Expectation

This exercise will be to write a solution to the below requirements while pairing with one of our engineers. This will last 1 hour.

● We expect candidates to drive their implementation from the tests (Test Driven Development)   
● We are looking for an understanding of how familiar you are with using the Red Green Refactor cycle [1] to work through the requirements and allow the solution to emerge   
● We are looking for evidence that you can design code to represent real world scenarios  
   
● We use this exercise to observe how you Communicate and collaborate in real time  
   
● We use this exercise to verify that you can use the language and tools you have chosen comfortably, as you demonstrate how you navigate around the implementation of the solution.  
   
We recommend not attempting to complete the exercise in advance, since we’re very interested in observing the design process you follow in real time. Although it is recommended to read the requirements and complete the preparation.

We have found that some candidates who have prepared an implementation in advance, find it much harder to collaborate with different ideas during the pairing exercise.  
   
Preparation  
Please follow these steps ahead of the interview  
   
Please set up a skeleton class library or module along with a testing framework, in your language and developer environment of choice, so you are ready to write code and run tests in the interview.  
   
Please let us know which language, testing library and IDE you plan to use ahead of the interview.  
   
Requirements

Our little theatre has 15 seats, arranged in 3 rows of 5

Rows are assigned a letter from A to C  
Seats are assigned a number from 1 to 5

**As a theatre box office manager**

**So that we can control reserved seating**

**I want to allocate seats to customers**

**When** a customer requests a number of seats for a performance

**And** there are not enough seats available to meet the request

**Then** the system should return a status indicating that there are not enough seats available

**When** a customer requests a number of seats for a performance

**And** there are some seats available, but less that the number requested

**Then** the system should return a status indicating that there are not enough seats available

**When** a customer requests a number of seats for a performance

**And** no seats are allocated

**And** there are enough seats to meet the request

**Then** the system should allocate the required number of seats starting from A1 and moving from left to right, front to back

**And** the system should return a value indicating which seats have been allocated

**When** a customer requests a number of seats for a performance

**And** some seats have been allocated

**And** there are other seats available

**Then** the system should allocated the required number of seats starting from the first seat available and moving from left to right, front to back

**And** the system should return a value indicating which seats have been allocated