Backend Assignment

Assignment Context

Attached is a roster from the airline **DTR** which uses a popular roster system (CCNX). The Airline crew upload their raw roster data to our service (pdf, excel, txt, html or webcal files). The business logic which we maintain per airline roster system is capable of extracting and parsing the relevant data from the raw roster data into our normalised models.

The following duties can e.g. appear on a roster.

- Day Off - Not scheduled to work.
- Report Event - Start for a day of working. A day can have multiple flights. Sometimes also called Check-In.
- Flight Events - Flight from Departure Airport to Arrival Airport
- Debrief Events - End for a day of working. Sometimes also called Check-Out.
- Layover Events - When you sleep at an Arrival Airport and fly out later.
- Simulator / Training Events - Training Course
- Standby Events - On reserve duty. Can be called by the airline any time.

For this assignment, we want you to extract all activities from the given roster. Activities can be distinguished by their type, for this roster we'll work with DO (Day Off), SBY (Standby), FLT (Flight), CI (Check-in), CO (Check-out) and UNK (Unknown, which is anything else not mentioned here). In the bottom of the roster there is a small section that explains certain codes of this particular roster. You may consider all columns *after ACReg* as irrelevant data for the context of the assignment.

For flights, the rule is that an activity should have 2 characters, followed by an undefined amount of numbers. For flight events we'll need to know their flight-number. Example: **DX77**

Rosters are published in either Zulu time (UTC) or Local time and sometimes both. The current roster shows both times on the events, but we want to only parse the **Zulu** times as event start/end times.

For timings, STD means Scheduled Time Departure and STA means Scheduled Time Arrival. Start/end variables are discussed as "Departure" and "Arrival" in aviation context.

Check-In/Out events are connected to flights and happen on the location of the flight. So Check-In uses the start location of the next flight, Check-Out uses the end location of the previous flight.

Concrete Steps

Usage of external libraries is allowed. Mind Abstraction, as different airlines have different roster layouts, and all have to be parsed with a single parse endpoint.

Create a Laravel (**version 10.x** running **PHP 8.2**) application with the following requirements:

- Parse the events from the given duty roster using the context above.
- Store the events in a **SQLite** database. DataBase should be re-creatable by using php artisan migrate.
- Provide API endpoint(s) to request events for certain scenarios:
  - give all events between date x and y.
  - give all flights for the next week (current date can be set to 14 Jan 2022)
  - give all Standby events for the next week (current date can be set to 14 Jan 2022)
  - give all flights that start on the given location.
  - ability to upload the roster by giving a file as input.
- The code has to be (unit/integration) tested and please show how much code coverage you've achieved with these tests.

If you can deliver the application as a Docker container, so we can easily execute it, this would be a big plus (but is not required).

We don’t expect any visual front-end, we are mostly interested in API endpoints we can call from e.g. Postman.