**Back Office Systems - Mini Project**

**For: Senior Software Engineer / Full Stack Developer**

**Overview**

This mini project is designed to gauge your existing knowledge and skillset and demonstrate how you approach problems as a Senior Software Engineer/Full Stack Developer. The tasks in this project reflect a simplified version of the type of work you can expect in this role. You should consider doing things like **Javadoc**, **unit tests**, and **industry standard naming conventions** as you'll want to demonstrate your understanding of best practices. In addition, you may want to check your code using **check style** (use Google’s default check style settings) as this is the standard our team uses. Please utilize **Maven** and **create a README.md** file to document how to compile and run your application, as well as any other important details.

Good luck and happy coding!

**Part One – ETL Demonstration**

**Description**

Create ETL jobs that load Tab Separate Values (TSV) files into MySQL database tables (You need to create SQL statement to create the tables too). Sample TSV files are provided with this write up. The first row contains the column names and each additional row contains values to be inserted into the corresponding database table.

Your solution must be configurable so it can be used to target any local or remote MySQL database, given the correct credentials.

**Gotchas**

Column order of data dump files can be random. Your program should be able to process for any permutation of columns.

**Part Two – Backend Demonstration**

**Description**

Program backend web service endpoints that allow CRUD operations on individual records of that table. Lastly, create an endpoint that provides the sum of inventory for all brands in the database. Endpoints should be able to provide the response in both XML and JSON formats. Your backend must be Java based and utilize the Spring framework. Be sure to follow standard rest conventions (http verbs and naming convention). Please utilize Swagger UI.

**Endpoints Checklist**

• Create Endpoint for Brand.

• Read Endpoint for Brand.

• Update Endpoint for Brand.

• Delete Endpoint for Brand.

• Sum of Inventory Endpoint for all Brands.

**Part Three - Front End Demonstration**

**Description**

UI Part 1) Interactive Display: Display the Brands’ Sum of Inventory in an interactive tabular format/report, where you can sort the data by clicking on the column headers. Data should be obtained by calling the endpoints you created.

UI Part 2) Create a basic front-end entry form with responsive design that will utilize your brand create end point.

**Check List**

• Tabular Display with Sorting Features

• Responsive Form for Creates

**Gotchas**

Are you utilizing common JavaScript libraries/technology?