jBilling Practical Test

I am happy to let you know that you passed the previous test, which tests general knowledge about Java, Spring and SQL.

This new test is a more practical exercise, where you need to write some code. The requirements are fictional and simplified so you can complete this in a reasonable time, but keep in mind that **this is your chance to show us** how well you can implement requirements. Do your best!

The Prizy Pricer

The company ‘Prizy’ wants to start conducting surveys of prices for different products, and then based on the collected information, calculate an ‘Ideal Price’ for a product.

The idea is to have a number of workers walk into a store, and with their iPads access a web application where they enter prices as they see them. In real time, an administrator will be able to see the entered prices using the same web application and the application will provide an ‘Ideal Price’ for a specific product.

There will be three basic screens (web pages) for this app:

Product loader

This is what the workers use to enter prices. It is basically a simple form with a bunch of fields:

- Store
- Product bar code (to id the product uniquely). This has to match a table with unique bar codes. That table also has a description for the product
- Price
- Notes

Product list

This is used by the administrator to browse and find a specific product. There can be thousands or millions of products

Product viewer

When a product is selected from the list, it shows the information about it:

- Bar code
- Product description
- Average price

- Lowest Price
- Highest Price
- Ideal Price. This price is calculated by taking all the prices of this product, removing the 2 highest and 2 lowest, then doing an average with the rest and adding 20% to it. It is known that this complicated formula **will be changed often**. Even different installations of Prizy Pricer will be using different versions of this ideal price formula. So the key here is to try to minimize the impact to the application when a new formula needs to be put in place (hint: use something like the ‘Strategy’ design pattern).
- Number of prices collected
- Anything else that might be useful?

What to implement

- Entering a new price for an existing product.
- The list of products, with a way to browse pages and a way to find a product by bar code.
- The product viewer screen, which is read only.

You do not need to bother with usernames and passwords (we don’t need to know who did what, or any form of authentication).

Testing

You have to provide automated tests for this application. In fact, the first thing we’ll evaluate is the quality of the testing in place. Make sure that the test pass! Most people fail here. If you need us to do something before running the tests (like initialize the DB), say so in some instructions.

Environment

This application has to be done using Grails version 2.4.3 (preferred) or 2.0.3 (acceptable). You have to use one of those versions. The DB has to be PostgreSQL (any version). If you are not familiar with Grails and PostgreSQL, do not worry. You will be able to learn Grails very fast thanks to your strong Java skills. Then you will be surprised at how easy is to put together an application like this with Grails.

Delivery

- A zip file with all the necessary files. Email this to abhishek.pandey@appdirect.com with the Subject “Java Practical Test”. Don’t forget to include your name to this email.
- A simple way to build the application (do not submit the binary).
- A simple way to run the tests.
- A DB dump with the database initialized, or some other way so we can use the application and see some data in it (products and prices for those products).

- Add to the zip file a README.txt file with any and all the instructions that we need in order to run the application and the tests.

Evaluation

We are not testing how good you are at Grails, but how good you are at designing and implementing a piece of code from scratch, specially the quality of the tests and design to ease the implementation of new ideal price formulas.

The position you are applying for is a Java position, but you will need to know your way around Grails and, most important, be able to learn new things fast.

Enjoy coding.