Coding exercise

We've reviewed your CV, gotten to know a part of you during our first meeting and talked to you about Auctionata, our Technology organization and the way we think about product development. Now we are very much interested in learning more about how you tackle and approach solving a real world problem.

Before we present the actual problem here are some notes up front:

- *Effort required*: Please do not spend more than 8h working on this. Rather adjust your scope to match this time frame. We have some suggestions for cutting scope later on in the exercise. But feel free to dice this up in any way you want.
- *Deliverables*: We want to be able to view your working solution and review your source code. How you allow us to do that is entirely up to you.
- *Technology*: Please feel free to pick whatever technologies you think solve the problem best. But please keep in mind that this story is part of our business critical application.
- *Solution and approach*: There are no right or wrong solutions here. Obviously different implementations will have different pros and cons. The goal of the exercise is to get in idea of how you approach problems and on which areas you like to focus.
- *Next steps*: If we like your solution and approach we'd love to have a brief pair programming session with you to implement a little piece of functionality together. Please keep this in mind when deciding how you make your deliverables available to us.
- *Think aloud*: It's really helpful for us if you could provide some context around the major decisions you made. Where and how you do this is again up to you. If you decide to add this information in the source code please mark it up somehow so that we can distinguish between regular comments in your code, i.e. those that you would normally add to your code, and those that you add just because of our request for you to think aloud.

Domain Background

If you are unsure about how auctions work please have a look at the following video which shows a summary of one of our auctions: http://auctionata.com/assets/videos/Lot_8197-1_NR7_LOT_33_en.webm (http://auctionata.com/assets/videos/Lot_8197-1_NR7_LOT_33_en.webm).

Requirements

Narrative

```
As a potential buyer in an online auction
I want to be able to bid on an item
So that I can participate in the auction
```

Scenario 1: displaying information about the current item

```
Given I am in the auction room
Then I see the current item picture, description and name
And I see the current highest bid with a button to place a new bid
```

Scenario 2: single user bidding on an item

```
Given I am in the auction room
When I place a bid on an item
And I am the only bidder
Then I am the highest bidder
```

Scenario 3: multiple users bidding on an item - you are first

```
Given I am in the auction room
When I place a bid on an item
And I am not the only bidder
And my bid was placed first
Then I am the highest bidder
```

Scenario 4: multiple users bidding on an item - you are not first

```
Given I am in the auction room
When I place a bid on an item
And I am not the only bidder
And my bid was not placed first
Then I am not the highest bidder
```

Scoping Options

We know that might look like too much work for an interview process. Please remember that we are only asking you to spend 8h on this exercise. Please consider the following options to cut scope:

Option A - Backend Implementation

- Focus heavily on backend implementation,

- Have a simple UI just to show what is happening behind the scenes
- Bonus points: Provide a nice API that could easily be consumed by a single page application or any other client.

Option B - Frontend Implementation

- Make it a single page application that uses API's for most of the heavy business logic
- You can mock the API to represent the result of you working together with another team to define the interfaces
- You could make the mocks convenient for you, but not too convenient, they should be constrained by realistic assumptions.

Option C - Your own scope

- Choose your own slice
- We want to see what you care about the most
- You could
    - spend copious amounts of time making the CSS look perfect everywhere,
    - focus on dealing with concurrency and build a prototype which is all about speed and scalability,
    - go the classic route and do a server side MVC implementation,
    - have a super slick build pipeline before anything else.
- All that that is fine! Go for it, just let us know why that was your choice.