|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Name: | **ASD** | | Client Name: | | **First Franklin** | |
| Started on: | 08-05-2024 | | Estimated End Date: | |  | |
| Weekly Review on: |  | | Estimated Hours | | Ongoing (Hrs) | |
| **Objective/Purpose** | The primary objective for a test plan is to produce documentation that describes how the tester will verify that the system works as intended. The document should describe what needs to be tested, how it will be tested, and who’s responsible for doing so. | | | | | |
| Requirements Document | Received? | Yes (User Stories) | | Knowledge Transfer? | |  |
| **In scope of Project** | | | | **Out Scope of Project** | | |
| **Source Data:**  **1.File retrieval from LN**  1.1 Receive files from the SFTP server specified by Lexis Nexis.   * 1. Frequency: Daily   2. Generate daily report of files read.     **2.Update ASD DB on SQL server.**  2.1 All files received from LN shall be processed and updated into the ASD database on the SQL server.  2.2 Frequency: Daily.  2.3 Create an active dataset of all customers with the latest information as new data is received.  2.4 Generate daily report of data loaded into DB.    **3.Match the ASD DB information with GPS Data / NLS Data**  3.1 Add additional columns to the active dataset created in 2.3 with information from GPS. Information like Number of accounts, Type of loans, collateral etc.    **4.Prepare customer file from GPS/NLS**  4.1Send weekly customer files to LN.  4.2 Once the file is received from LN, feed it into active ASD DB    **5.Define BK workflow on the FA app**  5.1 ASD workflow  5.2 Branch workflow    **6.Application**  6.1 Develop an application for every stage of the workflow.  6.2 Define the owners at every stage.  6.3 Define clear status.  6.4 Define passage of tasks.  6.5 Track action and time taken.   * 1. Track the user at every stage.     **7.Alerts and Reminders**  7.1 Daily task alerts – new BK case and events  7.2 Regular alerts/mails to respective stakeholder   * 1. Reminders on non-completed tasks   **8.Reports**  8.1 Generate required reports at every stage.  **9.User Level access**  9.1 ASD User   * 1. Branch User   **10.Define Multiple scenarios: Rule Engine**  10.1 BK Events  10.2 BK Active Cases  10.3 Multiple accounts  10.4 Pending task | | | |  | | |

**Introduction**

Develop a workflow to facilitate the communication and tracking of bankrupt cases between ASD and branches. The project involves developing an application that is designed to cater to the smooth and lean functioning of the ASD – Bankruptcy process.

The application will comprehensively monitor cases from initiation to resolution, providing essential alerts and reminders to all stakeholders involved. Its primary goal is to assign accountability and meticulously track the progress of the follow-through process for identified bankrupt customers.

This document covers the following aspects of testing

* Define the overall Scope and Testing approach!
* Define the testing tools and environments to be used to support testing.

Testing Feature in scope for release 1 are as follows:

* Track new Cases-Begin with the Workflow
* Update Case and Add Loan Document
* Review Submitted Document
* Upload POC
* Download POC
* Close Work Flow

# Testing Scope for the Release

## In Scope

* System Testing – ST (Functional tests)
* System Integration Testing – SIT
* Smoke and Sanity Testing
* Automation Testing for Regression tests
* Basic Web Page Performance Testing
* Performance Testing (Load Testing/Stress Testing) - Depending on Client requirement.

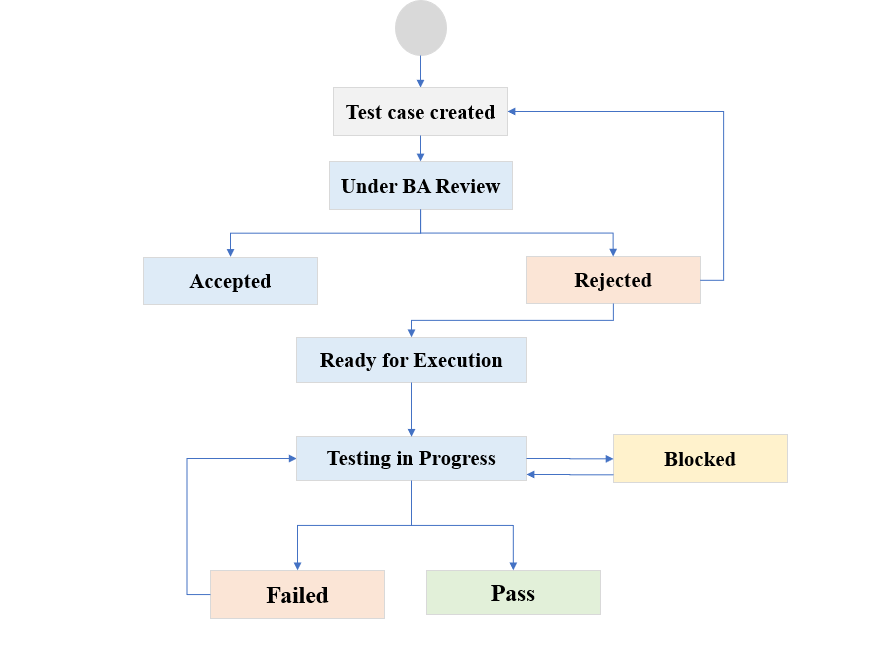
|  |  |
| --- | --- |
| **Testing Type** | **Uses** |
| Smoke and Sanity Testing | Smoke Testing is a preliminary test to determine if a software build is stable enough for further testing, while Sanity Testing verifies recent changes or fixes to ensure they haven't adversely affected core functionalities. |
| Functional Testing | Functional testing verifies that a software application's features and functionalities work as intended and meet the specified requirements. |
| Integration Testing | Integration testing ensures that individual software components or modules work together cohesively as a group, detecting any interoperability issues. |
| System Testing | System testing evaluates the complete software application in its entirety to ensure that it meets the specified requirements and functions correctly in its intended environment. |
| Re-Testing and Regression Testing | Retesting checks if specific issues reported earlier have been fixed, while regression testing verifies that new changes haven't introduced unintended side effects by retesting existing functionalities. |

**Defect Priority Definition and Risk Analysis**

|  |  |
| --- | --- |
| **Define Defect Priority Levels** | |
| **Assignment** | Assign the defect to the appropriate team |
|  | Development team |
|  | Testing team |
|  | BA |
| **Resolution** |  |
|  | Code fix |
|  | Unit testing |
|  | Code review |
|  | Merge to main branch |
|  | Deploy to testing environment |
|  |  |
| **Verification** | Verify the fix #Verification |
|  | Perform regression testing |
|  | Confirm defect is resolved |
|  | Update defect status in tracking system |
|  |  |
| **Closure** |  |
|  | Document resolution details |
|  | Update defect tracking system |
|  | Communicate resolution to stakeholders |
|  |  |
| **Post-Defect Analysis** | Conduct post-defect analysis |
|  | Identify cause of defect |
|  | Determine measures to prevent similar defects |
|  | Update test cases and test plans if necessary |
|  | Share learnings with the team (BA) |
|  |  |
| **Retrospective** | Hold a retrospective to discuss the defect and workflow |
|  | What went well |
|  | What could be improved |
|  | Action items for improvement |

|  |  |
| --- | --- |
| **Risk Analysis** | |
| **Identify potential risks & severity- technical** |  |
|  | High |
|  | Medium |
|  | Low |
|  |  |
| **Prioritize identified risks based on** |  |
|  | Impact |
|  | Probability of occurrence |
|  |  |
| **Defect Analysis** |  |
|  | Compile a list of all known defects |
|  | Prioritize defects based on defined priority levels |
|  | Analyze the potential risk of each defect |
|  | Develop a plan for addressing each defect (test cases) |
|  |  |
| **Implement Defect Fixes** |  |
|  | Assign defects to appropriate team members |
|  | Develop fixes for critical priority defects |
|  | Develop fixes for high, medium and low priority defects |
|  | Review and test all fixes and Implement fixes in the live application |

**Test Case Workflow**



**Test Schedule**

|  |  |  |
| --- | --- | --- |
| **Testing** | **Environment** | **Schedule** |
| Functional Testing | FIRE CONFIGURATOR test server | Functional Testing will be done after every build release |
| SIT | FIRE CONFIGURATOR test server | SIT will be done after unit testing and when new module is added to the system |
| UAT | FIRE CONFIGURATOR test server | Done by Business analyst at the end of every sprint |

**Test Tools**

|  |  |
| --- | --- |
| **Testing Activity** | **Tool** |
| Test Case management | MS Excel |
| Defects Management | MS Excel |
| Test Reporting | MS Excel |
| Performance Testing | JMeter |
| API testing | Postman |

**Test Environments**

|  |  |  |
| --- | --- | --- |
| **Environment** | **Level/Description** | **Usage** |
| DEV | Testing | This environment will be used by developers to build the systems and complete unit and system testing functions prior to integration with other units of code. No systematic controls on code, developers have access to implement code and change setups as needed. This environment often uses stubs or test harness tools to simulate other parts of code. |
| QA | Testing | This environment will be used by QA to verify the Functionality & UI design in desktop and mobile devices, All the recent coding will be committed in this environment and build information will be updated to QA by Dev team |
| UAT | UAT production  Environment | This environment will be used by the client to test the E2E functionality before it goes to production. |

**Defects Management**

The following stages are used to track defects as soon as they are identified:

1. **Open**: After a defect is created, the issue is marked as open and can be assigned to the Developer to start working on it.

2**. In Progress**: The developer actively starts working on the issue according to the requirements.

3. **Fixed:** The identified defect is analyzed, and the relevant code is fixed. The developer will change the status when the build is deployed. to developer will Release for testing.

4. **Released to QA**: At this stage, the defect is waiting to be verified by the Tester in the respective testing environment. If the defect retest is successful, it will be closed. If the defect retest fails, it will be reopened and assigned back to the developer.

5**. Reopen**: If the retested defect fails again, the developer will re-analyze the root cause and fix it. From the reopen stage, the fixed defect will be reassigned back to the Tester for retesting.

6. **Released to UAT**: Now, the defect is waiting to be verified by the UAT Tester or client in the Staging/Agreed environment. If the defect retest is successful, it will be closed. If the defect retest fails, it will be reopened and assigned back to the developer.

7. **Closed:** The defect is considered closed when it is confirmed to be working as per the client's requirements.

Note: Closed defects can be re-opened later based on the requirement or if the issue reoccurs.

8. **On Hold:** Defects meeting the following conditions will be changed to the "On Hold" status:

- Defects that can't be fixed in the current sprint and release.

- Defects with dependencies on stories/tasks.

9. **Deferred:** Defects that are not part of the current phase or requirement will have their status changed to "Deferred" for future fixes.

**10. Rejected:** Defects that are not part of the project requirements or were raised incorrectly by the tester will be changed to the "Rejected" status.

These stages help to keep track of defects throughout the development and testing process, ensuring effective defect management in the Scrum framework.

**Defect Workflow**

**A diagram of a process

Description automatically generated**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By** | **Reviewed by** | **Approved By** |
| **Name** | Pushpa B |  |  |
| **Role** | Associate Software Tester-L1 |  |  |
| **Date** |  |  |  |

# Automation Framework

**Key Objective**

The key objective of the automation framework is

* Zero Touch Automation
* Early Defect Detection
* Higher Return on Investment
* Reusable Commerce Regression Test Suite
* Ease of Maintenance
* Script-less Automation enabling Manual Functional Testers to automate test cases via keywords in Spreadsheet