МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського “ХАІ”

Кафедра 603

Лабораторна робота № 1

з дисципліни «Реінжиніринг програмного забезпечення»

Використання шаблону проектування MVC для розробки WEB-додатків.

ХАІ.603.657ІПЗ.15О.05010302.116301.ІПЗ

Виконав студент: гр. 657ІПЗ Апанасенко Ю.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (№ групи) (П.І.Б)

(підпис, дата)

Перевірив: доц., Розсоха С.В.

(наукова ступінь, вчене звання, П.І.Б)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(бал, підпис, дата)

2015

Цель: изучение средств методов реализации WEB-приложений на основе шаблона проектирования MVC в основных фреймворках.

**Постановка задачи**

Разработать WEB-приложение “Заметки”. Для реализации приложения использовать шаблон проектирования MVC в любой из популярных платформ.

Приложение должно выполнять следующие функции:

1. Добавление заметок.
2. Удаление заметок.
3. Пометка заметки как выполненной.
4. Отображение заметок по критериям:
5. выполненные заметки
6. не выполненные заметки
7. все заметки

**Ход работы**

В качестве платформы для реализации приложения была выбрана .NET 4.5. Так же был выбран ASP NET MVC 4 в котором уже реализован MVC для разработки web-приложений. Для реализации view приложения использовались серверные страницы cshtml. В качестве тестового web-сервера использовался IIS. Разработка велась в Visual Studio 2012.

В ходе реализации приложения были разработаны такие классы:

* ToDoModel – модель заметки
* ToDoController - все запросы и первичная обработка ответов происходит через этот класс.

Для отображения формы была создана страница Index.cshtml. На рисунке 1.1 изображена UML-диаграмма разработанного приложения.



Рисунок 1 - UML-диаграмма приложения Заметки.

Листинги файлов приложения приведены ниже.

Листинг файла ToDoController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using laba1MVC.Models;

namespace laba1MVC.Controllers

{

public class ToDoController : Controller

{

//

// GET: /ToDo/

List<ToDoModel> ToDoList = new List<ToDoModel>();

[HttpGet]

// [OutputCacheAttribute(VaryByParam = "\*", Duration = 1, NoStore = true)]

public ActionResult Index()

{

if ((List<ToDoModel>)Session["ToDoList"] != null)

ToDoList = (List<ToDoModel>)Session["ToDoList"];

Session["ToDoList"] = ToDoList;

return View(ToDoList);

}

[HttpGet]

public ActionResult Save()

{

return View();

}

[HttpPost]

public ActionResult Save(String toDo)

{

ToDoList = (List<ToDoModel>)Session["ToDoList"];

ToDoModel newModel = new ToDoModel();

newModel.ToDo = toDo;

newModel.State = false;

ToDoList.Add(newModel);

Session["ToDoList"] = ToDoList;

return RedirectToAction("Index") ;

}

[HttpPost]

public ActionResult SetChecked(string toDo, bool value)

{

ToDoList = (List<ToDoModel>)Session["ToDoList"];

ToDoModel editModel = ToDoList.First(x => x.ToDo == toDo);

editModel.State = value;

Session["ToDoList"] = ToDoList;

//return Redirect(Request.UrlReferrer.ToString());

return RedirectToAction("Index", "ToDo");

}

[HttpGet]

public ActionResult Delete(string toDo)

{

ToDoList = (List<ToDoModel>)Session["ToDoList"];

ToDoModel model = ToDoList.Find(x => x.ToDo == toDo);

ToDoList.Remove(model);

Session["ToDoList"] = ToDoList;

return RedirectToAction("Index");

}

[HttpGet]

public ActionResult ClearCompleted()

{

ToDoList = (List<ToDoModel>)Session["ToDoList"];

List<ToDoModel> models = ToDoList.FindAll(x => x.State == true);

ToDoList.RemoveAll(x => x.State == true);

Session["ToDoList"] = ToDoList;

return RedirectToAction("Index");

}

[HttpGet]

public ActionResult Sort(string condition)

{

ToDoList = (List<ToDoModel>)Session["ToDoList"];

List<ToDoModel> result = new List<ToDoModel>();

switch (condition)

{

case "All":

result = ToDoList;

break;

case "Active":

result = ToDoList.FindAll(x => x.State == false);

break;

case "Completed":

result = ToDoList.FindAll(x => x.State == true);

break;

}

Session["ToDoList"] = ToDoList;

return View("Index", result);

}

}

}

Листинг файла ToDoModel.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace laba1MVC.Models

{

public class ToDoModel

{

public String ToDo { get; set; }

public bool State { get; set; }

}

}

Листинг файла RouteConfig.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Routing;

namespace laba1MVC

{

public class RouteConfig

{

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "ToDo", action = "Index", id = UrlParameter.Optional }

);

}

}

}

Листинг файла Index.cshtml

@model ICollection<laba1MVC.Models.ToDoModel>

<script>

function handleClick(cb) {

$.ajax({

url:"@Url.Action("SetChecked","ToDo")",

type: 'POST',

contentType: 'application/json;',

data: JSON.stringify({ toDo: cb.name, value: cb.checked}),

success: function (valid) {

if (valid) {

location.reload(true);

} else {

//show that id is not valid

}

}

});

}

</script>

<h2>Todos</h2>

<body>

@Model.Count(x=>x.State == false) of @Model.Count() remaining<br />

@foreach (laba1MVC.Models.ToDoModel item in Model)

{

@Html.CheckBox(item.ToDo, item.State, new {@onclick = "handleClick(this)" })

if (item.State == true)

{

@Html.Label(item.ToDo, new { @class = "done-true" })

}

else

{

@Html.Label(item.ToDo)

}

@Html.ActionLink("Delete", "Delete", "ToDo" ,new { toDo = item.ToDo }, null)

<br />

}

@using (Html.BeginForm("Save","ToDo",FormMethod.Post))

{

@Html.TextBox("ToDo");

<input type="submit" value="Add" />

}

<br />

@Html.ActionLink("Clear complited", "ClearCompleted", "ToDo")

@Html.ActionLink("All", "Sort", "ToDo", new {condition = "All" }, null)

@Html.ActionLink("Active", "Sort", "ToDo", new {condition = "Active" }, null)

@Html.ActionLink("Completed", "Sort", "ToDo", new {condition = "Completed" }, null)

</body>

Результат работы приложения представлен на рисунке 2.

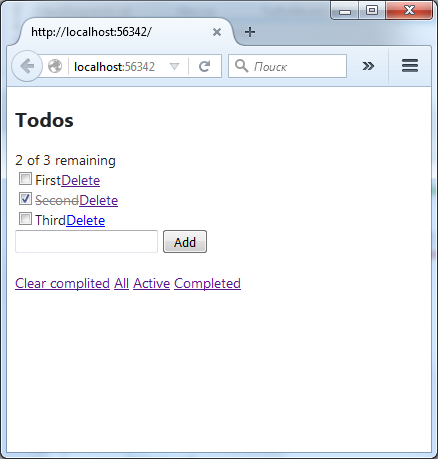


Рисунок 2 - Результат работы приложения.

Вывод: в ходе выполнения лабораторной работы с использованием ASP .NET MVC 4 было разработано web-приложение заметки. Разработка приложения позволила получить навыки использования ASP .NET MVC фреймворка для разработки подобного рода web-приложений.