**UDC 004.42**

# WEB PORTAL OF AMATEUR HANDBALL IN THE REPUBLIC OF BELARUS

*ULADZIMIR TRAPEZNIKOU, ARKADII OSKIN*
**Polotsk State University, Belarus**

*The article presents an analysis of the technologies that are used in the implementation of server-side Web applications. ADO.NET Entity Framework and CodeFirst approach are considered.*

The subject of the project is creating a web application containing information on amateur handball in Belarus, realized on the .net Framework platform.

Web applications or sometimes also web systems are different software products, access to which is provided via web interface. A web application is a client-server application in which the client is a browser, and the server is a Web server. The logic of a web application is distributed between the server and the client, data storage is carried out, preferentially, on the server, information exchange happens on a network. One of advantages of such approach is the fact that clients don't depend on a specific operating system of the user therefore web applications are cross-platform services.

Essential advantage of creating web applications to support fixed functions of a browser is that functions are performed irrespective of the client’s operating system. Instead of writing different versions for Microsoft Windows, Mac OS X, GNU/Linux and other operating systems, the application is created once for randomly selected platform.

A web application consists of client and server parts, thereby realizing client-server technology. A client part realizes the user interface, creates server requests and processes responses from it. A server part receives a request from the client, executes computation, after that creates the web page and sends it to the client on a network with the use of HTTP protocol.

Relevance of creating a web application is justified also by the fact that with development of portable computers and mobile devices, the client part of a web application can be used not only on desktop computers and laptops, but also on the mentioned mobile devices. It leads to the fact that the circle of people who will be able to use the developed product is much wider than if the web application was implemented only as a Desktop application.

The main objective of implementing a web application is to give the chance to users to be aware of all amateur competitions which are taking place in our country and also to view information on amateur commands and players.

Developed software should have the following functionality:

1 registered users can register sporting events;

2 all users can view news;

3 all users can view information on commands;

4 all users can view information on players;

5 all users can view information on actions;

6 registered users can change personal data in a private office;

7 registration of users with different roles;

8 search in a basis of actions;

9 editing all data domain by the user administrator;

10 authorized users can leave comments to results of actions;

11 all users can view photo and video reports;

12 the registered users can recover the password by means of email, in case of old password loss.

**Means for solving the problem**. Consider the basic tools and technologies of server-side Web applications, "the Republic of Belarus Web portal amateur handball."

The paper deals with ASP.NET MVC Framework - a framework for building web applications that implement the Model-view-controller pattern. This framework is added to Microsoft's ASP.NET. ASP.NET MVC platform is based on the interaction of three components: controllers, models and views. The controller receives the request, processes the user input, interacts with the model and the view, and returns the user to the result of the query.

The model layer describes the logic of data organization in the application. Submission receives data from the controller and generates UI elements for displaying information. When processing requests, ASP.NET MVC framework is based on the routing system that maps all incoming requests to specific routes in the system that indicates which controller and method should handle the request. A built default route involves a three-tier structure: a controller / action / parameter [1, 2].

In Ninject the IoC-container will be used. Actually IoC (Inversion of Control (Inversion of Control)) is an abstract concept, a set of guidelines for writing a weakly bound code, the essence of which is that each component of the system should be as isolated from others, instead of relying in their work on the details of specific implementation of other components. Dependency Injection (dependency injection) is one of the realizations of this principle and IoC-container is a framework that will allow you to simplify and automate the coding with the use of this approach as much as possible [1, 2].

When implementing the application architecture, "onion" architecture will be used. Onion-architecture is the separation of applications to the levels. And there is one independent layer that is in the center of the architecture. This level depends on the second level, the second level - on the third one, and so on. Thus, it turns out that the second (dependent) level is superimposed around the first (independent) level. Around the second level the third one accumulates, which may also depend on the first level. Figuratively, it can be expressed in the form of a bow, which also has a core around which all other layers are laminated up to the hull.

The number of levels may vary, but the center is always the domain model (Domain Model), that is, those classes of models that are used in the application objects stored in the database. The first level around the domain model forms interfaces that control the operation of the domain model. Usually they are repository interfaces through which we interact with the database.

The outer layer represents such components which are often changed. Typically, the outer layer forms user interface tests, some helper classes application infrastructure. This level also includes concrete implementations of interfaces, announced at lower levels. For example, the implementation of the repository interface, which is declared on the level Domain Services. In general, all the internal levels which can be combined in Application Core define only interfaces and the concrete realization of these interfaces are located on the outer level.

It is also worth noting that all the external storage, such as databases, files, external Web services, from which we can obtain data - all of which are external to the architecture, [1, 2].

ADO.NET Entity Framework and CodeFirst approach will be used for working with the database framework. ADO.NET Entity Framework (EF), an object-oriented data access technology, is object-relational mapping (ORM) solution for .NET Framework from Microsoft. It provides the ability to interact with objects, both through a LINQ LINQ to Entities, and using Entity SQL. To facilitate the construction of web-solutions, ADO.NET Data Services (Astoria) and a bunch of Windows Communication Foundation and Windows Presentation Foundation are used, which allows to build a multi-tier applications by implementing one of the MVC design patterns, MVP or MVVM [1,2].

When designing applications with CodeFirst approach, you first create the data model classes not paying any attention to the Entity Framework. Next, you needed to work with the database, you use a variety of tools, which project out of the database structure to create a class model. Then you can return to this model in the code, for example, to change it. These changes can then be reflected in the database with the help of the same tools.

**Conclusion**. The use of MVC model has allowed a clear distinction between the functional server side without mixing its components resulting in easy understanding and the following code change.

ADO.NET Entity Framework makes it possible to create a database containing all the information and functionality needed for web-based applications without understanding the subtleties of SQL comfortable work. With CodeFirst approach it is possible to create a database from scratch only on the basis of written class models.

Using an IoC container Ninject made web application flexible to changes. We just need to implement the interface fit our way to the top layer of the application without going to the lower layers.

## REFERENCES

1 Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/. – Дата доступа: 29.01.2017.

2 Metanit – сайт о программировании [Электронный ресурс]. – Режим доступа: http://metanit.com. – Дата доступа: 29.01.2017.

3 Habrahabr – многофункциональный сайт. [Электронный ресурс]. – Режим доступа: https://habrahabr.ru. – Дата доступа: 29.01.2017.

4 Сандерсон С., Фримен А. ASP.NET MVC3 Framework с примерами на C# для профессионалов (3-е изд., 2012) – 662 с.

5 Стиллмен Э., Грин Дж. Изучаем C#. – 3-е изд. – СПб. : Питер, 2014. – 816 с.: ил. – (Серия «HeadFirstO'Reilly»).

6 Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. – 6-е изд. : пер. с англ. – М. : ООО «И.Д. Вильямс», 2013. – 1312 с. : ил. – парал. тит. англ.