Work Test for Java/J2EE Developers

1 Overview

The purpose of this test is to give bwin Games an idea of an applicant's ability to solve a server programming assignment in a clear and structured way.

The test is expected to take 6-8 hours to complete.

1.1 Background

The task is to implement a simple online store application called "Pokerstore". In the application a customer can purchase items using his credit. The inventory of available items, the prices and the credit for each customer is managed by a store employee through the same application.

1.2 Current status

The application contains two service interfaces that need to be implemented:

- **StoreFrontService** – intended for customers using the online store
- **BackOfficeService** – intended for "Pokerstore" employees

A simple test case "PokerstoreTest" is available that demonstrates the basic functionality of the interfaces.

2 What to do

This section lists features that must be implemented to finish the assignment.

2.1 Implement Core Functionality

All methods in the StoreFrontService and BackOfficeService interfaces must be implemented to support employee functions and customer purchases. The following basic business rules apply:

- It must be possible to order products that are not currently in stock
- A purchase is not allowed if the customer has insufficient credit

2.2 Extend Functionality

The following additional functionality not expressed in the existing interfaces or test suite must be implemented:

- It must be possible for an employee to ask for a report of products out of stock.

You need to extend the BackOfficeService interface and the test suite to support this capability.

2.3 Non-functional Requirements

- The store must be able to handle *concurrent user requests*.
- The store must use a RDBMS for persistent data.
- The store may either run standalone in-process (with the JUnit test) or in a container managed environment.
- A GUI is not required.

3 How to do it

These are requirements on the way the work test is to be performed:

3.1 Implementation

- Working implementations of the BackOfficeService and StoreFrontService interfaces.
- A working implementation of the PokerstoreFactory that must be named "com.ongame.pokerstore.PokerstoreFactoryImpl" and provide instances of the service implementations.
- A working implementation for reporting products out of stock, including tests

All source code you write for the solution *must* be placed in the `solution` tree. You are allowed to create new packages.

Important! *It is not allowed to modify existing source code including the test class. Doing so will automatically fail the test.*

3.2 Documentation

The code you write must be clear and self-documenting. A *brief* description of the solution with motivation of any design choices made must be included (UML is not required). All documentation must be in either plain ASCII text or PDF format. Word documents are not acceptable. More details are available in the Deliverable section.

3.3 Target platform

The solution must use the Java2 platform, preferably JDK 1.5 or 1.6. Using third party tools and libraries is allowed and encouraged, as long as their license allows free unlimited use for development purposes.

Acceptable container choices are JBoss, Tomcat, Jetty or Winstone. If Jetty or Winstone is used, the container must be bundled in the solution delivery package with running instructions.

The only acceptable RDBMS choice is MySQL. Code and data must be compatible with version 5.x. A suitable JDBC driver is provided in the original *lib* directory.

3.4 Build and install

The solution must be built with Ant (1.7.x). A skeleton build.xml file is provided. The task names in the skeleton file must be used. Any additional tasks may only be used internally within the build.xml file (not called from command line). No environment variables may be required by the build.xml file other than JAVA_HOME and ANT_HOME.

Any additional properties required can be placed in a build.properties file. The solution shall be possible to build and install using the "dist" target. After building and installing, the provided PokerstoreTest shall be run by the "test" target in the build.xml.

Any additional jar-files necessary shall be bundled with the solution delivery placed in the *lib* dir. Ant should not be bundled in the solution.

If you wish, you may also use Maven2 to build the solution. A skeleton pom.xml is provided for that purpose. Note that it is **not** a requirement for your solution to support Maven. For further instructions see building.txt.

4 Deliverable

The delivery must consist of a single zip file. The zip shall contain the original directory hierarchy along with any added files or directories needed for the implementation, including:

- Full source code including new classes and any modified build files, SQL files and so forth.
- Any new 3rd party libraries placed in the lib directory.
- A brief SOLUTION.TXT file (or PDF) describing to the reviewer the following information:
    - Your major design choices and the reason for those choices
    - The exact version of JDK used
    - How to deploy the solution in case it requires a container
    - The location of any SQL scripts and how to execute them
    - The names of the files you have created with an optional high level description of the purpose of each

Reminders:

- All Java source code you write must be placed in the solution tree.
- The project must be able to build with Ant.
- Maven is optional.

5 Evaluation

In Bwin's evaluation of the submitted work test, we first run the test and ensure it fulfils all requirements. Then we focus on:

- The overall object-oriented design, the choice of data structures and patterns
- Robustness and transactional safety
- Technology choices and motivations
- Coding standards and reliability
- Code clarity and structure, general extensibility and maintainability
- Javadoc source documentation and comments
- Test coverage and testability

If you are an experienced J2EE developer we also expect you to demonstrate this by using J2EE techniques where applicable. Do not hesitate to use advanced techniques, even though Pokerstore is a very small server application.

Good luck!