Test task for Full Stack Software Engineer

02. Unit SEN

Exported on Jan 08, 2019

Table of Contents

1 Contents

2 Overview

This document describes the standard test task that is used to evaluate the skills and framework knowledge of applicants for the Full Stack Software Engineer position.

3 Goal

The goal of the task is to evaluate a number of criteria that are important for a Full Stack Software Engineer's daily job performance:

- Ability to understand business requirements.
- Ability to design a solution based on business requirements.
- Understanding of and proficiency in the tools and technologies that are commonly used in Westernacher's solution.
- Coding style.
- Eye for detail, correctness, ease of use.

3.1 Task description

3.1.1 Business requirements

We're building a small web application that manages user accounts in a database. The application has the following functionality:

- A list page that shows all user accounts in a sortable grid.
- Create new account dialog that adds new accounts to the database. The following properties should be supported
    - First name
    - Last name
    - Email address
    - Date of birth
- A Delete account dialog that removes accounts from the database.
- (optional). An Edit account dialog that allows to change account data of existing accounts.

Note that we don't accept any questions regarding these requirements. We'll leave the design of the solution up to you.

3.1.2 Architecture

The solution should use an architecture that's similar to the way applications are built at Westernacher. This means:

- An MVC architecture.
- A REST oriented architecture. Communication between the frontend and the backend services should only be done using RESTful HTTP services
- The front end should use a Javascript Framework, e.g. AngularJS, Angular, ExtJS, ...
- The back end should use Java 8 or above.
- Feel free to use a CSS framework like Bootstrap to do the styling or/and any CSS preprocessor.
- Feel free to use Spring, Spring MVC, Spring Boot...
- Feel free to use an ORM like Hibernate, JPA, etc.

3.1.3 Build

The build tool used should be Gradle (preferred) or Maven.
Make sure to add build tasks that make it easy to build and deploy the application from scratch.

3.1.4 System environment

The environment components should come from this list:

- Application server: Tomcat 7 / 8 / 9
- Database: HSQLDB, MySQL, Oracle, Postgre, Mongo, ...
- Java 8 or above
- Windows 7/8/10, Mac OS X 10.9 or higher, Ubuntu LTS 12.04 or higher

3.2 Documentation

Please, provide a short documentation that explains how we can build and deploy the application for testing. Make sure you list the system prerequisites for running the application.

We don't need end user or code documentation, we will figure that out ourselves.

3.3 Final thoughts

Try to make sure that your application is an example of a state-of-the-art Java Web Application with a good-looking, intuitive UI. These are the kinds of web applications we're building for our customers so you should convince us that you're the right person to join our team.

Try to keep the codebase as compact as possible. Try to programming defensively.Think for the small details.

Be creative. If you think you found something nice you want in the application to show us your skills: go ahead!

3.4 Reward

We don't want you to create this test application for free! If your work is approved by our reviewers and you pass your trial period at Westernacher, we'll pay you a bonus of 500 LEV as a compensation for your efforts.