HA Full Stack Developer Assessment

December 2018

2218. Space exploration is underway and mostly done by private companies. You joined Engineering Department of one of the private government contractors, Atlas Corporation.

*In that year and age everything is automated, so survey and data gathering are done by drones (model: Drone gen. G). Due to solar radiation the drones navigation system sometimes fails. We need a way to keep an eye on them. You, as the most promising recruit of Atlas Corp, were tasked to develop a g-drone position system (**GPS)**. The project has the following requirements:*

- *API first (it can have multiple different clients)*
- *Functional UI (must work, doesn't have to look pretty)*

Please, document all assumptions made, all design choices taken and questions if you'll have them.

- **API**
    - List all drones with **ID**, **quadrant** and **current position** (x, y float). Position is relative to quadrant. **Current position** should change over time. It's up to you to use a random values or a *formula* that makes movement over time look realistic.
    - Add new drone with **initial position** (x, y float) and **quadrant**
    - Remove drone by **ID**
- **UI**
    - Display all the drones in the selected quadrant. Position should be updated live (i.e. every X seconds via WebSocket or polling).
    - Display a message if a drone goes out bound of the selected quadrant (you are expected to only work with one quadrant for this first version).
    - Add new drones

Example **API** response

```
GET /drones
[
  {
    // x, y are coords
    // (values sent as strings, but must be treated as floating point number)
    "x": "123.12",
    "y": "456.56",
    "id": "45745c60-7b1a-11e8-9c9c-2d42b21b1a3e",
    "quadrant": 10
  }
]
```

Implementation requirements:

- **Production-quality code.** Business logic here is kept to minimal complexity to allow you to focus on quality and style.
- **Testing is required.** Implementations without any test coverage will be rejected.
- **DevOps-friendly.** Provide a way to package your application for deployment via Dockerfile.
- **Team-friendly**. Add documentation and short README.

Additional questions (please provide written answers to these):

- What instrumentation this service would need to ensure its observability and operational transparency?
- In general, how would you separate **technical decision** to **deploy** something from **business decision** to **release** something?

Code must be submitted in a git repository (GitHub, Bitbucket, GitLab), please share a link to the repository you have used.