

Test Strategy Branch Management

1. Introduction

This document outlines the test strategy for the Alfardan Exchange Branch Management Module. The module includes functionalities such as branch creation, branch viewing, area creation, area mapping, time extension requests, branch risk register, and day end activity report.

2. Objectives

- Ensure all functionalities of the Branch Management Module meet the specified requirements.
- Identify and resolve defects in the system before deployment.
- Verify the integration of the module with other systems.

3. Scope

The scope of testing includes the following components:

- Branch Creation
- Branch View
- Area Creation
- Area Mapping
- Time Extension Request
- Branch Risk Register
- Day End Activity Report

4. Testing Approach

4.1 Test Levels

- **Unit Testing:** Conducted by developers to verify individual components.
- **Integration Testing:** Ensure different components work together.
- **System Testing:** Verify the system as a whole meets the requirements.
- **Acceptance Testing:** Conducted by end-users to validate the system meets business needs.

4.2 Test Types

- **Functional Testing:** Verify each function of the software application.
- **Usability Testing:** Ensure the system is user-friendly.
- **Validation Testing:** Ensure that application is correctly meet the need

- **Regression Testing:** Verify that new code changes do not adversely affect the existing functionality.

4.3. Test Environment

- **Hardware:** Specify the hardware requirements.
- **Software:** Specify the software requirements, including operating systems, browsers, and database systems.
- **Test Data:** Prepare test data that covers all possible scenarios.

4.4. Test Tools

- Tools and frameworks to be used for testing (e.g., JIRA for bug tracking, slack for communication)

5. Risk Management

Identify potential risks and mitigation strategies:

- **Resource Availability:** Ensure backup resources are available.
- **Environment Issues:** Have a backup environment ready.
- **Delayed Deliverables:** Regularly review the schedule and adjust as needed.

6. Roles and Responsibilities

Identify and allocate resources, including:

- **Testers:** Assign specific testers for each functionality.
- **Developers:** Ensure developers are available for fixing defects.
- **Tools:** List the tools required for testing (e.g., JIRA for bug tracking, slack for communication)

7. Test Deliverables

Artifacts to produced during testing

- Test plans
- Test cases

8. Schedule and Milestones

Create a detailed schedule that includes:

- Test planning
- Test case development
- Test execution

- Defect tracking and resolution
- Test reporting
- Final sign-off