Desktop/Web application practical knowledge test for job applicants

Created by: Oldřich Urbiš
Last revision: 3. 5. 2016

1. Content of this document

This document contains requirements for Desktop/Web application, which supposed to verify job applicants qualification, especially ability to write clean, structured and maintainable code, and also ability to find and use information which applicant doesn't know and it's necessary to complete the task.
Application itself doesn't have to be fully functional, but the code has to be good enough.
The form is more important than the result.
In case of any unclear requirement, job applicant is obligated to resolve this issue using his best judgment.
Time necessary to fulfill all the requirements is not limited, but 6 hours should be enough.
Job applicant is obligated to work on the application without anyone's help.
Internet sources can be used for obtaining necessary information.

2. Requirements

Applicant should create Visual Studio solution which will contain all the necessary source codes to create database tables and the application itself. Selection of the type of the application domain Desktop or Web is on applicant. It should be selected based on the job position requirements.
Application should contain at least stuff based on these requirements, any other content like user handling, cyphering or authentication is not needed.
If not required functionality is implemented, it might be taken into account during evaluation.

a. Technologies used for implementation (select that suits)

Web

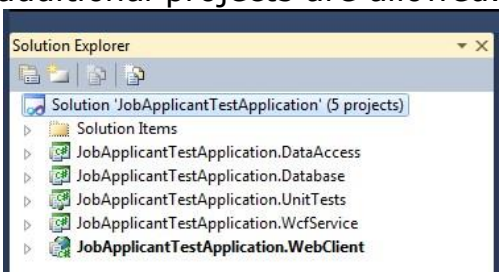
- ASP.NET C# Web forms
- ASP.NET MVC
- Angular.js
- Knockout
- Other JavaScript framework
- Bootstrap
- MS SQL
- jQuery
- WCF
- Web.API

Desktop

- Windows forms
- WPF
- MS SQL
- WCF
- Web services

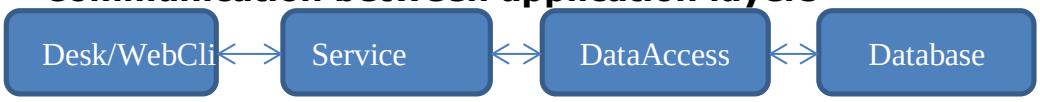
b. General solution structure

Solution of the application should like on the picture below. Any additional projects are allowed.



DataAccess – contains methods which queries database
Database – contains SQL queries for database creation
UnitTest – contains at least 1 unit test which will test some business logic
WcfService – contains definition for Wcf service method
Desktop/WebClient – contains GUI definition with all the necessary logic

Communication between application layers



c. General database structure

Implementation of a database layer should be performed using MS SQL. Data used for implementation are in attached Excel file. Applicant supposed to analyze data and design appropriate database structure including proper SQL scripts. Any necessary column can be added to the table. Foreign key constrains are required.
Only tables to store data from Excel files are necessary.
Loading data from Excel file into database is up to user, no import utility is required. Manual input with insert query generation is enough.

d. Desktop/Web client requirements

User interface must be implemented using one of the Desktop/Web application technologies and must consist of at least two screens/pages.
First screen/page must contain grid which will load data from the database, it will contain the list of motors with their properties. This screen/page must allow user to edit motor properties and to delete motors. Validation of input data is optional.
Screen/page layout is up to job applicant.
Second screen/page supposed to contain list of measurement for each motor during one hour. Each row must have timestamp and calculated difference between actual value and nominal value

(formula: nominal value – actual value) specified for each motor.
These values should be calculated in separated business logic (not in SQL), so it is possible to test it in unit tests.
Example of a row for motor with this property: Motor 1 – max current 8.7A

TimeStamp	Motor	Current	Difference	Revs.	Difference	Pressure	Difference
10:20:00	Motor 1	7	-1.7				

Grid should support paging and sorting, but it's not a must. Insert and filter functionality is not required.

e. Service

Application should use service, implemented with Wcf, Web.API or Web service, to obtain data from the database. Simplest way which will work is enough.

f. Data access layer

Data access layer should contain all the necessary methods which will handle loading, editing and deleting data from/in database.
Also all the necessary calculations should be implemented in this layer, but it doesn't have to be in the same project.
Usage of ORM framework like EntityFramework, PetaPoco, etc. is recommended, but not a must.

g. Unit test(s)

Unit test project should contain at least one unit test which will verify calculation of a difference between nominal and actual value, as it is described in paragraph *2.d*.

3. Attachments

- Excel file with data to be used - ASP.NETCandidatePracticalTestData.xlsx