**Final Project Requirements**

1. Create a new MVC application (suggestion: Personal Management).

Done

1. Allow users to register and login to the application.

Done

1. Include at least 3 entities (people, places, todos, events, hobbies, etc.)  
   Entities must be owned by a user of the application (a user cannot see another user's entities).  
   Create the actions and views to create, edit, display details and delete each of the entities.

Done – Books, Movies, Groups – most of the functionality involves the books entity

1. Encapsulate database access in a repository class that is accessed by the controllers via dependency injection.

Done

1. Create an application service class for one of the entities that encapsulates at least one function related to the entity  
   The service class must access the database via an injected instance of the repository class to accomplish it's task.  
   Call the service class function from the entity controller to performs the function.  
   Write at least one unit test for the service class that tests it's function based upon output from a mocked version of the injected data repository.  
   The mocked version of your the class can be created manually (a fake class that implements the same interface) or using a mocking framework (like FakeItEasy) as demonstrated in class.

Done

1. Create a separate project for a REST service that provides a function for one of the other entities that encapsulates at least one function related to the entity. + Injected EfDataRepository in service constructor  
   Create a "proxy" client class that calls the REST service.  
   Inject the proxy client class into the controller for the related entity.  
   Call the REST service from the entity controller to perform the function.

Done

1. Style your application with Bootstrap beyond the styles used by default in the views created by Visual Studio.

Done – navbar has dropdowns

1. Make use of some feature in jQuery (dialog, pop up message, etc.)

Done

1. Put your application on GitHub and send me a link to the repository prior to the deadline (Midnight, Friday Dec 9th).  
   I should be able to fork your GitHub repository, build the solution and run the application locally.  I will create pull requests on GitHub with comments.
2. Extra credit: Add at least one acceptance test using Selenium.
3. Extra credit: Set up a build project at www.appveyor.com that builds your application from a GitHub repo and runs the unit tests.  
   If you do this you'll need to add me as a user on your AppVeyor account.
4. Extra credit: Deploy your application to Azure (you can comment out the call to your REST service).  
   If you do this. send me a link to your application on Azure.

No extra credit attempted