Test PLAN

Project Name: GPTL bank

Prepared by: Suhas W

Table of Contents

[Overview 2](#_Toc58758203)

[Scope 3](#_Toc58758204)

[Inclusions 3](#_Toc58758205)

[Test Environments 3](#_Toc58758206)

[Exclusions 3](#_Toc58758207)

[Test Strategy 4](#_Toc58758208)

[Problem Tracking and Test Tracking Procedures 5](#_Toc58758209)

[Roles/Responsibilities 6](#_Toc58758210)

[Test Schedule 7](#_Toc58758211)

[Test Deliverables 8](#_Toc58758212)

[Pricing 9](#_Toc58758213)

# Overview

As part of the project, ‘TCS’ has asked Ninja to test few functionalities of ‘https://ninja.in” web application.

This document serves as high level test planning document with details on the scope of the project, test strategy, test schedule and resource requirements, test deliverables and schedule.

# **Scope**

The scope of the project includes testing the following features of gptl bank web application.

## Inclusions

* Manager
* New Customer
* Edit Customer
* Delete Customer
* New Account
* Edit Account
* Delete Account
* Deposit
* Withdrawal
* Customer
* Balance enquiry
* Fund Transfer
* Mini Statement
* Customized Statement
* Change Password
* Login & Logout

Above functionality we have to test.

Functional testing scenario will be against the following test data.

## Test Environments

* Windows 10 + Chrome Browser
* Windows 10 + Firefox Browser
* Windows 10 + IE Browser
* Windows 10 + Safari Browser
* Linux + Chrome Browser
* Linux + Firefox Browser

## 

## Exclusions

* All the features except that are mentioned under ‘Inclusions’
* Do not test third party application showing in the page

# 

# Test Strategy

**Functional Testing**

**TCS is communicated to the ninja and understood that we need to perform functional testing for given function in the scope**

We’ll apply below listed techniques to test the application:

**Functional Testing –**

**Step 1 – creation of test scenario and test cases for different functionality given in the scope**

* **Equivalence class partitioning**
* **Boundary values analysis**
* **Decision table testing**
* **State transition testing**
* **Use case testing**

**We will also use our expertise to create test scenario and test cases by following below technique:**

* + **Error guessing**
  + **Exploratory testing**

**We will give priority to the test cases.**

**Step 2- testing process, when we will receive the application for testing :**

* **Perform sanity testing on received application to check whether the application is stable or not.**
* **If sanity fail. Reject the build and wait for the stable build from the ninja.**
* **When receive stable build, we will go for the details testing or in-depth testing**
* **Multiple test resources will be testing the same application on multiple supported environment at the same time**
* **If bug found, report the bug in the “Jira” tool at the end of the day.**
* **As part of testing also perform :**
  + - **Smoke and sanity**
    - **Regression testing**
    - **Usability testing**
    - **UI testing**
    - **Accessibility testing**
* **Repeat all testing until application is good quality as per customer expectation**

**Step 3 – follow the below:**

* **Context driven testing perform the testing based on the context of the project means perform testing as much as possible depending on the criticality of the project.**

**if project is more difficult we will test more the application.**

* **Shift left testing 🡪 testing the application parallel with development stages.**
* **Exploratory testing 🡪 using expertise we perform testing with the normal testing.**
* **End to end flow testing 🡪 from register – login up to the payment and tracking.**

# Problem Tracking and Test Tracking Procedures

**Defect Reporting Procedure:**

During the test execution –

* Any deviation from expected behaviour by the application will be noted. If it can’t be reported as a defect, it’d be reported as an observation/issue or posed as a question.
* Any usability issues will also be reported.
* After discovery of a defect, it will be retested to verify reproducibility of the defect. Screenshots with steps to reproduce are documented.
* Every day, at the end of the test execution, defects encountered will be sent along with the observations.

Note:

* Defects will be documented in a word document.
* Test cases will be documented in an excel document.

Roles/Responsibilities

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| John Ibrahim | Test Manager | * Escalations |
| A. Don | Test Lead | * Create the Test Plan and get the client signoffs * Interact with the application, create and execute the test cases * Report defects * Coordinate the test execution. Verify validity of the defects being reported. * Submit daily issue updates and summary defect reports to the client. * Attend any meeting with client. |
| Suhas Wankar | Senior Test Engineer | * Interact with the application * Create and Execute the Test cases. * Report defects |
| Wankar suhas | Test Engineer | * Interact with the application * Create and Execute the Test cases. * Report defects |
| suhas | Test Engineer | * Interact with the application * Execute the Test cases. * Report defects |
| Wankar | Test Engineer | * Interact with the application * Execute the Test cases. * Report defects |

# 

# Test Schedule

Following is the test schedule planned for the project –

|  |  |
| --- | --- |
| Task | Time Duration |
| * Creating Test Plan | 20/09/2022 to 24/9/2022 |
| * Test scenario and test cases | 26/9/2022 to 5/10/20122 |
| * Test cases review | 6/10/2022 to 8/10/2022 |
| * Test Case Execution | 9/10/2022 to 30/10/2022 |
| * Summary Reports Submission | 1/11/2022 |

# Test Deliverables

The following are to be delivered to the client:

|  |  |  |  |
| --- | --- | --- | --- |
| Deliverables | Description | Responsible Owner | Target Completion Date |
| Test Plan | Details on the scope of the Project, test strategy, test schedule, resource requirements, test deliverables and schedule | xyz | 1/1/2023 |
| Functional Test Cases | Test Cases created for the scope defined | xyz | 1/3/2023 |
| Defect Reports | Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis. | xyz | NA |
| Summary Reports | Summary Reports –  Bugs by Bug#,  Bugs by Functional Area and  Bugs by Priority | xyz | NA |

# Pricing

Will be shared by Project Manager separately.

# Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

**Requirement Analysis**

Entry Criteria:

* Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

* List of Requirements are explored and understood by the Testing team
* Doubts are cleared

**Test Planning**

Entry Criteria:

* Testable Requirements derived from the given Requirements Documents or Project details
* Doubts are cleared

Exit Criteria:

* Test Plan document (includes Test Strategy) is signed-off by the Client (Ninja)

**Test Designing**

Entry Criteria:

* Test Plan Document is signed-off by the Client

Exit Criteria:

* Test Scenarios and Test Cases Documents are signed-off by the Client

**Test Execution**

Entry Criteria:

* Test Scenarios and Test Cases Documents are signed-off by the Client
* Application is ready for Testing

Exit Criteria:

* Test Case Reports, Defect Reports are ready

**Test Closure**

Entry Criteria:

* Test Case Reports, Defect Reports are ready

Exit Criteria:

* Test Summary Reports

# Suspension and Resumption Criteria

Based on the Client decision, we will suspend and resume the Project.

We will ramp up and ramp down the resources as per Client needs.

# Tools

The following are the list of Tools we will be using in this Project:

* JIRA Bug Tracking Tool
* Mind map Tool
* snipping Screenshot Tool
* FlashBack Video Recorder
* Word and Excel documents

# Risks and Mitigations

The following are the list of risks possible and the ways to mitigate them:

Risk: Non-Availability of a Resource

Mitigation/solution: Backup Resource Planning

Risk: Build URL is not working

Mitigation/Solution: Resources will work on other tasks

Risk: Less time for Testing

Mitigation/Solution: TCS will ramp up the resources based on the Client needs dynamically

# Approvals

TCS will send different types of documents for Client Approval like below:

* Test Plan
* Test Scenarios
* Test Cases
* Reports

Testing will only continue to the next steps once these approvals are done.