

Lab 2 – ASP.NET MVC: Domain Model + Supporting Views and Controllers

CS295N, Web Development 1: ASP.NET

Group A: Community Information Site

In this lab project, you will create a domain model that supports the features planned for your web site. The first section below describes the features, and the second section summarizes what you need to do to create the domain model.

Site features

This is just for your information. There is nothing for you to do here.

Starting Point

In the previous lab, you created a skeleton of a web site with the six pages shown below: (main bullet points are top-level web pages, and sub-points are pages linked from the top-level pages.)

- Home – General information about the community and the purpose of the site
 - History – A brief history of the community (can be a static page)
 - Contact – a page with a form where users can send messages to the site administrators
- Info – Highlights of the community
 - Important locations and links (displayed in a table based on the model)
 - Significant people and links (displayed in a table based on the model)

Additional features

You will need to support these additional features that weren't described in the first lab assignment:

- Contact page: users will be able to send messages to each-other and not just to the Administrators.
- ~~Messages can have replies and replies can have replies~~

TODO

Domain Model

1. Create a UML diagram for a domain model that reflects the features described for this web site. You can create the UML class diagram using one of these methods:
 - a. On paper and take a picture of it
 - b. Use a UML diagramming software tool like:
 - i. UMLet (Free and easy to use), <https://www.umlet.com>
 - ii. Visio (available free to students through Microsoft Imagine)
 - iii. Star UML (my favorite, but takes a while to learn), <http://staruml.io/download>

2. Implement (write the code) for the domain model in C#

More on the next page

Views and Controllers

- Complete the form on the contact view so that users can create messages
- Add a view that lets users view messages: both sent and received
- ~~Complete the locations and links view. In the future this will be editable via a view with a form, but for now you can load the model with hard-coded data~~
- ~~Complete the people and links view—same as above with the hard-coded data~~

For each of the views above, add a new controller class, if needed and write the appropriate controller methods

Submission to Moodle

Beta Version

Upload the following to the Code Review Forum:

1. A zip file containing your web app's Visual Studio solution folder.
Or, a link to a repository containing your web app's source code. You can put the link on the same document with the report on your tutorial exercise.
2. A code review of your lab partner's work. (You do this after your lab partner submits item 1 above and you have reviewed it.)

Production Version

1. Items 1 above, but revised as needed.
2. The code review your lab partner did of your work with the second column ("Production") completed by you.