

Show Booking System

You are required to build an application that lets users book for Live Shows. The day is divided into time slots of 1 hour each, starting from 9 am to 9 pm. Organizers of Live Shows can login to the portal and declare their show timings with the capacity for the given day. Users can login and book tickets for a particular live show / cancel existing bookings.

For simplicity you can assume that

- The Live Shows availability is declared for that particular day only.

Functionalities required:

- An organizer should be able to register new live shows, and mention genres among (Comedy, Theatre, Tech, Singing.)
- An organizer should be able to declare show's timings in each slot for the day. For example, the slots will be of 1 hour like 9am-10am, 10am-11am..
- Users should be able to login(Optional), and search available shows timings based on genre.
 - The show's timings should be displayed in a ranked fashion. Default ranking strategy should be to rank by start time. But we should be able to plugin more strategies like show's review etc in future.
- Users should be able to book 1 ticket for the available slot of show. A user can book multiple show tickets in a day.
 - **A user cannot book two show tickets with two different shows in the same time slot.**
 - **1 Ticket can contain multiple persons entry. It can not be partially booked (if available capacity is 2, one booking ticket requests for 3 persons entry then it can not be booked partially)**
- Users can also cancel a booking, in which case that slot becomes available for someone else to book.
- Build a waitlist feature:
 - If the user wishes to book a slot for a particular live show that is already booked, then add this user to the waitlist. If the user with whom the ticket is booked originally, cancels the booking, then the first in the waitlist gets the booking.
- A user/organizer should be able to view his/her bookings for the day.
- Overlapping slots cannot be provided for a particular show. But there could be running different parallel shows irrespective of the genre
- Implementing login feature is optional
- User registration is not mandatory
- Name of Live Show and UserName are their identifiers

Bonus functionality:

- Trending Live Show: Maintain at any point of time which live show has the most tickets booked.

Examples:

The input/output need not be exactly in this format but the functionality should remain intact

i: input o: output

i: registerShow: TMKOC -> Comedy

o: TMKOC show is registered !!

i: onboardShowSlots: TMKOC 9:00-11:00

o: Sorry, show timings are of 1 hour only

i: onboardShowSlots: TMKOC 9:00-10:00 3, 12:00-13:00 2, 15:00-16:00 5

o: Done!

i: registerShow: The Sonu Nigam Live Event -> Singing

o: The Sonu Nigam Live Event show is registered !!

i: onboardShowSlots: The Sonu Nigam Live Event 10:00-11:00 3, 13:00-14:00 2, 17:00-18:00 1

o: Done!

i: showAvailByGenre: Comedy

o: TMKOC: (9:00-10:00) 3

o: TMKOC: (12:00-13:00) 2

o: TMKOC: (15:00-16:00) 5

i: bookTicket: (UserA, TMKOC, 12:00, 2)

o: Booked. Booking id: 1234

O: Booking Id : 2345, Wait listing

i: showAvailByGenre: Comedy

o: TMKOC: (9:00-10:00) 3

o: TMKOC: (15:00-16:00) 5

i: cancelBookingId: 1234

o: Booking Canceled

i: showAvailByGenre: Comedy

o: TMKOC: (9:00-10:00) 3

o: TMKOC: (12:00-13:00) 2

o: TMKOC: (15:00-16:00) 5

i: bookTicket: (UserB, TMKOC, 12:00, 1)

o: Booked. Booking id: 5678

i: registerShow: The Arijit Singh Live Event -> Singing

o: The Arijit Singh Live Event show is registered !!
i: onboardShowSlots: The Arijit Singh Live Event 11:00-12:00 3, 14:00-15:00 2
o: Done!
i: showAvailByGenre: Singing
o: The Sonu Nigam Live Event: (10:00-11:00) 3
o: The Arijit Singh Live Event: (11:00-12:00) 3
o: The Sonu Nigam Live Event: (13:00-14:00) 2
o: The Arijit Singh Live Event: (14:00-15:00) 2
o: The Sonu Nigam Live Event: (17:00-18:00) 1

Guidelines:

- Time: 90mins
- Write modular and clean code.
- **A driver program/main class/test case is needed to test out the code by the evaluator with multiple test cases.** But do not spend too much time in the input parsing. Keep it as simple as possible.
- Evaluation criteria: Demoable & functionally correct code, Code readability, Proper Entity modeling, Modularity & Extensibility, Separation of concerns, Abstractions. Use design patterns wherever applicable
- You are not allowed to use any external databases like MySQL. Use only in memory data structures.
- No need to create any UI
- Please focus on the Bonus Feature only after ensuring the required features are complete and demoable.