**Assignment 4 – Going SOA via MSMQ**

**Assignment Due:** April 8th, by 5:00pm

**Requirements:**

This is a team assignment. For this assignment you will need to create a (or update your assignment 4) MVC 4.0 Web Application displaying k-state course offerings, which can be found here: <http://catalog.k-state.edu>. You should follow the MVC conventions throughout this assignment. A series of straightforward tutorials on this can be found here: <http://www.asp.net/mvc/tutorials/mvc-4/getting-started-with-aspnet-mvc4/intro-to-aspnet-mvc-4>. You will be using the MSMQ framework to implement a system bus, then refactoring your existing resources (authentication, courses/dependencies, and study plan visualization) to communicate over it. A description can be found at <http://msdn.microsoft.com/en-us/library/windows/desktop/ms711472%28v=vs.85%29.aspx> , and many tutorials are available on the Web, including <http://fransiscuss.com/2012/06/01/msmq-basic-tutorial/> .

1. **Prior Work** (10 points) – Your program should implement the full set of requirements from assignments 1-3.
2. **Distribute the Courses via MSMQ** (30 points) – You will be expanding your previous work on the degree programs to separate the database containing the courses and dependencies (with appropriate logic) and the flowchart view.
   1. *Courses and Dependencies –* Create a topic which allows your business logic to request courses and receive information containing the course and its dependencies.
   2. *Give your options –* Decide whether messages should be journaled, and whether they should have reliable delivery enabled.
3. **Plan of Study** (20 points) – Distribute the MVC resource (and supporting resources) used to represent each student’s customized plan of study. Follow the template in 2(a).
   1. *Updates* – Allow updates to a program of study to propagate back through the ESB to the database and visualizer.
   2. *Automatic redraw –* Users should be able to make a change to a program of study, and have any open windows automatically redraw.
4. **User Authentication and Authorization** (20 points) – Use the basic authentication system for your application, and add the service (via your ESB) for your modules to use.
5. **Tests** (10 points)– You should write unit tests for each of the model and controller classes you create.
6. **Writeup** (10 points) – You should write up a short document describing your architecture, topics used, and an analysis of how much bandwidth you expect to use with 100 simultaneous system users across the ESB.

Teams with CIS726 students are expected to spread their services connected by MSMQ across at least two VMs (more if desired). Teams with 526 students only are not required to distribute their services.

Your production version of the application should be hosted on a virtual machine(s) on your OpenStack. You will need to submit the name of your virtual machine running the application on K-State Online.