**Quality Assurance Test Plan**

|  |  |
| --- | --- |
| **WGU Student ID** | kbui4 |

# A. Overview

## 1. Software design plan summary

The design plan outlined in this document will address the issue from ticket #D480-AEN1. In short, the loan application page is prompting applicants with well established business for the *first* 5 years of historical fiscal data, rather than the *latest* 5 years of fiscal data.

## 2. Functional requirements objective

As outlined in the design plan, the main 2 functional requirements that need to be addressed are the applications ability to gather the proper fiscal data. Tightly related to that requirement is the application’s ability to generate relevant loan profiles for processing loan applications.

### 2a. Functional requirements objective metrics

Quality metrics related to these requirements are primarily correctness and trustworthiness. The fact that the form is asking for and collecting the wrong data is resulting in incorrect loan profiles. The fact that this issues are so visible to the end user reflects poorly on the organization in general and could cast doubt on the organization as a whole.

These metrics are extremely relevant to the software design solution. If the application is collecting a data, the loan applications can’t be used. If the end users don’t trust the integrity of the software they won’t use it and a source of customers is adversely affected.

## 3. Non-functional requirements objective

One non-functional requirements discussed in the design plan were a well-designed authentication implementation. The application needs to have secure authentication to protect and segregate user data. The other requirement discussed was that the application must be reasonably fast to load and interact with. Slow running applications are aggravating and may put off potential customers. There is also a level of professionalism that is displayed in a performant application.

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

Quality metrics related to these non-functional requirements include compliance and performance. The data this application deals in is extremely sensitive in nature and it is important to comply with established security standards. Furthermore, the software must comply with governmental regulations and standards for data security. Failing to meet these metrics could result in a security breach which would kill organizational reputation and result in legal action. As discussed above, a well performing application is absolutely critical to maintaining a healthy customer base. It is also a reflection of the quality of the organization in general.

# B. Scope

## 1. In-scope functional requirements

Two in-scope requirements to be addressed in this test plan are the the loan application page, and the resulting loan profile that gets produced.

After the fix is implemented the full loan application process will be tested. The displayed controls on the loan application page will be observed to check if they are prompting for the last 5 years of fiscal data, and the backing methods and event handlers will be stepped through to ensure they’re calculating the required fiscal years properly. After the loan application process has been tested and verified correct, the resulting loan profile that gets generated from these inputs will be tested and checked to ensure that the profile contains the fiscal data from the previous 5 years for the business, rather than the first 5 years of the business’ history.

These requirements are in scope of the design plan as they are directly related to the functional requirements outlined in the software design plan, as they govern the process of acquiring loan application data and processing it into a loan profile from start to finish.

## 2. In-scope non-functional requirements

2 in-scope non-functional requirements that will be tested during this QA process are that the pages on the site load in less than 5 seconds overall performance of the site, and that the site properly limits access to users, both logged in and not yet logged in.

Since a user must first log in in order to apply for a loan, we can observe that the authentication and authorization system is working properly. Attempting to apply for a loan before logging in should prompt the user to first log in, and once logged in, and the application should update to display the proper user profile for that user. Furthermore, at no point through the application process should the user have access to internal admin-only portions of the site, and once logged in the user should only have access to their own loan application and saved fiscal data. Finally, throughout the entire testing process, the performance of the site can easily be observed and it can be noted if any page takes longer than 5 seconds to load.

These requirements are in-scope with the previously described non-functional requirements of security and performance. Proper functionality of the log-in process, application access control, and segregation of loan application data is of the utmost importance with regards to organizational security, user privacy, and compliance to governmental regulations and well-established security standards.

Observing the overall site performance throughout the log-in and loan application process directly aligns with the need for the web application to provide a smooth and non-aggravating experience. While it is not currently an issue, it is something that is easy to keep an eye on and we should always be on the lookout for unexpected slow performance, and 5 seconds is a very lenient requirement for such a simple form to be loaded.

## 3. Out-of-scope functionalities

The following functionalities are considered out of scope and will not be addressed during this QA process:

* Automatic fiscal performance projection for businesses established inside of the past 5 years
* Redesigning the look and feel of the loan application form

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

Automatic fiscal performance projection for newly established business could be a very attractive feature, and could draw in more business for the organization. It could provide guidance for newer entrepreneurs, and help develop long term business relationships where both the customer and our organization could thrive together.

However, fixing the actual application process has a much higher priority, and requires much less development time. Trying to implement a brand-new feature while such an urgent and easily fixed bug would not be an efficient use of our time.

Upgrading the look and feel of the loan application form is also directly related to the loan application process. Having an interface that implements the newest technologies, and has the nicest UX features presents a professional face to our clients. Furthermore, making sure to implement things like accessibility features for differently abled users could even open up our business to clients that we would otherwise miss out on.

However, our UX is already built on a relatively modern tech stack, and usability testing of the application in general has been very positive. Furthermore, as we are a professional enterprise, it is easy to go too far the other way. Trying to make the absolute sleekest and shiniest interface can easily go too far and look unprofessional.

# C. Test Strategy

## 1. Testing overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Table** | | | | |
| **Test Type** | **Description of Test** | **Objective** | **Test Owner** | **Environment** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 2. Sequence of testing