**Software Design Plan**

|  |  |
| --- | --- |
| **WGU Student ID** | 002204262 |

# A. Business Case

## 1. Problem Statement

The web application is designed to provide financial data for a business’s history in one of two ways.

1. If a business is older than five years it will provide financial data for the previous five year
2. If the business is less than five years old it will provide financial data for the current life of the business and projections for the remaining years until a total of five years worth of data is available.

However, a recent ticket has been submitted saying that the results of the web application do not provide the desirable outcome for a business. Instead, the web application provides the first five years of financial information for a business instead of the most recent five years.

## 2. Business requirements

1. A user within the web application will be presented with the most recent five years of a businesses.

2. The web application will automatically identify the specific years of financial data based on a set of criteria: the business is older or younger than five years.

## 3. In-scope action items

1. Presenting the five most recent years of financial data for a business that is older than five years.

This is the sole concern of this ticket. When a query is made for a business that is older than five years, the user is presented with financial information for the first five years versus the desired outcome of the past five years.

2. Identifying if a business is older than five years or not to provide the appropriate information.

If a business is older than five years the web application should automatically provide the previous five years of financial data for the business. Appropriate determination of this is essential to fix the current problem. If a business is less than five years old, it does not experience the current issue being addressed.

## 4. Out-of-scope action items

1. Situation two where a business is less than five years old.

Nothing is said in the ticket about the second circumstance or if issues are involved there. It specifically identifies the first circumstance where a business is older than 5 years.

2. Forecasting financial information

The problem being brought forth involves companies that are older than five years old. Forecasting involes projecting future potential business information that may occur for a business that is less than five years old. The problem lies in business’s that are older than five years old, therefore forecasting is way beyond the scope of this problem to fix.

# B. Requirements

## 1. Functional requirements

1. Determining if a business is older than five years or not.

The ticket says that loan profiles are returning the first five fiscal years of financial data, giving the impression that this is occurring for every profile. In order to pinpoint the actual cause of the problem we need to make sure that businesses that are younger than five years are correctly being separated from businesses that are older than five years.

2. If a business if older than five years the loan profile should display financial data for the previous five years from the current year.

The web application currently displays the first five years of a business in the loan profile. However, the desired outcome is to have the previous five years of financial data displayed in the loan profile.

3. Accurate display of data and information.

It is quite in the realm of possibility that the correct financial data for the previous five years of financial data has been correctly obtained, but incorrectly labeled. In the example, it is saying that for a business that was established in 2000, that data is being propagated for 2000, 2001, 2002, 2003, 2004, 2005. However, it should be verified that those data elements actually correspond to those years.

## 2. Non-functional requirements

1. Security and encryption of data.

The web application transmits a large amount of financial data regularly. Security measures to maintain the integrity of this data is paramount for the safe use of this web application.

1. Time to finalize maintenance of the web application.

This web application is already in use. A significant bug has been identified that is vastly affecting business operations because reliable and accurate information is not being propagated. However, the actual cause of this bug is currently unknown, therefore the amount of time required to find the bug and correct it is unknown. This should be taken into consideration during the planning stages of tackling the issue that has been brought forth..

# C. Software Design

## 1. Software behavior

When a user utilizes the web application to obtain loan profile data the software must be able to automatically identify if a business is older than five years or younger than five years and then provide the appropriate data based on the age of the business. The inability to do this at all has put a standstill on the loan application process. Due to the large impact this is having on business processes, the amount of time to fix this issues is of utmost concern. The most positive scenario would involve quickly identifying the cause of the current issue and providing a fix with as little amount of time as possible so that the business can continue to resume operations.

Financial data for the business’s requesting loans needs to be readily available and more importantly accurate. If the financial data itself has been compromised and contains inaccurate data then this issue may not even be caused from something within the web application software.

The current year might be an issue in the web application that may have caused the issue currently being experienced by users. If the current year parameter that is utilized in the software algorithm is not in fact the actual real time year and instead a mistake was made during coding that grabs a year from financial data for whatever reason before doing the age check this could significantly cause issues. The correctly functioning web application in theory should derive the current year from something disconnected from application or data being utilized. For example, the year could be derived from the current time date stamp of the operating system.

## Software structure

Classes will be developed for this web application broken up into four categories: a core class to store business information called Business and a class called Age to determine the age of the business as well as whether the first five years of financial information or a combination of current age financial data mixed with project data should be displayed. Finally, project a young businesses financial data to make the 5 year mark was part of the prior development of the web application to satisfy that business requirement. Therefore a Projection class would need to be implemented in some form, however this is out of scope for the current issue but it is believed that due to being a core requirement before launch that this was implemented in some shape or form in the original code.

# D. Development Approach

## 1. Planned deliverables

A core class from the very beginning is the Business class. It will have data members that contain pertinent information such as name, date created, age, and financial information for each year the business has been alive. If this financial information is stored in an outside database or other location, functions will be created to allow the software to retrieve the information from its current location.

Determining whether a business is older than or younger than five years is critical for this application because each circumstance retrieved different sets of data. Therefore, a separate Age class will be utilized. This class will contain the functions to determine the age and which data to display when a query is made for a business. First, there will be a function retrieveCurrentYear to determine the current year by retrieving the date time stamp of the operating system.

After the current year has been retrieved, a function determineAge will retrieve the date created data member from the designated business. This function then will compare the two to determine the age of the business in the business class’s age data member. This class will also have a desiredFinancials class that will determine whether the financial data displayed should be the first five years of the business or the previous number years plus projected years till a total of five is available. This particular function would require the ability to call a function that will provide projections for a business that is younger than five years, however this is out of scope for the current project and considering the web application has already been released with the intent of this being a part of the web application from the beginning this particular function would have already been developed.

As previously stated a Projection class would be part of the program to project potential earning for a business younger than five years. This was a prerequisite requirement from the the initial launch of this web application, therefore it is currently out of scope for the current dilemma.

## 2. Sequence of deliverables

Creation of the Business class is the first step to resolving this issue, due to the fact that it contains all of the critical data that will be used by other classes to manipulate the data contained within this class to appropriately display the desired information to the user. Once this class, its data members, and more importantly the function to retrieve and store financial data for the business object further development can commence.

Once crucial business data can be saved in the business class the Age class should be developed to determine the age of the business and what financial data for the business should be displayed. Part of this class would utilize another class to project financial data for young business, but this is currently out of scope for this endeavour and furthermore believed to have been developed before launch considering it was a core requirement for the original application.

## 3. Development environment

This is a fix to a bug within the software that has been identified, so obviously consideration should be taken into what the current web application was developed in considering source code for the application would be geared towards that particular program language. Utilizing that particular programming language to fix this particular issue would be useful and time saving considering the issue involved is a bug for an already developed program. Therefore source code is written in a specific programming language already.

However, this developer would prefer to utilize python in a PyCharm environment. Python is a pretty intuitive programming language that has an added advantage of not requiring physical memory management while writing a program.

## 4. Development Methodology

The current dilema brought to our teams attention is for a web application that has already been launched. A particular issue has occurred within the application where appropriate financial data for a business requesting a loan is not being provided. This information is crucial for determining the status of a loan applicant and if the loan will be approved or not. The inability for the web application to accurately provide financial data has caused a stand still in business operations. Therefore, it is of the opinion that a RAD methodology will be most suitable for the issue that has been brought forth. This solution is more focused on fixing the current issue in the least amount of time possible. Implementation of corrections will be vetted by users as early as possible to verify if the issues identified have been resolved.