import java.awt.\*;

importjava.a wt.event.\*;

importjava.util.HashMap;

importjava.util.Map;

publicclassMovieBookingextendsFrame{

staticfinalintTICKET\_PRICE=200;//Ticketpriceperseat //Datastorageformovies,reservations,andpaymentstatusesstatic

Mapmovies=newHashMap<>();

staticMapreservations=newHashMap<>();

//GUIComponents

ChoicemovieChoice,timeChoice,paymentMethodChoice; TextFieldseatsField,reservationIdField,nameField,paymentDetailsField;

TextAre astatusArea;

ButtonbookButton,viewReservationButton,showMoviesButton,makePaymentButton; //Movieclasstostore

moviedetailsstaticclassMovie{

String name;String[]show Times;int[]available Seats; Movie(Stringname,String[]showTimes,int[]availableSeats)

this.name =name;

this.showTimes=show Times;

this.availableSeats=availableSeats;

}

}

//Reservationclasstostorereservationdetailsstatic

classReservation{

StringreservationId;

String movieName;

String showTime;

Int seatsBooked;

int amountToPay;

String customerName;

Boolean isPaid;

Reservation(StringreservationId,StringmovieName,StringshowTime,intseatsBooked, StringcustomerName) {

this.reservationId=reservationId;

this.movieName = movieName;

this.showTime=showTime;

this.seatsBooked=seatsBooked;

this. amountToPay=seatsBooked \*TICKET\_PRICE; //Calculatetotal amount

}

this.customerName=customerName;

this.isPaid=false; //Default:unpaid

}

//Constructortoinitialize

GUIpublic MovieBooking(){

setTitle("MovieTicketBookingSystem");

setSize(600,600);

setLayout(newFlowLayout());

//MovieSelection

add(newLabel("SelectMovie:"));

movieChoice = new Choice();

add(movieChoice);

//ShowTimeSelection

add(newLabel("SelectShowTime:"));

timeChoice = newChoice();

add(timeChoice);

//SeatsField

add(newLabel("EnterNumberofSeats:"));

seatsField=new TextField(10);

add(seatsField);

//CustomerName

add(newLabel("EnterYourName:"));

nameField = new TextField(20);

add(nameField);

// Buttons for booking and showing

moviesbookButton=newButton("BookTickets"); viewReservationButton=newButton("ViewReservation");

showMoviesButton=newButton("ShowAvailableMovies");add(bookButton); add(viewReservationButton); add(showMoviesButton); //ReservationID Field add(newLabel("EnterReservationID:"));r eservationIdField = new TextField(15);add(reservationIdField); //PaymentSection add(newLabel("PaymentMethod:"));paymen tMethodChoice = new Choice();paymentMethodChoice.add("Credi t Card");paymentMethodChoice.add("Debit Card");paymentMethodChoice.add("UPI"); paymentMethodChoice.add("Net Banking");add(paymentMethodChoice); add(newLabel("EnterPaymentDetails:"));p aymentDetailsField = new TextField(20);add(paymentDetailsField); makePaymentButton=newButton("MakePayment");add (makePaymentButton); //StatusArea statusArea=newTextArea(10,50);st atusArea.setEditable(false); 11 add(statusArea); //AddListeners bookButton.addActionListener(e- >bookTickets());viewReservationButton.addActionListener(e- >viewReservation());showMoviesButton.addActionListener(e- >showMovies());makePaymentButton.addActionListener(e->processPayment()); addWindowListener(new WindowAdapter() {publicvoidwindowClosing(WindowEvente){ System.exit(0); } }); //Initializemoviesi nitializeMovies(); } //Methodtoinitialize moviesandtheirshowtimesvoidinitializeMovies(){ movies.put("Black",newMovie("Black",newString[]{"10:00AM","1:00PM"," 4:00 PM"},newint[]{50,50,50})); movies.put("JurassicPark",newMovie("JurassicPark",newString[]{"11:00AM" ,"2:00 PM","5:00 PM"},newint[]{50,50,50})); movies.put("Veronica",newMovie("Veronica",newString[]{"12:00PM","3:00 PM","6:00 PM"},newint[]{50,50,50})); for(StringmovieName:movies.keySet()){movieChoice.a dd(movieName); 12 } movieChoice.addItemListener(e- >updateShowTimes());updateShowTimes();//Initializeshowtimesforthefir stmovie } //UpdateshowtimesinthetimeChoicedropdownbasedonselectedmovievoidup dateShowTimes() { timeChoice.removeAll(); StringselectedMovie=movieChoice.getSelectedItem();Mo viemovie =movies.get(selectedMovie); for(Stringtime:movie.showTimes){ti meChoice.add(time); } } //Methodtobookticketsvo idbookTickets(){ StringselectedMovie=movieChoice.getSelectedItem();Stri ng selectedTime = timeChoice.getSelectedItem();StringcustomerName = nameField.getText(); intseatsRequested; try{ seatsRequested=Integer.parseInt(seatsField.getText()); }catch(NumberFormatExceptione){ statusArea.setText("Invalidnumberofseats.Pleaseenteravalidnumber."); return; 13 } Moviemovie=movies.get(selectedMovie); intshowIndex=timeChoice.getSelectedIndex(); if (movie.availableSeats[showIndex] < seatsRequested) {statusArea.setText("Notenoughseatsavailableforthisshow.");re turn; } //GenerateauniquereservationID StringreservationId="RES"+(reservations.size()+1); Reservationreservation=newReservation(reservationId,selectedMovie,selectedTime,s eatsRequested,customerName); reservations.put(reservationId,reservation); //Deductseats movie.availableSeats[showIndex]-=seatsRequested; statusArea.setText("Bookingsuccessful!\nReservationID:"+reservationId +"\n"+ "Movie:"+selectedMovie+"\nShowTime:"+selectedTime+"\nSeats Booked:"+seatsRequested+ "\nAmounttoPay:₹"+reservation.amountToPay+"\nPayment Pending"); } //Methodtoprocesspaymentvo idprocessPayment(){ 14 StringreservationId=reservationIdField.getText(); if (!reservations.containsKey(reservationId)) {statusArea.setText("Reservationnotfound!Pleaseenteravalid ReservationID."); return; } Reservationreservation=reservations.get(reservationId);if( reservation.isPaid){ statusArea.setText("Paymentalreadycompletedforthisreservation.");return; } StringpaymentMethod=paymentMethodChoice.getSelectedItem();Stringp aymentDetails=paymentDetailsField.getText(); if(paymentDetails.isEmpty()){ statusArea.setText("Pleaseenterpaymentdetailstoproceed.");retu rn; } //Simulatepaymentprocessingr eservation.isPaid=true; statusArea.setText("PaymentSuccessful!\nReservationID:"+reservationId +"\n"+ "AmountPaid:₹"+reservation.amountToPay+"\nPaymentMethod: "+paymentMethod); 15 } //Methodtoviewreservationdetailsvo idviewReservation(){ StringreservationId=reservationIdField.getText(); if (!reservations.containsKey(reservationId)) {statusArea.setText("Reservationnotfound!");r eturn; } Reservationreservation=reservations.get(reservationId);statusArea.setText("Reservat ionDetails:\n"+ "ReservationID:"+reservation.reservationId+"\n"+"Movie:"+r eservation.movieName +"\n"+ "Show Time: " + reservation.showTime + "\n" +"SeatsBooked:"+reservation.seatsBooked+"\n"+ "Amount to Pay: ₹" + reservation.amountToPay + "\n" +"Customer Name: " + reservation.customerName + "\n" +"PaymentStatus:"+(reservation.isPaid ?"Paid":"Pending")); } //Methodtodisplayallavailablemoviesvoid showMovies() { StringBuilderdetails=newStringBuilder("AvailableMovies:\n");for( Moviemovie: movies.values()){ details.append("Movie:").append(movie.name).append("\n");