Event Calendar

Capabilities

- Onboard Users(include working hours of the day)
- Users can be grouped into a team. A user can be part of only one team.
- Create an event with participants, start time and end time for a given day(assume start and end time are of same day)
- Participants can be users or teams or both.
- User can participate in only 1 event at any given point in time
- When creating an event, user has an option to specify the number of required representatives(lets call it **N**) from teams(assume that the number of required representatives are same for all teams in an event)
- When a team is added, block the event against the **N** available members of the team. If the **N** users are not available for any team in the event, the event can't be created (Eg. If **N** is 2, then block the event against 2 available users of each team involved in the meeting).

Requirements

1. Create User
2. Create Team of users
3. Create Event for users/teams
4. Get events between a time range for a user. Include events for the given user which were blocked as part of the team being a participant.
5. Suggest available slots for given list of users/teams for a given day

Bonus

1. Update event time - if it is not possible return the status accordingly
2. Support recurring event(time, start_date, end_date, frequency)

Expectations

1. Make sure that you have working and demonstrable code for all the above requirements.
2. Feature requirements should be strictly followed. Work on the expected output first and then add good-to-have features of your own.
3. Use of proper abstraction, separation of concerns is required.

4. Code should easily accommodate new requirements with minimal changes.
5. Proper exception handling is required.
6. Code should be modular, readable and unit-testable.

Important Notes

- Do not use any database store, use in-memory data structure.
- Do not create any UI for the application.
- Do not build a Command line interface, Executing test cases or simple Main function should be sufficient
- **Do not make any assumption, please ask it out.**

Sample Test Cases

1. Onboard 5 Users
    a. Create user A with working hours 10am to 7pm
    b. Create user B with working hours 9:30am to 5:30pm
    c. Create user C with working hours 11:30am to 6:30pm
    d. Create user D with working hours 10am to 6pm
    e. Create user E with working hours 11am to 7:30pm
    f. Create user F with working hours 11am to 6:30pm
2. Create Teams
    a. Create team T1 with users C & E
    b. Create team T2 with users B, D & F
3. Create events
    a. Create event Event1 with user A & team T1 for next day 2PM to 3PM, with 2 representations
    b. Create event Event2 with user C for next day 2PM to 3PM //should fail since C is already part of Event1
    c. Create event Event3 with Teams T1, T2 for current day 3PM to 4PM, with 2 representations
    d. Create event Event4 with user A & team T2 for current day 3PM to 4PM, with 1 representation
    e. Create event Event5 with users A and F for current day 10AM to 11AM //should fail since it's outside F's working hours
4. Get Events for a time range
    a. Get events for user A from current day 10AM to next day 5PM
       Output: Event1, Event4
    b. Get events for user B from current day 10AM to next day 5PM
       Output: Event3 or Event4 depending on which members of T2 were chosen for these events
5. Get available time slots
    a. Get available time slots for user A and team T1 with 1 representation for current day
       Output: 10AM to 3PM and 4PM to 7PM (Reasoning: C's working hours end

at 6:30, but since E will be available to represent team T1 until 7:30PM.
Taking overlap of A and E as 7PM)