.NET DEVELOPER TEST TASK

- ✓ for execution
- ✓ version 1.1 dated 12 / 10 / 2015

1. GENERAL REQUIREMENTS

- ✓ Make sure to estimate the test project in hours before the actual start.
- ✓ Make sure to log the time spent on the test project with enough details.
- ✓ Make sure to use Github, a web-based Git repository hosting service, for the test project.
- ✓ Make sure to use Azure / AppHarbor in order to demonstrate the test project.
- ✓ Make sure to store the test project login / password information in one easily accessed place.
- ✓ Make sure to share the above mentioned information after the test task is complete.

2. FUNCTIONAL REQUIREMENTS

- ✓ The application is dedicated to covering a piece of the bus tickets sales cycle.
- ✓ The application allows to register, authenticate, and authorize users.
- ✓ The application allows the User to search for bus stops by name.
- ✓ The application allows the User to search for bus stops by departure bus stop name, arrival bus stop name, and departure date and time.
- ✓ The application allows the User to reserve tickets for the specified voyage.
- ✓ The application allows the User to buy out the reserved tickets.
- ✓ The application stores the data about users: first name, last name, E-mail, password, birth date.
- ✓ The application stores the data about bus stops: ID, name, description, status.
- ✓ The application stores the voyage data: ID, departure bus stop ID, arrival bus stop ID, departure date and time, arrival date and time, travel time, voyage number, voyage name, number of seats, one ticket cost.
- ✓ The application stores the data about orders: order ID, voyage ID, status (reserved, bought out).
- ✓ The application stores the data about tickets: ticket ID, order ID, passenger's first and last names, passenger's document number, passenger's seat number, status (reserved, bought out).

4. FOR CONSIDERATION

- ✓ Make sure to check the functional performance of the application as a finished product (**User Stories**).
- ✓ Make sure to store the passwords in a secure way.
- ✓ Make sure to use best practices of **C# and ASP.NET MVC**, in particular:
    - ■ use **OAuth** to authorize the users;
    - ■ use **Bootstrap** to develop a responsive and mobile first user experience (UX);
    - ■ use **MS SQL Server** to have a robust relational database management system;
    - ■ use **Entity Framework** to work with relational data using domain-specific objects;
    - ■ use **ASP.NET Web API** to build a qualified web API on top of the .NET Framework.
- ✓ Make sure to grant possibility to test the application for a business user (e.g. not from the console).
- ✓ Make sure to qualitatively use Git as a source control tool understanding the Git Flow and CI + CD principles.
- ✓ Make sure to provide qualitative comments in English when committing the code in Git.
- ✓ Make sure to write 1-2 Unit tests to cover certain core functionality at a maximum level.
- ✓ Make sure to develop the application with no contradictions to the test task in case there is lack of information.
- ✓ Make sure to develop a user-friendly and intuitive user interface (UI) from the User's perspective.
- ✓ Make sure to develop the test task on your own, the only help should come from Google / books.