Technical Exercise		
for an IoT Developer position @ Concept Reply		

INSTRUCTIONS

1. General note

The exercise is split into two sections: the first section is a coding task, where you should write some code to manipulate the provided data. The second section focuses on software design and architecture.

Please try to work on every question and to avoid getting hung up on questions you cannot answer well. Comments and additional explanations or drawings are a good way to show us your thought process.

2. Source code

Source code should be written for a professional audience. You can choose a language/framework (our preferences are: **JavaScript/Typescript, Java or Golang**).

3. Test duration

Please upload your result no more than *3 hours* after downloading the exercise. With each task, we provide you with a high-level indication regarding our estimated effort, which is not binding, but should give a reference as to our expected complexity of a task.

4. Submitting results

Please upload the result of your work as one zip file containing PDF documents, images, and your source code to the same portal from where you downloaded the test.

Please don't put your solution on a public code repository (e.g., GitHub).

SECTION 1: CODING (75 MIN)

In the following JSON snippet, you will find sample data describing electric scooters and users.

```json
{
  "scooters": [
    {
      "id": "45601e30-01af-4dd9-9575-78661cf2a398",
      "assigned": true,
      "assigned_user": "user2",
      "decommissioned": false
    },
    {
      "id": "dacc2b76-57be-4fc5-8d02-36c7e43b6e0f",
      "assigned": false,
      "assigned_user": "",
      "decommissioned": true
    },
    {
      "id": "51a4a69a-7cd7-4a89-8c16-4b31547450b9",
      "assigned": true,
      "assigned_user": "user3",
      "decommissioned": true
    },
    {
      "id": "a35703f4-eea8-4e2b-be84-ad07fe603662",
      "assigned": false,
      "assigned_user": "",
      "decommissioned": false
    },
    {
      "id": "8fe477c9-02fc-4896-addb-c0fed19be4ff",
      "assigned": false,
      "assigned_user": "",
      "decommissioned": true
    },
    {
      "id": "48aaee55-f8ad-4c58-be67-73def318753a",
      "assigned": false,
      "assigned_user": "",
      "decommissioned": false
    }
  ],
  "users": [
    {
      "id": "user1",
      "current_scooter": ""
    },
    {
      "id": "user2",
      "current_scooter": "45601e30-01af-4dd9-9575-78661cf2a398"
    },
    {
      "id": "user3",
      "current_scooter": "51a4a69a-7cd7-4a89-8c16-4b31547450b9"
    },
    {
      "id": "user4",
      "current_scooter": ""
    }
  ]
}
```

written by:	Burak Atalay	approved by:	Vincent Ohana	review:	n.a.
unit:	Concept Reply GmbH	issue date:	21.11.2023	page:	2/5
privacy note:	*Reply confidential*				

Create a small program or script that implements at least the following:

- Load the json containing scooters & users into memory.
- Scooters that are not assigned to a user and are decommissioned should be deleted.
- Scooters that are not assigned to a user and not decommissioned should be assigned to users without a scooter.

Additionally, please add tests for the functions that you implement. Feel free to add any more relevant functionality to showcase your coding skills. Your program/script should be runnable, and it should logically log the operations that it is performing. When all operations are done the final version of the data should be printed.

written by:	Burak Atalay	approved by:	Vincent Ohana	review:	n.a.
unit:	Concept Reply GmbH	issue date:	21.11.2023	page:	3/5
privacy note:	*Reply confidential*				

Section 2: SOFTWARE DESIGN AND ARCHITECTURE

Scenario: Mobility-On-Demand

For an urban mobility customer, Concept Reply is developing a backend for urban transportation (think of another uber). The project seeks to provide users with a convenient and efficient way of finding and renting rides based on their location and preferences. The project scope only includes the backend of the application, since the mobile and web apps will be developed internally by the customer.

The following requirements have been communicated to Concept Reply and should be supported by the backend service:
1. create and manage users (incl. name, address and age)
2. create and manage user demands (incl. pickup / drop-off location and time, number of passengers, …)
3. create and manage cars (incl. relevant features like engine type, available seats, etc.)

The project utilizes a fleet of cars that transmit their real-time location. This way users can track the location of their cars, whereas administrators can track the location of all cars in the fleet. The project uses real-time data based on car availability and user preferences to recommend cars to the users.

Task 1: Database schema (30 min)
Design a database schema according to the requirements of the above *mobility-on-demand* scenario. Use for example entity-relationship diagrams for illustrating the database schema and be creative about properties of your entities.

Photographs of hand drawings are also acceptable.

Task 2: API Specification (45 min)
Write an API specification (a specification is not a web server!!!) for the mobility-on-demand scenario based on the database schema you created in the first task. The API should provide interfaces for accessing and manipulating data related to cars, users, and demands.

Feel free to showcase your creativity and attention to detail in designing an API that connects the frontend and backend systems.
Remember to include appropriate request and response structures, as well as any necessary validation mechanisms to ensure data integrity and security.

Rather focus on a few routes with a high level of detail than trying to cover everything, but only on a very high level.

written by:	Burak Atalay	approved by:	Vincent Ohana	review:	n.a.
unit:	Concept Reply GmbH	issue date:	21.11.2023	page:	4/5
privacy note:	*Reply confidential*				

TASK 3: SYSTEM DESIGN / ARCHITECTURE (30 MIN)

Design an architecture that can handle the complexities of our mobility-on-demand scenario. Please include the DB from task 1 and the API from task 2 in your design. If you are missing information, make educated guesses.

As non-functional requirements, please consider in your design:
- performance
- scalability
- reliability & availability

Feel free to sketch out the system architecture using diagrams to convey your vision effectively. Photographs of hand drawings are also acceptable.

written by:	Burak Atalay	approved by:	Vincent Ohana	review:	n.a.
unit:	Concept Reply GmbH	issue date:	21.11.2023	page:	5/5
privacy note:	*Reply confidential*				