Backend exercise

This is a small project draft, consisting in the development of a **Web API** for **appointments scheduling**.

Candidates are kindly asked to read the requirements and implement them, following the guidelines about technologies and tools.

The exercise will be evaluated not only for its soundness and completeness, but also for the quality and efficiency of the design and the design patterns used.

In case you find some specs or implementation details unclear, please proceed as you think it is more appropriate, and be ready to explain the reasons for your choices.

REQUIREMENTS

- The appointments scheduling system can handle a number of appointments.
- There are two types of appointments: meeting and reminder.
    - Meetings can be recurrent (with a weekly schedule) or one-off, and have a list of attendees.
    - Reminders are always one-off and don't have attendees.
- The Web API should provide at least the following functionalities:
    - Creation of an appointment (either recurrent/one-off meeting or reminder)
    - Update the date of an existing appointment
    - Update the list of attendees for a meeting
    - Retrieval of all the appointments on a given day
    - Retrieval of all the appointments for a specific attendee on a given day

GUIDELINES

- You can choose the **language** you prefer, but an OOP language would be preferable (C#, Java…). You may also use a specific **framework** if needed (.NET Core, Spring Boot...).
- For the **database**, you can just use a simple file (json, xml, csv…) or any database management system with a free license (SQLite…).
- You should implement at least a couple of **unit test** cases using the testing framework you prefer.
- You should implement at least one **integration test** using the framework you prefer.

EXPECTED DELIVERABLES

- A compressed archive containing the source code, or a link to a repository.
- A minimal documentation of the project in the form you prefer.
- Some instructions on how to build and execute the project (including examples of how to call the API directly, and data population scripts if relevant).