Thank you for applying to be a part of Rubicon Water!

We have reviewed your CV and would like to proceed your application to the next phase. Please find below the instructions for our coding challenge.

Rubicon Coding Challenge

The aim of this exercise is to implement a water ordering API so farmers can request water to irrigate their farms. Farmers can use this API to place water orders, view existing water orders and cancel water orders before they are delivered.

A basic water order has the following attributes:

- farmId – A unique ID for identifying a farm.
- Start date time – The date and time when water should be delivered.
- Duration – The duration of the order (e.g. Duration of 3 hours means water will flow into the farm for 3 hours from the start date time).

As part of this exercise, you will need to:

1. Design and implement an API to accept new orders from a farmer.
    - A single API request may contain multiple water orders from the same farmer.
2. Design and implement an API for cancelling an existing order if it hasn't been delivered.
3. Design and implement an API so farmers can query existing orders. When querying orders, the farmer should be able to see the status of each order. Possible status of a water order:
    - Requested – *"Order has been placed but not yet delivered."*
    - InProgress – *"Order is being delivered right now."*
    - Delivered – *"Order has been delivered."*
    - Cancelled – *"Order was cancelled before delivery."*
4. The API must ensure the water orders for a farm do not overlap.
    - For example, if Farm X already has an order for 30 Jan 2019 starting at 6am with a 3 hours duration, it should not allow Farm X to place an order starting at 8am on the same day.
5. To simulate water delivery, your application should output a line each time the status of a water order changes. This include –
    - When a new water order is placed;
    - When a water order starts (Start date time of the order);
    - When a water order is delivered (i.e. start date time + duration);
    - When a water order is cancelled;
6. For the purpose of this exercise, you can assume orders are stored in memory. Orders do not need to persist between application restarts.

Submission Instructions

The solution should be implemented in Java.

You are welcome to make use of open source libraries in your solution.

Please send your completed solution to kwong.lai@rubiconwater.com.

Your solution will be evaluated based on the following criteria:

- Functional correctness
- API Design
- Readability
- Maintainability
- Extensibility
- Supportability