|  |
| --- |
| System Test Plan |
| Cylinders & Orders Management System (COMS) Project |
| This document provides a plan for the testing work to be performed during the development of the Cylinders & Orders Management System. |

**This page has been intentionally left blank**

|  |  |  |  |
| --- | --- | --- | --- |
| Filing Reference : | GG/COMS/MP.2/1 | | |
| Document Title: | System Test Plan | | |
| Version : | 1 | | |
| Prepared by : | Alvin Chang Parkk Khiong | | |
| Date Created : | Tuesday, November 1st, 2011 | | |
|  |  | |  |
| Approved by : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |  |
|  | Maung Tin Kyaw Oo | | Date |
|  | Technical Lead | |  |
| Authorized by : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |  |
|  | Tran Ba Tien | | Date |
|  | Project Manager | |  |
| **© 2011 Hoang Kim Joint Stock Company.**  **The information contained in this document is the property of Hoang Kim Joint Stock Company. The contents must not be reproduced, wholly or in part, for purposes other than for which it has been supplied, without the prior permission of Hoang Kim Joint Stock Company, or, if it has been furnished under contract to another party, as expressly authorised under that contract. Hoang Kim Joint Stock Company shall not be liable for any errors or omissions.** | |  | |

**Document Reference:** GG/COMS/MP.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Remarks** | **Authorised by** |
| 1.0 | 01/11/2011 | Alvin Chang Parkk Khiong | Created Document |  |
|  |  |  |  |  |

**Cylinders & Orders Management System (COMS)  
 Project**

**System Test Plan**

**Distribution:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Department** | **Organisation** |
| Tran Ba Tien | SE18-Team2S | MTech (ISS) |
| Maung Tin Kyaw Oo | SE18-Team2S | MTech (ISS) |
| Chang Parkk Khiong Alvin | SE18-Team2S | MTech (ISS) |
| Koh Ming Jin | SE18-Team2S | MTech (ISS) |
| Junaith | SE18-Team2S | MTech (ISS) |
| Phyo Phyo Lwin | SE18-Team2S | MTech (ISS) |

Table of Contents

[1. INTRODUCTION 1](#_Toc313213093)

[1.1 Purpose 1](#_Toc313213094)

[1.2 Audience 1](#_Toc313213095)

[1.3 Organisation 1](#_Toc313213096)

[1.4 References 1](#_Toc313213097)

[2. TEST PROCEDURE 4](#_Toc313213098)

[2.1 Approach 4](#_Toc313213100)

[2.2 Scope 4](#_Toc313213101)

[2.3 Procedure 4](#_Toc313213102)

[3. TEST DATA 6](#_Toc313213103)

[3.1 Sales Orders 6](#_Toc313213105)

[3.2 Cylinders 6](#_Toc313213106)

[3.3 Employees 6](#_Toc313213107)

[3.4 Departments 6](#_Toc313213108)

[3.5 Workflow and Steps 6](#_Toc313213109)

[3.6 Roles and Access Rights 6](#_Toc313213110)

[3.7 Performance Formula 6](#_Toc313213111)

[3.8 Error Codes 6](#_Toc313213112)

[4. TEST SPECIFICATION 8](#_Toc313213113)

[4.1 Change Cylinder Priority 8](#_Toc313213115)

[4.1.1 Test: Update for Cylinder Priority 8](#_Toc313213116)

[4.2 Export Cylinder Queues 8](#_Toc313213117)

[4.2.1 Test: Export Cylinder Queue 8](#_Toc313213118)

[4.3 Login 9](#_Toc313213119)

[4.3.1 Test: User Login 9](#_Toc313213120)

[4.4 Logout 9](#_Toc313213121)

[4.4.1 Test: User Logout 9](#_Toc313213122)

[4.5 Manage Employee-Role 9](#_Toc313213123)

[4.5.1 Test: Add Employee-Role 9](#_Toc313213124)

[4.5.2 Test: Remove Employee-Role 9](#_Toc313213125)

[4.6 Manage Error 10](#_Toc313213126)

[4.6.1 Test: Create Error Code 10](#_Toc313213127)

[4.6.2 Test: Remove Employee-Role 10](#_Toc313213128)

[4.7 Manage Performance Formula 10](#_Toc313213129)

[4.8 Manage Rights 10](#_Toc313213130)

[4.9 Manage Role Approval 10](#_Toc313213131)

[4.10 Manage Role 10](#_Toc313213132)

[4.11 Manage Sales Order 11](#_Toc313213133)

[4.12 Manage User Account 11](#_Toc313213134)

[4.13 Manage Workflow-Step 11](#_Toc313213135)

[4.14 Print Step List 11](#_Toc313213136)

[4.15 Print Worker Marks Report 11](#_Toc313213137)

[4.16 Send Cylinder To A Particular Step 11](#_Toc313213138)

[4.17 Start Cylinder Production Process 11](#_Toc313213139)

[4.18 Stop Cylinder Production Process 11](#_Toc313213140)

[4.19 Update Cylinder Status 12](#_Toc313213141)

[4.20 View & Print Cylinder Information 12](#_Toc313213142)

[4.21 View Cylinder Progress Logs 12](#_Toc313213143)

[4.22 View Employee Details 12](#_Toc313213144)

[4.23 View Error 12](#_Toc313213145)

[4.24 View Order Progress Logs 12](#_Toc313213146)

[4.25 View Sales Order 12](#_Toc313213147)

[4.26 View Workflow Queues 12](#_Toc313213148)

[Approval Record GG/Forms/Approval 13](#_Toc313213149)

# 1. INTRODUCTION

Hoang Kim Joint Stock Company is one of the leading providers of printing cylinders in Vietnam. They are currently using the latest technologies from Germany and Japan, and their client bases include various Vietnamese corporations as well as across Southeast Asia.

The company would like to have an integrated IT system that can:

* Manage the main manufacturing process, i.e. sales orders and cylinders.
* Benchmark employees’ performance to calculate bonuses.
* Give management reporting tools for daily operations.
* Be easy to maintain and to add new features in the future.

## Purpose

The purpose of this document is to provide a plan for the testing work to be performed during the development of the Cylinders & Orders Management System.

## Audience

The intended reader of this plan is the project leader, who is responsible for carrying out the testing of the system. This document should provide all the necessary instructions.

## Organisation

The testing approach and procedure is summarised in Section 2. The test data set is defined in Section 3. Finally, Section 4 specifies each of the system tests to be performed.

## References

To fully understand the background to this project, the reader should also be familiar with:

1. COMS Project Plan, reference GG/COMS/MP.1/2, version 2.0, dated 1 February 2011.
2. COMS Quality Plan, reference GG/COMS/MQ.1/3, version 3.0, dated 13 September 2011.
3. COMS User Requirement Specifications, reference GG/COMS/TS.1/1.1, version 1.1, dated 1 February 2011.
4. COMS High Level Design Specifications, reference GG/COMS/TS.2/1, version 1.0, dated 7 April 2011.
5. COMS Change CylinderPriority UCRR, reference GG/COMS/TS.2/1, version 1.0, dated 29 April 2011
6. COMS Export Cylinder Queues UCRR, reference GG/COMS/TS.2/2, version 1.0, dated 29 April 2011
7. COMS Login UCRR, reference GG/COMS/TS.2/3, version 1.0, dated 29 July 2011
8. COMS Logout UCRR, reference GG/COMS/TS.2/4, version 1.0, dated 29 July 2011
9. COMS Manage Employee-Role UCRR, reference GG/COMS/TS.2/5, version 1.0, dated 30 July 2011
10. COMS Manage Error UCRR, reference GG/COMS/TS.2/6, version 1.0, dated 15 May 2011
11. COMS Manage Performance Formula UCRR, reference GG/COMS/TS.2/7, version 1.0, dated 14 June 2011
12. COMS Manage Rights UCRR, reference GG/COMS/TS.2/8, version 1.0, dated 14 June 2011
13. COMS Manage Role Approval UCRR, reference GG/COMS/TS.2/9, version 1.0, dated 30 July 2011
14. COMS Manage Role UCRR, reference GG/COMS/TS.2/10, version 1.0, dated 14 June 2011
15. COMS Manage SalesOrder UCRR, reference GG/COMS/TS.2/11, version 1.0, dated 15 July 2011
16. COMS Manage User Account UCRR, reference GG/COMS/TS.2/12, version 1.0, dated 14 June 2011
17. COMS Manage Workflow-Step UCRR, reference GG/COMS/TS.2/13, version 1.0, dated 30 July 2011
18. COMS Print Step List UCRR, reference GG/COMS/TS.2/14, version 1.0, dated 14 June 2011
19. COMS Print Worker Marks Report UCRR, reference GG/COMS/TS.2/15, version 1.0, dated 14 June 2011
20. COMS Send CylinderToAParticularStep UCRR, reference GG/COMS/TS.2/16, version 1.0, dated 29 April 2011
21. COMS Start CylinderProductionProcess UCRR, reference GG/COMS/TS.2/17, version 1.0, dated 29 April 2011
22. COMS Stop CylinderProductionProcess UCRR, reference GG/COMS/TS.2/18, version 1.0, dated 29 April 2011
23. COMS Update Cylinder Status UCRR, reference GG/COMS/TS.2/19, version 1.0, dated 14 June 2011
24. COMS View&Print CylinderInformation UCRR, reference GG/COMS/TS.2/20, version 1.0, dated 29 April 2011
25. COMS View Cylinder Progress Log UCRR, reference GG/COMS/TS.2/21, version 1.0, dated 14 June 2011
26. COMS View EmployeeDetails UCRR, reference GG/COMS/TS.2/23, version 1.0, dated 29 April 2011
27. COMS View Error UCRR, reference GG/COMS/TS.2/24, version 1.0, dated 29 April 2011
28. COMS View Order Progress Log UCRR, reference GG/COMS/TS.2/25, version 1.0, dated 14 June 2011
29. COMS View SalesOrder UCRR, reference GG/COMS/TS.2/27, version 1.0, dated 29 July 2011
30. COMS View WorkflowQueues UCRR, reference GG/COMS/TS.2/22, version 1.0, dated 29 April 2011

# 2. TEST PROCEDURE

The aim of understanding system testing is to enable the project to demonstrate, with an acceptable degree of confidence, that the Cylinders & Orders Management System satisfies the requirement as defined in the system specification (ref 4).



## Approach

The basic method to be adopted will be to use a test data set, as defined in Section 3, to exercise and demonstrate the functions and features of the system. This will be done through a series of defines tests given in Section 4.

## Scope

The tests defined in Section 4 will attempt to demonstrate that the features and functions specified in the system specification (ref 4) operate correctly. However it should be noted that features that have no specific user requirement, have no tests identified to verify that the facilities operate correctly.

## Procedure

The system tests to be performed are defined in Section 4. System testing will be deemed complete when all the defined tests have been performed, documented, and approved by the Project Manager. For each test defined in Section 4, the following step shall be performed:

1. Determine the expected results of the test;

2. Carry out the test instructions and create any required hardcopy print-outs;

3. Compare the expected with the actual results. If the required results have not been achieved then define the required corrective action;

4. Fill out a test log form, as shown in Figure 2.1. Attach to the form with all the required print-outs, and file in the system testing workfile.

5. If the required results were not achieved then:

1. Implement the corrective action, as specified on the test log form;
2. Perform steps 2, 3 and 4 above. Create a NEW test log form for each repeated test;
3. Repeat (a) and (b) above until the test is successful.

When the tests have been completed, the system testing workfile should submitted to the Project Manager for approval.

**Figure 2.1 :** Test Log Form.

|  |  |
| --- | --- |
| Testing Log Form GG/Forms/Testing | |
| Project Name **Cylinders & Orders Management System** | |
| Test Identifier | File Ref **GG/COMS/TW.3/** |
| Tested by | Date |
| Approved by | Date |
| Test Description **(give brief description)** | |
| Expected Results **(refer to attached documents if necessary)** | |
| Actual Results **(refer to attached documents if necessary)** | |
| Test Status **(either** SUCCESSFUL **or** ERROR) | |
| Corrective Action or Remarks **(refer to attached documents if necessary)** | |

# 3. TEST DATA

The aim of this section is to define the basic data set to be used in the system tests defined in Section 4.



## Sales Orders

Figure 3.1 defines the sales orders to be used in the system tests.

## Cylinders

Figure 3.2 defines the cylinders to be used in the system tests.

## Employees

Figure 3.4 defines the employee information to be used in the system tests

## Departments

Figure 3.5 defines the departments to be used in the system tests.

## Workflow and Steps

Figure 3.6 defines the workflow and steps to be used in the system tests.

## Roles and Access Rights

Figure 3.7 defines the roles and access rights to be used in the system tests.

## Performance Formula

Figure 3.8 defines the performance formulas to be used in the system tests

## Error Codes

Figure 3.9 defines the error codes to be used in the system tests.

Figure 3.1: Sales Orders to be used in System Tests

Figure 3.2: Cylinders to be used in System Tests

Figure 3.3: Employees to be used in System Tests

Figure 3.4: Departments to be used in System Tests

Figure 3.5: Workflows and steps to be used in System Tests

Figure 3.6: Roles and Access Rights to be used in System Tests

|  |  |
| --- | --- |
| **Roles** | **Access Rights** |
| Accountant |  |
| Director |  |
| Operations Manager |  |
| Administrator |  |

Figure 3.7: Performance Formulas to be used in System Tests

Figure 3.8: Error Codes to be used in System Tests

# 4. TEST SPECIFICATION

The system tests to be performed, using the procedure defined in Section 2 and the test data given in Section 3, are listed in the following subsections. Note that unless otherwise stated, all tests assume that:

* The tester has access privileges set such that read, write, edit and delete operations are permissible for all record types



## Change Cylinder Priority

The aim of this test is to verify that the facilities to change the priority of a particular cylinder operate as specified in the specifications (ref 4, 5).

### Test: Update for Cylinder Priority

1. Before commencing, ensure that there is valid cylinder information in the system. Refer to section 4.20 for more details.
2. Select the **Change Cylinder Priority** command from the **Cylinder Menu**. The list of existing cylinders will be shown.
3. Select one of the cylinders in the list which has a low priority and change it to high priority. Save the change.
4. Reload the list to ensure the changes have been saved.
5. Select one of the cylinders in the list which has a high priority and change it to low priority. Save the change.
6. Reload the list to ensure the changes have been saved.
7. Print out the screen shot of the cylinder status before and after the priority change to demonstrate the correct information is stored.

## Export Cylinder Queues

The aim of this test is to verify that the facilities to export cylinder queues operate as specified in the specifications (ref 4, 6).

### Test: Export Cylinder Queue

1. Before commencing, ensure that there is valid cylinder information in the system. Refer to section 4.20 for more details.
2. Select the **Export Cylinder Queue** command from the **Cylinder Menu**.
3. Print out the exported queue information, and print out the screenshots of the step list (section 4.14) and cylinder list (section 4.20) to demonstrate the correct information is exported.

## User Login / Logout

The aim of this test is to verify that the facilities to allow a user to log in and out of the system operate as specified in the specifications (ref 4, 7, 8).

### Test: User Login

1. Before commencing, ensure that no user is currently logged in to the system.
2. Select the **Login** command from the **Login Menu**. Enter a valid user account id and password from the employee data in Figure 3.3
   1. UserID: Jacob.smith / Password: password123
3. Print out the subsequent screenshot to show the successfully log-in of the user account.

### Test: User Logout

1. Before commencing, ensure that a valid user is currently logged in to the system. (See section 4.3).
2. Select the **Logout** command from the **Login Menu**.
3. Print out the subsequent screenshot to show the successfully log-out of the user account.

## Manage Employee-Role Relationship

The aim of this test is to verify that the facilities to manage the employee-role relationships operate as specified in the specifications (ref 4, 9).

### Test: Add Employee-Role Relationship

1. Before commencing, ensure that the database table storing the employee-role data is empty.
2. Select the **Add Employee-Role Relationship** command from the **Employee Menu**. Select the employee and add the role as given in Figure 3.3, 3.6
3. Use **View Employee Details** command from the **Employee Menu** to check that the role has been successfully added to the employee.
4. Print out the screen shots of the Employee Details with their roles to demonstrate the correct data was stored.

### Test: Remove Employee-Role Relationship

1. Select the **Remove Employee-Role Relationship** command from the **Employee Menu**. Select the employee from 4.5.1, and remove the role previously added to it.
2. Use **View Employee Details** command from the **Employee Menu** to check that the role has been successfully removed.
3. Print out the screen shots of the Employee Details with their roles to demonstrate the correct data was stored.

## Manage Error

The aim of this test is to verify that the facilities to manage the error codes operate as specified in the specifications (ref 4, 10, 27).

### Test: Create Error Code

1. Before commencing, ensure that the database table storing the error code data is empty.
2. Select the **Create Error Code** command from the **Workflow Error Message Menu**. Add the error code data as shown in Figure 3.8.
3. Use **View Error Code** command from the **Workflow Error Message Menu** to check that the error code has been successfully entered into the database.
4. Print out the screen shots of the following error codes to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Edit Error Code

1. Select the **Edit Error Code** command from the **Workflow Error Message Menu**.
2. Edit the first error code to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Error Code** command from the **Workflow Error Message Menu** to check that the error code has been successfully entered into the database.
4. Print out the screen shots of the edited error code to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original error code.

### Test: Delete Error Code

1. Select the **Delete Error Code** command from the **Workflow Error Message Menu**.
2. Delete the following error codes:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Error Code** command from the **Workflow Error Message Menu** to check that the error code has been successfully entered into the database.
4. Print out the screen shots of the list of error codes to demonstrate the error codes are no longer stored in the database.

## Manage Performance Formula

The aim of this test is to verify that the facilities to manage the performance formula operate as specified in the specifications (ref 4, 11).

### Test: Create Performance Formula

1. Before commencing, ensure that the database table storing the performance formula data is empty.
2. Select the **Create Performance Formula** command from the **Formula Menu**. Add the performance formula data as shown in Figure 3.7.
3. Use **View Performance Formula** command from the **Formula Menu** to check that the performance formula has been successfully entered into the database.
4. Print out the screen shots of the following performance formula to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Edit Performance Formula

1. Select the **Edit Performance Formula** command from the **Formula Menu**.
2. Edit the first performance formula to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Performance Formula** command from the **Formula Menu** to check that the performance formula has been successfully entered into the database.
4. Print out the screen shots of the edited performance formula to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original performance formula.

### Test: Delete Performance Formula

1. Select the **Delete Performance Formula** command from the **Formula Menu**.
2. Delete the following performance formula:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Performance Formula** command from the **Formula Menu** to check that the performance formula has been successfully entered into the database.
4. Print out the screen shots of the list of performance formulas to demonstrate the performance formulas are no longer stored in the database

## Manage Rights

The aim of this test is to verify that the facilities to manage the access rights operate as specified in the specifications (ref 4, 12).

### Test: Create Access Rights

1. Before commencing, ensure that the database table storing the Access Rights data is empty.
2. Select the **Create Access Rights** command from the **Role Menu**. Add the Access Rights data as shown in Figure 3.6.
3. Use **View Access Rights** command from the **Role Menu** to check that the Access Rights has been successfully entered into the database.
4. Print out the screen shots of the following Access Rights to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Edit Access Rights

1. Select the **Edit Access Rights** command from the **Role Menu**.
2. Edit the first Access Rights to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Access Rights** command from the **Role Menu** to check that the Access Rights has been successfully entered into the database.
4. Print out the screen shots of the edited Access Rights to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original Access Rights.

### Test: Delete Access Rights

1. Select the **Delete Access Rights** command from the **Role Menu**.
2. Delete the following Access Rights:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Access Rights** command from the **Role Menu** to check that the Access Rights has been successfully entered into the database.
4. Print out the screen shots of the list of Access Rights to demonstrate the Access Rights are no longer stored in the database

## Manage Role-Access Rights Relationship

The aim of this test is to verify that the facilities to manage the access rights operate as specified in the specifications (ref 4, 12).

### Test: Add Role-Rights Relationship

1. Before commencing, ensure that the database table storing the Role-Rights data is empty.
2. Select the **Add Role-Rights Relationship** command from the **Roles Menu**. Select the role and add the rights as given in Figure 3.6
3. Use **View Role Details** (section 4.10) to check that the role-right access has been successfully added to the employee.
4. Print out the screen shots of the Role Details with their roles to demonstrate the correct data was stored.

### Test: Remove Role-Rights Relationship

1. Select the **Remove Role-Rights Relationship** command from the **Roles Menu**. Select the employee from 4.5.1, and remove the role previously added to it.
2. Use **View Role Details** (section 4.10) to check that the role-right access has been successfully removed.
3. Print out the screen shots of the Role Details with their roles to demonstrate the correct data was stored.

## Manage Roles

The aim of this test is to verify that the facilities to manage roles and role approval operate as specified in the specifications (ref 4, 14, 13).

### Test: Create Role

1. Before commencing, ensure that the database table storing the Role data is empty.
2. Select the **Create Role** command from the **Role Menu**. Add the Role data as shown in Figure 3.6.
3. Use **View Role** command from the **Role Menu** to check that the Role has been successfully entered into the database.
4. Print out the screen shots of the following Role to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Edit Role

1. Select the **Edit Role** command from the **Role Menu**.
2. Edit the first Role to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Role** command from the **Role Menu** to check that the Role has been successfully entered into the database.
4. Print out the screen shots of the edited Role to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original Role.

### Test: Delete Role

1. Select the **Delete Role** command from the **Role Menu**.
2. Delete the following Role:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Role** command from the **Role Menu** to check that the Role has been successfully entered into the database.
4. Print out the screen shots of the list of Role to demonstrate the Role are no longer stored in the database

### Test: Role Approval

1. Before commencing, ensure that the user account that is logged in has a **Director** role.
2. Select the **Role Approval** command from the **Role Menu**.
3. Approve the following Role Approval requests: #todo
   1. <ABC>#todo
   2. <DEF>#todo
4. Use **View Role** command from the **Role Menu** to check that the Role has been successfully entered into the database.
5. Print out the screen shots of the following Role to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

## Manage Sales Order

The aim of this test is to verify that the facilities to manage sales orders operate as specified in the specifications (ref 4, 15, 29).

### Test: Create Sales Order

1. Before commencing, ensure that the database table storing the Sales Order data is empty.
2. Select the **Create Sales Order** command from the **Sales Order Menu**. Add the Sales Order data as shown in Figure 3.1.
3. Use **View Sales Order** command from the **Sales Order Menu** to check that the Sales Order has been successfully entered into the database.
4. Print out the screen shots of the following Sales Order to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Edit Sales Order

1. Select the **Edit Sales Order** command from the **Sales Order Menu**.
2. Edit the first Sales Order to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Sales Order** command from the **Sales Order Menu** to check that the Sales Order has been successfully entered into the database.
4. Print out the screen shots of the edited Sales Order to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original Sales Order.

### Test: Delete Sales Order

1. Select the **Delete Sales Order** command from the **Sales Order Menu**.
2. Delete the following Sales Order:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Sales Order** command from the **Sales Order Menu** to check that the Sales Order has been successfully entered into the database.
4. Print out the screen shots of the list of Sales Order to demonstrate the Sales Order are no longer stored in the database

## Manage Employee User Account

The aim of this test is to verify that the facilities to manage user accounts operate as specified in the specifications (ref 4, 16, 26).

### Test: Create Employee User Account

1. Before commencing, ensure that the database table storing the Employee User Account data is empty, and that the user account that is logged in has an **Admin** role.
2. Select the **Create Employee User Account** command from the **Employee User Account Menu**. Add the Employee User Account data as shown in Figure 3.3.
3. Use **View Employee User Account** command from the **Employee User Account Menu** to check that the Employee User Account has been successfully entered into the database.
4. Print out the screen shots of the following Employee User Account to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Edit Employee User Account

1. Select the **Edit Employee User Account** command from the **Employee User Account Menu**.
2. Edit the first Employee User Account to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Employee User Account** command from the **Employee User Account Menu** to check that the Employee User Account has been successfully entered into the database.
4. Print out the screen shots of the edited Employee User Account to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original Employee User Account.

### Test: Delete Employee User Account

1. Select the **Delete Employee User Account** command from the **Employee User Account Menu**.
2. Delete the following Employee User Account:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Employee User Account** command from the **Employee User Account Menu** to check that the Employee User Account has been successfully entered into the database.
4. Print out the screen shots of the list of Employee User Account to demonstrate the Employee User Account are no longer stored in the database

## Manage Workflows

The aim of this test is to verify that the facilities to manage the workflow operate as specified in the specifications (ref 4, 17).

### Test: Create Workflow

1. Before commencing, ensure that the database table storing the Workflow data is empty, and that the user account that is logged in has an **Admin** role.
2. Select the **Create Workflow** command from the **Workflow Menu**. Add the Workflow data as shown in Figure 3.5.
3. Use **View Workflow** command from the **Workflow Menu** to check that the Workflow has been successfully entered into the database.
4. Print out the screen shots of the following Workflow to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Modify Workflow Properties

1. Select the **Modify Workflow Properties** command from the **Workflow Menu**.
2. Edit the first Workflow to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Workflow** command from the **Workflow Menu** to check that the Workflow has been successfully entered into the database.
4. Print out the screen shots of the edited Workflow to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original Workflow.

### Test: Deactivate Workflow

1. Select the **Deactivate Workflow** command from the **Workflow Menu**.
2. Deactivate the following Workflows:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Workflow** command from the **Workflow Menu** to check that the Workflow has been successfully deactivated.
4. Print out the screen shots of the list of Workflow to demonstrate the Workflow has been deactivated.

### Test: Activate Workflow

1. Select the **Activate Workflow** command from the **Workflow Menu**.
2. Activate the following Workflows:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Workflow** command from the **Workflow Menu** to check that the Workflow has been successfully activated.
4. Print out the screen shots of the list of Workflow to demonstrate the Workflow has been activated.

## Manage Steps

The aim of this test is to verify that the facilities to manage the steps operate as specified in the specifications (ref 4, 17, 18).

### Test: Create Step

1. Before commencing, ensure that the database table storing the Step data is empty, and that the user account that is logged in has an **Admin** role.
2. Select the **Create Step** command from the **Step Menu**. Add the Step data as shown in Figure 3.5.
3. Use **View Step** command from the **Step Menu** to check that the Step has been successfully entered into the database.
4. Print out the screen shots of the following Step to demonstrate the correct data was stored:
   1. <ABC>#todo
   2. <DEF>#todo

### Test: Modify Step Properties

1. Select the **Modify Step Properties** command from the **Step Menu**.
2. Edit the first Step to the following:
   1. <FJKGHFDJ>#todo
3. Use **View Step** command from the **Step Menu** to check that the Step has been successfully entered into the database.
4. Print out the screen shots of the edited Step to demonstrate the correct data was changed
5. Un-do the edit changing the entry back to the original Step.

### Test: Deactivate Step

1. Select the **Deactivate Step** command from the **Step Menu**.
2. Deactivate the following Steps:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Step** command from the **Step Menu** to check that the Step has been successfully deactivated.
4. Print out the screen shots of the list of Step to demonstrate the Step has been deactivated.

### Test: Activate Step

1. Select the **Activate Step** command from the **Step Menu**.
2. Activate the following Steps:
   1. <ABC>#todo
   2. <DEF>#todo
3. Use **View Step** command from the **Step Menu** to check that the Step has been successfully activated.
4. Print out the screen shots of the list of Step to demonstrate the Step has been activated.

### Test: Print Step List Report

1. Select the **Print Step List Report** command from the **Step Menu**.
2. Print the report for the following period:
   1. 01 Jan 2011 to 31 Dec 2011#todo
3. Print out the screen shots of the list of Step to demonstrate the data printed is accurate.

## Manage Workflow-Step Relationship

The aim of this test is to verify that the facilities to manage the workflow-step relationship operate as specified in the specifications (ref 4, 17).

### Test: Add Workflow-Steps Relationship

1. Before commencing, ensure that the database table storing the Workflow-Steps Relationship data is empty.
2. Select the **Add Workflow-Steps Relationship** command from the **Workflow Menu**. Select the role and add the rights as given in Figure 3.6
3. Use **View Workflow Details** command from the **Workflow Menu** to check that the workflow-step has been successfully added to the employee.
4. Print out the screen shots of the Role Details with their roles to demonstrate the correct data was stored.

### Test: Remove Workflow-Steps Relationship

1. Select the **Remove Workflow-Steps Relationship** command from the **Workflow Menu**. Select the employee from 4.5.1, and remove the role previously added to it.
2. Use **View Workflow Details** command from the **Workflow Menu** to check that the workflow-step has been successfully removed.
3. Print out the screen shots of the Role Details with their roles to demonstrate the correct data was stored.

## Manage Cylinder Production Process

The aim of this test is to verify that the facilities for the cylinder production process operate as specified in the specifications (ref 4, 20, 21, 22, 23).

### Test: Start Cylinder Production Process

1. Before commencing, ensure that the database table storing the cylinder production data is empty.
2. Select the **Start Cylinder Production Process** command from the **Workflow Menu**.
3. Use **View Cylinder Information** command from the **Workflow Menu** to check that the production process for the cylinder has been successfully started.
4. Print out the screen shots of the notification of the started cylinder production process to demonstrate the correct data was stored.

### Test: Stop Cylinder Production Process

1. Use **View Cylinder Information** command from the **Workflow Menu** to view the list of Cylinders available. Select the first cylinder in the list which is started.
2. Select the **Stop Cylinder Production Process** command.
3. Use **View Cylinder Information** command from the **Workflow Menu** to check that the production process for the cylinder has been successfully stopped.
4. Print out the screen shots of the notification of the stopped cylinder production process to demonstrate the correct data was stored.

### Test: Send Cylinder to a Particular Step

1. Use **View Cylinder Information** command from the **Workflow Menu** to view the list of Cylinders available. Select the first cylinder in the list which is not started.
2. Select the **Send Cylinder to a Particular Step** command from the **Workflow Menu**.
3. Select the following step to send the cylinder to:
   1. <ACB>#todo
4. Use **View Cylinder Information** command from the **Workflow Menu** to check that the the cylinder has been successfully sent to the step.
5. Print out the screen shots of the notification to demonstrate the correct data was stored.

## Manage Cylinder Status

The aim of this test is to verify that the facilities to manage the cylinder status and print the worker marks report operate as specified in the specifications (ref 4, 15, 19).

### Test: Update Cylinder Status

1. Before commencing, ensure that the cylinder production process has started for the following cylinder:
   1. <ABC>#todo
2. Use **Update Cylinder Status** command from the **Workflow Menu.** Use the barcode reader to scan the bar code of the cylinder. To simulate, input the following code instead:
   1. <ABHCBSN>#todo
3. Use **View Cylinder Information** command from the **Workflow Menu** to check that the cylinder has been successfully moved onto the next step.
4. Print out the screen shots of the cylinder details to demonstrate the correct data was stored.

### Print Worker Marks Report

1. Before commencing, ensure that the user account that is logged in has an **Accountant** role.
2. Use **Print Worker Marks Report** command from the **Workflow Menu.** Select the following employee from the list shown, and select the **Print** command.
   1. <AKHKJDHD>#todo
3. Print out the screen shots of the employee details to demonstrate the correct data was printed accurately.

## Print Cylinder Information Report

The aim of this test is to verify that the facilities to print the cylinder information report operate as specified in the specifications (ref 4, 24).

### Print Cylinder Information Report

1. Before commencing, ensure that the user account that is logged in has an **Operations Manager** role.
2. Use **Print Cylinder Information Report** command from the **Workflow Menu.** Select the following cylinder from the list shown, and select the **Print** command.
   1. <AKHKJDHD>#todo
3. Print out the screen shots of the cylinder details to demonstrate the correct data was printed accurately.

## View Cylinder Progress Logs

The aim of this test is to verify that the facilities to view the cylinder progress logs operate as specified in the specifications (ref 4, 25).

### View Cylinder Progress Logs

1. Before commencing, ensure that the user account that is logged in has an **Operations Manager** role.
2. Use **View Cylinder Progress Logs** command from the **Workflow Menu.** Select the following cylinder from the list shown:
   1. <AKHKJDHD>#todo
3. Print out the screen shots of the cylinder progress log details to demonstrate the data was displayed correctly.

## View Order Progress Logs

The aim of this test is to verify that the facilities to view the order progress logs operate as specified in the specifications (ref 4, 28).

### View Order Progress Logs

1. Before commencing, ensure that the user account that is logged in has an **Operations Manager** role.
2. Use **View Order Progress Logs** command from the **Sales Order Menu.** Select the following order from the list shown:
   1. <AKHKJDHD>#todo
3. Print out the screen shots of the order progress log details to demonstrate the data was displayed correctly

## View Workflow Queues

The aim of this test is to verify that the facilities to view workflow queues operate as specified in the specifications (ref 4, 30).

### View Workflow Queues

1. Before commencing, ensure that the user account that is logged in has an **Operations Manager** role.
2. Use **View Workflow Queues** command from the **Workflow Menu.**
3. Print out the screen shots of the workflow queue details to demonstrate the data was displayed correctly

|  |  |  |
| --- | --- | --- |
| Approval Record GG/Forms/Approval | | |
| **Project Name COMS Project** | | |
| **Document Ref** GG/COMS/MP.2/1 |  | |
| Approved by | Date | |
| Authorised by | Date | |
| The document authorisation appears on the title page. | |  |
| The structure of the publication is logical. | |  |
| The distribution list is correct. | |  |
| The title page is signed. | |  |
| Calculations appear reasonable, are neatly presented and have been checked. | |  |
| Theory and formulae are correct and properly applied. | |  |
| Illustrations are relevant, readable and logically placed. | |  |
| There are no typographical errors. | |  |
| Units are consistent throughout. | |  |
| The security classification is correct. | |  |
| There are no obvious omissions. | |  |
| The document complies with the Client's requirements, however specified. | |  |
| Responsibility is accepted for all opinions, conclusions and recommendations. | |  |
| The document does not run counter to company policy. | |  |
|  | |  |
|  | |  |