Modi Corp Billing System

**Test Strategy for UI**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| 24-01-2023 | V1.0 | Mayank Mahadane | Initial draft version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Scope 3](#_Toc534636696)

[2. Test Approach 4](#_Toc534636697)

[3. Test Environment 5](#_Toc534636698)

[4. Testing Tools 6](#_Toc534636699)

[5. Release Control 7](#_Toc534636700)

[6. Risk Analysis 8](#_Toc534636701)

[7. Review and Approvals 9](#_Toc534636702)

# Scope

Modi Corporation Billing system will communicate with banking servers to recharge the wallet for adding funds. Also, it will communicate with billers for paying unpaid bills, getting unpaid bills. There will be a portal for end customers where they can register and make all type of utility bill payments. Below functionalities will be covered in our testing.

**Functionalities –**

1. Register new user on Billing portal
2. Get all billers from the system
3. Get unpaid bills for utilities
4. Wallet for registered users
5. Wallet recharge and balance enquiry
6. Bill payment from wallet
7. Login for registered users (Assumed functionality)

As part of initial testing, the above functionalities will be tested manually as per Signed off requirements.

**Out of Scope –**

Anything which is not mentioned in this document.

This document has to be reviewed by Product Owner and Business Team and should be approved with comments wherever required. Version Info should be updated each time.

# Test Approach

**Type of testing to be performed –**

1. System Testing - Manual
2. Integration Testing - Manual
3. Regression Testing – Manual/Automation

Test cases to be written in BDD format and should be reviewed and approved by Business Team. Coverage matrix to be prepared and updated timely manner.

Credit Card/Debit Card/Netbanking credentials should be provided for testing purpose. Team should not use any other medium to make any kind of payment.

**Team Members -**

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| Mayank Mahadane | Senior Automation Engineer |
| ABC | Automation Engineer |
| XYZ | Tester |

**Defect Management –**

All bugs should be logged in JIRA and all evidences should be attached. SLA for bug fixes should be defined by Product Owner and Client side application manager.

**Test Management –** To be confirmed by project management

# Test Environment

There will be 2 environments for testing as below –

1. **QA –** This environment will be used for new build testing and all the bugs found on this environment should be reported, get fixed and retested.
2. **Staging –** This environment will be replica of Production setup but will have testing data. This environment should be used for all regression and automation testing.

**Database -**

1. **QA –** This database will have all data related to application’s QA environment.

1. **Staging –** This database will have all data related to application’s Staging environment.

# Testing Tools

* **Automation Tools –**
  1. Selenium
  2. Cucumber
  3. Apache POI
  4. Extent Reports
  5. IntelliJ Idea as IDE
  6. GitHub
  7. Jenkins CI/CD
  8. JDK/JRE
  9. Maven
* **Functional Testing Tools –**
  1. Chrome/Edge/Firefox/Safari browser
  2. MySQL

# Release Control

* Release management plan with appropriate version history that will make sure test execution for all modification in that release

# Risk Analysis

1. If QA server has any technical issues so Staging environment can be used for the testing purpose until QA server is resolved and vice versa.
2. In case of conditional sign off, prior approval from stakeholders should be taken and documented.
3. If QA DB is down then mapping to Staging DB should be done and vice-versa.

# Review and Approvals

* All these activities are reviewed and sign off by the business team, project management, development team, etc.
* Summary of review changes should be traced at the beginning of the document along with approved date, name, and comment