Guru99 Banking Project

- Test Plan -

Revision History

Date	Description	Author	Comments
27.03.2024	Test Plan for Guru99 Web Version 1.3	Batrinu Diana	Test plan version 1 (initial version)
03.04.2024	Added more details for 2.4 Test Implementation	Ioana Popescu	Test Plan version 1.1

Table of Content:

1. **Introduction**

1.1 Project objective

1.2 Functionalities in scope

1.3 Functionalities and tests out of scope

2. **Test process**

2.1 Test planning

2.2 Test analysis

2.3 Test design

2.4 Test implementation

2.5 Test execution

2.6 Test closure

2.7 Test monitoring and control

3. Test deliverables

3.1 Test plan

3.2 Test conditions

3.3 Test cases

3.4 Daily test summary reports

3.5 Traceability matrix

3.6 Test case results

3.7 Bugs report

3.8 Test completion report

1. Introduction

The Guru99 Bank project aims to provide net banking facility to its customers.

This release will have limited features. Over a period of time, new and new functionalities will be added to the site.

1.1 Project Objective

The objective of this project is to increase the trust in the quality of the application, to find defects in the live software application, to evaluate the product risks of the application, and to send back to the stakeholders information with regards to the testing process and the level of quality of the application after the improvement process. Also, all project members will have a clear understanding about what is tested and what is not.

Application under test:

https://demo.guru99.com/V4/manager/Managerhomepage.php

Application documentation:
https://docs.google.com/document/d/1rPW5DV82VJT6vtA1VDSrfxaCBuAduxW0zb1yfTh_VMk/edit

1.2 Functionalities in scope

For this version of the application the functionalities in the scope of testing are:
Some of the features of Manager module for the Web application which were defined in Guru99 Banking Project software requirements specification:

- ✓ New Customer
- ✓ Edit Customer
- ✓ Delete Customer

From the point of view of the testing techniques we are going to use mostly blackbox testing with the following test design techniques:
- equivalence partitioning
- boundary value analysis
- decision table

From a testing type perspective we are going to use non-functional testing, where we are going to cover only usability testing and compatibility testing.

Also, positive testing and negative testing are to be done, and (according to the needs) retesting and regression testing will be done when defects are going to be fixed or when modifications of any type are going to be brought to the code.

The testing process will be focused on the Chrome browser version 27 and above

1.3 Functionalities and tests out of scope

From the perspective of the modules covered, any other functionality that is located outside of the specified modules are not to be tested.

We are not going to cover during the testing process any techniques related to whitebox testing.

Also, security testing and non-functional testing like performance, stress, volume or logical database testing will not be performed during this session of testing.

Test process for mobile application is out of scope.

Other browsers except Chrome are out of scope.

Automation testing is beyond scope.

2. Test process

2.1. Test planning

Roles and responsibilities

Role 1	Batrinu Diana- Test lead - Will monitor the proper functionality of the test process, the involvement of the teams and the reach of the defined deadlines
Role 2	Tester 2 - Junior Tester - Will write the test conditions and test cases for: New Customer, Edit Customer, Delete Customer
Role 3	Tester 5 - Tester - Will execute the test cases for New Customer, Edit Customer, Delete Customer and report the defects

Entry criteria:

- Testing environment is up and running
- Business requirements are completed by the analysis team and are delivered to the appropriate testing team for evaluation
- Potential project risks are detected and mitigated
- Roles and responsibilities are allocated
- Test plan should be finalized before entering the next phase of testing

- Define the objectives of testing and the accepted level of quality

Exit criteria:

- All the test cases were executed, run rate is mandatory to be 100%
- 90% or more of the test cases are passed
- All detected errors have been reported and closed
- No critical issues have status open
- The project budget was reached
- The project deadline was reached
- The remaining defects/bugs have low severity and low priority
- Test completion report has been created and sent to the stakeholders
- Product risks have been identified and mitigated

Project risks:

- The team does not have the proper knowledge or experience in order to guarantee the desired level of quality for the application
- Not enough time has been allocated in order to properly test and cover all the functionalities in scope
- All that the data that is going to be used will have to be created explicitly in the scope of testing, which will cut off from the time allocated for testing, generating a risk of not reaching the deadline
- Test environment not working
- Due to the company changes we might have levers

Product risks:

- All the data that is going to be used will be test data, which will not give us an experience of the application close enough to the ones that the user will experience
- Taking into account that only 3 modules are in the scope of testing, the rest of the modules will still be at risk of not fulfilling the user needs
- Validation constraints on some of the fields might be too restrictive

2.2 Test analysis

- Analyze the business requirements to make sure that we have all the details for creating the test conditions
- Write test conditions that will be tested in our testing process

2.3 Test design

In this phase we will create the test cases based on the previously defined test conditions

2.4 Test implementation

- Verify if the test environment is up and running
- Access to the test environment is given: valid username and valid password
- Create test suites (Test cycles)
- We prioritize the test cases based on risks and business priority

2.5 Test execution

- All functional test cases cases created in Test Design phase are executed and we set the test case status(passed/failed)
- For the test cases failed we will raise defects
- Regression pack has been ran

- Retesting for the bugs that are fixed by the developers to validate that they issues are not reproduced

2.6 Test closure

- Analyze if the exit criteria were met and the testing process can be closed
- Generate the traceability matrix
- Generate the test completion report

2.7 Test monitoring and control

In this phase various periodic reports will be generated to reflect the current status of the testing process. In case of major problems, control measures could be taken.

3. Test deliverables

The following test deliverables will be provided by the end of the testing process and sent to the stakeholders in order to create the basis of informed decision:

- Test plan
- Test conditions
- Test cases
- Daily test summary report
- Traceability matrix
- Test case results
- Bugs report
- Test completion report

4. Schedule

- The testing process will take place over a period of 1.5 months and will involve all the activities defined in the previous section
- All the resources will be adapted accordingly in case new testing resources are detected as necessary