PROJECT REPORT

ON

Code Coverage Using JaCoCo

FOR

PTC Software (India) Pvt. Ltd.

BY

Kiran Shejwal

SAVITRIBAI PHULE PUNE UNIVERSITY
MASTER IN COMPUTER APPLICATION
MAHARASHTRA EDUCATION SOCIETY's
INSTITUTE OF MANAGEMENT AND CAREER COURSES
(IMCC), PUNE-411038
2016-17

Maharashtra Education Society's

INSTITUTE OF MANAGEMENT AND CAREER COURSES (IMCC)

(Recognized by Savitribai Phule Pune University & Approved by AICTE)

131, Mayur Colony, Kothrud, Pune 411 038

Telefax . +91-20-25466271, 25463453 • E-mail: directorimcc@ vsnl.net

DR. V. H. INAMDAR Director
Tel: +91-20-25445067 (Director) Telefax: +91-20-25466271, 25463453 (Registrar)
Tel: +91-20-25440196 (TPO) Web Site: <www.mesimcc.org>

Ref. No : MCA-MCM/Project/328/2016-17

Date : 05/4/17

CERTIFICATE

This is to certify that the Project Report entitled ***"Code Coverage Using JaCoCo"*** is prepared by ***Kiran Shejwal a*** student of ***M.C.A.*** Course for the Academic Year 2016-17 at M.E.Society's Institute of Management & Career Courses (IMCC), Pune - 411038. M.C.A Course is affiliated to Savitribai Phule Pune University.

To the best of our knowledge, this is original study done by the said student and important sources used by him/her have been duly acknowledged in this report.

The report is submitted in partial fulfillment of M.C.A Course for the Academic Year 2016-17 as per the rules & prescribed guidelines of Savitribai Phule Pune University.

Dr. Santosh Deshpande
HOD, Computer Dept. IMCC

Dr. Vikas Inamdar
Director, IMCC

External Examiner

Internal Examiner

ptc

March 23, 2017

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Kiran Shejwal** final year MCA student of **Institute of Management & Career Courses (MCA), Pune** has been associated with us as **"Project Intern"** from July 2016 till date.

As per Company policy we cannot provide any kind of source code or executable to the Trainee. During project the candidate has worked with utmost dedication & sincerity. The candidate has worked on:

"Code Coverage Using JaCoCo"

The project was carried out in partial fulfillment of requirement for the award of degree of **Master in Computer Applications (MCA)** from **Savitribai Phule Pune University** for the academic year 2016-2017.

Your Sincerely,

for PTC Software (India) Pvt. Ltd.

Rucha Mantri
Assistant Manager - Human Resources

CERTIFICATE

This is to certify that **Kiran Shejwal** has completed the project work entitled "***Code Coverage Using JaCoCo***" under my guidance. The report is submitted in partial fulfillment of M.C.A. Course for the Academic Year 2016-2017 as per the rules & prescribed guidelines of Savitribai Phule Pune University.

His/Her work is found to be satisfactory and complete in all respects.

Dr. Minakshi More
(Internal Project Guide)

Acknowledgement

I wish to express deep sense of gratitude towards PTC Software (India) Pvt. Ltd. for providing me the help opportunity to work with the company and to provide wonderful and challenging environment for development of "Code Coverage Using JaCoCo".

I would like to thank Dr. V.H. Inamdar, Director, IMCC, Dr. Santosh Deshpande, HOD of computer department, IMCC, Dr. Manasi Bhate, TPO of IMCC, Dr. Minakshi More for their valuable guidance and kind co-operation throughout the period of the work undertaken.

Kiran Shejwal

INDEX

CHAPTER 1: INTRODUCTION

1.1 Company Profile

PTC Software is at the early stages of a fundamental transformation, marking what could be one of the most significant disruptions to the manufacturing industry since the Industrial Revolution. As one of the world's largest and fastest-growing technology companies, PTC helps manufacturing leaders address these transformative forces to achieve and sustain product and service advantage.

PTC solutions for Computer Aided Design (CAD), Product Lifecycle Management (PLM), Application Lifecycle Management (ALM), Service Lifecycle Management (SLM), and Internet of Things (IoT) enable process transformation and deliver closed-loop lifecycle management for products and services that are increasingly smart and connected. Our solutions help you optimize the activities within individual functions of your organization, and align them across your entire enterprise—from engineering to supply chain and manufacturing, to sales and service.

Today, PTC works with more than 28,000 businesses around the world to help them create, operate, and service products in rapidly-evolving, globally distributed manufacturing industries, including industrial equipment, automotive, high tech and electronics, aerospace and defence, retail, consumer, and medical devices. By partnering with PTC, you can cultivate and exploit your best product and services ideas in an increasingly smart and connected world.

1.2 Existing System and Need for System

Code coverage data gives us an important insight on how effective our tests are, what parts of our source code are thoroughly executed. You can also look at the code coverage report and find out specific areas of code which are not exercised by our tests. If you are following TDD, then you must and will be having 100% coverage. The test early principle helps you in detecting and fixing bugs early in the life cycle of your project. If you want your project to be as successful as Spring, then you know what % of coverage you need. Let's all strive to "Aim for the highest".

Code coverage reports should be used on a daily basis

to: 1.See what percentage of tests actually test our code.

2. Improve the coverage by writing additional tests as more functionality is added.

1.3 Scope of Work

Code coverage and test coverage metrics are both measurements that can be useful to assess the quality of your application code.

Code coverage is a measurement of how many lines/blocks/arcs of your code are executed while the tests are running. In this expert response, you'll learn how quality assurance professionals use both of these metrics effectively.

Coverage information has to be collected at runtime. For this purpose, JaCoCo creates instrumented versions of the original class definitions.

The instrumentation process happens while execution during class loading using Java agents. There are several different approaches to collect coverage information. For each approach different implementation techniques are known.

The Java agent is loaded by the application class loader. Therefore, the classes of the agent live in the same namespace like the application classes which can result in clashes especially with the third party library ASM. The JaCoCo build therefore moves all agent classes into a unique package.

Code coverage helps in identifying the code which is not covered by the test case execution so that we can write the new test cases to improve the Coverage.

Code Coverage is a measurement of how many lines/blocks/path of your code are executed while the tests are running.

Code coverage is collected by using a specialized tool to instrument the binaries to add tracing calls and run a full set of tests against the instrumented product. A good tool will give you not only the percentage of the code that is executed, but also will allow you to drill into the data and see exactly which lines of code were executed during particular test.

while code coverage is a good metric of how much testing you are doing, it is not necessarily a good metric of how well you are testing your product. There are other metrics you should use along with code coverage to ensure the quality.

1.4 Operating Environment –Hardware and Software

Client Software Requirements:

- Windows 7 64 bit
- Web Browser: IE 11, Mozilla 10.0X ESR

Client Hardware Requirements:

- RAM 2GB
- 2 processors
- 50 GB HDD for software components

Server Hardware Requirements:

- 8GB RAM
- 2 processors
- 50 GB HDD for software components

Server Software Requirements:

- Windows Server 2008 R2 64-bit

CHAPTER 2:
TESTING ENVIRONMENT

2.1 Objective of system:

Identify the dead code: After running all the defined use cases, if the coverage output shows that if some functions are not called, after analysing we can identify whether the code is un-touched as no required use case exists or code is dead code (i.e. not required)

Identify the missing test cases: Coverage output can also be useful in identifying the extra tests (exceptional cases), which we were missed out earlier after analysing the coverage report.

Function and line coverage: Coverage output reports how many functions and how may lines have been covered in the application code. This also gives "hit" of the functions, i.e. how many times a particular method is been called. This gives additional information such as which code is being accessed more.

2.2 User Requirements:

Code coverage analysis is the process of:

Finding areas of a program not exercised by a set of test cases,

Creating additional test cases to increase coverage, and

Determining a quantitative measure of code coverage, which is an indirect measure of quality.

An optional aspect of code coverage analysis is:

Identifying redundant test cases that do not increase coverage.

2.3 Detail Description of Technology Used:

Java: Java is a general-purpose computer programming language that is concurrent, class-based, oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform.

JaCoCo: JaCoCo is an open source toolkit for measuring and reporting Java code coverage. JaCoCo is distributed under the terms of the Eclipse Public License. It was developed as a replacement for EMMA under the umbrella of the EclEmma plug-in for Eclipse.

Features

JaCoCo offers instructions, line and branch coverage. In contrast to Clover, which requires instrumenting the source code, JaCoCo can instrument Java byte code using two different approaches:

> like JCov on the fly while running the code with a Java agent,
>
> like JCov and Cobertura prior to execution (offline)
>
> And can be configured to store the collected data in a file, or send it via TCP. Files from multiple runs or code parts can be merged easily. Unlike Cobertura and Emma it fully supports Java 7 and Java 8.

Selenium Web Driver: Selenium is a portable software-testing framework for web applications. Selenium provides a record/playback tool for authoring tests without the need to learn a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests can then run against most modern web browsers. Selenium deploys on Windows, Linux, and OS X platforms. It is open-source software, released under the Apache 2.0 license: web developers can download and use it without charge.

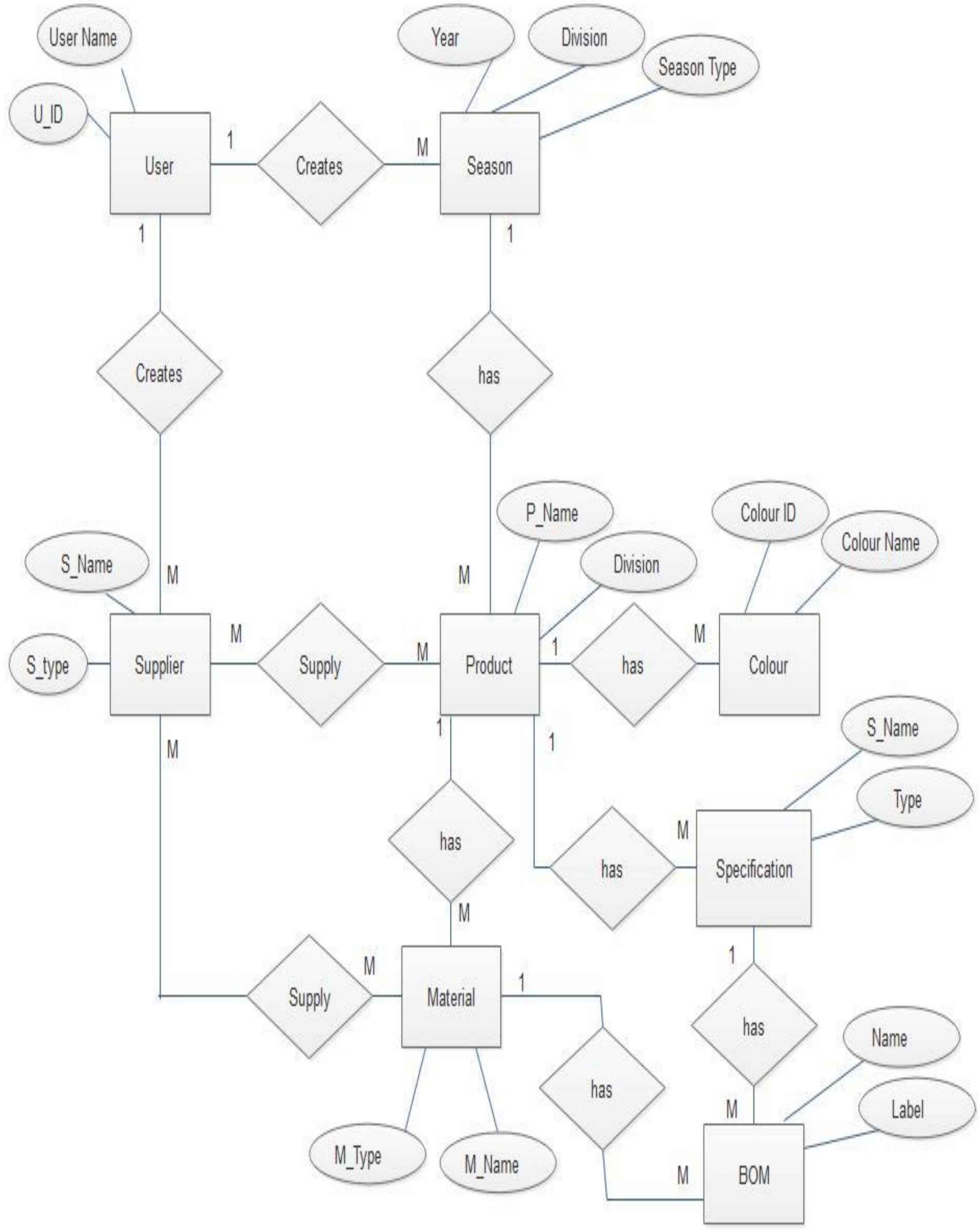
Selenium Web Driver is the successor to Selenium RC. Selenium Web Driver accepts commands (sent in Selenese, or via a Client API) and sends them to a browser. This is implemented through a browser-specific browser driver, which sends commands to a browser, and retrieves results. Most browser drivers actually launch and access a browser application (such as Firefox, Chrome or Internet Explorer); there is also an Html Unit browser driver, which simulates a browser using Html Unit.

Jenkins: Jenkins is an open source automation server written in Java. The project was forked from Hudson after a dispute with Oracle. Jenkins helps to automate the non-human part of the whole software development process, with now common things like continuous integration, but by further empowering teams to implement the technical part of a Continuous Delivery. It is a server-based system running in a servlet container such as Apache Tomcat. It supports SCM tools including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands. The creator of Jenkins is Kohsuke Kawaguchi.[3] Released under the MIT License, Jenkins is free software.

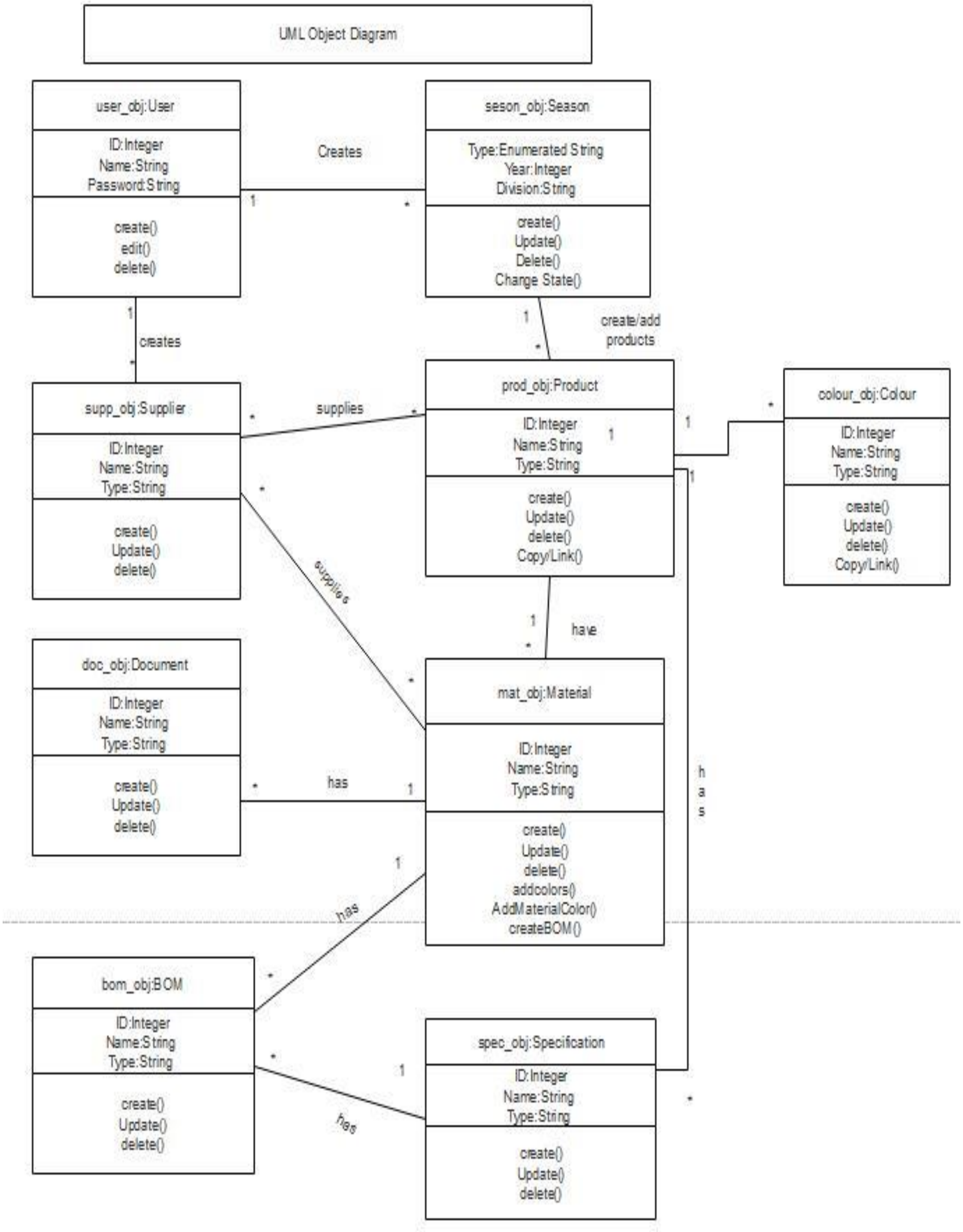
Builds can be triggered by various means, for example by commit in a version control system, by scheduling via a cron-like mechanism and by requesting a specific build URL. It can also be triggered after the other builds in the queue have completed. Jenkins functionality can be extended with plugins.

CHAPTER 3:
ANALYSIS & DESIGN

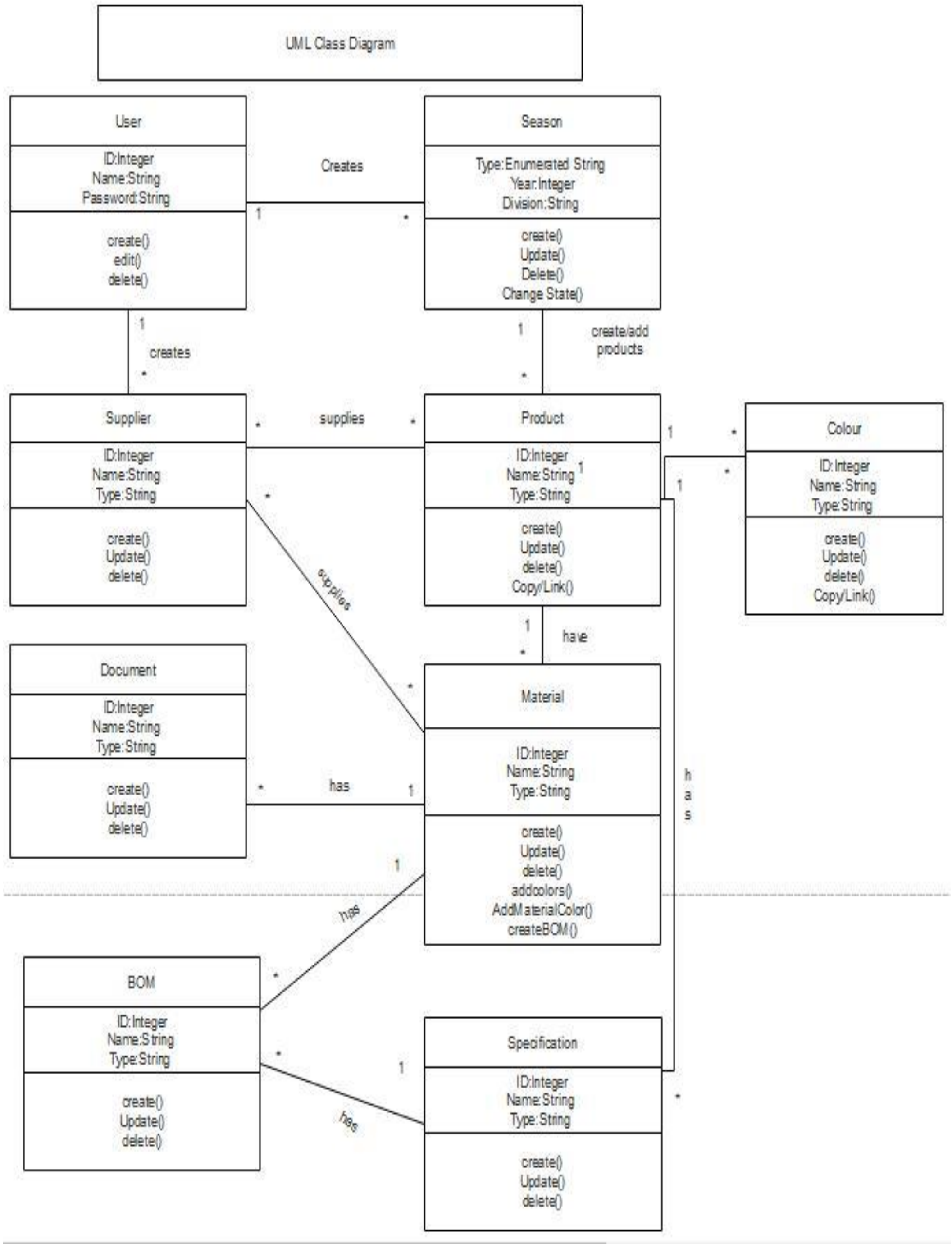
3.1 Entity Relationship Diagram:



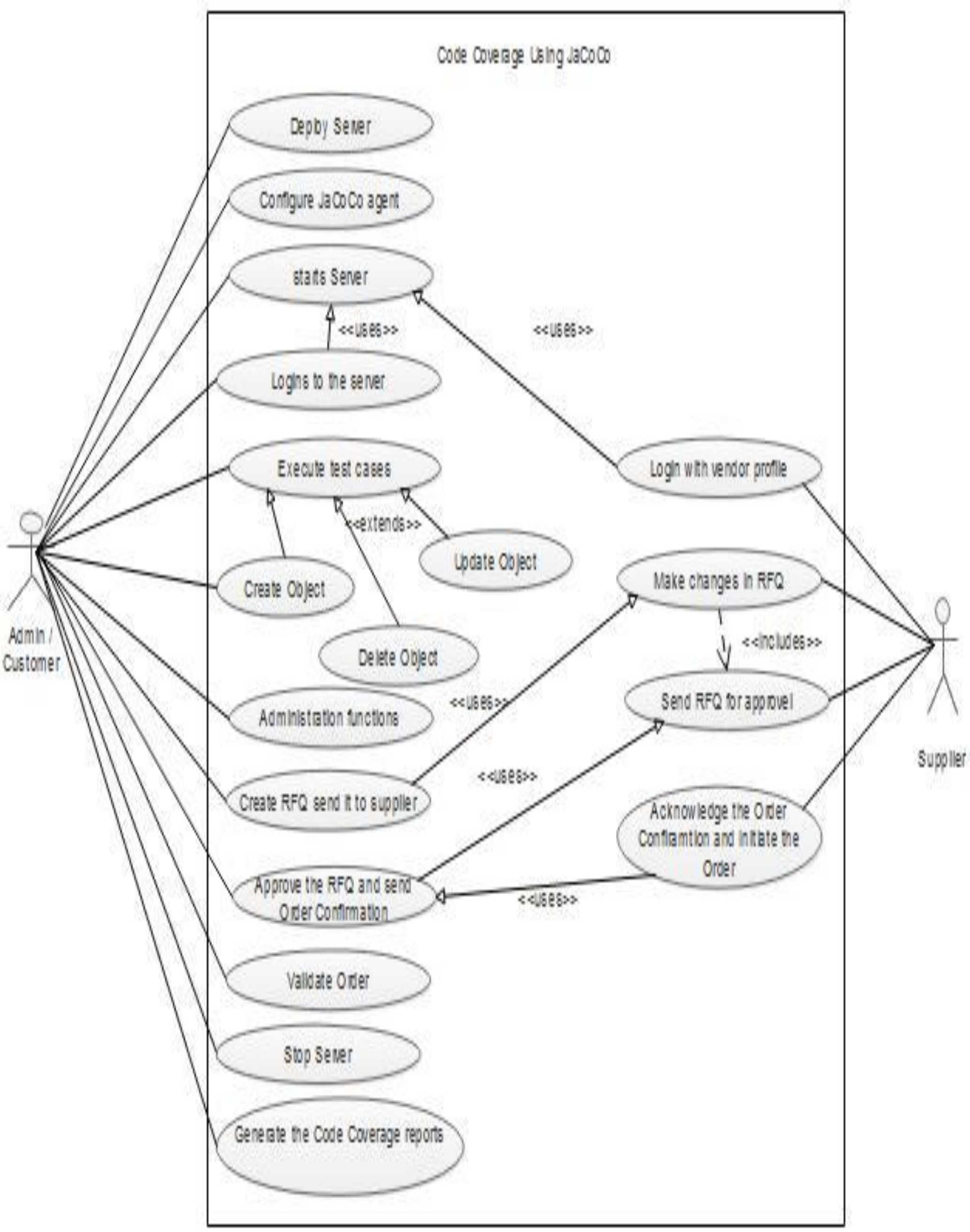
3.2 Object Diagram:



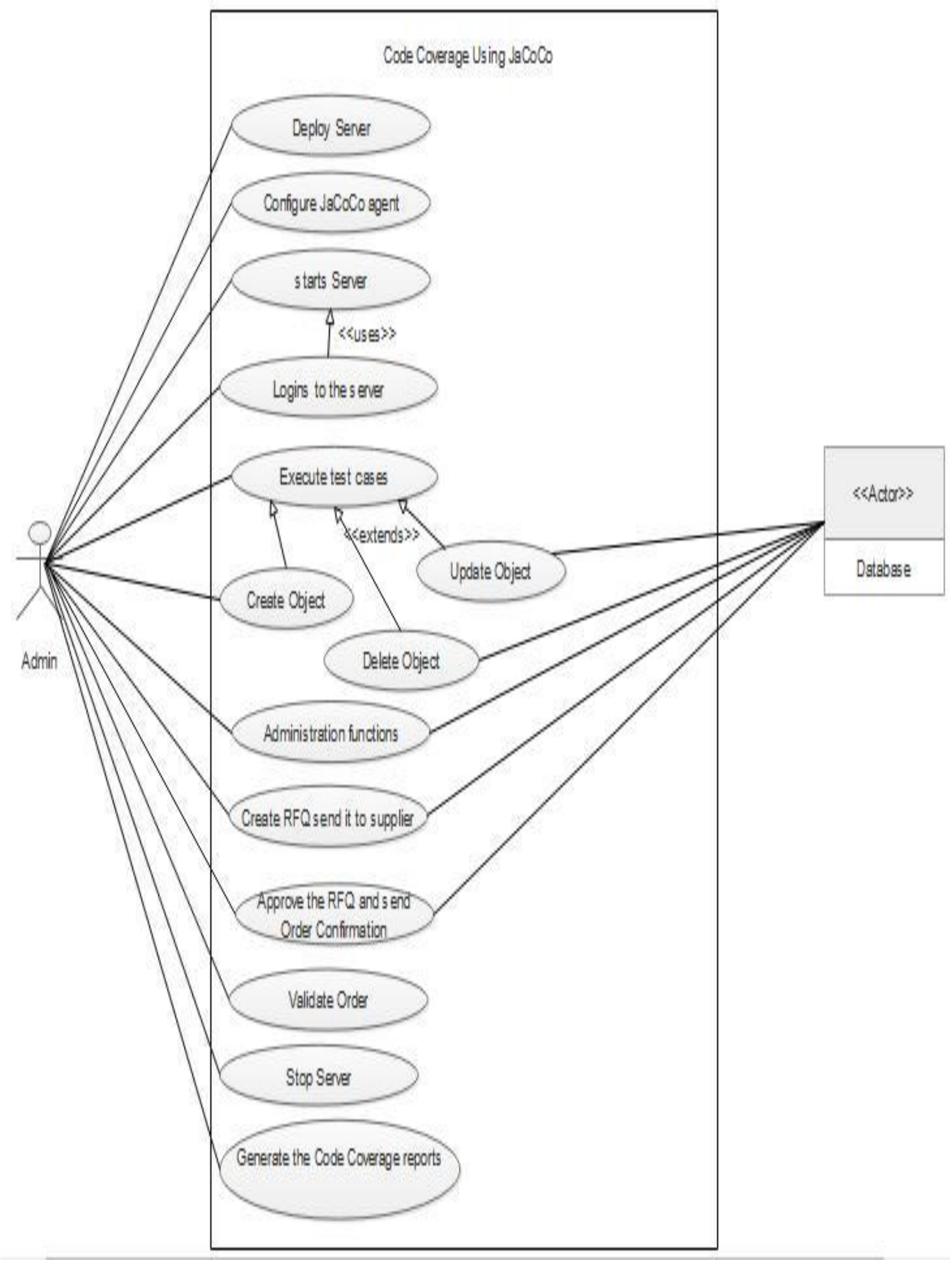
3.3 Class Diagram:



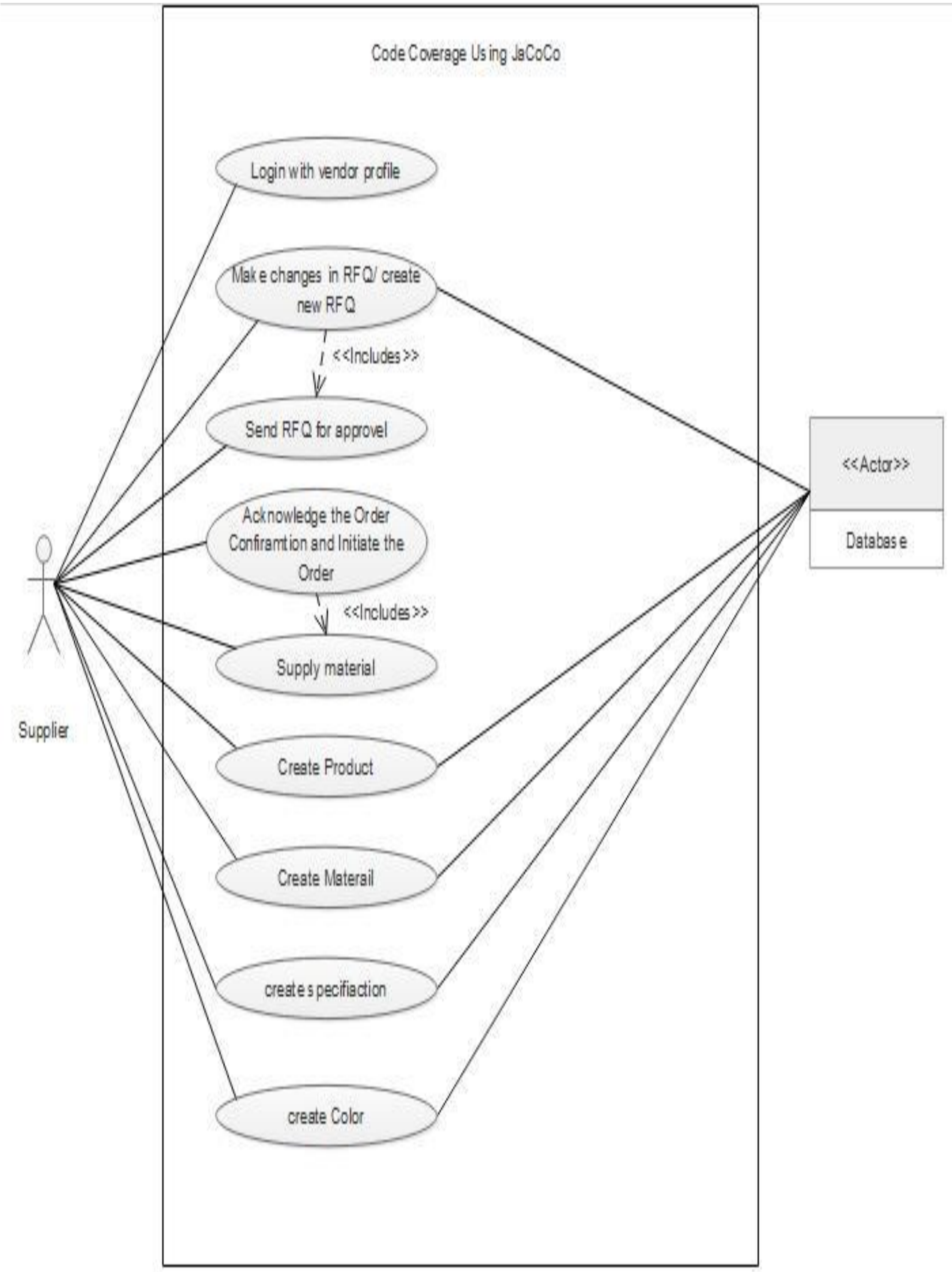
3.4 Use case Diagram of System:



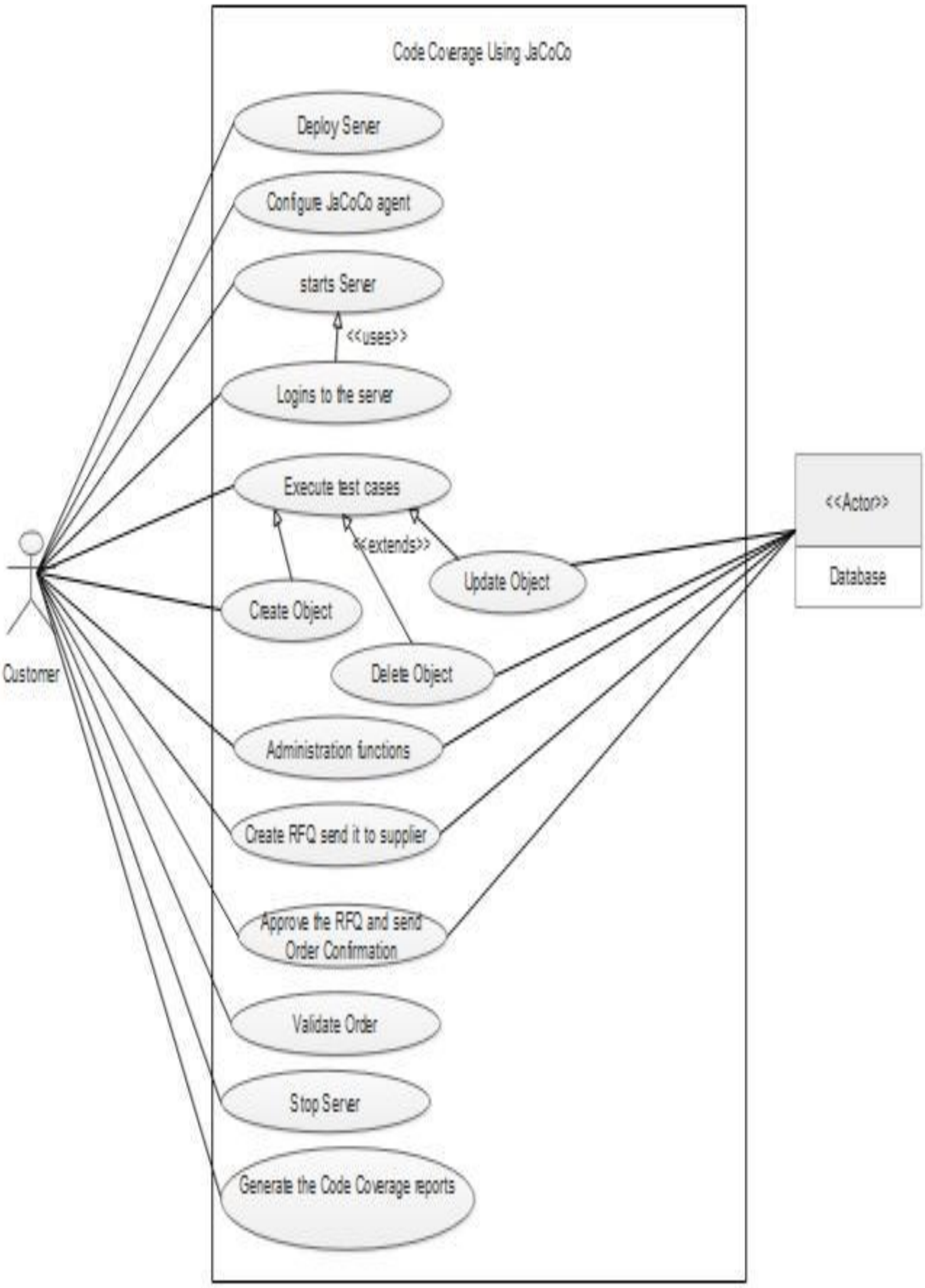
3.4.1 Use case Diagram of Admin User:



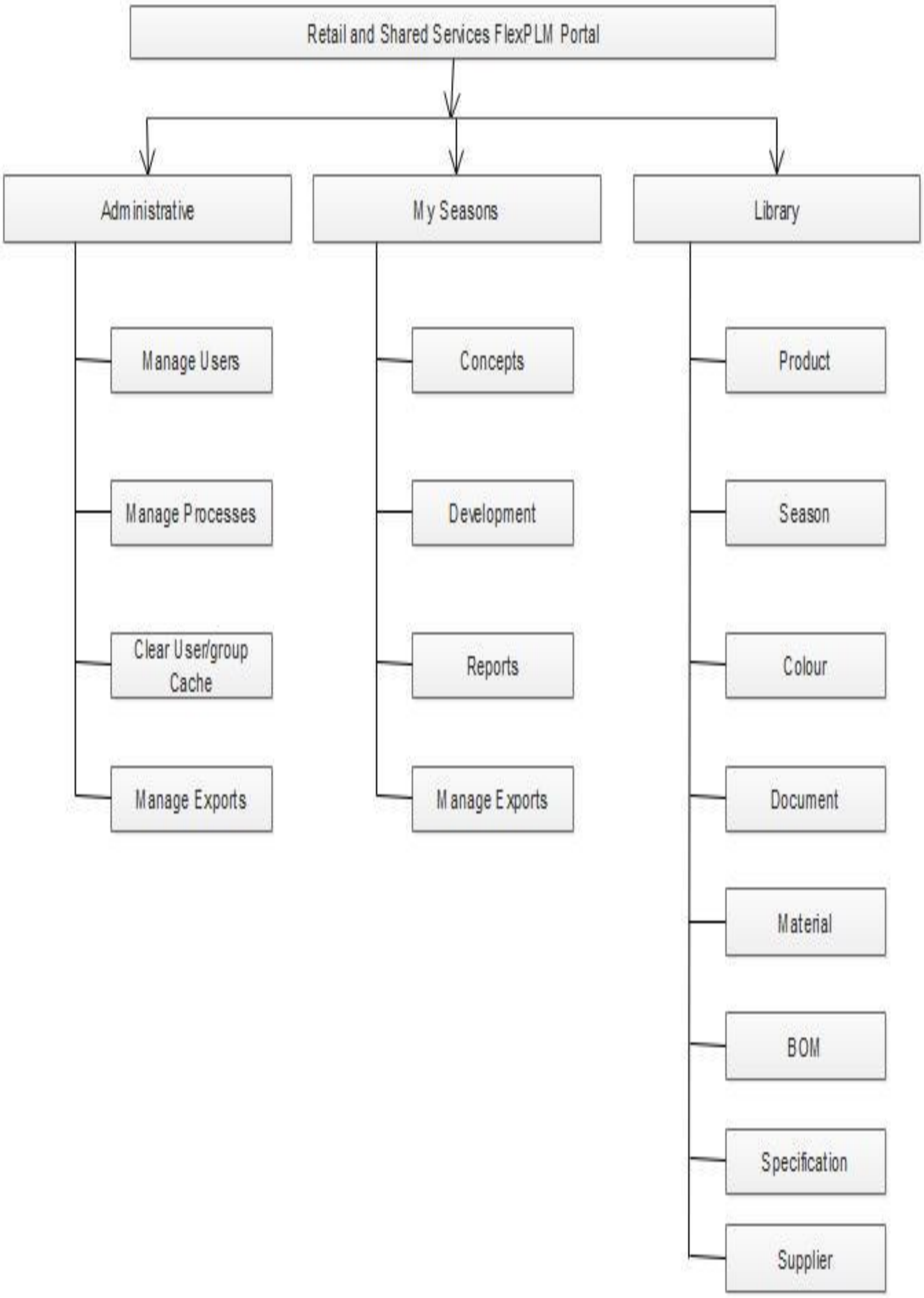
3.4.2 Use case Diagram of Supplier User:



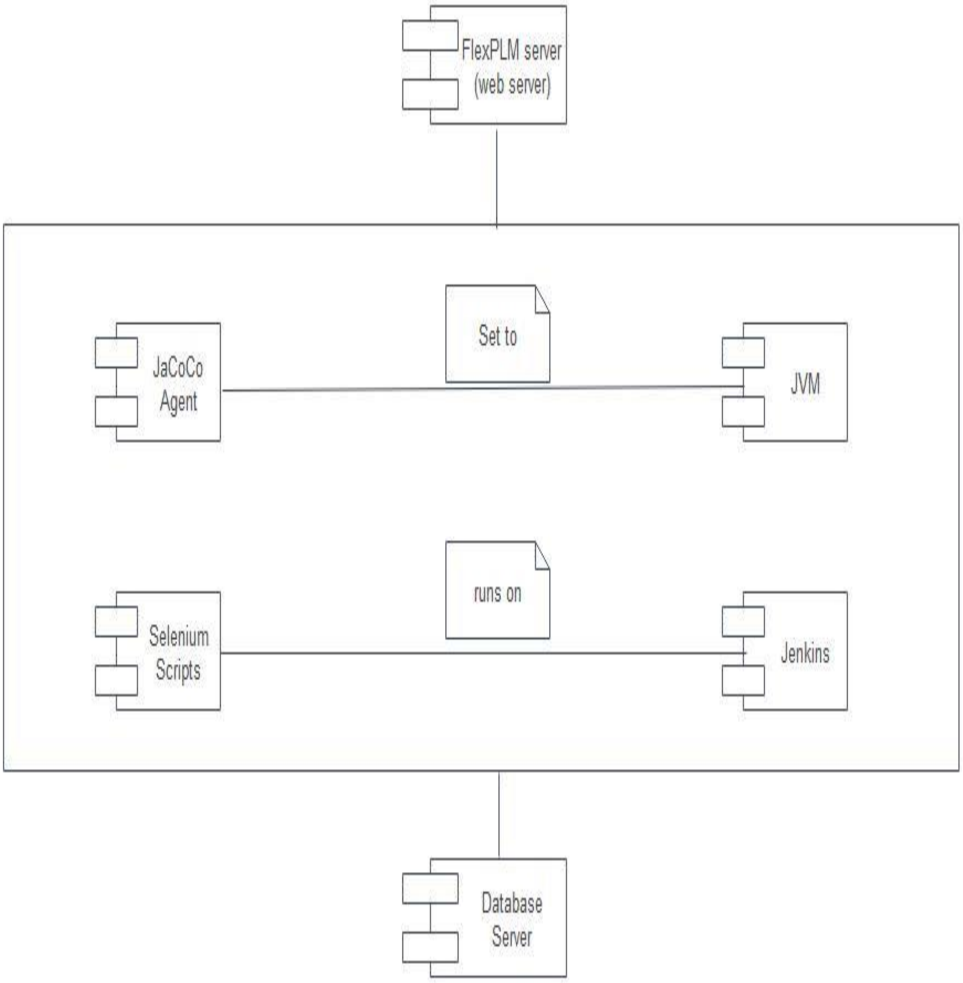
3.4.3 Use case Diagram of Customer:



3.5 Module Hierarchy Diagram:



3.6 Component Diagram



3.7 Testing Strategy and Test Plan

3.7 [1] Verification and Validation

Software Test Specification

1.0 Introduction

This section provides an overview of the entire testing process documentation. This document describes both the test plan and the test procedure.

1.1 Goals and objectives

“Software Testing is a process of finding defects”. A good test case is one that has a capability of finding yet undiscovered errors. Our test objective is to design test processes that systematically uncover different classes of errors and do so with minimum amount of time and effort.

1.2 Statement of scope

Testing scope includes complete Black Box Testing techniques. In this we are testing input & desired output i.e. functional testing.

In case of non-functional testing, User Interface testing, usability & performance testing is performed.

For bug fixes retesting & regression testing will be performed against written test cases & logged defects.

Integration testing will be performed for functional point of view.

2.0 Test Plan

This section describes the overall testing strategy and the project management issues that are required to properly execute effective tests.

2.1 Software to be tested

The software to be tested is identified by name. Exclusions are noted explicitly.

2.2 Testing strategy

FlexPLM System will be tested in functional point of view with the help of Black Box Testing Technique.

2.2.1 Unit testing

Unit is a smallest building block of software system. The basic unit testing will be performed by developers before releasing code of that unit. When it is available for testing, tester test it to check if module is generating desired output as well as it redirects user to enter valid data.

e.g.

% Product is getting created or not.

% Create button should create & display the details of that object.

2.2.2 Integration testing

In integration testing, system will be tested by joining 2 modules for their inter-dependencies. Critical level integration tests are performed by integrators & rest of

the functional integration testing will be performed by testers. In this tester checks whether data generated by one module is properly reflected on another module.
e.g. 1) Product created from season will be listed in the line sheet.

2.2.3 Validation testing

To maintain maximum control over the testing criteria, all data fields will be made specified for testing purposes. The level builder will be tested to ensure proper communication between the interface and the database. Valid input is saved in database and for invalid testing appropriate error message is displayed. e.g.

3. Cost field should display error message if user enters negative value.
4. Application prompt user if any mandatory field is pending to update before record is saved.

2.2.4 High-order testing

After unit, validation & integration testing, all the ready modules are integrated & tested for their correctness, completeness & performance. This is also called as System Testing. At the time of system testing, critical test cases of each level are re-executed & ensured that system is generating output as per given requirement. e.g.

- When all details are entered via different modules like product, supplier, colourway it generates a consolidated report in non-editable table format as per given input.
- It is also tested when all modules are integrated system performance is well manageable.

2.3 Testing tools and environment

Testing tools are used as Integrity to prepare test cases, its execution, defect reporting etc. Integrity is used as a test management tool. Application & server is deployed on server with the set configuration & then it is validated on Mozilla Firefox, Google Chrome, Internet Explorer browser by considering known issues. Selenium

WebDriver is used to Automate the Test Cases. JaCoCo tool is used to find the Code Coverage report of the System.

2.4 Test schedule

Test schedule is prepared by considering time required for development of each module. Accordingly, tester will execute the test cases for functional & non-functional aspects. For testing of all module 1 person & approximately 20 man days (i.e. 20 * 8 = 160 Hrs.) were allocated in which tested has to execute test cases, report defects, retest bug fixes, regress all important test cases

- complete testing for the system as a whole. Also tester test system for User Acceptance Testing – Alpha Testing against finalized user requirements. This testing will ensure that system is working as per user requirements and it is ready to go live.

3.0 Test Procedure

This section describes as detailed test procedure including test tactics and test cases for the software.

3.1 Software to be tested Functional Requirement	**Functionality Details**
1) Login	• Admin • Other Users • Project Manager • Accounts Manager
2) Dashboard	• All the data of proposal & other details are displayed on dashboard.
2) Season	• Season should be created with the given data.
3) Product	• Product should get created and get added to the season.

3.2 Testing procedure

- Collection & analysis of requirements
- Identification of test scenarios
- Generating module-wise test cases
- Creating test cases to validate each field (low-level) with valid & invalid input.
- Written test cases are executed & pass or fail result is mentioned.

6. For failed test case defect is reported in defect reporting template.

7. Verify if the reported bugs are fixed & retested to ensure bug fixes & regressed to check affection to other modules.

8. When bug is fixed that test case status is made as "Retested – As expected" or opened if bug is not fixed properly.

3.2.1 Unit test cases

The procedure for unit testing is described for each software component (that will be unit tested) is presented. This section is repeated for all components.

3.2.1.2Stubs and/or drivers for component

Not applicable

3.2.1.3 Test cases

Component: Login

Validated for all text fields with valid/invalid data according to its length for accepting data Forgot Password link & its functionality.

Functionality of buttons.

Component: Manage Library Objects Like Season, Product, Colour, Specification, Document

Validated for all text fields with valid/invalid data according to its length & data type for accepting data.

Dropdown combo’s data selection

Validation of different functional button

View displayed on pages according to set values

Functionality of breadcrumbs

Warning & confirmation messages.

UI of each webpage.

Consolidated Report generation after each details addition.

3.2.2 Integration testing:

Modules can work individually but they may not work together. Integration testing is done to ensure interface integrity. The transfer of data between modules is tested. Integration testing is done to ensure for correct output is generated when two or more inter-dependent modules are combined together to form system or sub-system. In Project Management System it is ensured that data generated by one module is displayed & can be used successfully for input which will be used to generate consolidated report on Manage Proposal page. Integration testing is performed when delete operation is used to delete proposal then related all data should be deleted from all modules.

3.2.2.1 Testing procedure for integration:

Verify whether names generated in Manage Master Forms are displayed on Manage Proposal > Add New Proposal page while entering new record.

Proposal details & Purchase Order details are displayed while entering project details.

When project is submitted corresponding invoice is generated by system.

All the related details of proposal, purchase order, project details 7 invoice details are displayed on Manage Proposal > View page in non-editable table format.

When proposal is deleted, related all POs, Project details & Invoice details will be deleted.

When project is deleted, all project data & invoice details generated against that project will also be deleted.

In above all scenarios inter dependencies are tested when all modules are combined.

3.2.2.2 Test cases and their purpose

Test cases are written for above scenarios just to ensure that system will generate desired output when combined all module together.

3.2.2.4 Expected results

Expected result for integration testing ensures that modules are communicating in desired fashion to generate high level output.

3.2.3 Validation testing

Validation is the process of checking “are we doing the right job?” It is done with actual execution of the system. Here, we are validating Project Management System after integrating all modules. This type of testing is performed to check if system is generating desired output as per customer’s need when valid input is entered. Also for invalid input, user is asked to enter valid data by warning or error message.

3.2.3.1 Testing procedure for validation

In validation testing, considering valid & invalid input & its desired output, test scenarios are prepared. For this Black Box testing techniques like Equivalence Class Partitioning, Boundary Value Analysis, and State Transition method are used & output is validated for its desired format. Validation testing can be done in each level of testing like unit, integration & system testing.

3.2.3.2 Expected results

1) Application should be able to generate desired output in well formatted manner as per customer's requirement.

2) Application should prompt user for invalid input data entry. Warning/Error messages should be designed in easily understandable format.

3.2.3.3 Pass/fail criterion for all validation tests

For invalid input application should give proper error message.

Valid input data should be saved; confirmation message should be displayed & that data is also displayed in proper format. All the records should be displayed in non-editable table format in respective sections. This can be viewed on module level also.

3.2.4 High-order testing (System Testing)

1) Functionality of buttons
2) Warning & Error messages on invalid input entries, blank mandatory fields.

3) Reports generated – for each module differently & consolidated.

3.2.4.1 Security testing

This tools are used to evaluate the security characteristics of software. For to protect data confidentiality, integrity, authentication, authorization, availability and non-repudiation.

3.2.4.2 Performance testing

Performance testing is performed, to determine how fast some aspect of a system performs under a particular workload.

As per the given requirement it include following things:

What is the time requirement for any/all backend batch processes?

For user interface how many concurrent users are expected for each

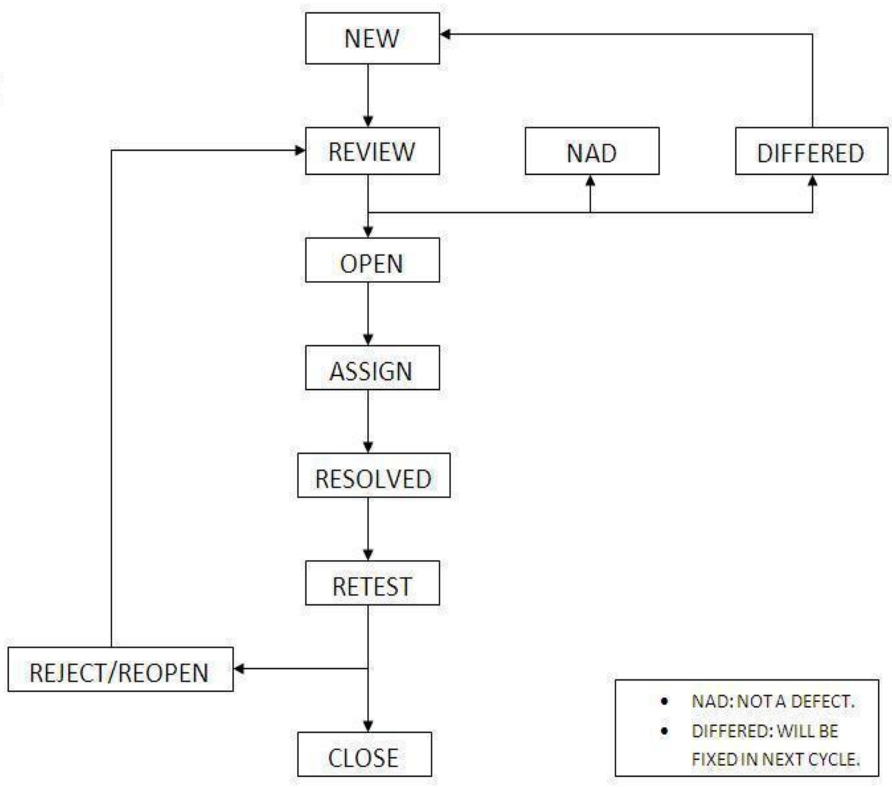
How much time it takes for performing any operation.

Performance testing is carried out using automation tool.

3.7[2] Defect Lifecycle for project

Defect life cycle is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring it’s not reproduced.

The bug has different states in the Life Cycle. The Life cycle of the bug can be shown diagrammatically as follows:



Bug or defect life cycle includes following steps or status:

1. New – When the bug is identified and logged in bug tracking tool for the first time, its state will be "NEW".

2. Open / Closed – After a QA has posted a bug, the QA lead validates the bug. If bug is valid then he changes the state as "OPEN" and if the bug is invalid, then the lead changes its state to "CLOSED".

3. Assign – After QA lead changes the state as "OPEN", he assigns the bug to corresponding developer or developer team lead and the bug status is changed to "ASSIGN".

4. Rejected – If the developer feels that the bug is not valid or it has some technical limitations and cannot be fixed, he rejects the bug. He changes the state of bug to "REJECTED".

5. Duplicate – If the bug is logged is repeated twice or the two bugs reported has alike results and steps to reproduce, then one bug status is changed to "DUPLICATE".

6. Deferred – If the development team lead decides to fix the bug in next release due to lack of time or the priority of the bug is low then he changes the state to "DEFERRED", which is later changes to "ASSIGN" when the bug is taken in consideration to be fixed.

7. In Test – Once the developer fixes the bug, he assigns the bug to the testing team for next round of testing. Before that he changes the state of bug to "IN TEST". It specifies that the bug has been fixed and is released to testing team.

8. Verified – Once the bug is fixed and the status is changed to "IN TEST", the tester tests the bug. If the bug is not reproducible in the software, he changes the status to "VERIFIED".

9. Reopened – Once the bug is fixed and the status is changed to "IN TEST", the tester tests the bug. If the bug is reproducible in the software, he changes the status to "REOPENED”.

10. Closed – After the bug status is changes to "REJECTED" or "DUPLICATE" or "VERIFIED" the QA Lead verifies the comments added by the development or Testing team. When he is satisfied with the comments he changes the state to "CLOSED".

In Project Management System –

Defect Life Cycle used for Project Management System is as follows –

NEW – Tester reports bugs in the shared bug sheet with severity. Test Lead verifies that defect & change its status to OPEN if it is a bug or NOT A BUG if it is not a bug or

DUPLICATE if it is previously reported or DEFFERRED in case of not considered in this build.

Developer has access of bug sheet, so developer starts to resolve reported open bugs as per assigned severities & priorities.

When developer fixes any issue, the status was changed to RESOLVED & the corresponding bug row made coloured in light grey colour, so that tester will come to know which bugs are resolved & need retesting.

Tester will retest & regress for the resolved issue & change status as:

CLOSED: – If issue is successfully resolved with no impact on other functionality.

REOPENED: – If issue is not resolved properly or changes caused for bugs in other modules.

When status becomes CLOSED – Defect life cycle stops there & the defect is mad coloured as Green.

When status is updated as REOPENED then step 2 onwards followed & the defect is made coloured as Orange. At the end, all bugs will be retested to verify there is no major bugs.

3.8 Module Specifications:

1. Product: Product can be of any type of product like apparel, footwear etc. Product is having submodules like Size, Measurements etc.

2. Colour: Colour or colour is a property of light as seen by people. Colour can be created as combination of multiple colours and solid colours etc.

3. Material: The matter from which something can be made. Material can include but is not limited to raw and processed material, components, parts small tools and accessories that may be consumed directly or indirectly. Material can have the sub modules like sources, Colour, Images, Documents etc.

4. Document: A piece of written, printed, or electronic matter that provides information or evidence or that serves as an official record. Document is nothing but the cover pages, Image pages etc.

5. Specification: A detailed description of the design and materials used to make something. Exact statement of the particular needs to be satisfied, or essential characteristics that a customer requires (in a good, material, method, process, service, system, or work) and which a vendor must deliver. Specifications are written usually in a manner that enables both parties (and/or an independent certifier) to measure the degree of conformance. They are, however, not the same as control limits (which allow fluctuations within a range), and conformance to them does not necessarily mean quality (which is a predictable degree of dependability and uniformity). Specifications are divided generally into two main categories: (1) Performance specifications: conform to known customer requirements such as keeping a room's temperature within a specified range.

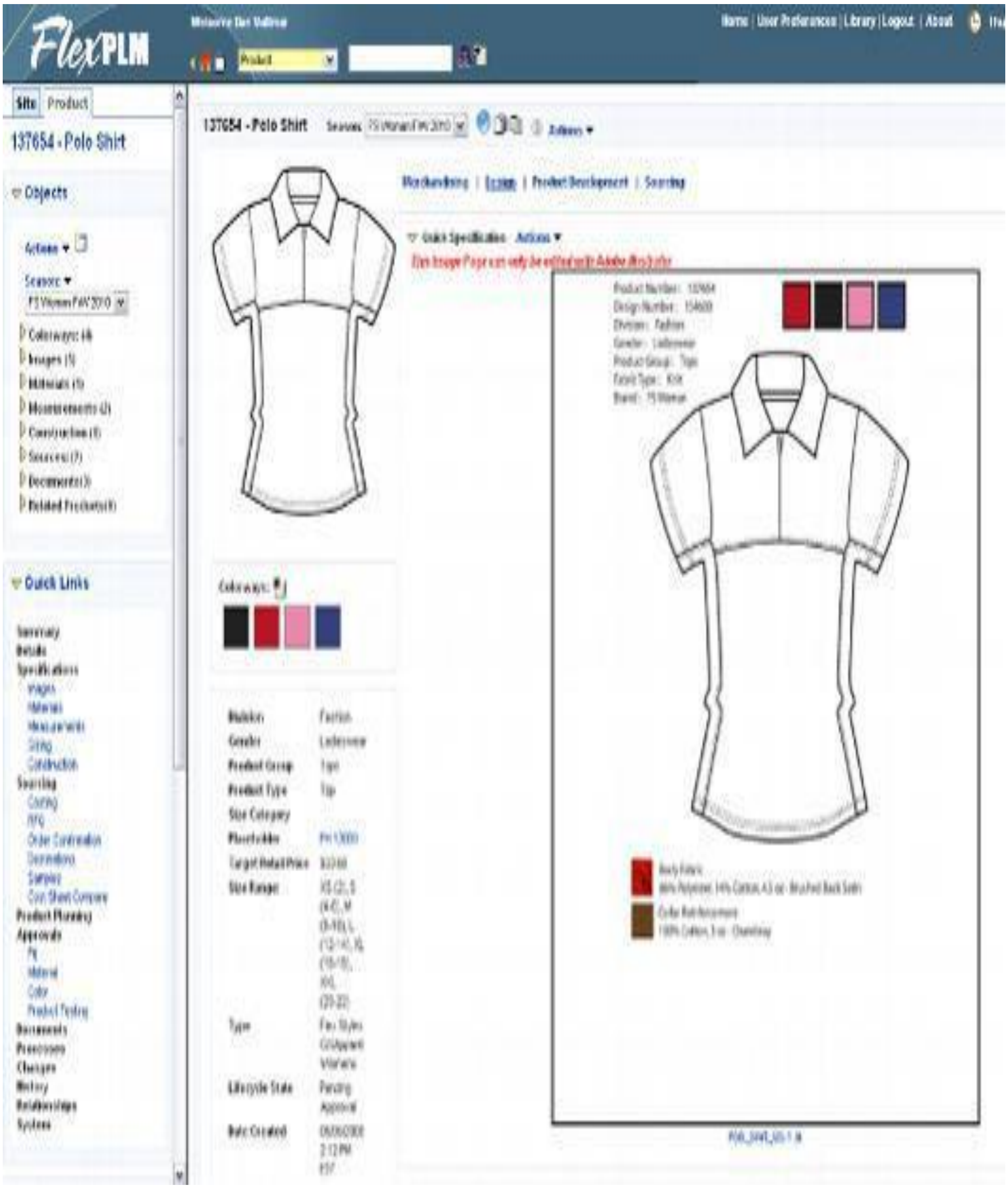
6. Bill of Material: The bill of materials typically includes part names, part numbers, part revisions and the quantities required to build an assembly.

7. Supplier: A person or organization that provides something needed such as a product or service. This can be of many types like vendor supplier, distributor retailer etc.

All the above objects can be created, updated and deleted.

3.9 User Interface Design

3.9 .1. FlexPLM Product Details Page



Division	Fashion
Gender	Ladieswear
Product Group	Tops
Product Type	Top
Size Category	
Placeholder	PH 13000
Target Retail Price	$32.00
Size Range	XS (2), S (4-6), M (8-10), L (12-14), XL (16-18), XXL (20-22)
Type	Flex Styles CC/Apparel Womens
Lifecycle State	Pending Approval
Date Created	06/06/2008 2:12 PM PST

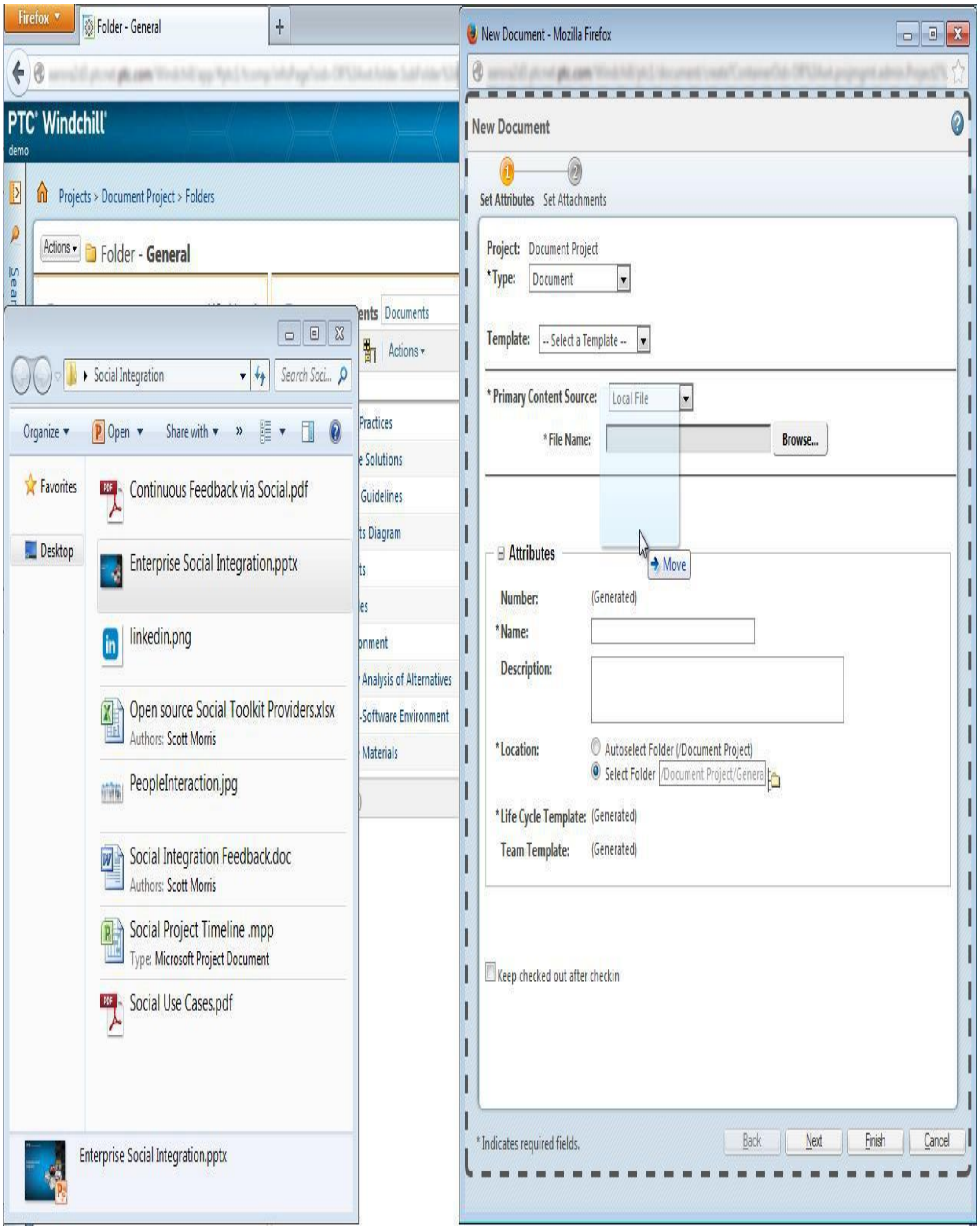
This Image Page can only be edited with Adobe Illustrator

Product Number: 137654
Design Number: 154600
Division: Fashion
Gender: Ladieswear
Product Group: Tops
Fabric Type: Knit
Brand: FS Woman

Body Fabric
86% Polyester, 14% Cotton, 4.5 oz. Brushed Back Satin

Collar Rib Accent
100% Cotton, 5 oz. Chambray

3.9 .2. FlexPLM Create New Document Page



Continuous Feedback via Social.pdf

Enterprise Social Integration.pptx

linkedin.png

Open source Social Toolkit Providers.xlsx
Authors: Scott Morris

PeopleInteraction.jpg

Social Integration Feedback.doc
Authors: Scott Morris

Social Project Timeline .mpp
Type: Microsoft Project Document

Social Use Cases.pdf

Enterprise Social Integration.pptx

New Document

Project: Document Project
*Type: Document
Template: -- Select a Template --

*Primary Content Source: Local File
*File Name:

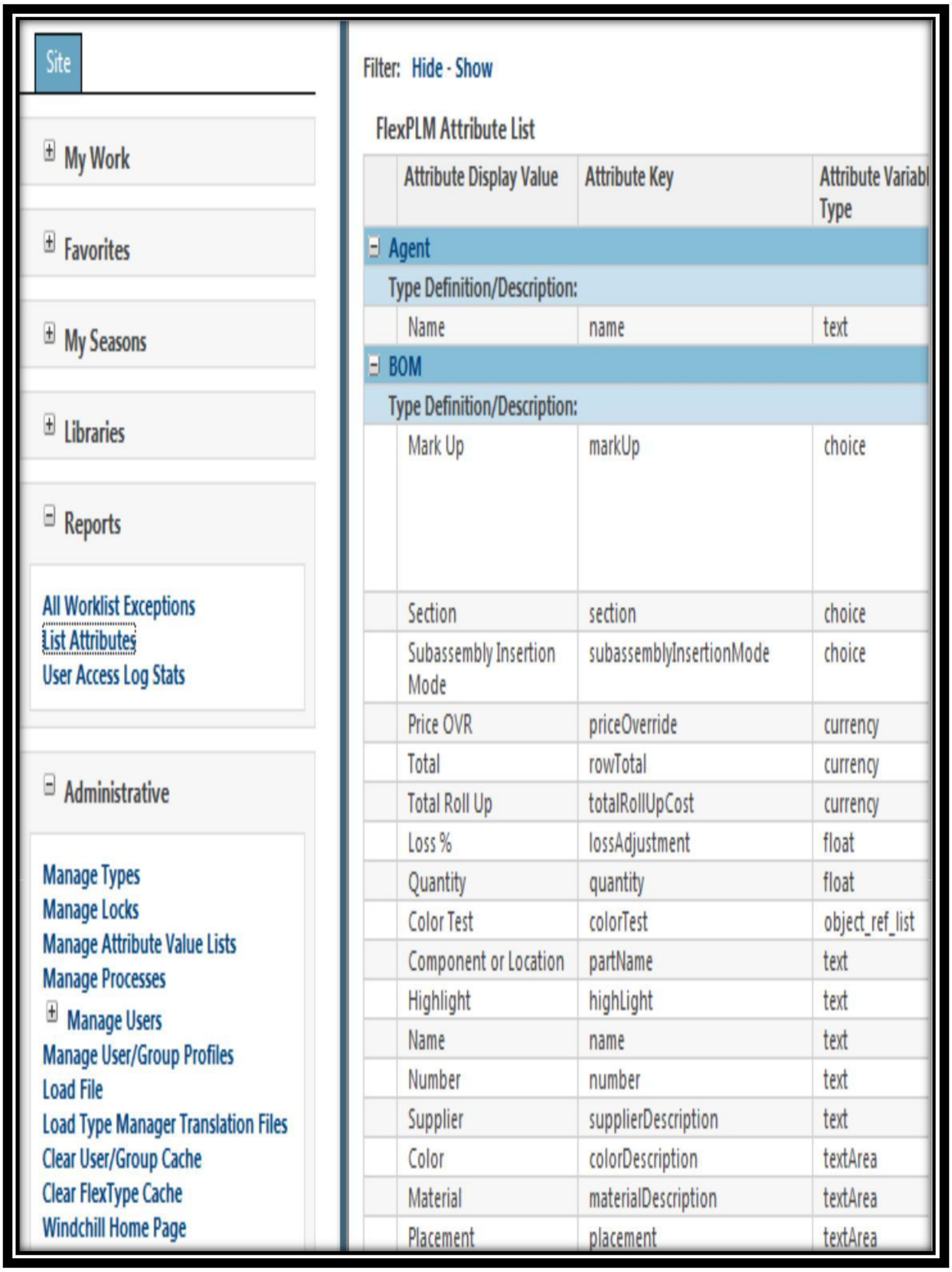
Attributes

Number: (Generated)
*Name:
Description:
*Location: ○ Autoselect Folder (/Document Project)
● Select Folder /Document Project/General
*Life Cycle Template: (Generated)
Team Template: (Generated)

□ Keep checked out after checkin

* Indicates required fields.

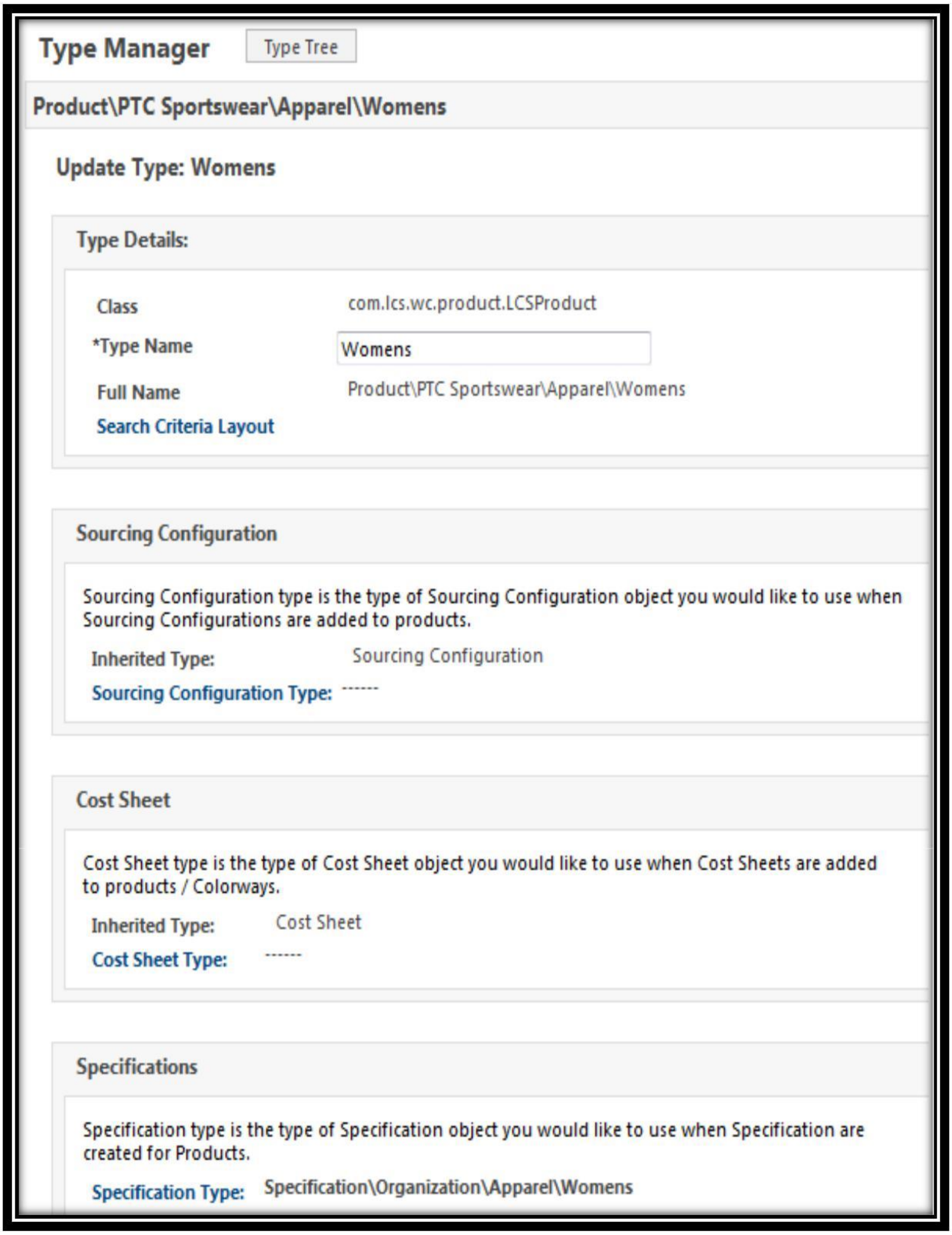
3.9 .3. FlexPLM Attribute List Page



FlexPLM Attribute List

	Attribute Display Value	Attribute Key	Attribute Variabl Type
Agent			
Type Definition/Description:			
	Name	name	text
BOM			
Type Definition/Description:			
	Mark Up	markUp	choice
	Section	section	choice
	Subassembly Insertion Mode	subassemblyInsertionMode	choice
	Price OVR	priceOverride	currency
	Total	rowTotal	currency
	Total Roll Up	totalRollUpCost	currency
	Loss %	lossAdjustment	float
	Quantity	quantity	float
	Color Test	colorTest	object_ref_list
	Component or Location	partName	text
	Highlight	highLight	text
	Name	name	text
	Number	number	text
	Supplier	supplierDescription	text
	Color	colorDescription	textArea
	Material	materialDescription	textArea
	Placement	placement	textArea

3.9.4. FlexPLM Type Manager Details Page



Product\PTC Sportswear\Apparel\Womens

Update Type: Womens

Type Details:

Class	com.lcs.wc.product.LCSProduct
*Type Name	Womens
Full Name	Product\PTC Sportswear\Apparel\Womens
Search Criteria Layout	

Sourcing Configuration

Sourcing Configuration type is the type of Sourcing Configuration object you would like to use when Sourcing Configurations are added to products.

Inherited Type:	Sourcing Configuration
Sourcing Configuration Type:	------

Cost Sheet

Cost Sheet type is the type of Cost Sheet object you would like to use when Cost Sheets are added to products / Colorways.

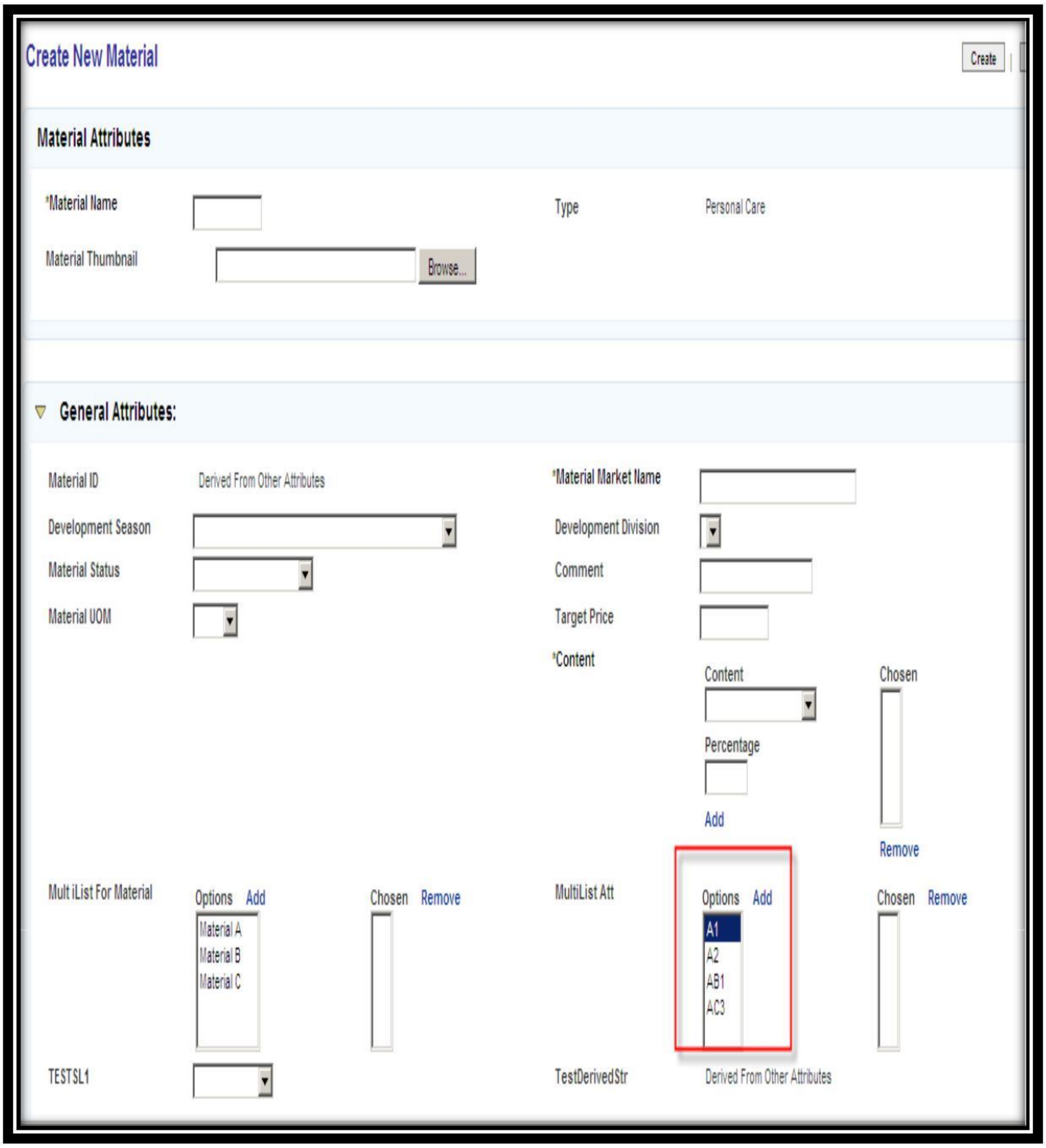
Inherited Type:	Cost Sheet
Cost Sheet Type:	------

Specifications

Specification type is the type of Specification object you would like to use when Specification are created for Products.

Specification Type:	Specification\Organization\Apparel\Womens

3.9 .5. FlexPLM Type Manager Details Page



Create New Material

Create

Material Attributes

*Material Name

Type: Personal Care

Material Thumbnail

Browse...

General Attributes:

Material ID: Derived From Other Attributes

*Material Market Name

Development Season

Development Division

Material Status

Comment

Material UOM

Target Price

*Content

Content

Percentage

Add

Chosen

Remove

Mult iList For Material

Options Add

Material A
Material B
Material C

Chosen Remove

MultiList Att

Options Add

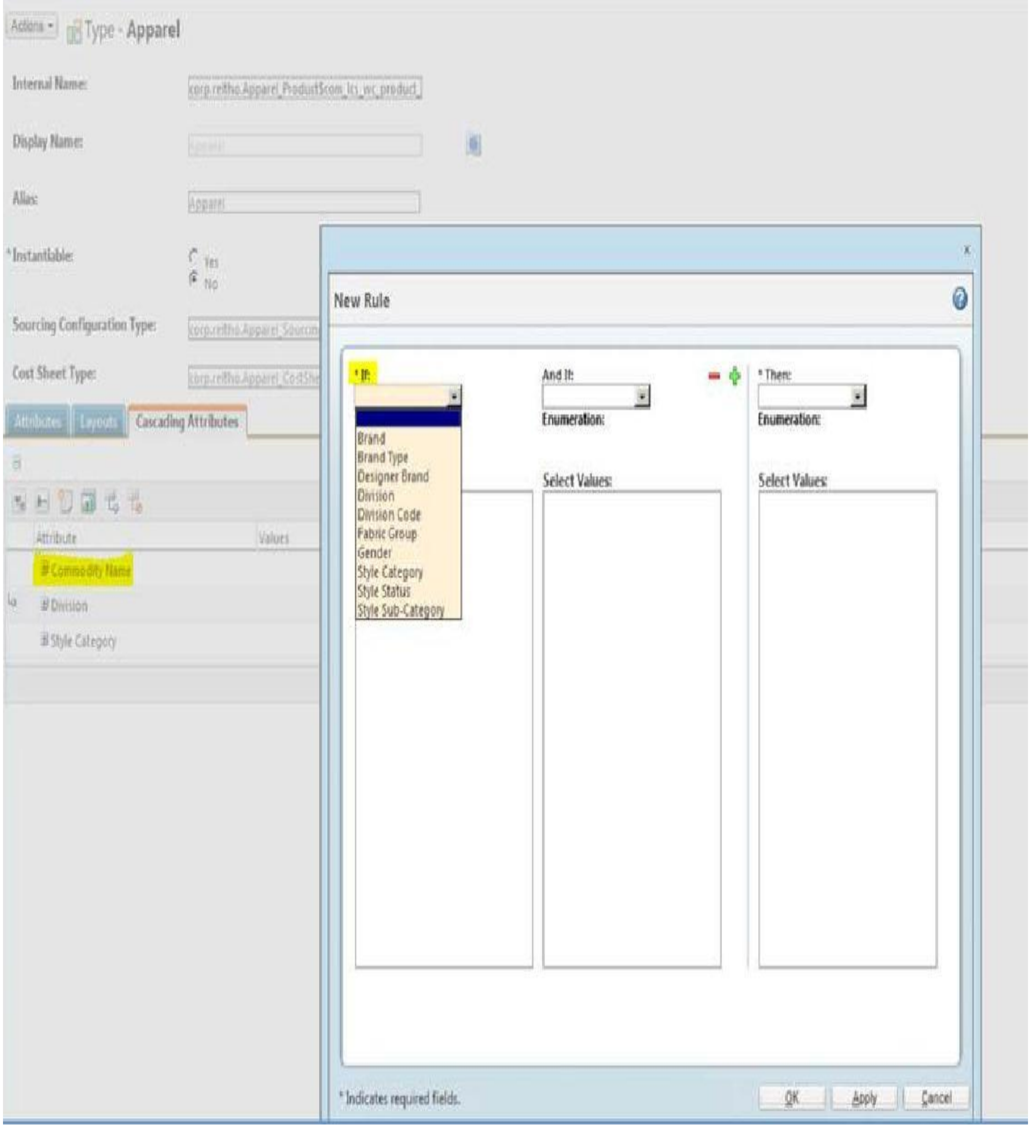
A1
A2
AB1
AC3

Chosen Remove

TESTSL1

TestDerivedStr: Derived From Other Attributes

3.9 .6. FlexPLM Type Manager New Rule Creation Page



Type - Apparel

Internal Name: corp.reitho.Apparel_Produt$com_lcs_wc_product_

Display Name: Apparel

Alias: Apparel

*Instantiable: ○ Yes ● No

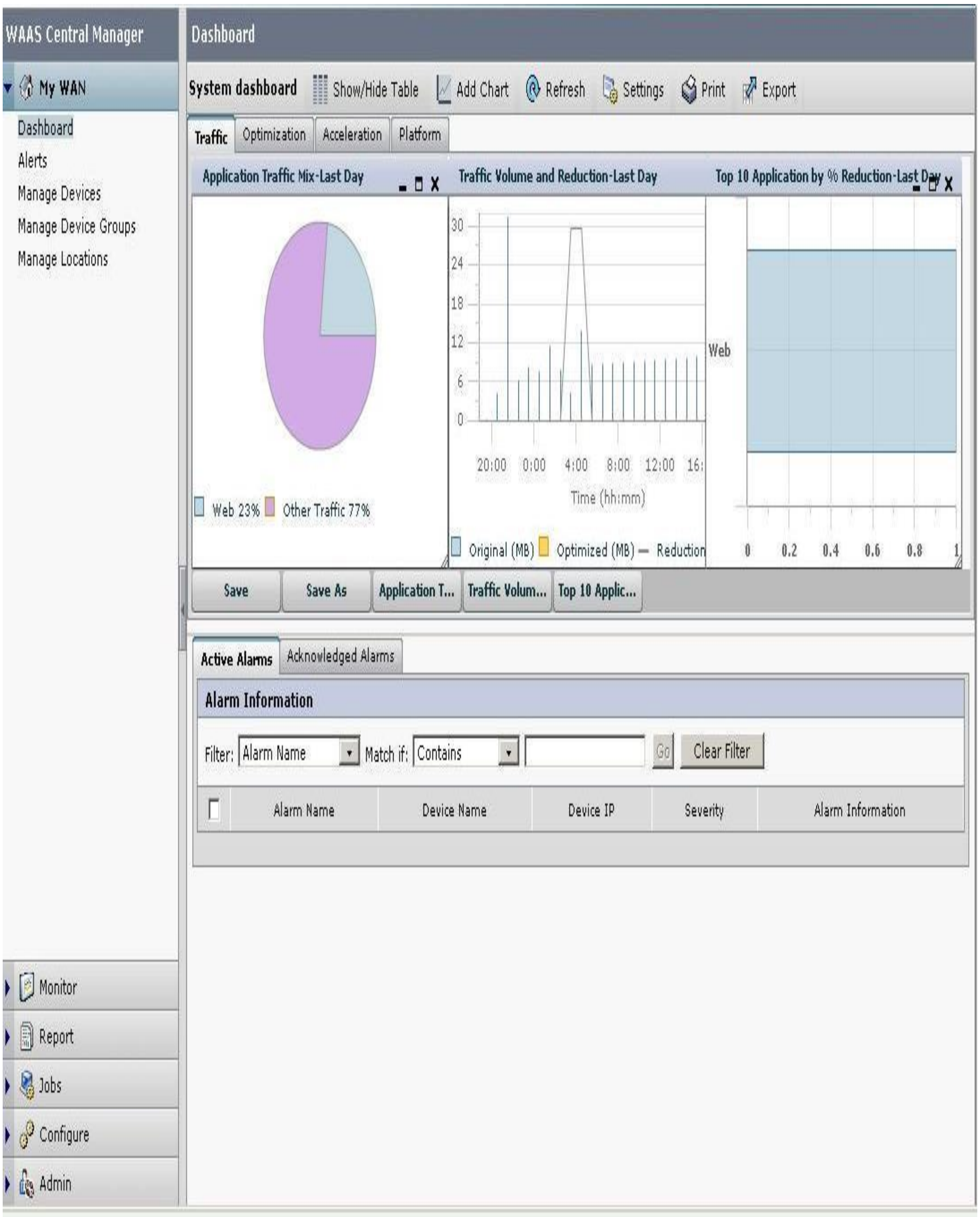
Sourcing Configuration Type: corp.reitho.Apparel_Sourcin

Cost Sheet Type: corp.reitho.Apparel_CostShe

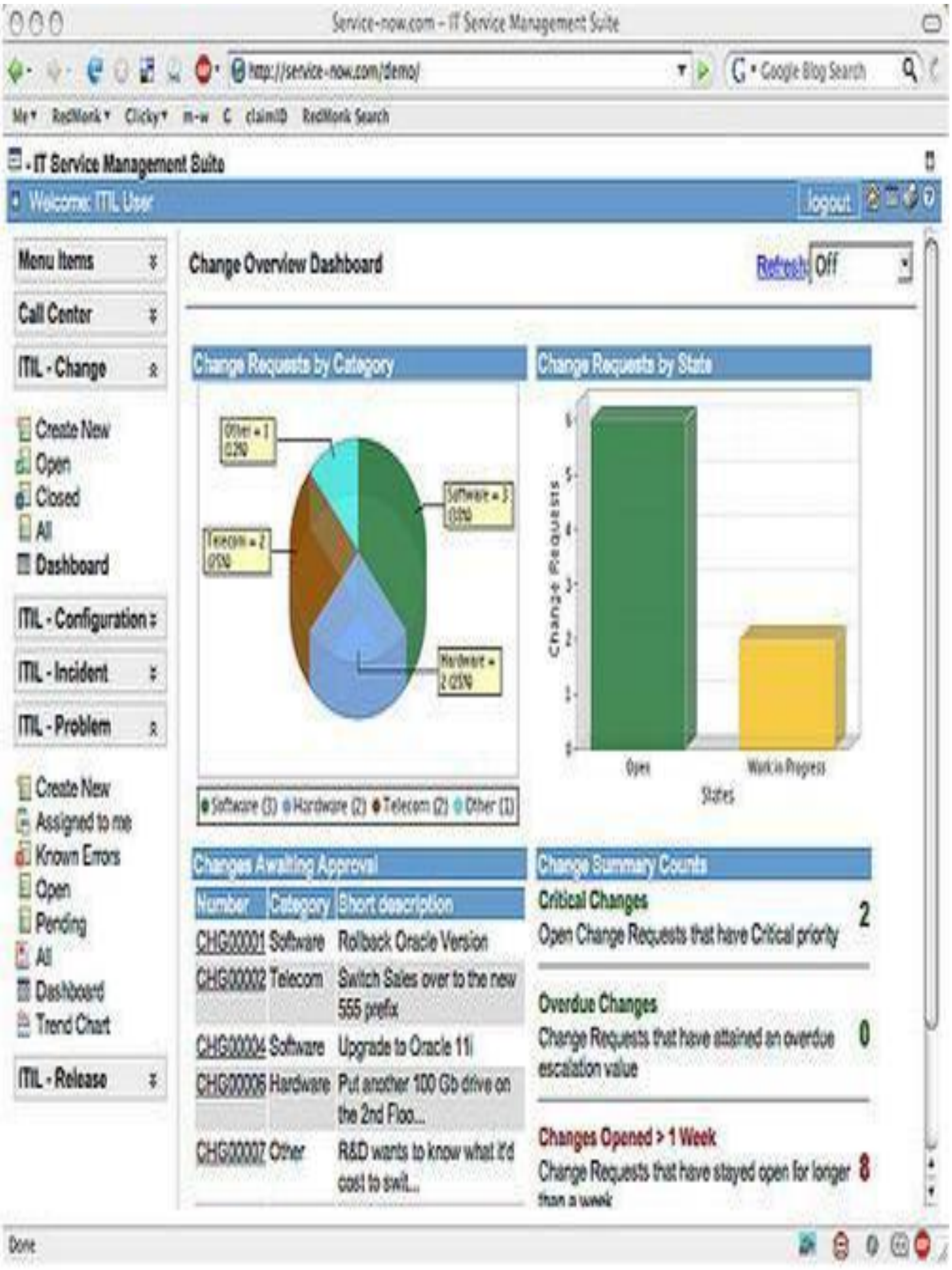
New Rule

*Indicates required fields.

3.9.7. FlexPLM Season Dashboard Page



3.9.8. FlexPLM Change Overview Dashboard Page



Changes Awaiting Approval

Number	Category	Short description
CHG00001	Software	Rollback Oracle Version
CHG00002	Telecom	Switch Sales over to the new 555 prefix
CHG00004	Software	Upgrade to Oracle 11i
CHG00006	Hardware	Put another 100 Gb drive on the 2nd Floo...
CHG00007	Other	R&D wants to know what it'd cost to swit...

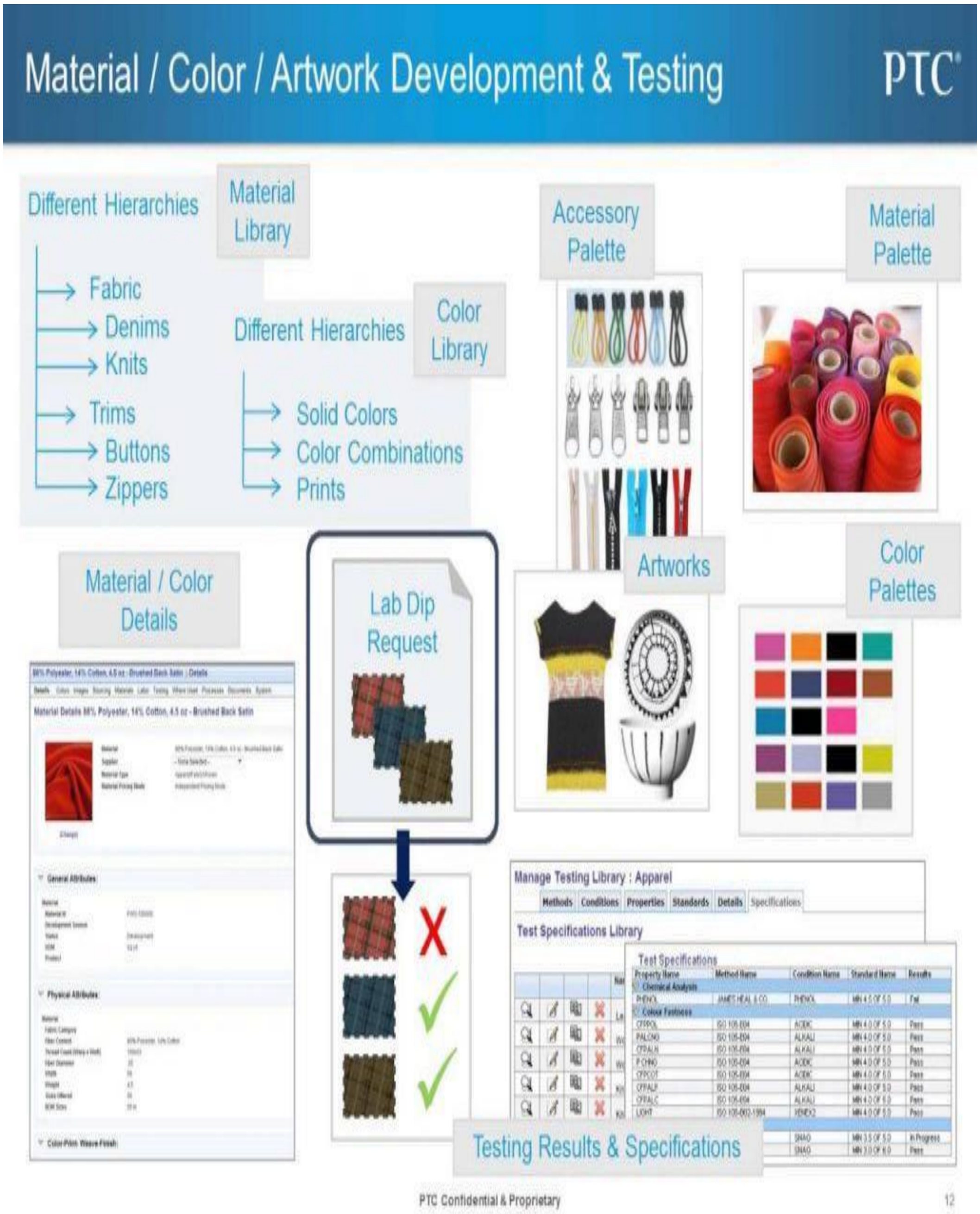
Change Summary Counts

Critical Changes 2
Open Change Requests that have Critical priority

Overdue Changes 0
Change Requests that have attained an overdue escalation value

Changes Opened > 1 Week 8
Change Requests that have stayed open for longer than a week

3.9.9. FlexPLM Material/Color/Artwork Page



Material / Color / Artwork Development & Testing

PTC®

Material Library

Different Hierarchies

- Fabric
  - Denims
  - Knits
- Trims
  - Buttons
  - Zippers

Color Library

Different Hierarchies

- Solid Colors
- Color Combinations
- Prints

Accessory Palette

Material Palette

Artworks

Color Palettes

Material / Color Details

86% Polyester, 14% Cotton, 4.5 oz - Brushed Back Satin : Details

Material Details 86% Polyester, 14% Cotton, 4.5 oz - Brushed Back Satin

General Attributes

Physical Attributes

Color-Print-Weave-Finish

Lab Dip Request

Manage Testing Library : Apparel

Test Specifications Library

Test Specifications

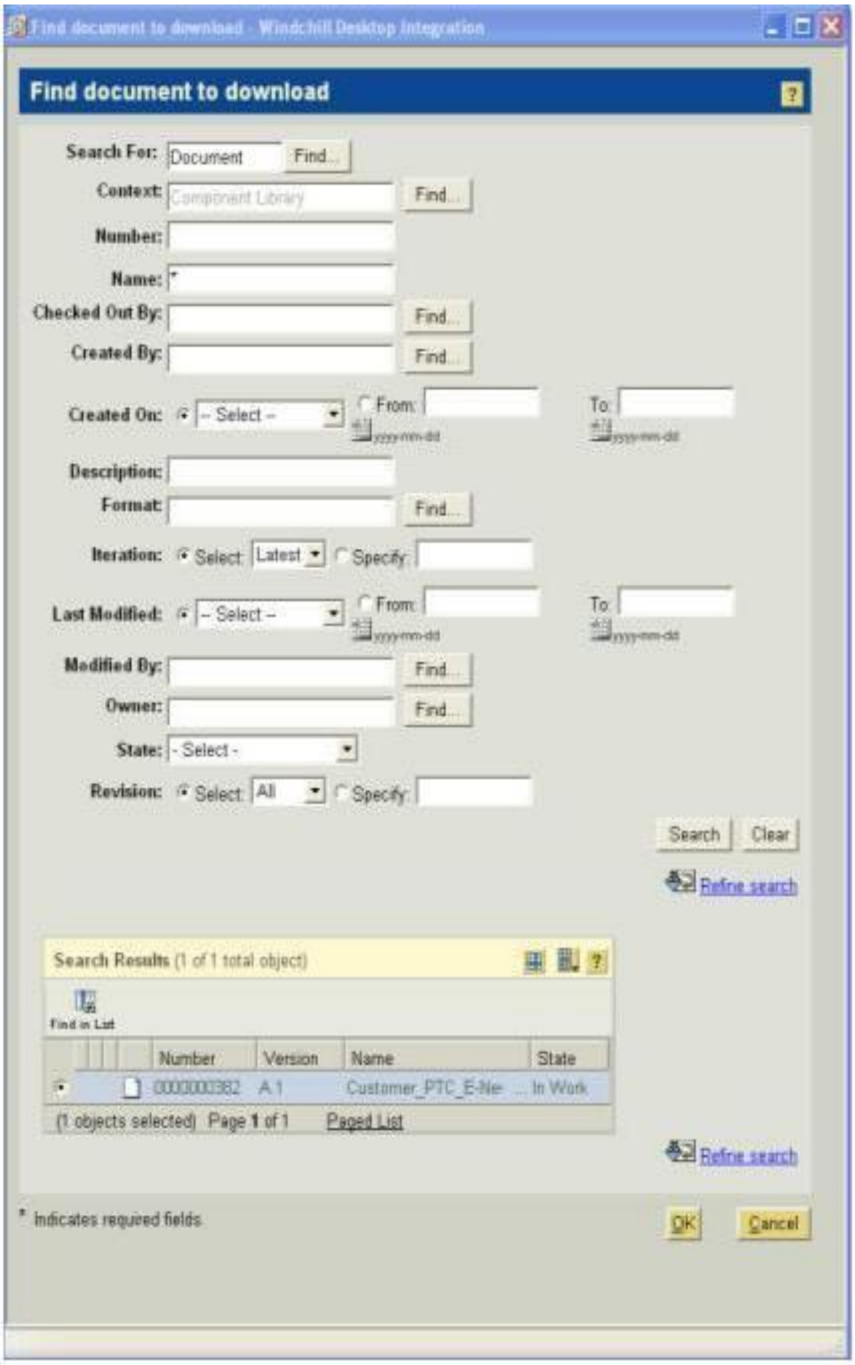
Property Name	Method Name	Condition Name	Standard Name	Results
Chemical Analysis				
PHENOL	JAMES HEAL & CO	PHENOL	MIN 4.5 OF 5.0	Fail
Colour Fastness				
CFPPOL	ISO 105-E04	ACIDIC	MIN 4.0 OF 5.0	Pass
PALCNO	ISO 105-E04	ALKALI	MIN 4.0 OF 5.0	Pass
CFPALN	ISO 105-E04	ALKALI	MIN 4.0 OF 5.0	Pass
P CHNO	ISO 105-E04	ACIDIC	MIN 4.0 OF 5.0	Pass
CFPCOT	ISO 105-E04	ACIDIC	MIN 4.0 OF 5.0	Pass
CFPALP	ISO 105-E04	ALKALI	MIN 4.0 OF 5.0	Pass
CFPALC	ISO 105-E04	ALKALI	MIN 4.0 OF 5.0	Pass
LIGHT	ISO 105-B02-1994	XENEX2	MIN 4.0 OF 5.0	Pass
		SNAG	MIN 3.5 OF 5.0	In Progress
		SNAG	MIN 3.0 OF 6.0	Pass

Testing Results & Specifications

PTC Confidential & Proprietary

12

3.9.10. FlexPLM Find Document Page



Find document to download - Windchill Desktop Integration

Find document to download

Search For: Document

Context: Component Library

Number:

Name: *

Checked Out By:

Created By:

Created On: ● -- Select -- ○ From: yyyy-mm-dd To: yyyy-mm-dd

Description:

Format:

Iteration: ● Select: Latest ○ Specify:

Last Modified: ● -- Select -- ○ From: yyyy-mm-dd To: yyyy-mm-dd

Modified By:

Owner:

State: - Select -

Revision: ● Select: All ○ Specify:

Search Results (1 of 1 total object)

	Number	Version	Name	State
●	0000000382	A.1	Customer_PTC_E-Ne ...	In Work

(1 objects selected) Page 1 of 1 Paged List

* Indicates required fields

3.10 Table Specifications

3.10.1User Object Table:

Sr.no.	Field Name	Data Type	Key Constraint
1	User ID	Varchar2(10)	Primary key
2	User Name	Varchar2(10)	Not Null
3	Email	Varchar2(10)	Not Null
4	Org Number	Number (20)	Not Null
5	Org Name	Varchar2(30)	Not Null

3. 10.2 Product Table:

Sr.no.	Field Name	Data Type	Key Constraint
1	Product ID	Varchar2(10)	Primary key
2	User ID	Varchar2(10)	Foreign key
3	Product Name	Varchar2(20)	Not Null
4	Division	Varchar2(10)	Not Null
5	Date	Date	Not Null

3.10.3 Season Table:

Sr.no.	**Field Name**	**Data Type**	**Key Constraint**
1	Season Name	Varchar2(20)	Primary key
2	Division	Varchar2(10)	Not Null
3	Year	Number (10)	Not Null
4	Division	Varchar2(10)	Not Null
5	Product Type	Varchar2(10)	Not Null
6	Product ID	Varchar2(10)	Foreign keys

3.10.4 Supplier Table:

Sr.no.	**Field Name**	**Data Type**	**Key Constraint**
1	Product ID	Varchar2(10)	Primary key
2	User ID	Varchar2(10)	Foreign key
3	Supplier Name	Varchar2(10)	Not Null
4	Supplier Type	Varchar2(10)	Not Null
5	Date	Date	Not Null

3.10.5 Colour Table:

Sr.no.	Field Name	Data Type	Key Constraint
1	Colour ID	Varchar2(10)	Primary key
2	Colour Type	Varchar2(10)	Not Null
3	Colour Name	Number (10)	Not Null

3.11.6 Material Table:

Sr.no.	Field Name	Data Type	Key Constraint
1	Material ID	Varchar2(10)	Primary key
2	Material Name	Varchar2(10)	Not Null
3	Material Market Name	Varchar2(10)	Not Null
4	Material type	Number (10)	Not Null
5	Product ID	Varchar2(10)	Foreign keys

3.10.7 BOM Table:

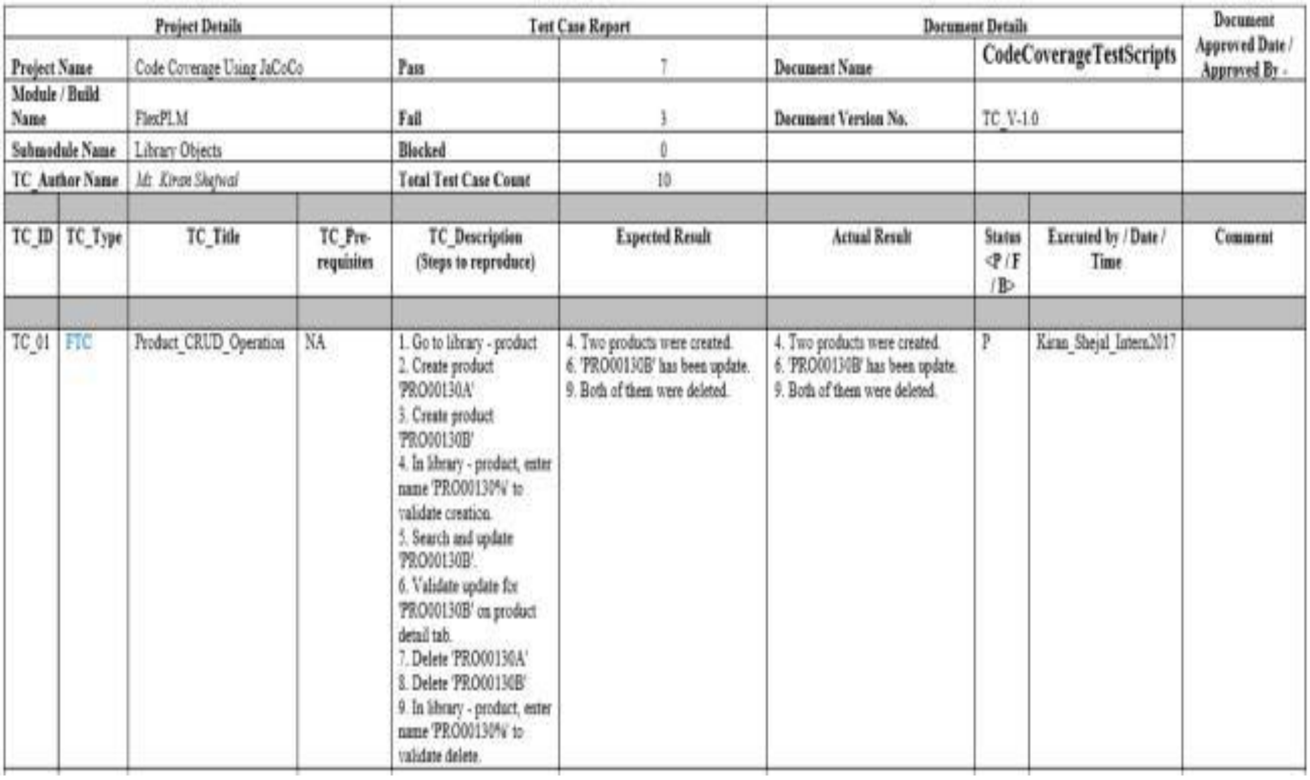
Sr.no.	Field Name	Data Type	Key Constraint
1	BOM ID	Varchar2(10)	Primary key
2	BOM Name	Varchar2(10)	Not Null
3	Material ID	Varchar2(10)	Foreign key
4	Colour ID	Number (10)	Foreign key

3.10.8 Specification Table:

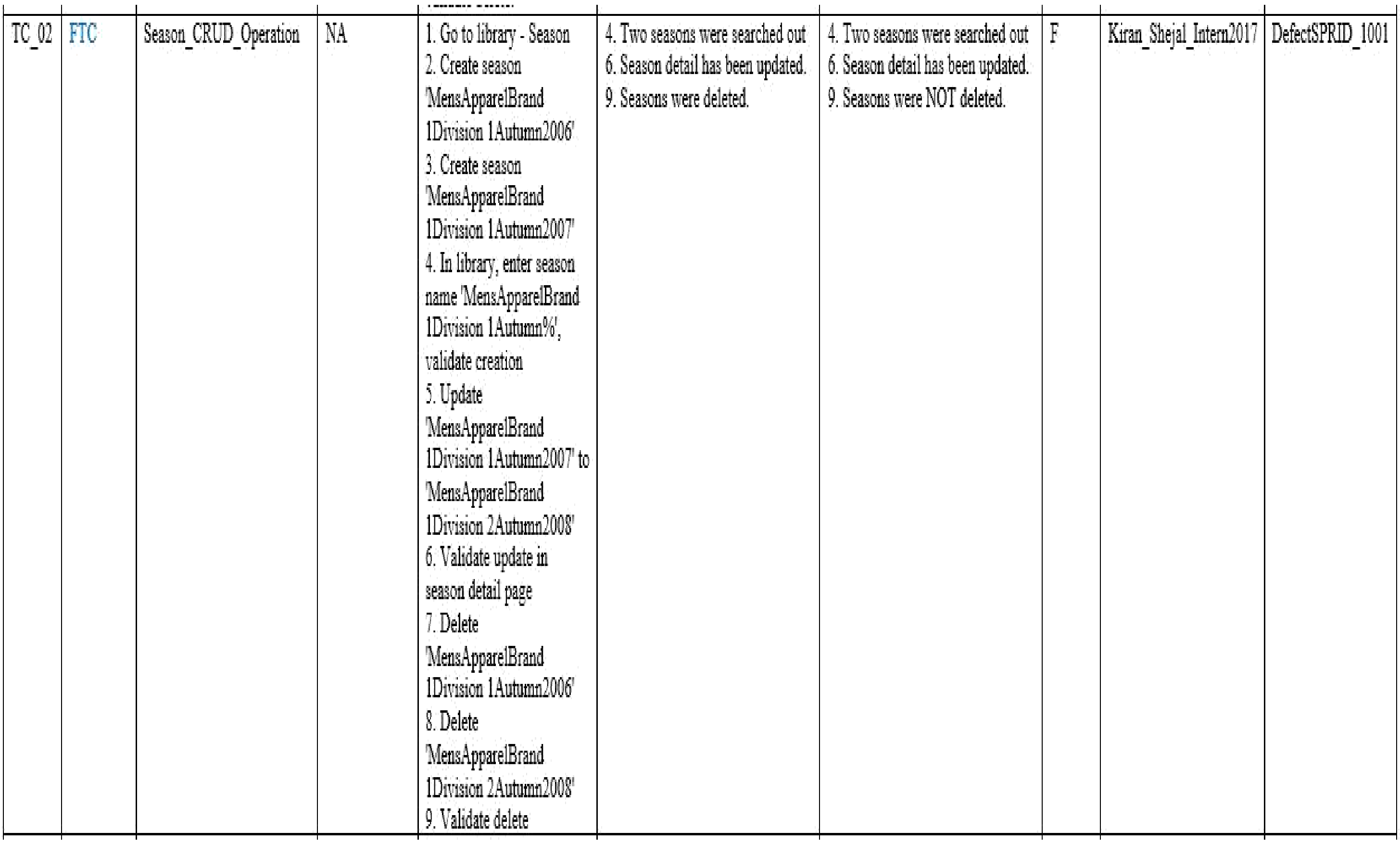
Sr.no.	Field Name	Data Type	Key Constraint
1	Specification ID	Varchar2(10)	Primary key
2	Product ID	Varchar2(10)	Foreign key
3	Specification Name	Varchar2(10)	Not Null
4	Specification Type	Varchar2(10)	Not Null

3.11 Test Procedures and Implementation

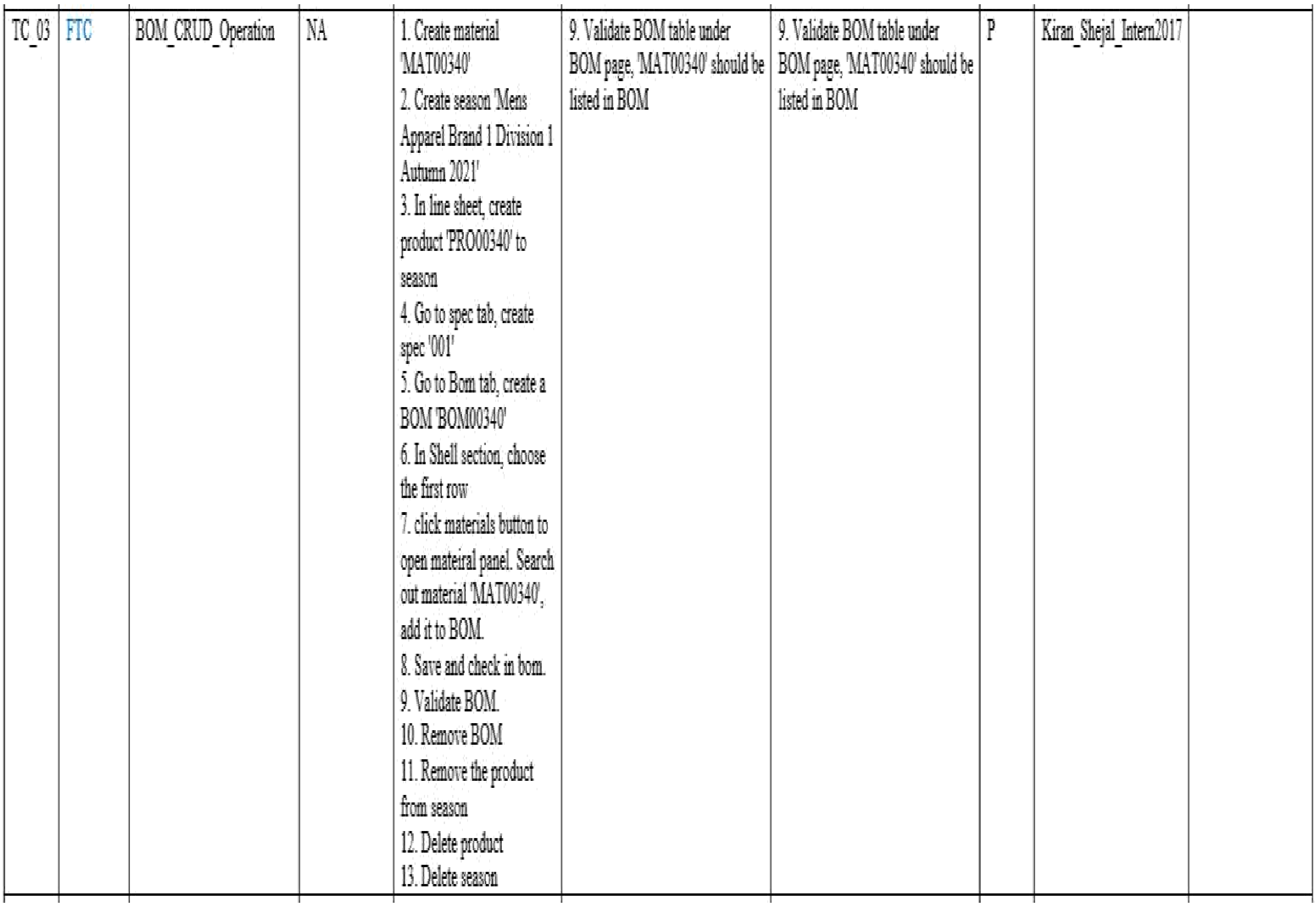
1.Test Cases



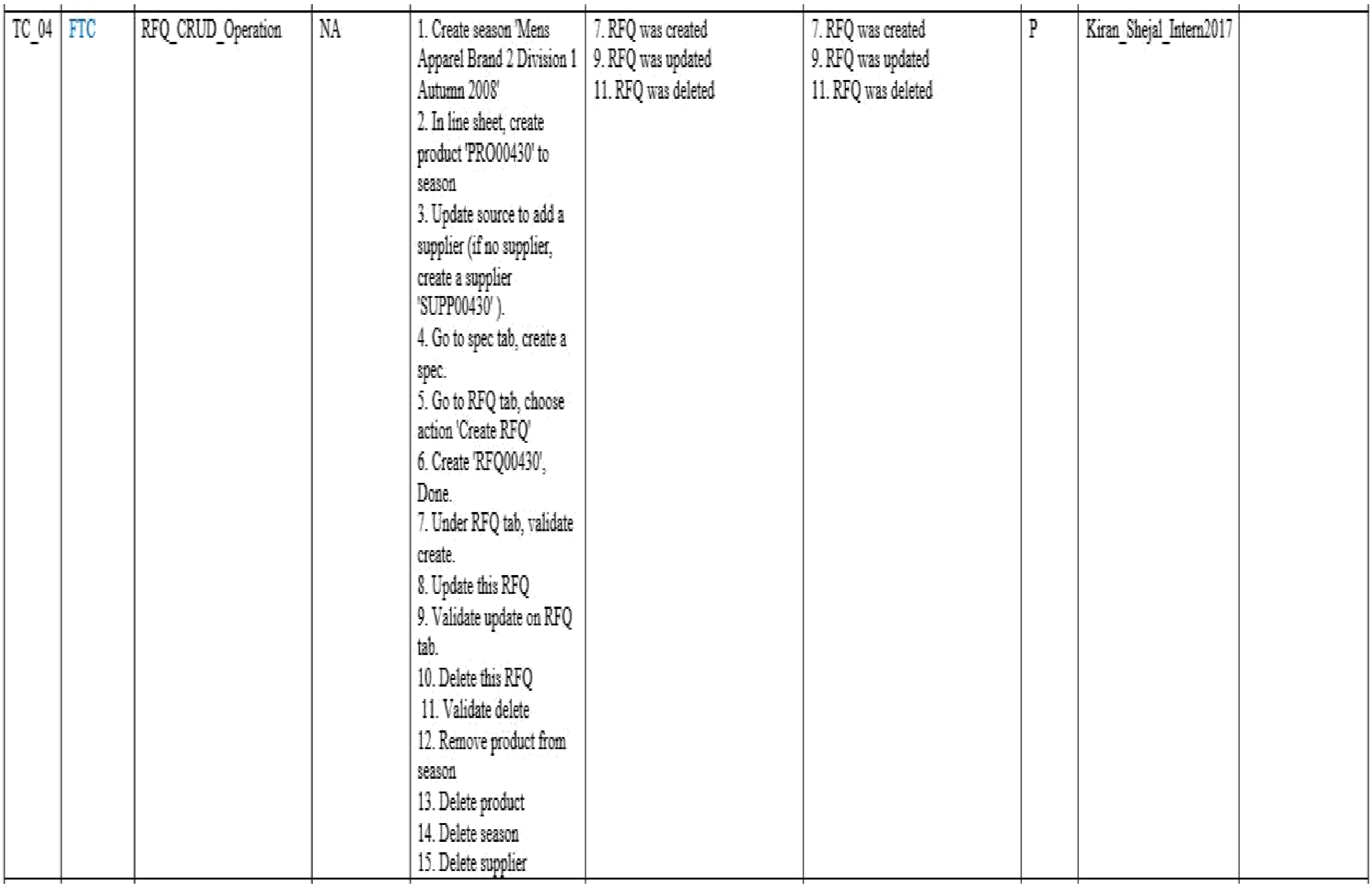
Project Details				Test Case Report		Document Details			Document Approved Date / Approved By -
Project Name		Code Coverage Using JaCoCo		Pass	7	Document Name	CodeCoverageTestScripts		
Module / Build Name		FlexPLM		Fail	3	Document Version No.	TC_V-1.0		
Submodule Name		Library Objects		Blocked	0				
TC_Author Name		Mr. Kiran Shejwal		Total Test Case Count	10				
TC_ID	TC_Type	TC_Title	TC_Pre-requisites	TC_Description (Steps to reproduce)	Expected Result	Actual Result	Status <P / F / B>	Executed by / Date / Time	Comment
TC_01	FTC	Product_CRUD_Operation	NA	1. Go to library - product 2. Create product 'PRO00130A' 3. Create product 'PRO00130B' 4. In library - product, enter name 'PRO00130%' to validate creation. 5. Search and update 'PRO00130B'. 6. Validate update for 'PRO00130B' on product detail tab. 7. Delete 'PRO00130A' 8. Delete 'PRO00130B' 9. In library - product, enter name 'PRO00130%' to validate delete.	4. Two products were created. 6. 'PRO00130B' has been update. 9. Both of them were deleted.	4. Two products were created. 6. 'PRO00130B' has been update. 9. Both of them were deleted.	P	Karan_Shejal_Intern2017	



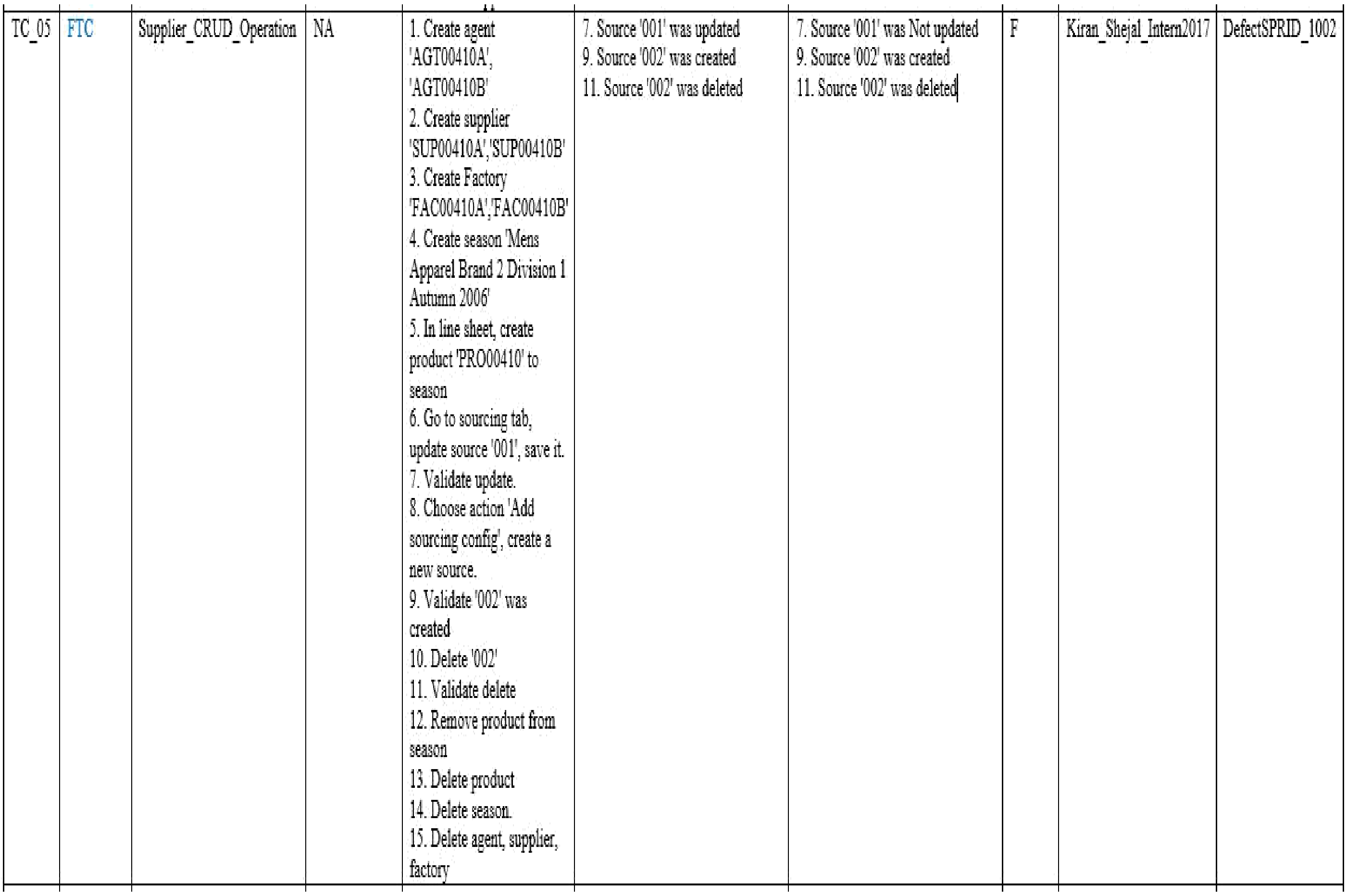
TC_02	FTC	Season_CRUD_Operation	NA	1. Go to library - Season 2. Create season 'MensApparelBrand 1Division 1Autumn2006' 3. Create season 'MensApparelBrand 1Division 1Autumn2007' 4. In library, enter season name 'MensApparelBrand 1Division 1Autumn%', validate creation 5. Update 'MensApparelBrand 1Division 1Autumn2007' to 'MensApparelBrand 1Division 2Autumn2008' 6. Validate update in season detail page 7. Delete 'MensApparelBrand 1Division 1Autumn2006' 8. Delete 'MensApparelBrand 1Division 2Autumn2008' 9. Validate delete	4. Two seasons were searched out 6. Season detail has been updated. 9. Seasons were deleted.	4. Two seasons were searched out 6. Season detail has been updated. 9. Seasons were NOT deleted.	F	Kiran_Shejal_Intern2017	DefectSPRID_1001



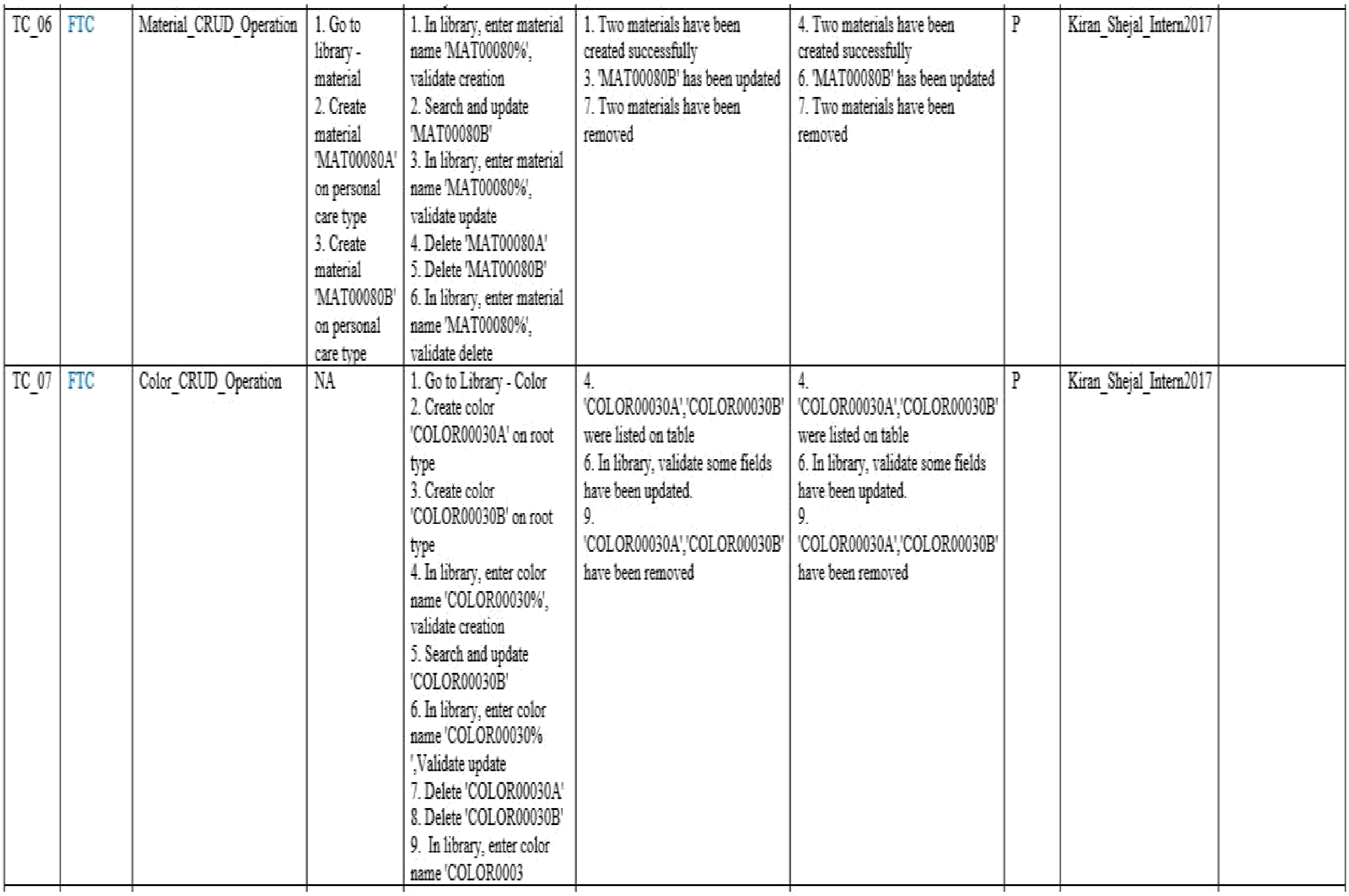
TC_03	FTC	BOM_CRUD_Operation	NA	1. Create material 'MAT00340' 2. Create season 'Mens Apparel Brand 1 Division 1 Autumn 2021' 3. In line sheet, create product 'PRO00340' to season 4. Go to spec tab, create spec '001' 5. Go to Bom tab, create a BOM 'BOM00340' 6. In Shell section, choose the first row 7. click materials button to open mateiral panel. Search out material 'MAT00340', add it to BOM. 8. Save and check in bom. 9. Validate BOM. 10. Remove BOM 11. Remove the product from season 12. Delete product 13. Delete season	9. Validate BOM table under BOM page, 'MAT00340' should be listed in BOM	9. Validate BOM table under BOM page, 'MAT00340' should be listed in BOM	P	Kiran_Shejal_Intern2017	



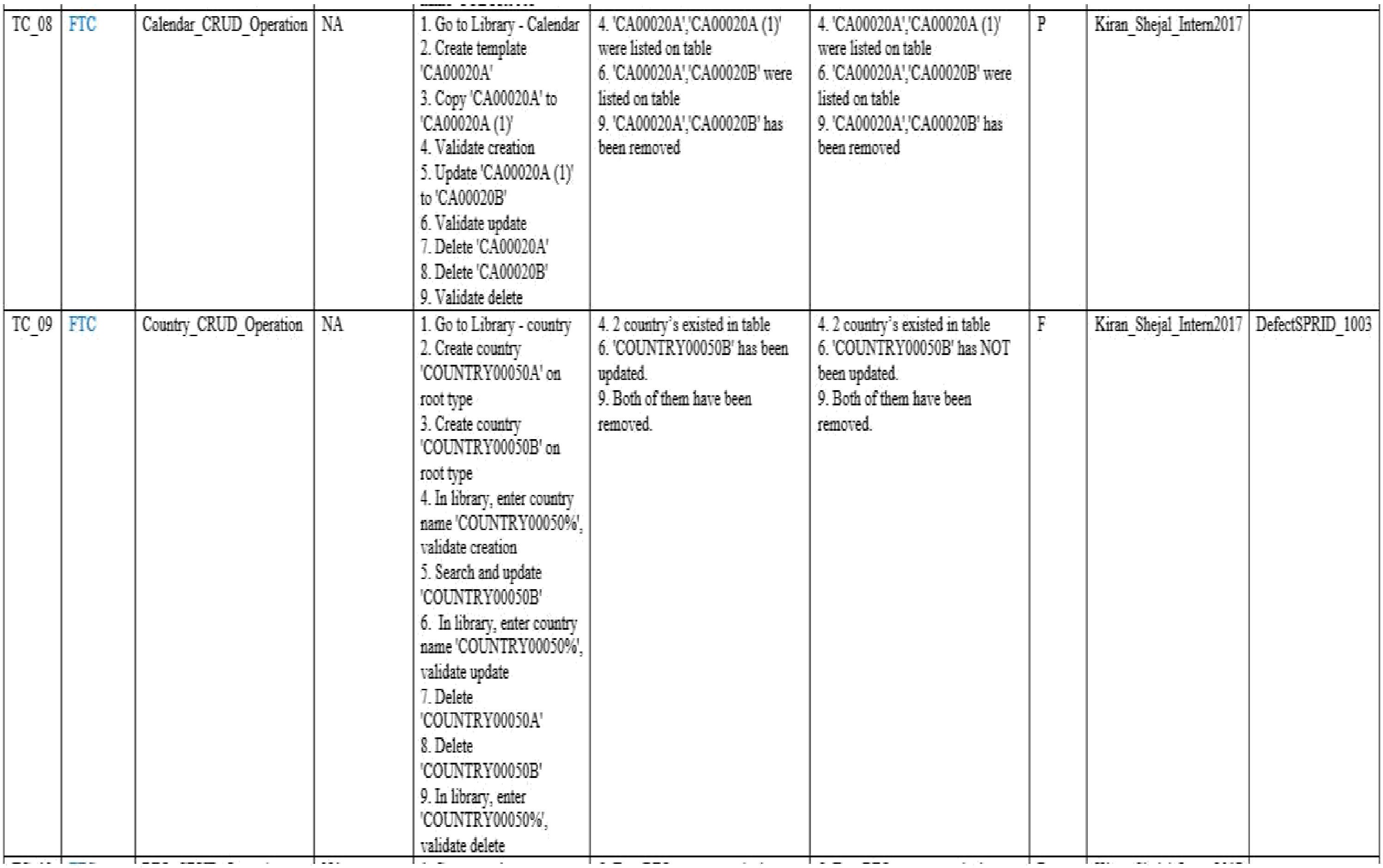
TC_04	FTC	RFQ_CRUD_Operation	NA	1. Create season 'Mens Apparel Brand 2 Division 1 Autumn 2008' 2. In line sheet, create product 'PRO00430' to season 3. Update source to add a supplier (if no supplier, create a supplier 'SUPP00430'). 4. Go to spec tab, create a spec. 5. Go to RFQ tab, choose action 'Create RFQ' 6. Create 'RFQ00430', Done. 7. Under RFQ tab, validate create. 8. Update this RFQ 9. Validate update on RFQ tab. 10. Delete this RFQ 11. Validate delete 12. Remove product from season 13. Delete product 14. Delete season 15. Delete supplier	7. RFQ was created 9. RFQ was updated 11. RFQ was deleted	7. RFQ was created 9. RFQ was updated 11. RFQ was deleted	P	Kiran_Shejal_Intern2017	



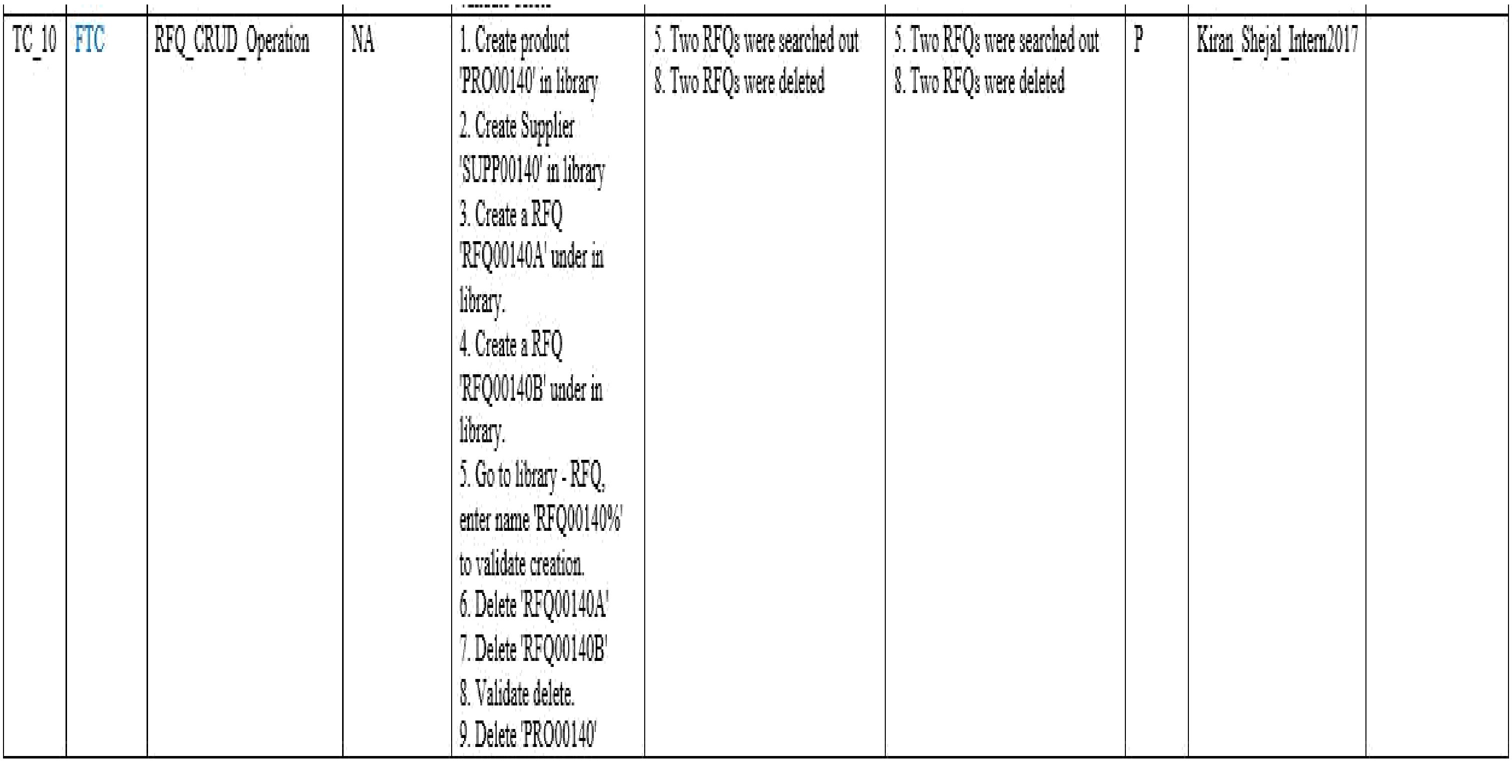
TC_05	FTC	Supplier_CRUD_Operation	NA	1. Create agent 'AGT00410A', 'AGT00410B' 2. Create supplier 'SUP00410A','SUP00410B' 3. Create Factory 'FAC00410A','FAC00410B' 4. Create season 'Mens Apparel Brand 2 Division 1 Autumn 2006' 5. In line sheet, create product 'PRO00410' to season 6. Go to sourcing tab, update source '001', save it. 7. Validate update. 8. Choose action 'Add sourcing config', create a new source. 9. Validate '002' was created 10. Delete '002' 11. Validate delete 12. Remove product from season 13. Delete product 14. Delete season. 15. Delete agent, supplier, factory	7. Source '001' was updated 9. Source '002' was created 11. Source '002' was deleted	7. Source '001' was Not updated 9. Source '002' was created 11. Source '002' was deleted	F	Kiran_Shejal_Intern2017	DefectSPRID_1002



TC_06	FTC	Material_CRUD_Operation	1. Go to library - material 2. Create material 'MAT00080A' on personal care type 3. Create material 'MAT00080B' on personal care type	1. In library, enter material name 'MAT00080%', validate creation 2. Search and update 'MAT00080B' 3. In library, enter material name 'MAT00080%', validate update 4. Delete 'MAT00080A' 5. Delete 'MAT00080B' 6. In library, enter material name 'MAT00080%', validate delete	1. Two materials have been created successfully 3. 'MAT00080B' has been updated 7. Two materials have been removed	4. Two materials have been created successfully 6. 'MAT00080B' has been updated 7. Two materials have been removed	P	Kiran_Shejal_Intern2017	
TC_07	FTC	Color_CRUD_Operation	NA	1. Go to Library - Color 2. Create color 'COLOR00030A' on root type 3. Create color 'COLOR00030B' on root type 4. In library, enter color name 'COLOR00030%', validate creation 5. Search and update 'COLOR00030B' 6. In library, enter color name 'COLOR00030% ',Validate update 7. Delete 'COLOR00030A' 8. Delete 'COLOR00030B' 9. In library, enter color name 'COLOR0003	4. 'COLOR00030A','COLOR00030B' were listed on table 6. In library, validate some fields have been updated. 9. 'COLOR00030A','COLOR00030B' have been removed	4. 'COLOR00030A','COLOR00030B' were listed on table 6. In library, validate some fields have been updated. 9. 'COLOR00030A','COLOR00030B' have been removed	P	Kiran_Shejal_Intern2017	

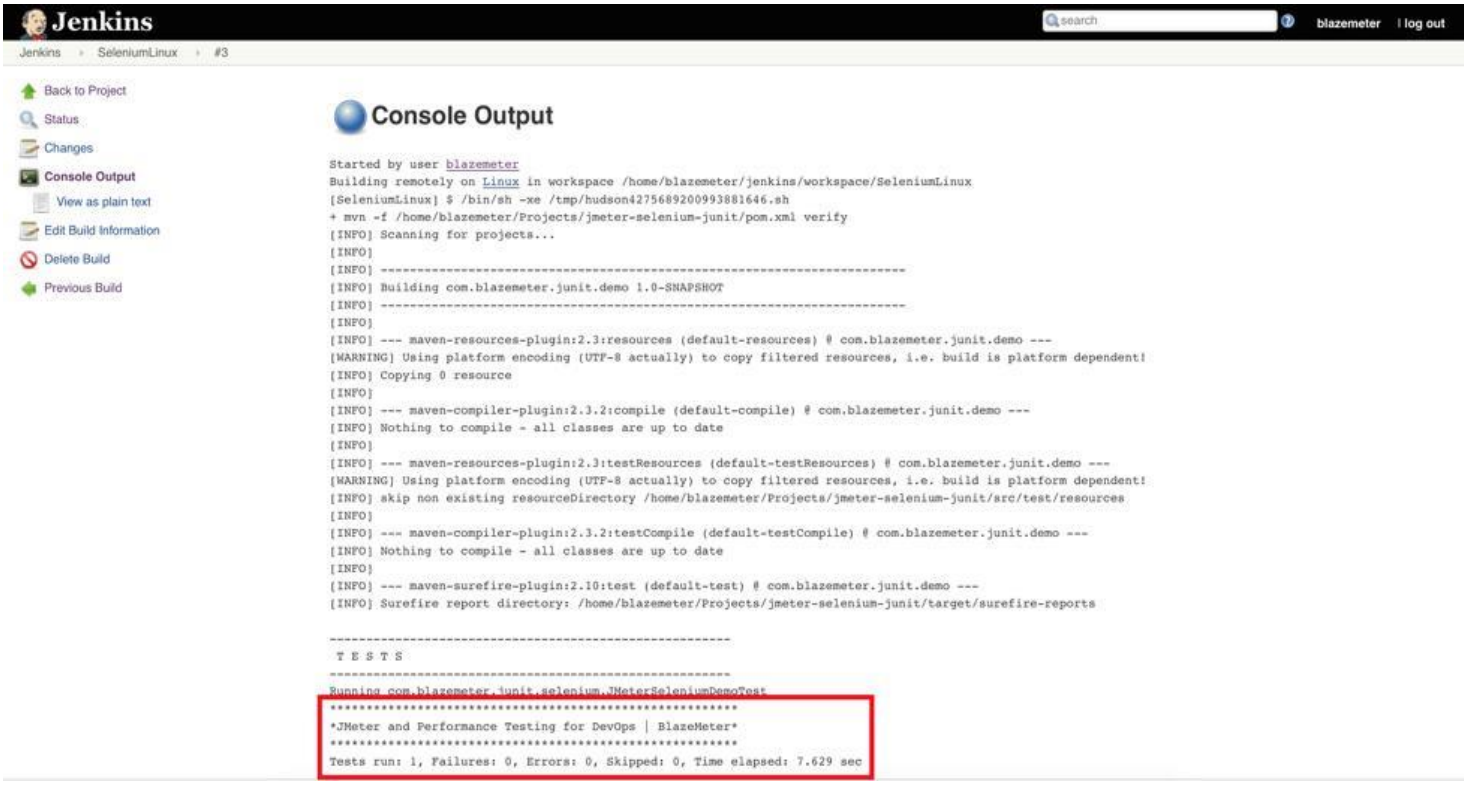


TC_08	FTC	Calendar_CRUD_Operation	NA	1. Go to Library - Calendar 2. Create template 'CA00020A' 3. Copy 'CA00020A' to 'CA00020A (1)' 4. Validate creation 5. Update 'CA00020A (1)' to 'CA00020B' 6. Validate update 7. Delete 'CA00020A' 8. Delete 'CA00020B' 9. Validate delete	4. 'CA00020A','CA00020A (1)' were listed on table 6. 'CA00020A','CA00020B' were listed on table 9. 'CA00020A','CA00020B' has been removed	4. 'CA00020A','CA00020A (1)' were listed on table 6. 'CA00020A','CA00020B' were listed on table 9. 'CA00020A','CA00020B' has been removed	P	Kiran_Shejal_Intern2017	
TC_09	FTC	Country_CRUD_Operation	NA	1. Go to Library - country 2. Create country 'COUNTRY00050A' on root type 3. Create country 'COUNTRY00050B' on root type 4. In library, enter country name 'COUNTRY00050%', validate creation 5. Search and update 'COUNTRY00050B' 6. In library, enter country name 'COUNTRY00050%', validate update 7. Delete 'COUNTRY00050A' 8. Delete 'COUNTRY00050B' 9. In library, enter 'COUNTRY00050%', validate delete	4. 2 country's existed in table 6. 'COUNTRY00050B' has been updated. 9. Both of them have been removed.	4. 2 country's existed in table 6. 'COUNTRY00050B' has NOT been updated. 9. Both of them have been removed.	F	Kiran_Shejal_Intern2017	DefectSPRID_1003



TC_10	FTC	RFQ_CRUD_Operation	NA	1. Create product 'PRO00140' in library 2. Create Supplier 'SUPP00140' in library 3. Create a RFQ 'RFQ00140A' under in library. 4. Create a RFQ 'RFQ00140B' under in library. 5. Go to library - RFQ, enter name 'RFQ00140%' to validate creation. 6. Delete 'RFQ00140A' 7. Delete 'RFQ00140B' 8. Validate delete. 9. Delete 'PRO00140'	5. Two RFQs were searched out 8. Two RFQs were deleted	5. Two RFQs were searched out 8. Two RFQs were deleted	P	Kiran_Shejal_Intern2017	

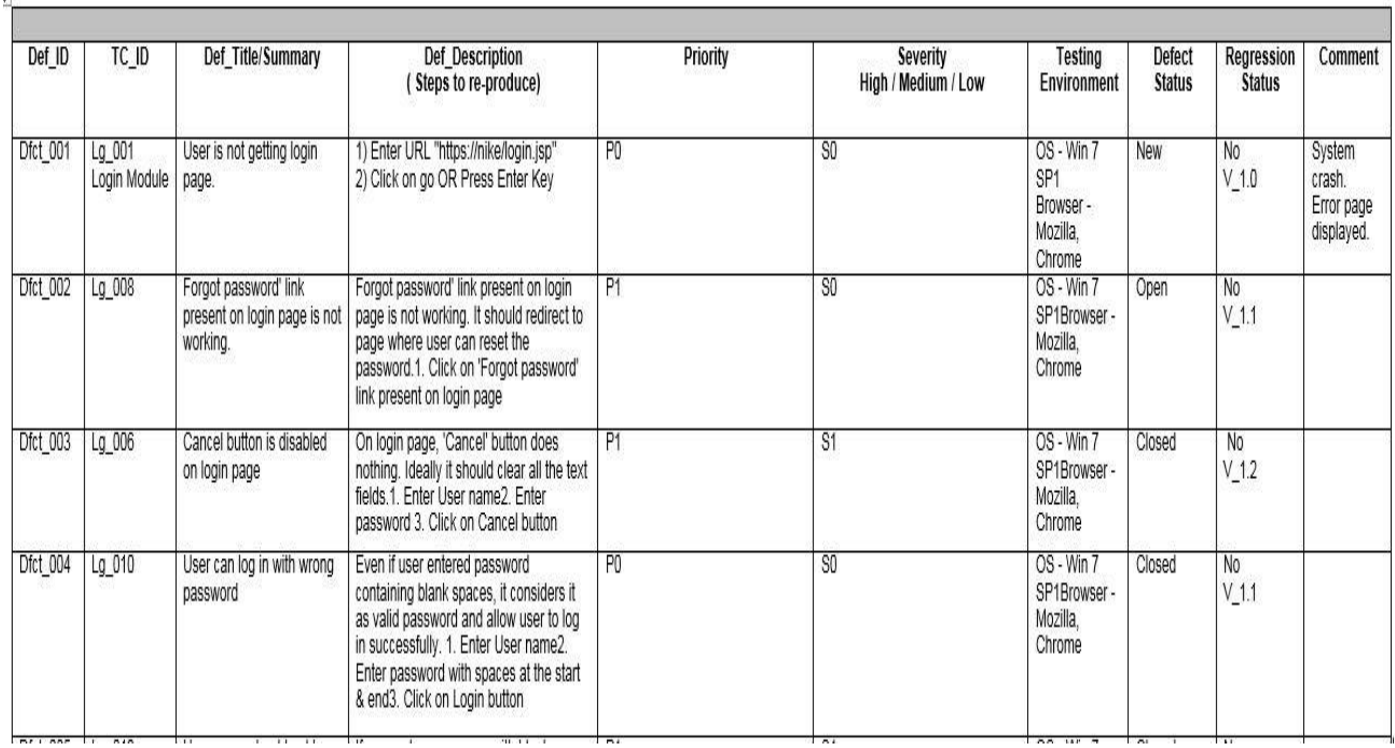
1.Test Case Execution log



Console Output

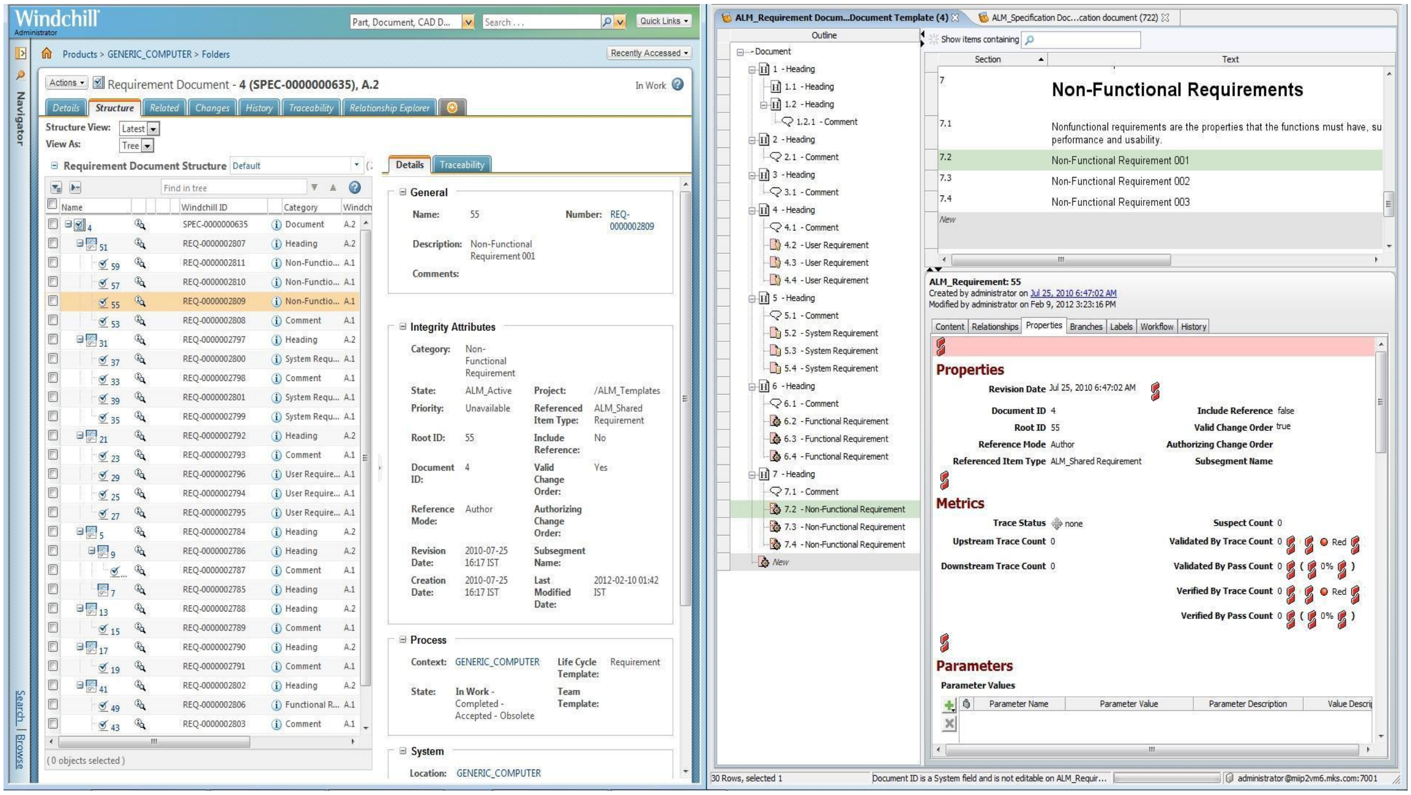
```
Started by user blazemeter
Building remotely on Linux in workspace /home/blazemeter/jenkins/workspace/SeleniumLinux
[SeleniumLinux] $ /bin/sh -xe /tmp/hudson4275689200993881646.sh
+ mvn -f /home/blazemeter/Projects/jmeter-selenium-junit/pom.xml verify
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building com.blazemeter.junit.demo 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.3:resources (default-resources) @ com.blazemeter.junit.demo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ com.blazemeter.junit.demo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.3:testResources (default-testResources) @ com.blazemeter.junit.demo ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/blazemeter/Projects/jmeter-selenium-junit/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ com.blazemeter.junit.demo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ com.blazemeter.junit.demo ---
[INFO] Surefire report directory: /home/blazemeter/Projects/jmeter-selenium-junit/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.blazemeter.junit.selenium.JMeterSeleniumDemoTest
*******************************************************
*JMeter and Performance Testing for DevOps | BlazeMeter*
*******************************************************
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 7.629 sec
```

2.Defect Metrics



Def_ID	TC_ID	Def_Title/Summary	Def_Description (Steps to re-produce)	Priority	Severity High / Medium / Low	Testing Environment	Defect Status	Regression Status	Comment
Dfct_001	Lg_001 Login Module	User is not getting login page.	1) Enter URL "https://nike/login.jsp" 2) Click on go OR Press Enter Key	P0	S0	OS - Win 7 SP1 Browser - Mozilla, Chrome	New	No V_1.0	System crash. Error page displayed.
Dfct_002	Lg_008	Forgot password' link present on login page is not working.	Forgot password' link present on login page is not working. It should redirect to page where user can reset the password.1. Click on 'Forgot password' link present on login page	P1	S0	OS - Win 7 SP1Browser - Mozilla, Chrome	Open	No V_1.1	
Dfct_003	Lg_006	Cancel button is disabled on login page	On login page, 'Cancel' button does nothing. Ideally it should clear all the text fields.1. Enter User name2. Enter password 3. Click on Cancel button	P1	S1	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.2	
Dfct_004	Lg_010	User can log in with wrong password	Even if user entered password containing blank spaces, it considers it as valid password and allow user to log in successfully. 1. Enter User name2. Enter password with spaces at the start & end3. Click on Login button	P0	S0	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.1	

3.Defect screen along with defect



Requirement Document - **4 (SPEC-0000000635), A.2**

In Work

Structure View: Latest

View As: Tree

Requirement Document Structure Default

Name	Windchill ID	Category	Windch
4	SPEC-0000000635	Document	A.2
51	REQ-0000002807	Heading	A.2
59	REQ-0000002811	Non-Functio...	A.1
57	REQ-0000002810	Non-Functio...	A.1
55	REQ-0000002809	Non-Functio...	A.1
53	REQ-0000002808	Comment	A.1
31	REQ-0000002797	Heading	A.2
37	REQ-0000002800	System Requ...	A.1
33	REQ-0000002798	Comment	A.1
39	REQ-0000002801	System Requ...	A.1
35	REQ-0000002799	System Requ...	A.1
21	REQ-0000002792	Heading	A.2
23	REQ-0000002793	Comment	A.1
29	REQ-0000002796	User Require...	A.1
25	REQ-0000002794	User Require...	A.1
27	REQ-0000002795	User Require...	A.1
5	REQ-0000002784	Heading	A.2
9	REQ-0000002786	Heading	A.2
...	REQ-0000002787	Comment	A.1
7	REQ-0000002785	Heading	A.1
13	REQ-0000002788	Heading	A.2
15	REQ-0000002789	Comment	A.1
17	REQ-0000002790	Heading	A.2
19	REQ-0000002791	Comment	A.1
41	REQ-0000002802	Heading	A.2
49	REQ-0000002806	Functional R...	A.1
43	REQ-0000002803	Comment	A.1

(0 objects selected)

Details | Traceability

General

Name: 55 Number: REQ-0000002809

Description: Non-Functional Requirement 001

Comments:

Integrity Attributes

Category: Non-Functional Requirement

State: ALM_Active Project: /ALM_Templates

Priority: Unavailable Referenced Item Type: ALM_Shared Requirement

Root ID: 55 Include Reference: No

Document ID: 4 Valid Change Order: Yes

Reference Mode: Author Authorizing Change Order:

Revision Date: 2010-07-25 16:17 IST Subsegment Name:

Creation Date: 2010-07-25 16:17 IST Last Modified Date: 2012-02-10 01:42 IST

Process

Context: GENERIC_COMPUTER Life Cycle Template: Requirement

State: In Work - Completed - Accepted - Obsolete Team Template:

System

Location: GENERIC_COMPUTER

- Document
  - 1 - Heading
    - 1.1 - Heading
    - 1.2 - Heading
      - 1.2.1 - Comment
  - 2 - Heading
    - 2.1 - Comment
  - 3 - Heading
    - 3.1 - Comment
  - 4 - Heading
    - 4.1 - Comment
    - 4.2 - User Requirement
    - 4.3 - User Requirement
    - 4.4 - User Requirement
  - 5 - Heading
    - 5.1 - Comment
    - 5.2 - System Requirement
    - 5.3 - System Requirement
    - 5.4 - System Requirement
  - 6 - Heading
    - 6.1 - Comment
    - 6.2 - Functional Requirement
    - 6.3 - Functional Requirement
    - 6.4 - Functional Requirement
  - 7 - Heading
    - 7.1 - Comment
    - 7.2 - Non-Functional Requirement
    - 7.3 - Non-Functional Requirement
    - 7.4 - Non-Functional Requirement
  - New

Section	Text
7	**Non-Functional Requirements**
7.1	Nonfunctional requirements are the properties that the functions must have, su performance and usability.
7.2	Non-Functional Requirement 001
7.3	Non-Functional Requirement 002
7.4	Non-Functional Requirement 003
New	

ALM_Requirement: 55
Created by administrator on Jul 25, 2010 6:47:02 AM
Modified by administrator on Feb 9, 2012 3:23:16 PM

Properties

Revision Date Jul 25, 2010 6:47:02 AM

Document ID 4 Include Reference false

Root ID 55 Valid Change Order true

Reference Mode Author Authorizing Change Order

Referenced Item Type ALM_Shared Requirement Subsegment Name

Metrics

Trace Status none Suspect Count 0

Upstream Trace Count 0 Validated By Trace Count 0 ● Red

Downstream Trace Count 0 Validated By Pass Count 0 (0%)

Verified By Trace Count 0 ● Red

Verified By Pass Count 0 (0%)

Parameters

Parameter Values

Parameter Name	Parameter Value	Parameter Description	Value Descri

30 Rows, selected 1 Document ID is a System field and is not editable on ALM_Requir... administrator@miip2vm6.mks.com:7001

CHAPTER 4:
USER MANUAL

4.1 User Manual

4.1.1 Flex PLM

Many forward-looking companies are implementing PLM solutions to meet strategic corporate goals and emerge stronger as the global economy fluctuates. With profit margins diminishing, it is more important than ever to make smart IT investments. Today, the industry is under pressure to deliver innovative, “trend-right” products to market faster – without sacrificing margin or quality. Reducing cycle time is therefore key to capturing trends that fuel consumer demand. Achieving these goals requires collaboration across key roles in both the enterprise and the supply chain – early in the product development process. For the world’s top companies in retail, footwear & apparel and consumer products, PTC Windchill FlexPLM is the solution.

4.1.2 JaCoCo Tool: JaCoCo uses a set of different counters to calculate coverage metrics. All these counters are derived from information contained in Java class files which basically are Java byte code instructions and debug information optionally embedded in class files. This approach allows efficient on-the-fly instrumentation and analysis of applications even when no source code is available. In most cases the collected information can be mapped back to source code and visualized down to line level granularity. Anyhow there are limitations to this approach. The class files have to be compiled with debug information to calculate line level coverage and provide source highlighting. Not all Java language constructs can be directly compiled to corresponding byte code. In such cases the Java compiler creates so called *synthetic* code which sometimes results in unexpected code coverage results**.** You can instrument your source code with JaCoCo in two ways

1. Offline Instrumentation and 2. Run time Instrumentation

4.1.2 Key Features

Ease-of-use: designed specifically for the retail, footwear & apparel and consumer products markets

5. Breadth of functionality: supports collaboration – early in the product development process – between merchandising, design, product development and sourcing

6. Depth of functionality: supports the unique requirements of designing and manufacturing apparel, footwear, accessories, home goods and other consumer-related products with PTC Windchill FlexPLM, PTC customers have realized significant improvements in cycle time, margin, product quality and overall operational efficiency.

Built on Proven PTC Windchill Architecture:

PTC Windchill FlexPLM leverages PTC Windchill architecture – a fully integrated platform that's been proven in process-intensive industries to improve the productivity of thousands of concurrent users. Some of the many business benefits of the software include:

- Increased operational efficiency. Enables streamlined collaboration across a global environment, ensuring that trend-right products get to market faster, without sacrificing margin or quality

- Reduced costs. Provides visibility to key cost-drivers in the product development process, enabling decision-makers to easily identify savings

- Reduced cycle time. Impacts all roles in the product development process. By enabling greater collaboration and visibility across all roles, you can eliminate inefficiencies in the product development process and ultimately reduce cycle time

4.1.3 Capabilities:

- A centralized product repository to share and control all forms of product development information throughout the value chain

- A scalable, Web-based architecture that can be optimized for maximum performance in small and large implementations, as well as across organizational and geographical boundaries

- A process management engine that enables administration of seasonal calendars, milestone tracking, and supply chain collaboration

- Direct integration with design tools such as Adobe® Illustrator™ to increase design efficiency and allow designers to work with familiar applications

- Easy-to-use vendor portal that allows suppliers to log in and collaborate in real-time on core product data

4.1.4 Supported Business Processes

4.1.4.1 Calendar Management/New Product Introduction

9. Define seasonal calendars, including key product development milestones

10. Track progress of individual products against the seasonal calendar

4.1.4.2 Concept Development

3) Create storyboards of design trends, themes and concepts

4) Collect and control all forms of digital product data

4.1.4.3 Merchandise Planning

• Define business plans per financial, attribute or assortment goals

3) Identify seasonal contribution/targets per business objectives

• Track line development against seasonal targets

4) Create product plans containing size and colourway matrices, indicating which combinations should be developed

5)Communicate product plans to vendors when requesting quotes

• Leverage product plans when sending commitments to vendors

4.1.4.4 Line Planning

6. Create your own role-based home page and customize how the application looks based on your specific position

7.Generate product placeholders from a merchandise plan, either manually or automatically

• Optimize the merchandising mix across company, brands, regions and distribution channels

10. Create line sheets and line presentation boards

11. Track actual product adoption against merchandise plans

4.1.4.5 Colour Development

- Establish and enforce colour definition using industry

standards

11. Define and regulate colour usage per seasonal palettes

12. Associate colour to material and supplier

13. Automate lab dip process

14. Integrate colour definition and quality control systems

Materials Development & Testing

5. Formalize a comprehensive materials library structure

6. Create material specifications

7. Manage materials- and supplier-specific information on

cost and quality

8. Determine materials usage – including product and

seasonal commitments

9. Manage materials testing

• Track “where-used” for materials Product Specifications Management

• Create comprehensive product specifications, including images, CAD data, bills-of-materials, size definition, measurements, construction details and bills-of-labour

• Standardize points of measure and graded measurements for apparel

7. Maintain multiple levels of the specification at the product/colourway, size, and region and supplier level

8.Generate zip files containing both product specifications and reports in PDF format, as well as product documents. Easily export zip files and/or automatically store them with products

8. Carry over best-selling products from previous seasons

- Apply multiple easy to use product-specification creation features, including Quick Specs, Advanced Copy and Linked Products

4.1.4.6 Change Tracking

- Track line item changes made to products, colours, sources, cost sheets and specification data

- Identify product changes using change notification icons

- Create change tracking reports to efficiently communicate and manage change throughout product development

Document version management

- Create multiple versions of a document, allowing them to evolve while maintaining traceability

- Leverage document versions to describe seasons, Products, and product specifications Sample management

- Request samples (e.g., concept, photo, prototype, fit, production, etc.) from internal or external resources

- Evaluate and provide disposition on received samples

- Track sample status and sample statistics Supplier management

- Maintain a database of suppliers

- Manage supplier profiles, including capabilities and capacity, as well as social, technical and environmental compliance

- Search for suppliers

4.1.4.7 Early sourcing

- Create RFQ across multiple products for multiple vendors simultaneously

- Create costing templates that guide vendors in the bidding process

- Manage pricing negotiations by providing vendors with RFQ response capabilities

- Award products to one or more vendors Commitment management

- Generate order confirmations to selected vendor(s) containing product specifications (complete specs or Quick Specs)

- Include product plan information (size/colourway combinations and volumes) for one or more products with an order confirmation

- Capture agreed-upon/negotiated costing from theft process within the confirmation

- Leverage the order commitment as a PO, or export the data to ERP/PO systems costing

- Generate product cost based on bill-of-materials or FOB

- Conduct multi-level costing by colour, size, and region and distribution channel

- Create “what-if” costing scenarios.

4.2 Operations Manual / Menu Explanation:

4.2.1 My Home:

This menu provides link to direct us to the of FlexPLM server homepage

4.2.2 My work

This menu shows the assigned work for the logged in user

4.2.3 My Favourites

This menu displays the favourite contents marked by the logged in user

4.2.4 My season

This menu displays the different object created for the function season

4.2.5 Libraries

This menu contains all the out of the box Flex Typed object on which we can work i.e. Season, Product.

4.2.6 Reports:

This menu contains links for reports for specific data that user want check.

4.2.6 Administrative:

This option available only for the Administrative user. From this menu we can explore the Type manager section.

Drawbacks and Limitations:

JaCoCo tool is only used with the Java. It does not support the other languages like VB, C, C++ etc. Customized reports are not supported.

It is giving only the percentage wise code coverage report It does not give the proper package and the classes which has the lowest coverage.

Code coverage reports does not satisfy the user in terms of detailed level code coverage.

It does not generate the customized reports so that we need to implement it by our self.

Proposed Enhancements

Customised reports are created in future so that the classes and packages having less code coverage will be highlighted so that the tester should get to know that these areas are not covered by the test cases and he will draft the new test cases for that specific area.

Going further, will map the uncovered classes to the test cases, so that we will get to know the classes and related test cases to cover

Also with this the newly added code will be able to pick the test cases which are candidate for automation in order to cover and test the newly added code.

Conclusion

Code coverage and test coverage metrics are both measurements that can be useful to assess the quality of your application code.

Code coverage is a measurement of how many lines/blocks/arcs of your code are executed while the tests are running. In this expert response, you'll learn how quality assurance professionals use both of these metrics effectively.

Coverage information has to be collected at runtime. For this purpose, JaCoCo creates instrumented versions of the original class definitions.

The reports will be early analysis to expose the uncovered code to test ensuring 100% coverage for the project.

The code coverage will help you to do early analysis of how much code is covered by the test case execution.

Bibliography

http://www.jacoco.org

https://en.wikipedia.org/wiki/Java_Code_Coverage_Tools

http://www.eclemma.org

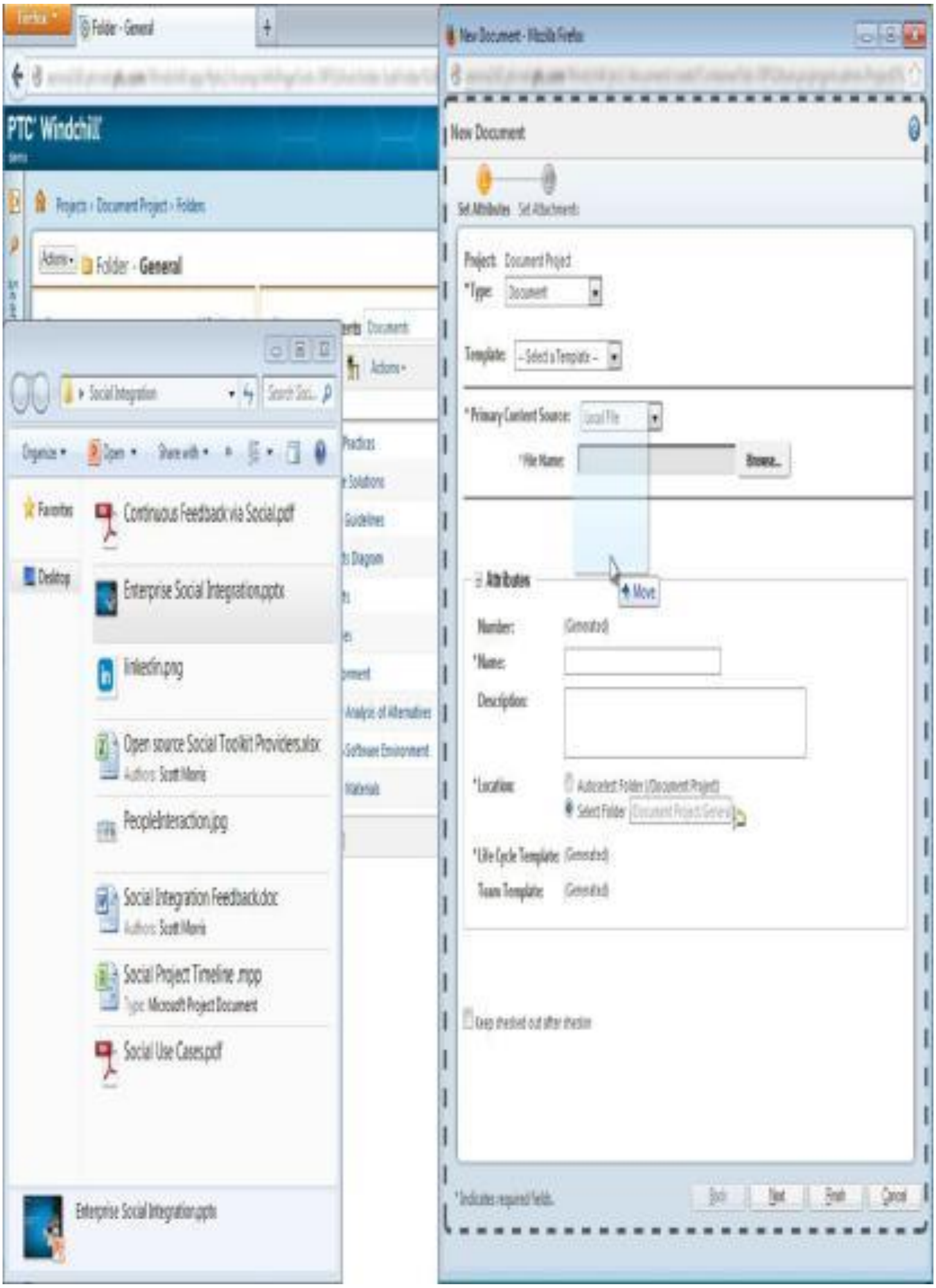
https://www.codeproject.com

http://automationrhapsody.com

http://stackoverflow.com/

ANNEXURES

Annexure 1: User Interface Screen



Continuous Feedback via Social.pdf

Enterprise Social Integration.pptx

linkedin.png

Open source Social Toolkit Providers.xlsx
Authors: Scott Morris

PeopleInteraction.jpg

Social Integration Feedback.doc
Authors: Scott Morris

Social Project Timeline .mpp
Type: Microsoft Project Document

Social Use Cases.pdf

Enterprise Social Integration.pptx

Project: Document Project

*Type: Document

Template: -- Select a Template --

*Primary Content Source: Local File

*File Name:

Number: (Generated)

*Name:

Description:

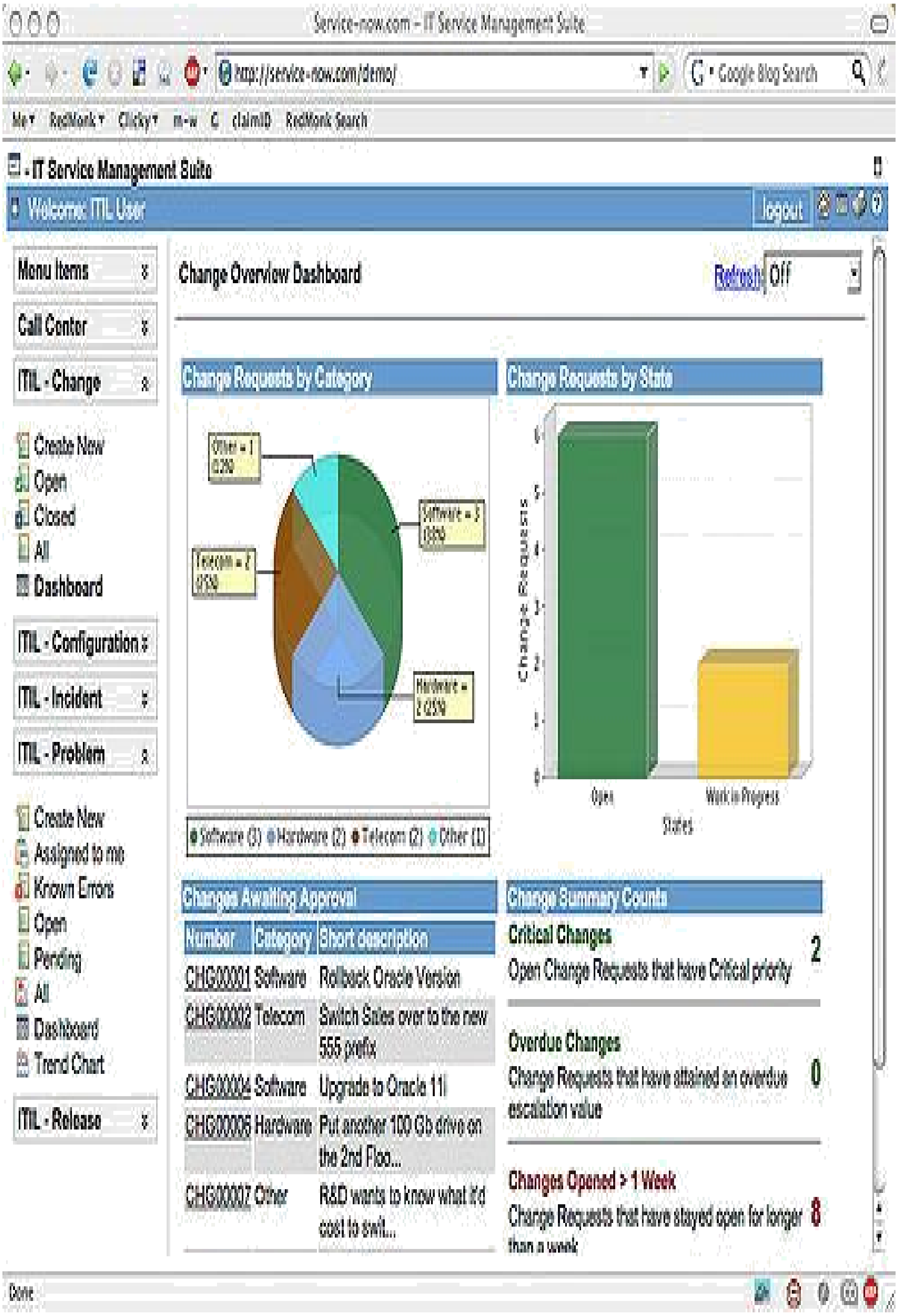
*Location: ○ Autoselect Folder (/Document Project)
● Select Folder: /Document Project/General

*Life Cycle Template: (Generated)

Team Template: (Generated)

□ Keep checked out after checkin

*Indicates required fields.



Change Overview Dashboard

Change Requests by Category

Change Requests by State

Changes Awaiting Approval

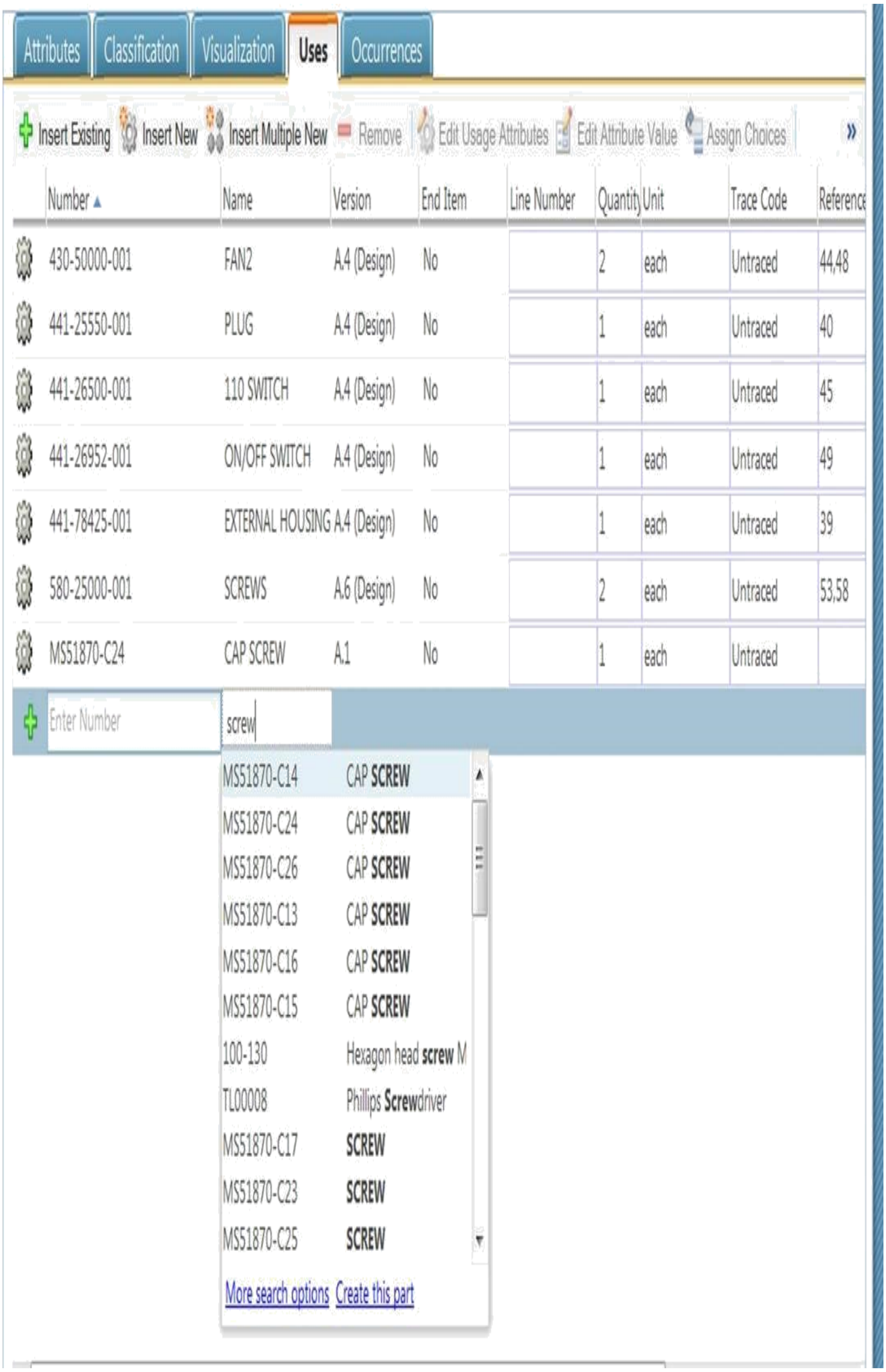
Number	Category	Short description
CHG00001	Software	Rollback Oracle Version
CHG00002	Telecom	Switch Sales over to the new 555 prefix
CHG00004	Software	Upgrade to Oracle 11i
CHG00006	Hardware	Put another 100 Gb drive on the 2nd Floo...
CHG00007	Other	R&D wants to know what it'd cost to swit...

Change Summary Counts

Critical Changes 2
Open Change Requests that have Critical priority

Overdue Changes 0
Change Requests that have attained an overdue escalation value

Changes Opened > 1 Week 8
Change Requests that have stayed open for longer than a week

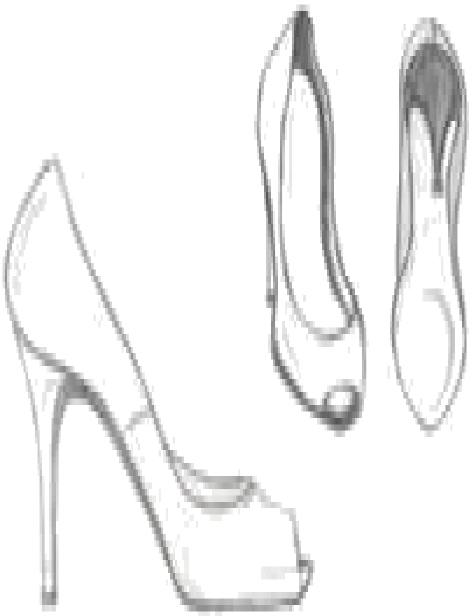


Number ▲	Name	Version	End Item	Line Number	Quantity	Unit	Trace Code	Reference
430-50000-001	FAN2	A.4 (Design)	No		2	each	Untraced	44,48
441-25550-001	PLUG	A.4 (Design)	No		1	each	Untraced	40
441-26500-001	110 SWITCH	A.4 (Design)	No		1	each	Untraced	45
441-26952-001	ON/OFF SWITCH	A.4 (Design)	No		1	each	Untraced	49
441-78425-001	EXTERNAL HOUSING	A.4 (Design)	No		1	each	Untraced	39
580-25000-001	SCREWS	A.6 (Design)	No		2	each	Untraced	53,58
MS51870-C24	CAP SCREW	A.1	No		1	each	Untraced	
Enter Number	screw							

MS51870-C14	CAP **SCREW**
MS51870-C24	CAP **SCREW**
MS51870-C26	CAP **SCREW**
MS51870-C13	CAP **SCREW**
MS51870-C16	CAP **SCREW**
MS51870-C15	CAP **SCREW**
100-130	Hexagon head **screw** M
TL00008	Phillips **Screw**driver
MS51870-C17	**SCREW**
MS51870-C23	**SCREW**
MS51870-C25	**SCREW**

Product Information

Clase Producto : 137658
Numero diseño : 134600
División : Moda
Genero : Dama
Product Group : Zapatilla
Tipo tela : tafeta
Marca : FS Woman



Materiales

A

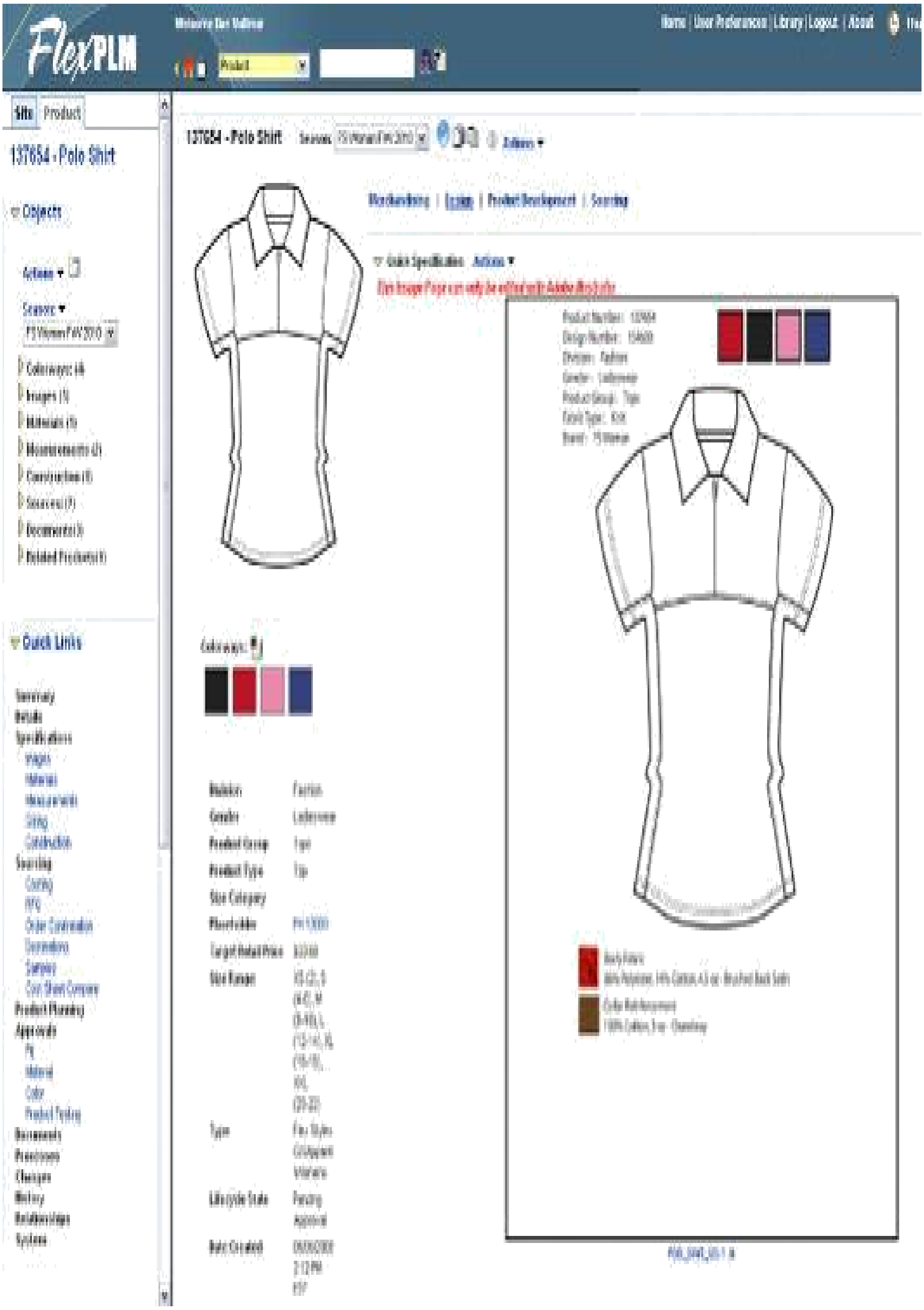
B

C

D

Colorido

	Materiales	Rojo	Negro	Rosa	Azul
A	Tela del cuerpo 86% Poliester, 14% Algodon, 4.5 oz	🟥	⬛	🟪	🟦
B	Refuerzo 100% Algodon, 5 oz - Chambray	110	110	110	110
C					
D					



Welcome Dan Sullivan

Product

Home | User Preferences | Library | Logout | About

Site | Product

137654 - Polo Shirt

Objects

Actions

Season

FS Women F/W 2010

Colorways (4)

Images (1)

Materials (1)

Measurements (2)

Construction (1)

Sourcing (7)

Documents (1)

Related Products (1)

Quick Links

Summary

Details

Specifications

Images

Materials

Measurements

Sizing

Construction

Sourcing

Costing

RFQ

Order Confirmation

Downloading

Samples

Cost Sheet Compare

Product Planning

Approvals

Fit

Material

Color

Product Testing

Documents

Processes

Changes

History

Relationships

Systems

137654 - Polo Shirt Seasons: FS Women F/W 2010 Actions

Merchandising | Design | Product Development | Sourcing

Quick Specification Actions

This Image Page can only be edited with Adobe Illustrator

Colorways:

Division	Fashion
Gender	Ladieswear
Product Group	Tops
Product Type	Top
Size Category	
Placeholder	FH 1000
Target Retail Price	$32.00
Size Range	XS (2), S (4-6), M (8-10), L (12-14), XL (16-18), XXL (20-22)
Type	Flex Styles Cl/Apparel Womens
Lifecycle State	Pending Approval
Date Created	09/09/2009 2:13 PM PST

Product Number: 137654

Design Number: 15400

Division: Fashion

Gender: Ladieswear

Product Group: Tops

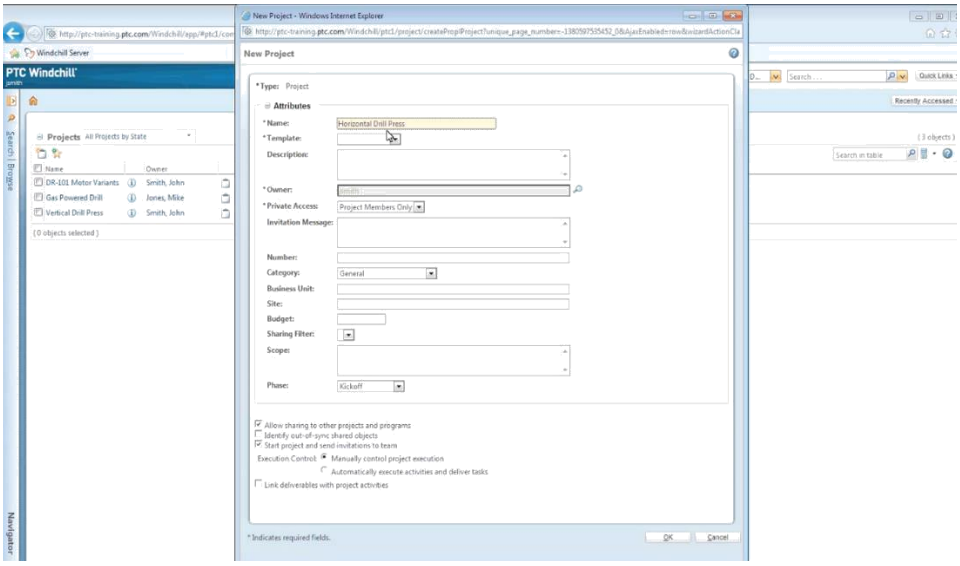
Retail Type: Knit

Brand: FS Women

Body Fabric
86% Polyester, 14% Cotton, 4.5 oz - Brushed Back Satin

Collar Rib Assortment
100% Cotton, 5 oz - Chambray

POLO_0445_SS-1 .ai



Projects All Projects by State

☐ Name		Owner
☐ DR-101 Motor Variants	ⓘ	Smith, John
☐ Gas Powered Drill	ⓘ	Jones, Mike
☐ Vertical Drill Press	ⓘ	Smith, John

(0 objects selected)

New Project - Windows Internet Explorer

http://ptc-training.ptc.com/Windchill/ptc1/project/createProp!Project?unique_page_number=-1380597535452_0&AjaxEnabled=row&wizardActionCla

New Project

*Type: Project

Attributes

*Name: Horizontal Drill Press

*Template:

Description:

*Owner: smith

*Private Access: Project Members Only

Invitation Message:

Number:

Category: General

Business Unit:

Site:

Budget:

Sharing Filter:

Scope:

Phase: Kickoff

☑ Allow sharing to other projects and programs

☐ Identify out-of-sync shared objects

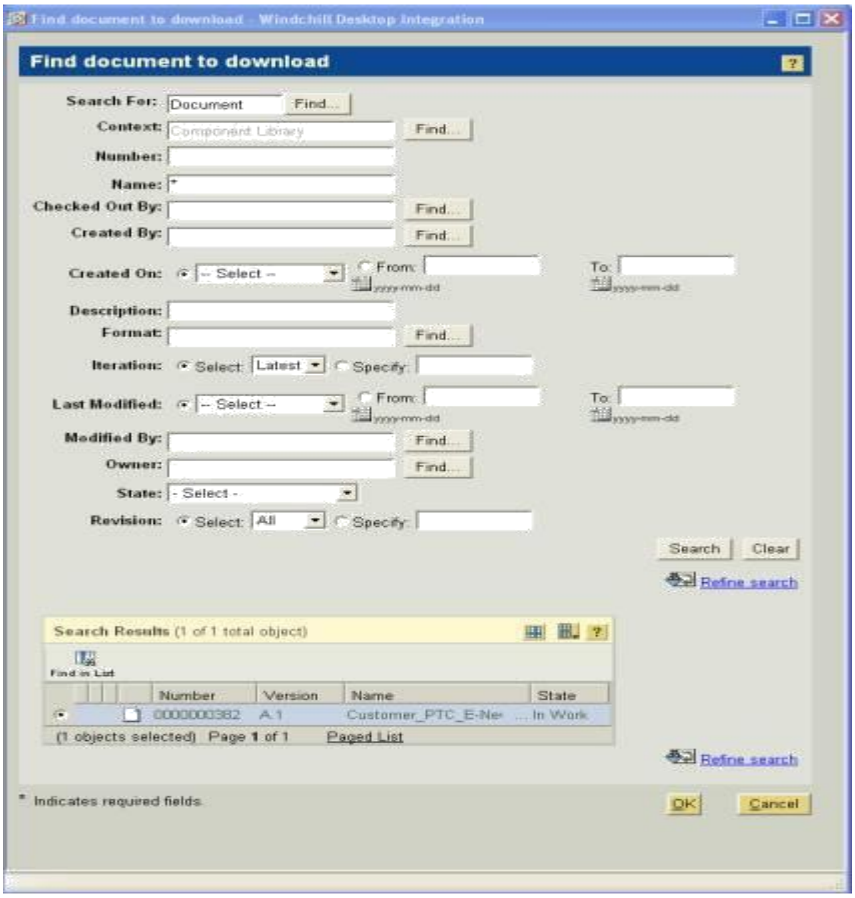
☑ Start project and send invitations to team

Execution Control: ● Manually control project execution

○ Automatically execute activities and deliver tasks

☐ Link deliverables with project activities

* Indicates required fields.



Find document to download

Search For: Document

Context: Component Library

Number:

Name: *

Checked Out By:

Created By:

Created On: ● -- Select -- ○ From: yyyy-mm-dd To: yyyy-mm-dd

Description:

Format:

Iteration: ● Select: Latest ○ Specify:

Last Modified: ● -- Select -- ○ From: yyyy-mm-dd To: yyyy-mm-dd

Modified By:

Owner:

State: - Select -

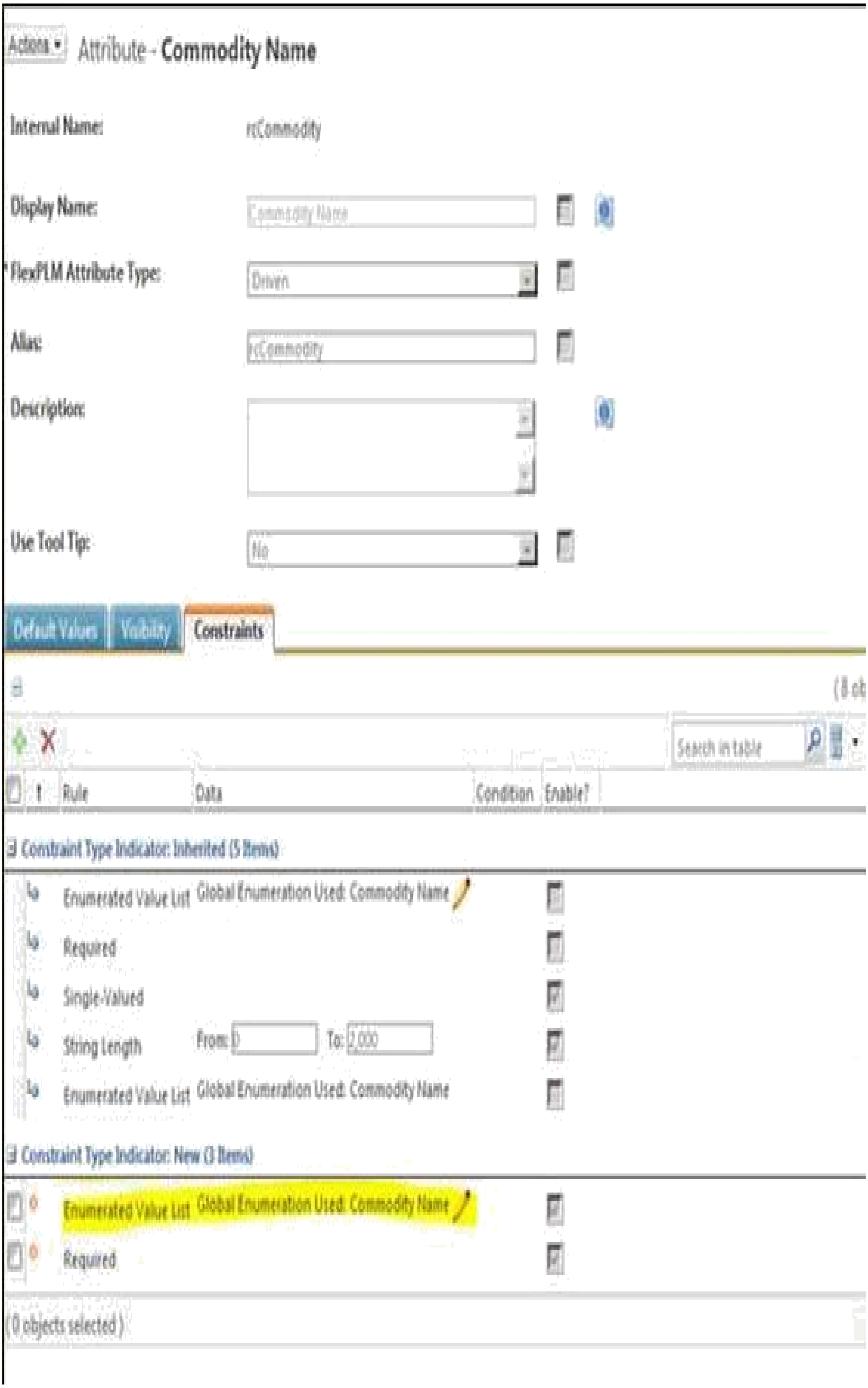
Revision: ● Select: All ○ Specify:

Search Results (1 of 1 total object)

	Number	Version	Name	State
●	0000000382	A.1	Customer_PTC_E-Ne ...	In Work

(1 objects selected) Page 1 of 1 Paged List

* Indicates required fields.



Actions ▾ Attribute - **Commodity Name**

Internal Name: rcCommodity

Display Name: Commodity Name

* FlexPLM Attribute Type: Driven

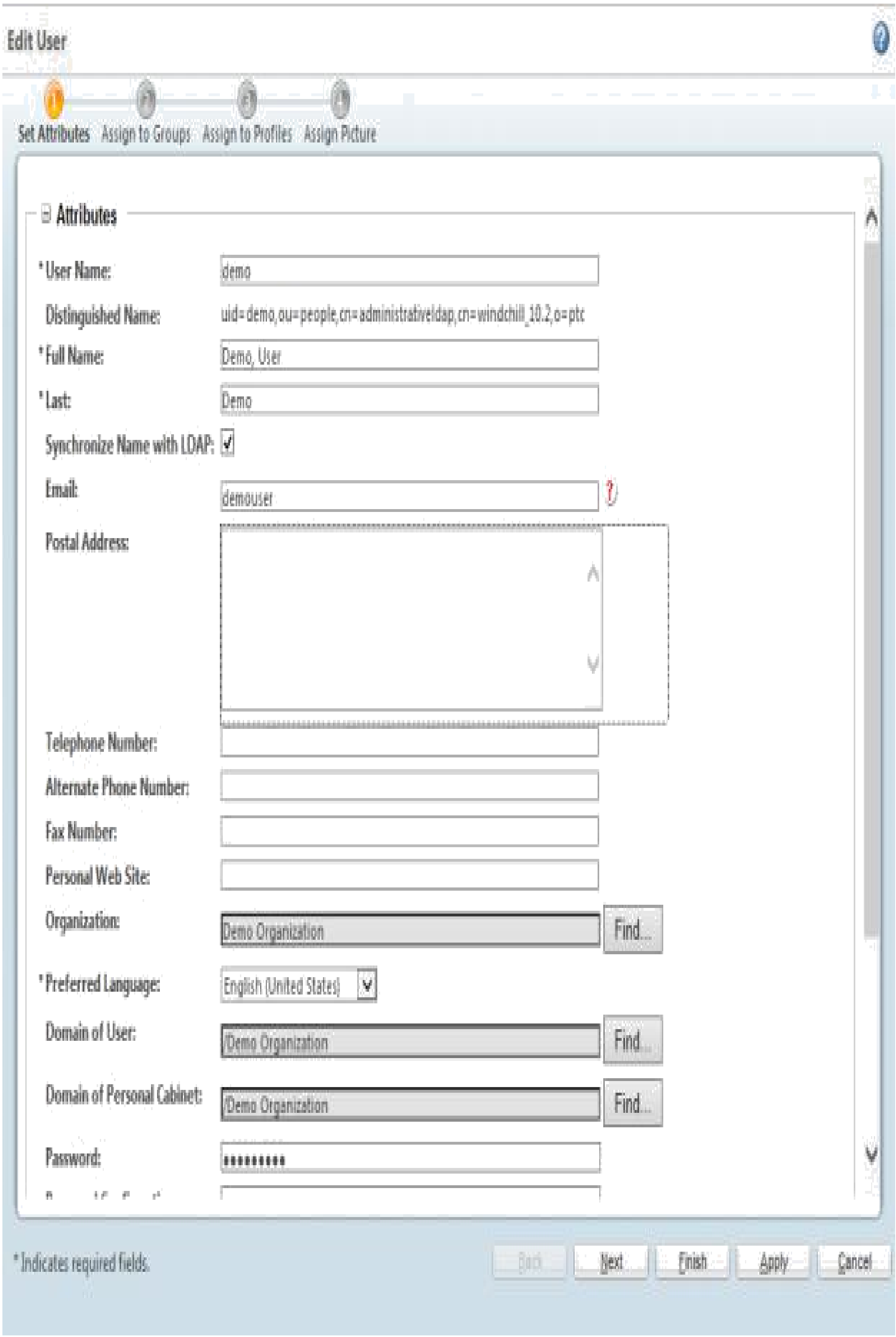
Alias: rcCommodity

Description:

Use Tool Tip: No

□	Rule	Data	Condition	Enable?
⊟ Constraint Type Indicator: Inherited (5 Items)				
	Enumerated Value List	Global Enumeration Used: Commodity Name		☐
	Required			☐
	Single-Valued			☑
	String Length	From: 0 To: 2,000		☑
	Enumerated Value List	Global Enumeration Used: Commodity Name		☐
⊟ Constraint Type Indicator: New (3 Items)				
□	Enumerated Value List	Global Enumeration Used: Commodity Name		☑
□	Required			☑

(0 objects selected)

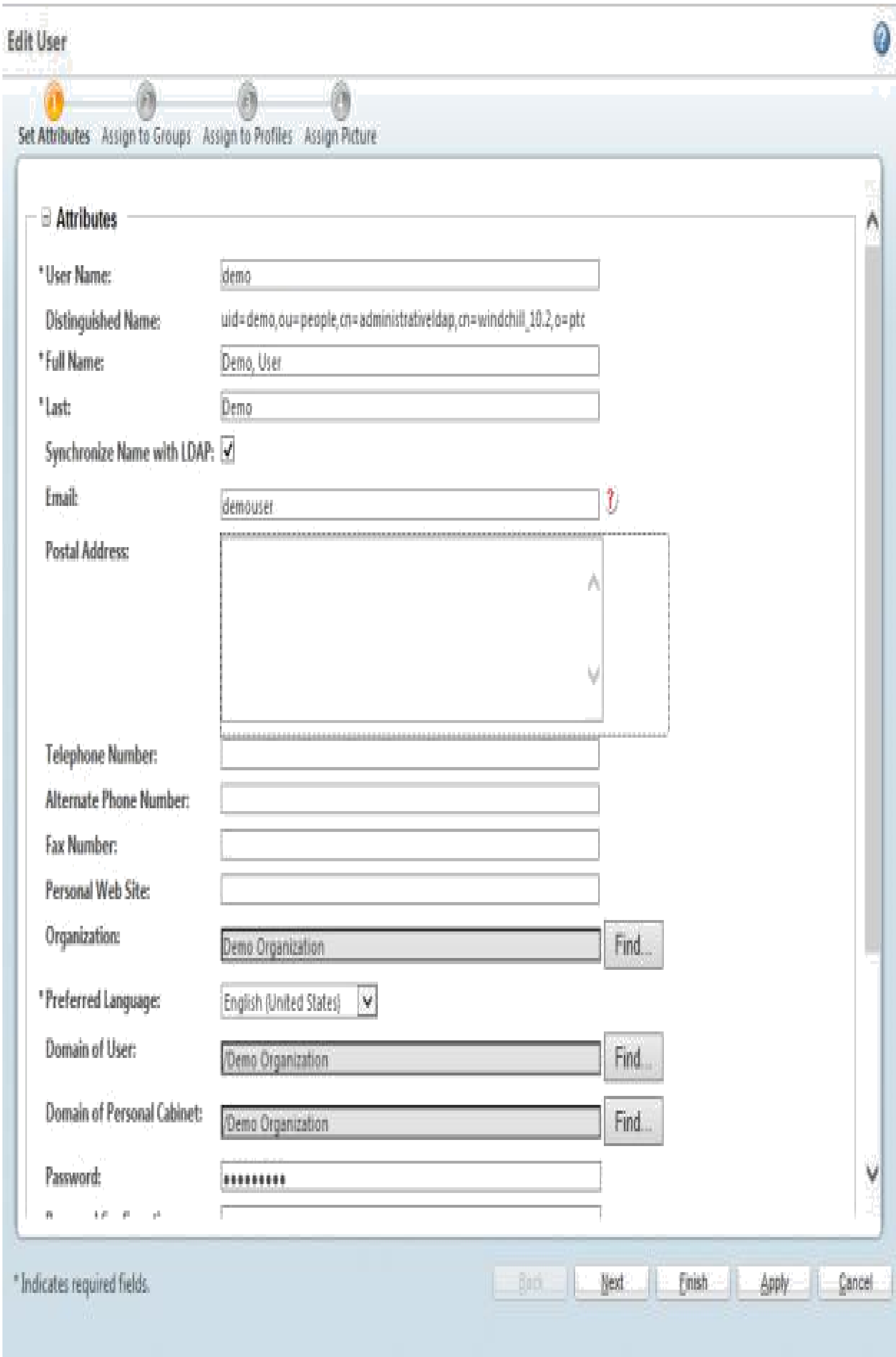


Edit User

⊟ Attributes

* User Name:	demo
Distinguished Name:	uid=demo,ou=people,cn=administrativeldap,cn=windchill_10.2,o=ptc
* Full Name:	Demo, User
* Last:	Demo
Synchronize Name with LDAP:	☑
Email:	demouser
Postal Address:	
Telephone Number:	
Alternate Phone Number:	
Fax Number:	
Personal Web Site:	
Organization:	Demo Organization
* Preferred Language:	English (United States)
Domain of User:	/Demo Organization
Domain of Personal Cabinet:	/Demo Organization
Password:	●●●●●●●●●

* Indicates required fields.

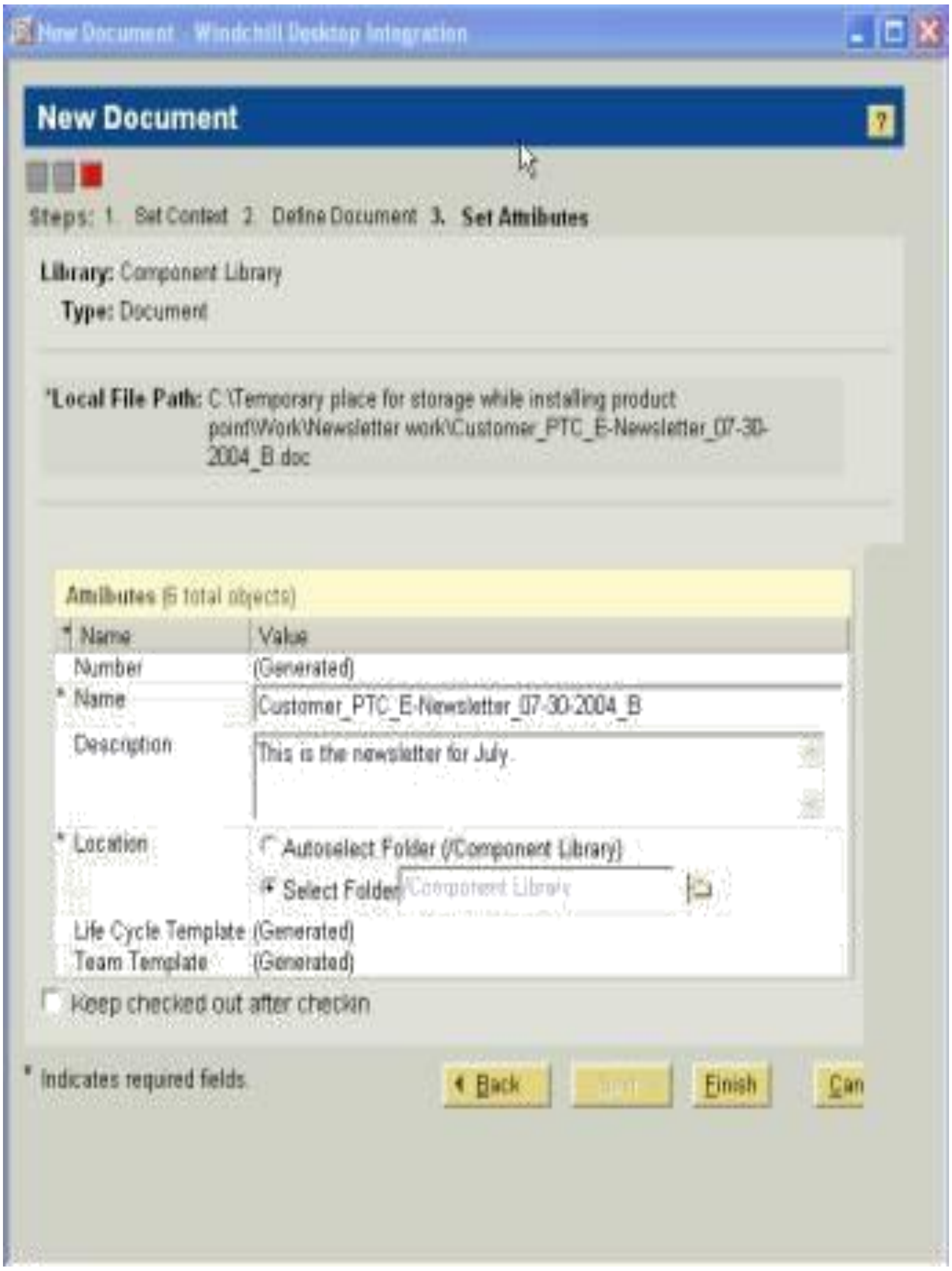


Edit User

Attributes

*User Name:	demo	
Distinguished Name:	uid=demo,ou=people,cn=administrativeldap,cn=windchill_10.2,o=ptc	
*Full Name:	Demo, User	
*Last:	Demo	
Synchronize Name with LDAP:	☑	
Email:	demouser	
Postal Address:		
Telephone Number:		
Alternate Phone Number:		
Fax Number:		
Personal Web Site:		
Organization:	Demo Organization	Find...
*Preferred Language:	English (United States)	
Domain of User:	/Demo Organization	Find...
Domain of Personal Cabinet:	/Demo Organization	Find...
Password:	•••••••••	

*Indicates required fields.



New Document

Steps: 1. Set Context 2. Define Document **3. Set Attributes**

Library: Component Library

Type: Document

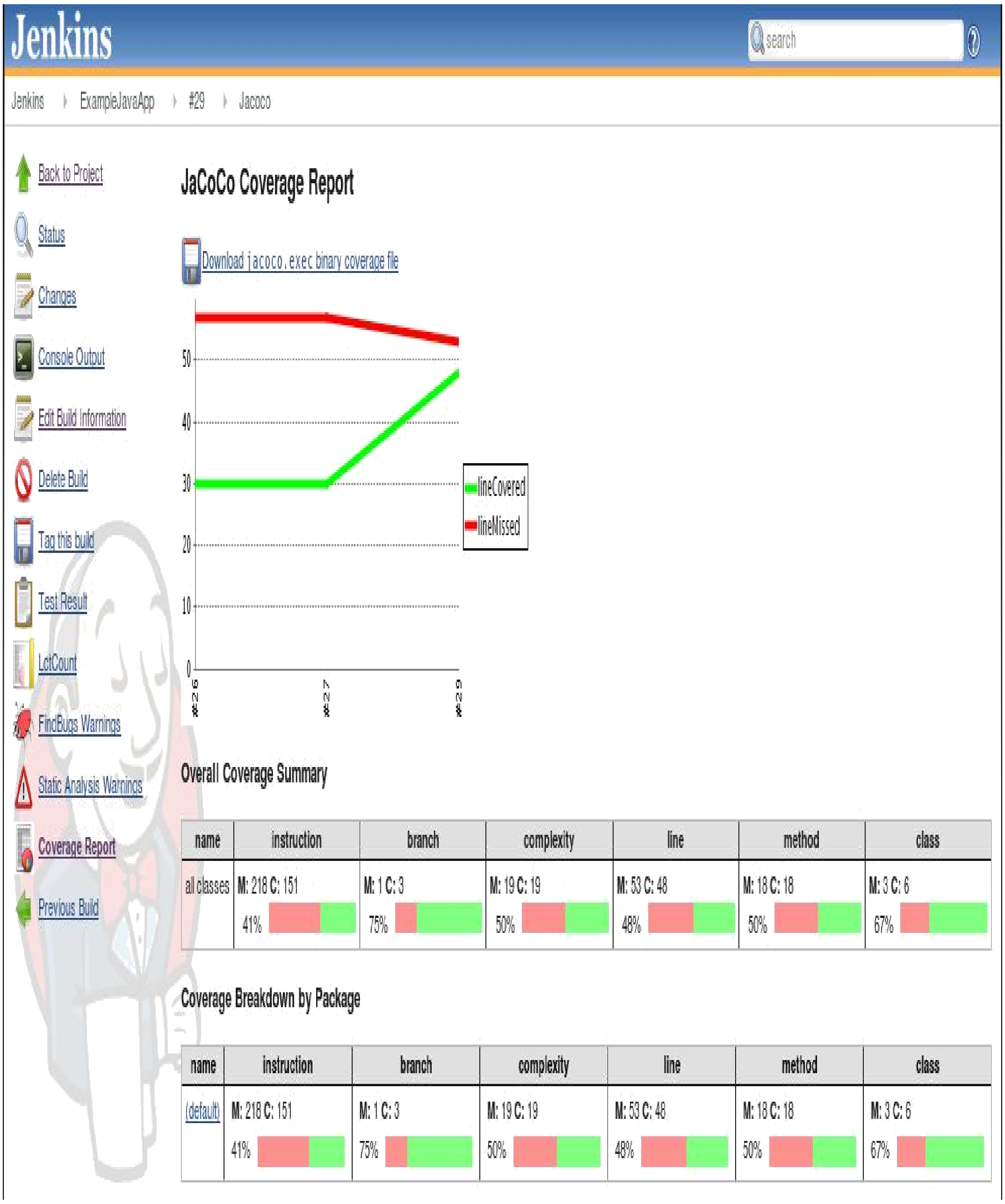
***Local File Path:** C:\Temporary place for storage while installing product point\Work\Newsletter work\Customer_PTC_E-Newsletter_07-30-2004_B.doc

Attributes (6 total objects)

* Name	Value
Number	(Generated)
* Name	Customer_PTC_E-Newsletter_07-30-2004_B
Description	This is the newsletter for July.
* Location	○ Autoselect Folder (/Component Library) ● Select Folder /Component Library
Life Cycle Template	(Generated)
Team Template	(Generated)

□ Keep checked out after checkin

* Indicates required fields.



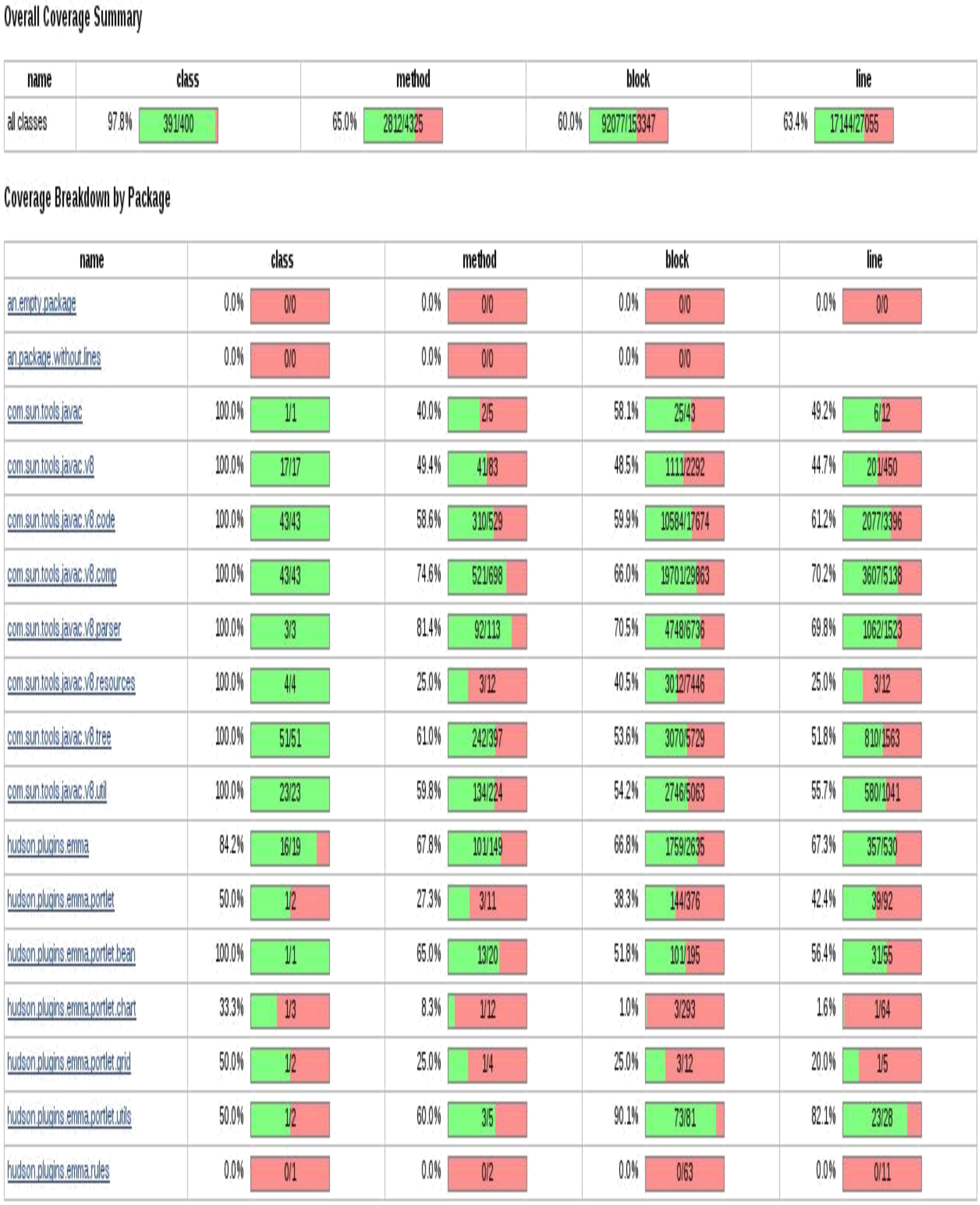
JaCoCo Coverage Report

Overall Coverage Summary

name	instruction	branch	complexity	line	method	class
all classes	M: 218 C: 151 41%	M: 1 C: 3 75%	M: 19 C: 19 50%	M: 53 C: 48 48%	M: 18 C: 18 50%	M: 3 C: 6 67%

Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	class
(default)	M: 218 C: 151 41%	M: 1 C: 3 75%	M: 19 C: 19 50%	M: 53 C: 48 48%	M: 18 C: 18 50%	M: 3 C: 6 67%

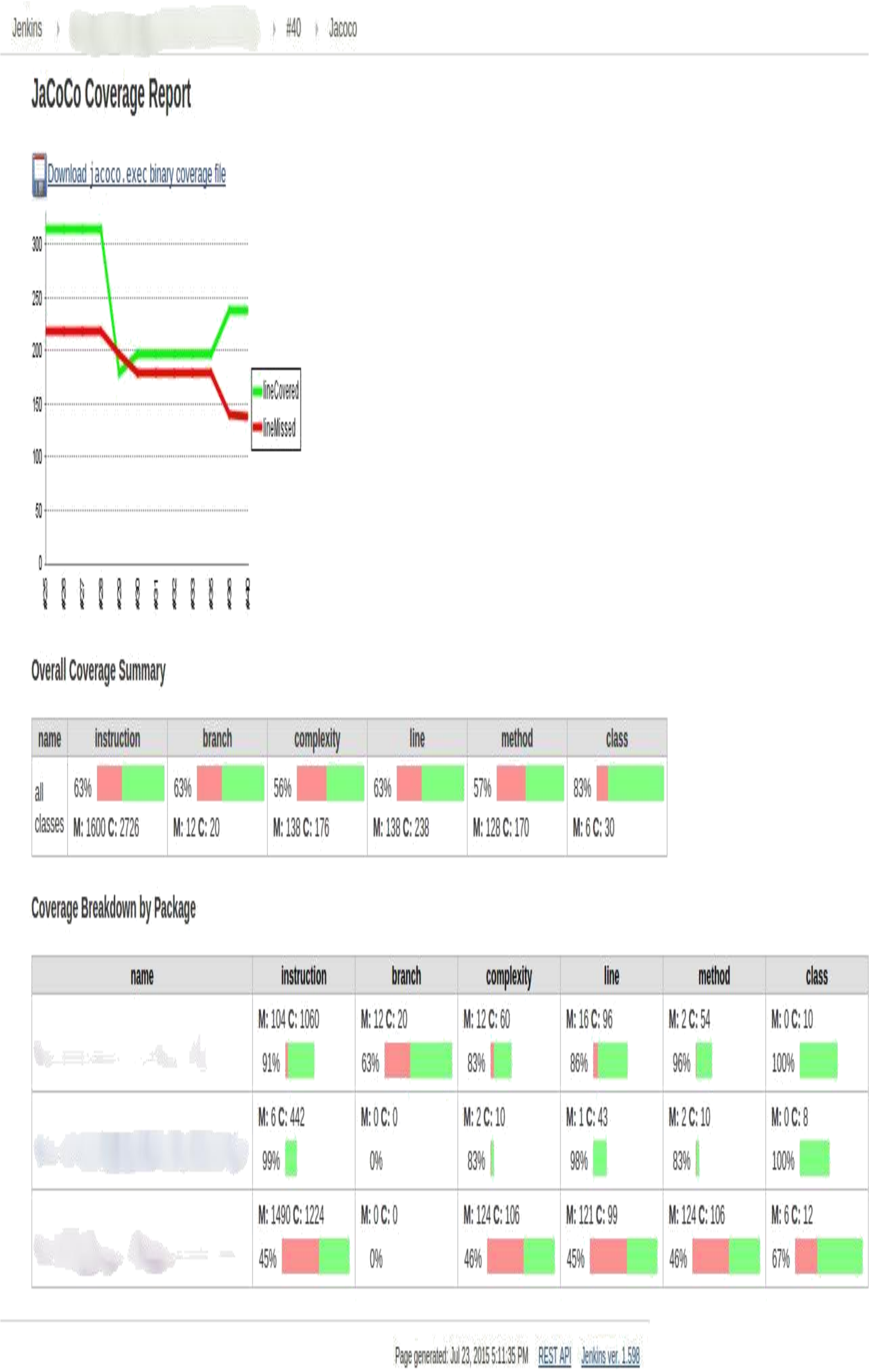


Overall Coverage Summary

name	class		method		block		line	
all classes	97.8%	391/400	65.0%	2812/4325	60.0%	92077/153347	63.4%	17144/27055

Coverage Breakdown by Package

name	class		method		block		line	
an.empty.package	0.0%	0/0	0.0%	0/0	0.0%	0/0	0.0%	0/0
an.package.without.lines	0.0%	0/0	0.0%	0/0	0.0%	0/0		
com.sun.tools.javac	100.0%	1/1	40.0%	2/5	58.1%	25/43	49.2%	6/12
com.sun.tools.javac.v8	100.0%	17/17	49.4%	41/83	48.5%	1111/2292	44.7%	201/450
com.sun.tools.javac.v8.code	100.0%	43/43	58.6%	310/529	59.9%	10584/17674	61.2%	2077/3396
com.sun.tools.javac.v8.comp	100.0%	43/43	74.6%	521/698	66.0%	19701/29863	70.2%	3607/5138
com.sun.tools.javac.v8.parser	100.0%	3/3	81.4%	92/113	70.5%	4748/6736	69.8%	1062/1523
com.sun.tools.javac.v8.resources	100.0%	4/4	25.0%	3/12	40.5%	3012/7446	25.0%	3/12
com.sun.tools.javac.v8.tree	100.0%	51/51	61.0%	242/397	53.6%	3070/5729	51.8%	810/1563
com.sun.tools.javac.v8.util	100.0%	23/23	59.8%	134/224	54.2%	2746/5063	55.7%	580/1041
hudson.plugins.emma	84.2%	16/19	67.8%	101/149	66.8%	1759/2635	67.3%	357/530
hudson.plugins.emma.portlet	50.0%	1/2	27.3%	3/11	38.3%	144/376	42.4%	39/92
hudson.plugins.emma.portlet.bean	100.0%	1/1	65.0%	13/20	51.8%	101/195	56.4%	31/55
hudson.plugins.emma.portlet.chart	33.3%	1/3	8.3%	1/12	1.0%	3/293	1.6%	1/64
hudson.plugins.emma.portlet.grid	50.0%	1/2	25.0%	1/4	25.0%	3/12	20.0%	1/5
hudson.plugins.emma.portlet.utils	50.0%	1/2	60.0%	3/5	90.1%	73/81	82.1%	23/28
hudson.plugins.emma.rules	0.0%	0/1	0.0%	0/2	0.0%	0/63	0.0%	0/11



Jenkins › [redacted] › #40 › Jacoco

JaCoCo Coverage Report

Download `jacoco.exec` binary coverage file

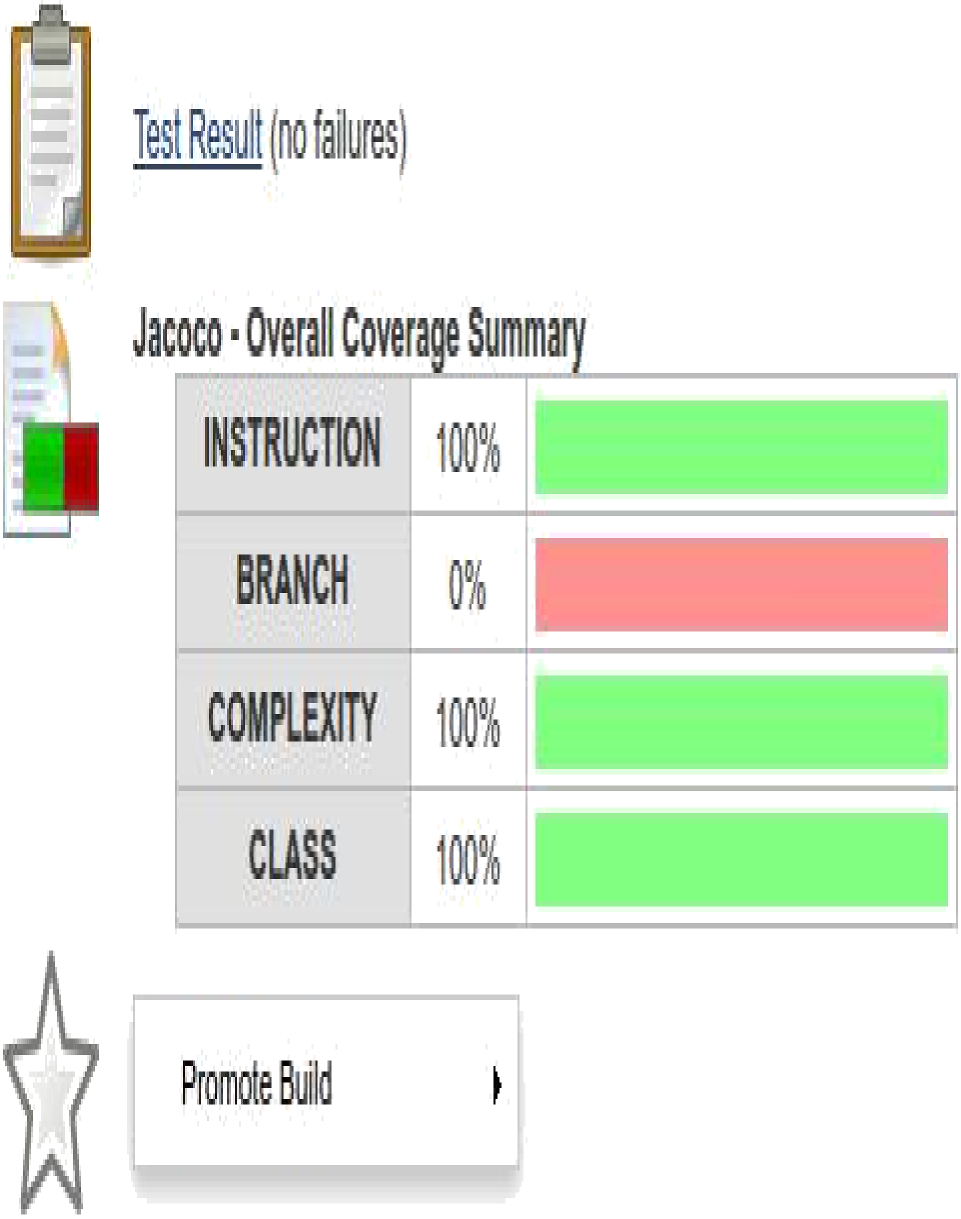
Overall Coverage Summary

name	instruction	branch	complexity	line	method	class
all classes	63% **M:** 1600 **C:** 2726	63% **M:** 12 **C:** 20	56% **M:** 138 **C:** 176	63% **M:** 138 **C:** 238	57% **M:** 128 **C:** 170	83% **M:** 6 **C:** 30

Coverage Breakdown by Package

name	instruction	branch	complexity	line	method	class
[redacted]	**M:** 104 **C:** 1060 91%	**M:** 12 **C:** 20 63%	**M:** 12 **C:** 60 83%	**M:** 16 **C:** 96 86%	**M:** 2 **C:** 54 96%	**M:** 0 **C:** 10 100%
[redacted]	**M:** 6 **C:** 442 99%	**M:** 0 **C:** 0 0%	**M:** 2 **C:** 10 83%	**M:** 1 **C:** 43 98%	**M:** 2 **C:** 10 83%	**M:** 0 **C:** 8 100%
[redacted]	**M:** 1490 **C:** 1224 45%	**M:** 0 **C:** 0 0%	**M:** 124 **C:** 106 46%	**M:** 121 **C:** 99 45%	**M:** 124 **C:** 106 46%	**M:** 6 **C:** 12 67%

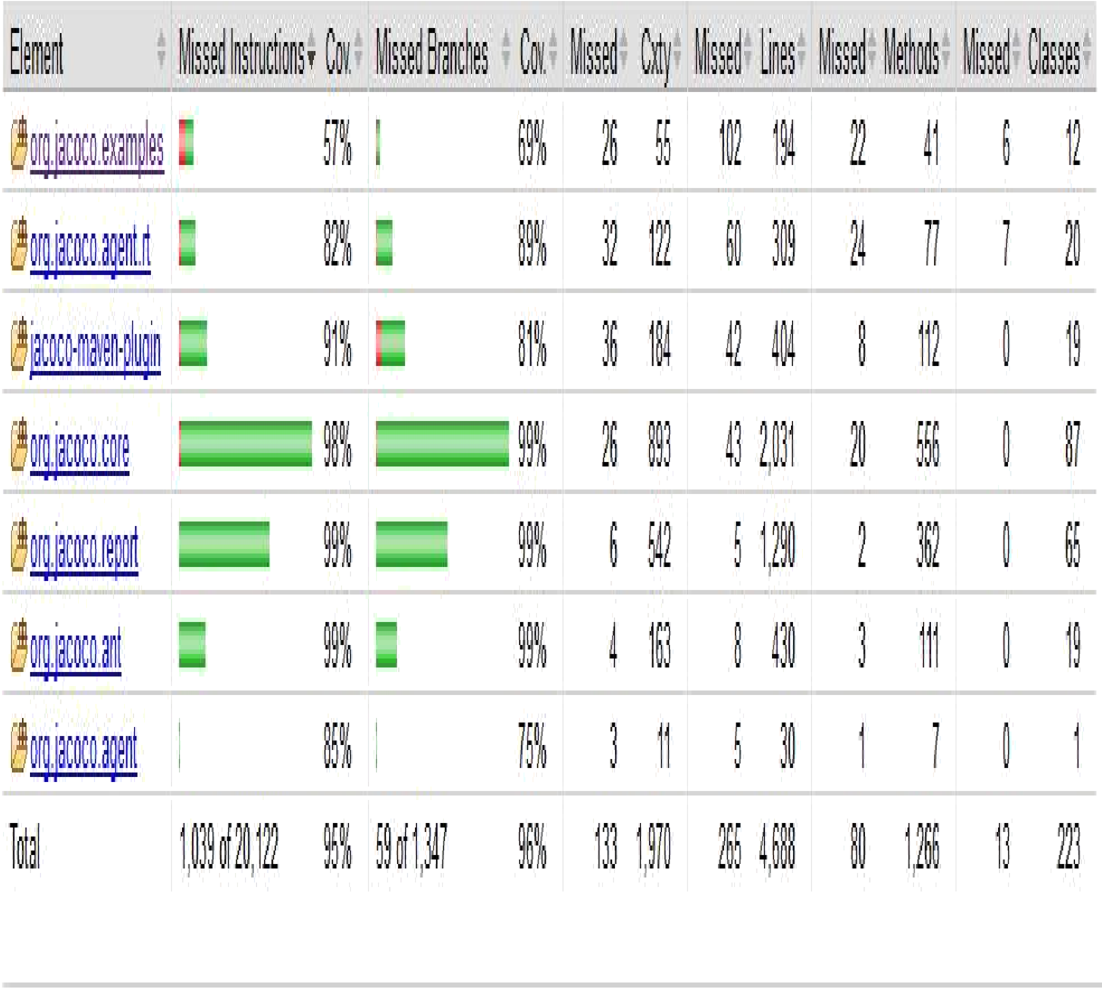
Page generated: Jul 23, 2015 5:11:35 PM REST API Jenkins ver. 1.598



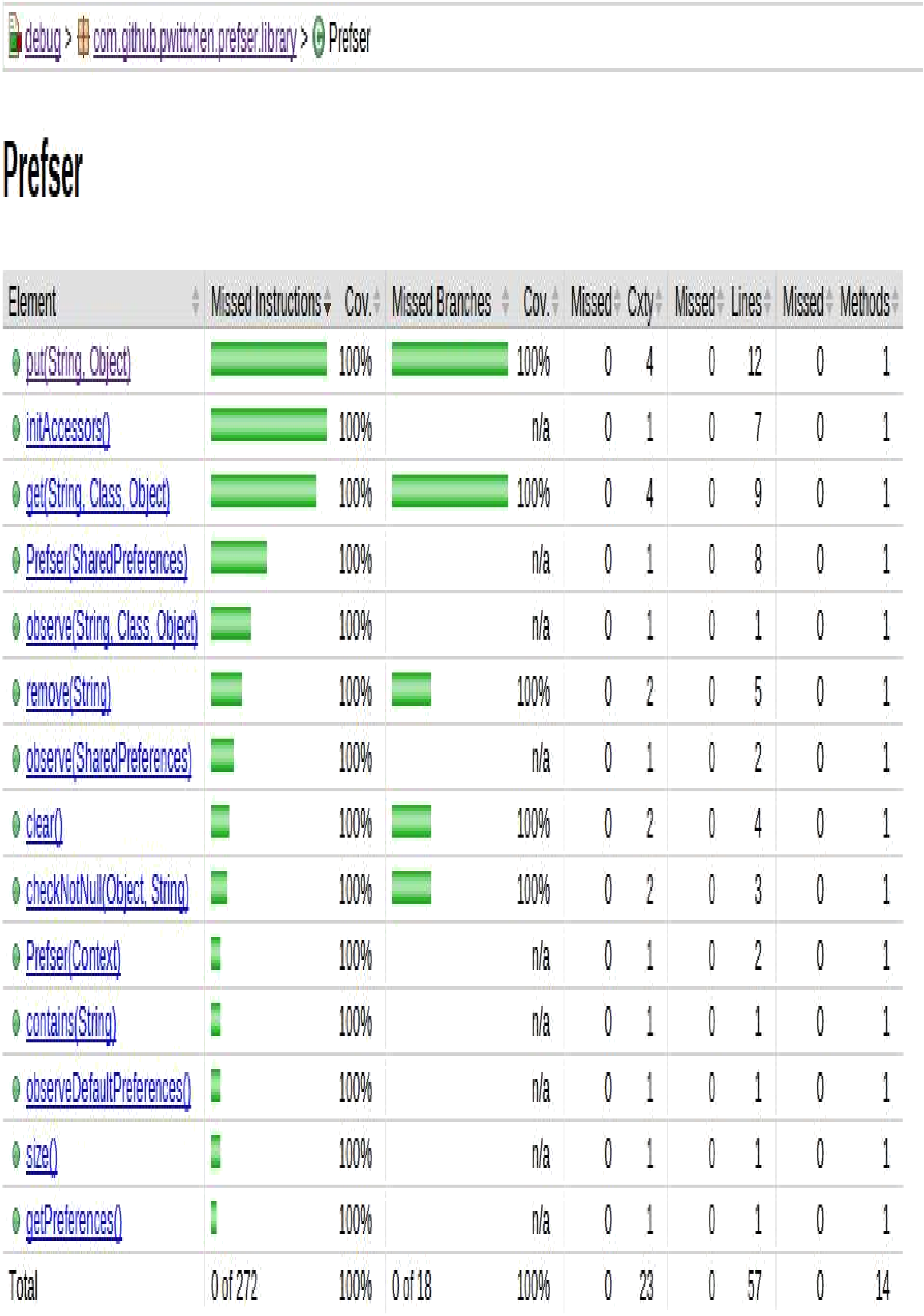
Test Result (no failures)

Jacoco - Overall Coverage Summary

INSTRUCTION	100%	🟩
BRANCH	0%	🟥
COMPLEXITY	100%	🟩
CLASS	100%	🟩



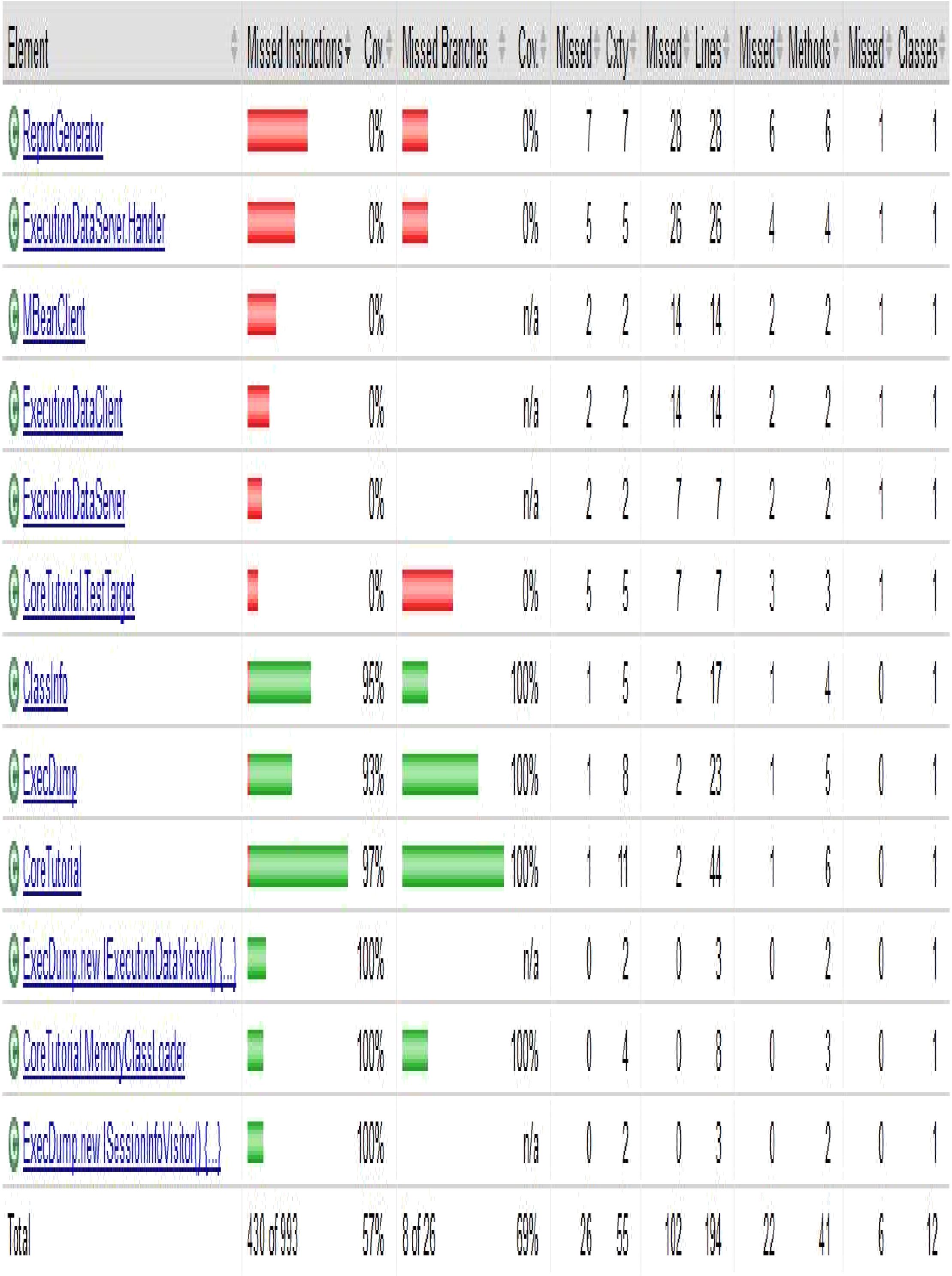
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
org.jacoco.examples		57%		69%	26	55	102	194	22	41	6	12
org.jacoco.agent.rt		82%		89%	32	122	60	309	24	77	7	20
jacoco-maven-plugin		91%		81%	36	184	42	404	8	112	0	19
org.jacoco.core		98%		99%	26	893	43	2,031	20	556	0	87
org.jacoco.report		99%		99%	6	542	5	1,290	2	362	0	65
org.jacoco.ant		99%		99%	4	163	8	430	3	111	0	19
org.jacoco.agent		85%		75%	3	11	5	30	1	7	0	1
Total	1,039 of 20,122	95%	59 of 1,347	96%	133	1,970	265	4,688	80	1,266	13	223



debug > com.github.pwittchen.prefser.library > Prefser

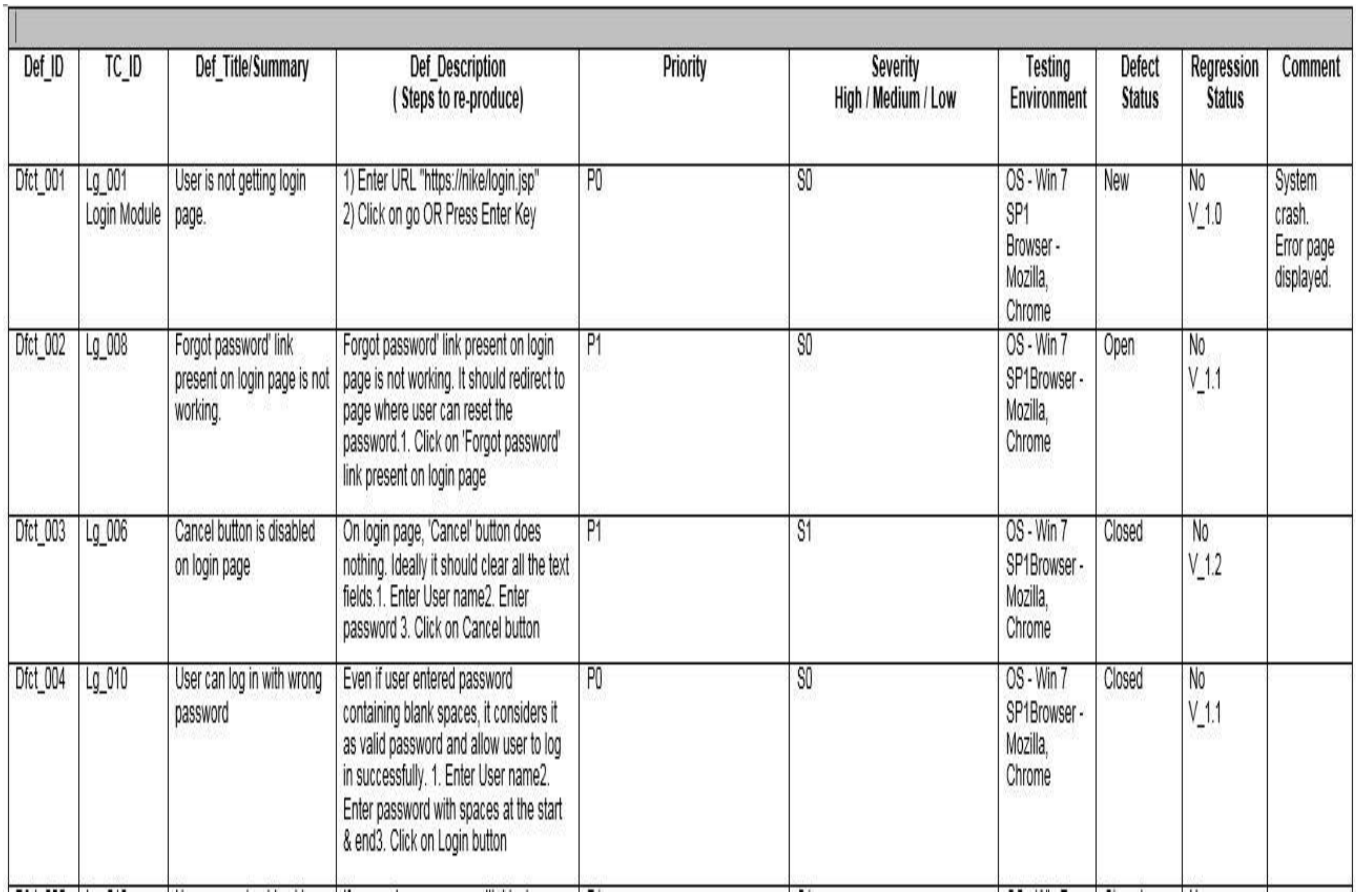
Prefser

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
put(String, Object)	🟩	100%	🟩	100%	0	4	0	12	0	1
initAccessors()	🟩	100%		n/a	0	1	0	7	0	1
get(String, Class, Object)	🟩	100%	🟩	100%	0	4	0	9	0	1
Prefser(SharedPreferences)	🟩	100%		n/a	0	1	0	8	0	1
observe(String, Class, Object)	🟩	100%		n/a	0	1	0	1	0	1
remove(String)	🟩	100%	🟩	100%	0	2	0	5	0	1
observe(SharedPreferences)	🟩	100%		n/a	0	1	0	2	0	1
clear()	🟩	100%	🟩	100%	0	2	0	4	0	1
checkNotNull(Object, String)	🟩	100%	🟩	100%	0	2	0	3	0	1
Prefser(Context)	🟩	100%		n/a	0	1	0	2	0	1
contains(String)	🟩	100%		n/a	0	1	0	1	0	1
observeDefaultPreferences()	🟩	100%		n/a	0	1	0	1	0	1
size()	🟩	100%		n/a	0	1	0	1	0	1
getPreferences()	🟩	100%		n/a	0	1	0	1	0	1
Total	0 of 272	100%	0 of 18	100%	0	23	0	57	0	14



Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
ReportGenerator	🟥	0%	🟥	0%	7	7	28	28	6	6	1	1
ExecutionDataServer.Handler	🟥	0%	🟥	0%	5	5	26	26	4	4	1	1
MBeanClient	🟥	0%		n/a	2	2	14	14	2	2	1	1
ExecutionDataClient	🟥	0%		n/a	2	2	14	14	2	2	1	1
ExecutionDataServer	🟥	0%		n/a	2	2	7	7	2	2	1	1
CoreTutorial.TestTarget	🟥	0%	🟥	0%	5	5	7	7	3	3	1	1
ClassInfo	🟥🟩	95%	🟩	100%	1	5	2	17	1	4	0	1
ExecDump	🟥🟩	93%	🟩	100%	1	8	2	23	1	5	0	1
CoreTutorial	🟥🟩	97%	🟩	100%	1	11	2	44	1	6	0	1
ExecDump.new IExecutionDataVisitor() {...}	🟩	100%		n/a	0	2	0	3	0	2	0	1
CoreTutorial.MemoryClassLoader	🟩	100%	🟩	100%	0	4	0	8	0	3	0	1
ExecDump.new ISessionInfoVisitor() {...}	🟩	100%		n/a	0	2	0	3	0	2	0	1
Total	430 of 993	57%	8 of 26	69%	26	55	102	194	22	41	6	12

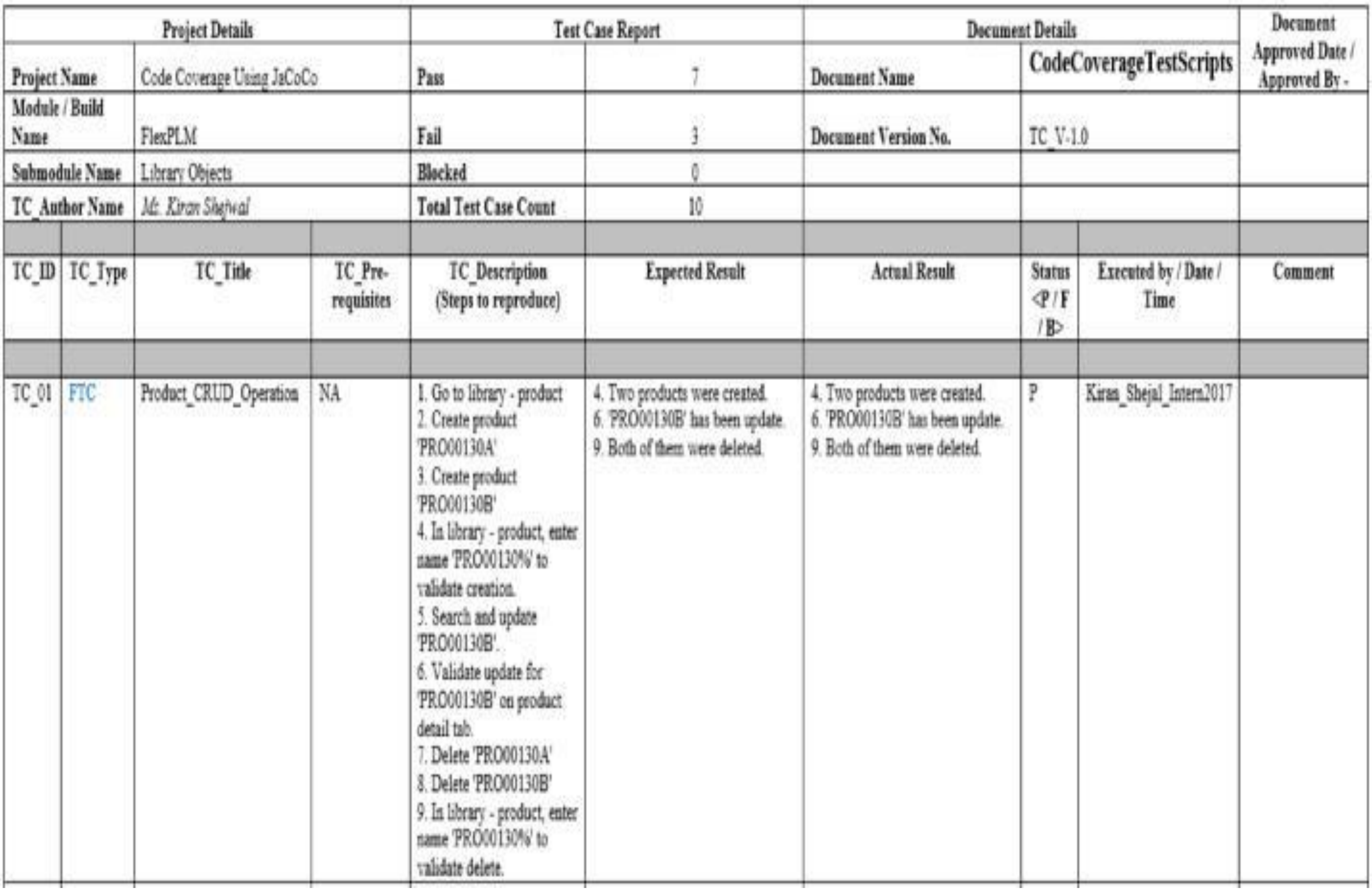
Annexure 2: Defect Report



Def_ID	TC_ID	Def_Title/Summary	Def_Description (Steps to re-produce)	Priority	Severity High / Medium / Low	Testing Environment	Defect Status	Regression Status	Comment
Dfct_001	Lg_001 Login Module	User is not getting login page.	1) Enter URL "https://nike/login.jsp" 2) Click on go OR Press Enter Key	P0	S0	OS - Win 7 SP1 Browser - Mozilla, Chrome	New	No V_1.0	System crash. Error page displayed.
Dfct_002	Lg_008	Forgot password' link present on login page is not working.	Forgot password' link present on login page is not working. It should redirect to page where user can reset the password.1. Click on 'Forgot password' link present on login page	P1	S0	OS - Win 7 SP1Browser - Mozilla, Chrome	Open	No V_1.1	
Dfct_003	Lg_006	Cancel button is disabled on login page	On login page, 'Cancel' button does nothing. Ideally it should clear all the text fields.1. Enter User name2. Enter password 3. Click on Cancel button	P1	S1	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.2	
Dfct_004	Lg_010	User can log in with wrong password	Even if user entered password containing blank spaces, it considers it as valid password and allow user to log in successfully. 1. Enter User name2. Enter password with spaces at the start & end3. Click on Login button	P0	S0	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.1	

Dfct_005	Lg_012	User name should not be accepted if entered with spaces at beginning or at end	If user enter user name with blank spaces at beginning or at the end, it should prompt error message. Currently it accepts such user name and allows user to log in.1. Enter User name with spaces at the start & end2. Enter password 3. Click on Login button	P1	S1	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.2	
Dfct_006	Lg_07	After clicking on create button season should get created successfully	1. Go to library - Season 2. Create season 'MensApparelBrand 1Division 1Autumn2006' 3. Create season 'MensApparelBrand 1Division 1Autumn2007'	P1	S1	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.2	
Dfct_007	Lg_05	After clicking on create button Product should get created successfully	1. Go to library - product 2. Create product 'PRO00130A'	P2	S2	OS - Win 7 SP1Browser - Mozilla, Chrome	Open	No V_1.3	
Dfct_008	Lg_06	After clicking on create button Material should get created successfully	1. Go to library - Material 2. Create Material 'Mat00130A'	P3	S3	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.3	
Dfct_009	Lg_07	After clicking on create button BOM should get created successfully	1. Go to library - BOM 2. Create BOM 'BOM00130A'	P4	S4	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.2	
Dfct_010	Lg_08	After clicking on create button Specification should get created successfully	1. Go to library - Specification 2. Create Specification 'Specification00130A'	P5	S5	OS - Win 7 SP1Browser - Mozilla, Chrome	Closed	No V_1.1	

Annexure 3: All test cases

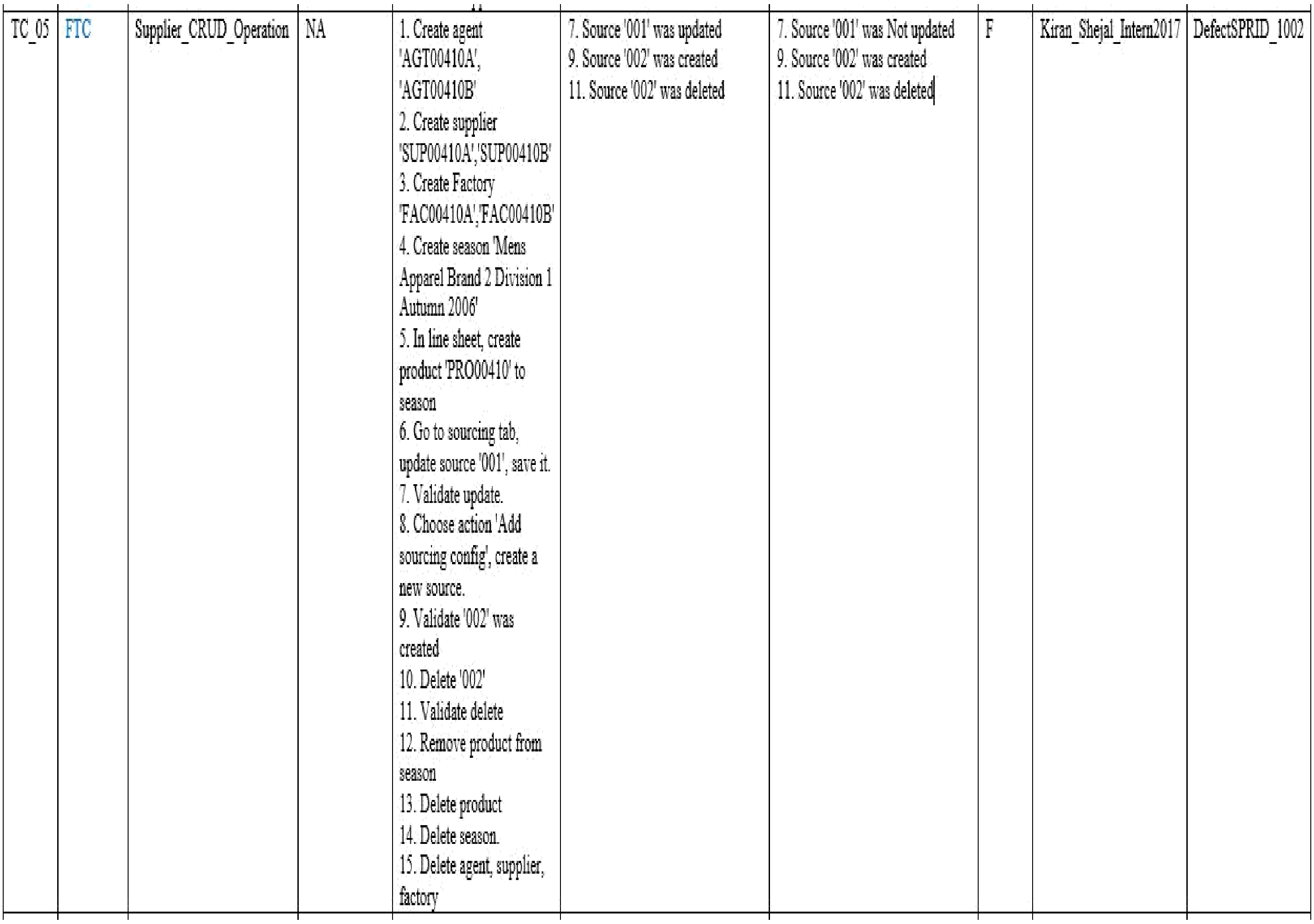


Project Details				Test Case Report		Document Details			Document Approved Date / Approved By -
Project Name		Code Coverage Using JaCoCo		Pass	7	Document Name	CodeCoverageTestScripts		
Module / Build Name		FlexPLM		Fail	3	Document Version No.	TC_V-1.0		
Submodule Name		Library Objects		Blocked	0				
TC_Author Name		*Mr. Kiran Shejwal*		Total Test Case Count	10				
TC_ID	**TC_Type**	**TC_Title**	**TC_Pre-requisites**	**TC_Description (Steps to reproduce)**	**Expected Result**	**Actual Result**	**Status <P / F / B>**	**Executed by / Date / Time**	**Comment**
TC_01	FTC	Product_CRUD_Operation	NA	1. Go to library - product 2. Create product 'PRO00130A' 3. Create product 'PRO00130B' 4. In library - product, enter name 'PRO00130%' to validate creation. 5. Search and update 'PRO00130B'. 6. Validate update for 'PRO00130B' on product detail tab. 7. Delete 'PRO00130A' 8. Delete 'PRO00130B' 9. In library - product, enter name 'PRO00130%' to validate delete.	4. Two products were created. 6. 'PRO00130B' has been update. 9. Both of them were deleted.	4. Two products were created. 6. 'PRO00130B' has been update. 9. Both of them were deleted.	P	Kiran_Shejal_Intern2017	

TC_02	FTC	Season_CRUD_Operation	NA	1. Go to library - Season 2. Create season 'MensApparelBrand 1Division 1Autumn2006' 3. Create season 'MensApparelBrand 1Division 1Autumn2007' 4. In library, enter season name 'MensApparelBrand 1Division 1Autumn%', validate creation 5. Update 'MensApparelBrand 1Division 1Autumn2007' to 'MensApparelBrand 1Division 2Autumn2008' 6. Validate update in season detail page 7. Delete 'MensApparelBrand 1Division 1Autumn2006' 8. Delete 'MensApparelBrand 1Division 2Autumn2008' 9. Validate delete	4. Two seasons were searched out 6. Season detail has been updated. 9. Seasons were deleted.	4. Two seasons were searched out 6. Season detail has been updated. 9. Seasons were NOT deleted.	F	Kiran_Shejal_Intern2017	DefectSPRID_1001

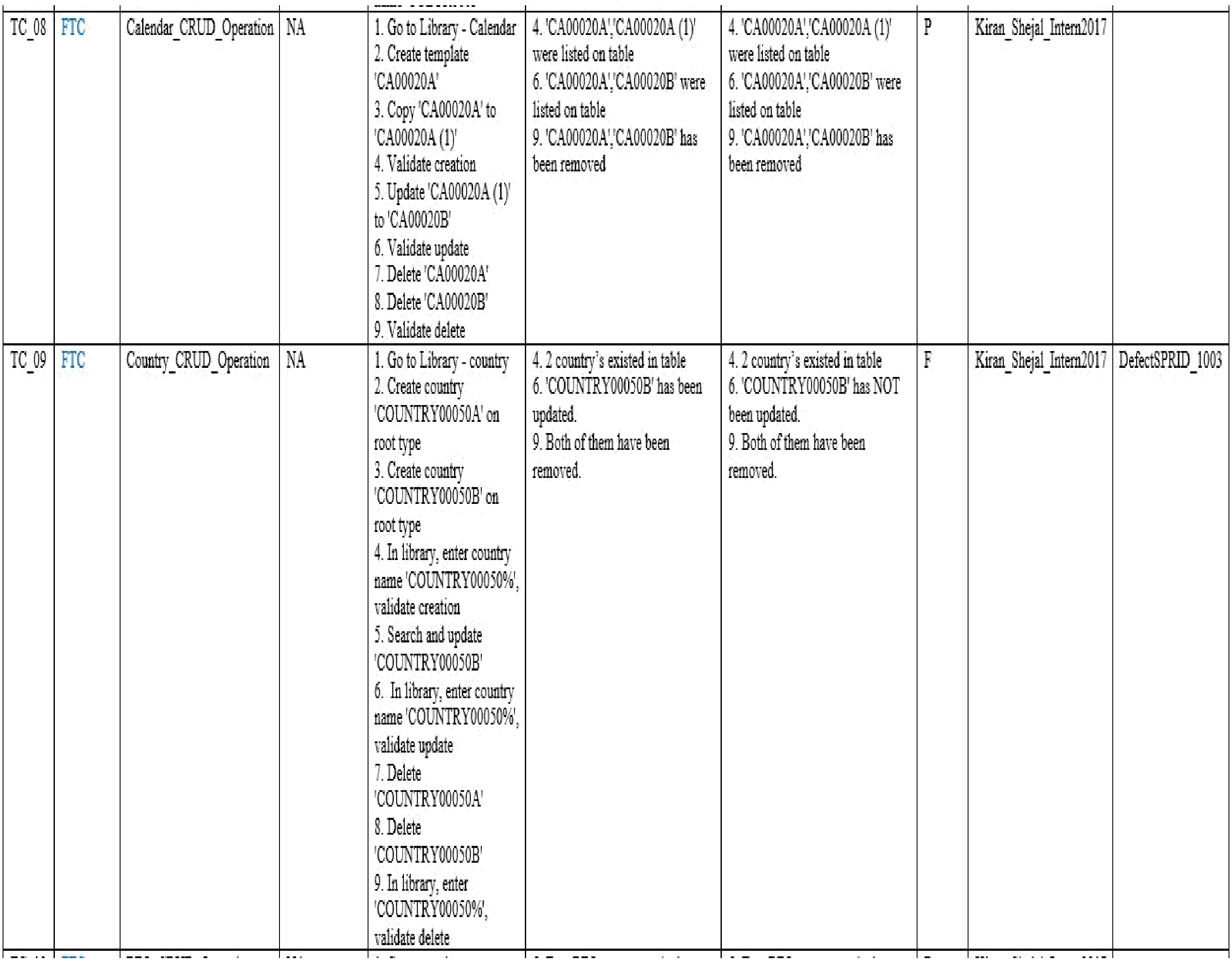
TC_03	FTC	BOM_CRUD_Operation	NA	1. Create material 'MAT00340' 2. Create season 'Mens Apparel Brand 1 Division 1 Autumn 2021' 3. In line sheet, create product 'PRO00340' to season 4. Go to spec tab, create spec '001' 5. Go to Bom tab, create a BOM 'BOM00340' 6. In Shell section, choose the first row 7. click materials button to open mateiral panel. Search out material 'MAT00340', add it to BOM. 8. Save and check in bom. 9. Validate BOM. 10. Remove BOM 11. Remove the product from season 12. Delete product 13. Delete season	9. Validate BOM table under BOM page, 'MAT00340' should be listed in BOM	9. Validate BOM table under BOM page, 'MAT00340' should be listed in BOM	P	Kiran_Shejal_Intern2017	

TC_04	FTC	RFQ_CRUD_Operation	NA	1. Create season 'Mens Apparel Brand 2 Division 1 Autumn 2008' 2. In line sheet, create product 'PRO00430' to season 3. Update source to add a supplier (if no supplier, create a supplier 'SUPP00430'). 4. Go to spec tab, create a spec. 5. Go to RFQ tab, choose action 'Create RFQ' 6. Create 'RFQ00430', Done. 7. Under RFQ tab, validate create. 8. Update this RFQ 9. Validate update on RFQ tab. 10. Delete this RFQ 11. Validate delete 12. Remove product from season 13. Delete product 14. Delete season 15. Delete supplier	7. RFQ was created 9. RFQ was updated 11. RFQ was deleted	7. RFQ was created 9. RFQ was updated 11. RFQ was deleted	P	Kiran_Shejal_Intern2017	



TC_05	FTC	Supplier_CRUD_Operation	NA	1. Create agent 'AGT00410A', 'AGT00410B' 2. Create supplier 'SUP00410A','SUP00410B' 3. Create Factory 'FAC00410A','FAC00410B' 4. Create season 'Mens Apparel Brand 2 Division 1 Autumn 2006' 5. In line sheet, create product 'PRO00410' to season 6. Go to sourcing tab, update source '001', save it. 7. Validate update. 8. Choose action 'Add sourcing config', create a new source. 9. Validate '002' was created 10. Delete '002' 11. Validate delete 12. Remove product from season 13. Delete product 14. Delete season. 15. Delete agent, supplier, factory	7. Source '001' was updated 9. Source '002' was created 11. Source '002' was deleted	7. Source '001' was Not updated 9. Source '002' was created 11. Source '002' was deleted	F	Kiran_Shejal_Intern2017	DefectSPRID_1002

TC_06	FTC	Material_CRUD_Operation	1. Go to library - material 2. Create material 'MAT00080A' on personal care type 3. Create material 'MAT00080B' on personal care type	1. In library, enter material name 'MAT00080%', validate creation 2. Search and update 'MAT00080B' 3. In library, enter material name 'MAT00080%', validate update 4. Delete 'MAT00080A' 5. Delete 'MAT00080B' 6. In library, enter material name 'MAT00080%', validate delete	1. Two materials have been created successfully 3. 'MAT00080B' has been updated 7. Two materials have been removed	4. Two materials have been created successfully 6. 'MAT00080B' has been updated 7. Two materials have been removed	P	Kiran_Shejal_Intern2017	
TC_07	FTC	Color_CRUD_Operation	NA	1. Go to Library - Color 2. Create color 'COLOR00030A' on root type 3. Create color 'COLOR00030B' on root type 4. In library, enter color name 'COLOR00030%', validate creation 5. Search and update 'COLOR00030B' 6. In library, enter color name 'COLOR00030% ',Validate update 7. Delete 'COLOR00030A' 8. Delete 'COLOR00030B' 9. In library, enter color name 'COLOR0003	4. 'COLOR00030A','COLOR00030B' were listed on table 6. In library, validate some fields have been updated. 9. 'COLOR00030A','COLOR00030B' have been removed	4. 'COLOR00030A','COLOR00030B' were listed on table 6. In library, validate some fields have been updated. 9. 'COLOR00030A','COLOR00030B' have been removed	P	Kiran_Shejal_Intern2017	



TC_08	FTC	Calendar_CRUD_Operation	NA	1. Go to Library - Calendar 2. Create template 'CA00020A' 3. Copy 'CA00020A' to 'CA00020A (1)' 4. Validate creation 5. Update 'CA00020A (1)' to 'CA00020B' 6. Validate update 7. Delete 'CA00020A' 8. Delete 'CA00020B' 9. Validate delete	4. 'CA00020A','CA00020A (1)' were listed on table 6. 'CA00020A','CA00020B' were listed on table 9. 'CA00020A','CA00020B' has been removed	4. 'CA00020A','CA00020A (1)' were listed on table 6. 'CA00020A','CA00020B' were listed on table 9. 'CA00020A','CA00020B' has been removed	P	Kiran_Shejal_Intern2017	
TC_09	FTC	Country_CRUD_Operation	NA	1. Go to Library - country 2. Create country 'COUNTRY00050A' on root type 3. Create country 'COUNTRY00050B' on root type 4. In library, enter country name 'COUNTRY00050%', validate creation 5. Search and update 'COUNTRY00050B' 6. In library, enter country name 'COUNTRY00050%', validate update 7. Delete 'COUNTRY00050A' 8. Delete 'COUNTRY00050B' 9. In library, enter 'COUNTRY00050%', validate delete	4. 2 country's existed in table 6. 'COUNTRY00050B' has been updated. 9. Both of them have been removed.	4. 2 country's existed in table 6. 'COUNTRY00050B' has NOT been updated. 9. Both of them have been removed.	F	Kiran_Shejal_Intern2017	DefectSPRID_1003

TC_10	FTC	RFQ_CRUD_Operation	NA	1. Create product 'PRO00140' in library 2. Create Supplier 'SUPP00140' in library 3. Create a RFQ 'RFQ00140A' under in library. 4. Create a RFQ 'RFQ00140B' under in library. 5. Go to library - RFQ, enter name 'RFQ00140%' to validate creation. 6. Delete 'RFQ00140A' 7. Delete 'RFQ00140B' 8. Validate delete. 9. Delete 'PRO00140'	5. Two RFQs were searched out 8. Two RFQs were deleted	5. Two RFQs were searched out 8. Two RFQs were deleted	P	Kiran_Shejal_Intern2017
TC_11	FTC	My Season Line Sheet: Create sourcing config	NA	1. Create a supplier 'SUPP00300' 2. Create season 'Mens Apparel Brand 1 Division 1 Autumn 2017' 3. In line sheet, create product 'PRO00300' to season 4. Choose this product, select action 'Create New: Sourcing Config' 5. In pop up window, search and select supplier 'SUPP00300', create this source. 6. Open this product and go to souring tab, choose the new source. 7. Remove the product from season 8. Delete product 9. Delete season 10. Delete supplier	6. The new source should be listed in source list, and verify the supplier is 'SUPP00300'	6. The new source should be listed in source list, and verify the supplier is 'SUPP00300'	P	Kiran_Shejal_Intern2017

TC_12	FTC	Materials tab: BOM CRUD	NA	1. Create material 'MAT00340' 2. Create season 'Mens Apparel Brand 1 Division 1 Autumn 2021' 3. In line sheet, create product 'PRO00340' to season 4. Go to spec tab, create spec '001' 5. Go to Bom tab, create a BOM 'BOM00340' 6. In Shell section, choose the first row 7. click materials button to open mateiral panel. Search out material 'MAT00340', add it to BOM. 8. Save and check in bom. 9. Validate BOM. 10. Remove BOM 11. Remove the product from season 12. Delete product 13. Delete season	9. Validate BOM table under BOM page, 'MAT00340' should be listed in BOM	9. Validate BOM table under BOM page, 'MAT00340' is not listed in BOM	F	Kiran_Shejal_Intern2017
TC_13	FTC	Specification tab: new spec, initialize spec from existing spec	NA	1. Create season 'Mens Apparel Brand 1 Division 1 Autumn 2019' 2. In line sheet, create product 'PRO00320' to season 3. Go to Spec tab, choose one source, create spec '001'. 4. Validate creation 5. Under spec tab, in existing spec drop down list, choose spec '001', initialize another one. 6. Validate creation 7. Remove the product from season 8. Delete product 9. Delete season	4. '001' has been created. 6. '002' has been created.	4. '001' has been created. 6. '002' has been created.	P	Kiran_Shejal_Intern2017

TC_14	FTC	CRUD for material - personal care type	NA	1. Go to library - material 2. Create material 'MAT00080A' on personal care type 3. Create material 'MAT00080B' on personal care type 4. In library, enter material name 'MAT00080%', validate creation 5. Search and update 'MAT00080B' 6. In library, enter material name 'MAT00080%', validate update 7. Delete 'MAT00080A' 8. Delete 'MAT00080B' 9. In library, enter material name 'MAT00080%', validate delete	4. Two materials have been created successfully 6. 'MAT00080B' has been updated 7. Two materials have been removed	4. Two materials have been created successfully 6. 'MAT00080B' has been updated 7. Two materials have been removed	P	Kiran_Shejal_Intern2017
TC_15	FTC	RFQ CRUD: Create RFQ in a product, then search out RFQ in library.	NA	1. Create product 'PRO00140' in library 2. Create Supplier 'SUPP00140' in library 3. Create a RFQ 'RFQ00140A' under in library. 4. Create a RFQ 'RFQ00140B' under in library. 5. Go to linrary - RFQ, enter name 'RFQ00140%' to validate creation. 6. Delete 'RFQ00140A' 7. Delete 'RFQ00140B' 8. Validate delete. 9. Delete 'PRO00140'	5. Two RFQs were searched out 8. Two RFQs were deleted	5. Two RFQs were searched out 8. Two RFQs were deleted	P	Kiran_Shejal_Intern2017