Draft Test Plan
Of
Electronic Remittance (eRemit)
[File Processing Module]

Prepared By
Md. Shahidul Islam
Date: 23 April 2020

Table of Contents

Document Control and Distribution History

Title: Draft Test Plan

Revision History

Rev No.	Date	Description	Author	Review & Approve
1	22 April 2020	Draft Test Plan	Md. Shahidul Islam	Ashfaqur Rahman Tahashin
2	23 April 2020	Draft Test Plan	Md. Shahidul Islam	Sultan Ahmed Sagor

1. INTRODUCTION

A remittance is a transfer of money, often by a foreign worker to an individual in their home country. Money is sent home by migrants competes with international aid as one of the largest financial inflows to developing countries. **eRemit** system is being designed to track and handle the transactions of remittance send by the foreign workers. The file processing module process the file of transactions list with legal info and AML verification and then finalize the transaction.

1.1. Purpose

This Test plan describes the plan for testing the architectural prototype of **eRemit** system file processing module. The test plan document supports following objectives:

- Execution Strategy: describes how the functional and nonfunctional test of file processing module will be performed and to fix and implement fixes.
- Estimate of the test efforts.
- List the deliverable elements of the test activities.

1.2. Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- Test Lead Plans for the testing activities in the overall project schedule, reviews the documents.
- The stakeholders' representatives may take part in the UAT test to ensure the business is aligned with the results of the test.

2. TEST STRATEGY

2.1. Test Objectives

The main objective of the test suite is to provide adequate coverage metrics, requirements validation and system quality data such that sufficient data is provided for those making the decision to release.

The final product of the test is to fold:

- File Processing module should be in production level
- A set of stable test scripts that can be reused for Functional and UAT test execution.

2.2. Test Assumptions

Key Assumptions:

- It is assumed that AML API is fully functional and it does not require testing.
- Production like data required and be available in the system prior to the start of Functional Testing.
- Functional testing, User Interface Testing and User Acceptance testing will be executed according to test cases.

2.3. Test Principles

- Testing shows presence of defects
- Exhaustive testing is not possible
- Early testing
- Absence of errors fallacy

2.4. Scope and Levels of Testing

2.4.1. Functional Testing

Smoke Testing

Purpose: Smoke testing is perform to comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing.

Testers: Testing Team.

Methods: Manual process

Timing: Smoke Testing performed after software build to ascertain that the critical functionalities of the program are working fine. It is executed "before" any detailed functional or regression tests are executed on the software build.

Regression Testing and Retesting:

Purpose: Retesting for fixed bugs will be done by respective QA once it is resolved by respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

Regression testing is the selective retesting of a software system that has been modified to ensure that any bugs have been fixed and that no other previously working functions have failed as a result of the reparations and that newly added features have not created problems with previous versions of the software.

Testers: Testing Team.

Methods: Manual test and writing automation script by selenium Tools.

Timing: After deployment in the test environment if there are bugs found in the system or change request submitted by the client.

Test Acceptance Criteria:

- Test cases approved and signed-off prior to the start of Test execution.
- Development completed, unit tested with pass status and results shared to the Testing team to avoid duplicate defects.
- Test environment with application installed, configured and ready to use state.

Test Deliverables:

S. No	Deliverable Name	Author	Reviewer
1	Test Plan	Test Engr.	Test Lead
2	Functional Test Cases	Test Team	Test Lead
3	Daily/weekly status report	Test Team/ Test Lead	Test Lead/ Software Architect
4	Test Evaluation Report	Test Lead	Software Architect

Milestone List:

The milestone list is tentative and may change due to below reasons:

- Any issues in the System environment readiness.
- Any change in scope/addition in scope.
- Any other dependency that impacts efforts and timelines.

2.4.2. User Acceptance Test (UAT)

Purpose: This test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.

Testers: UAT is performed by the end users.

Timing: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

Test Deliverables

S. No	Deliverable Name	Author	Reviewer
1	UAT Test Cases	Test Team	Test Lead

2.4.3. Non Functional Testing

Performance Testing

Purpose: Performance testing measures response times, transaction rates, and other time sensitive requirements. The goal of Performance testing is to verify and validate the performance requirements have been achieved. Performance testing is usually executed several times, each using a different "background load" on the system. The initial test should be performed with a "nominal" load, similar to the normal load experienced (or anticipated) on the target system. A second performance test is run using a peak load.

Testers: Testing Team

Performance/Capability Goals (Expected Results) and Pass/Fail Criteria: The following are performance requirements (success criteria) for the performance tests:

1. The average response time (measured by the Time to last byte metric) is less than 3 seconds
2. The worst response time (measured by the Time to last byte metric) is less than 30 seconds
3. The average CPU utilization of the database server is less than 75%
4. The average CPU utilization of the application server is less than 75%
5. Each blade server must be capable of handling 500 concurrent users
6. The maximum number of acceptable server errors, non HTTP-200 status codes on client requests, will be less than 2% of all client requests.

Methods:

Number of Users: Performance testing will run with a maximum of 1000 users. The users will be created in beforehand and be accessible via eRemit Login. Each request will login with a different userID.

Load Profiles: Load profiles describe the loading under which the system operates in terms of how many users are operating the system at once. Load Profiles will be recorded as below:

1. Log In to the eRemit application
2. File Processing: Number of uploading files

Tools: Jmeter, Blazemeter add on, Splunk / Nagios for logging

Test Deliverables

S. No	Reports
1	Statistical Summary: 1. Maximum running concurrent users 2. Total throughput 3. Average throughput 4. Average hits per second
2	Transaction Summary: 1. Total passed transactions 2. Total Failed Transactions
3	HTTP Response Summary

4	Running concurrent users graph
5	Response times graph
6	Server resources utilization summary: 1. CPU utilization graph 2. Physical memory utilization graph 3. Disk I/O utilization graph 4. Network I/O (Kb/s) graph 5. Top 5 memory consumers graph 6. Top 5 CPU consumers graph

2.4.4. Automation Testing

After completion of manual testing and regression testing system is ready for automation testing. To automate the system Selenium webdriver , Python and its various testing tools will be used. Python testing tools:

i. **Robot**
ii. **Pycharm**

Test Deliverables

S. No	Deliverable Name	Author	Reviewer
1	Automation Scripts	Test Team	Test Lead
2	Testing Report for Automated Testing	Test Team	Test Lead

3. Execution Strategy

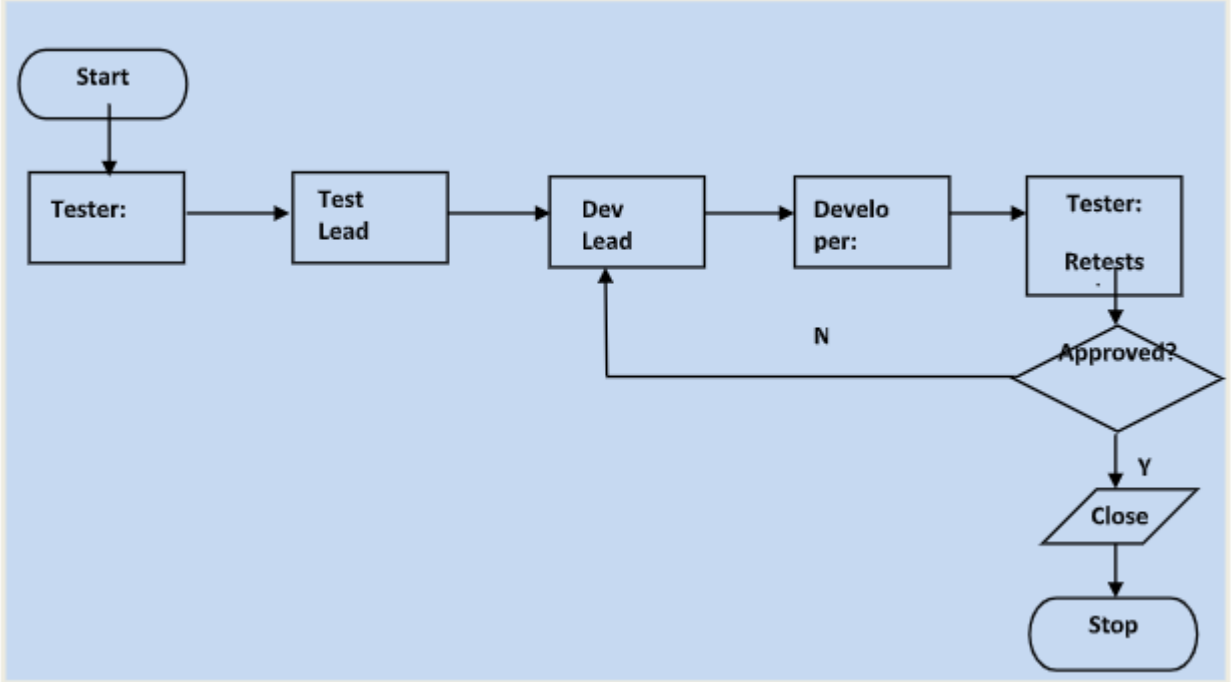
3.1. Entry and Exit Criteria

- Availability of the test environment supporting necessary hardware, software, network configuration, settings and tools for the purpose of test execution.

- Availability of AML API server.
- Presence of proper excel data.
- Requirements should be clearly defined and approved.
- Execution of all test cases.
- All the identified defects are corrected and closed.
- No high priority or severity or critical bug has been left out.

3.2. Defect Tracking and Reporting

Following flowchart depicts Defect Tracking Process:



4. Test Management Process

4.1. Test Management Tool

Trello is the tool used for Test Management. It works on a “Card” basis. All testing artifacts such as Test cases, test results, bugs will be updated in the Trello cards.

4.2. Test Execution Process

- Once all Test cases are approved and the test environment is ready for testing, testers will start exploratory testing of the application to ensure the application is stable for testing.
- Each Tester is assigned Test cases directly written in excel sheets.
- Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the steps directly in the Test Cases sheet.
- Daily Test execution status as well as Defect status will be reported to all stakeholders.
- Testers upload their test cases and bug reports in Trello cards.

4.3. Role Expectation

S. No	Role	Name
1	Test Lead	Ashfaqur Rahman Tahashin
2	Development Lead	Sultan Ahmed Sagor
3	Testing Team	SQA General
4	Development Team	The Intellects

4.3.1. Test Team:

- Develop test conditions, test cases, expected results, and execution scripts.
- Perform execution and validation.
- Identify, document and prioritize defects according to the guidance provided by the Test lead.
- Re-test after software modifications have been made according to the schedule.
- Prepare testing metrics and provide regular status.

4.3.2. Test Lead:

- Acknowledge the completion of a section within a cycle.

- Give the OK to start the next level of testing.
- Facilitate defect communications between testing team and technical / development team.

4.3.3. Development Team:

- Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
- Assist in the validation of results (if requested).
- Implement fixes to defects according to schedule.

5. TEST ENVIRONMENT

5.1. Test Site URL

The following site will be tested by the testers :

Site URL: **http://10.11.201.92:4200**

5.2. Tools List

These following tools will be used to test the system:

Tools Name	Purpose
MYSQL Workbench	Database Monitoring
Trello	Project/Test management
Jmeter	Load and performance testing
Selenium Webdriver	Regression and Retesting automation
PyCharm	IDLE
Splunk / Nagios	Performance testing
Blazemeter add on	Performance testing
Robot Framework	Python Testing tools