Technical Interview Exercise

In an effort to round out our interview process, the engineering team would like you to complete the following tech task. This is to ensure that candidates are able to not only complete the task at hand but package and ship production-ready code (with tests, deployment mechanisms, etc.).

Create a clock application (written in Go) that will print the following values at the following intervals to stdout:

- "tick" every second
- "tock" every minute
- "bong" every hour

Only one value should be printed in a given second, i.e. when printing "bong" on the hour, the "tick" and "tock" values should not be printed.

It should run for three hours and then exit.

A mechanism should exist for the user to alter any of the printed values while the program is running, e.g. after the clock has run for 10 minutes I should, without stopping the program, be able to change it so that it stops printing "tick" every second and starts printing "quack" instead. Any of the clock output values should be able to be changed to any value at any time.

Include a Dockerfile for containerized deployment and store in a DB each of the signals that clock triggers.

We would also like you to use the standard Go libraries as much as possible to keep the solutions standardized across candidates to make them easier to review. Please try to make this code as production-ready as possible.

The solution should be placed on your github and made private and password protected. You will be requested to share the github link and the password so we can access it.

As a bonus (not required): Include a mechanism to set the initial interval values for the clock messages. e.g: tick-> every 5 seconds, tock -> every 3 minutes, bong-> every 2 hours

Presentation and Code Review

Once you have completed the exercise, you will be required to present your project to the technical interview panel. During the presentation, you should explain your user story, design choices, the technical architecture, and as an optional demonstrate the functionality of the deployment. This will be done over Zoom and you will screen share either your private GitHub repository or IDE.

After the presentation, the interview panel will conduct a code review of your project. You will be asked to explain your coding decisions and answer any questions related to the code. The interview panel will evaluate your project based on the following criteria:

- Go skills: Ability to develop with the Go language and its standard libraries. Code quality and
- extensibility. Logical reasoning skills.
- Clean Architecture: Your architecture should adhere to Clean Architecture principles.
- Code Testing: Your project should have sufficient testing coverage.
- Code quality: Your code should be well-organized, readable, and adhere to best practices.
- Functionality: Your application should perform the required operations.
- User Story: Your user story should drive the development of the application and be included in
- your presentation.
- Presentation: Your presentation should be clear, concise, and demonstrate a good understanding of the project

Scoring

You will be scored based on your adherence to Clean Architecture principles, code testing (unit testing), code quality, user story, and code review remarks. Each criterion will be weighted equally.

We hope this exercise challenges your technical abilities and helps us assess your suitability for the position. Good luck!