Truecaller “Java/Object Oriented Principles” Assignment (v2)

This is an assignment designed specifically to test how much a candidate knows about code quality, object oriented principles, coding best practices, design patterns and written communication skills. The output of this assignment would be the base for evaluation and will be followed by a technical interview if qualified.

Project Specification:

1) Each API call in the project should have a user as parameter (Assume numeric auto incrementing ids)
2) Design a REST API endpoint to store the views of user profiles
    a. UserX viewing the profile of UserY should be recorded
    b. Date and time of the view should be recorded
3) Design a REST API endpoint to view the users who viewed this user’s profile in the past.
    a. The list should include the viewer’s user id and also the date/time of the view
    b. The list should NOT include more than 10 items
    c. The list should NOT include views older than 10 days
4) Design any relevant support code to ensure that the requirements in the third specification are met at all times.
5) Assuming you have millions of views every hour, try to come up with the most efficient database schema in terms of storage space, latency and throughput. Write inline comments to justify the design decisions taken, so that the next developer after you will understand the design and can easily maintain it.

Instructions:

1) Grab Dropwizard and create a maven project with it: http://dropwizard.codahale.com/
2) Setup your project so that it can read from and write to a local file based database. (This is essential for us to test your solution locally without creating databases and schemas). Use any Sql compliant library here; H2, Derby, HsqlDb, Sqlite etc.
3) Create the source code to meet the following requirements
4) Create a design document that briefly discusses the chosen database schema and the reasoning behind it. Please make sure these questions are answered and discussed:
    a. Do you delete any data from the database?
        i. If yes, When? Why?
        ii. If no, Why?
    b. Do you have any periodic task type of batch jobs to maintain data?
        i. If yes, Why needed?
        ii. If no, Why not needed?
    c. What type of compromise (for example; tradeoff between storage on disk vs latency) did you see? What was your decision? Why?
5) Also include the instructions to run the project and test if it works as expected

Please focus on clean/maintainable code, best possible data structures and corresponding database schemas. Use your good judgment to include the right amount of design patterns in your code. It should be structured but also not over engineered.
Please run clean goal to avoid big file attachments. This is very important since some of our communications are done over emails, different size limits on different providers might cause an inconvenience otherwise.
Please be pragmatic and thank you in advance for your interest in Truecaller and for the time you have spent for us.

We sincerely hope to see you as part of our team.

Umut Alp, CTO

Rosario Leo, Director of engineering, Platform, rosario.leo@truecaller.com