James Elam

SWDV 691-1

Application: Student Loan Calculator

I. Problem

Over the course of this Capstone, I will be planning and developing the “Student Loan Calculator” application. The primary goal is to allow users to enter their loan information and be given simple numbers and a visual representation of their loan repayment plan. “Alternative plans” can be created to compared repayment plans with the “current” plan. As discussed in my product pitch video, many students are taking a submissive approach to repayment. They jump on their student loan site, enter their minimum auto-payment information, and never think about it again. Many of these same students are jumping into well-paying post-college jobs and have the income to pay over the minimum. I believe that there is a general unawareness to the benefits and there are few tools that give the borrower insight about the entire loan and the ripple-effects of squeezing a little more money into their auto-payment.

This application will mostly target current and post-college students. However, there will be great effort to develop the application loosely enough to support any type of loan in future features. This would broaden the target demographic to anyone with a credit history. In the current iteration, a typical user persona could be derived from any ethnicity, however, likely within an age range of 20-29. Future iterations would target all ages and ethnicities with credit.

As previously mentioned, the main goal is to take a user’s student loans and payment plan to display simple aggregate data and charts that are easily digestible. After they see their current loan repayment outlook, they would have the ability create “alternative payment plans”. These allow a user to adjust lump sums and auto-payments without overwriting the “current plan”. The application would recalculate interest paid and repayment time on the alternative plan and overlay this data onto to the current plan to show interest and months saved on the new plan. The main benefit to users is receiving data that is personalized and specific to their loans. Rather than reading generic information on this topic, they have a chance to make a new plan that they can be proactive with and take immediate action with.

II. Minimum Viable Product

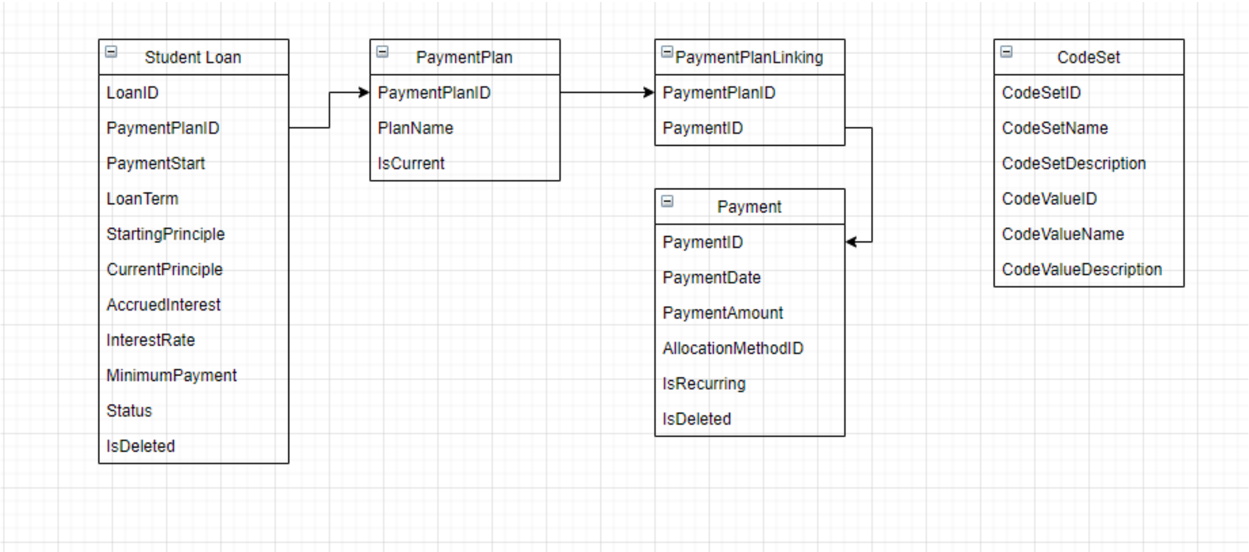
For the sake of personalization and mobility, users should be able to create an account and persist their data between devices. The basic application will utilize Basic Authentication, however, stretch goals would be implementing SSO through Google and Facebook accounts. Users should be able to add the principle, accrued interest, interest rate, status (deferred, forbearance, ect.), minimum required monthly payment, and the current monthly payment for each student loan they have. Statistics and charts should then clearly be presented to the user. Users should then be able to create “alternative plans”. Users will need to enter a new monthly payment for each loan and enter recurring lump sum payments or single payments with specified dates.

1. Data Presentation

   a. Graphs – ability to switch between a single line to represent all loans or a line for each individual loan.
      i. y-axis: Loan Balance after monthly payment, x-axis: month-year
      ii. y-axis: Total interest paid since beginning of loan, x-axis: month-year
      iii. y-axis: Interest being accrued daily, x-axis: month-year
   b. Aggregate Statistics
      i. Estimated Payoff date
      ii. Current principal amount
      iii. Current interest on top of principle
      iv. Expected total interested paid
      v. Average interest accrued daily
2. High-Level Architecture
   a. View Layer
      i. ReactJS
   b. Service Layer
      i. NodeJS
      ii. Rest based API
      iii. Express Framework
   c. Persistence Layer
      i. Microsoft SQL Server
      ii. Azure Cloud SQL Database

3. High-Level Data Overview

This application will be heavily data driven. User's individual loan information will need to be stored in the database and queries can be run against this to provide complex financial calculations. Calculations should not be handled by any other layer to ensure accuracy. Each loan can have multiple payment plans (current and alternative). Each payment plan can have a variety of monthly payments (ex. 2 monthly recurring payments and 1 lump sum every 6 months). Loan information should refence a payment plan as "IsCurrent" and users will be able to elect an alternative plan as their new current plan. Consideration will be put into creating a "Codeset" table for generating dropdowns and text to allow for different languages in future iterations. AllocationMethodID references the CodeSet table and refers to whether a payment goes to the loan with highest interest or with the highest balance first.


Student Loan
LoanID
PaymentPlanID
PaymentStart
LoanTerm
StartingPrinciple
CurrentPrinciple
AccruedInterest
InterestRate
MinimumPayment
Status
IsDeleted
PaymentPlan
PaymentPlanID
PlanName
IsCurrent
PaymentPlanLinking
PaymentPlanID
PaymentID
Payment
PaymentID
PaymentDate
PaymentAmount
AllocationMethodID
IsRecurring
IsDeleted
CodeSet
CodeSetID
CodeSetName
CodeSetDescription
CodeValueID
CodeValueName
CodeValueDescription