**Capstone Project Report**

**CCSE Project Management System**

<https://sites.google.com/view/ccse-pmo-capstone/home>

**IT – 4983**

**Syed Ahmed, Wendy Claver, Faith Kinchen, Devin Greene**

**Computer College of Software Engineering**

**27 April 2020**

**Executive Summary**

This project was proposed by the College of Computing and Software Engineering to design and create a database repository to manage the CCSE Capstone, and Contract for Hire projects.. Early on the team decided to use Microsoft Teams for weekly interactions, and the GroupMe app for daily communication. Instead of depending on CCSE or requiring school resources, one of our team members hosted a virtual machine to use as our workstation. We used the Anydesk software to access our virtual machine, so that team members could work at the same time if need be. We begin by creating ER diagrams and table schema’s to map our database and proceeded to use these documents to begin creation of the Database in Visual Studio. After team discussion we decided to use Visual Studio as our main IDE. Members of the group have used this IDE before, and we were aware that it could process and implement a wide range of languages, including both front end and back in languages. The team decided that we would use the built in SQL server kit that comes with Visual Studio, and the database was subsequently coded in Transact SQL. After development of the database the team began development of the GUI. After much discussion it was decided to use C# and other built in functions of Visual Studio to create a web-based GUI as opposed to a in application GUI. Originally the team attempted to use the MVC framework to create the GUI, but after experimentation and team discussion, the team made a change to ASP.Net Core using Razor pages. Progress was made in a quicker fashion after the change and the GUI and the Web portal are now implemented with Razor pages on the Net core 2.2 framework. A basic version of Role Based Access is implemented with the same framework at the Portal, and the ability to handle PDF artifacts was developed using C#.

Contents

[Background 5](#_Toc38712100)

[Business and Project background 5](#_Toc38712101)

[Project Scope and Deliverables 5](#_Toc38712102)

[Technical Background 5](#_Toc38712103)

[Project Outcomes and Achievements 6](#_Toc38712104)

[General Project Success by Objective 6](#_Toc38712105)

[Project Success by deliverables 6](#_Toc38712106)

[Technical Summary 6](#_Toc38712107)

[Solution Approaches 6](#_Toc38712108)

[Summary of System development and design 7](#_Toc38712109)

[Project Management System Functionality and Description 7](#_Toc38712110)

[Manage Projects Component: 8](#_Toc38712111)

[Manage Sponsors: 8](#_Toc38712112)

[Manage Project Proposals: 8](#_Toc38712113)

[Manage Faculty Assignment 9](#_Toc38712114)

[Manage Student Assignment 9](#_Toc38712115)

[Portal with RBAC Summary: 9](#_Toc38712116)

[PDF Handler Summary 10](#_Toc38712117)

[Project Planning and Management 10](#_Toc38712118)

[Overview 10](#_Toc38712119)

[Project Processes by Milestones 11](#_Toc38712120)

[Milestone 1 11](#_Toc38712121)

[Milestone 2 11](#_Toc38712122)

[Milestone 3 12](#_Toc38712123)

[Contributions 12](#_Toc38712124)

[Overview 12](#_Toc38712125)

[By Team Members 12](#_Toc38712126)

[Workload Summary by Milestones 13](#_Toc38712127)

[Team Reflection 14](#_Toc38712128)

[Project Success factors 14](#_Toc38712129)

[Team Collaboration 14](#_Toc38712130)

[Meeting arrangements and experiences 14](#_Toc38712131)

[Collaboration System Use 14](#_Toc38712132)

[Challenges 15](#_Toc38712133)

[Technical 15](#_Toc38712134)

[Non-Technical 15](#_Toc38712135)

[Areas to improve 15](#_Toc38712136)

[Appendix 16](#_Toc38712137)

[Project Files 16](#_Toc38712138)

[Report List 16](#_Toc38712139)

# Background

## Business and Project background

This project was proposed by the College of Computing and Software Engineering at Kennesaw State University. Dawn Tatum was our project sponsor, with Adriana Clark serving and intermediary for logistics and administration needs. CCSE requires a repository for to manage the multiple projects that they are responsible for. These include Capstone projects, and contract for hire projects for both graduate and undergraduate students. The project sponsor indicates these projects are currently scattered across different sections of the CCSE and that they desire a software that will allow them to store and easily manage all the different kinds of projects in one place. Multiple people are associated with each project including Sponsors, KSU faculty, and Students that are involved in the project. Many of these projects also require MOUs and NDAs. This leads to a large assortment of Data and documents to be kept up with per project, and this complexity is further compounded by the fact that some sponsors may sponsor multiple projects, and that multiple staff can be associated with one project, or one staff member with multiple projects. The general goal of this project was to design and create a database and the accompanying GUI that would allow CCSE to store and organize projects and their relative information in one application.

## Project Scope and Deliverables

The scope of the project originally encompassed three different kinds of projects but was later reduced to the two kinds of projects mentioned earlier in the document. (Capstone, and Contract for Hire). The project also includes tracking the information located on the project proposal pdf documents, which include the Sponsors’ name, contact information, business, and position in the business. The project was also expected to track student and faculty information, such as name and email. The main objectives of this project where to create a database to track the above information, and to allow the user to run reports and view information relative to each project.

## Technical Background

We used a diagraming software called Dia, that allowed us to chart our Entity relationship diagrams, and our Table Schema. This software was chosen out of familiarity as the team had prior experience using this software in the past. Both documents are included in the final report folder.

This project was developed using the community version of Visual Studio 2019. Several packages where installed alongside our version of Visual Studio, including SQL server, ASP.Net MVC andASP.Net Core frameworks. Visual Studio is one of the leading IDE to develop for windows environments and can develop front end and backend. We also chose it for its extensive amount of documentation, and wide variety of third-party tutorials. The front end of the application was developed using Razor Pages which allows pages to handle request directly, without the help of a controller. The back-end database was developed using Visual Studio with the SQL Server package installed.

A basic version of Role Based Access Control (RBAC) was implemented at the portal, in which admins are given the ability to create new accounts. There are three levels of accounts, Admin, user, and viewer. We often found during research that RBAC is normally implemented after the deployment of a product into its environment, and uses its own database, but due to sponsor request, our team created the above roles and implemented a system that allows each differing levels of control over the GUI.

# Project Outcomes and Achievements

## General Project Success by Objective

The project was overall successful. The team was able to deliver the basic requirements listed on the Project Proposal documents of a GUI, database, and the ability to enter data and run reports. The GUI allows for the manipulation of all the tables in the database while simultaneously protecting important keys and identifiers. The team was also able to deliver the objectives of implementing RBAC and handling of PDF documents that where requested by the project sponsor. The portal includes the ability to sign into the application under different accounts Some parts of the project may not live up to par to the sponsors expected image, but all parts should fulfill their intended function, and satisfy the objective given to the project team.

## Project Success by deliverables

The three deliverables where divided into the follow milestones:

The first milestone was our ER-Diagram charts and Table Schemas. The team was highly successful at this milestone. The sponsor was pleased with this milestone, with the exception that our original schema had too much detail. The team was able to provide a less intricate and more streamlined table schema at the next sponsor meeting.

The second milestone deliverable was the database itself, and the team was also highly successful wit this milestone, as the database was completed before the milestone was due, and had been through multiple iterations and improvements before the milestone was presented to the project sponsor. The database continued to see changes and improvements as the project progressed and new information was brought to light by the project sponsor. The team managed to take the new information into account while maintaining the more streamlined database that was presented at the second milestone meeting.

The third milestone was the database with the GUI and the ability to input data and run reports. The team was also able to provide this milestone, with the caveat that certain upgrades and quality of life improvements that are in the final product where not shown in the demoed version of the project. The PDF handling functionality was also demoed and showed that it could store files to the Database that the GUI interacted with. The only flaw with this section being that it was not available through the GUI at that point during the project. This milestone was successful in the sense that the group was able to demo all the requirements listed on the project proposal, and the requests of the project sponsor such as RBAC and the handling of PDF documents. The team’s largest qualm with this milestone was the inability to demo as a singular app in one solution file.

# Technical Summary

## Solution Approaches

The team chose to begin design of the database with the creation of an ER diagram and table schema, which were used to create the original database after the approval of the project sponsor. The database was designed next using the table schema and saw multiple changes throughout the project life.

After the creation of the database, the team went into a research phase to gain understanding of foreign concepts. During this time drawings and ideas for the GUI where created and discussed. Many meetings revolved around how exactly higher level concept such as role based access would be implemented, and the best way to avoid creating a system that would be needlessly complex in the event that the database was integrated into the KSU Active Directory domain. A large majority of discussion was centered around debate on creating and in application GUI or a web-based application. After drawings and ideal sketches of the GUI where finished, and the team had decided the best way to implement requested concepts work was divided between team members. Originally, the project was created using the ASP.net MVC framework, but as the project progressed, the team found some concepts extremely difficult to implement using this framework, and soon after the second milestone the team chose to change from the MVC framework to the ASP.Net Core framework, which allowed team members to make faster progress than was seen using MVC.

The team was able to implement both the GUI and a basic version of role based access control using ASP.Net Core and eventually settled on a presentation of the GUI that allowed for admins to view data and sort it by select database columns. The GUI also includes a report page that allows for columns from multiple tables to be viewed together, namely the Project number, sponsor name, student names, and faculty names. All these columns are also searchable within their respective tables.

Role based access was implemented using the database’s Faculty and Student tables, and at the request of the project sponsor supports three levels of users: Admin, User and Viewer. Each role has an individual log in portal, to facilitate the routing of each role to its own version of the GUI. The only real difference between the GUI pages for each role being their access the CRUD(Create, Read, Update, Delete) functions.

## Summary of System development and design

The Project Management System has been developed using Asp. NET Entity Framework CORE in Visual Studios 2019. We have used SQL Server Express and the LocalDB for development purposes. The KSU CCSE PMO includes the following features:

* Managing Projects
* Managing Sponsors
* Managing Faculty Project Assignment
* Viewing Project Proposals
* Searching projects
* Querying/Exporting and Printing reports

## Project Management System Functionality and Description

**This section gives a brief accounting of the basic components of the CCSE PMO:**

### Manage Projects Component:

The goal of this component is to manage all functionality related to CCSE projects. In this component, all CRUD (Create, Read, Update, Delete) operation have been developed. These operations allow the administrator to manage and edit all projects commissioned through the CCSE and related departments.

**Functions of this component include**:

* Administrator can view a list of projects
* Administrator can view a list of all projects
* Administrator can sort, search and filter projects
* Administrator can add projects
* Administrator can edit projects
* Administrator can delete projects
* Administrator can view project details.

### Manage Sponsors:

The goal of this component is to manage all CCSE project sponsors. In this component, all CRUD(Create, Read, Update, Delete) operation have been developed. These operations allow the administrator to record, view, manage and edit important Sponsor information. This includes the number of projects sponsored and the presence of any Non-Disclosure Agreements

**Functions of this component include**:

* Administrator can view all Sponsor related details
* Administrator can sort, search and filter Sponsors
* Administrator can add Sponsors
* Administrator can edit sponsor details
* Administrator can delete sponsors
* Administrator can view sponsors list

### Manage Project Proposals:

The goal of this component is to view all CCSE project proposals submissions . In this component, all CRUD (Create, Read, Update, Delete) operation have been developed. These operations allow the administrator to record, view, manage and edit important proposal related information used for project reports. This includes the number of pros

* Administrator can view proposal related information
* Administrator can sort, search and filter proposals
* Administrator can add project proposal
* Administrator can edit project proposal
* Administrator can delete project proposals
* Administrator can see a list of project proposals

### Manage Faculty Assignment

The goal of this component is to view and edit faculty project assignments. In this component, CRUD (Create, Read, Update, Delete) operations have been developed. These operations allow the administrator to record, view, manage and edit details relating to faculty project assignments. This includes the number of projects a faculty member is assigned to and what department they are in.

**Functions of this component include**:

* Administrator can view faculty assigned to the project
* Administrator can sort, search and filter faculty
* Administrator can add Faculty
* Administrator can edit Faculty details
* Administrator can delete Faculty
* Administrator can view faculty list

### Manage Student Assignment

The goal of this component is to view and edit student project assignments. In this component, CRUD (Create, Read, Update, Delete) operations have been developed. These operations allow the administrator to record, view, manage and edit details relating to student project assignment.

**Functions of this component include**:

* Administrator can view students assigned to the project
* Administrator can sort, search and filter students
* Administrator can add students
* Administrator can edit student details
* Administrator can delete student
* Administrator can view student list

### Portal with RBAC Summary:

The PMO Portal was created using Razor Pages along with ASP.NET Core 2.2 in Microsoft Visual Studio 2019. The programming language used was C#. It relies on the use of two NuGet Packages. These packages are were BCrypt.Net-Core 1.6.0 by Steve Donaghy and Microsoft Entity FrameworkCore SQL Server .2.4 by Microsoft for ASP.NET Core 2.2 Compatibility. It writes to the credentials database that stores all user login information. Login information consists of a user’s Net id or ID, in case for Sponsors, along with their password and name. Each user has their own login portal that redirects them to their respective role. They are redirected to a razor page that has developed for the portal. For example, Faculty users are Admins. Student users are Users. Student users are Viewers. Faculty users can view all GUI functionalities. They can view, edit, and delete all data such as project, student, sponsor data. Faculty members also have the functionality of adding users to the PMO Portal. Students can only view and edit their own data. Sponsors can only view their data such proposals. You are denied access if you try to log in to different role that you were not assigned to. For example, sponsors cannot log into faculty portal and only faculty member have the ability create new users. BCrypt.Net-Core is used to compare the password input by user on the portal to the hashed password stored on the data base. If they match, then they start a new session and are successfully logged in to their respective GUI role.

### PDF Handler Summary

This project runs separately to the main website but is connected through the database. This project is in the Project PDF UploadDownload folder. The main code is in the Default.aspx.cs file, coded in C#. The first method connects the page to the database by calling the connection string to the database that is in the Web.config file. The Upload button is connected to the Button1\_Click method which checks if a file has been uploaded, and if it is a PDF file. If both are true, then it gets the filename, extension, type and adds it into the PDFFiles table in the database. The View files button is connected to the Button2\_Click method which queries the PDFfiles table. The GridView1\_SelectedIndexChanged pulls the project files from the database and presents them in a viewable, downloadable format. The Default.aspx file is the visual for the page. The Web.config file is what contains the connection string which can be changed to link to any database. The PDFFiles sql file is a table which must be created in the database for the files uploaded to be stored there. The Project\_ID must be manually entered for each file uploaded.

# Project Planning and Management

## Overview

The team chose to meet virtually once every week until the end of the project. The team met Friday of every week with the exception the weeks that there was a meeting with the project sponsor. The team met on the day before the meetings (Wednesday) with the project sponsor to ensure that we were prepared to demo for that meeting. The project team went ahead of schedule by creating the database soon after the sponsor gave approval on the new table schema, which was some weeks ahead of the time that the database was due for the milestone two update.

The project was divided up into section for each group member. After the creation of the table schema, and the database the team was assigned separate roles in order to “divide and conquer”. Much of the project teams time was spent doing research on computer science concepts foreign to information technology students. Many resources where used during research such as:

* Microsoft Documentation for Visual Studio, ASP.net, and C#
* Written 3rd party tutorials
* 3rd party video tutorials
* Other used sources such as W3 schools, Stack overflow, and Auth0.

The research was broken into multiple sections. Each team member was expected to develop and implement the requirement that they researched. These research fields where divided into the following:

* GUI development
* Artifact(PDFs) handling
* Role Based Access Control
* Querying and Reporting

The team members where each assigned one of these concepts for research early in the semester, almost as soon as the database was completed. It was necessary eventually for teammates to swap assignments at some points for a more rounded view of the concepts being researched, and to assure that each member was able to implement their assignment when it came time to develop.

After the research phase the team continued to work separately, occasionally an assignment was traded between, or handed off to another groupmate when obstacles arose, and one groupmate was stuck. This allowed the team to develop each requirement simultaneously, while also allowing the members to decide if some implementations where to in depth for the scope of this project or beyond the capabilities of the group. Each requirement went through multiple iterations, some were rolled back to previous states due to the complexity of a chosen method, and others simply saw improvements.

## Project Processes by Milestones

### Milestone 1

The required deliverable for this milestone was documents showing a solid plan for the design of the database. Originally, research was done to refresh the team on database concepts such as primary and foreign keys, junction tables, and general SQL syntax such as join statements. The team successfully drafted an ER diagram followed by a table schema for the database repository. After the presentation of the ER Diagram, the team received feedback that the diagram was to detail and involved unnecessary tables and data columns. By the date of the first milestone, the team had edited both the ER diagram and the table schema to the liking of the project sponsor.

### Milestone 2

The required objective for this milestone was the completion of the database. Originally, the team continued its research into database concepts in relation to linking tables and joining tables in SQL queries. The project team was able to develop the database based on the table schema well before the due date of the milestone and spent a significant amount of time on research of concepts requested by the sponsor for the third milestone. The project team was able to sufficiently link tables and write queries that allowed information from across tables to be accessed and began drafting sketches and ideas for the GUI. The original version of the portal for the application was also completed.

It was during this milestone that the project team was forced to change from the MVC framework to the asp.net Core framework that the project is developed in. After a significant amount of work and experimentation with the MVC framework, the project team found that progress was not being made at an adequate pace. The team made the decision to change the GUI and the portal to the less complex ASP.Net Core framework and was able to progress the project at a pace more consistent with the team’s Gannt chart.

### Milestone 3

This was the final milestone and required the completion of the GUI, and the ability to enter and edit data into the database. This was also the last chance for the team to demonstrate any features requested by the project sponsor such as role-based access and artifact handling. The team was able to demo the portal and the final version of the GUI during this milestone and was also able to show the applications ability to upload and store PDF documents. The team presented some of the code and the overall concepts behind the implementation of role-based access in the application. They also demonstrated the GUIs ability to sort and search data, as well as the screen that pulls information from multiple tables to show a report. The team’s largest qualm with this milestone was that it was unable to demonstrate all of these capabilities in a single project solution, as technical failure of the teams virtual workstation caused the loss of several days of work. Despite this, the team worked diligently to recombine the work of the team members back into a single solution file.

## Contributions

### Overview

The team was not divided up into traditional roles such as researcher and documenter. This allowed the team to be flexible with the work and research that they did, while also allowing them the freedom to “trade” work or research assignments that may have been frustrating them. While each team member was eventually designated a field (RBAC, GUI development, etc) to research and develop in the early stages of the project the entire team researched concepts such as role based access, artifact handling, and database development, and this early stage of broad research often allowed team members to assist other members during development.

### By Team Members

**Faith Kinchen:**

* Sketches of GUI
* Development of GUI
* Development of reporting feature
* Documentation

**Wendy Claver:**

* Drafts of web portal
* Development of web portal
* Implementation of RBAC
* Documentation

**Syed Ahmed:**

* Upkeep of project website
* Upkeep of database
* Development of artifact handling
* Documentation

**Devin Greene:**

* Development of Database
* Creation of SQL queries
* Submission of weekly reports
* Documentation

## Workload Summary by Milestones

This section is a summary of the man hours of the project broken up by milestone and deliverables. Deliverable times may not equal total milestone times, as often work and research was done for upcoming milestones during a previous milestone, and testing and documentation are included in the total milestone hours. It should be noted that almost every part of the project saw changes or improvements throughout the project duration, that are not recorded on the Gannt chart, due to the minutia of the changes.

**Milestone 1:** Approx. 110 hours total

**Deliverables:**

ER Diagram: Approx. 10 hours

Table Schema: Approx. 12 hours

**Milestone 2:** Approx. 144 hours total

**Deliverables:**

Developed Database: Approx. 23 hours

**Milestone 3:** Approx. 134 hours

**Deliverables:**

GUI: Approx. 35 hours

Reporting function: Approx. 26 hours

# Team Reflection

## Project Success factors

The most important factors to the success of this project where the team’s willingness to study and learn concepts and the ability to relinquish responsibility to other groupmates. The team also showed aplomb with the ability to change gears in the middle of the project when the group realized that the current framework was not going to allow for an adequate rate of progress. They readily changed to a simpler framework that allowed for a faster rate of progress, despite this change leading to a significant difference in the development compared to the sketches and plans already made.

The Project owner was impressed with our attention to detail regarding the database ER diagram and table schema and was happy with the amount of thought that went into the planning database. We also received praise when demonstrating or the portal to the app with a sign in for Sponsors (which is viewer level) and where told that this was not something the sponsor considered and was valid option of users for the database.

## Team Collaboration

The team chose to use Microsoft Teams for meetings, information sharing and general collaboration. We used a virtual machine hosted on a team members system as a virtual workstation and used the Anydesk software to access the VM. After a certain point, the team chose to divide work and continue to develop on our personal machines, as the VM often had difficulties when multiple where accessing the VM. The team has decided in hindsight that was likely one of the biggest mistakes made during the project life, as bringing together multiple solutions into a single solution file led to undue stress and required more time than anticipated by the team.

### Meeting arrangements and experiences

Meetings were an overall positive experience for the group. Meetings generally stayed on task and each group member participated in conversation and presented ideas for the best way to tackle problems. Conversation was generally consistent and members where happy to share information that they found during research and collaboration of information and concepts was generally high.

## Collaboration System Use

The team chose to use Microsoft Teams for our weekly meetings and the GroupMe app for day to day communications.

**Microsoft Teams:**

This is a collaboration and video call software developed by Microsoft that includes shared calendars, file sharing, screen share, and other collaboration tools. The team often found the screen sharing feature useful for collaboration and teamwork on exceptions and errors. The software as easy to use and well designed.

**GroupMe:**

Is a popular mobile app that allows for the creation of groups. It allows for mass messages to be sent easily, but also generally functions like most other text-based communication apps such as Discord, Hangouts, or Slack. The team mostly used this app for day to day communications, and to clarify questions brought up during team meetings. This application is simple and easy to use after a group has been established.

**Anydesk:**

Anydesk is a remote access software that allowed for the team to access the VM hosted on a team members machine. This was the main purpose of the software, but it also allowed for team members to easily upload and download files between the VM and their personal machines. The software was also chosen for its ability to allow multiple users to access the VM at the same time, but issues where had with this later on.

## Challenges

### Technical

Some of the technical challenges the team faced included issues with both Anydesk and GroupMe. One of our group mates stopped receiving GroupMe messages late into the project, but this was quickly remedied by switching to email, and was late enough into the project life that it was not a terrible challenge.

The Anydesk software posed a slightly more difficult challenge. It became obvious during development that the software (or possibly the VM) was extremely slow whenever the entire team was present on the VM, and despite its capability to allow all team members to access the VM simultaneously, it offered no way for members to work on separate assignments at the same time. This was one of the main reasons that team member eventually chose to develop on personal machines and then upload solutions to the VM.

### Non-Technical

The biggest non-technical challenge that faced the team was the research of foreign concepts and frameworks, followed by the division of development on separate machines. While the team did not face a plethora of issues when handing over assignments to other team members, it did face much difficulty in combining solutions built on separate machines. Due to the technical difficulties described above, major pieces of the project where developed separately from one another on individual team member machines. This led to an unexpected amount of work to combine solutions into a cohesive project file. This remains the number one mistake in the eyes of the team.

## Areas to improve

Overall, the team agrees that finding a way to develop in a single project file would have made the greatest overall improvement. This would have required a tighter structured work schedule, and perhaps a schedule of when the VM was open, and when it was being used would have been a viable solution to the technical difficulties mentioned above, and would have kept the developments inside a single solution. Alternatively, the team could have looked for a different software that allowed for team members to work on the VM at the same time, but work on separate assignments.

# Appendix

## Project Files

**Analysis and Design:** This folder contains documents such as the ER Diagram and Table schema that was used for developing the database. It also contains the original questions from the first sponsor Q&A, as well as sketches for the first version of the GUI.

**CCSE PMO Project:** This file is the project itself. It includes READMEs, the database and database context that the project uses, as well as the projects solution files.

**Database Copies:** These are copies of the Database, database longs, and the database context that the project uses. They are the same as the files found in the CCSE PMO Project folder.

**Documentation:** This folder contains the read me files written by the team members, as well as the summaries of the different parts of the application

**Learning Materials:** This is a list of resources and links that the team used to create parts of the application, or to learn foreign concepts.

**Photos:** This folder contains screenshots used in PowerPoints or shown to the project sponsor.

## Report List

**Milestone presentations:** This file contains all the milestone presentations in powerpoint and PDF formats.

**Weekly Reports:** This file contains all of the weekly reports that where due during the semester, as well as the final report log.

**Project Plan:** This document is the project plan approved at the start of the semester by the project sponsor.

**Project Workload Planning and Tracking:** This document is the Gannt chart used to track hours worked and to gauge progress on the project throughout the semester.