**Final Project Guidelines - .NET III**Build an ASP.NET MVC web application with the following features:

***Solution:***

* \* n-tier architecture with an ASP.NET MVC presentation layer  
  OR  
  an ASP.NET MVC project that is all Entity Framework  
  OR  
  any combination of the two above.
* \* References between projects set properly

***Database***

* \* A SQL script and batch file for deploying your database   
  (unless it is all in Entity Framework)
* \* Sample data in the script
* \* Stored procedures for all data mutation operations and queries  
  (unless it is all in Entity Framework)

***Application:***

* \* Responsive front-end is preferred
* \* Support login and register featurs
* \* Screens for basic data operations CRUD

***Planning (not strictly required, but adds to your points):***

* \* A Statement of Work
* \* An ER Diagram for your database
* \* Data Dictionary
* \* UI Drawings
* \* List of User Stories

 Theme/Topic is up to you.

* Pick something fun if you like
* Useful projects are often best
* You may reuse your .NET II project if you like.

**Base Points (r*equired to avoid deductions; gets you to a B):***

* \* Multiple users and at least two roles **(Identity System preferred)**
* \* Reasonable division of classes
* Program builds and runs
* Results correct / database updates as expected
* Database script works
* Features work

***Quality Plusses (can take you to an A):***

* App: well-conceived and tested application
* \* Unit tests for your logic layer
* \* Code to interface, not implementation
* \* Dependency Inversion
* Data: good data types in app, DTOs, normalized tables
* Code: Well-written, well commented, good method names
* UI: UX focus, looks good, inputs constrained

***Quality Deductions (can cost you points):***

* Poor naming conventions or other poor coding style
* Copy/Paste code
* Unvalidated, unlabeled, or confusing inputs
* Unsafe code (try/catches missing)