Kingdom of Saudi Arabia
Ministry of Education
Imam Abdulrahman bin Faisal University
College of Sciences and Humanities in Jubail
Computer Science Department

المملكة العربية السعودية
وزارة التعليم
جامعة الإمام عبدالرحمن بن فيصل
كلية العلوم و الدراسات الإنسانية بالجبيل
قسم علوم الحاسب الآلي

Data Structure - Level 6

Dr. Enas EL. Sharawy

T.A. Hessa Almutairi

Kingdom of Saudi Arabia
Ministry of Education
Imam Abdulrahman bin Faisal University
College of Sciences and Humanities in Jubail
Computer Science Department

جامعة الإمام عبدالرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

المملكة العربية السعودية
وزارة التعليم
جامعة الإمام عبدالرحمن بن فيصل
كلية العلوم و الدراسات الإنسانية بالجبيل
قسم علوم الحاسب الآلي

Project Goal:

Designing a program using the Java language that helps users to review the available matches and see the dates of their performances. The reserved tickets can be reviewed and access to a specific match. Managers can also manage all reservations and access for users. There is a section to organize entry, through which it is possible to review the reserved places and tickets for each match.

Project Requirements:

Project proposal:

project title and idea:

Matches Reservation program was designed to organize matches seats and tickets at specific dates and times.

The goal of your program:

We solved the problem organizing reservations for users who want to attend matches has been resolved by giving users the ability to choose the match they want to watch, choose the appropriate time and day for them, and then choose certain seats to be reserved, and the amount incurred by them and the card number that was reserved for this match is displayed.

Main characters:

We have three types of users:

- Regular Users: can use the following services:
  - ✔ Registration within the program.
  - ✔ Browse available matches.
  - ✔ Browse the special shows for these matches.
  - ✔ Reservation of tickets and reservation of seats.
  - ✔ Browse reserved tickets.
  - ✔ Entering a certain seat within a match.
- Admins: can use the following services:
  - ✔ Browse available matches.
  - ✔ View reserved and unreserved seats for each match.
  - ✔ Browse reserved tickets.
  - ✔ Browse seats reserved for a specific ticket
  - ✔ Allow users to enter a specific match
- Organizers: can use the following services:
  - ✔ Browse available matches.
  - ✔ View reserved and unreserved seats for each match.
  - ✔ Browse seats reserved for a specific ticket

Design the program:

We have designed the program using Object-oriented programming (OOP). We created class for each entity in the program. And we customized functions to each class.

We used Data Structure for building block of programs.

1. Array: declared an array like int[]reservedSeats, which is an array of primitive int types to store the index of reserved seats.
2. LinkedList: is dynamic array to store tickets like List<Ticket> AllTickets.
3. 2d array: to store match shows like MatchShows[][] AllMatchsShows.

Solution to solve the problem:

Designing the main classes:

1- Match Class: containing the main attributes of match as the name of match and the index of match in the list of All Matches. We used get and set modifiers to encapsulate fields. We also use constructor and some functions in this class.

```
public class Match {
    private int matchIndex;
    private String matchName;
    public Match() {
    }
    public Match(int matchIndex, String matchName) {
        this.matchIndex = matchIndex;
        this.matchName = matchName;
    }
    public int getMatchIndex() {
        return matchIndex;
    }
    public void setMatchIndex(int matchIndex) {
        this.matchIndex = matchIndex;
    }
    public String getMatchName() {
        return matchName;
    }
    public void setMatchName(String matchName) {
        this.matchName = matchName;
    }
   public void print()
   {
      System.out.println("Index: "+matchIndex +" Name: "+matchName)
   }
}
```

2- Match Shows: containing the main attributes of shows as the date and time that match available at and list containing seats customized for this show. It also contains functions return or print available and not available seats for this show.

```java
public class MatchShows {
    private int showIndex;
    private String date;
    private String time;
    private List<Seat> showSeats;
    public MatchShows() {
    }
    public MatchShows(int showIndex, String date, String time, List<Seat> showSea
        this.showIndex = showIndex;
        this.date = date;
        this.time = time;
        this.showSeats = showSeats;
    }
    public void print()
    {
       System.out.println("-Show Index: "+showIndex +" in: "+date +" at " +time);
    }
     public void printAvailableSeats()
    {
       List<Seat> availableSeats=getAvailable();
       for (int i = 0; i < availableSeats.size(); i++) {
           availableSeats.get(i).print();
       }
    }
   public List<Seat>  getAvailable()
    {
       List<Seat> availableSeats=new ArrayList<>();
       for (int i = 0; i < showSeats.size(); i++) {
           if(showSeats.get(i).isIsAvaialble())
           {
               availableSeats.add(showSeats.get(i));
           }
```

3- Seat Class: containing the main attributes of seat as the name and if the seat is available or reserved. It also contains functions print seat's information.

```java
public class Seat {
    private int seatIndex;
    private String seatName;
    private boolean isAvaialble=true;

    public Seat() {
    }

    public Seat(int seatIndex, String seatName, boolean isAvaialble)
        this.seatIndex = seatIndex;
        this.seatName = seatName;
        this.isAvaialble = isAvaialble;
    }
```

4- Reservation Class: containing the main attributes of reservation such as the index of reserved seat and reserved show, if this reservation allowed to enter the user to it or not.

```
public class Reservation {
    private int reservationIndex;
    private int ticketIndex;
    private int seatIndex;
    private boolean allowUser;
    public Reservation() {
    }
    public Reservation(int reservationIndex, int ticketIndex, int se
        this.reservationIndex = reservationIndex;
        this.ticketIndex = ticketIndex;
        this.seatIndex = seatIndex;
        this.allowUser = allowUser;
    }
    public int getReservationIndex() {
        return reservationIndex;
    }
    public void setReservationIndex(int reservationIndex) {
        this.reservationIndex = reservationIndex;
    }
    public int getTicketIndex() {
        return ticketIndex;
    }
    public void setTicketIndex(int ticketIndex) {
        this.ticketIndex = ticketIndex;
    }
    public int getSeatIndex() {
        return seatIndex;
    }
    public void setSeatIndex(int seatIndex) {
```

5- Ticket Class: containing the main attributes of ticket such as the index of ticket, the total payment, the id of match and show, the list of reservation entered in this ticket.

```
public class Ticket {
    private int ticketIndex;
    private String TicketNumber;
    private int payment;
     private int userId;
     private int matchIndex;
     private int showIndex;
    public List<Reservation> AllReservations;
    public Ticket() {
    }
    public Ticket(int ticketIndex, String TicketNumber, int payment,int userId,in
        this.ticketIndex = ticketIndex;
        this.TicketNumber = TicketNumber;
        this.payment = payment;
        this.userId=userId;
        this.matchIndex=matchIndex;
        this.showIndex=showIndex;
    }
    public int getTicketIndex() {
        return ticketIndex;
    }
    public void setTicketIndex(int ticketIndex) {
        this.ticketIndex = ticketIndex;
    }
    public String getTicketNumber() {
        return TicketNumber;
    }
    public void setTicketNumber(String TicketNumber) {
        this.TicketNumber = TicketNumber;
    }
    public int getPayment() {
```

Main Flow of the Program:

When the program is run it displays three options to enter the user to it:

- Login as regular user.
- Login as Admin.
- Enter program as organizer.

```
.............WELCOME TO Match Reservation PROGRAM.............
enter your Choice
 1: login as User
 2: login as Admin
 3: Match Organizers
 4: exit program?
```

After enter your choice you can browse each type of user menu:

Regular User Menu:

```
.............User Menu.............
1-View Matchs
2-Register
3-Reserve Match
4-Show Reserved Match
5-enter match
6-Return to Home Menu
Your option :
```

Admin Menu:

```
.............Admin Menu.............
1-View Matchs
2-Reservations
3-Show Ticket Seats
4-Show Match Details
5-enter user to match
6-Return to Home Menu
Your option :
```

Organizers Menu:

```
.............Organizers Menu.............
1-View Matchs
2-Reservations
3-Show Ticket Seats
4-Return to Home Menu
Your option :
```

For each menu we used switch case to process user's options.

Like this:

```
String Main_Menu = "\n.............User Menu.............\n"
        + "1-View Matchs\n"
        + "2-Register \n"
        + "3-Reserve Match \n"
        + "4-Show Reserved Match \n"
        + "5-enter match\n"
        + "6-Return to Home Menu \n"
        + "Your option : ";
while (choice != 6) {
    System.out.println(Main_Menu);
    choice = reader.nextInt();
    switch (choice) {
        case 1: {
            ViewMatchs();
            break;
        }
        case 2: {
            Register();
            break;
        }
        case 3: {
            ReserveMatch();
            break;
        }
        case 4: {
            ShowReserveMatch();
            break;
        }
        case 5: {
            EnterMatch();
            break;
        }
```

We also process the errors of the users such as entering index not exists or entering number more than the available size of the seats. We check if the user entering number of reserved seat and show warning message.