Elevator System

Vague, high-level problem statement, as delivered in an interview: Design an elevator system for a building with multiple elevators.

What that typically means: Design an API to operate an elevator. Your API will be used by the “UI” that could be either the elevator itself, or could be an app.

1. Need to queue elevator call requests

2. Need to queue floor requests within the elevator

2.1 Elevator needs to keep the floor requests in ascending order if going UP

2.2 Elevator needs to keep the floor requests in descending order if going DOWN

3. Strategy for assigning an elevator for a given call request

4. Elevators can be in maintenance or out-of-order states

5. Elevator doors can ONLY be open if the elevator is standing and it is on a particular floor

6. Elevator must be able to take call requests while moving and must be able to stop at the floors even if the requests came late but elevator didn't pass the floor yet

Deliverables:

1. A class diagram, showing relationships with each other where appropriate. Classes should show state and methods. Use any convenient notation. UML is more widely known.

2. Main() method, showing how you'll initialize your system and start using it.

3. Identify the design pattern.

4. A REST API