NORTHWEST NAZARENE UNIVERSITY

Enterprise Resource Planning System Implementation


THESIS
Submitted to the Department of Mathematics and Computer Science
in Partial fulfillment of the requirements
for the degree of
BACHELOR OF ARTS

Cody Robbins
2023

THESIS
Submitted to the Department of Mathematics and Computer Science
in Partial fulfillment of the requirements
for the degree of
BACHELOR OF ARTS

Cody Robbins
2023

Enterprise Resource Planning System Implementation

Author: _____
Cody Robbins
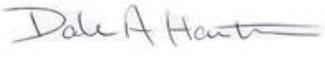
Approved: _____
Dale Hamilton, Ph.D., Department of Mathematics and Computer
Science, Faculty Advisor

Approved: _____
Kevin Mark, Project Management Office, ERP Consultant

Approved: _____
Barry Myers, Ph.D., Chair,
Department of Mathematics and Computer Science

Abstract

Enterprise Resource Planning System Implementation.

Robbins, Cody (Department of Mathematics and Computer Science),
HAMILTON, DR. DALE (Department of Mathematics and Computer Science)

The purpose of an Enterprise Resource Planning System (ERP) is to assist in managing software from different vendors to create a unified system to increase the effectiveness of a company. When attempting an ERP implementation, requirements gathering of current processes is essential because it provides a guide for what the company needs the new system to accomplish. How we achieved this on the Northwest Nazarene University ERP migration was first to meet with the department and get a general idea of some of their business processes. Once enough information was gathered, mapping the information started; the goal was to capture a high-level flow that allowed for the design of the future state.

Table of Contents

Table of Figures

Introduction

Background

The goal of the Denali project is to transition Enterprise Resource Management systems from Current System to Future System. There are some major differences in how these systems operate. The current System is an Informix-based data management system that uses a Unix-style command to manage and move data. One of the reasons for the switch was that the Informix database is a non-relational system, and any user of the CX software can change the undelaying database, and it is challenging to cover. The Future System will replace the Current System; the future system is a Microsoft-based system that runs in the cloud and uses native Microsoft features such as SQL Server, SharePoint, Teams Etc...

Involvement in the NNU ERP Migration was primary in current state analysis, SharePoint permissions, and database mapping. Time was split, and all work was done concurrently. The biggest challenge was database mapping because data access was needed to validate the correct IDs being mapped. It was not essential to the project because that was not the map's goal.

Body

SharePoint

JavaScript Object Notation is used to do advanced customization to share point sites. JSON is like XML for the structure of code. In Denial's SharePoint site, a custom display was needed to show the project stats and if the project was on time. There are great

templates to get a good idea of what you want and customize from there. Figure 1 shows

the code that was used for the custom timeline, and Tecchan1107 made the template.

```json
{
  "$schema": "https://developer.microsoft.com/json-schemas/sp/v2/board-formatting.schema.json",
  "hideSelection": true,
  "formatter": {
    "elmType": "div",
    "style": {
      "display": "flex",
      "justify-content": "center",
      "align-items": "flex-start",
      "width": "100%"
    },
    "children": [
      {
        "elmType": "div",
        "style": {
          "display": "flex",
          "flex-direction": "column",
          "border-left": "10px solid",
          "padding-left": "18px",
          "margin-top": "1px",
          "width": "85%",
          "position": "relative"
        },
        "attributes": {
          "class": "= if([$Title] != 'June 2022' , 'ms-bgColor-blue','')"
        },
        "children": [
          {
            "elmType": "div",
            "txtContent": "[$Title]",
            "inlineEditField": "[$Title]",
            "style": {
              "font-weight": "bold",
              "font-size": "20px"
            }
          },

          {
            "elmType": "div",
            "txtContent": "[$Description]",
            "inlineEditField": "[$Description]",
            "style": {
              "padding-top": "10px",
              "padding-bottom": "10px"
            }
          }
        ]
      },
      {
        "elmType": "div",
        "style": {
          "width": "30px",
          "height": "30px",
          "border-radius": "50%",
          "position": "absolute",
          "left": "-1%"
        },
        "attributes": {
          "class": "ms-bgColor-themePrimary"
        }
      }
    ]
  }
}
```

Figure 1 JSON code for Timeline

Site Timeline design

the approach that was taken for gathering the requirements for a project timeline was based on a meeting with the project management office, and we discussed what elements he wanted, such as color, placement, and the general structure of the project timeline. He had a template in mind that he liked but wanted changes. In Figure 2, this was the final version that was on the site for two months, tell it was replaced with just a time scale for project updates.
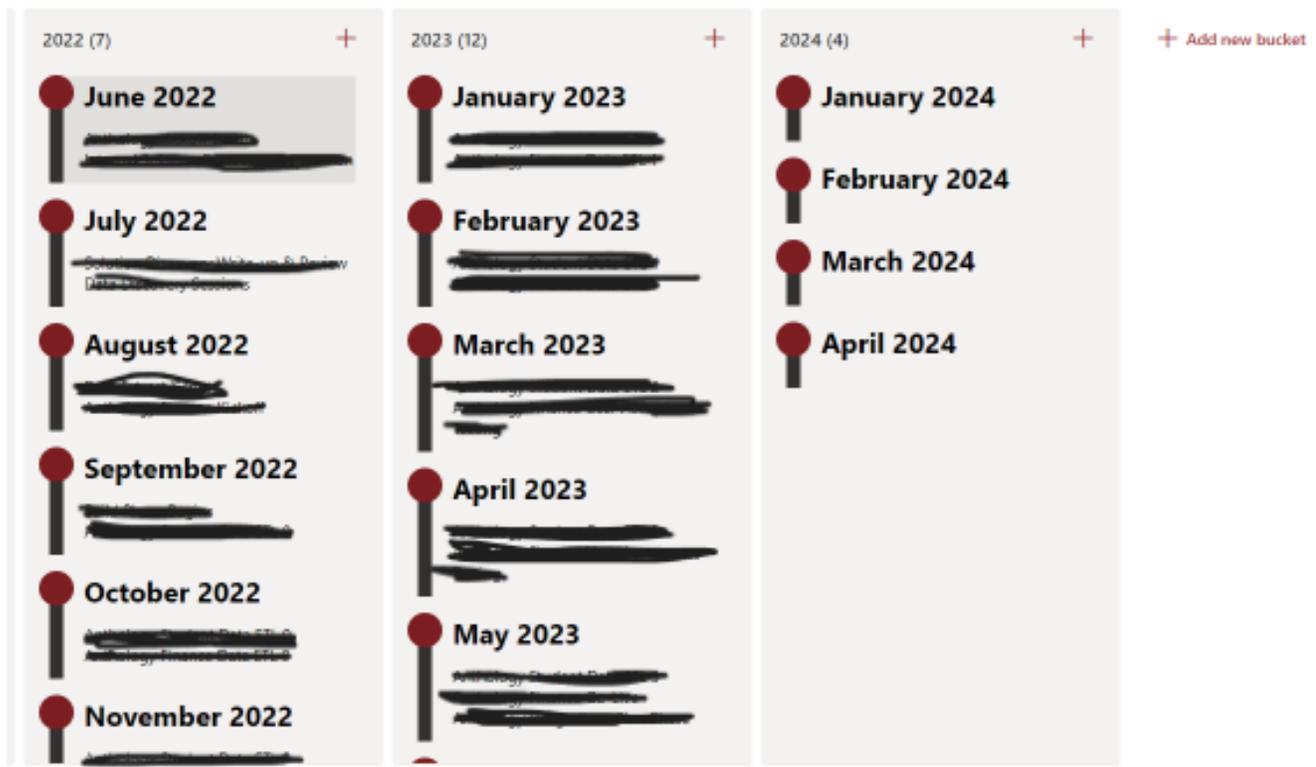
Figure 2 Project timeline site view

Access levels

When working with access levels to a SharePoint site, there are three significant options owner, member, and visitor. These are the main groups applied to users of the Denali site. There are three owners and many active Denali members, but they have more restricted access to the site and views, which can only look at documents and not download or change them. Figure 3 shows the 3 base classes that are used in SharePoint.

Manage who has access to this site.

Add members ∨

∨   Site owners - full control ⓘ

∨   Site members - limited control ⓘ

∨   Site visitors - no control ⓘ
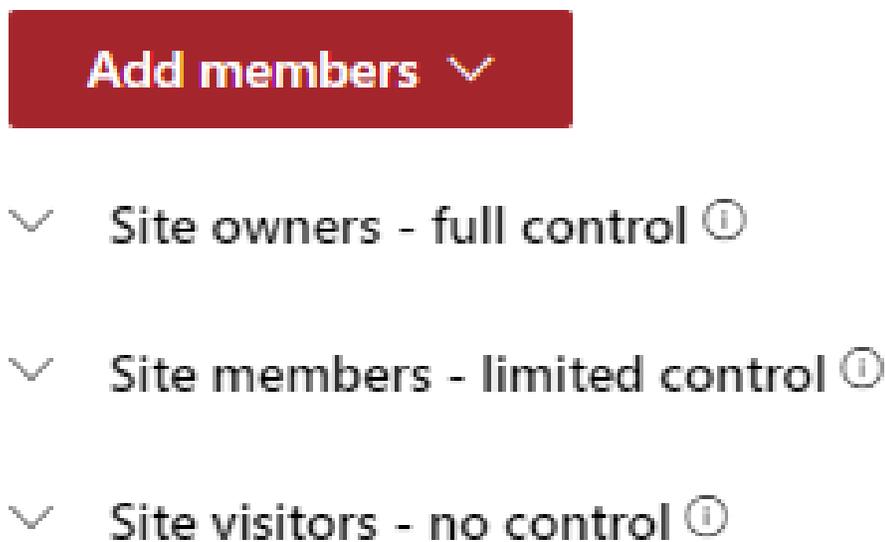
Figure 3 shows different types of users.

Stock Permissions

Permissions are an integral part of making a successful SharePoint site. When giving an individual edit access to a site, there are many things to consider because this lets a user

change the whole site, which is the default permission on a site. Figure 4 shows the

different types of people that can be given and what their effect is on a user.

Users:
  Denali Group

Permissions:
☐ Full Control - Has full control.
☐ Design - Can view, add, update, delete, approve, and customize.
☑ Edit - Can add, edit and delete lists; can view, add, update and delete list items and documents.
☐ Contribute - Can view, add, update, and delete list items and documents.
☐ Review - Can view and review list items and documents.
☐ Read - Can view pages and list items and download documents.
☐ Restricted View - Can view pages, list items, and documents. Documents can be viewed in the browser but not downloaded.
☑ Limited Access - Can view specific lists, document libraries, list items, folders, or documents when given permissions.
☑ Web-Only Limited Access - Can only view the web when given permissions.
☐ Approve - Can edit and approve pages, list items, and documents.
☐ Manage Hierarchy - Can create sites and edit pages, list items, and documents.
☐ Restricted Read - Can view pages and documents, but cannot view historical versions or user permissions.
☐ Restricted Interfaces for Translation - Can open lists and folders, and use remote interfaces.
☐ Modified edit - Can edit lists; can view, add, update and delete list items and documents.

Figure 4 Default user groups

Data Model

Understanding the Current System data schema took work because the original system

was implemented in 2002 and has had many modifications over the years to fit the needs

of the university in its daily tasks. Meeting with an NNU Development team member,

and they ran through the naming scheme and how they tracked different tables with sub-

names. The Current System has a set schema with their product, which includes set fields

and other attributes for use in their front-end software. A set of requirements was

established, including looking for a custom field or table. The purpose of finding the

custom tables and fields was to provide this information to the architects of the future

System team about the changes to the base table attributes because their custom script

that pulls for a stock Current System schema would fail if there were added fields and

tables. An example of a table is aa shown in Figure 7. For this table to be mapped

correctly, the base id table would need to be checked because there were many different

forms of an ID in the schema, and verification was required to find the correct attribute in

the aa table. Verification was needed to ensure there were no custom fields in the table,

which required checking the base schema table provided by Current System.
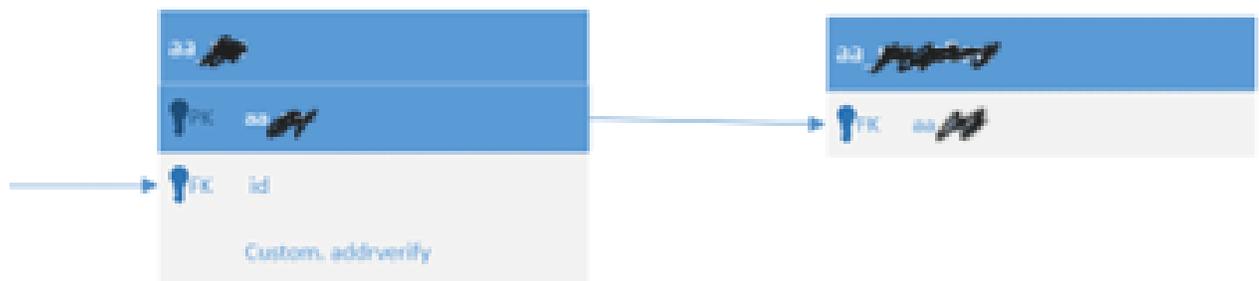
Figure 5 Relationship of id and aa

Data Mapping Design Process

When the project started, the map key was the first piece made because it is the guild for

the tables, and what is essential is that the key shown in Figure 6 is version 3. The first

version of the map was just all the information from the tables, but that was not necessary

after a few meetings with the future System and the NNU technical team. This was

narrowed down to tables not included in the stock schema of the current system. There

are only one-to-one relationships in the overview map shown in Figure 5 because of how

the Informix database was set up in 2002 by Current System.
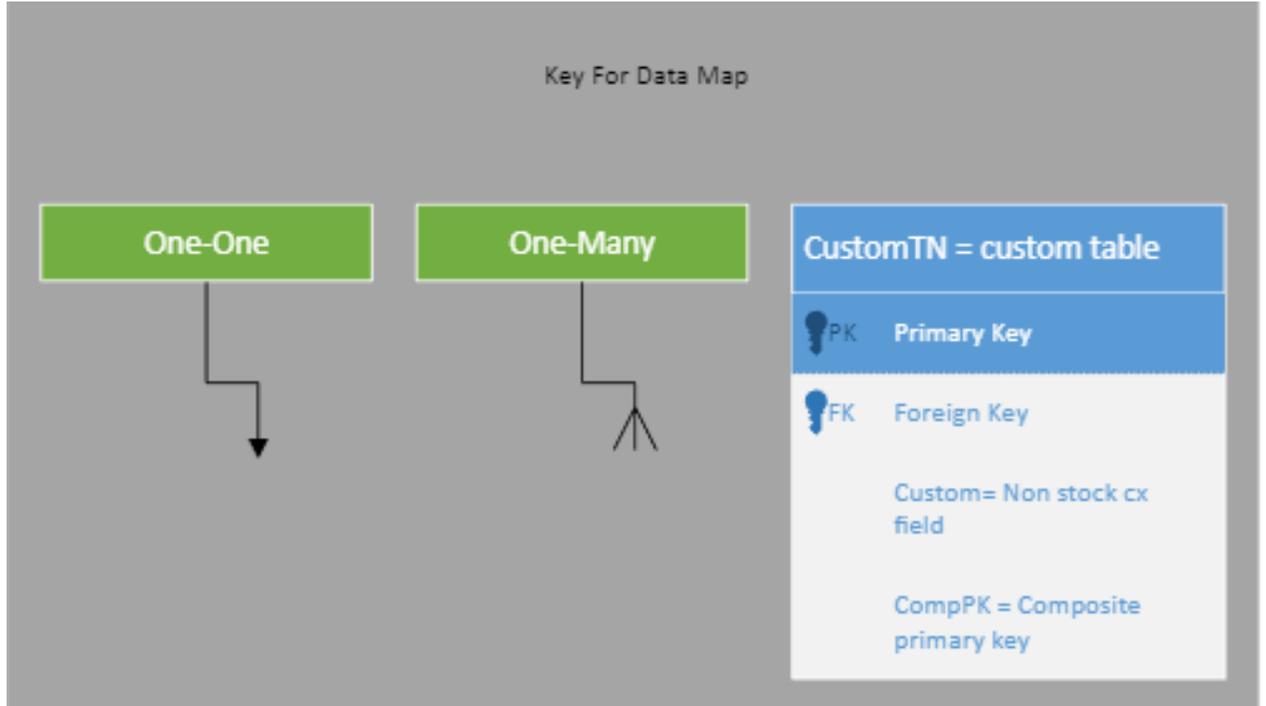
Figure 6 shows the chosen format.

Map Splitting

When designing a map containing 300, when complete, splitting it into sub-sections is essential because the overview map would be impossible to make readable. The decision to split was necessary, and the split was accomplished by taking tables with extensive branching paths. For example, in Figure 8, the ctc table has nine tables that branched off.
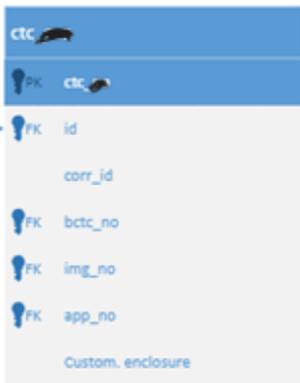
Figure 7 base ctc table from the high-level map.
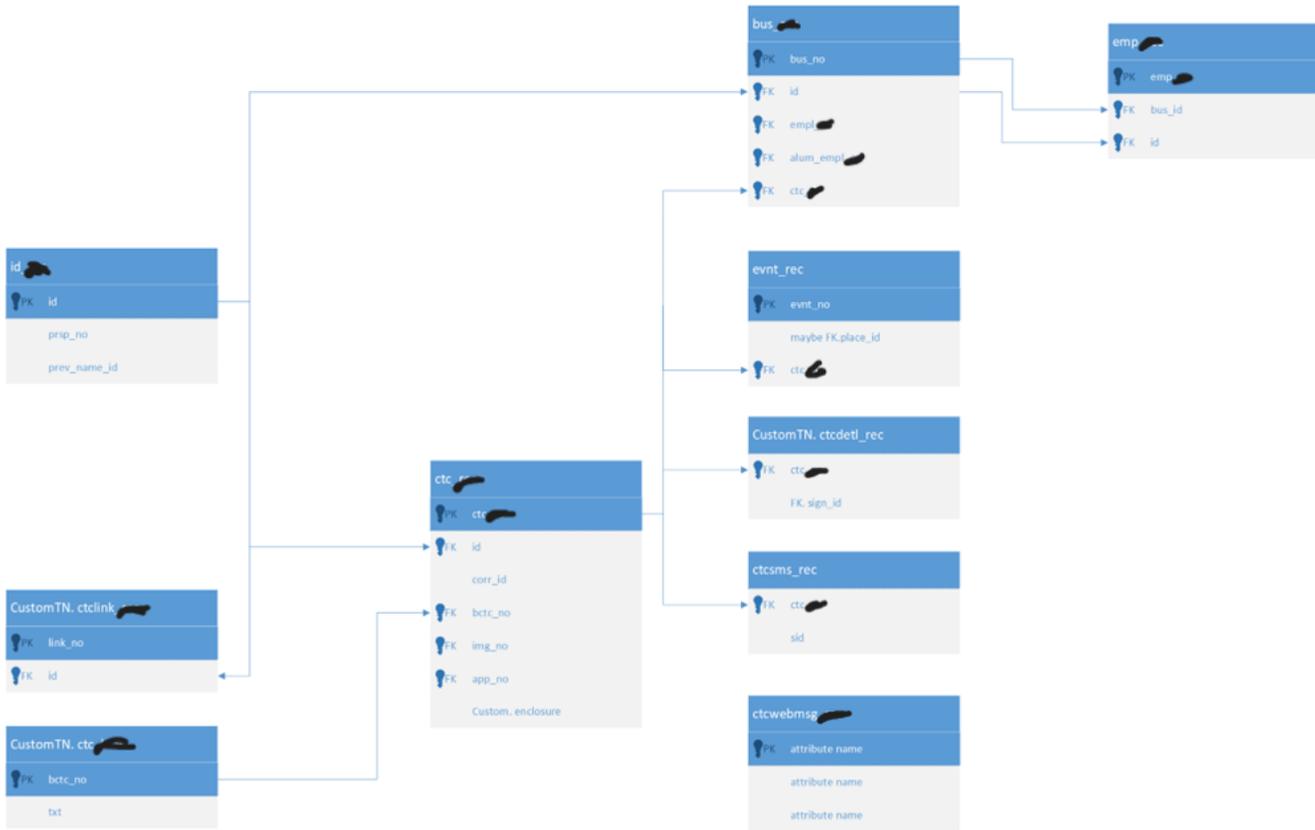
Figure 8 ctc expanded table map.

Final Solution

In the beginning, the idea was to create an application that would read and then distinguish the tables, but this was going to require a massive amount of time and would have been a one-off application that would never be used again, so it was decided to proceed manually, which took a long time. The competition rate of the tables was 60 out of 300. It was swift to copy the whole table to a Visio map, but this proved useless because some tables had 40 fields. The following approach was to use only the custom tables and custom fields in base tables and their ID, which reduced the map size and gave a much more helpful map. Figure 9 shows a portion of the overview map that was created based on stock and custom schema of the current system.
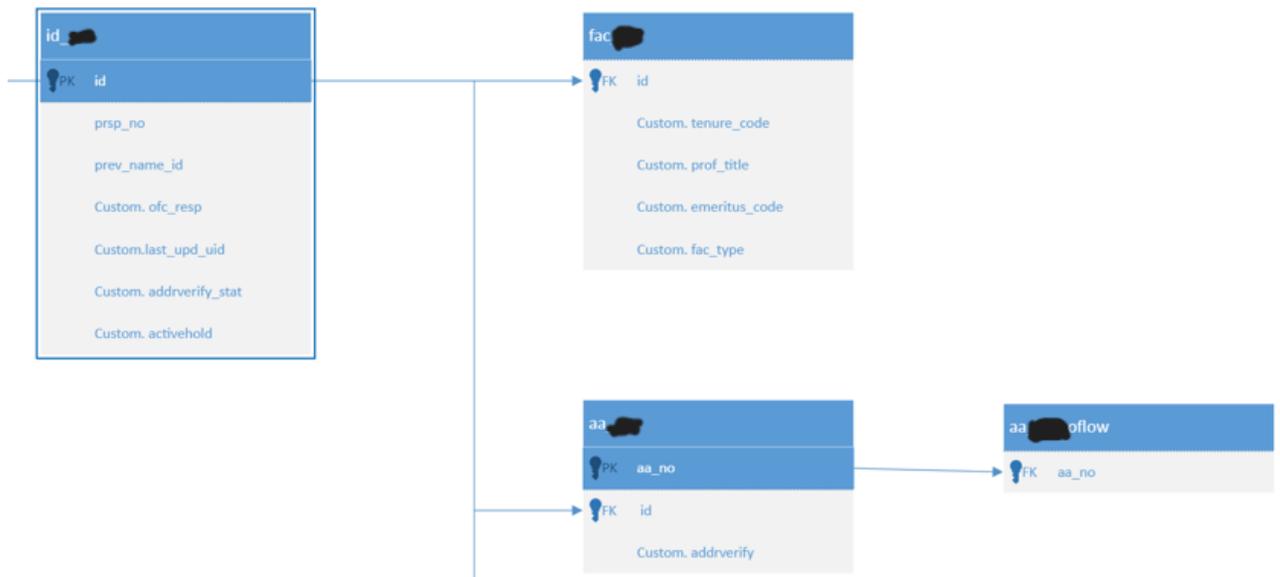
Figure 9 Part of the Database Overview map

Financial Aid's Current Process

It was crucial to understanding how the financial aid department performs its daily tasks in the Current System. The start was creating a spreadsheet with a few questions about their work. The document was distributed to everyone in the department about a week before the meeting to give time for accuracy. In Figure 10, this was the spreadsheet delivered to them.

	Is this procedure required to comply with governing bodies? If so, which one?	Which staff member performed the proccedure?	How often is the procedures performed?
Short descriptive phrase describing the process		Staff Member Name	Please enter D for daily, W for weekly, M for monthly, S for per Session , Y for yearly.
e.g. Awarding new student aid	e.g. Department of Ed	e.g. Tom	e.g. S

Figure 10 questions list

Current State Requirement Documentation

Before the meeting, the NNU technical team thoroughly reviewed the questions and created follow-up and new questions for the financial aid department. In the meeting, the team asked them to walk through how a new student is processed in the current system, and current students document the flow and pain points of the current process. Collecting the pain points is crucial because it narrows the focus and what is essential to the department for the future state. After the meeting, a high-level map shown in Figure 11 was created and sent to financial aid for approval. This process took about a week because many changes were made to the map's flow.
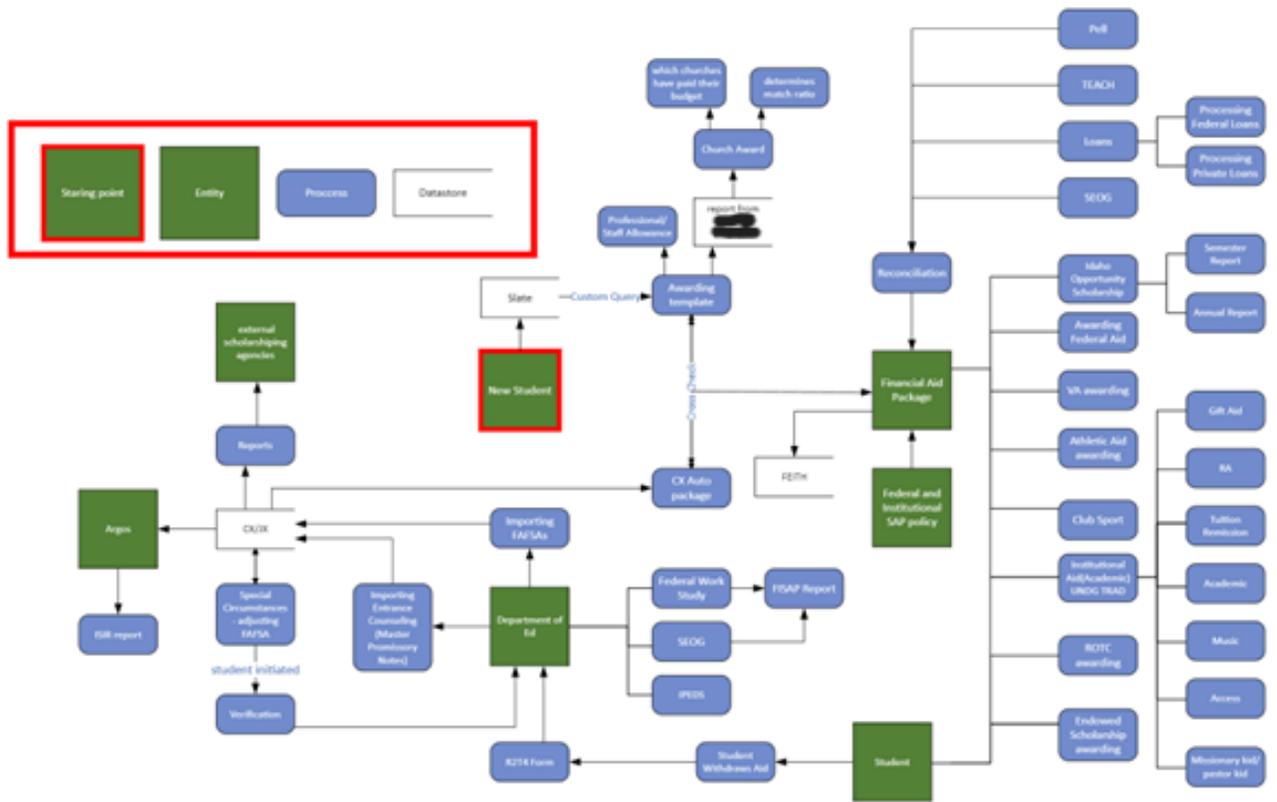
Figure 11 High-level Overview

Subsections to High-level map

After approving the high-level map, the sub-mapping process began for our original
spreadsheet. We had names that corresponded to specific functions in the current system.
The following steps were to have a person for the financial aid department walk us
through their daily tasks, document their process, and record if it was manual, automated,
or a combination. This was because if a process was automated, the NNU technical team
could try to find the process to understand better how to reimplement the function if the
new system does not have this functionality. Figure 12 is one of the sub-maps created
from the office based on what different employees told us about their job.

Legend

- **Staring point** (green box with red border)
- **Entity** (green box)
- **Proccess** (blue box)
- **Datastore** (white box)

Process Flow

Elmone.com → List of students → Financial Aid Office → Manually enter records → CX

CX → Export loan process. Creates file → FileZilla → LSC must be in file name to work → Elmone.com → Must add "AC" in front of file name. Manually upload → CX

CX → Makes loan package → 3 files made when "dispersed" → Open "DS2" file in word, save as "Rich Text Format" → FileZilla

Open "DS2" file in word, save as "Rich Text Format" → Chris

Figure 12 Private Loans Map

Conclusion

Overall, the Denali project has been challenging and a great introduction to how complex and difficult it can be to implement an ERP system. Going into this project, the experience from the work done with the General Education Council helping to improve their data collection process and create charts and other pieces. This gave me the knowledge needed to help with database mapping because of my experience working with and transforming GE data. The Denali project also introduced me to new tools such as SharePoint, teams, Jason programming, and Visio.

The project used many skills learned throughout my courses at NNU. This project used my knowledge of Database design, systems analysis, and machine learning for XML used in the final project to help me understand how the structure of JSON works.

References

MikePlumleyMSFT. (2023b, February 21). *Introduction to SharePoint and OneDrive -
SharePoint in Microsoft 365*. Microsoft Learn. https://learn.microsoft.com/en-
us/sharepoint/introduction

Tecchan1107. (2022.d.). *List-Formatting/view-samples/timeline-board at master ·
pnp/List-Formatting*. GitHub. https://github.com/pnp/List-
Formatting/tree/master/view-samples/timeline-board