```java
//importend libraries
import java.util.ArrayList;
import java.util.Scanner;
import classes.*;

//Booking App class
public class BookingApp {

    //the mainMenu method call mainMenu
    public static String mainMenu(){

        //the format of the the main Menu
  System.out.println("University of Knowledge - COVID test");
  System.out.println();
  System.out.println("Manage Bookings");
  System.out.println();
  System.out.println("Please, enter the number to select your option: ");
  System.out.println();
  System.out.println("To manage Bookable Rooms: ");
  System.out.println(" 1. List ");
  System.out.println(" 2. Add ");
  System.out.println(" 3. Remove ");
  System.out.println("To manage Assistants on Shift: ");
  System.out.println(" 4. List ");
  System.out.println(" 5. Add ");
  System.out.println(" 6. Remove ");
  System.out.println("To manage Bookings: ");
  System.out.println(" 7. List");
  System.out.println(" 8. Add ");
  System.out.println(" 9. Remove ");
  System.out.println(" 10. Conclude ");
        System.out.println("After selecting one the options above, you will be presented other screens.");
        System.out.println("If you press 0, you will be able to return to this main menu.");
        System.out.println("Press -1 (or ctrl+c) to quit this application.");
        System.out.println("");

        //scanner for user input
  Scanner sc = new Scanner(System.in); //System.in is a standard input stream
  String user_input = sc.nextLine(); //reads string

        //this method will return the input from the user
  return user_input;
 }


/**
 * @param args
 */
public static void main(String[] args){
    //flag for calling the main menu
    boolean flag = true;

    //initializing system
    //rooms
    Room room1 = new Room("A1",3);
    Room room2 = new Room("B23", 5);
    Room room3 = new Room("C13", 2);
    //array of rooms
    Room[] rooms = {room1,room2,room3};
    //populate Univercity Resources
    UniversityResources.addRooms(rooms);

    //assistants
 Assistant as1 = new Assistant("jo97@uok.ac.uk", "John");
    Assistant as2 = new Assistant("cate123@uok.ac.uk", "Kate");

    Assistant as3 = new Assistant("geog2000@uok.ac.uk", "George");
    //array of assitants
    Assistant[] assistans = {as1,as2,as3};
    //populate Univercity Resources
    UniversityResources.addAssistans(assistans);

    //bookable rooms
    //bookable rooms
    BookableRoom br1 = new BookableRoom(1, "07:00", "25/02/2021", room1);// Availible
    BookableRoom br2 = new BookableRoom(3, "08:00", "26/02/2021", room1);// Full
    BookableRoom br3 = new BookableRoom(1, "09:00", "27/02/2021", room1);// Availible
    BookableRoom br4 = new BookableRoom(4, "07:00", "26/02/2021", room2);// Availible
    BookableRoom br5 = new BookableRoom(1, "08:00", "28/02/2021", room2);// Availible
    BookableRoom br6 = new BookableRoom(0, "09:00", "26/02/2021", room2);// Empty
    BookableRoom br7 = new BookableRoom(2, "07:00", "23/02/2021", room3);// Full
    BookableRoom br8 = new BookableRoom(1, "08:00", "23/02/2021", room3);// Availible
    BookableRoom br9 = new BookableRoom(0, "09:00", "21/02/2021", room3);// Empty
    //array of bookable rooms
    BookableRoom[] bookablerooms = {br1,br2,br3,br4,br5,br6,br7,br8,br9};
    //passing them to the bookable room array list in Booking System
    BookingSystem.populateBookableRooms(bookablerooms);

    //assistant on shift
    AssistantOnShift asos1 = new AssistantOnShift("25/02/2021", "FREE", as1);// Free
    AssistantOnShift asos2 = new AssistantOnShift("21/02/2021", "BUSY", as1);// Busy
    AssistantOnShift asos3 = new AssistantOnShift("23/02/2021", "FREE", as2);// Free
    AssistantOnShift asos4 = new AssistantOnShift("28/02/2021", "FREE", as2);// Free
    AssistantOnShift asos5 = new AssistantOnShift("23/02/2021", "BUSY", as3);// Busy
    AssistantOnShift asos6 = new AssistantOnShift("26/02/2021", "FREE", as3);// Free
    //array of assistants on shift
    AssistantOnShift[] assistantonshifts = {asos1,asos2,asos3,asos4,asos5,asos6};
    //passing them to the bookable room array list in Booking System
    BookingSystem.populateAssistantOnShifts(assistantonshifts);

    //main while loop for calling the main menu
  while (flag == true) {
        //calling main menu
   String user_input = BookingApp.mainMenu();

        //if -1 is pressed then systme exists
   if (user_input.equals("-1")) {
    System.out.println("Quit");
    System.exit(0);
   }

        //if 1 is pressed we go to the next page
   if (user_input.equals("1")) {
            //format of the 1st page
    System.out.println("University of Knowledge - COVID test");
            System.out.println();

            //print all bookable rooms
            for (BookableRoom br:BookingSystem.getBookableRooms()){
                br.printThe();
            }

            //formating the rest of the page
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag to continue calling until 0 or -1 pressed
            boolean f = true;
            //loop for calls
            while (f == true){
                //getting users input
                Scanner sc = new Scanner(System.in);
                String us_in = sc.nextLine();

                //if 0 then exit from page
       if (us_in.equals("0")){
        f = false;
                //if -1 then exit from the system
       } else if (us_in.equals("-1")){
        System.exit(0);
                //otherwise an error
       } else {
                    System.out.println("Error! Please enter a vaild input");
                    flag = true;
                }
        }
}

      //second page
if (user_input.equals("2")){
        //format of this page
        System.out.println("University of Knowledge - COVID test");
        System.out.println();
        System.out.println("Adding bookable room");
        System.out.println();
        System.out.println(" Please, enter one of the following: ");
        System.out.println();

        //printnig out all rooms of the system
        int k = 1;
        for (Room r : rooms) {
            System.out.print(k + ". ");
            r.printThe();
            k = k + 1;
        }

        //rest of the format of the page
        System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
        System.out.println("separated by a white space.");
        System.out.println("0. Back to main menu.");
        System.out.println("-1. Quit application.");
        System.out.println();

        //flag for continue calling
        boolean new_flag = true;

        //the loop with the flag
        while (new_flag == true) {

            //getting user input
            Scanner sc = new Scanner(System.in);
            String[] us_in = sc.nextLine().split(" ");

            //exit to main menu
            if (us_in[0].equals("0")){
                new_flag = false;

            //exit the system
            } else if (us_in[0].equals("-1")){
                System.exit(0);
            }

            //try to add a bookable room
            try {
                //if statment to control user input
                if (new_flag == true){
                    //converting the id to int

                    int i = Integer.parseInt(us_in[0]);

                    //validating users input
                    BookingSystem.validateTime(us_in[2]);
                    BookingSystem.validateDate(us_in[1]);

                    //adding the room
                    //if invalid data is entered, stystem prints out the catch
                    BookingSystem.addBookableRoom(0, us_in[2], us_in[1], rooms[i-1]);

                    //formating of the suuccess page
                    System.out.println("Bookable Room added successfully");
                    //printing out the new bookable room
                    System.out.println(BookingSystem.getBookableRooms().get(BookingSystem.getBookableRooms().size()-1).printThe());
                    System.out.println("Please, enter one of the above!");
                    System.out.println();
                    System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
                    System.out.println("separated by a white space.");
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();
```

```
                        System.out.println();
                    }else{
                        System.out.println();
                    }

                //catch if an error occures
                } catch(Exception e) {
                    //format of a fail page
                    System.out.println("Error!");
                    System.out.println("This input is invaild, please follow the correct syntax");
                    System.out.println("Please, enter one of the above!");
                    System.out.println();
                    System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
                    System.out.println("separated by a white space.");
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();

                }
            }
        }

        //third page
        if (user_input.equals("3")){

            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Removing bookable room");
            System.out.println();
            System.out.println("Please, enter one of the following:");
            System.out.println();

            //printing out all bookabel rooms that can be removed
            for (BookableRoom br : BookingSystem.getBookableRooms()){
                if (br.findOutStatusRoom() == "EMPTY"){
                    int indx = BookingSystem.getBookableRooms().indexOf(br);
                    System.out.print(indx + 1 + ". ");
                    System.out.println(br.printThe());
                }
            }

            //rest of the formatting
            System.out.println("The sequential ID listed to a room, a date (dd/mm/yyyy), and a time (HH:MM),");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean new_new_flag = true;

            //loop with the flag
            while (new_new_flag == true){

                //getting user input
                Scanner sc = new Scanner(System.in);
                String us_in = sc.nextLine();

                //exit to main menu
                if (us_in.equals("0")){
        new_new_flag = false;

                //exit the system
       } else if (us_in.equals("-1")){
        System.exit(0);
                }

                //try to remove the bookable room
                try{
                    //if statment to control users input
                    if (new_new_flag == true) {
                        //converting user input to integer
                        int i = Integer.parseInt(us_in);
                //format of the success page
                System.out.println("Bookable Room removed successfully:");
                //printing the element which will be removed
                System.out.println(BookingSystem.getBookableRooms().get(i-1).printThe());

                //removing the room which was picked
                BookingSystem.removeBookableRoom(BookingSystem.getBookableRooms().get(i-1));

                //rest of the format
                System.out.println("Please, enter one of the above!");
                System.out.println();
                System.out.println("The sequential ID to select the bookable room to be removed.");
                System.out.println("0. Back to main menu.");
                System.out.println("-1. Quit application.");
                System.out.println();
            } else{
                System.out.println();
            }

        // if error occures
        } catch (Exception e){
            //format of the fail page
            System.out.println("Error!");
            System.out.println("This input is invaild, please follow the correct syntax");
            System.out.println("Please, enter one of the above!");
            System.out.println();
            System.out.println("The sequential ID to select the bookable room to be removed.");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

        }
    }
}

//forth page
if (user_input.equals("4")){

    //format of the page

    System.out.println("University of Knowledge - COVID test");
    System.out.println();

    //printing the assistants on shift
    for (AssistantOnShift asos: BookingSystem.getAssistantOnShifts()){
        asos.printThe();
    }

    //rest of the format
    System.out.println("0. Back to main menu.");
    System.out.println("-1. Quit application.");
    System.out.println();

    //flag for continious calls
    boolean fla = true;
    while (fla == true){

        //getting user input
        Scanner sc = new Scanner(System.in);
        String us_in = sc.nextLine();

        //exit to main menu
if (us_in.equals("0")){
 fla = false;
        //exit the system
} else if (us_in.equals("-1")){
 System.exit(0);
        //otherwise error
} else {
            System.out.println("Error! Please enter a vaild input");
            fla = true;
        }
    }
```

```java
    }

        //fifth page
        if (user_input.equals("5")){

            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Adding assistant on shift");
            System.out.println();
            System.out.println("Please, enter one of the following: ");
            System.out.println();

            //printing all assistans of the system
            int k = 1;
            for (Assistant a : assistans) {
                System.out.print(k + ". ");
                System.out.println(a.printThe());
                k = k +1;

            }

            //rest format of the page
            System.out.println("The sequential ID of an assistant and date (dd/mm/yyyy), separated by a white space.");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean new_flg = true;

            //loop with the flag

            while (new_flg == true) {

                //getting user input
                Scanner sc = new Scanner(System.in);
                String[] us_in = sc.nextLine().split(" ");

                //exit to main menu
                if (us_in[0].equals("0")){
                    new_flg = false;

                //exit the system
                } else if (us_in[0].equals("-1")){
                    System.exit(0);
                }

                //try to add assistant on shift
                try{
                    //if statment to control user input
                    if (new_flg == true){
                        //coverting the user input
                        int i = Integer.parseInt(us_in[0]);

                        //valiadting the user input (date)
                        BookingSystem.validateDate(us_in[1]);

                        //adding the assistat on shift
                        BookingSystem.addAssistanOnShift(us_in[1], "FREE", assistans[i-1]);

                        //format of the page
                        System.out.println("Assistant on Shift added successfully:");

                        //print the assistant on shift
                        System.out.println(BookingSystem.getAssistantOnShifts().get(BookingSystem.getAssistantOnShifts().size()-1).printThe());
                        System.out.println("Please, enter one of the above!");
                        System.out.println();
                        System.out.println("The sequential ID of an assistant and date (dd/mm/yyyy), separated by a white space.");
                        System.out.println("0. Back to main menu.");
                        System.out.println("-1. Quit application.");
                        System.out.println();
                    } else {
                        System.out.println()
                        System.out.println();
                    }
                //if error occures
                } catch (Exception e){
                    //format of the fail page
                    System.out.println("Error!");
                    System.out.println("This input is invaild, please follow the correct syntax");
                    System.out.println("Please, enter one of the above!");
                    System.out.println();
                    System.out.println("The sequential ID of an assistant and date (dd/mm/yyyy), separated by a white space.");
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();

                }
            }
        }

        //sixth page
        if (user_input.equals("6")){

            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Removing assistant on shift");
            System.out.println();

            System.out.println("Please, enter one of the following:");
            System.out.println();

            //printing assistans on shift that can be removed
            for (AssistantOnShift a : BookingSystem.getAssistantOnShifts()){
                if (a.findOutStatusAssistan() == "FREE"){
                    int indx = BookingSystem.getAssistantOnShifts().indexOf(a);
                    System.out.print(indx + 1 + ". ");
                    a.printThe();
                }
            }

            //rest format of the page
            System.out.println("The sequential ID to select the bookable room to be removed.");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean new_new_flg = true;

            //loop with the flag
            while (new_new_flg == true){

                //getting user input
                Scanner sc = new Scanner(System.in);
                String us_in = sc.nextLine();

                //exit to main menu
                if (us_in.equals("0")){
        new_new_flg = false;

                //exit the sytem
       } else if (us_in.equals("-1")){
        System.exit(0);
                }
                //try to remove assistant
                try{
                    //if statement to control user input
                    if (new_new_flg == true){
                        //converting user input to int
                        int i = Integer.parseInt(us_in);
                        //printing out the assitatnt on shif that will be removed
                        System.out.println("Assistant on Shift removed successfully:");
                        System.out.println(BookingSystem.getAssistantOnShifts().get(i-1).printThe());

                        //removig the assitant on shift
                        BookingSystem.removeAssistanOnShift(BookingSystem.getAssistantOnShifts().get(i-1));

                        System.out.println("Please, enter one of the above!");
                        System.out.println();
                        System.out.println("The sequential ID to select the assistant on shift to be removed.");
                        System.out.println("0. Back to main menu.");
                        System.out.println("-1. Quit application.");
                        System.out.println();
                    }else{
                        System.out.println();
                    }

                //if error occures
                } catch (Exception e){
                    //format of the fail page
                    System.out.println("Error!");
                    System.out.println("This input is invaild, please follow the correct syntax");
                    System.out.println("Please, enter one of the above!");
                    System.out.println();

                    System.out.println("The sequential ID to select the assistant on shift to be removed.");
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();
                }

            }
        }

        //seventh page
        if (user_input.equals("7")){
            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Select which booking to list:");
            System.out.println("1. ALL");
            System.out.println("2. Only bookings status:SCHEDULED");
            System.out.println("3. Only bookings status:COMPLETED");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean nw_flg = true;

            //loop with the flag
            while (nw_flg == true){

                //getting user input
                Scanner sc = new Scanner(System.in);
                String us_in = sc.nextLine();

                //exit to main menu
       if (us_in.equals("0")){
        nw_flg = false;

                //exit the system
       } else if (us_in.equals("-1")){
        System.exit(0);

                //print all scheduled bookings
                } else if (us_in.equals("2")){
                    BookingSystem.showSCHEDULEDBookings();
                    nw_flg = true;
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();

                //print all completed bookings
                } else if (us_in.equals("3")){
                    BookingSystem.showCOMPLETEDBookings();
                    nw_flg = true;
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println("-1. Quit application.");
                    System.out.println();

                //print all bookings
                } else {
                    BookingSystem.showALLBookings();
                    nw_flg = true;
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();
                }
            }
        }


        //eighth page
        if (user_input.equals("8")){
            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Adding booking (appointment for a COVID test) to the system");
            System.out.println();
            System.out.println("List of available time-slots:");
            System.out.println();
            System.out.println("Please, enter one of the following:");
            System.out.println();

            //4 lists to search for availible time-slots
            ArrayList<AssistantOnShift> listAssistant = new ArrayList<AssistantOnShift>();
            ArrayList<BookableRoom> listBookableRooms = new ArrayList<BookableRoom>();

            ArrayList<AssistantOnShift> listAssistantFinal = new ArrayList<AssistantOnShift>();
            ArrayList<BookableRoom> listBookableRoomsFinal = new ArrayList<BookableRoom>();

            //find all bookable rooms which are empty or availible
            for (BookableRoom b : BookingSystem.getBookableRooms()){
                String statusBooking = b.findOutStatusRoom();
                if (statusBooking == "EMPTY" || statusBooking == "AVAILABLE"){
                    listBookableRooms.add(b);
                }
            }

            //find all assistatns which are free
            for (AssistantOnShift a : BookingSystem.getAssistantOnShifts()){
                String statisAssistant = a.findOutStatusAssistan();
                if (statisAssistant == "FREE"){
                    listAssistant.add(a);
                }
            }

            //look where both lists have same dates
            //meaning finding free time spaces
            int indx = 1;
            for (BookableRoom b : listBookableRooms){
                int c = b.carryCampacity();
                for (int i = 0; i <= c; i++){
                    for (AssistantOnShift a :listAssistant){
                        if (b.getDate().equals(a.getDate())){
                            System.out.println(indx + ". " + a.getDate() + " " + b.getTime());
                            listAssistantFinal.add(a);
                            listBookableRoomsFinal.add(b);
                            indx = indx + 1;
                        }
                    }
                }
            }

            //rest format of the page
            System.out.println("The sequential ID of an available time-slot and the student email, separated by a white space.");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean nw_flag = true;
            boolean nw_flag = true;

            //loop with the flag
            while (nw_flag == true) {

                //getting user input
                Scanner sc = new Scanner(System.in);
                String[] us_in = sc.nextLine().split(" ");

                //exit to main menu
                if (us_in[0].equals("0")){
                    nw_flag = false;

                //exit the system
                } else if (us_in[0].equals("-1")){
                    System.exit(0);
                }

                //try adding the booking
                try {
                    //if statment to control user input
                    if (nw_flag == true) {
                        //put user input to correct format
                        int i = Integer.parseInt(us_in[0]);
                        String studentEmail = us_in[1];

                        //validate the student email
                        BookingSystem.validateStudentEmail(studentEmail);

                        //get the correct assistant on shift and bookable room
                        BookableRoom br = listBookableRoomsFinal.get(i-1);
                        AssistantOnShift as = listAssistantFinal.get(i-1);

                        //create the booking
                        BookingSystem.addBooking(as, br, br.getTime(), studentEmail, "SCHEDULED");
                        System.out.println("Booking added successfully:");

                        //print that booking
                        System.out.println(BookingSystem.getBookings().get(BookingSystem.getBookings().size()-1).printThe());
                        System.out.println();
                        System.out.println("Please, enter one of the above!");
                        System.out.println();
                        System.out.println("The sequential ID of an available time-slot and the student email, separated by a white space.");
                        System.out.println("0. Back to main menu.");
                        System.out.println("-1. Quit application.");
                        System.out.println();
                    } else {
                        System.out.println();
                    }

                //if error occures
                } catch (Exception e) {
                    //print the fail form
                    System.out.println("Error!");
                    System.out.println("This input is invaild, please follow the correct syntax");
                    System.out.println("Please, enter one of the above!");
                    System.out.println();
                    System.out.println("The sequential ID of an available time-slot and the student email, separated by a white space.");
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();
                }
            }
        }

        //ninth page
        if (user_input.equals("9")){

            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Removing booking from the system");

            System.out.println();
            System.out.println();
            System.out.println("Please, enter one of the following:");
            System.out.println();

            //print all bookings that can be removed
            for (Booking b: BookingSystem.getBookings()){
                String status = b.findOutStatusBooking();
                if (status == "SCHEDULED"){
                    int k = BookingSystem.getBookings().indexOf(b);
                    System.out.print(k + 1 + ". ");
                    System.out.println(b.printThe());
                    k = k + 1;
                }
            }

            //rest format of the page
            System.out.println("The sequential ID to select the booking to be removed from the listed bookings above.");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean new_new_flg = true;

            //loop with the flag
            while (new_new_flg == true){

                //getting user input
                Scanner sc = new Scanner(System.in);
                String us_in = sc.nextLine();

                //exit to main menu
                if (us_in.equals("0")){
        new_new_flg = false;

                //exit the system
       } else if (us_in.equals("-1")){
        System.exit(0);
                }

                //try removing the booking
                try{
                    //if statment ot control user input
                    if (new_new_flg == true){
                        //convert user input to integer
                        int i = Integer.parseInt(us_in);

                        //print the booking which will be removed
                        System.out.println("Booking removed successfully:");
                        System.out.println(BookingSystem.getBookings().get(i-1).printThe());

                        //remove booking
                        BookingSystem.removeBooking(BookingSystem.getBookings().get(i-1));

                        //rest of the format
                        System.out.println("Please, enter one of the above!");
                        System.out.println();
                        System.out.println("The sequential ID to select the booking to be removed from the listed bookings above.");
                        System.out.println("0. Back to main menu.");
                        System.out.println("-1. Quit application.");
                        System.out.println();
                    }else{
                        System.out.println();
                    }
                //if error occures
                } catch (Exception e){

                    //print the fail form
                    System.out.println("Error!");
                    System.out.println("This input is invaild, please follow the correct syntax");
                    System.out.println();
                    System.out.println("Please, enter one of the above!");
                    System.out.println();
                    System.out.println("The sequential ID to select the booking to be removed from the listed bookings above.");
                    System.out.println("0. Back to main menu.");
```

```java
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();

                }
            }
        }

        //tenth page
        if (user_input.equals("10")){

            //format of the page
            System.out.println("University of Knowledge - COVID test");
            System.out.println();
            System.out.println("Conclude booking");
            System.out.println();
            System.out.println("Please, enter one of the following:");
            System.out.println();

            //print the bookings which can be concluded
            for (Booking b: BookingSystem.getBookings()){
                String status = b.findOutStatusBooking();
                int k = BookingSystem.getBookings().indexOf(b);
                if (status == "SCHEDULED"){
                    System.out.print(k+1 + ". ");
                    System.out.println(b.printThe());
                    k = k + 1;
                }
            }

            //rest format of the page
            System.out.println("The sequential ID to select the booking to be completed.");
            System.out.println("0. Back to main menu.");
            System.out.println("-1. Quit application.");
            System.out.println();

            //flag for continious calls
            boolean new_new_flg = true;

            //loop with the flag
            while (new_new_flg == true){

                //getting user input
                Scanner sc = new Scanner(System.in);
                String us_in = sc.nextLine();

                //exit to main menu
                if (us_in.equals("0")){
        new_new_flg = false;

                //exit the system
       } else if (us_in.equals("-1")){
        System.exit(0);
                }

                //try to conclude booking
                try{
                    //if statment to control user input
                    if (new_new_flg == true){

                        //convert user input to integer
                        int i = Integer.parseInt(us_in);
                        //change status to completed
                        BookingSystem.getBookings().get(i-1).setStatus("COMPLETED");
                        //print the form and the booking which is now concluded
                        System.out.println("Booking completed successfully:");
                        System.out.println(BookingSystem.getBookings().get(i-1).printThe());
                        System.out.println("Please, enter one of the above!");
                        System.out.println();
                        System.out.println("The sequential ID to select the booking to be completed.");
                        System.out.println("0. Back to main menu.");
                        System.out.println("-1. Quit application.");
                        System.out.println();

                    }else{
                        System.out.println();
                    }

                //if error occures
                } catch (Exception e){
                    //print the fail form
                    System.out.println("Error!");
                    System.out.println("This input is invaild, please follow the correct syntax");
                    System.out.println("Please, enter one of the above!");
                    System.out.println();
                    System.out.println("The sequential ID to select the booking to be completed.");
                    System.out.println("0. Back to main menu.");
                    System.out.println("-1. Quit application.");
                    System.out.println();
                }
            }
        }

    //way to exit from the main menu
 if (flag == false)
  break;
 }

}
}
```