James Elam

SWDV 691-1

Database Re-Design: Loan Payoff Calculator

1. Preface

This documentation has been updated with changes detailed in the “Design Feedback Response” document. These changes include adding “DateCreated”, “DateUpdated”, and “DateDeleted” to all database models. The User’s table will now include “CreationMethod”, “FacebookID”, and “GoogleID” to allow for account creation through these social media OAuth methods. The Loan model will also no longer have a foreign key for “PaymentPlanID”. The Payments table will relate directly to a user rather than through their Loans. My justification for MSSQL has not changed since I feel that relationships between each table are very important and MSSQL does an excellent job for this. An image of the new Database design can be found at the end of this document.

1. Justification

The Loan Payoff Calculator application gives users an analytical view of how making lump sum payments or changing their auto-payment would affect their loan repayment. As a requirement, the user must be able to creation an account and sign-in. This allows for their loans and payments to be related to their user account and persist their data between application visits. The data points driving this application must be related to one another and back to the user. As a result, I will be using the relational database Microsoft SQL Server to store data.

1. Data Structure

All the data being presented to the User is personal and relevant only to them thus everything needs to link back to their UserID. Careful consideration was put towards not allowing any personally identifiable loan data or PII that could allow a loan to be linked back to a real person. User’s can enter any number of Student Loans that they have. Without entering any PaymentPlans, calculations can be made on the Loan Table alone since they are required to enter their minimum payment per month. Thus, we can display their current loan repayment trajectory.

Every Loan is linked to a Payment Plan which has many Payment associated with it. Payment Plans are meant to overwrite or compare with their Minimum payment plan. Each payment has an AllocationMethodID which is how that payment will be applied to the StudentLoan such as going towards the high interest loans or towards the highest balance loan first.

The CodeSet table is for any values that need to be expanded on and group them into categories. All of the possible AllocationMethodID’s would be in the codeset table. Each value would have the same CodeSetID (category), but have a different CodeValueID. This CodeSet table will also be used to generate dropdowns and certain texts on the front-end. This will allow most aspects of the site to be data-driven and for changes to be made via the database.

