Blix Coding Test

At Blix, we build and deploy infrastructures that track foot traffic at clients' premises.

Let's consider the following scenario where we want to build a **reporting app** for our **foot traffic** data.

For this simple app, we have two data concepts, **location** and **visit**. Each location has a **name**. Each visit is assigned to a **location**, and has a **start time**, an **end time** and a **signal strength**. A signal strength is a negative integer which ranges from -1 to -100 (FYI a greater number means stronger signal in this case).

Now what we need is to build a **Ruby on Rails** app that does the following:

- Build a **database** that would allow us to store those two data types. We expect to see **validations** in Rails models according to the given constraints. SQLite is fine but you can also choose PostgreSQL or some other RDBMS.

- Build an **API** that outputs **JSON** encoded data for each of the following two use cases. Each API endpoint should allow filtering by a date range.
    - A **time series (daily) view** of visits, i.e. number of visits vs day-truncated start time
    - A **histogram view** of visits' signal strengths, i.e. number of visits vs signal strength

- Build a simple **UI** to consume and visualise the aforementioned data views provided by the API. E.g. you might choose to use a line chart for the time series and a bar chart for the histogram. Or use your out-of-the-box thinking! The use of some JavaScript framework such as Ember or Angular is desirable, but it is also okay to use plain Rails templates and JavaScript libraries.

You **don't** have to consider any authentication for this task.

We are looking for quality in *design*, *simplicity*, *testability*, *coding style*, *modularity* and *creativity*, among other things. That means you may want to use:

- Git/Mercurial or a similar version control system

- RSpec or similar test frameworks

- A proper README file to document setup procedures and any special design decisions