**Quality Assurance Test Plan**

|  |  |
| --- | --- |
| **WGU Student ID** | 011205284 |

# A. Overview

## 1. Software design plan summary

The Software Design Plan aims to address a significant issue in the Endothon Finance web application’s financial data retrieval logic. The application incorrectly retrieves financial data from the first five fiscal years of a business’s operation instead of the most recent fiscal years. This issue results in inaccurate loan profiles and compromises the reliability of financial assessments used for loan qualification.

**Action Items and Functional Requirements:**

To resolve this issue, the design plan includes the following essential action items:

1. **Update the Data Retrieval Logic:**
   * The application will be modified to retrieve financial data from the most recent five fiscal years for businesses that have been operational for over five years.
   * For businesses with less than five years of operation, the system will retrieve historical financial data and supplement it with forecasted financial projections to create a complete five-year financial profile.
2. **Implement Logic for Special Cases:**
   * Ensure that businesses that are exactly five years old are handled correctly by retrieving the past five years without requesting forecasted data.
3. **Validate Business Establishment Year:**
   * The system will validate the input for the business establishment year to ensure accurate logic flow and prevent errors in data retrieval.

**Non-Functional Requirements:**

The design plan also outlines the non-functional requirements that focus on performance and security:

* **Performance:** The application should retrieve financial data within a maximum of 2 seconds, even under peak load conditions.
* **Security:** All financial data must be encrypted both at rest and in transit, ensuring compliance with data protection standards.

**Software Structure and Deliverables:**

The design structure is broken down into several key modules:

1. **Data Input Validation Module:** Validates the business establishment year before querying financial data.
2. **Data Retrieval Module:** Queries the correct financial data based on business age.
3. **Forecasting Module:** Generates forecasted financial data for businesses younger than five years.
4. **Error Handling and Logging Module:** Captures and logs any errors during the data retrieval process.

The key deliverables include:

* **Updated Financial Data Retrieval Logic:** Refactored logic to correctly retrieve financial data based on the business's age.
* **Forecasting Functionality:** A module to handle financial projections for newer businesses.
* **Testing and Validation:** Comprehensive test cases and scripts to validate the updated logic.
* **Documentation Update:** Technical documentation covering the new logic and expected behavior was updated.

**Development Methodology:**

The development approach follows an Agile methodology, which allows for iterative development with frequent testing and review. This approach ensures flexibility in addressing any unforeseen issues during the bug-fixing process, enabling quick adjustments based on early feedback. Agile was chosen over Waterfall to allow more rapid development and adaptation, especially given the critical nature of this bug fix.

By implementing these changes, the design plan ensures that the loan profiling process will be accurate, reliable, and aligned with the business’s functional requirements.

## 2. Functional requirements objective

The functional requirements are intended to ensure that the web application

accurately retrieves financial data according to the business's operational age.

Specifically:

The system shall retrieve financial data from the five most recently completed fiscal years for businesses that have been in operation for more than five years.

For entities with less than five years of operational history, the system shall obtain all available historical financial data and supplement it with forecasted data for future periods to complete a five-year profile.

### 2a. Functional requirements objective metrics

 **Accuracy:** Ensures the correct financial years are retrieved based on business age.

 **Completeness:** All requested data fields must be fully populated for each loan application.

 **Consistency:** The logic should behave the same across different cases, ensuring uniformity in data collection.

These metrics are essential to maintain the integrity of loan profiles and to avoid errors in financial qualification for loans, which could lead to incorrect loan decisions.

## 3. Non-functional requirements objective

The non-functional requirements emphasize performance, security, and usability to ensure the app operates efficiently while delivering a seamless user experience.

Specifically:

* The app should load promptly, ideally within two seconds, to sustain user engagement.
* It must ensure secure data encryption during financial data retrieval, adhering to established security standards.

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

 **Performance:** The loan application page should load within 2 seconds to maintain a high user experience.

 **Security:** Data must be encrypted at rest and in transit, adhering to security protocols like SSL encryption.

 **Usability:** The interface should be user-friendly, provide clear prompts, and minimize user input errors during the loan application process.

# B. Scope

## 1. In-scope functional requirements

 **Retrieve financial data from the most recent five fiscal years for businesses that have been operational for over five years.**

* This requirement ensures the core functionality of creating loan profiles with accurate and up-to-date financial information.

 **Request forecasted data for businesses younger than five years.**

* This ensures that loan profiles for newer businesses are built with appropriate data, including future projections when historical data is not fully available.

## 2. In-scope non-functional requirements

 **Page load time should be within 2 seconds.**

* This ensures performance efficiency, contributing to a smooth and responsive user experience.

 **Data encryption during transmission and at rest.**

* This ensures compliance with data security standards and protects sensitive financial information during and after transmission.

## 3. Out-of-scope functionalities

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

 **Redesign of the user interface (UI):**

* **Alignment with the business requirement**: While the UI impacts user experience, the current issue concerns backend data retrieval, not the UI.
* **Why out of scope:** The scope of the bug fix only involves backend logic updates and does not require UI changes.

 **Expanding loan offerings:**

* **Alignment with the business requirement:** While changes to loan offerings could influence data requirements, they are outside the scope of the current logic bug.
* **Why out of scope***:* This is a feature-related enhancement, not a bug fix related to the current data retrieval issue.

# C. Test Strategy

## 1. Testing overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Table** | | | | |
| **Test Type** | **Description of Test** | **Objective** | **Test Owner** | **Environment** |
| Unit Test | **In-scope requirement:** Test the data retrieval logic by simulating requests for businesses older than five years. Input business establishment year and verify that the system retrieves the five most recent fiscal years. **Sample input:** business established in 2010;  **Expected output:** financial data from 2018-2022. | Ensure correct financial data is pulled for older businesses. | QA Engineer | Local Development |
| Unit Test | |  | | --- | |  |  |  | | --- | | **In-scope requirement:** Test the logic for businesses younger than five years by simulating requests and verifying that forecasted data is included.  **Sample input:** business established in 2020; **Expected output:** financial data for 2020-2022 and forecasts for 2023-2024. | | Ensure forecasted data is requested for newer businesses. | QA Engineer | Local Development |
| Integration | **In-scope requirement:** Test integration with the database by sending data queries based on business age and verifying that the correct data (historical and forecasted) is returned.  **Sample input:** query for business established in 2015; **Expected output:** data for 2017-2021. | Ensure accurate data retrieval from the database in real-world scenarios. | Database Admin | QA Server |
| Performance | **In-scope requirement:** Simulate multiple user interactions on the loan application page to measure load time.  **Sample input:** 500 users access the page simultaneously.  **Expected result:** page loads in under 2 seconds. | Ensure the page performance meets the 2-second load time requirement. | Performance Tester | Performance Test Tool |
| Security | **In-scope requirement:** Test encryption of financial data both in transit and at rest. **Sample input:** Send a request for financial data and inspect the data packets for encryption status.  **Expected result:** data should be encrypted in accordance with security protocols. | Ensure that sensitive financial data is securely handled and transmitted. | Security Tester | Staging Environment |

## 2. Sequence of testing

 **Unit Testing**: The first step is to ensure that the core logic for data retrieval is correct. Unit tests will verify whether the app retrieves suitable financial years and accurately handles forecast data.

* **Justification:** Unit tests are critical for validating individual components before testing interactions between components.

 **Integration Testing**: After confirming that the logic works in isolation, integration tests ensure that the app interacts correctly with the database and retrieves the proper data fields.

* **Justification:** Integration tests check the interaction between modules, which is necessary after validating the core logic.

 **Performance Testing**: Once the functionality is confirmed, performance tests will verify that the app can handle user load efficiently and meet the 2-second load time requirement.

* **Justification:** Ensuring performance after functionality testing guarantees that the user experience is not compromised.

 **Security Testing**: Finally, security tests will ensure that all sensitive financial data is encrypted and handled securely throughout the data retrieval process.

* **Justification**: After functional and performance tests, security testing is critical to protect sensitive information during operations.