**Software Testing Plan**

***LendingFront***

***Fecha: [26/04/2020]***

**Table of contents**

Version History 3

Project Information 3

Approvals 4

Summary 4

Scope of the tests 4

Test Elements 4

Regression tests 5

Functionalities that should not be tested 5

Test approach 5

Acceptance or rejection criteria 5

Suspensión criterio 6

Deliverables 6

Resources 6

Enviroment requirements – Software 6

Required testing tools 6

Personal 6

Planification and organization 6

Testing procedures 7

Timeline 7

Dependencias y Riesgos 7

References 7

# Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Versión** | **Autor** | **Compnay** | **Description** |
| **26/04/2020** | **1.0** | **Jhoan Lopez** | **LendingFront** | **Initial Document** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Project Information

|  |  |
| --- | --- |
| Company | LendingFront |
| Project | Lending Platform |
| Date | 26/04/2020 |
| Company Manager | Dario Vergara // John Roa |

# Approvals

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name and Lastname** | **Position** | **Area** | **Date** | **Firm** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Summary

LendingFront’s platform provides its clients with a set of portals designed for businesses that need fast access to capital. LendingFront makes available to its clients traditional and nontraditional data sources to improve the loan decisioning process. LendingFront understands the importance of QA in the software development life cycle. It is imperative that the platform is fully tested before it is made available to its clients. At the same time, the company has an aggressive schedule for launching new products. The testing cycle needs to be short, reliable and efficient.

# Scope of the tests

## Test Elements

* **Inscription process**
* **Login**
* **Check current status of process(Solicitude)**
* **Loans**
* **Loan balance**
* **Payment activity**
* **Language and currency**
* **Reports**

## Regression tests

* **Loans**
* **Loan balance**
* **Payment activity**

## Functionalities that should not be tested

* **Login**

## Test approach

Traditional software development methodologies work on the premise that software requirements remain constant throughout the project. But with an increase in complexity, the requirements undergo numerous changes and continuously evolve. In Agile methodology, software is developed in incremental, rapid cycles. As soon as iteration is completed, the entire system is subjected to testing. Feedback from testing is immediately available and is incorporated in the next cycle. The testing time required in successive iteration can be reduced based on the experience gained from past iterations.

Interactions amongst customers, developers and client are emphasized rather than processes and tools. The agile methodology focuses on responding to change rather than extensive planning.

# Acceptance or rejection criteria

Acceptance:

* Unit tests
* Functional tests passed more than 90%
* Coverage of all components
* Corrected defects

Rejection:

* Open defects
* No unit tests
* Failed functional tests

## Suspensión criterio

* Blocking defects
* No testing environment

# Deliverables

* Test plan document
* Test cases
* Logs
* Evidence
* Project closure document

# Resources

## Enviroment requirements – Software

* Access to lending Platform
* Databases

## Required testing tools

* Oracle sql developer
* Selenium
* Maven
* Bug Tracker
* Jira

## Personal

List of personnel required to complete the testing activities, specifying their roles, for example:

One (1) test leader

Five (3) test analysts

Two (2) specialists in test automation, among others.

# Planification and organization

## Testing procedures

According to the established methodology, 3 sprints must be performed each lasting 2 weeks, after the sprint, feedback must be performed.

## Timeline

Imagen example

## Dependencies and risks

* Dependencies with developments.
* Dependencies with other projects.
* Availability of resources.
* Time restrictions.
* Premises that turn out not to be true.

Risks can be classified according to their probability and impact, each must consider a mitigation plan to prevent it from occurring or a contingency plan when the risk cannot be mitigated and has to be accepted.

# References

File “Senior Automation Enginner.docx”.