Student: Simge Haksal
ID: 923606
Course Name: Advanced Computer Programming
Project's Name: XML Processing
Programming Language: Java

The aim of my project is writing a program that provides a conversation between a database and an XML file. I prefer to divide the project into **two different parts**:

1. Given database, the program will read the data and produce an XML file to present the data in database.
2. Given in XML files, the program will insert the information given in the XML files to the same database (it will read the different XML files at the same time).

I will prepare a file which contains necessary SQL commands to create tables and also populate them with sample data. I am thinking of preparing a file related to Bulletin Board System on MySQL Workbench. I will use WampServer (phpMyAdmin). For me, **three tables** are enough to develop the application. These tables can be:

1. "**exam**" table: keeps track of exams. It stores the exam number, exam title, the time allowed for this exam, and the number of questions per page.
2. "**question**" table: keeps track of questions in a given exam. It stores information about the exam, question number, question text, correct answer for a given question.
3. "**answerOption**" table: keeps track of different options for a given question in a given exam.

Let's discuss the specifications. I will complete the project with this order:

1. I will parse an XML file using DOM parser, and I will also parse the file and traverse the resultant tree and insert the extracted info into the database.
2. I will reconstruct the XML file from the database

The Xerces Java Parser supports the XML recommendation and contains advance parser functionality, and it is the most popular one. To sum up, I will apply **three techniques** from six of them that you published on your website. These are:

1. ADT, Parametric Types, Object Orientation
2. Concurrent Programming
3. Generic Programming and Metaprogramming

But there is a restriction for Java, I cannot use the metaprogramming technique with Java programming language. Not all metaprogramming involves generative programming. If programs are modifiable at runtime or if incremental compilation is available (such as in C#, Forth, Frink, Groovy, JavaScript, Lisp, Elixir, Lua, Perl, PHP, Python, REBOL, Ruby, Rust, SAS, Smalltalk, and Tcl), the techniques can be used to perform metaprogramming without actually generating source code. Because of this, I will only use these techniques:

1. ADT, Parametric Types, Object Orientation
2. Concurrent Programming
3. Generic Programming

If it is not a problem for you, I want to start to develop as soon as possible.

Block Diagrams

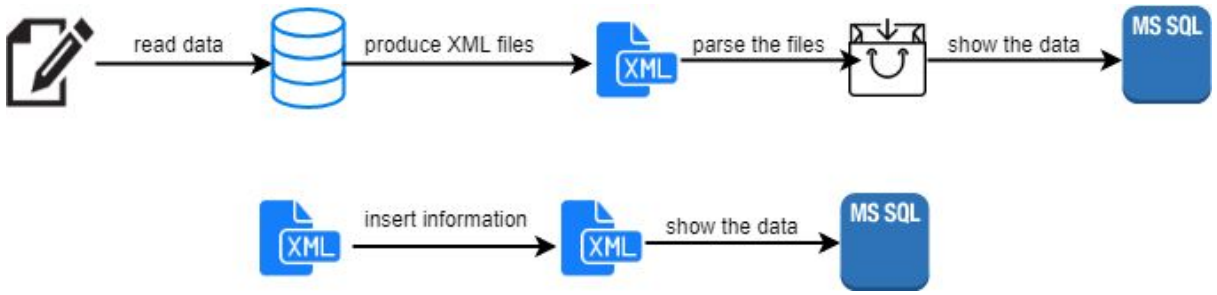

Figure 1. *Simple Visual for the Process*

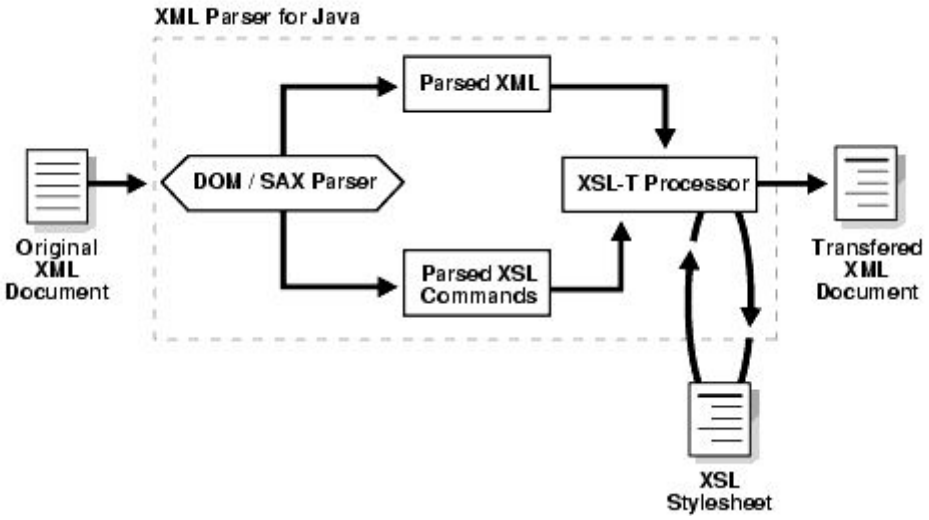

Figure 2. *XML Parser for Java*