

Goal of the application

To automate everyday work of an administrator at a service station.

Application functionality

The only user of the application should be the administrator of the service station. That user serves two client types: new clients and returning clients. Clients specify their name and surname to the administrator to register an order. The administrator should check whether the client is registered in the client base or not.

It should be possible to open a client card with personal data (see the description below) for existing users. Each client card has the following information about related cars:

- Make
- Model
- Year
- VIN

The administrator can edit the list of cars for the client: add, edit or delete (deletion is only possible if there are no orders related to the car).

It should also be possible to see the list of orders per each car. Each order contains the following information:

- Date
- Order Amount (US\$ 0,0 – 10,000)
- Order Status (Completed / In Progress / Cancelled)

The administrator can edit the list of orders for the car: add, edit or delete.

It should be possible to create a client card for new clients who haven't been registered in the application before. This card should have the following parameters:

- First Name
- Last Name
- Date of Birth
- Address
- Phone
- Email

Application requirements

The whole user interface of the application should be developed in English (Cyrillic support is not necessary).

The user interface of the application should be user friendly and efficient. That means users can quickly perform required actions without discomfort.

The system should be implemented as a web-application. The test task consists of two parts:

- Front-end part (Client). You can use any **JavaScript (AngularJS, React and etc.)**
- Back-end part (Server). You can use any **.NET Framework (Core or MVC)** or **Python** and MongoDB/MySQL/MSSQL for work with database.

There are two ways of implementing Intetics' test task:

- As a front-end developer – implement only the client using mock data for back-end part.
- As a full stack developer – implement both parts.

How we assess

Your work will be assessed based on the following criteria:

- Code structure and object model usage
- Usability of application user interface
- Application data structure and data processing
- Error handling and validation of user data

How to send work results

The results of the work should include the source code of the application, a database dump, short instructions about setting up and execution of the application. They should be uploaded to your personal (free) account at GitHub (<https://github.com/>). As a result, you should send us the link to the source code repository.

And remember: together we can build a better world. Long live and prosper.

