Insurance Illustrator Specification

Version 1.0

March 1, 2021

Created by:

Sean Atchison

# Introduction

The purpose of this document is to outline the components needed to create a SaaS service that takes information from an http post, allows for some user input, does a few calculations, then displays the results in text and chart format to the user. This is uses as a sales tool for insurance agents.

# Services offered

The service will be hosted on Microsoft Azure and use web hosting and DB components and/or Microsoft Excel object for calculations and graphing (note: If you have other thoughts on more efficient ways to accomplish the calculations and graphs please let me know)

# Risks, Mitigations, Contingencies

A risk is anything that may affect the schedule, cost, timeline or quality of the project. A mitigation is what we are doing to avoid the risk. A trigger is something that tells us that the risk is being realized. A contingency is what we will do if the risk is realized, and the mitigation didn’t work. Passed is the date we no longer have to worry about that risk.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Trigger | Mitigation | Contingency | Passed |
| Time Zone Differences Slow work |  |  |  |  |
| Are developers familiar with AZURE and Excel API |  |  |  |  |
| Customers may have refining input that changes scope |  |  |  |  |

# Business issue we are trying to solve

Insurance agents are selling Index Universal Life (IUL) and Whole Life (WL) insurance policies that have accelerated cash value premium payments as a way to refinance debt by making loans against the insurance policy. There are currently no standard “Illustrations” to help them to make consumers easily see the benefit of this strategy. This software purpose is to calculate out the benefits and illustrate them in the areas of:

1. Interest savings
2. Debt payoff time (Note: This will be payoff by financing into the policy. It will not include repayment into the policy.)
3. Added benefits (death benefit and additional cash value(s))
4. Monthly payment differences

# How does the process work today?

The current process is that agents are left to try to explain and calculate this on their own. There is one illustration software that does this in the market, but for only one small carrier and only for their whole life product.

# How do we want it to work with Debt Eliminator, LLC

An insurance agent (“user”) will be on a carrier site. They will run what is called a basic illustration on that third party site/application. As an example; Male age 35 in good health contributing $10,000 per year for 10 years with minimal death benefit. This will create an amortization schedule in a csv file. The user would then click a link/button in the app “Debt Eliminator”. This will make an http post with the following payload:

* CSV file of amortization schedule
* Policy files in PDF format to include in our Print or email.
* Insurance company UID for the basic illustration
* Client name

User will be brought to Debt eliminator service where they can enter a clients debt information. User then clicks show illustration and is brought to a display page with result of calculations. User can print or email the illustration with supporting documents.

# What are the success metrics?

Delivery of standard HTTP Post call for insurance companies to use.

Creation of the web portal hosted on the Azure service that is fully functional for two initial insurance carriers.

Acceptance testing completed.

Debt Eliminators, LLC will be doing testing.

# Assumptions

Target Chrome and Edge browsers

Web pages should render for Mobile but not be optimized for mobile

Agile mindset – rapid prototyping and feedback loop over documentation. Bias for action.

# Use Cases

1. Making the HTTP post with expected payload – We are to write HTTP Post code for insurance companies to host in their applications/site. That code should have the following elements:
   1. CSV file with product amortization schedule. (\*See appendix for examples)
   2. UID – Generated by the insurance carrier
   3. Proposed Insured (PI) full name
   4. PI Age
   5. PI Sex
   6. PI medical classification
2. Creating the DB – Each insurance carrier will have a slightly different format for their file. We will initially start with just 2 carriers and the CSV formats are in exabits attached.
3. Completing the calculations – the calculations happen at 2 main times:
   1. When entering debts: The user will enter their debts one at a time on the web page. They will enter:
      1. Debt Type: “Consumer”; “Student Debt”, “Real Estate”
      2. Company: String – debtors name
      3. Balance: Dollar Amount
      4. Minimum Payment: Dollar amount
      5. Interest Rate: Expressed annually
   2. For each debt entered, the we will calculate how many years until the debt is payed off if making the minimum payment, and how long it will take to pay it off in years
      1. To calculate the number of payments until paid off is a “NPER” calculation. In Excel this would look like: =ROUND(SUM(ABS(NPER(D2/12,E2\*-1,C2\*-1,0,1))/12),2) or in VBA might look like: <https://docs.microsoft.com/en-us/dotnet/api/microsoft.visualbasic.financial.nper?view=net-5.0>
      2. Store in DB
      3. To calculate the total interest “CUMIPMT”. In Excel this would look like: =IF(L2=0,0,CUMIPMT(D2/12,L2\*12,K2,1,L2\*12,1))

You can find an article for the math here: <http://pistulka.com/Other/?p=1678>

* + 1. Store in DB
  1. To calculate the interest savings;
     1. Sort The debt list by amount owed, lowest first
     2. Look up in amortization schedule what year the first debt amount would be available.
     3. Make a variable (or DB entry ) to store total borrowed from policy. Note: this is a cumulative number.
     4. Calculate total interest and total principal paid on existing loan until it can be paid from policy funds. Subtract the principal amount of payments from total principal and store in DB
     5. Calculate total interest on new balance at 4% annual interest rate
     6. Subtract interest until funds available from total interest on loan.

1. Creating the illustration using excel API (or other???)
2. Creating the web pages

# Work flow(s)

Diagram

Description automatically generated

Diagram

Description automatically generated

Explanation of what happens on each event:

OnHTTPSuccess:

* Call DBTransform
* Redirec user via new tab to DebtEliminator site, debt input page.

OnHTTPFail:

* Error log
* Redirect user via new tab to DebtEliminator site, error page.
* Send email to admin (Sean Atchison)

OnDBTransformSuccess:

* UpdateDB

OnDBTransformFail:

* Error log
* Redirect user via new tab to DebtEliminator site, error page with msg.
* Send email to admin (Sean Atchison)

OnDebtEntry:

* UpdateDB
* Update Table on page

OnDebtEdit:

* Show Edit and Remove Icon next to each table entry

OnEdit:

* + Populate that row into edit boxes
  + UpdateDB and table

OnRemove:

* + Remove that debt entry
  + UpdateDB and table

OnShowIllustration:

* Call IllustrationCalculation
* UpdateDB
* Redirect user to Illustration page

OnCalculationSuccess:

* Create amounts and graphs for illustration page

OnCalculationFail:

# Branding

* We will need to hold a few colors and up to two branding images
* Based on the Post initiation, we will display the proper branding on the site
* There are currently no branding images or details

# Schedule (Phases and Key Dates)

TBD for details.