We'll cover the following Components of a meeting scheduler User Interval Meeting room Meeting

Calendar

Notification

Association

Composition

Aggregation

Design pattern

meeting scheduler.

The class definition is shown below:

approach.

the invite.

**Interval** 

meeting.

**Meeting room** 

desired number of people.

time, and the meeting room.

The class diagram of the Meeting class is provided below:

- id : int

- participants : User {list} - participantsCount : int - interval : Interval - room : MeetingRoom - subject : string

meeting.

Meeting

meeting.

Calendar

The class definition is provided below:

Meeting scheduler

**Notification** 

meeting scheduler.

**Association** 

and Meeting classes.

Composition

**Aggregation** 

Interval

**Notification** 

notificationId : int - content : string

- organizer : User - calendar : Calendar rooms : MeetingRoom {list}

creationDate : date/time

+ sendInvite(user) : void

+ cancelNotification(user) : void

generates

scheduleMeeting(user {list}, interval) : bool

Al-powered trainer

Powered by AI

← Back

Use Case Diagram for the Meeting Schedu...

+ cancelMeeting(user {list}, interval) : bool

MeetingScheduler

User

Al-powered trainer

Meeting scheduler

Relationship between the classes

Class diagram of the meeting scheduler

In this lesson, we'll identify and design the classes, abstract classes, and interfaces

As mentioned earlier, we'll design the meeting scheduler using a bottom-up

as their name, email, and can also accept or reject an invitation to a meeting.

name : string - email: string

Components of a meeting scheduler

based on the requirements that we have previously gathered from the interviewer in a

The User class is responsible for representing the personal information of a user such

User

+ respondInvitation(invite) : void

The class diagram of the User class

∵∴ R5: Meeting scheduler

**R5:** Users will receive an invite regardless of whether or not they are available at

the interval. Users can respond to the invitation by either accepting or rejecting

The Interval class contains the start time and end time of a meeting. The visual

Interval

startTime : date/time endTime: date/time

The class diagram of the Interval class

்டு R3: Meeting scheduler

R3: If not reserved already, each meeting room should have the ability to be

booked, along with setting an interval, a start time, and an end time for the

The MeetingRoom class contains the details of any particular room, such as its

of intervals to keep track of when the room is booked for a meeting.

- id : int capacity : int

The class diagram of the MeetingRoom class is provided below:

capacity and a status, to identify whether it is currently available. It also contains a list

**Meeting Room** 

- bookedIntervals : Interval {list}

The class diagram of the MeetingRoom class

∵்റ்∙ R2 and R3: Meeting scheduler

R2: Each meeting room should have a specific capacity to accommodate the

**R3:** If not reserved already, each meeting room should have the ability to be

booked, along with setting an interval, a start time, and an end time for the

The Meeting class displays the meeting details, such as the participants, the meeting

Meeting

+ addParticipants(participants {list}) : void

The class diagram of the Meeting class

்டு R3: Meeting scheduler

R3: If not reserved already, each meeting room should have the ability to be

booked, along with setting an interval, a start time, and an end time for the

The Calendar class keeps track of all the meetings that are scheduled or being held.

Calendar

meetings : Meeting {list}

The class diagram of the Calendar class

்டு R6: Meeting scheduler

**R6:** Each user should have access to a calendar that can be used to track the date

The MeetingScheduler class contains an organizer that is responsible for scheduling

MeetingScheduler

+ checkRoomsAvailability(numberOfPersons, interval) : MeetingRoom

Class diagram of the MeetingScheduler class

∵ R6: Meeting scheduler

**R6:** Each user should have access to a calendar that can be used to track the date

The Notification class will send a notification for an invitation to a user regarding

any new meeting. It will also send a cancelation notification to a user as well, in case

**Notification** 

The class diagram of the Notification class

R4: Meeting scheduler

Now, we'll discuss the relationships between the classes we have defined above in the

The User class has a one-way association with the Notification class and a

• The MeetingScheduler class has a one-way association with the Notification

The association relationship between classes

Calendar

The composition relationship between classes

The MeetingScheduler class is aggregated from the User and MeetingRoom class.

The aggregation relationship between classes

User

+ respondInvitation(invite) : void

creates

MeetingScheduler

User

Meeting

Meeting

User

Meeting

+ addParticipants(participants {list}) : void

Interval

startTime : date/time endTime : date/time

20 Prompts Remaining

Complete

Next  $\rightarrow$ 

Sequence Diagram for the Meeting Sched...

participants : User {list}

- room : MeetingRoom

- interval : Interval

- subject : string

MeetingRoom

Calendar

meetings : Meeting {list}

- capacity : int

- id : int

notificationId : int content : string

- creationDate : date/time

+ sendInvite(user) : void + cancelNotification(user) : void

**R4:** A notification should be sent to all the people invited to the meeting.

Relationship between the classes

The class diagram has the following association relationships:

**Notification** 

MeetingScheduler

The class diagram has the following composition relationships:

The Calendar class is composed of the Meeting class.

The class diagram has the following aggregation relationships:

MeetingRoom

Class diagram of the meeting scheduler

email : string

Here's the complete class diagram for the meeting scheduler:

sent to

MeetingScheduler

+ checkRoomsAvailability(numberOfPersons, interval) : MeetingRoom

created and that this instance has a global point of access.

The MeetingRoom class is aggregated from the Interval class.

The MeetingScheduler class is composed of the Calendar class.

two-way association with the Meeting class.

a meeting using the calendar. It can also cancel a meeting, check the availability of

The visual representation of the MeetingScheduler class is provided below:

+ scheduleMeeting(user {list}, interval) : bool + cancelMeeting(user {list}, interval) : bool

+ releaseRoom(room, interval) : bool

and time, as well as to schedule or cancel meetings.

any meeting gets canceled or is postponed.

The UML representation of the class is shown below:

+ bookRoom(room, numberOfPersons, interval) : bool

and time, as well as to schedule or cancel meetings.

meeting rooms, and book or release a meeting room.

- organizer : User - calendar : Calendar - rooms : Room {list}

isAvailable: bool

representation of the Interval class is as follows:

Learn to create a class diagram for a meeting scheduler using the bottom-up approach.

Class Diagram for the Meeting Scheduler

- bookedIntervals : Interval {list} + bookRoom(room, numberOfPersons, interval) : bool - isAvailable : bool + releaseRoom(room, interval) : bool The class diagram of the meeting scheduler Design pattern In the meeting scheduler design, the entire system revolves around the scheduler which is responsible for scheduling meetings. To create a robust design, it is not possible that there can be more than one instance for the scheduler. Therefore, we use the Singleton design pattern to ensure that only one instance of the scheduler is

At this stage, everything should be clear. If you encounter any confusion or ambiguity,

feel free to utilize the interactive AI-enabled widget below to seek clarification. This

tool is designed to assist you in strengthening your understanding of the concepts.

We have completed the class diagram of the meeting scheduler according to the requirements. Now let's design the sequence diagram of the meeting scheduler in the next lesson.

**Prompt Al Widget**