

**Prepared by eMomentum Interactive Systems Limited**

**For First Chartered Securities**

**Submitted on May 2016**

**FC S Governance Risk and Compliance System**

Incident Management Module Use Case Test Scenario

# **Introduction**

This document serves to guide the Quality Review Team through testing the Incident Management module of the Governance Risk and Compliance System developed for First Chartered Securities Group.

For each product functional area, this Quality Review Record is composed of the following:

1. A process flow image showing how the system is designed to work, based on User Requirements Document and their subsequent revisions.
2. Success criteria describing the expected outcome as per User Requirements and their subsequent revision.
3. Test results as observed by the Quality Review Team.

# Responsibilities

Team Chair and Chief Reviewer- Catherine Omasete, Dorothy Maseke

Product Presenter: Loise Mwende

Reviewers: Charles Waweru, Sullivan Kiprotich, Joy Gachoka, Kelvin Njoroge, Stanley Kamau

# Quality Review Summary:

* Functional Areas Quality Reviewed passed without reservations:
* Functional Areas Quality Review failed and require additional review:
* Functional Areas Quality Review accepted with Concession and follow-up actions Recommended:

Contents

[Introduction 1](#_Toc466013701)

[Responsibilities 1](#_Toc466013702)

[Quality Review Summary: 1](#_Toc466013703)

[Incident Management 3](#_Toc466013704)

[Test Cases 3](#_Toc466013705)

[Create Incident Control Category 3](#_Toc466013706)

[Prerequisites 3](#_Toc466013707)

[Success Criteria 3](#_Toc466013708)

[Test use Case 3](#_Toc466013709)

[Edit Incident Category 3](#_Toc466013710)

[Prerequisites 3](#_Toc466013711)

[Success Criteria 3](#_Toc466013712)

[Test use Case 3](#_Toc466013713)

[Delete Incident Category 3](#_Toc466013714)

[Prerequisites 3](#_Toc466013715)

[Success Criteria 3](#_Toc466013716)

[Test use Case 3](#_Toc466013717)

[Create Incident 3](#_Toc466013718)

[Prerequisites 3](#_Toc466013719)

[Success Criteria 3](#_Toc466013720)

[Test Use Case: Scenario 1 4](#_Toc466013721)

[Test Use Case: Scenario 2 4](#_Toc466013722)

[Validate Incident 4](#_Toc466013723)

[Prerequisites 4](#_Toc466013724)

[Success Criteria 4](#_Toc466013725)

[Test use Case 4](#_Toc466013726)

[Create Action 5](#_Toc466013727)

[Prerequisites 5](#_Toc466013728)

[Success Criteria 5](#_Toc466013729)

[Test Use Case 5](#_Toc466013730)

[Edit Action 5](#_Toc466013731)

[Prerequisites 5](#_Toc466013732)

[Success Criteria 5](#_Toc466013733)

[Test Use Case 5](#_Toc466013734)

[Delete Action 5](#_Toc466013735)

[Prerequisites 5](#_Toc466013736)

[Success Criteria 5](#_Toc466013737)

[Test Use Case 5](#_Toc466013738)

[Update Action 5](#_Toc466013739)

[Prerequisites 5](#_Toc466013740)

[Success Criteria 5](#_Toc466013741)

[Test Use Case 5](#_Toc466013742)

# Incident Management

It’s important for every company/organization to keep track of all the incidents that take place. The Incident Management Module is connected to Risk Module in that all incidents that occur in the organization are related to the specific risks. If a specific risk is not found, then the user will be prompted to propose a new risk in the selected risk category. This helps in identifying, tracking and managing high level risk areas.

# Test Cases

The table below show the processes in the Incident Management module and the actors who perform them;

|  |  |
| --- | --- |
| Process | Actors |
| Create Incident Control Category | Risk Manager |
| Edit Incident Control Category | Risk Manager |
| Delete Incident Control Category | Risk Manager |
| Create an Incident | Unit Owner |
| Validate an Incident | Risk Manager |
| Create Action | Risk Manager |
| Edit Action | Risk Manager |
| Delete Action | Risk Manager |
| Update Action Status | Risk Manager |

## Create Incident Control Category

This process involves creating an Incident category.

1. The user should have successfully logged into the system.
2. The user should have navigated to the Incident Management dashboard.

### Success Criteria

1. Should be able to easily find the New Category button.
2. Should be able to locate New category button:
   1. On the dropdown when creating a new Incident.
   2. On the filters located on the dashboard.

### Test use Case

#### Actors

* **Risk Manager:** Creates Incident Category

#### Use Case Scenario: create Risk Category

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk manager clicks on the New Category button on the dashboard |  |  |  |
| 1. Risk Manager fills the details |  |  |  |
| 1. Risk manager saves form |  |  |  |

**Overall Comments:**

## Edit Incident Category

This process involves editing an Incident category

### Prerequisites

1. The user should have successfully logged into the system.
2. The user should have navigated to the Incident Management dashboard.
3. There should be existing incident categories in the system.

### Success Criteria

1. Should be able to easily navigate to the Incident category page.
2. Should be able to easily find the edit incident category function.
3. Should successfully edit a Category.
4. The edited category should appear on the list of all categories.

### Test use Case

#### Actors

* **Risk Manager:** Edits Incident Category

#### Use Case Scenario: Edit Incident Category

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk Manager clicks on New Category Button |  |  |  |
| 1. On the pop-up the risk manager locates desired incident and clicks on the pencil icon to edit |  |  |  |
| 1. Risk Manager fills the details |  |  |  |
| 1. Risk manager saves changes |  |  |  |

**Overall Comments:**

## Delete Incident Category

This process involves deleting an Incident category

### Prerequisites

1. The user should have successfully logged into the system.
2. The user should have navigated to the Incident Management dashboard.
3. There should be existing Incident Categories in the system.
4. Desired categories should not have existing Incidents linked to them.

### Success Criteria

1. Should be able to easily navigate to the Incident category page.
2. Should be able to easily find the delete incident category function.
3. Should successfully delete a Category.
4. The deleted category should not appear:
   1. On the dropdown when creating a new Incident
   2. On the filters located on the dashboard

**Note:** You cannot delete and incident category that has existing incidents linked to it

### Test use Case

#### Actors

* **Risk Manager:** Deletes Incident Category

#### Use Case Scenario: Delete Incident Category

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk Manager clicks on New Category Button |  |  |  |
| 1. On the pop-up the risk manager locates desired incident and clicks on the trash icon to delete |  |  |  |

**Overall Comments:**

## Create Incident

This process involves creating an incident:

### Prerequisites

1. The user should have successfully logged into the system.
2. The system should have existing business units.
3. The system should have existing risk existing in their risk categories.
4. The user should have successfully navigated to the Incident Management dashboard.
5. The system should have existing incident categories.

### Success Criteria

1. Should be able to easily navigate to the Incident Management dashboard.
2. Should be able to easily find the Create Incident button.
3. Should be able to Create Incident.
4. Should be able to view details of Incident.
5. Should be able to find created incident from dashboard.

### Test Use Case: Scenario 1

#### Actors

**Unit Owner:** Creates an Incident

**Risk Manager:** Receives notification once an Incident has been created

#### Use Case Scenario 1a: Create Incident where a Risk exists

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Unit owner clicks on Create Incident Report |  |  |  |
| 1. Unit owner fills in the Incident details |  |  |  |
| 1. Unit owner associates the incident with an existing risk |  |  |  |
| 1. Unit owner submits/saves |  |  |  |
| 1. Risk Manager receives a notification |  |  |  |
| 1. Validate incident process |  |  |  |

#### Use Case Scenario 1b: Create Incident where a risk does not exist

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Unit owner clicks on Create Incident |  |  |  |
| 1. Unit owner fills in the Incident details |  |  |  |
| 1. Unit Owner proposes a risk to be associated with Incident |  |  |  |
| 1. Unit owner submits/saves |  |  |  |
| 1. Risk Manager receives two notifications namely: 2. A proposed risk 3. A new incident |  |  |  |
| 1. Validate incident process |  |  |  |

**Note**: The Risk Manager can only validate an incident after validating the proposed risk

**Overall Comments**

### Test Use Case: Scenario 2

#### Actors

**Risk Manager:** Creates an Incident

#### Use Case Scenario 1a: Create Incident where a Risk does exists

**Note:** The Risk Manager cannot propose a risk

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk Manger clicks on Create Incident Report |  |  |  |
| 1. Risk Manager owner fills in the Incident details |  |  |  |
| 1. Risk Manager associates the incident with an existing risk |  |  |  |
| 1. Risk Manager owner submits/saves |  |  |  |
| 1. Responsible Manager receives a notification |  |  |  |

**Overall Comments**

## Validate Incident

This process involves selecting the proposed incidents to approve/reject.

### Prerequisites

1. The Risk Manager should have successfully logged into the system.
2. The Risk Manager should have accessed the specific incident from the notification sent.

### Success Criteria

1. Should be able to access the specific incident from the notification sent.
2. Should be able to approve/reject the incident.

### Test use Case

#### Actors

* Risk Manager: Approves/Rejects Incident

#### Use Case Scenario: Validate Incident

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk Manager accesses the specific proposed incident |  |  |  |
| 1. Risk Manager clicks on the Approve/Reject button |  |  |  |
| 1. System sends email notification to Responsible Manager |  |  |  |

**Overall Comments:**

## Create Action

The actions module records actions taken to resolve an incident.

### Prerequisites

1. The user should have successfully logged into the system.
2. The system should have existing Incidents
3. The user should have successfully navigated to the Incident Management dashboard.

### Success Criteria

1. The user should be able to find the add action function.
2. The user should be able to locate created action.
   1. From the Incident Management dashboard.
   2. From an Incident’s page.

### Test Use Case

#### Actors

**Risk Manager:** Create new actions

#### Use Case Scenario: create actions

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk manager navigates to the desired incident |  |  |  |
| 1. Risk Manager clicks on the New Action button |  |  |  |
| 1. Risk manager saves form |  |  |  |
| 1. Incident’s responsible manager is notified |  |  |  |
| 1. Action Owner is notified |  |  |  |

**Overall Comments**

## Edit Action

### Prerequisites

1. The user should have successfully logged into the system.
2. The system should have existing Incidents.
3. The user should have successfully navigated to the Incident Management dashboard.
4. The system should have existing actions.

### Success Criteria

1. The user should be able to find the edit action function.
2. The user should be able to locate edited action.

### Test Use Case

#### Actors

Risk Manager: Edit Action

#### Use Case Scenario: edit action

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk manager navigates to the desired incident |  |  |  |
| 1. Risk Manager locates the desired action and clicks on the pencil action to edit |  |  |  |
| 1. Risk manager saves form |  |  |  |
| 1. Incident’s responsible manager is notified |  |  |  |
| 1. Action Owner is notified |  |  |  |

\*\**Edit the due date of the action to a date before today’s date and note whether the action owner, risk manager and responsible manager are notified via email.*

**Overall Comments**

## Delete Action

### Prerequisites

1. The user should have successfully logged into the system.
2. The system should have existing Incidents.
3. The user should have successfully navigated to the Incident Management dashboard.
4. The system should have existing actions.

### Success Criteria

1. The user should be able to find the delete action function.
2. The user should be able to locate deleted action.

### Test Use Case

#### Actors

Risk Manager: Delete Action

#### Use Case Scenario: delete action

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Risk manager navigates to the desired incident |  |  |  |
| 1. Risk Manager locates the desired action and clicks on the trash action to delete |  |  |  |
| 1. Risk manager saves form |  |  |  |
| 1. Incident’s responsible manager is notified |  |  |  |
| 1. Action Owner is notified |  |  |  |

**Overall Comments**

## Update Action

### Prerequisites

1. The user should have successfully logged into the system.
2. The system should have existing Incidents.
3. The user should have successfully navigated to the Incident Management dashboard.
4. The system should have existing actions that are incomplete.

### Success Criteria

1. The user should be able to find the edit action function.
2. The user should be able to locate updated action marked as complete.

### Test Use Case

#### Actors

Unit Owner: Update Action

#### Use Case Scenario: update action

|  |  |  |  |
| --- | --- | --- | --- |
| **STEP** | **Was Information Readily Available** | **PASS/FAIL?** | **CONCESSION IF FAIL** |
| 1. Unit Owner navigates to the desired incident |  |  |  |
| 1. Unit Owner locates the desired action and clicks on the pencil action to edit |  |  |  |
| 1. Unit Owner changes the status field to complete and saves form |  |  |  |
| 1. Risk Manager is notified |  |  |  |
| 1. Action Owner is notified |  |  |  |

**Overall Comments**