DVR Scheduler

A client is asking you to build software for their new DVR device.

Input to the system will be:

- Date
- Timespan
- Channel

At minimum, your system needs to provide an interface for the DVR system that allows users to enter schedules, and also allows the recording equipment to query what channel it should be recording at a certain time.

Notes

For this system, account for any implementation details that you feel are missing in whatever way you think makes the most sense. Realize that details have been left out intentionally so that you can demonstrate your ability to make reasonable decisions about what you implement (and how) in the time you have available to you, and for you to explain and justify those decisions to us.

Example

For example, if the user has entered the following schedule:

Date	**Time**	**Channel**
02/11/2014	5:00pm-8:00pm	314
02/11/2014	7:30pm-10:00pm	215
02/11/2014	9:00pm-11:00pm	182

Your program would respond with output somewhat equivalent to this:

Input	**Output**
2/11/2014 6:10pm	Record 314
2/11/2014 8:30pm	Record 215
2/11/2014 10:30pm	Record 182
2/11/2014 3:00pm	Not recording
2/11/2014 7:45pm	?

Note that the behavior is undefined at 7:45, since the DVR has been instructed to record on channel 215 and 314 at that time. Your job is to come up with a reasonable conflict resolution system.

Tips

- This problem is bit more complicated than meets the eye. You don't have to solve all the problems, but being aware of the concerns should guide your implementation without making it overly complicated.
- Consider the cases where a user schedules more than two programs to record at the same time.
- The recording software is able to query by time for what channel to record on, but how does it know where to save the files?
- How does the DVR know which recordings go with which instructions from the user when the user goes to play the recordings?

Problem Extensions

Consider these in the context of the expectations below.

Only being able to record a single episode is a bit archaic. What happens if you have a second tuner?

- How does this change the method of conflict resolution?
- Assuming it's unacceptable to interrupt a recording to switch tuners, is it important for the scheduler to consider which tuner to start recording a program on?

In the described implementation users only enter dates and times.

- What problems arise from this simplification?
- How might you extend the available input methods?
- For example, would you support recurring shows?

Optionally implement a user interface for the DVR

- What features are important for the user to be able to see to make changes?
- Once a user schedules recordings, can they view them? How? In a list, a calendar, something else?

Provide a well-defined external API for both the User actions and Recording software.

- What features/endpoints would you need to support?
- Do you use this "external" API internally for your own apps as well? Why/why not?
-

Expectations

Junior Engineer

Read and understand the problem extensions.
Reasonable and complete test cases (edge cases are critical for this problem)
Good solution for conflict resolution, points for creativity and pragmatism.
Understands that the instructions are a minimum requirement, and that creativity and out-of-the-box thinking is required for an acceptable solution.

Intermediate Engineer

Choose at least one of the following problem extensions to implement, or invent your own. Be prepared to defend your choice of feature set over another, as if to a project manager.
Should show significantly more creative thinking than a Junior engineer.
Will be able to discuss implications of other features, even if not implemented.
Notices gaps in the described system not described. Is able to discuss solutions to those problems even if not implemented.
All of the expectations of a junior engineer.

Senior Engineer

Implement one or two of the problem extensions.
Be able to discuss the project at the architecture level.
Be able to discuss a high level plan to take the project to completion.
Does a good job triaging features, and is able to talk about why the features they chose to implement were more important than others.
All the expectations of an intermediate engineer and a junior engineer.