Software Documentation : Java Custom LIMS - SQUIT Spec Mapper

This page last changed on Mar 25, 2010 by klause.

1. Scope

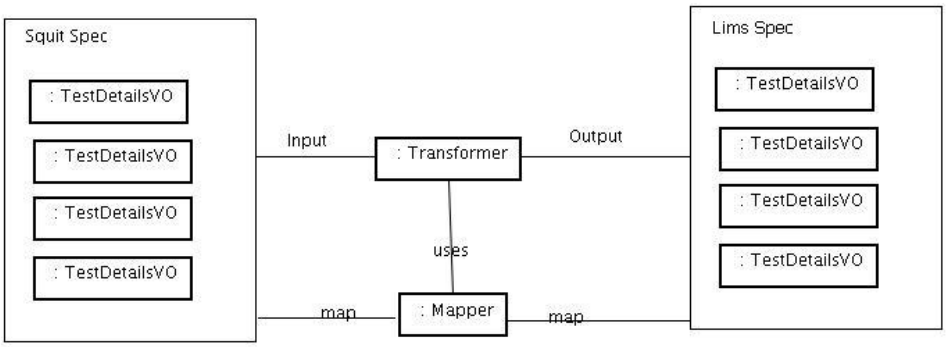
The Laboratory Information Management System (LIMS) is an inventory tracking system. The system is a JEE web application that relies on the application server logging to track execution issues.

SQUIT is a system which receives supplier information from customers and export it to SAP systems.

1.1 Overview

This middleware component will do a data mapping between supplier information retrieved by SQUIT so it can be imported on LIMS. Both systems provide Java classes that list a supplier information record called "Spec". Specs are lists of TestDetailsVO objects described in 2.1.2.

The purpose of this component is to create a mapping tool which would allow to map SQUIT specs and LIMS Specs through a Mapper. For each TestDetailVO on a SQUIT Spec the "Mapper" would store a corresponding LIMS TestDetailVO and an operation that should be applied to SQUIT TestDetailsVO values to convert it to a LIMS
TestDetailsVO as shown on Picture 1.



Picture 1

1.1.1 Version

1.0

1.2 Logic Requirements

1.2.1 List Specs Mapped

1. The user selects Spec Mapping in LIMS application menu (this item is out of the scope and does not need to be in design)
2. A page listing the existing mappings is shown (Spec mapping list). The results of the list must me paged in 20 records per page. The user also can use the search fields to filter the records shown.

1.2.2 Add a Spec Mapping

1. In the spec list page (Spec mapping list) the user clicks in Add button
2. The edit page (Spec mapping edit) is shown with empty fields
3. The user clicks in Browse button to select a SQUIT Spec.
4. The SQUIT spec browser window pops up (SQUIT spec browser popup)
5. The user filter the SQUIT specs using the search fields
6. The user selects the SQUIT spec to be mapped by clicking in the corresponding record line
7. The edit page is updated with the selected SQUIT spec
8. The user clicks in Browse button to select a LIMS Spec.

9. The LIMS spec browse window pops up (LIMS spec browser popup)
10. The user filter the LIMS specs using the search fields
11. The user selects the LIMS spec to be mapped by clicking in the corresponding record line
12. The edit page is updated with the selected LIMS spec
13. The user optionally selects the corresponding unit
14. The user optionally selects operation (multiply or divide)
15. The user optionally selects operand value
16. If the user clicks save:

    16.1. The operand field is validated to check whether it is a number. A error message is shown.
    16.2. It is verified if this mapping already exists. A error message is shown.
    16.2. The record is stored in the database if all validations passed.

17. If the user clicks cancel, the data is abandoned and it returns to list page

1.2.3 Edit a Spec Mapping

1. In the spec list page (Spec mapping list) the user clicks in existing spec mapping record
2. The edit page (Spec mapping edit) is shown with selected mapping data
3. The user may click in Browse button to select another SQUIT Spec.
4. The SQUIT spec browser window pops up (SQUIT spec browser popup)
5. The user filter the SQUIT specs using the search fields
6. The user selects the new SQUIT spec to be mapped by clicking in the corresponding record line
7. The edit page is updated with the new selected SQUIT spec
8. The user may click in Browse button to select a LIMS Spec.
9. The LIMS spec browse window pops up (LIMS spec browser popup)
10. The user filter the LIMS specs using the search fields
11. The user selects the new LIMS spec to be mapped by clicking in the corresponding record line
12. The edit page is updated with the new selected LIMS spec
13. The user optionally selects the corresponding unit
14. The user optionally selects operation (multiply or divide)
15. The user optionally types operand value
16. If the user clicks save:

    16.1. The operand field is validated to check whether it is a number. A error message is shown.
    16.2. It is verified if this mapping already exists. A error message is shown.
    16.2. The record is updated in the database if all validations passed.

17. If the user clicks cancel, the data is abandoned and it returns to list page

1.2.4 Delete Spec Mapping

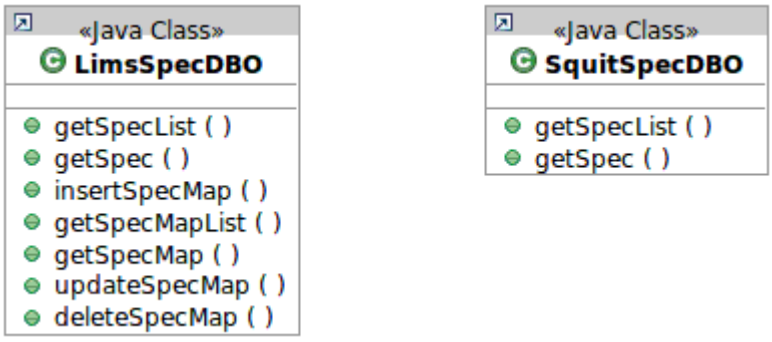
1. In the spec list page (Spec mapping list) the user selects the specs that should be deleted
2. The user clicks in Delete button
3. The system shows a message dialog asking the user to confirm the deletion of selected mappings
4. If the user confirms, the mappings are deleted from the database
5. If the user does not confirm, no actions are taken

1.2.5 Use of existing DBO classes

In LIMS, database data is accessed using its Database Objects (DBOs). Normally, each domain class has a corresponding DBO, which is responsible for acquiring a database connection and executing SQL statements transparently.

The design does not need to specify database access, ORM mapping and JNDI look because the existing DBO classes makes it transparent.

The mapper should use information from existing SquitSpecDBO and LimsSpecDBO (picture 2) classes to create, delete, show and update mappings on the database.



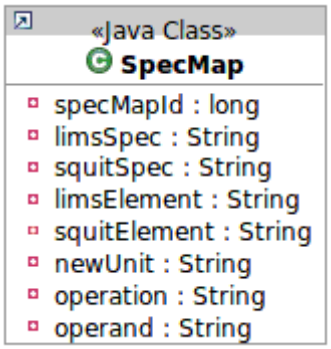
Picture 2

(the stub for these classes are attached: LimsSpecDBO, SquitSpecDBO)

1.2.6 Spec Mapping class

Mapping should be able to map a SQUIT spec name into a LIMS spec name, an SQUIT element name into a LIMS element name, change a SQUIT unit into a LIMS unit and Squit value can be transformed(multiplied or divided) by a constant to a LIMS value.

The mapping class must be provided in the design. Spec mapping class solution is a design decision. Just for clarification, a suggestion for this class is shown on Picture 3.



Picture 3

Database table design for the mapping class is out of the scope.

1.2.7 Log of messages

All error messages sould be logged using Log4J logging framework.

1.2.7 Correspondence between TestDetailsVO and GUI fields

The GUI fields (where applicable) are linked to TestDetailsVO class as follow:

TestDetailsVO.type = Spec code
TestDetailsVO.SampleID = Spec name
TestDetailsVO.Element = Element
TestDetailsVO.Units = Unit

1.3 Required Algorithms

N/A

1.4 Example of the Software Usage

It would be possible to use JSP UI to list/search, view, create, update and delete mappings and to configure transformations for these mappings. An example of a mapping is as follow:

	SQUIT	LIMS
Spec name	01L8582--J48496	01L8582
Element	AU	AU- GOLD
New unit	(PPM)	PPB
Operation	N/A	Divide
Operand	N/A	1000

1.5 Future Component Direction

None

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

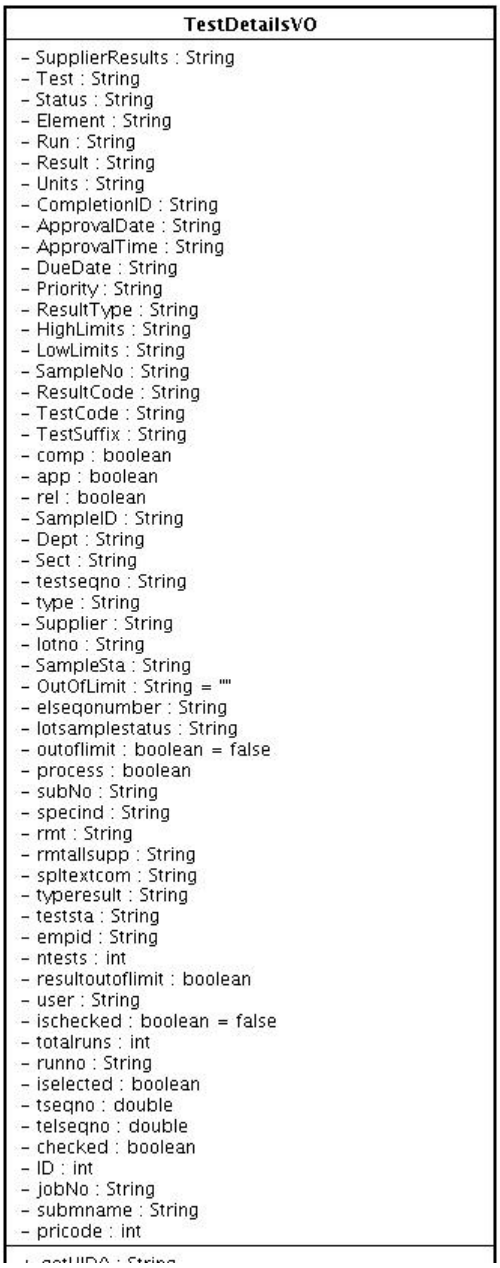
A JSP should be provided to allow to create, update and delete mappings between SQUIT specs and LIMS Specs. This JSP should be based on a MVC model and use Struts 1.1 actions to access the function provided by the Mapper class.

The pages must be compatible with Internet Explorer 6 and Mozilla Firefox 3.

2.1.2 External Interfaces

SQUIT is a external system that interfaces with LIMS to provide some additional data. The specs from SQUIT must be accessed through SquitSpecDBO.

A Spec is represented by a TestDetailsVO class:



Picture 4

(the stub for this class is attached: TestDetailsVO.java)

2.1.3 Environment Requirements

- Development language: Java 1.5
- Compile target: Java 1.5
- DB2 database 8.1
- Websphere 6.1

2.1.4 Package Structure

- com.ibm.lims.businesslogic - Business logic classes
- com.ibm.lims.actions.transformer - Struts action classes for transformer/mapper

- com.ibm.lims.databaseobject - Database objects

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

N/A

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- J2EE 2.4
- UI should be based on Struts 1.1
- UML 2.0

3.2.2 TopCoder Software Component Dependencies:

Do Not Include Any TopCoder Generic components in this design.

3.2.3 Third Party Component, Library, or Product Dependencies:

- Struts 1.1 (Struts 1.x)
- Log4J 1.2 (Log4J 1.2)

3.2.4 QA Environment:

- IBM Websphere 6.1
- IBM DB2 8.1

3.3 Design Constraints

The component design diagrams must use UML 2.0 notation.

The design should state that all error messages must be logged using Log4J.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification