Advise System Requirements

The proposed advising system will:

* Adhere to the predetermined vision and scope
* Require executive sponsorship
* Operate within approved budget
* Run on premise on Linux-based operating systems
* Run on virtualized infrastructure
* Utilize an Oracle database as its back-end repository
* Utilize ASP.NET as its front-end user/application interface
* Allow users to log in locally initially (future versions to allow unified organizational credentials or SSO)
* Utilize configurable role-based security to provide proper access while restricting access to unauthorized individuals necessary
* Utilize a High-Availability solution by utilizing multiple application servers and hardware load balancing
* Allow data to be input manually or uploaded via comma-separated value (CSV) file (later versions to integrate with Student Information System (SIS) to reduce the management workload

Database structure will:

* Be a relational database design with tables for storing users, courses, academic history, program requirements while utilizing primary/foreign key relationships for referential integrity

User interface will provide the ability for:

* Users to log in
* Administrators to customize the system to meet the organization’s branding requirements
* Administrators to create roles for use within the role-based security framework
* Administrators to input or upload information (students, instructors, course data, academic history, program requirements)
* Users to see current course data as well as academic history
* Student users to review his/her academic history
* Student users to compare his/her academic history (along with current course load) to his/her program/degree requirements
* Student users to export the degree requirements and/or comparison reports
* Advisors to find students
* Advisors to find the program requirements for the program/degree in question
* Advisors to review a student’s academic history
* Advisors to compare a student’s academic history (along with current course load) to the student’s program/degree requirements
* Functional administrators (Registrars) to create, view, and edit program/degree requirements
* Functional administrators (Registrars) to create, view, and edit term definitions
* Functional administrators (Registrars) to view all students and their information
* All users to perform a “What-If” analysis to compare proposed course schedules against program/degree requirements
* Users to access help features and contact appropriate help personnel from within the software

Documentation will:

* Provide information regarding the database schema and database/application servers
* Provide information regarding all the aforementioned user interface abilities
* Be provided to users via a web site for convenient access and update capabilities

Testing procedure will:

* Train support staff
* Develop training specification and train using appropriate method

Software Release task will:

* Identify pilot group
* Train pilot group with appropriate method
* Obtain user feedback
* Release to whole organization

Post Implementation will:

* Gather feedback from organization
* Gather feedback from helpdesk
* Document lessons learned
* Document issues to maintenance team