Information Technology Services (ITS) Program Database

Lora Erland
Dakota State University

Follow this and additional works at: https://scholar.dsu.edu/theses
Part of the Databases and Information Systems Commons

Recommended Citation
https://scholar.dsu.edu/theses/322

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses & Doctoral Dissertations by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.
INFORMATION TECHNOLOGY SERVICES (ITS)  
PROGRAM DATABASE

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

November 15, 2015

By
Lora Ersland

Project Committee:

Stephen Krebsbach
Chris Olson
Marilyn Halgerson
We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Lora Ersland

Master’s Project Title: Information Technology Services (ITS) Program Database

Faculty supervisor: Stephen Krebsbach Ph.D. Date: 12/11/2025

Committee member: Christopher Olson Date: 12/11/2015

Committee member: Date: 12/11/2015
ACKNOWLEDGMENT

I would like to express my sincere appreciation to everyone who has contributed to the completion of my master’s degree. Without the support, the completion of this degree would not have been possible.

Marilyn Halgerson, Amy Olson, and John McNary for their ideas, support and guidance through this project.

K-12 Data Center staff members James Booze and Allen Thiele for their guidance through the project and technical expertise.

Dr. Stephen Krebsbach, Dr. Christopher Olson, and Marilyn Halgerson for taking the time to serve on my project committee. And to Dr. Krebsbach for serving as my advisor throughout my degree completion.

My mother, Lyla, mother-in-law, Carol, and father-in-law, David, sisters, brother, and their families for their support, encouragement, and help with the kids when I needed to get homework and projects done.

And last, but certainly not least, to my husband, Paul, and our boys, Andrew, Alex, Aaron, and Ashton for their constant love, support, and encouragement. When I went back to school, I added an additional craziness to our already crazy schedules. Their willingness to work around my classes and projects made this accomplishment possible.
ABSTRACT

The South Dakota Board of Regents (SDBOR) system, comprised of the six public universities, has undertaken a project to migrate the Colleague Student Information System from a proprietary Unidata database to an Oracle database. The conversion to the Oracle database will allow the ITS Administrative Computing staff more options to create programs that access the Colleague Student Information System.

During the conversion of the existing Unidata programs, it was discovered that the migration was causing us to lose our ability to keep track of programs that were developed to access the Colleague system. In addition, while we are gaining the option of using different tools to develop programs, we are also losing our ability to easily search through developed programs to find a specific program or search for a program that might contain a certain aspect that is desired. Users across campus have also asked multiple times if a certain program might exist to extract information they are looking for. We determined that there was a need for a system that would allow program information to be stored and accessed by ITS Administrative Computing staff and staff members across campus who have access to run programs against the Colleague Student Information System.

An ITS Program Database was incorporated into the existing Project Request System to allow access to the database in a location that is already accessed by staff. The ITS Program Database contains SQL tables to store program information and uses ASP.NET and C# WebForms to access the program information from the SQL tables. Table structure, WebForm designs, programming languages, and programming structure was determined from the existing Project Request System.
The outcome of the ITS Program Database project was a system that is user-friendly and easily maintainable. The ITS Program Database is accessible through the DSU Portal and linked to the existing Project Request System that is already used by staff across campus. Campus staff members are able to access a list of programs they have access to run. They also have the ability to search through their list of programs to find a program based on search terms. This project has given the ITS Administrative Computing staff a tool that allows the tracking of completed projects to avoid duplicate creation and the ability to search for programs created with a variety of tools.
DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

[Signature]

<Student name>
# TABLE OF CONTENTS

**PROJECT APPROVAL FORM** ................................................................. II

**ACKNOWLEDGMENT** ........................................................................ II

**ABSTRACT** ................................................................................... IV

**DECLARATION** ........................................................................... VII

**TABLE OF CONTENTS** ................................................................... VIII

**LIST OF TABLES** .......................................................................... IX

**LIST OF FIGURES** ......................................................................... IX

**INTRODUCTION** ........................................................................ 1
  - **BACKGROUND OF THE PROBLEM** ....................................... 1
  - **STATEMENT OF THE PROBLEM** ........................................... 2
  - **OBJECTIVES OF THE PROJECT** ............................................ 4

**SYSTEM DESIGN** ....................................................................... 9
  - **PRELIMINARY INFORMATION** ............................................... 9
  - **PLANNING PHASE** ............................................................... 10
  - **DESIGN PHASE** ................................................................. 13
  - **DEVELOPMENT PHASE** ..................................................... 18
  - **IMPLEMENTATION PHASE** .................................................. 21

**CASE STUDY** ............................................................................. 233

**CONCLUSIONS** .......................................................................... 27

**REFERENCES** ............................................................................. 30

**APPENDICES** ............................................................................ 31

**APPENDIX A: SQL CODE** .......................................................... 31

**APPENDIX B: ER DIAGRAM AND DATA DICTIONARY** ............... 33

**APPENDIX C: CODE FOR WEBFORMS** ...................................... 37

**APPENDIX D: SCREEN SHOTS OF WEBFORMS** ......................... 38

**APPENDIX E: WORK BREAKDOWN STRUCTURE (WBS)** ............. 44
LIST OF TABLES

Table 1. ITS Program Database Objectives ............................................................... 8
Table 2. ITS Program Database Programs Table Variables ..............................12
LIST OF FIGURES

Figure 1. Rapid Application Development Methodology ........................................... 10

Figure 2. ITS Program Database Primary Tables ....................................................... 14

Figure 3. ITS Program Database Support Tables ...................................................... 16

Figure 4. Project Database ER Diagram ................................................................. 19

Figure 5. WebForm Flow ....................................................................................... 21
INTRODUCTION

Background of the Problem

The Colleague Student Information System, housed and maintained by the South Dakota Board of Regents through the Regents Information System (RIS), is the student information system that is used by the six state institutions in South Dakota, Black Hills State University (BHSU), Dakota State University (DSU), Northern State University (NSU), South Dakota School of Mines & Technology (SDSMT), South Dakota State University (SDSU), and the University of South Dakota (USD). The one instance of the Colleague Student Information System is used to maintain applicant, student, faculty, course, section, and student finance information for all six universities. The Administrative Computing staff in the Information Technology Services (ITS) are responsible for administering the Colleague Student Information System for DSU and the development of various programs run against the system.

The Colleague Student Information System used by the South Dakota universities is currently based on a Unidata database, which is a proprietary database that Colleague was originally developed on. The development options for programs on the Unidata database is limited to a few options; Uniquery paragraphs, Unidata subroutines, WebWizard subroutines, and Data Ready reports. Although four options were available for programming on the Unidata database, Uniquery paragraphs were used the tool for about 90% of all programs that were developed.

Over the past few years, an Oracle Migration project has been underway to migrate the Colleague Student Information System from a Unidata database to an Oracle SQL database platform. The migration will allow programs to be produced against the Oracle database
using SQL tools. The options that will be available to us in the Oracle environment include PL/SQL, Cognos, IBM’s business intelligence (BI) and performance management software suite, Colleague Studio, and SLCR, a process available in Colleague to produce a list of IDs to be used in other processes within Colleague. Each of these development options will be used for a different purpose with no one tool being a primary tool. The migration from the Unidata database to the Oracle database is expected to take place over spring break in March 2016.

**Statement of the problem**

As part of the migration to the new database platform, a conversion of every program developed accessing the Unidata database of the Colleague Student Information System is required. This conversion is occurring at each of the public university campuses and at RIS. The conversion consists of analyzing all programs that were developed in the Unidata environment, analyzing what the program does, determining what tool will be used for the conversion, converting the program, and testing the new program in the Oracle environment.

While the Colleague Student Information System was based on a Unidata database and there was only one main tool used for development of programs, it was fairly easy to keep track of where the programs had been written. In the event that it was unknown where the program existed, a search option was available in Uniquery allowing keyword searches to be performed on the programs in the library. Users across campus have access to run a limited number of programs that have been developed by ITS Administrative Computing staff. These programs were written as Uniquery paragraphs as either menu mnemonics or Data Ready reports. The menu mnemonics are accessible by a menu mnemonic in Colleague UI. The Data Ready reports are accessible through a secured hyperlink accessing information from the
Colleague Student Information System. Users are able to see the menu mnemonics on a menu listing in Colleague UI. The Data Ready report hyperlinks accessible to a user are stored in an email or a document, depending on how the user chooses to keep track of the options they have access to run.

A benefit of migrating to the Oracle database is that there are more useful tools that can be used to develop programs. However, more development options means that the programs will be developed in different places, which will make accessing one location to find programs no longer feasible. ITS Administrative Computing staff no longer have the ability to search through one library to find the programs that have been developed, determine if a program has already been created, or to find a program that is known to have been developed. Users on campus will still have the menu mnemonics available in Colleague UI to run programs that have been created with Colleague Studio. Some campus users will also have the ability to run SLCR to create lists of IDs to be used in processes being run in Colleague. Another new option to campus users will be the availability of programs to be run using the new Cognos reporting tool.

The ITS Administrative Computing staff and DSU campus users would benefit from having a central ITS Program Database that would be used to store information about all custom program development on the Colleague Student Information System. The information stored in the ITS Program Database would include information such as program name, program description, process name, program system, access location, data type, and who created the program.

One benefit of the ITS Program Database is that it would be helpful for ITS Administrative Computing staff to determine if a program already exists. While working on
the Oracle migration, it was determined that many programs accessing the Colleague Student Information System from the Unidata environment were created with similar specifications. The ITS Program Database would be useful to search for programs that might already exist using keyword searches.

Another benefit is that the ITS Administrative Computing staff and university staff would have the ability to search through all programs that have been created using the various tools available for the Oracle environment from one location. A program created using any of the new tools, PL/SQL, Cognos, Colleague Studio, or SLCR, would be entered into the Program Database along with pertinent information about each of the programs. Information that would be entered would indicate which tool was used to create the program and how to access that program to either run the program or make updates.

A third benefit is that users across campus would have the ability to have a list of programs that they have access to run available to them in one location. There have been many questions from users asking if a certain program exists and, if it does, how to access the information. Users would also be able to look for program information in one location. This will eliminate the need for all users to keep their own list of programs they have access to.

**Objectives of the project**

The Program Database will have two different types of users. Each type of user will use the Program Database in a different capacity. Even though there are different uses for the Program Database, the objectives of the project will be the same.

The first type of users are the ITS Program Database administrators, who would be the ITS Administrative Computing staff. The staff members from this area will use the ITS Program Database as a means to store information about all programs they have developed
which access information from the Colleague Student Information System. The ITS Program Database will provide the ITS Administrative Computing staff a way to keep track of all programs that have been completed, with the expectation that we will eliminate the duplicate development of like programs.

The second group of users that will use the ITS Program Database are the users on the DSU campus. Any user that has access to run a program accessing data from the Colleague Student Information System will have access to view report listings in the ITS Program Database. The users will utilize the ITS Program Database to view a list of programs they have access to run. They will also be able to use the search option to search for a program from their access list.

The primary objectives for the ITS Program Database project are as follows:

1. The ITS Program Database will be structured in a straight-forward, user friendly interface.
2. The entry of program related information will be uncomplicated for ITS Administrative Computing staff.
3. The ITS Program Database will be accessible both on and off campus. Accessing the database must be compatible with policy that is established which indicates how systems are accessed on campus. Users also must not feel that they need to go out of their way to use the ITS Program Database.
4. The ITS Program Database will contain all pertinent program information as identified by ITS Administrative Computing staff. All program information will be available to the ITS Administrative Computing staff. Program information
identified as being accessible to users on the DSU campus will be made available to those users.

5. The ITS Program Database will allow for easy searching of programs and related program information.

ITS Administrative Computing staff released a Project Request System to campus users. The Project Request System is used to submit requests to the ITS Administrative Computing staff. The Project Request System was developed by staff in the K-12 Data Center. The K-12 Data Center is an outreach department located at DSU. While evaluating the accessibility of the ITS Program Database, it was determined that the best method to access the database is through the existing Project Request System. The Project Request System is accessible from the DSU Portal, which is compatible with DSU’s policy indicating how staff are able to access systems on campus. Campus users are set up with access to the Project Request System and have been accustomed to accessing the site on a regular basis. Security to the Project Request System is controlled using DSU’s active directory for user identification.

The ITS Program Database will contain a series of SQL tables that will be used to store the program information. To access the information stored in the tables, WebForms will be developed. WebForms will be available to ITS Administrative Computing staff and other DSU users based on security assigned in the ITS Program Database that will allow them to view information controlled by the security. The design and configurations of the WebForms will follow the design and standards established for the Project Request System.

The SQL tables and WebForm files will be stored on the same hardware as the Project Request System that is maintained by the staff in the K-12 Data Center. The SQL tables for
the ITS Program Database will be inserted into the existing Project Request System SQL database. The WebForms will be created in the same folder structure as the Project Request System.

Visual Studio will be used to create the SQL tables and develop the WebForms that will be used to access the program information in the tables, which is the same software used to create the Project Request System. The WebForms will be developed with C# and ASP.NET. The WebForms which will be available to access the ITS Program Database data include:

- Program display and search page for campus users. This page will be the initial page which all campus users will access to display all programs the user has access to.

- Program detail page for campus users. To view more detailed information about that program, a hyperlink for the program will be included on the program display and search page. The hyperlink will allow the users to access a program detail page that will display the additional information.

- Program display page for ITS Administrative Computing staff. This page will be developed to display all programs entered into the ITS Program Database. The ITS Administrative Computing staff who have access to this section will have access to view program information for all programs.

- Program detail page for ITS Administrative Computing staff. To view the more detailed information of each of the programs, this page will be accessed from a hyperlink on the program admin program display page. The programs admin program detail page will also allow staff to edit information about the program.
- Add Program page for ITS Administrative Computing staff. Additional programs and program information may be added to the ITS Program Database from the add programs page. After adding the programs, the programs admin program details page will be displayed. The keywords and departments for the program will then be added from this page.

- Table maintenance pages for ITS Administrative Computing staff. Support tables included in the SQL database will be maintainable from a web form. These tables will contain values that will be selected for each of the programs.

Table 1. ITS Program Database Objectives

<table>
<thead>
<tr>
<th>Objective</th>
<th>Deliverable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interface will be straight-forward and user friendly for all users. WebForms clearly designed with users in mind.</td>
</tr>
<tr>
<td>2</td>
<td>Uncomplicated entry of programs and related information. Minimal WebForms will be created for program entry.</td>
</tr>
<tr>
<td>3</td>
<td>Accessible Embedded in the Project Request System accessible through the DSU Portal.</td>
</tr>
<tr>
<td>4</td>
<td>Contain pertinent program information The tables in the database will be designed to include information deemed important by the ITS Administrative Computing staff.</td>
</tr>
<tr>
<td>5</td>
<td>Allow searching The ITS Program Database will be designed to allow campus users and ITS Administrative Computing staff the option for multiple search criteria.</td>
</tr>
</tbody>
</table>
SYSTEM DESIGN

Preliminary Information

The first vision of the ITS Program Database planned an application as a standalone database with web pages that would be used to access the information in the database. The web pages were going to be accessible through the DSU Portal. The program information would have been stored in tables in a SQL database. The web pages were to be developed using PHP.

After further discussions with ITS Administrative Computing staff, it was determined that this database should be implemented as an option in the existing Project Request System. The Project Request System was developed by staff in the K-12 Data Center located on the DSU campus. The Project Request System is housed on a system maintained by the K-12 Data Center. The ITS Program Database will follow the same format and design scheme that has been established in the Project Request System. The coding of the ITS Program Database will be in C# and ASP.NET accessing an SQL database, as does the existing Project Request System.

The ITS Program Database is being created for use by the ITS Administrative Computing staff and the Colleague users on the DSU campus. Information about each of the programs must be stored and easily accessible by ITS staff and campus users. Updating the ITS Program Database will be completed using manual entries into the database by the ITS Administrative Programming staff upon completion of the programs.

The development method being utilized for the ITS Program Database project is the rapid application development method. Rapid Application Development (RAD) allows for
quicker development of an application. The result of a RAD application is an application that it is better and more maintainable because of updates to the specifications taking place during the development. Using the RAD development approach, specifications were developed first. The following stages of the processes included development, testing, and refining in reiterative stages to allow for changes to be made to the application (“Rapid Application Development”, n.d.).

![Rapid Application Development Methodology](image)

Figure 1. Rapid Application Development Methodology

**Planning Phase**

The development of the ITS Program Database is for use by ITS Administrative Computing staff and DSU campus users who have access to run programs in Colleague. The inability of easy tracking of programs that have been developed using the different options led to the determination of a need for a central application that will allow for program information...
to be stored. The application must be usable by both ITS Administrative Computing staff and DSU campus users.

The ITS Program Database will be developed as part of the Project Request System. The Project Request System was developed by staff in the K-12 Data Center and is located on hardware maintained by them. The ITS Program Database will follow the Project Request System in location and functionality. Incorporating the project into an existing application meant that the programming language used and the design of the WebForms would match the existing Project Request System.

To accomplish the objective that the ITS Program Database be straight-forward and user friendly for all users, the design of the WebForms which are used to access the program information will be user friendly. The number of WebForms will be limited to eliminate the need to click through to multiple WebForms to retrieve the necessary information. The functionality on the page will also be straightforward with users not needing to look around to figure out what they need to do.

Program information will be entered and maintained by ITS Administrative Computing staff through minimal WebForms. A WebForm will be developed to enter the program information. After entering the program information, the program details form will be displayed. On the program details form, the keywords for the program and department access will be assigned.

The ITS Program Database will be easily accessible through the existing Project Request System. Users across campus already have access to the Project Request System and have been using the system to submit project requests to ITS Administrative Computing. Campus users will be able to search through the list of programs they have access to,
determining if they have access to run a program, prior to submitting a request for a report to be run. The Project Request System is available on the DSU Portal, which is compatible with DSU’s policy on how systems are accessed on campus.

The ITS Program Database will contain pertinent information for each of the programs that are developed to access data from the Colleague Student Information System. ITS Administrative Computing staff will determine the type of information that is included for the programs in the ITS Program Database. Staff in ITS Administrative Computing will have access to view, add, and maintain all program information. Users on the DSU campus will have access to certain aspects of information for each of the programs that will allow them to determine what programs are available and how to run the program. The table below lists and describes information that will be recorded for all programs:

Table 2. ITS Program Database Programs Table Variables

<table>
<thead>
<tr>
<th>Item to Record</th>
<th>Description</th>
<th>Access to Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Title</td>
<td>The short title of the program</td>
<td>Campus Users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program Description</td>
<td>A description of the program. This may include the purpose the program was</td>
<td>Campus Users</td>
</tr>
<tr>
<td></td>
<td>developed, key selection criteria, or details about data that is retrieved.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program System</td>
<td>The system where the program is run from. At this time, only the Colleague</td>
<td>Campus Users</td>
</tr>
<tr>
<td></td>
<td>Student Information System is included. Other systems, such as the Banner</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td></td>
<td>Finance System, may be included at a later date.</td>
<td></td>
</tr>
<tr>
<td>Program Data Type</td>
<td>The type of data that is used in the program. At this time, data types that</td>
<td>Campus Users</td>
</tr>
<tr>
<td></td>
<td>are available include Live, Historical, and ODS.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program Process Name</td>
<td>The process name that is used to run the program. This could be a mnemonic</td>
<td>Campus Users</td>
</tr>
<tr>
<td></td>
<td>or a program name.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program Creates Saved List</td>
<td>An indicator that the program creates a saved list to be used in the Colleague Student Information System.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Where Accessed</td>
<td>An indication of where the program was created. At this time, the values will include PL/SQL, Menu, Cognos, and SLCR.</td>
<td>Campus Users ITS Administrative Computing</td>
</tr>
<tr>
<td>Initial Programmer</td>
<td>This will be the initial programmer who created the program.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program Update Notes</td>
<td>Information about any updates made to a program.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program Creation Notes</td>
<td>Information about the creation of a program.</td>
<td>ITS Administrative Computing</td>
</tr>
<tr>
<td>Program Creation Date</td>
<td>Date the program was originally created.</td>
<td>ITS Administrative Computing</td>
</tr>
</tbody>
</table>

The ITS Program Database will have search features that will allow for searching of the program title, program description, program creation notes, program update notes, and keywords identified for the program. The search option will allow users to easily determine if a certain type of information is included in a particular program. The search option will also allow users to quickly find a program created for a particular purpose.

**Design Phase**

The design phase of the project was used to analyze the existing Project Request System and determine what would be needed to develop the ITS Program Database within the Project Request System. The existing table and folder structure would be the template that would be used for the Program Database. Existing tables and information set up in the Project Request System would be used when the same information would be needed.

The existing SQL tables were analyzed to determine the type of information that was used for the Project Request System. After analyzing the existing SQL tables in the Project Request System database, a number of primary tables were identified as needed for the ITS
Program Database. To store relevant program information, tables such as Programs, ProgramKeywords, and ProgramAssignments would be created. The Programs table will contain detailed information about each of the programs as identified in the Planning Phase and listed on pages 12-13. The ProgramKeywords table will contain each of the keywords assigned to a program. The ProgramAssignments table will contain the values of all departments who have access to view the information of a particular program.

Figure 2. ITS Program Database Primary Tables
Additional support tables will be created to store values that will be used to populate the Programs, ProgramKeywords and ProgramAssignments tables. The support tables identified include Programmers, ProgramPossibleKeywords, ProgramSystem, and ProgramWhereAccessed. All four tables will be maintained through WebForms developed in C# and ASP.NET which will allow ITS Administrative Computing staff, who have access to add program information in the ITS Program Database, to maintain the support tables. The Programmers table will contain the names of all programmers who can create programs that will be added to the ITS Program Database. The format of the Programmers table will be first_name last_name, with a space between the first and last names. The ProgramPossibleKeywords table will contain all keywords that can be assigned to programs and used to search for the programs. The keywords included in this table will be single word keywords. Examples of keywords included in this table are: admit, enroll, status, cumulative, term, and GPA. The ProgramSystem table will contain information about any system that the programs listed in the ITS Program Database can access. The initial value in this table will be Colleague, but we are allowing this option in the event that we would like to expand this database to include programs written against other systems, such as the Banner Finance System. The ProgramWhereAccessed table will store where the program should be run from. For example, an entry of “Menu” will be accessed from Colleague UI whereas an entry of “Cognos” will be accessed from the Cognos reporting utility.
The relationship between each of the main program tables and the support tables can be viewed in Appendix B on page 33.

The Project Request System has security already established to allow for user access, department admin access, and university admin access. The user access for campus users is determined by the department the user is employed in. The ITS Program Database security will model the existing security for the Project Request System. The access for each of the
programs will be assigned to a department. Not all users in a department listed in the Project Request System will have access to run reports against the Colleague Student Information System. A flag that will indicate if a user has access to run programs will be assigned on the department screen where users are assigned access to the Project Request System. A combination of the department being added to a program and the flag indicating if a user has access to run programs will determine if users will have access to view a particular program’s information. The entries for the department assignment to a program will be stored in the ProgramAssignments table.

Using an existing application meant using the already established programming languages, coding structures, and web site design. The same programming languages for the WebForms, C# and ASP.NET, will be used to create the WebForms for the ITS Program Database. Access and creation for the WebForms and SQL tables will be through Visual Studio. To confirm the move to using the same programming structure of ASP.NET as the Project Request System, it is beneficial to know the advantages of ASP.NET. ASP.NET has advantages such as source code and HTML that are together on the same pages, which makes it easier to write and maintain pages. ASP.NET also provides simplicity by making it easier to perform common tasks with less code. The code written in ASP.NET is executed on the server, before it hits the browser. This provides for more power in development and flexibility in developing the web pages (Kozyk, “What is ASP.NET? – Top 12 Advantages of ASP.NET”).

A tab structure is used in the Project Request System to separate the different functions of the system. The same tab structure is being included for the ITS Programs Database. A Programs tab will be accessible to users who have access to view programs
listed in the ITS Program Database. A Programs Admin tab will be accessible for ITS Administrative Computing staff who have access to add and update program information. The ability to view these tabs is dependent on security built into the ITS Program Database and Project Request System.

**Development Phase**

The development phase of the ITS Program Database project started with creating a local copy of the existing Project Request System. The copy of the system was used to develop the tables and WebForms needed for the ITS Program Database. After the development was completed, the changes were uploaded to the existing Program Request System with the help of K-12 Data Center staff.

Because the determination was previously made to use the Project Request System as a basis of the ITS Program Database project, the existing code, code structure, and design was used for the ITS Program Database. Analyzing the code structure and WebForm flow took some time to become familiar.

Table updates were initially made to the database to store program information and to allow the program information to be available when developing the WebForms. The SQL statements used to create the tables can be found in Appendix A on page 31. The relationship of each of the tables for the ITS Program Database is indicated by Figure 4 on the following page.
The WebForms were developed in two stages. The first stage was the display of program information for campus users. The WebForms needed for the campus users included a programs page and a program details page. The programs page lists all programs the user has access to see. A hyperlink is included on the programs page that then displayed the program details page. The program details pages includes additional program information for the user.

The second stage of WebForms created were the administrative WebForms that will be used by ITS Administrative Computing staff. The initial page used by the administrative staff is a program page similar to the programs page used by campus users, except security is not assigned to limit the programs that are displayed. All programs entered into the Programs table are displayed on the initial pages for ITS Administrative Computing staff. A hyperlink on the title will direct the user to the program admin details page. The program admin details page displays all information stored for the program. An option to edit all program
information is available on the program admin details page. The assignment of keywords and departments also takes place on this page.

The initial programs page for ITS Administrative Computing staff also contains options to maintain the support SQL tables. Clicking on one of those links will direct the user to a new page that will allow the user to update the SQL tables. An option to add an additional program to the programs database is also included on the initial programs page for ITS Administrative Computing staff. A WebForm will display which will allow the user to enter all necessary information for that program. After all initial program information is added and the add program button is clicked on, the program admin details page will display. The user is then able to add keywords and departments for the program or update information for the program. The keywords and departments for a program are added separately from the initial program information because of the how the table structure was defined in the SQL database.

Figure 5, on the following page, displays the page flow of the WebForms.
During the development of the tables and WebForms, routine meetings were held with ITS Administrative Programming to review the functionality. Changes were identified during the meetings. These changes were then made and implemented followed by more reviews until all specifications were completed.

**Implementation Phase**

The implementation phase is occurring in two phases. During the first phase, the Project Request System was updated to include the ITS Program Database. K-12 Data Center staff performed the actual update to the Project Request System because of permissions that are set up in the system. The Project Request System was updated with the SQL tables and WebForms that were developed for the ITS Program Database in the local copy of the Project
Request System. After the updates were made to include the ITS Program Database, the ITS Administrative Computing staff were given access to the site. The ITS Administrative Computing staff entered the values into the support tables. Programs that were already developed for the Oracle migration project were also entered into the ITS Program Database.

The second phase of the implementation took place during the 2015 Fall term. After the initial group of programs that have been converted for Oracle were entered into the ITS Program Database, access to the site was given to DSU campus users. Users will have access to all of the programs as we add them to the system. As programs continue to be developed, the program information will be added to the ITS Program Database.
CASE STUDY

The development of the ITS Program Database was incorporated into the existing Project Request System that is already utilized across campus. The Project Request System was developed using C# and ASP.NET. The same development languages and WebForms design used in the Project Request System will be used in the ITS Program Database.

The development of the ITS Program Database was a challenge because of limited previous knowledge of C# and ASP.NET. The initial steps of the ITS Program Database development included determining what code on the WebForm ASPX and CS pages were needed for the development of the ITS Program Database WebForms. The way that the pages were built and the way the WebForms functioned needed to be consistent with the Project Request System. While previous knowledge was lacking, technical guidance and expertise were provided by staff at the K-12 Data Center to work through any issues that arose.

The ITS Program Database was determined to be functionally needed by two different groups of users; the campus users and ITS Administrative Computing staff. To allow for access by the different groups and to be consistent with the design of the site previously implemented, separate tabs were created for the groups of users to access their WebForms. These tabs provided the access to the WebForms by each group.

A Programs tab was created for campus users to have access to the program information they are able to run. When accessing the Programs tab, the user views a list of programs by viewing the program title and description. To access additional information for a program, the program title is hyperlinked to a program details page. The program details page
will give the user more detailed information about the program, including which system the program should be run from and the type of data that is used in the program.

The Programs Admin tab was developed for the ITS Administrative Computing staff to have access to all programs in the ITS Program Database. When accessing the Programs Admin tab, a list of all programs in the database is displayed. To view more detailed information or update information about a program, a hyperlink is included with the program title to access a program admin program details page. To add a new program, an option is included on the program admin’s program details page that will direct the user to the add program page.

While defining the information that needed to be recorded for each of the programs, there were some fields that were identified as having a need to be table driven. This information would be used by all of the programs that were entered and the decision was made to put the values on a table to control the format and values used for this information. SQL tables were developed to contain this support information. To allow ITS Administrative Computing staff to have access to these tables without needing access to the actual SQL tables, WebForms were developed that would allow ITS Administrative Computing staff to add additional entries to these support tables. The WebForms to update these tables is located on the programs page that is accessed from the Programs Admin tab.

A search capability was requested that allowed both campus users and ITS Administrative Computing staff the ability to search through their list of programs for a specific program. Initial programming for this search capability included searching the program title, program description, and program keywords only. After discussing some of the program aspects that would be stored in the program update notes and program creation notes,
the determination was made to include both program update notes and program creation notes as fields in the program search.

The addition of the keywords and department assignments were the last of the program information to be included. Both of these pieces of information can have multiple values for each of the programs. After discussion with the staff from the K-12 Data Center, it was determined that this information would be included in tables separate from the table that contained the program information that was already identified as being needed. A separate table was created to contain the keywords for each of the programs. An additional table was created to contain the departments assigned to each of the programs. The keys to these tables were ProgramId and KeywordId for the program keyword table and ProgramId and DepartmentId for the program department table.

When adding additional programs to the ITS Program Database, the initial program information is entered on the Add Program WebForm which is accessed from the Program Admin programs page. After the initial program information is entered and the add programs button is selected, the program admin program details page is displayed which will allow for changes to be made to the program information which was originally entered. The program details page is also where the keywords and departments for the programs will also be entered. Having these entries separate is necessary because of the separate SQL tables that are updated with this information.

Security for the ITS Program Database is handled separately for the ITS Administrative Computing staff and DSU campus users. Both of these security options needed additional help from staff at the K-12 Data Center. A security class already existed in the Project Request System for University access. This access was defined with limited user
access and allowed updates to the high level tables in the Project Request System. To allow ITS Administrative Computing staff access to the Program Admin tab and related WebForms, K-12 Data Center staff created an additional option in the University access security class that allowed access to the Program Admin tab. ITS Administrative Computing staff were then added to the University security class with Program Admin access.

K-12 Data Center staff also had to make a change to allow campus users to see the Programs tab. To allow access to the Project Request System, staff are added as members to a department. The department listed in the Project Request System is the department in which the user is employed in. Not all members of a department need to have access to see the Programs tab as not all members of a department have access to run the programs. Security for a program is being controlled by the department level. To allow access to programs for certain department members, K-12 staff added a flag on the departments screen which gives members of a department access to see the Programs tab. A combination of the user’s department being added to the program and the user’s program access flag being indicated on the departments options allows a user to have access to program information.

The actual implementation to update the Project Request System with the ITS Program Database tables and WebForms involved the help of K-12 Data Center staff. The files from the local machine where the ITS Program Database was developed were transferred to the production site.
CONCLUSIONS

The Oracle Migration project for the Colleague Student Information brought about an issue that there was no way to easily track all programs developed against the Colleague Student Information System. Having the ability to develop programs in many different options added to the problem of not knowing what is developed, where the program was developed, and for what purpose.

The objectives of the ITS Program Database project was to provide a way for ITS Administrative Computing staff and campus users to have access to programs in a user-friendly way, that is easily accessible, contains the pertinent program information, is searchable, and with a simple program entry point.

The development of the ITS Program Database met the required objectives of this project. The WebForms that have been developed are straight-forward and user friendly in the design and navigation. Adding additional program information allows for most information to be added on one WebForm. Access to programs for additional users can be given easily and quickly. The ITS Program Database was made easily accessible to all users through the existing Project Request System which is available through the DSU Portal. The DSU Portal is used to provide access to software and other information for all DSU Faculty, Staff, and Students. All programs contain information that is pertinent for both ITS Administrative Computing staff and campus users. The access to the ITS Program Database is allowed through ASP and C# WebForms. The search functionality of the ITS Program Database allows all users to easily find programs with search terms from the program title, program description, or keywords identified for the program.
The Program Database was developed with maintenance pages for the support tables in the database, which will allow ITS Administrative Computing staff to add additional entries directly to the tables on their own without direct access to the SQL tables. As the ITS Program Database is used, additional fields may be determined as needed for each of the programs. Additional information may also be determined as needed for campus users who have access to view the program information. These changes can be made in later additions to the ITS Program Database to make it even more valuable for users.

By using WebForms to allow ITS Administrative Computing staff the ability to add additional support table entries, maintenance of the ITS Program Database can be done mostly outside of having direct access to the SQL tables. Any additional program information that would need to be recorded would need to be updated to the SQL tables directly. Adding additional programs created in different systems would be possible with an addition to the system support table.

Further enhancements of the system may include allowing users to see a further listing of programs that have been created to access the Colleague Student Information System, but that they do not have access to run themselves. Additional fields would need to be stored in the program database and displayed in the WebForms to indicate which programs users have access to run and which programs are informational to let them know the program exists.

The entry of program information into the ITS Program Database is manual at this time. The systems being used to create the programs do not allow for any type of output of program information that could be used to electronically input the data into the ITS Program Database. A way to electronically update the system with additional programs and program information would be beneficial, but is not in the foreseeable future.
The ITS Program Database may develop into a tracking mechanism in the future to allow ITS Administrative Computing staff to know how utilized a particular program is. While we cannot link directly to the system in which the program exists to see if the program has been run, we may be able to track the programs that are accessed from the ITS Program Database. If another conversion were to take place or if updates are being made to particular programs because of data changes, knowing if a program is being utilized can help the ITS Administrative Computing staff know if the program needs to be updated.

An option is available for both ITS Administrative Computing and campus users to create an Excel download file all programs the user has access to. Additional reporting options might be needed in the future to determine the types of programs that have been developed, utilization of the programs, etc. A reporting section could be created for this purpose.
REFERENCES


Rapid Application Development, Retrieved October 31, 2015 from
APPENDICES

APPENDIX A: SQL CODE

**Code for creating tables**

Programmers Table:
CREATE TABLE [dbo].[Programmers]
(
    [Programmer] NVARCHAR (256) NOT NULL,
    PRIMARY KEY CLUSTERED ([Programmer] ASC)
);

ProgramAssignments Table:
CREATE TABLE [dbo].[ProgramAssignments]
(
    [ProgramId] INT NOT NULL,
    [DepartmentId] BIGINT NOT NULL,
    PRIMARY KEY CLUSTERED ([DepartmentId] ASC, [ProgramId] ASC)
);

ProgramKeywords Table:
CREATE TABLE [dbo].[ProgramKeywords]
(
    [ProgramId] BIGINT NOT NULL,
    [Keyword] NVARCHAR (256) NOT NULL,
    PRIMARY KEY CLUSTERED ([ProgramId] ASC, [Keyword] ASC)
);

ProgramPossibleKeywords Table:
CREATE TABLE [dbo].[ProgramPossibleKeywords]
(
    [Keyword] NVARCHAR (256) NOT NULL,
    PRIMARY KEY CLUSTERED ([Keyword] ASC)
);
### Programs Table:

```sql
CREATE TABLE [dbo].[Programs]
(
    [ProgramId] INT IDENTITY (200, 1) NOT NULL,
    [ProgramTitle] NVARCHAR (256) NOT NULL,
    [ProgramDescription] NVARCHAR (MAX) NOT NULL,
    [ProgramSystem] NVARCHAR (50) NOT NULL,
    [ProgramUpdateNotes] NVARCHAR (MAX) NULL,
    [ProgramCreationNotes] NVARCHAR (MAX) NULL,
    [ProgramDataType] NVARCHAR (50) NOT NULL,
    [ProgramProcessName] NVARCHAR (256) NOT NULL,
    [ProgramCreatesSavedlist] NVARCHAR (5) NULL,
    [ProgramWhereAccessed] NVARCHAR (256) NOT NULL,
    [ProgramInitialProgrammer] NVARCHAR (256) NOT NULL,
    [ProgramDateCreated] DATE NOT NULL,
    PRIMARY KEY CLUSTERED ([ProgramId] ASC)
);
```

### ProgramSystems Table:

```sql
CREATE TABLE [dbo].[ProgramSystems]
(
    [System] NVARCHAR (256) NOT NULL,
    PRIMARY KEY CLUSTERED ([System] ASC)
);
```

### ProgramWhereAccessed Table:

```sql
CREATE TABLE [dbo].[ProgramWhereAccessed]
(
    [WhereAccessed] NVARCHAR (256) NOT NULL,
    PRIMARY KEY CLUSTERED ([WhereAccessed] ASC)
);
```
APPENDIX B: RELATIONAL MODEL AND DATA DICTIONARY

Relational Model

- **Programmer**
- **ProgramAssignments**
  - ProgramId
  - DepartmentId
- **ProgramKeywords**
  - ProgramId
  - Keyword
- **ProgramPossibleKeywords**
  - Keyword
- **ProgramSystems**
  - System
- **WhereAccessed**
  - WhereAccessed

Diagram showing relationships and attributes of the relational model.
Entity Types

1. Name: PROGRAMMERS
   Type: Strong Entity
   Definition: A PROGRAMMERS is a list of possible programmers for a PROGRAMS. Only one PROGRAMMERS can be assigned to a PROGRAMS.
   Identifier: Programmer

2. Name: PROGRAMASSIGNMENTS
   Type: Strong Entity
   Definition: A PROGRAMASSIGNMENTS is a list of PROGRAMS and DEPARTMENTS. A PROGRAMS can be assigned to multiple DEPARTMENTS.
   Identifier: DepartmentId, ProgramId

3. Name: PROGRAMKEYWORDS
   Type: Strong Entity
   Definition: A PROGRAMKEYWORDS is a list of PROGRAMS and PROGRAMPOSSIBLEKEYWORDS. A PROGRAMS can be assigned to multiple PROGRAMPOSSIBLEKEYWORDS.
   Identifier: ProgramID, Keyword

4. Name: PROGRAMPOSSIBLEKEYWORDS
   Type: Strong Entity
   Definition: A PROGRAMPOSSIBLEKEYWORDS is a list of possible keywords for PROGRAMS. Multiple KEYWORD can be assigned to a PROGRAMS.
   Identifier: Keyword

5. Name: PROGRAMS
   Type: Strong Entity
   Definition: PROGRAMS is a list of PROGRAMS.
   Identifier: ProgramId

6. Name: PROGRAMSYSTEMS
   Type: Strong Entity
   Definition: A PROGRAMSYSTEMS is a list of possible systems for PROGRAMS. A PROGRAMS may only be assigned to one PROGRAMSYSTEMS.
   Identifier: System

7. Name: PROGRAMWHEREACCESSSED
   Type: Strong Entity
   Definition: A PROGRAMWHEREACCESSSED is a list of possible access locations for a PROGRAMS. A PROGRAMS may only be assigned to one PROGRAMWHEREACCESSSED.
   Identifier: WhereAccessed
Dictionary of Entity Attributes

Programmers
1. Name: Programmer
   Definition: A programmer’s name. This composite attribute has two attributes, first and last.
   Null: No

ProgramAssignments
1. Name: ProgramId
   Definition: This numeric value serves as the unique identifier of any program.
   Null: No
2. Name: DepartmentId
   Definition: This numeric value serves as the unique identifier of any department.
   Null: No

ProgramKeywords
1. Name: ProgramId
   Definition: This numeric value serves as the unique identifier of any program.
   Null: No
2. Name: Keyword
   Definition: This character value serves as the unique identifier of any keyword.
   Null: No

ProgramPossibleKeywords
1. Name: Keyword
   Definition: This character value serves as the unique identifier of any keyword.
   Null: No

ProgramSystems
1. Name: System
   Definition: This character value serves as the unique identifier of any system.
   Null: No

ProgramWhereAccessed
1. Name: WhereAccessed
   Definition: This character value serves as the unique identifier of any access location.
   Null: No
Programs

1. Name: ProgramId
   Definition: This numeric value serves as the unique identifier of any program
   Null: No
2. Name: ProgramTitle
   Definition: The title of a program.
   Null: No
3. Name: ProgramDescription
   Definition: The description of a program.
   Null: No
4. Name: ProgramSystem
   Definition: The system a program is located on.
   Null: No
5. Name: ProgramUpdateNotes
   Definition: The update notes of a program.
   Null: Yes
6. Name: ProgramCreationNotes
   Definition: The program creation notes of a program.
   Null: Yes
7. Name: ProgramDataType
   Definition: The data type of a program.
   Null: No
8. Name: ProgramProcessName
   Definition: The process name that is used to run a program.
   Null: No
9. Name: ProgramCreatesSavedlist
   Definition: Whether or not a program creates a saved list.
   Null: Yes
10. Name: ProgramWhereAccessed
    Definition: Where a program can be accessed at.
    Null: No
11. Name: ProgramInitialProgrammer
    Definition: The initial developer of a program.
    Null: No
12. Name: ProgramDateCreated
    Definition: The creation date of a program.
    Null: No
APPENDIX C: CODE FOR WEB FORMS

Code is redacted for printing.
APPENDIX D: SCREEN SHOTS OF WEB FORMS

Screen Shot of Programs Default Page:

Screen Shot of Program Details Page:
Screen shot of Programs Admin Page:

Programs Admin

<table>
<thead>
<tr>
<th>Program Title</th>
<th>Program Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athlete Reports</td>
<td>Creates a file of active athletes. Can run by either all sports or for a particular sport.</td>
</tr>
<tr>
<td>Download of Advisees</td>
<td>Creates a download of active advisees for an advisor.</td>
</tr>
<tr>
<td>Faculty Not Returning from Semester to Semester</td>
<td>A report of faculty not returning from one semester to another semester.</td>
</tr>
</tbody>
</table>

Screen Shot of Programs Admin Details Page:
Screen shot of Add Program Page:

![Add Program Page](image)

### Add a Program

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Title</td>
<td></td>
</tr>
<tr>
<td>Program Description</td>
<td></td>
</tr>
<tr>
<td>Process Name</td>
<td></td>
</tr>
<tr>
<td>System Name</td>
<td></td>
</tr>
<tr>
<td>Where Accessed</td>
<td></td>
</tr>
<tr>
<td>Data Type</td>
<td></td>
</tr>
<tr>
<td>Program Creates Savedlist</td>
<td>Select if program creates saved list</td>
</tr>
<tr>
<td>Initial Programmer</td>
<td></td>
</tr>
<tr>
<td>Program Creation Date</td>
<td></td>
</tr>
<tr>
<td>Creation Notes</td>
<td></td>
</tr>
</tbody>
</table>

**Buttons:** Add Program, Cancel
Screen shot of Add Keywords Page:
Screen shot of Add Programmer Page:

Screen Shot of Add Systems Page:
Manage Where Accessed

Where Accessed

Add Where Accesssed

Where Accessed

Cognos

Menu

SLCR
APPENDIX E: WORK BREAKDOWN STRUCTURE (WBS)

1 Identify
   1.1 Determine system flow
   1.2 Determine how to connect with existing system
   1.3 Determine fields needed
   1.4 Determine keywords to be used in searches
   1.5 Determine SQL tables needed
   1.6 Determine levels of user access
   1.7 Identify users
   1.8 Determine location for system link

2 Design
   2.1 Design landing page
   2.2 Design program entry form
   2.3 Design ITS search form
   2.4 Design campus search form

3 Develop
   3.1 Create SQL tables
   3.2 Create Program WebForm
   3.3 Create Program Details WebForm
   3.4 Create Program Admin WebForm
   3.5 Create Program Admin Details WebForm
   3.6 Create Add Program WebForm
   3.7 Create Update Table WebForms
   3.8 Configure user authentication

4 Testing
   4.1 Test WebForms
   4.2 Test AddProgram WebForm
   4.3 Test Security

5 Implement
   Update Project Request System with ITS Program
   5.1 Database Code
   5.2 Populate Support Tables
   5.3 Enter existing program information into database
   5.3 Notify campus users
# APPENDIX F: GANTT CHART

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Mode</th>
<th>Task Name</th>
<th>Duration</th>
<th>Deadline</th>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Project Database</td>
<td>75 days</td>
<td>NA</td>
<td>Mon 8/10/15</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Identify</td>
<td>10 days</td>
<td>Fri 8/23/15</td>
<td>Mon 8/30/15</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1.2 Determine system flow</td>
<td>2 days</td>
<td>NA</td>
<td>Mon 9/10/15</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1.2 Determine how to connect with existing system</td>
<td>2 days</td>
<td>NA</td>
<td>Wed 8/25/15</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>1.5 Determine SQL tables needed</td>
<td>1 day</td>
<td>NA</td>
<td>Wed 8/25/15</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1.6 Determine level of user access</td>
<td>3 days</td>
<td>NA</td>
<td>Wed 8/25/15</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1.8 Determine location for system link</td>
<td>1 day</td>
<td>NA</td>
<td>Thu 8/24/15</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1.3 Determine fields needed</td>
<td>10 days</td>
<td>NA</td>
<td>Mon 9/6/15</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>1.4 Determine keywords to be used in searches</td>
<td>5 days</td>
<td>NA</td>
<td>Wed 8/26/15</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>1.7 Identify users</td>
<td>10 days</td>
<td>NA</td>
<td>Mon 9/6/15</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>2 Design</td>
<td>8 days</td>
<td>Mon 8/31/15</td>
<td>Thu 9/3/15</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Design landing page</td>
<td>1 day</td>
<td>NA</td>
<td>Sat 8/29/15</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Design search form</td>
<td>2 days</td>
<td>NA</td>
<td>Sat 8/29/15</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>2.2 Design program entry form</td>
<td>1 day</td>
<td>NA</td>
<td>Sun 8/30/15</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>2.4 Design campus search form</td>
<td>2 days</td>
<td>NA</td>
<td>Sun 8/30/15</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Develop</td>
<td>99 days</td>
<td>Fri 10/2/15</td>
<td>Tue 10/27/15</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>3 Create SQL tables</td>
<td>5 days</td>
<td>NA</td>
<td>Sat 9/5/15</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>3.2 Create Program WebForm</td>
<td>10 days</td>
<td>NA</td>
<td>Sat 9/5/15</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>3.3 Create Program Details WebForm</td>
<td>4 days</td>
<td>NA</td>
<td>Thu 9/3/15</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>3.4 Create Program Admin WebForm</td>
<td>5 days</td>
<td>NA</td>
<td>Sun 9/13/15</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>3.5 Create Program Admin Details WebForm</td>
<td>4 days</td>
<td>NA</td>
<td>Fri 9/10/15</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>3.6 Create Add Program WebForm</td>
<td>5 days</td>
<td>NA</td>
<td>Thu 10/2/15</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>3.7 Create Update Table WebForms</td>
<td>7 days</td>
<td>NA</td>
<td>Sun 10/14/15</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>3.8 Configure user authentication</td>
<td>5 days</td>
<td>NA</td>
<td>Mon 10/19/15</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>Testing</td>
<td>22 days</td>
<td>Wed 10/28/15</td>
<td>Tue 11/3/15</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>4 Test WebForms</td>
<td>27 days</td>
<td>NA</td>
<td>Sun 10/13/15</td>
</tr>
<tr>
<td>26</td>
<td></td>
<td>4.2 Test AddProgram WebForm</td>
<td>6 days</td>
<td>NA</td>
<td>Mon 11/5/15</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>4.3 Test Security</td>
<td>7 days</td>
<td>NA</td>
<td>Mon 11/5/15</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>Implement</td>
<td>16 days</td>
<td>Fri 11/20/15</td>
<td>Wed 11/26/15</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>5.1 Update Project Request System with ITS Program Database Cafe</td>
<td>1 day</td>
<td>NA</td>
<td>Fri 10/28/15</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>5.2 Populate Support Tables</td>
<td>1 day</td>
<td>NA</td>
<td>Sun 11/1/15</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>5.3 Enter Existing program information into database</td>
<td>11 days</td>
<td>NA</td>
<td>Sun 11/1/15</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>5.4 Notify campus users</td>
<td>13 days</td>
<td>NA</td>
<td>Mon 11/5/15</td>
</tr>
</tbody>
</table>