

James Elam

SWDV 691-1

Database Design

I. Justification

The Student Loan 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 login for the application to persist and associate inputted information to the user later. Users should be able to enter their loan information and payment plans and return at any point. Much of the data being stored relates to one another. As a result, I will be using the relational database Microsoft SQL Server to store and relate data.

II. Data Structure

All of 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.

