

Web Applications – Exam #3 (deadline 2024-09-13 at 23:59)

“Concert Seats”

Question and answers already given in the google doc

1. So, do we need to store the size in the database as 'small,' 'medium,' or 'large,' or is that information not relevant?

That information should be stored in the database with the size, so that you can provide the name as well as the sizes.

2. In this database, should we store the dimension as text (small/medium/large) or using integers? The latter is more flexible

The database is up to you and subject to evaluation. However, considering that you must be able to add new sizes, the latter is, in practice, required, since you need to store the sizes anyway to avoid modifying the code.

3. So, the admin could add new sizes that are not small, medium, large?

Yes. This is done offline, for simplicity, directly by the developer, by ONLY modifying the database. Doing it by the admin requires coding much more and is not required.

4. A possible problem is to go beyond 26 columns because you run out of letters in the alphabet. Should we take this problem into account or assume that the developer will never enter in the db a theatre with more than 26 columns?

For simplicity do not care about more than 26 columns. It is ok to assume that the developer will never enter a theater with more than 26 columns.

5. "with its size" it means that if a theater is added it will be small(4,8) or medium(6,10) or large(9,14) and not a personalized size? in the sense that small,medium, large are standardized sizes and we do not need to think of theaters with different measures.

You do need to be able to add theaters with different sizes

6. so for example a theater with r=5 and c=7?

Yes, for example.

7. So, we ONLY select the concert, the theaters are not to be chosen. Am I correct?

Yes. The theater associated with the concert is already defined in the db and cannot be changed/chosen

8. Is it ok if show a list of the theaters and for each theater we show the concerts for that theater?

The text says the user should select a concert

9. In the same page where it is possible to select a concert, right?

Not necessarily

10. it means that we have to implement a part of the page where it says the seats stats or we can leave it to the user to discern by looking at the color of the seat in the 2d view where all seat are rapresented?

You need to implement a part of the page with the seat stats

11. Should we implement both methods or only one?

Both are mandatory

12. the reservation can be made in a different page (like a page for the first method and another for the second one) or the same page of the 2d visualization?

No. The text says "in THAT visualization", so it is the same.

12b. when the user writes in the form the number of seats, he must see in real time the seats requested (which one is decided by the client app) in the matrix in real time? In other words, the client app sends the array of seats to be booked regardless of the booking mode to the server or it can send the number of seats to be booked to the server and only then does the server decide which seats to book?

12c. so is it wrong to put the second method in the 2d view page and then add a button to open a form in another page for the first method, as we saw in class during the lectures with the button to add a new answer?

1) With the first method, the user does not have to see the seats requested: in fact, nowhere the text says that the seats need to be required by the client, in that modality

2) Here the "view" (as you call it) is the same. This is just a functional requirement for testing convenience. Since you have React it is not that difficult to make things appear and disappear on the same page. What we saw in class during the lectures is just one of the possibilities (indeed when we had no routes we did not use another page).

13. The current user's reservation after the confirmation or after the automatic request will appear for him as requested (yellow) or as occupied (red)? The current user should see his own reservation in red as the other ones or it should remain yellow?

Red. It must not remain yellow.

13b. in the first method the user specify only the amount of the the seats that he wants , not the number of the seat rappedresented by its code, correct? like "4 seats" and not "4c,2b,3a,5f"

Yes

13c. So, it's okay if a user, with their view, selects random seats by entering the number, and the seats are randomly or logically selected. Then a request is made to the server to check their status. Therefore, is it acceptable that even with a random selection, some seats may already be occupied by others?
(and so use the blue seats strategy)?

You seem to talk about the second method (while the comment is attached to the first). For the first method, the user does not select seats, it is the server that selects the seats. Random (as all other methods etc.) means "among the available", otherwise it is certain that the booking will fail. For the second method, the user should be restricted to avoid already occupied seats.

14. the requested seat refers to only the ones currently requested by the current user and not to the requested by all the other users in that moment?

Yes, of course. Probably you do not even know which other users have requested seats.

15. Previously it has been said "The seat visualization page must also display the number of occupied seats, available seats, and the total number of seats." But now there's also the "requested" field, should we always show this field or only when we first interact with a seat?

It is ok to always show the number of requested seats. This will be clarified in the revised version of the text. Thanks.

16. so we don't have to show the status of the seats to the user in "real time", that is, if the seats are booked or requested by another user we don't have to show this change to users who are still booking but only if they reopen the booking page?

No need to show in real time the bookings from other users (we also did not see how to do this during the course...)

17. I am not sure: If i click on the submit button after i have selected my seats with the red/green/yellow matrix then appears a pop-up that asks me to confirm my choice?

Yes. It can be a pop-up, another page, or just a new button, this is up to you to decide.

18. This means that we have to save in the database the requested seats with the status "requested" and if after logout and a new login the seats have been occupied by another user the confirmation and transformation into occupied won't be possible or only that if during the user session another user will confirm and occupy some of that seats the confirmation won't be possible? Summarizing my question is: we have to store in the database also the requested seats or only the occupied once? The user should be able to see the seats that he has selected previously in the previous login(the requested but not occupied ones) or only the seats that he has confirmed and occupied?

No need to store the requested seat information. Only final information (occupied seats) has to be persistent when users log out and log in.

19. this case can happen only when the seats are occupied by user2 before the first user selects them because when he opened the reservation page those were listed as available(green), as the server wasn't already updated with user2 reservation?

In practice yes.

20. I do not understand which is the sense of this part. If I do not permit a user to press on an occupied seats, is not enough?

It is not enough. Please review the course contents.

21. Should the user be allowed to attempt reserving seats that are already visibly occupied (indicated by the red color), only to then receive a 5-second error message? Or should we prevent the user from even trying to reserve these already occupied seats, with the 5-second error mechanism being reserved for cases where seats become occupied simultaneously by others, which the user cannot see in real-time?

The second option. It makes no sense to allow reserving occupied seats.

22. if a user has already done a reservation, could it add new seats by reserving other seats or if a reservation exists, when selecting the concert, the system could say sth like 'error: a reservation has already been done by the authenticated user for this concert'

The latter (which is also easier)

23. where should i see the reservation for a logged user? in the initial page with the concerts that i can choose? or in a page apart?

Up to you. Of course, after user authentication.

24. If the user deletes the reservation, can they reserve seats for the same concert again?

Yes. No constraints about it.

25. Is it still an acceptable solution to allow an user to make more reservations to the same concert?

It is not what is required. However, it is something in addition, not required, so it simply will not be counted (if it is working),

26. the delete button must be always visible or can i use a structure so that the user can delete the reservation only if he clicks on the reservation details?

Both are ok

27. The second server should have another db to retrieve informations about the user's reservation or it just receives the reservations data from the client and return the discount computed?

It receives the reservation data and returns the computed discount. No db.

28. so we don't have to store it somewhere in the db, right?

Exactly, no need to store it in the db

29. so the discount should be saved in the reservation details in order for the user to see it everytime he wants or it can be only shown to the user one time and after it disappears forever and it will be impossible to retrieve?

As you like

30. Is it acceptable that when the user makes a reservation, the discount is displayed only for 5 seconds or, in any case, for a limited time?

No. This is not to be confused with what happens in the 2-dimensional visualization.

31. So if a user makes a booking, they will see the discount. If they return to the main menu and then reselect the concert, they will see the discount again (which might change due to the token), but even when returning to the page, they will still see it.

Yes, that is fine. Or showing it only after successful reservation is fine as well

32. is it possible to implement sql views? (This feature of sql was not used during the lectures, then I ask)

If they work with SQLite 3 and you are able to explain why you need them, yes

33. since there are six concerts but only 3 theatres it is right to make the assumption that the theaters have more than one concerts, obviously all different from each other?

Of course

34. there are 6 users in total?

- 4 user with two reservation for each user, and the reservations must be for two different concerts; and 2 users out of the 4 users must be loyal.

The reservation can be same for each user (like user1: concert1(2 seats), concert2(2 seats) and user2: concert1(2 seats), concert2(2 seats)) or the must be different between all of the users?

-2 users without any reservation and one of the 2 users must be loyal.

Yes