# *Development Project II (420-E63-HR)*

# *Assignment 6 – Operational & System Documentation*

Date assigned: Monday, April 8, 2018

Date due: **Friday, April 13, 2018, 11:50PM Part A – Runbook Draft**

**Monday, April 23, 2018, 11:50PM Part A – Runbook Final**

**Friday, April 27, 2018, 11:50PM Part B - System**

**Learning Objectives**

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

* Describe how to copy databases in SQL Server
* Copy an existing database in SQL Server
* Deploy a new web application to a web server
* Secure a database connection on an IIS web server
* Update a Runbook for deploying a web application and a database
* Create a System document for the Maintenance team

**Students are required to pass this assignment in order to pass the course.** Students who complete and fail this assignment before late marks are deducted may correct and resubmit the assignment for a maximum mark of 60% less the late marks on the original assignment.

You may do research and investigation as a team, however the documentation and submissions for this assignment is **individual work only**.

To do:

Milestones

|  |  |
| --- | --- |
| Date | Milestone |
| Friday, April 13 | Runbook skeleton laid out, sections identified, intro, initial recipe on how to deploy (this is needed on setting up for system test) |
| Monday April 23 | Final Runbook submitted |
| Friday April 27 | System documentation submitted |

## Part A - Operational Documentation - Runbook

You will create a [Runbook](https://en.wikipedia.org/wiki/Runbook) that will capture your operational documentation. A template has been provided for you in Moodle.

You are free to modify the template, layout or organization as long as the required information is provided and well organized.

One of your goals is that with just the Runbook, access to your software (tools, scripts, repository, etc.) and access to the production environment, the system administrators are equipped to install, configure, run, monitor and maintain your system.

Recall, that a typical production environment may have 100’s or 1000’s of systems simultaneously running and that system administrators depend on well documented, well-packaged systems.

Save your Runbook as **YourUserName\_Teamxx\_E63\_A06\_RunBook.docx.** Include your (short) username and team number.

### Setup and Prep

1. Take a look at the Runbook template. Determine how to organize and capture all the relevant details of deployment in the Runbook. Add additional sections to the Runbook if needed.
2. Figure out how and where you will store your deployment and migration items (i.e. any scripts or tools that you will reference from the Runbook that your users will need).
3. Research and ensure that you know:
   1. How to publish an application
   2. How to copy and deploy a database
   3. How to securely setup an application and a database (consider ~~HTTPS,~~ App Pools, no hard-coded usernames/passwords).
   4. How to use the Redgate tools to perform version control, copy and deploy databases.

#### Terminology and Concepts

Target Server – A specific location or server that will host a service such as a database or application components.

Deployment instructions in your Runbook should be able to work on a variety of instances. For example, you shouldn’t hard-code in your instructions the names of target servers (i.e. which server the Database will run on, or what server the Application will run on). You’ll generally explain how to work on any Target Server as long as the Target Server meets your clear and precise requirements.

For example, in our environment, the System Administrator should be able to follow the document to install the application on CSTEST or CSPROD.

### Installation

1. Describe the requirements for installation of your system.
2. Describe how the steps to install on a selected Target Server.
3. Make it more concrete by tying this into how this would work on installing it on a specific system. Select our CSTEST environment.
4. How to confirm that the installation is successful.

Hints:

1. You can highly reference the general section, and indicate the details (perhaps in a table format) to substitute in for the concrete example. i.e. a table with parameters, such as Target DB server, Target Application Server, and all the other selectables). Do not list all the steps and details again.
2. Read and Understand Appendix I

### Upgrade/Downgrade

If your system supports in-place upgrades, with or without the persistence of users’ data, you need to explicitly describe what is possible.

Specifically, if there is already an instance of your application running on the Target servers and a new release is available.

### Monitoring

Monitoring is understanding what is going on in the system.

It should cover:

1. How to determine if the application is functioning properly
2. How to determine if the application is performing properly

### Maintenance

Maintenance includes any tasks that needs to be done for your system to keep it healthy.

Details for each task should include task purpose, steps and frequency (or what is the trigger to do this task).

Typical tasks include:

1. How to startup or shutdown the application (ideally, in a graceful way)
2. How to trim logs or other output that, left unchecked, might blow up a file system.

## Part B –System Documentation

Consider maintenance teams that will inherit the wonders of your creation. This is the audience for the System documentation. It must cover details on how the system is implemented, including (but not limited to):

1. Coding and organizational standards (where to find what)
2. Architecture (i.e. layers and blocks), high level design of the system
3. Data Model
4. How to find the code, build and run it.

This does not have to be very large, but sufficient details to give the maintenance team context so that they can work with the project.

You may freely copy and update from your past design documents.

Once again, your audience is 3rd yr Computer Science student that has no concept of what your project does, how it’s made or how to run it.

Present this in a report format (Title page, TOC, Introduction, References, Body, ….)

Save this as **YourUserName\_TeamXX\_E63\_A06\_SystemDoc.docx**.

**To submit**

Submit portions to the appropriate folders in Moodle. Note the Rubric also provided in Moodle on how this assignment will be assessed.

# Appendix I – Installation Notes and Hints

## Installation Overview

This section involves writing an installation guide.

Installation instructions should include the following details:

* 1. Instructions on how to open the application from source control in Visual Studio, including steps on connecting to the TFS server.
  2. Instructions on how to deploy both databases to the test database server using the Redgate tools. Assume that the Redgate tools are already installed.
  3. Instructions on how to deploy the application.
  4. Instructions on how to add the application pool if it doesn’t exist.
  5. Instructions on how to set up the security on the databases in SQL Server.
  6. Instructions on how to verify that the application has been correctly deployed, including a link to the deployed application.
  7. Any other instructions required.
  8. An appropriate number of screen shots which would enhance the instructions. You must use your own screen shots. Refer to the following source on how to take proper screen shots: <http://www.hanselman.com/blog/TakingProperScreenshotsInWindowsForBlogsOrTutorials.aspx> .
  9. References to any sources used.

## Deployment of Database to a DB Server

This section involves how to create a deployment of your database environment in the development server using Redgate tools.

1. Install the Redgate tools using the SQL Toolbelt.exe in the S:\Computer Sciences\Projects\Development\Redgate folder.
2. Using the Redgate Schema Compare/Deploy tool in SQL Server Management Studio, deploy the database **schema** for your application to a new database named in CSDev. Save the script with the default name in the folder for the assignment. Do not save the project.
3. Using the Redgate Data Compare/Deploy tool, deploy the **data** for the tables in the your application database to the new database on the target database server. Save the script with the default name in the folder for the assignment. Do not save the project.

## Deployment of Web Application to an Application Server

This section involves deploying the application to the development server. You may need an application pool created has been created (by a prof or system administrator) which will be used to run the application on the web server and a corresponding login has been created on the database server.

I’ve put in some details for getting this on CSDEV, but you need to abstract so that this works on other target servers.

1. Make sure that you have the latest version of the application and ensure that the web.config file is using the individual database created in Part B and your individual PASS database.
2. Change all the connection strings in your config files use Windows integrated authentication instead of a username and password.
3. Ensure there are no hardcoded details that wouldn’t survive an environmental change. This includes people. For example, support emails and other contact information should not be hardcoded by be configuration file or database driven.
4. Build the application.
5. Deploy the application to a site in the **Projects** folder with your *uName*/*projectName* on the CSDev web server (<http://csdev.cegep-heritage.qc.ca>).
6. Login to the CSDev development web server using Remote Desktop Connection.
7. Start Internet Information Services Manager (IIS).
8. In IIS, change the application to use the default AppPool, which is the identity of the application pool under which the application will run on the web server, and make sure that the Login page is the first item in the list of default documents.
9. In SQL Server, change the security on your databases to assign the required rights for the default AppPool login.
10. Test the installed application to make sure it runs.
11. **Be sure to logoff or sign out of the server when you are done using the connection, but do NOT shutdown the server**.