# *Maintenance Project (420-E71-HR)*

# *Task 2 – The Project – Ramp up & Analysis*

Date assigned: Monday, January 29, 2018

Date due: **Monday, January 29, 2018, 12:00pm**

**Learning Objectives**

Upon successful completion of this task, the student will be able to:

* Understand the maintenance requirements for the systems in question.

To do:

**Task Set-Up**

Save this document with the name **YourUserName\_E71\_A02\_Project.docx** in the 420E71 folder of your H drive.

## Part A – The System

**NOTE:** As you are going through the systems, if you find what you think is a bug, document it the project’s TFS. This includes any artifact, software, documentation, etc.

**CSAdmin**

* CSAdmin an the administrative system to other systems at Heritage such as FACE, Track, CES, Hers..
* Currently in VB.net, needs to be rewritten in C#
* Need to remove all stored procedures and convert them to the business logic layer
* Need to remove direct ties to Clara (currently a stored procedure), must use the Clara views
* Ned to standardize the user login
* Used LDAP authentication

The user base for CSAdmin is Allan, but the system has already become a very important part of the business of the college. Changes to CSAdmin involve adding new features and removing older ones; we will also solicit the user for change requests for CSAdmin.

This is a production system so a significant portion of the maintenance planning has to include detailed testing to ensure that new problems are not introduced.

The CSAdmin system is a specialised system that will only be used by one or two people a term. CSAdmin helps the Computer Science System Administrators (Allan) in maintaining the remaining systems.

Given the above list of tasks we need to make some hard decisions. Your task at this point is to determine which area(s) you think we should concentrate on in which order, or, perhaps, how the work should be split up amongst our nine-person team to complete the work appropriately.

To do this you will need to look at:

* The current databases on CSDEV (CSAdmin and the Clara database on CSOracle) to see what needs to be done to them;
* The administration features that have already been implemented, which ones are missing, which ones are incorrect or incomplete.
* Acquaint yourself with the CSAdmin system. Logon to the system on CSDEV under csdev.cegep-heritage.qc.ca/Projects/CSAdmin. However, I would recommend that you take a copy of the system from Team Foundation Server (CSAdmin-2018-MR-team1/2) and run it through Visual Studio. That way you will be able to see the code, etc.
* The defects and backlog items themselves in Team Foundation Server for each system. You will need to group/rank them and see what kinds of defects are there, how many appear to be related to each other, the severity of the defects and the potential ease of fixing them (you will likely need to look at the code around where some of the defects are found). Note: this list will evolve as we baseline these systems.
* Here is a list of work items to start off with: <http://cstfs.cegep-heritage.qc.ca:8080/tfs/CSAdmin/CSAdmin/_workitems#_a=query&path=Shared%20Queries%2FAll%20Work%20Items%20Not%20Closed>

**Question 1: Answer the following questions:**

1. What is CSAdmin used for?

**CSAdmin is used to control the roles in all of the applications that the CS department creates**

1. How many users does CSAdmin have, and who are they?

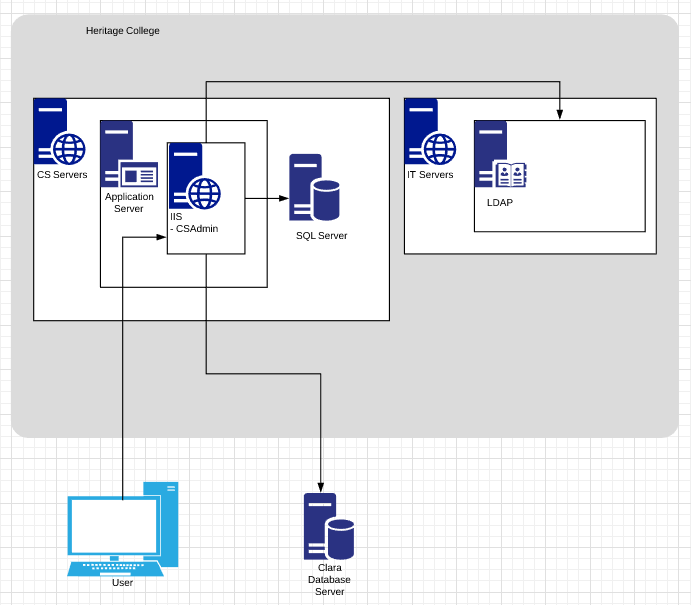
**Currently one user, and its Allan.**

1. Explain how you are going to remove the ties to Clara, and what difficulty you will encounter.

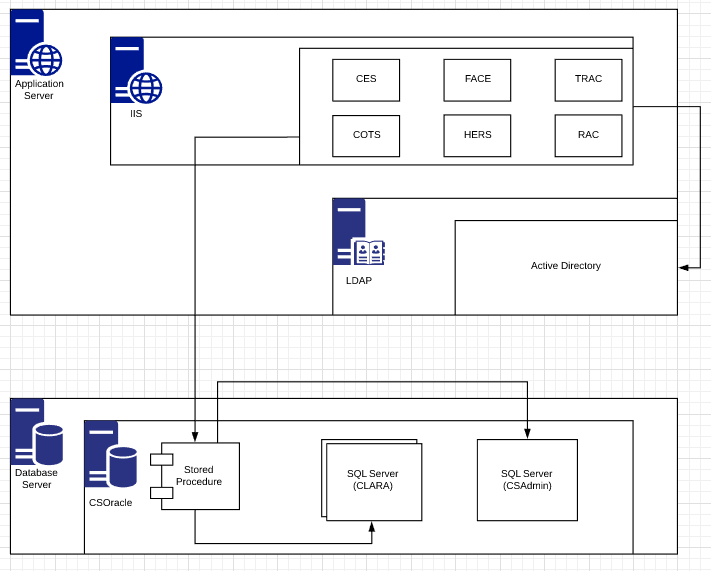
**We’re going to make views to pull in data from Clara.**

**Question 2: Please create the following architecture diagrams for the CSAdmin system:**

1. one **physical architecture** diagram



1. one **component** diagram



1. two **deployment** diagrams
   1. one for the **dev** environment
   2. one for the **test** environment

## Part B – Voyage of Discovery

**TO BE DONE INDIVIDUALLY with a lot of consultation with each other**

At the beginning of many maintenance projects is a discovery phase where you ramp up on the project and discover what the current artifacts of the system are and what state they are in.

**Question 1:**

Identify the current state (available/Not available, complete/not complete) and where to find the following artifacts:

|  |  |  |
| --- | --- | --- |
| **Artifacts** | **State** | **Location** |
| Application Stakeholders and Roles | Not Available | Not available, but we know who they are |
| Project History | Available | TFS > Code > History |
| Documentation | Available / Not complete | TFS > Work > Stories |
| Software, Database, Other technical artifacts | Available | TFS/Visual Studio, SQL Server |
| Bugs and Issues | Available | TFS > Work > Queries > All work items not closed |

**Question 2:**

Black box analysis and walk through. Logon and use the application.

Fill in the tables below:

Work items raised by Allan:

|  |  |  |
| --- | --- | --- |
| Work Item # | Title | Severity |
| 1 | As a user I do NOT want to see default error screens. |  |
| 142 | As a user, I am only allowed to create or update application user roles that are duplicates |  |
| 160 | Need to remove direct ties to Clara |  |
| 164 | CSAdmin must use the IIS AppPool |  |
| 165 | As an Administrator I need to add users for applications even though the user is not in Clara |  |

New bugs raised by me:

|  |  |  |
| --- | --- | --- |
| Work Item # | Title | Severity |
| 1 | Search filters don’t apply when you enter them, search, and then change the radio buttons on the side | 3 |
| 2 | The page doesn’t compress to a smaller screen size and is not at all mobile responsive | 4 |
| 3 | Clicking the “replace” button when looking at users will crash the system. | 1 |
| 4 | Entering a new application with the same name and code as an existing application won’t throw an error, but it will not be added to the list. | 3 |
| 5 | You can edit an application to have the same name and code as another application | 2 |
| 6 | You do JS injections in textboxes on the lookups > languages page | 1 |
| 7 | You can break the edit application functionality by adding an opening script tag into any of the text boxes | 1 |
| 8 | No edits will validate against <script> | 1 |

**Question 3:**

It is your job to be familiar with what’s in TFS Backlog items and ensure you don’t submit a duplicate (generating more work for others to triage). We will review the Backlog items later.

Come up with one or two suggestions for how we (as a project team) should approach the maintenance cycle. You need to consider the fact that we have about 80 hours x 9 people. You also need to consider that some changes may be closely tied together and may need to be worked on at the same time. In your suggestion(s) you must think about working on one area as a team (everyone bug fixing) or dividing up the areas to individual members of the group. Remember, one of the tasks of a project manager is to make sure everyone gets to learn all parts of a system and that no one person is stuck doing the same tasks all the time. This means that having someone always working on one thing (like user interface) is not a viable option.

When you have determined your suggestions, document them and provide reasons for choosing these suggestions. Be prepared to support your answers in a group discussion.

The system should be maintained following an agile development methodology. The system will be maintained over the course of the next 14 weeks, which is a very short amount of time. Because of this, the system maintenance should follow an agile development methodology.

I think that we should create different roles for our team; people that work on converting the old VB code, people that work on adding new functionality, testing, etc., and then rotate these roles at each sprint. This way everyone is responsible for certain pieces of functionality over the course of an entire sprint, but everyone still gets to learn all the different pieces to the system. One person also won’t just get tired of working on the same thing constantly, so there’s overturn in what you get to work on.

**To submit**

When you have completed the task:

* Save this file and copy it to the appropriate location in Moodle.