxt.getPassword());

String newPassword = new String(this.newPasswordTxt.getPassword());

String passwordConfirm = new String(this.passwordConfirmTxt.getPassword());

if (StringUtil.isEmpty(oldPassword)) {

JOptionPane.showMessageDialog(this, "原密码不能为空!");

return;

}

if (StringUtil.isEmpty(newPassword)) {

JOptionPane.showMessageDialog(this, "新密码不能为空!");

return;

}

if (StringUtil.isEmpty(passwordConfirm)) {

JOptionPane.showMessageDialog(this, "必须重复输入新密码!");

return;

}

Connection con = null;

con=database.getCon();

String currentP\_id=登录.currentpassenger.GetP\_id();

passenger passenger = new passenger();

passenger.setP\_id(currentP\_id);

String rightOldPassword="";

try {

rightOldPassword = co.QueryoldPassword(con, passenger);

} catch (SQLException ex) {

Logger.getLogger(修改密码.class.getName()).log(Level.SEVERE, null, ex);

}

if (!oldPassword.equals(rightOldPassword)) {

JOptionPane.showMessageDialog(this, "原密码错误，请重新输入!");

return;

}

if (!newPassword.equals(passwordConfirm)) {

JOptionPane.showMessageDialog(this, "两次输入的密码必须相同!");

return;

}

passenger.setP\_password(newPassword);

try {

con = database.getCon();

int modifyNum = co.ModifyPassword(con, passenger);

if (modifyNum == 1) {

JOptionPane.showMessageDialog(this, "密码修改成功!");

this.resetValue();

} else {

JOptionPane.showMessageDialog(this, "密码修改失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "修改失败!");

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

} catch (Exception ex) {

Logger.getLogger(修改密码.class.getName()).log(Level.SEVERE, null, ex);

}

}

点击重置按钮，清空输入信息

点击返回按钮，返回乘客个人中心

1. 订单详情1（<60岁的乘客出现该界面）



该界面实现两个功能：展示该乘客的火车订单和高铁动车订单

Filltable程序的实现：

private void fillTable(train train) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

torder sc=new torder();

String P\_id= 登录.currentpassenger.GetP\_id();

passenger.setP\_id(P\_id);

System.out.println(passenger.GetP\_id());

try {

con = database.getCon();

ResultSet rs = co.Querytrainorder1(con,passenger);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

v.add(rs.getString(9));

dtm.addRow(v);

}

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

private void fillTable1(hsrailway hsraiway) {

DefaultTableModel dtm = (DefaultTableModel) jTable2.getModel();

dtm.setRowCount(0);

Connection con = null;

hsorder sc=new hsorder();

String P\_id= 登录.currentpassenger.GetP\_id();

passenger.setP\_id(P\_id);

System.out.println(passenger.GetP\_id());

try {

con = database.getCon();

ResultSet rs = co.Queryhsrailwayorder(con,passenger);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

v.add(rs.getString(9));

dtm.addRow(v);

}

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

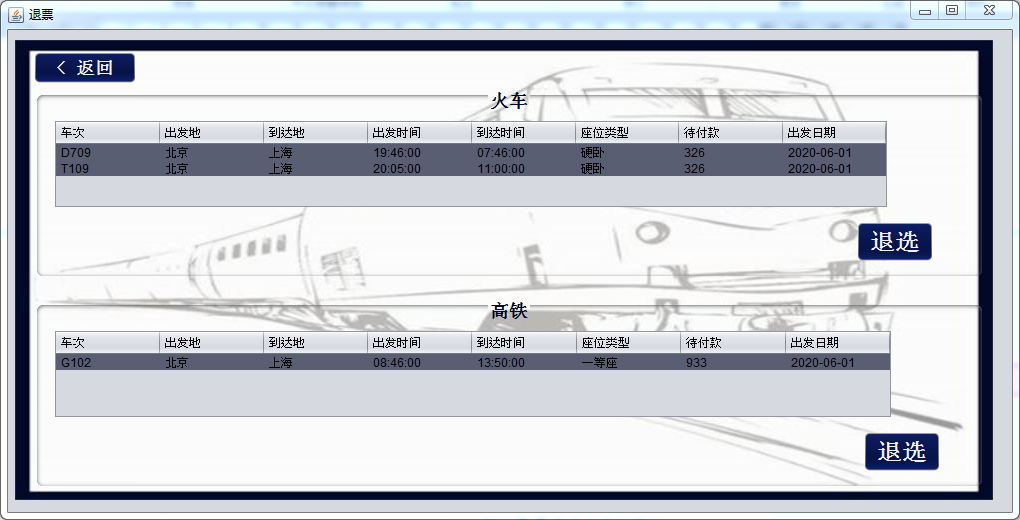
1. 订单详情（>=60岁的乘客出现该界面）



该界面实现两个功能：展示该乘客的火车订单和高铁动车订单

Filltable程序与5.6的filltable差不多，少了软卧备注和硬卧备注，不赘述。

1. 退票



该界面实现两个功能：火车和高铁动车退票。乘客通过点击表中数据进行退票

该界面中有三个按钮：退票，退票，返回。

第一个退票按钮下的程序：

private void jToggleButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if (T\_name=="" && T\_seattype=="") {

JOptionPane.showMessageDialog(this, "请选择要退选的车次！");

return;

}

int n = JOptionPane.showConfirmDialog(this, "确认要退选这个车次吗?");

if (n == 0) {

Connection con = null;

String currentP\_id = 登录.currentpassenger.GetP\_id();

torder sc= new torder();

sc.setT\_name(T\_name);

sc.setT\_date(T\_date);

sc.setT\_seattype(T\_seattype);

sc.setP\_id(currentP\_id);

try {

con = database.getCon();

int selectionNum = sc.SelectionCancel(con);

int selectedNum = sc.T\_numberAdd(con,T\_name,T\_seattype,T\_date);

if (selectionNum == 1 && selectedNum == 1) {

JOptionPane.showMessageDialog(this, "退选成功!");

this.fillTable1(new torder());

} else {

JOptionPane.showMessageDialog(this, "退选失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "退选失败!");

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

第二个退票按钮下的程序：

private void jb\_selectionCancelActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if (H\_name=="" && H\_seattype=="") {

JOptionPane.showMessageDialog(this, "请选择要退选的车次！");

return;

}

int m = JOptionPane.showConfirmDialog(this, "确认要退选这个车次吗?");

if (m == 0) {

Connection con = null;

String currentP\_id = 登录.currentpassenger.GetP\_id();

hsorder sc= new hsorder();

sc.setP\_id(currentP\_id);

sc.setH\_seattype(H\_seattype);

sc.setH\_name(H\_name);

sc.setH\_date(H\_date);

try {

con = database.getCon();

int selectionNum = sc.SelectionCancel(con);

int selectedNum = sc.H\_numberAdd(con,H\_name,H\_seattype,H\_date);

System.out.println(selectionNum);

System.out.println(selectedNum);

if (selectionNum == 1 && selectedNum >= 1) {

JOptionPane.showMessageDialog(this, "退选成功!");

this.fillTable2(new hsorder());

} else {

JOptionPane.showMessageDialog(this, "退选失败!");

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

JOptionPane.showMessageDialog(this, "退选失败!");

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

鼠标功能的实现：

private void tordertableMousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = tordertable.getSelectedRow();

T\_name= (String) tordertable.getValueAt(row, 0);

T\_date = (String) tordertable.getValueAt(row, 7);

T\_seattype= (String) tordertable.getValueAt(row, 5);

}

private void hsordertableMousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = hsordertable.getSelectedRow();

H\_name= (String) hsordertable.getValueAt(row, 0);

H\_date = (String) hsordertable.getValueAt(row, 7);

H\_seattype= (String) hsordertable.getValueAt(row, 5);

}

Filltable功能的实现：

private void fillTable1(torder torder) {

DefaultTableModel dtm = (DefaultTableModel) tordertable.getModel();

dtm.setRowCount(0);

Connection con = null;

torder sc=new torder();

String P\_id= 登录.currentpassenger.GetP\_id();

passenger.setP\_id(P\_id);

System.out.println(passenger.GetP\_id());

try {

con = database.getCon();

ResultSet rs = sc.PassengerList(con, passenger);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

private void fillTable2(hsorder hsorder) {

DefaultTableModel dtm = (DefaultTableModel) hsordertable.getModel();

dtm.setRowCount(0);

Connection con = null;

hsorder sc=new hsorder();

String P\_id = 登录.currentpassenger.GetP\_id();

passenger.setP\_id(P\_id);

System.out.println(passenger.GetP\_id());

try {

con = database.getCon();

ResultSet rs = sc.PassengerList(con, passenger);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString(1));

v.add(rs.getString(2));

v.add(rs.getString(3));

v.add(rs.getString(4));

v.add(rs.getString(5));

v.add(rs.getString(6));

v.add(rs.getString(7));

v.add(rs.getString(8));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

点击返回按钮，返回乘客个人中心界面。

1. 留言界面



该界面实现了一个功能：留言。乘客把留言写在界面中，输入到Mysql的message表中。管理员在留言显示中可以查看并删除。

该界面有两个按钮：确定，返回。

确定按钮下的程序如下：

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

// TODO add your handling code here:

try{

database database=new database();

int m=0;

String M\_message = this.message.getText();

message message=new message();

message.setM\_message(M\_message);

if(StringUtil.isEmpty(M\_message)){

fo.ShowInfo(this, "公告不能为空!");

}

Connection con=null;

con=database.getCon();

if (co.CanmessageAdd(con,"message","M\_message",M\_message)){

m = co.messageAdd(con,message);

if(m==1){

fo.ShowInfo(this, "发布成功!");

this.resetValue();

}else{

fo.ShowInfo(this, "发布失败!");

}

} else {

}

} catch (Exception ex) {

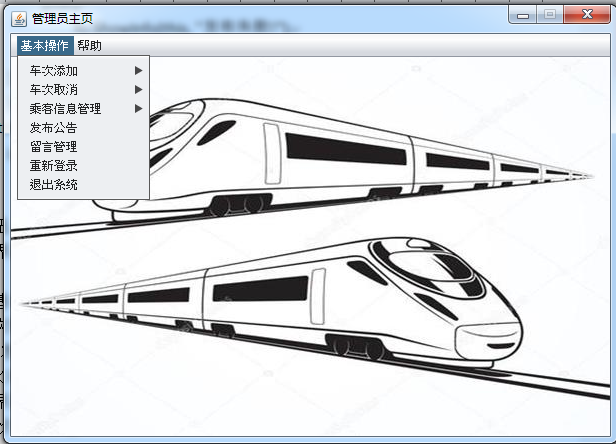
Logger.getLogger(留言.class.getName()).log(Level.SEVERE, null, ex);

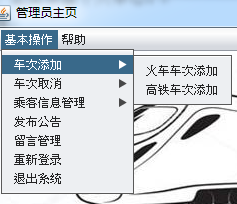
}

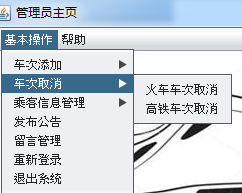
}

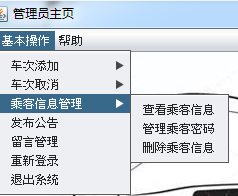
点击返回按钮，返回乘客个人中心。

1. 管理员界面



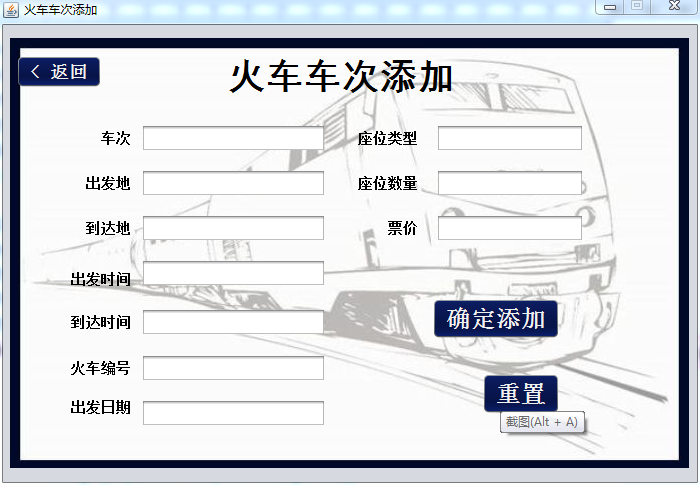






该界面中的基本操作下，有七个选项：车次添加（火车车次添加，高铁车次添加），车次取消（火车车次取消，高铁车次取消），乘客信息管理（查看乘客信息，管理乘客密码，删除乘客信息），发布公告，留言管理，重新登录，退出系统。分别连接了火车车次添加，高铁车次添加，火车车次取消，高铁车次取消，查看乘客信息，管理乘客密码，管理员删除乘客信息，登录界面。点击退出系统，会出现提示。

1. 火车车次添加界面



该界面实现了车次添加的功能。

管理员依次填入火车相关信息，点击确定按钮，系统识别Mysql数据库的相关数据，若无重复则添加成功，数据加入到Mysql的train和train\_price的表中。

该界面有三个按钮：确定添加，重置，返回。

确定添加按钮下的程序：

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

try{ // TODO add your handling code here:

database database=new database();

int n=0;

int m=0;

String T\_departure=this.jTextField2.getText();

String T\_destination=this.jTextField3.getText();

String T\_leavetime=this.jTextField4.getText();

String T\_arrivetime=this.jTextField6.getText();

String T\_name=this.jTextField1.getText();

String T\_id=this.jTextField5.getText();

String T\_seattype=this.jTextField7.getText();

String T\_price=this.jTextField9.getText();

String T\_number=this.jTextField8.getText();

String T\_date=this.jTextField10.getText();

SimpleDateFormat format = new SimpleDateFormat("hh:mm:ss");

java.util.Date d = null;

java.util.Date a= null;

try {

d = format.parse(T\_leavetime);

a = format.parse(T\_arrivetime);

} catch (Exception e) {

e.printStackTrace();

}

Time date = new java.sql.Time(d.getTime());

Time c = new java.sql.Time(a.getTime());

train train=new train();

train\_price train\_price=new train\_price();

train.setT\_departure(T\_departure);

train.setT\_destination(T\_destination);

train.setT\_leavetime(date);

train.setT\_arrivetime(c);

train.setT\_name(T\_name);

train.setT\_id(T\_id);

train.setT\_date(T\_date);

train\_price.setT\_id(T\_id);

train\_price.setT\_name(T\_name);

train\_price.setT\_seattype(T\_seattype);

train\_price.setT\_price(T\_price);

train\_price.setT\_number(T\_number);

train\_price.setT\_date(T\_date);

if(StringUtil.isEmpty(T\_name)){

fo.ShowInfo(this, "车次不能为空!");

return;

}

if(StringUtil.isEmpty(T\_departure)){

fo.ShowInfo(this, "出发地不能为空!");

return;

}

if(StringUtil.isEmpty(T\_destination)){

fo.ShowInfo(this, "到达地称不能为空!");

return;

}

if(StringUtil.isEmpty(T\_leavetime)){

fo.ShowInfo(this, "出发时间不能为空!");

return;

}

if(StringUtil.isEmpty(T\_arrivetime)){

fo.ShowInfo(this, "到达时间不能为空!");

return;

}

if(StringUtil.isEmpty(T\_seattype)){

fo.ShowInfo(this, "座位类型不能为空!");

return;

}

if(StringUtil.isEmpty(T\_number)){

fo.ShowInfo(this, "座位余量不能为空!");

return;

}

if(StringUtil.isEmpty(T\_price)){

fo.ShowInfo(this, "票价不能为空!");

return;

}

if(StringUtil.isEmpty(T\_date)){

fo.ShowInfo(this, "出发日期不能为空!");

return;

}

Connection con=null;

con=database.getCon();

if(co.Cantrain\_priceAdd(con,"train\_price", "T\_name", T\_id)&&co.CantrainAdd(con,"train","T\_name",T\_id)){

m = co.train\_priceAdd(con, train\_price);

n = co.trainAdd(con,train);

if(m==1&&n==1){

fo.ShowInfo(this, "添加成功");

this.resetValue();

}else{

fo.ShowInfo(this, "添加失败!");

}

}

else

{

fo.ShowInfo(this, "火车编号重复!");

}

} catch (Exception ex) {

Logger.getLogger(火车车次添加JFrame.class.getName()).log(Level.SEVERE, null, ex);

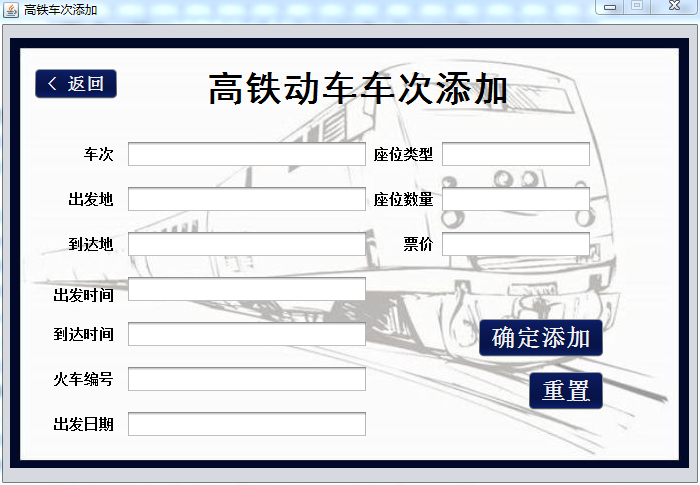
}

}

点击重置按钮，清空火车添加的信息。

点击返回按钮，返回管理员界面。

1. 高铁车次添加界面



该界面实现了车次添加的功能。

管理员依次填入高铁动车相关信息，点击确定按钮，系统识别Mysql数据库的相关数据，若无重复则添加成功，数据加入到Mysql的hsrailway和hsraiway\_price的表中。

该界面有三个按钮：确定添加，重置，返回。

高铁车次添加和火车车次添加异曲同工，在这里不赘述。

1. 火车车次取消界面



该界面实现了车次取消的功能。

管理员输入需要取消的火车编号，日期和车次。点击删除按钮，系统识别Mysql数据库的相关数据，若识别到信息，则删除Mysql的train和train\_price表中的相关信息。

该界面有两个按钮：删除，返回。

删除按钮下的程序如下：

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

// TODO add your handling code here:

String T\_id = this.id.getText();

String T\_name = this.t\_name.getText();

String T\_date = this.date.getText();

int deleteNum = 0;

int deleteNum1 = 0;

if (StringUtil.isEmpty(T\_name)) {

fo.ShowInfo(this, "车次不能为空!");

return;

}

int n = co.ConfirmDelete(this,"车次", T\_name);

if (n == JOptionPane.YES\_OPTION) {

Connection con = null;

try {

con = db.getCon();

} catch (Exception ex) {

Logger.getLogger(火车车次取消JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

try {

deleteNum=co.trainDelete(con,T\_id,T\_date,T\_name);

deleteNum1 = co.train\_priceDelete(con,T\_id,T\_date,T\_name);

} catch (SQLException ex) {

Logger.getLogger(火车车次取消JFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

else{

return;

}

if (deleteNum == 1&&deleteNum1 == 1) {

fo.ShowInfo(this, "删除成功!");

this.resetValue();

}

else {

fo.ShowInfo(this, "删除失败!");

}

}

点击返回按钮，返回管理员界面。

1. 高铁车次取消界面



该界面实现了车次取消的功能。

管理员输入需要取消的高铁编号，日期和车次。点击删除按钮，系统识别Mysql数据库的相关数据，若识别到信息，则删除Mysql的hsrailway和hsraiway\_price表中的相关信息。

该界面有两个按钮：删除，返回。

高铁车次取消和火车车次取消异曲同工，在这里不赘述。

1. 查看乘客信息界面



该界面实现了查看乘客信息的功能。

管理员输入想要查看的乘客信息（P\_id），点击查询。系统识别数据，查询Mysql数据库passenger表中的数据。

该界面有两个按钮：查询，返回。

查询按钮下的程序如下：

private void jb\_searchActionPerformed(java.awt.event.ActionEvent evt) {

{

String P\_id = this.SnoTxt.getText();

Connection con=null;

passenger passenger=new passenger();

this.fillTable1(passenger,P\_id);

if (StringUtil.isEmpty(P\_id)) {

fo.ShowInfo(this, "信息为空!");

return;

}

}

}

Filltable功能的实现：

private void fillTable(passenger passenger) {

DefaultTableModel dtm = (DefaultTableModel) studentTable.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.Queryallpassenger(con);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("P\_id"));

v.add(rs.getString("P\_name"));

v.add(rs.getString("P\_password"));

v.add(rs.getString("P\_phone"));

v.add(rs.getString("P\_address"));

v.add(rs.getString("P\_email"));

v.add(rs.getString("P\_age"));

v.add(rs.getString("P\_sex"));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

private void fillTable1(passenger passenger,String P\_id) {

DefaultTableModel dtm = (DefaultTableModel) studentTable.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.Querypassenger(con,P\_id);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("P\_id"));

v.add(rs.getString("P\_name"));

v.add(rs.getString("P\_password"));

v.add(rs.getString("P\_phone"));

v.add(rs.getString("P\_address"));

v.add(rs.getString("P\_email"));

v.add(rs.getString("P\_age"));

v.add(rs.getString("P\_sex"));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

点击返回按钮，返回管理员界面。

1. 乘客密码管理（管理乘客密码界面）



管理员点击表格内容或者输入乘客身份证号和乘客密码。系统识别Mysql passenger中的（P\_id），再对（P\_password）进行修改。

该界面有两个按钮：修改，返回。

修改按钮下的程序如下：

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String P\_id = this.jTextField2.getText();

String P\_password = this.jTextField3.getText();

passenger.setP\_id(P\_id);

passenger.setP\_password(P\_password);

if (StringUtil.isEmpty(P\_id)) {

JOptionPane.showMessageDialog(this, "身份证号不能为空!");

return;

}

if (StringUtil.isEmpty(P\_password)) {

JOptionPane.showMessageDialog(this, "用户密码不能为空!");

return;

}

Connection con = null;

try {

con=database.getCon();

} catch (Exception ex) {

Logger.getLogger(管理乘客密码.class.getName()).log(Level.SEVERE, null, ex);

}

int num =0;

try {

if (co.CanModify(con, "passenger", "P\_id", P\_id)) {

try {

num = co.ModifyPassword(con, passenger);

} catch (Exception ex) {

Logger.getLogger(管理乘客密码.class.getName()).log(Level.SEVERE, null, ex);

}

if (num == 1 ){

fo.ShowInfo(this, "修改成功!");

this.resetValue();

passenger passenger =new passenger();

this.fillTable(passenger);

}

else {

fo.ShowInfo(this, "修改失败!");

this.resetValue();

}

}

else

{

fo.ShowInfo(this, "没有这个用户");

}

} catch (SQLException ex) {

Logger.getLogger(管理乘客密码.class.getName()).log(Level.SEVERE, null, ex);

}

}

Filltable功能的实现

private void fillTable(passenger passenger) {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.passengerList(con, passenger);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("p\_name"));

v.add(rs.getString("p\_id"));

v.add(rs.getString("p\_password"));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

鼠标功能的实现：

private void jTable1MousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = jTable1.getSelectedRow();

this.jTextField2.setText((String) jTable1.getValueAt(row, 1));

this.jTextField3.setText((String) jTable1.getValueAt(row, 2));

}

点击返回按钮，返回管理员界面。

1. 删除乘客信息界面



该界面中，实现了删除乘客信息的功能。

管理员点击表格或者输入乘客身份证号码（P\_id）和乘客姓名（P\_name），点击删除，系统识别Mysql中passenger表的P\_id和P\_name，进行删除。

该界面有两个按钮：删除，返回。

删除按钮下的程序如下：

private void deleteActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String P\_id = this.jTextField2.getText();

String P\_name = this.jTextField3.getText();

passenger.setP\_id(P\_id);

passenger.setP\_name(P\_name);

int deleteNum = 0;

if (StringUtil.isEmpty(P\_id)) {

fo.ShowInfo(this, "身份证号不能为空!");

return;

}

int n = co.ConfirmDelete(this,"乘客", P\_name);

if (n == JOptionPane.YES\_OPTION) {

Connection con = null;

try {

con = database.getCon();

} catch (Exception ex) {

Logger.getLogger(管理员删除乘客信息.class.getName()).log(Level.SEVERE, null, ex);

}

try {

deleteNum=co.passengerDelete(con,P\_id);

} catch (SQLException ex) {

Logger.getLogger(管理员删除乘客信息.class.getName()).log(Level.SEVERE, null, ex);

}

}

else{

return;

}

if (deleteNum == 1) {

fo.ShowInfo(this, "删除成功!");

this.resetValue();

try {

this.fillTable();

} catch (Exception ex) {

Logger.getLogger(管理员删除乘客信息.class.getName()).log(Level.SEVERE, null, ex);

}

}

else {

fo.ShowInfo(this, "删除失败!");

};

}

鼠标功能的实现：

private void jTable1MousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = jTable1.getSelectedRow();

this.jTextField2.setText((String) jTable1.getValueAt(row, 0));

this.jTextField3.setText((String) jTable1.getValueAt(row, 1));

}

Filltable功能的实现：

private void fillTable() {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.passengerList(con, passenger);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("P\_id"));

v.add(rs.getString("P\_name"));

v.add(rs.getString("P\_phone"));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

点击返回按钮，返回管理员界面。

1. 发布公告



该界面实现管理员发布公告功能。管理员输入公告，添加进Mysql notice表中。在主页和查询界面表格中显示出来。

该界面与留言界面类似，不做赘述。

1. 留言管理（留言显示界面）



该界面实现了留言的删除功能。管理员点击表格内容，点击删除按钮，进行删除。

该界面有两个按钮：删除，返回。

private void deleteActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String M\_message = this.jTextField1.getText();

int deleteNum = 0;

if (StringUtil.isEmpty(M\_message)) {

fo.ShowInfo(this, "留言内容不能为空!");

return;

}

int n = co.ConfirmDelete(this,"message", M\_message);

if (n == JOptionPane.YES\_OPTION) {

Connection con = null;

try {

con=database.getCon();

} catch (Exception ex) {

Logger.getLogger(留言显示.class.getName()).log(Level.SEVERE, null, ex);

}

try {

deleteNum=co.messageDelete(con,M\_message);

} catch (SQLException ex) {

Logger.getLogger(留言显示.class.getName()).log(Level.SEVERE, null, ex);

}

}

else{

return;

}

if (deleteNum == 1) {

fo.ShowInfo(this, "删除成功!");

this.resetValue();

try {

this.fillTable();

} catch (Exception ex) {

Logger.getLogger(留言显示.class.getName()).log(Level.SEVERE, null, ex);

}

}

else {

fo.ShowInfo(this, "删除失败!");

};

}

鼠标功能实现：

private void jTable1MousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

int row = jTable1.getSelectedRow();

this.jTextField1.setText((String) jTable1.getValueAt(row, 0));

}

Filltable功能实现：

private void fillTable() {

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();

dtm.setRowCount(0);

Connection con = null;

try {

con = database.getCon();

ResultSet rs = co.messageList(con,message);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("M\_message"));

dtm.addRow(v);

}

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

database.closeCon(con);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

点击返回按钮，返回管理员界面。

四、课程设计体会

颜雨祺 18110218：

本学期的课设真的让我成长了很多，无论是心态还是知识获得，都让我有了一个飞跃。作为组长我认为有责任去承担责任，尽量让小组气氛积极和谐。这三个月里，我们小组三个人共同经历了探讨系统功能、系统开发、错误探讨。从最开始的逻辑不对，每个人都很崩溃到后来心平气并且积极面对一个个难题。尤其是最后几周，让我真的很感动，三个人每天上完课晚上编写程序，相互鼓励。虽然在写程序的时候会有一些分歧与矛盾，但我们积极沟通，让组内更加团结。

通过这次的课设我对实践出真知这句话有了全新的体会。从最开始的什么也不会，对于程序功能开发一点概念都没有，到后来通过看书或者问老师，慢慢变得有了想法也有了信心与能力写下这一串串的代码。最后看到自己写出来的系统能够正常运行，真的很自豪。

最后，我很感谢老师的帮助。谢谢老师在我们每次问一些很简单的问题的时候耐心的讲解。甚至帮我一起找程序里面的错误。老师的鼓励让我变得更加有信心去写好每一个代码，实现每一个功能。

王慧 18110219：

为期三个月的课设，令我感到体会满满，收获满满！心理素质也得到了磨炼，突然就接近尾声了，写着这个个人体会竟然还有一丝感动，从起初的不敢想象如何完成，到一个个界面成功运行，内心的激动和喜悦是用那些感叹词都无法描述的。

我第一个做的界面是登录，当时我认为代码都没有问题，但就是一直报错，后来问老师才知道，就是查询数据库的sql语句出现了问题，而这个问题就是少了一个空格，知道问题所在的我真的既难过又兴奋，难过自己找了好几天的错竟然就是一个空格的问题，兴奋这个登录功能实现了！这个登录功能的实现给了我信心，让我有决心做好后面的工作。但其实每个功能的实现都是充满坎坷的，每次点击运行文件时都会有那么点紧张与期盼，总是会遇到各种小问题，小组内进行讨论解决，找老师寻求帮助，每次问题得以解决，都会觉得自己又进步了一点，在各种报错中成长，其实感觉也很好！

课设虽然接近尾声了，但是做课设的这股劲儿要继续保持，课设带给我的不仅是知识储备增加，更是一种克服困难的力量！过程艰辛，但一切都值得！

自来乐·白克力 18110260：

这个学期经历了“艰难险阻”的对象编程与数据库技术这门课要结束了，经过这三个月来的历练，我的收获颇丰。

首先，对于上个学期数据库没有好好学的人来说，这无疑是巨大的挑战。无论是具体的步骤设计，还是从代码开发，都是补足上个学期遗憾的好机会。从最开始编写代码的迷惘，和到现在能够独立自主完成某一个界面的开发，这其中的历练只有自己懂（跪了。）

其次，团队合作也是一种新的挑战。团队合作并非易事，特别是之前从来没有这么长时间的共同工作的经历，这次课程设计也是给了我们磨合到完美配合的机会。我相信如果有下次相同的课程设计作业，小组成员之间一定能够更加完美的配合。

最后，我想说的是，此次课程设计给了我一次磨炼自己和重新认识自己专业的机会。让我从排斥计算机、排斥编程，到慢慢接受，并且不断提升自己，这个过程是充实的。看到成品的心情，不是买一两件衣服或者看两个小时电影能够比拟的。

这次课程设计，让我有了很多收获，也让我懂了团队合作的重要性以及计算机行业的深奥。我们接触的只是皮毛，希望自己以后能够拥有更多的学习机会。