**Quality Assurance Test Plan**

| **WGU Student ID** | 010771133 |
| --- | --- |

# A. Overview

## 1. Software design plan summary

The software design plan in question is the proposed solution to the problem of incorrect financial data being retrieved to use to generate a loan profile for the applicants. The main issue is that the web application is most likely using incorrect logic when selecting the fiscal years used to generate loan profiles. Instead of selecting the most recent 5 years of fiscal data, the web application is using the earliest first 5 years of fiscal data. The proposed solution will be a logic written in Python in Visual Studio IDE using object-oriented programming principles. GitLab will be used for version control and for continuous integration and deployment practices. Agile is our methodology of choice for continuous feedback. The proposed solution is a creation of a new logic script that will replace the old logic used for selecting relevant fiscal years, with the new logic script being able to select the correct relevant fiscal years used for loan profiles.

## 2. Functional requirements objective

The functional requirements objective is to make sure the web application is correctly identifying and retrieving the relevant 5 years of fiscal data. This means that for applicants that are 5 years or older, it is requesting the most recent 5 years of fiscal data. For applicants that are younger than 5 years, it is retrieving the available fiscal data they do have and then requesting additional forecasted fiscal data to make in total 5 years of fiscal data.

### 2a. Functional requirements objective metrics

The metrics Data Accuracy Rate, System Response Time, Error Rate in Data Retrieval, and Business Age Validation Success Rate will be used to evaluate the functional requirements.

Data Accuracy Rate will measure how often the correct fiscal years are retrieved and the target value should be 100%.

System Response Time will measure how quickly the web application retrieves the data, calculates the fiscal years, and displays back the information. The target response time is under 10ms.

Error Rate in Data Retrieval will be used to measure how often the incorrect fiscal years are being retrieved, and the target for this metric is 0%.

Business Age Validation Success Rate will be used to measure the percentage of cases where the system correctly differentiates between businesses younger than 5 years and businesses 5 years and older. The target percentage for this metric is 100%.

## 3. Non-functional requirements objective

The non-functional requirements objective is to make sure the web application is secure and accessible. The application should be readily available and should not have a significant amount of downtime. The data gathered from this application should also be secure and only accessible by the relevant parties.

### 3a. Non-functional requirements objective metrics

The metrics System Uptime Percentage, Number of Security Breaches, and Data Encryption Coverage Percentage will be used to evaluate the non-functional requirements.

System Uptime Percentage will measure the percentage of the time the web application is available and not experiencing downtime. The target value for this metric is 99%.

Number of Security Breaches will be used to measure how many times unauthorized access happens and the target value is 0 breaches per year.

Data Encryption Coverage Percentage will be used to measure how much of the confidential and sensitive data gathered is encrypted both in storage and during transmission. The target value for this metric is 100%.

# B. Scope

## 1. In-scope functional requirements

* successful and accurate identification and retrieval of the 5 most recent fiscal years for loan profiling for applicants that are 5 years or older
* successful and accurate identification and retrieval of available fiscal years and necessary forecasted fiscal years for applicants that are younger than 5 years

## 2. In-scope non-functional requirements

* create a secure authentication login process to prevent unauthorized access
* host the web application and make sure that it can maintain operations even under high load

## 3. Out-of-scope functionalities

### 3a. Out-of-scope functionalities explanation

* Using Predictive AI to Generate Forecasted Fiscal Years Based on Previous Data
  + integrating a new feature that can generate the necessary forecasted fiscal years based off of existing fiscal year data and other business data supplied by the applicant. This will make it so applicants no longer need to supply their own forecasted fiscal data if their business is younger than 5 years old. It is out-of-scope because this feature is not necessary for the completion of this ticket and doesn’t resolve the issue identified in the ticket
* Optimize Data Retrieval Process to Minimize Resources Used
  + find ways to optimize the data retrieval process so that there is less processing power used when doing data retrievals. This is out-of-scope because it doesn’t address the main issue of incorrectly requesting the wrong fiscal years for loan profiling.

# C. Test Strategy

## 1. Testing overview

| **Test Case Table** | | | | |
| --- | --- | --- | --- | --- |
| **Test Type** | **Description of Test** | **Objective** | **Test Owner** | **Environment** |
| Unit | Test the retrieval logic to see if it will correctly gather and identify the most recent 5 fiscal years of the business for loan profiling for businesses 5 years of age or older  Input: Business creation year and current year  Expected Output: Most recent 5 years of fiscal data for the business  Sample Input: 01/01/2000; 2025  Sample Output: 2025, 2024, 2023, 2022, 2021 | To make sure that the web application logic will select the most recent 5 years of fiscal data for businesses over the age of 5 | Software Developer | Visual Studio IDE  Development Environment |
| Unit | Test the retrieval logic to see if it will correctly gather and identify the available fiscal years and appropriate forecasted fiscal years of the business for loan profiling for businesses under 5 years of age  Input: Business creation year, current year  Expected Output: Available years of fiscal data, and enough forecasted fiscal years for a total of 5  Sample Input: 01/01/2023; 2025  Sample Output: 2023, 2024, 2025, 2026 (forecasted), 2027 (forecasted) | To make sure that the web application logic will select the available fiscal years and necessary forecasted fiscal years of a business under the age of 5 | Software Developer | Visual Studio IDE Development Environment |
| Security | Test the authentication process to make sure that it correctly validates provided credentials and provides appropriate access  Input: user credentials  Expected Output: homepage with connection and access to relevant information, details, and functions  Sample Input: testUser01; testUserPassword01  Expected Output: successful login | To make sure the login page and authentication process correctly authenticates users and connects it with appropriate access and roles | Quality Assurance Engineer | Selenium Testing Environment |
| Performance | Test that the web application functions as intended under expected user load and ensures uptime is maintained  Inputs: Various Numbers of Users  Expected Output: Latency is under 2 seconds. No crashes.  Sample Input: 1000 Users  Sample Output: Average Latency is 0.89 seconds and no crashes | To test that the web applications continues to function at acceptable performance levels under various levels of load and maintains 99% uptime | Quality Assurance Engineer | Apache JMeter and LoadRunner Testing Environment |

## 2. Sequence of testing

Testing process will be done progressing from low level to higher level functions and features. The sequence of testing should start from the two unit tests that makes sure the logical part of the web application functions as intended and correctly identifies and selects the relevant fiscal years for a business for loan profiling. Both unit tests need to be done and passed before moving on with testing since they are core low level functions needed for the web application to function correctly. The next test step should be the security test for the login authentication process. This will make sure that users are properly authenticated and can proceed to the page where they can supply confidential business data and fiscal years. This is done before the Performance test because the Performance test needs the Security test to be done before it can simulate users. Lastly, the Performance test will be done to test that the web application maintains acceptable performance metrics like latency and maintains 99% uptime under various user loads. It is done last because it is the highest level of requirements and requires the security part to be completed before it can simulate users logging in to the web application. This testing sequence ensures that the web applications core functionality, security, and performance are thoroughly tested.