Problem #1 => Calendar App

About Events

- An event would typically consist of {start, end, location, Owner, user-list, title}.
- Events can either be like meetings (with a dedicated location and appropriate guest-list) or as well be like holidays, birthdays, reminders etc.
- An event once created, can be either accepted or rejected by the constituent users - if neither it should be in neutral state.

Implement any 2 API's:

1. User API to create, update, or cancel their events. Create/Update operation should be successful only in case of no conflicts of resources.
2. Given a user, provide an API to fetch list of event/meetings for the given date range.
3. Given a Meeting ID, provide an API to give details of invitees with their responses i.e., ACCEPT, DECLINE etc.
4. API to find available rooms in specific building at particular time-slot

Expectations

1. Code quality should be production ready for merge and deployment.
2. Guidelines have the highest weightage than finishing more api's.
3. Code should be demo able. Create the sample data yourself in a file, test case or main driver program itself (no external data store). Don't spend time parsing the inputs.
4. Code should be readable, modular (no monoliths), testable, extensible with proper naming conventions.
5. Code should handle edge cases properly and fail gracefully.

Guidelines:

- o Define a detailed object model for entities required by system
- o Make proper use of Inheritance, Abstraction, interfaces, exception handling
- o Have proper commenting in code and should follow best coding practises
- o Use design patterns like Builder, Factory, Visitor etc wherever applicable
- o Justify his/her coding choices i.e. why did he/she choose to define a separate function for a feature or why did he/she not define constructor for initialising class
- o Define Enums, Singleton classes where applicable
- o Separation of concern is addressed
- o Implement unit test cases for key sections of his/her code
- o Use Java 8/7 features like functional interfaces, Auto Closable resources etc.