NodeJS Test Assignment

General Requirements

- For implementation use TypeScript and provide all necessary data typings.
- For assignments assume that the latest version of NodeJS is used.
- Provide solution as `zip` archive with separate files for each assignment or as link to GitHub repository with these files uploaded.

Assignments

1. HTTP Request Handling

Write down code snippet using following requirements:

- Use only vanilla TypeScript, without additional libraries.
- Use REST Rick and Morty API for this assignment.
- Request all episodes using API.
- Replace URLs in “characters” array with character JSON objects taken from API.
- Log final array into console.

2. Counter Function

Implement the `counter` function according to requirements:

- Function accepts a number as the first argument. This number represents the initial value for the counter.
- If no value passed to a function, use `0` as initial value.
- Function returns array with two function:
  - First function allows us to get the current counter value.
  - Second function increases the internal counter value by one.
- Multiple calls of `counter` function create independent instances of counter

Example:

```
export function counter() {
}


const [getA, nextA] = counter(1);

getA(); // 1
nextA();
```

```
getA(); // 2

const [getB, nextB] = counter(10);

nextB();
getA(); // 2
getB(); // 11

nextA();
getA(); // 3
getB(); // 11
```

3. Services Interaction

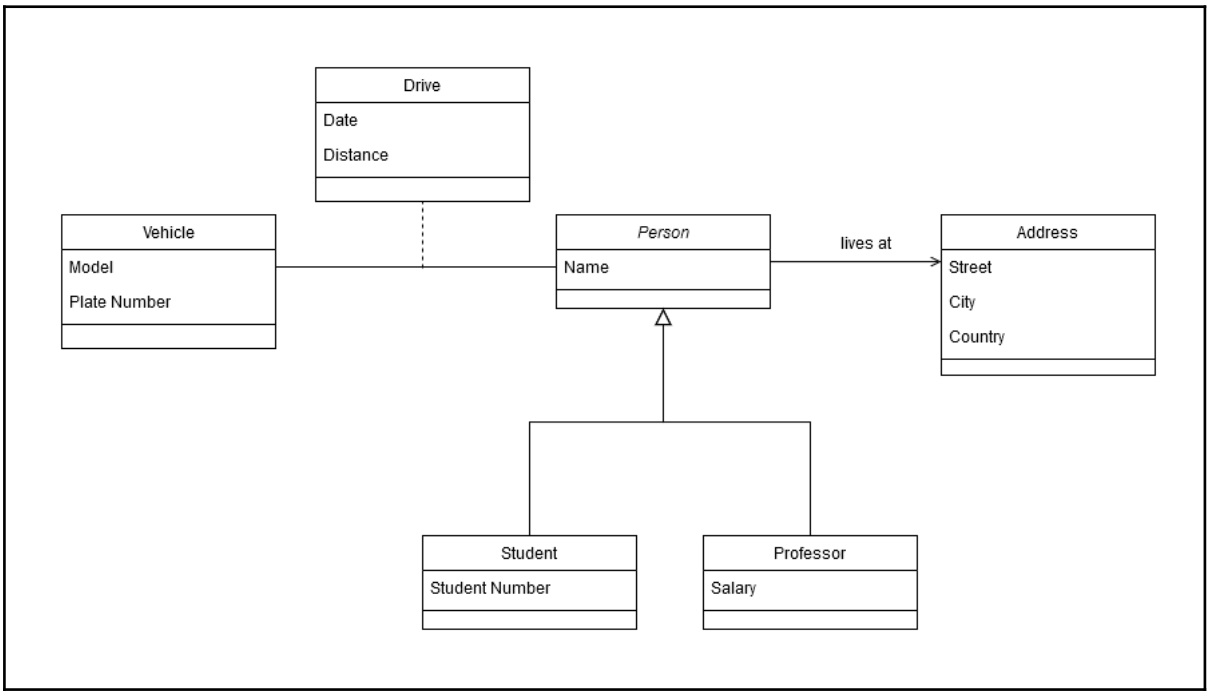
Imagine that we have two services - ServiceA and ServiceB. Both services have to communicate with each other in some way.

Given this information, please answer following questions:

1) What options do we have to establish such communication?
2) For each option describe what are the pros and cons of this solution?
3) For each option describe what are the cases the solution fits best?

4. Database Structure

Given following class diagram, prepare relational database structure description:



You can provide one of the following:

1) ER diagram.
2) Relational database initial migration or dump.
3) Models prepared using your favorite ORM framework.