UDC 621.3

I.V. Ponomarev, A.R. Chuhalo

FEATURES OF THE DEVELOPMENT SITE WITH ASP.NET MVC FRAMEWORK

Annotation. A sequence of actions to create a website using MVC technology. At the heart of the developed product architecture template used layered architecture that simplifies programming job.

Tags: ASP.NET, FRAMEWORK, multilayer landscaping, the MVC, the model, the view, the controller.

Introduction. As an alternative to Web Forms technology, ASP.NET MVC uses a different approach to structuring of web applications. This means that do not have to deal with the ASPX-pages and controls, or the ViewState postback, as well as the life cycle of complex events. Instead determined controllers, actions and views.

Formulation of the problem. It is necessary to consider the characteristics of the development of the site using ASP.NET MVC FRAMEWORK, which allows to solve the bone application design problem and concentrate on developing a specific program layer.

Main part.

Description ASP.NET MVC FRAMEWORK.

MVC - it is a fundamental pattern that has found applications in many technologies, given the development of new technologies and every day makes life easier for developers. [1] Consider platform components [2].

Model.

Under the model refers to a part of the program, which contains a functional business logic of the application. The model should be completely independent from the rest of the product. Model layer should not know anything about the design elements, and how it will be displayed. Thus, the result is achieved, which allows to change the view of the data, and how they are displayed, without touching the model itself. The model has the following features:

- model - is the application business logic;

- model has the knowledge about itself and does not know about the views and controllers;
- For some projects, the model - it's just a data layer (DAO, database, XML-file);
- model for other projects - a database manager, a set of objects or just the application logic.

Performance.

In the presentation responsibility to display the data received from the model. However, the view can’t directly affect the model. You can say that the representation has access "read-only" data. Representation has the following features:

- in the representation of the data mapping is implemented, which can be obtained from the model in any way;
- representation, in some cases, may have code that implements some business logic.

Examples of presentation: HTML-page, WPF Form, Windows Form.

Controller.

The controller converts the user's actions (in this context, the user - not necessarily people) in the input parameters for the model, and transmits to the model management:

- loads the environment variables (POST / GET variables, command line parameters, URL parameters, etc...);
- performs primary processing environment variables (variables test types, availability, default values, etc...);
- implement mechanisms to monitor extraordinary events;
- implementing mechanisms for logging (no authentication, and Logging).

The sequence of development of the site.

First we need to develop a specification for the user and the administrative part. They should include a description of the main elements and the View-relevant functionality that can be configured. The basis of the architecture of the product using a template layered architecture (View Model). In this software solution developed consists of five layers, each of which has no rigid links with other.

1. The layer server, which contains a user interface and administration forms, their controllers.

2. The repository, which contains the interface with a set of common CRUD operations and class with their implementation.

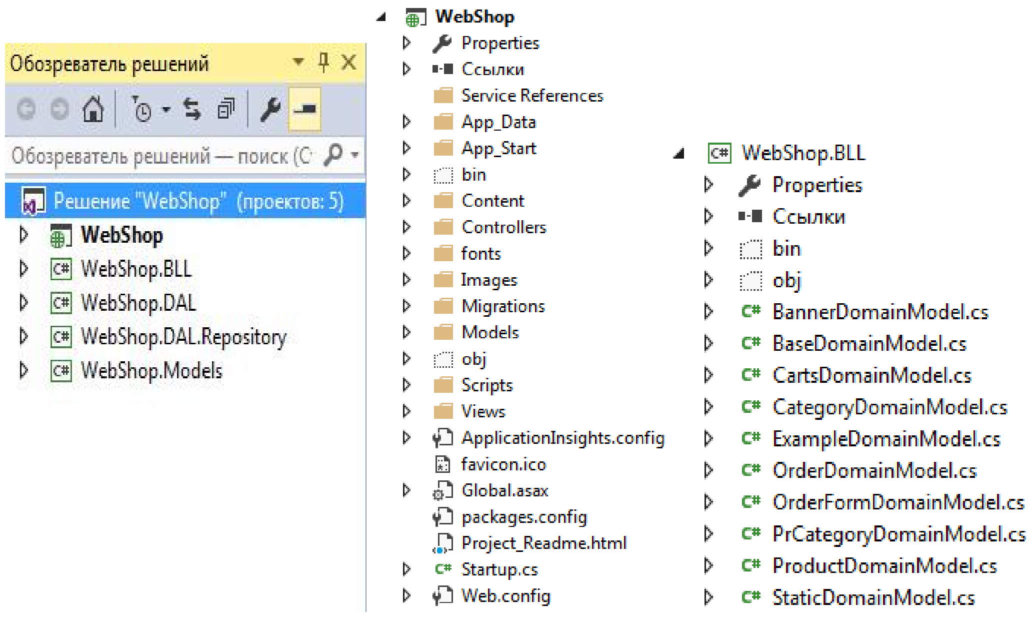
3. DAL-layer in which the establishment of a database by Entity Framework.

4. A layer of business logic that implements the basic functionality for a layer model.

5. Layer models, which are an intermediate layer between the database and business logic.

Example of implementation.

As a platform for the implementation of the selected .Net Framework - a powerful, modern, easy-to-use application development tool that provides extensive standard library.



```
Обозреватель решений
Обозреватель решений — поиск (C
Решение "WebShop" (проектов: 5)
  WebShop
  WebShop.BLL
  WebShop.DAL
  WebShop.DAL.Repository
  WebShop.Models
```

```
WebShop
  Properties
  Ссылки
  Service References
  App_Data
  App_Start
  bin
  Content
  Controllers
  fonts
  Images
  Migrations
  Models
  obj
  Scripts
  Views
  ApplicationInsights.config
  favicon.ico
  Global.asax
  packages.config
  Project_Readme.html
  Startup.cs
  Web.config
```

```
WebShop.BLL
  Properties
  Ссылки
  bin
  obj
  BannerDomainModel.cs
  BaseDomainModel.cs
  CartsDomainModel.cs
  CategoryDomainModel.cs
  ExampleDomainModel.cs
  OrderDomainModel.cs
  OrderFormDomainModel.cs
  PrCategoryDomainModel.cs
  ProductDomainModel.cs
  StaticDomainModel.cs
```

Picture 1. Figure solutions and core layers of the site

As an example of designed site selling electronics «WebShop» (Picture 1). Consider «WebShop» static page.

The tables in the database: public class ShopEntities: DbContext

```
{
public DbSet<Client> Clients { get; set; }
public DbSet<ClState> ClStates { get; set; }
public DbSet<Image> Images { get; set; }
public DbSet<Order> Orders { get; set; }
public DbSet<OrderState> OrderStates { get; set; }
public DbSet<Product> Products { get; set; }
public DbSet<ProductCategory> ProductCategories { get; set; }
public DbSet<ProductOrder> ProductOrders { get; set; }
public DbSet<PrCategory> PrCategories { get; set; }
public DbSet<Static> Statics { get; set; }
public DbSet<Cart> Carts { get; set; }
public DbSet<OrderCart> OrderCarts { get; set; }
public DbSet<Comment> Comments { get; set; }
}
```

Model of the table:

```
public class StaticModel : ViewModelBase<int>
  {
        public Int64 Static_id { get; set; }
        public string About_shop { get; set; }
        public string Sales { get; set; }
        public string Delivery { get; set; }
  }
```

Business logic:

```
public class StaticDomainModel : BaseDomainModel
  {
     public void AddStatic()
     {
        using (var repository = new BaseRepository<Static, int>())
        {
           //repository.Insert(new Static { Title = "Test1" });
        }
     }
     public IEnumerable<StaticModel> GetAll()
     {
        using (var repository = new BaseRepository<Static, int>())
        {
           var list = repository.Query().Select(x => new StaticModel
        { Static_id = x.Static_id, About_shop = x.About_shop,
        Delivery = x.Delivery, Sales = x.Sales }).ToList();
           return list;
        }
     }
     public Static GetById(int id)
     {
        using (var repository = new BaseRepository<Static, int>())
        {
           return repository.Get(id);
        }
     }
```

An example of "Specials" page:

```
@model IEnumerable<WebShop.Models.StaticModel>
@{
   ViewBag.Title = "Акции";
}
<h2>@ViewBag.Title</h2>
<p>
   @foreach (var item in Model)
         {
      <tr>
         @Html.DisplayFor(model => item.Sales)
      </tr>
   }</p>
```

Conclusions. The paper considers the basis of the software solution creating a website using MVC technology. Determine the product architecture, developed a common scheme of interaction between the layers of the program. ASP.NET MVC platform allows to create programs with a weakly bound layers. This makes it possible to program each layer independently, thus facilitating the work of developers, allowing to concentrate on a specific task.

LITERATURE

1. Martin Fowler. Patterns of Enterprise Application Architecture. - Addison Wesley, 2002-560 p.

2. Adam Freeman. Pro ASP.NET MVC 5.- Apress, 2013-736 p.

УДК 621.3

Пономарев И.В., Чухало А.Р. Особенности разработки сайта с помощью ASP.NET MVC FRAMEWORK. Описываются основные компоненты технологии MVC. Предложены принципы построения архитектуры сайтов и указываются основные преимущества такого подхода.

Библ. 2., илл.1

УДК 621.3

Пономарьов І.В., Чухало А.Р. Особливості розробки сайту за допомогою ASP.NET MVC FRAMEWORK. Описуються основні компоненти технології MVC. Запропоновані принципи створення архітектури сайтів й зазначають основні переваги цього підходу.

Бібл. 2., іл.1

UDC 621.3

Ponomarev I.V., Chuhalo A.R. Website development using ASP.NET MVC FRAMEWORK. It describes the basic components of the MVC technology. The principles of construction of architectural sites and identifies the main advantages of this approach.

Bibl. 2., pic.1