# Programming Skills Assessment, Part II

This exercise is intended to give you an opportunity to demonstrate your skills in software engineering, object-oriented analysis and design, and implementation of quality software. In addition, you will have the opportunity to present your solution and approach to other software engineers, as you might during a technical review with an engineering team. Please approach this project as you would in a real-world production environment.

You previously created an application that loads a Project containing study and experiment information, displays user-selected content in a window, and provides basic sorting features. Please extend this application to include the following new features, described below.

You will have approximately 90 minutes for your implementation. You are not expected to complete all of these during that time–just do as much as you can.

**1) Search feature**

Recall that the previous exercise involves sequence ofexperiments, e.g.:

<Study ID="Study01">

<Identity>

<Name>Cancer Study 001</Name>

<Description>Use of Benzene for curing cancer</Description>

</Identity>

<Experiments OrderBy="Index">

<Experiment ID="Expt0009" Index="15" CreationDate="07/10/2015 17:31:32">Create Benzene powder at 90 degrees Celsius</Experiment>

<Experiment ID="Expt0014" Index="14" CreationDate="07/10/2015 08:39:00">Create Benzene powder at room temperature</Experiment>

…

</Experiments>

</Study>

Sample input is provided in the file Project.xml.

Our hypothetical customer has now requested that cross-referencing search capabilities. A second “experimental record” contains further details about the execution of these experiments, e.g.:

<ExperimentalRecord>

<ExperimentRun ExperimentID="Expt1012" LabTechnician="JCC" RunDate="2016-12-21" />

<ExperimentRun ExperimentID="Expt1005" LabTechnician="PJM" RunDate="2015-01-09" />

<ExperimentRun ExperimentID="Expt0009" LabTechnician="AJG" RunDate="2015-01-01" />

. . .

</ExperimentalRecord>

Sample input is provided in the file Experimental Record.xml.

The application should now allow the user to search for an experiment by *Index*, *CreationDate*, or the value of the *Experiment* element. When an experiment is specified, then the values of the corresponding *LabTechnician* and *RunDate* attributes should be displayed.

For example, given the above input, if the user specifies *Index 15*, then the application should display that the lab technician was “AJG” and that the run date was “2015-01-01.”

**2) Filter feature**

When a study is selected, the application currently displays all of the experiments within that study. However, in some cases, it may be desirable to simplify the display. Add a feature that allows the user to specify the initials of a lab technician, in which case only the experiments run by that technician will be displayed. Specifying a filter should not modify the data stored in the data files or database.

**3) Report feature**

Export a simple summary report of studies and experiments in your project. If your application is client-based, use an HTML format for your report. If your application already is web-based, then use a text file format.

You will have approximately 90 minutes for your implementation. At the end of that period, you will be asked to give a brief technical presentation about your solution.