Banker Display System

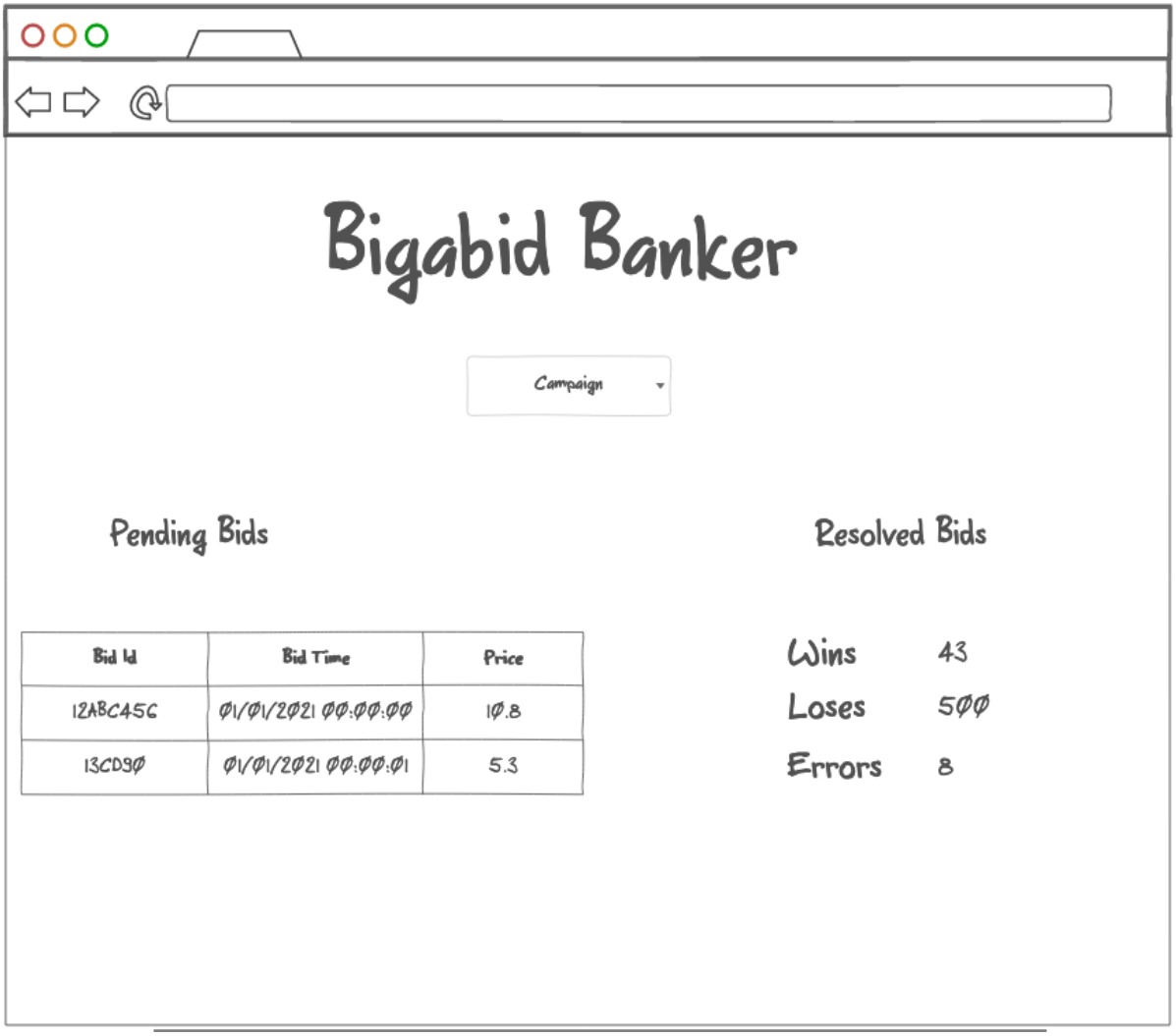
During the interview we spoke about the banker, a distributed system working with Redis as a caching engine, where the banker inserts new bids and eventually resolves them.

The goal of this exercise is to create a web interface to display a “real time” representation of the banker’s actions, basically presenting the data we have in Redis.

The web interface will include the following components:

- A dropdown showing the available campaigns
- A table showing the bid ids, their price, and the time of the bid
- A section showing the sum of bids resolved grouped by their status

Example design:



The general flow will be as such:

After the user selects a campaign, it will start fetching bids starting from that exact moment (by time).
All of the unresolved bids should be displayed in the table.
Once a bid is resolved with a status, add it to the sum of resolved bids of that status.

Technical Details:

- A list of campaigns will be in a set under the key “LIST_OF_CAMPAIGNS”
- The bid ids will be in a sorted set named “LIST_OF_BIDS” with their score being the unix timestamp of the time of the bid
- The data of the bid will be under a key which is the bid id itself, the value will be a string of JSON with the following format - { price: number, campaign: string, status: number }
- Status value is an enum
    - 0 - Pending
    - 1 - Error
    - 2 - Lose
    - 3 - Win
- **Every** bid must be resolved after no more than 60 seconds

Attached with this doc is a link to a project named Seeder which will seed data for you so all you need to do is just build the website. All of the details on how to run it is under the README file. Remember, your responsibility is to only deal with the website, treat the seeder as a third party you’re working with.

The example design attached is just an example, the way it looks doesn’t matter at all.
Make sure it runs smoothly with no issues, try and see what is the maximum amount of bids you’re able to deal with per second, performance is the key here.

Post the result on your personal Github \ Bitbucket \ any other platform, try and split your commits into rational portions, and send it when you’re done.

This exercise shouldn’t take you more than 3 hours, you shouldn’t continue if it took longer - manage your time accordingly, and prioritize - if you need to give up on certain features, do so.

If you have any questions feel free to e-mail me:
amirs@bigabid.com

Good luck!