Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет непрерывного и дистанционного обучения

Кафедра информатики

**Индивидуальная практическая работа №1**

по дисциплине

"ИЗБРАННЫЕ ГЛАВЫ ИНФОРМАТИКИ"

Вариант 10

Выполнил студент гр. 493551 Авхимович Алексей Валерьевич

E-mail: [droon2112@gmail.com](mailto:droon2112@gmail.com)

Минск 2018

Содержание

[Введение 3](#_Toc476990950)

[**1** ASP.NET 4](#_Toc476990951)

[**1.1** Общая архитектура системы 4](#_Toc476990952)

[**1.2** Компоненты бизнес-логики и слоя доступа к данным 4](#_Toc476990953)

[**1.3** Общая схема сайта 6](#_Toc476990954)

[**1.4** Макет сайта 7](#_Toc476990955)

[**1.5** Разработка разметки и программного кода 8](#_Toc476990956)

[**1.6** Демонстрация работы 12](#_Toc476990957)

[**2** ASP.NET MVC 17](#_Toc476990958)

[**2.1** Постановка задачи 17](#_Toc476990959)

[**2.2** Разработка программного кода 17](#_Toc476990960)

[**2.3** Демонстрация работы 18](#_Toc476990961)

Введение

Цель работы: создание сайта с использованием ASP.NET.

Для проекта необходимо продумать и реализовать:

* общую архитектуру;
* компоненты бизнес-логики и слоя доступа к данным;
* общую схему сайта;
* макет сайта.

Необходимо создать набор ASP.NET страниц сайта и связать компоненты страниц со слоем бизнес-логики веб-проекта. При создании страниц обязательным является использование эталонных страниц, тем, проверочных элементов управления, табличных элементов управления. Страницы должны демонстрировать возможности сохранения состояния и кэширования.

На сайте должны быть реализованы следующие элементы:

* выделены страницы, доступные на сайте пользователям определённой роли;
* реализованы страницы администратора для управления пользователями;
* предусмотрена возможность регистрации пользователей.

Постановка задачи: вариант 10 – Форум.

Необходимо реализовать следующие функции:

* стандартные операции, присущие форуму – добавление тем, сообщений;
* модерирование записей;
* работа с пользователями форума.

# ASP.NET

## Общая архитектура системы

Общая архитектура системы приведена на рисунке 1.1.



Рисунок . – Общая архитектура системы

## Компоненты бизнес-логики и слоя доступа к данным

Бизнес-логика реализуется с помощью обработчиков, прикрепляемых к веб-формам.

Роли и функции пользователей приведены в таблице 1.1.

В системе выделены следующие роли пользователей:

* администратор;
* модератор;
* авторизованный пользователь;
* забаненный пользователь
* гость (незарегистрированный пользователь).

Таблица . – Распределение функций по ролям

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функция | Админи­стратор | Моде­ратор | Пользо­ватель | Забанен­ный | Гость |
| Установка роли пользователя | + | + | – | – | – |
| Запрет активности пользователя | + | + | – | – | – |
| Модерирование сообщений | + | + | – | – | – |
| Создание разделов | + | – | – | – | – |
| Создание тем | + | + | + | – | – |
| Создание сообщений | + | + | + | – | – |
| Редактирование сообщений | + | + | + | – | – |
| Просмотр сообщений | + | + | + | + | + |
| Авторизация, изменение пароля | + | + | + | – | – |
| Регистрация | – | – | – | + | + |

Структура базы данных приведена на рисунке 1.2.

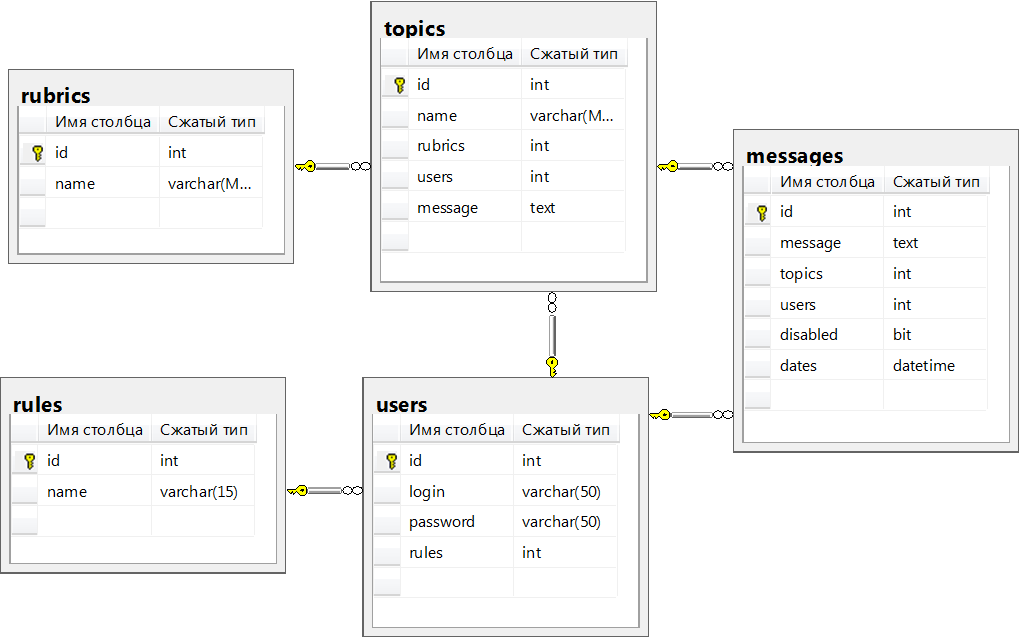


Рисунок . – Схема базы данных

Доступ к данным реализован с использованием DataSet. Схема DataSet приведена на рисунке 1.3.

Для управления авторизацией кроме классов DataSet дополнительно используем классы User и LoginHelper.

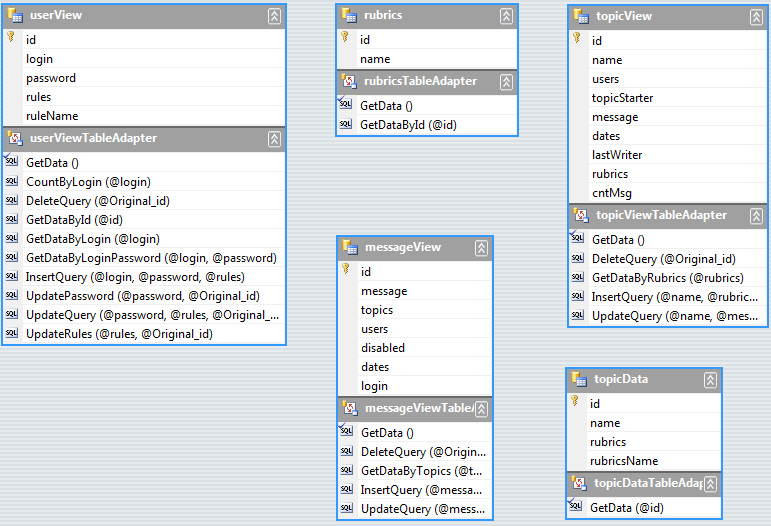


Рисунок . – Схема DataSet

## Общая схема сайта

Сайт форума включает следующие основные страницы:

* Default.aspx – главная страница, включающая список разделов; для администратора доступны редактирование и добавление разделов; обеспечивает переход к странице со списком тем выбранного раздела;
* Topics.aspx – страница со списком тем раздела; для администратора, модератора и пользователя доступно редактирование и добавление тем; обеспечивает переход к странице со списком сообщений темы;
* Messages.aspx – страница со списком сообщений темы; для администратора и модератора доступно редактирование и скрытие сообщений; для пользователя доступно редактирование своих сообщений.

Дополнительные страницы обеспечивают механизмы авторизации и управление пользователями:

* Auth.aspx – страница с формой авторизации; позволяет зарегистрированному пользователю авторизоваться на сайте; доступна неавторизованному пользователю;
* Register.aspx – страница с формой регистрации; позволяет пользователю создать аккаунт; доступна неавторизованному пользователю;
* Users.aspx – страница со списком пользователей; доступна администратору; обеспечивает переход к странице профиля пользователя;
* Profile.aspx – страница профиля пользователя; позволяет администратору и модератору изменить роль пользователя; позволяет зарегистрированному пользователю изменить пароль; дополнительно доступ к странице профиля обеспечивается со страниц со списком тем и списком сообщений через гиперссылку на логине пользователя.

Доступ к дополнительным страницам и главной странице сайта обеспечивается из главного меню. Меню доступно на всех страницах сайта.

Структура страниц сайта приведена на рисунке 1.4.

Рисунок . – Структура страниц сайта

## Макет сайта

Макет сайта приведен на рисунке 1.5.

Страница сайта включает:

* главное меню;
* панель пользователя;
* основное содержание;
* подвал.

Главное меню включает следующие пункты:

* наименование сайта – ссылка на главную страницу;
* пользователи – ссылка на страницу списка пользователей;
* регистрация – ссылка на страницу с формой регистрации;
* вход – ссылка на страницу авторизацию;
* выход – ссылка на процедуру обработки выхода с сайта.

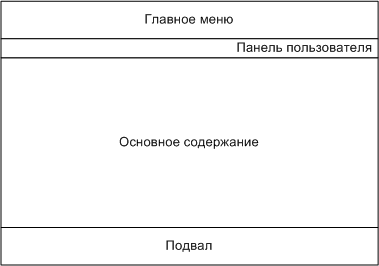


Рисунок . – Макет сайта

## Разработка разметки и программного кода

Дизайн и разметка сайта разработана с использованием фреймворка Bootstrap.

Сайт предусматривает единственную тему Fixed, применяемую в целях стандартизации оформления элементов.

Сайт предусматривает одну мастер-страницу Template.Master.

Главное меню формируется с помощью Bootstrap.

Управление скрытием и отображением элементов осуществляется через элемент управления LoginView.

Пункт меню «Пользователи» доступен только администратору:

<asp:LoginView ID="menuLeftLoginView" runat="server">

<RoleGroups>

<asp:RoleGroup Roles="Admin">

<ContentTemplate>

<li><asp:HyperLink ID="menuUsers" runat="server" NavigateUrl="~/Users.aspx">Пользователи</asp:HyperLink></li>

</ContentTemplate>

</asp:RoleGroup>

</RoleGroups>

</asp:LoginView>

Пункты меню «Регистрация» и «Вход» доступны неавторизованному пользователю, пункт меню «Выход» – авторизованному пользователю:

<asp:LoginView ID="menuRightLoginView" runat="server">

<AnonymousTemplate>

<li><asp:HyperLink ID="menuRegister" runat="server" NavigateUrl="~/Register.aspx">Регистрация</asp:HyperLink></li>

<li><asp:HyperLink ID="menuAuth" runat="server" NavigateUrl="~/Auth.aspx">Вход</asp:HyperLink></li>

</AnonymousTemplate>

<LoggedInTemplate>

<li><asp:LinkButton ID="menuLogout" runat="server" OnClick="menuLogout\_Click">Выход</asp:LinkButton></li>

</LoggedInTemplate>

</asp:LoginView>

Панель пользователя включает логин авторизованного пользователя и гиперссылку на его профиль. Содержание формируется динамически в коде при загрузке страницы:

// Получаем логин авторизованного пользователя

string login = HttpContext.Current.User.Identity.Name;

User user = new User(login);

// Если пользователь авторизован

if (Context.User.Identity.IsAuthenticated)

{

// Помещаем данные пользователя в соответствующие поля

// и показываем панель пользователя

profileHyperLink.Text = user.login;

profileHyperLink.NavigateUrl = string.Format("~/Profile.aspx?UserID={0}", user.id);

userNamePanel.Visible = true;

}

// Иначе - скрываем панель данных пользователя

else

userNamePanel.Visible = false;

В формах, в поля которых вводятся пользовательские данные (формы регистрации, авторизации и изменения пароля, формы добавления и изменения разделов, тем, сообщений), применяется валидация данных с использованием следующих элементов управления:

* RequiredFieldValidator – валидация на пустое значение;
* CustomValidator – настраиваемая валидация;
* RegularExpressionValidator – валидация с помощью регулярного выражения;
* CompareValidator – валидация на совпадение значений.

Так к полю Login формы регистрации применяются три валидатора – проверка на пустое значение; настраиваемая валидация на проверку присутствия логина в БД; валидация с помощью регулярного выражения на допустимость используемых символов (2-20 букв латинского алфавита и цифр, начинается с буквы):

<asp:TextBox runat="server" ID="Login" CssClass="form-control" />

<span class="help-block">

<asp:RequiredFieldValidator runat="server" ID="reqLogin" ControlToValidate="Login" Display="Dynamic">Логин не может быть пустым</asp:RequiredFieldValidator>

<asp:CustomValidator runat="server" ID="cstmLogin" ControlToValidate="Login" onServerValidate="cstmLogin\_ServerValidate" Display="Dynamic" EnableClientScript="False">Такой логин уже существует</asp:CustomValidator>

<asp:RegularExpressionValidator runat="server" ID="rglrLogin" ControlToValidate="Login" ValidationExpression="^[a-zA-Z][a-zA-Z0-9-\_\.]{1,20}$">Логин может включать 2-20 букв и цифр, должен начинаться с буквы</asp:RegularExpressionValidator>

</span>

Для вывода информации из базы данных на страницах со списками разделов, тем, сообщений и пользователей применяется элемент управления ListView, использующий в зависимости от страницы следующие шаблоны:

* LayoutTemplate – общая обертка элемента управления;
* ItemTemplate – шаблон элемента списка;
* InsertItemTemplate – шаблон вставки данных;
* EditItemTemplate – шаблон редактирования данных.

Допустимые операции с данными страниц приведены в таблице 1.1.

Таблица . – Операции с данными страниц по ролям

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Страница | Просмотр | Добавление | Изменение | Удаление |
| Default.aspx | Все | Admin | Admin | – |
| Topics.aspx | Все | Admin, Moder, User | Admin, Moder | Admin, Moder |
| Messages.aspx | Все | Admin, Moder, User | Admin, Moder, User (только свои, не скрытые) | Admin, Moder, User (только свои, не скрытые) |
| Users.aspx | Admin | – | – | – |

Доступ к данным по ролям на этих страницах ограничивается в коде страницы через атрибут Visible, поскольку использование элемента управления LoginView ограничивает доступ к данным:

<div runat="server" visible='<%# AllowEdit(Eval("login"), Eval("disabled")) %>'>

<asp:LinkButton ID="EditButton" runat="server" CommandName="Edit" CssClass="btn btn-default btn-sm"><span class="glyphicon glyphicon-pencil"></span></asp:LinkButton>

<asp:LinkButton ID="DeleteButton" runat="server" CommandName="Delete" CssClass="btn btn-default btn-sm"><span class="glyphicon glyphicon-remove"></span></asp:LinkButton>

</div>

Функция проверки прав доступа задается в коде страницы:

// Проверка права на редактирование сообщения

protected bool AllowEdit(object login, object disabled)

{

// Редактировать могут те, кто может скрыть

if (AllowDisabled())

return true;

// Редактировать может пользователь, если это его сообщение и оно не скрыто модератором

else if (HttpContext.Current.User.IsInRole("User") && HttpContext.Current.User.Identity.Name == login.ToString() && disabled.ToString() == "False")

return true;

else

return false;

}

Для доступа к данным используется элемент управления ObjectDataSource, который настраивается на необходимый объект DataSet:

<asp:ObjectDataSource ID="dsMessages" runat="server" DeleteMethod="DeleteQuery"

InsertMethod="InsertQuery" OldValuesParameterFormatString="original\_{0}"

SelectMethod="GetDataByTopics"

TypeName="ForumSimple.ForumDataSetTableAdapters.messageViewTableAdapter"

UpdateMethod="UpdateQuery">

<DeleteParameters>

<asp:Parameter Name="Original\_id" Type="Int32" />

</DeleteParameters>

<InsertParameters>

<asp:Parameter Name="message" Type="String" />

<asp:ControlParameter ControlID="hiddenTopicID" DefaultValue="0"

Name="topics" PropertyName="Text" Type="Int32" />

<asp:ControlParameter ControlID="hiddenUserID" DefaultValue="0"

Name="users" PropertyName="Text" Type="Int32" />

</InsertParameters>

<SelectParameters>

<asp:ControlParameter ControlID="hiddenTopicID" DefaultValue="0" Name="topics"

PropertyName="Text" Type="Int32" />

</SelectParameters>

<UpdateParameters>

<asp:Parameter Name="message" Type="String" />

<asp:Parameter Name="disabled" Type="Boolean" />

<asp:Parameter Name="Original\_id" Type="Int32" />

</UpdateParameters>

</asp:ObjectDataSource>

На сайте используется собственная система авторизации. При прохождении процедуры авторизации в случае, когда логин и пароль пользователя найдены в базе данных, создается билет авторизации:

FormsAuthenticationTicket ticket = new FormsAuthenticationTicket(1, user.login, DateTime.Now, DateTime.Now.AddDays(7), true, user.role);

string strEncrypted = FormsAuthentication.Encrypt(ticket);

response.Cookies.Add(new HttpCookie("\_\_AUTH", strEncrypted));

Данные билета используются для установки роли пользователя:

HttpCookie authCookie = Request.Cookies[FormsAuthentication.FormsCookieName];

if (authCookie != null)

{

FormsIdentity identity = (FormsIdentity)appl.User.Identity;

FormsAuthenticationTicket authTicket = FormsAuthentication.Decrypt(authCookie.Value);

string roles = authTicket.UserData;

appl.Context.User = new GenericPrincipal(identity, new string[] {roles});

}

## Демонстрация работы

На рисунке 1.6 приведена главная страница со списком разделов для неавторизованного пользователя.

Форма регистрации с результатом валидации приведена на рисунке 1.7.

Форма авторизации приведена на рисунке 1.8.

Главная страница для администратора в режиме редактирования данных приведена на рисунке 1.9.

Страница списка тем раздела для авторизованного пользователя приведена на рисунке 1.10.

Страница списка сообщений приведена на рисунке 1.11.

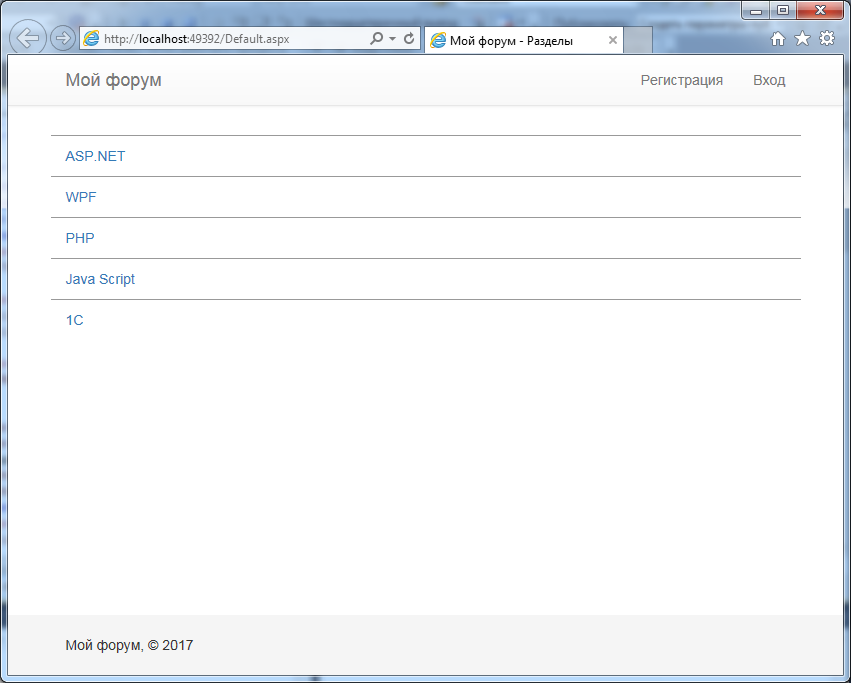


Рисунок . – Главная страница для неавторизованного пользователя

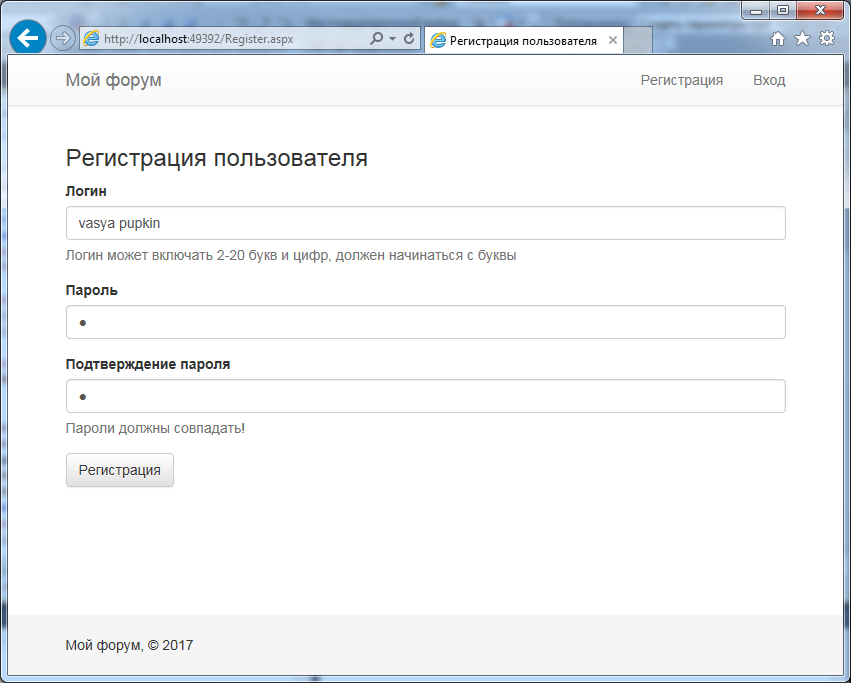


Рисунок . – Форма регистрации

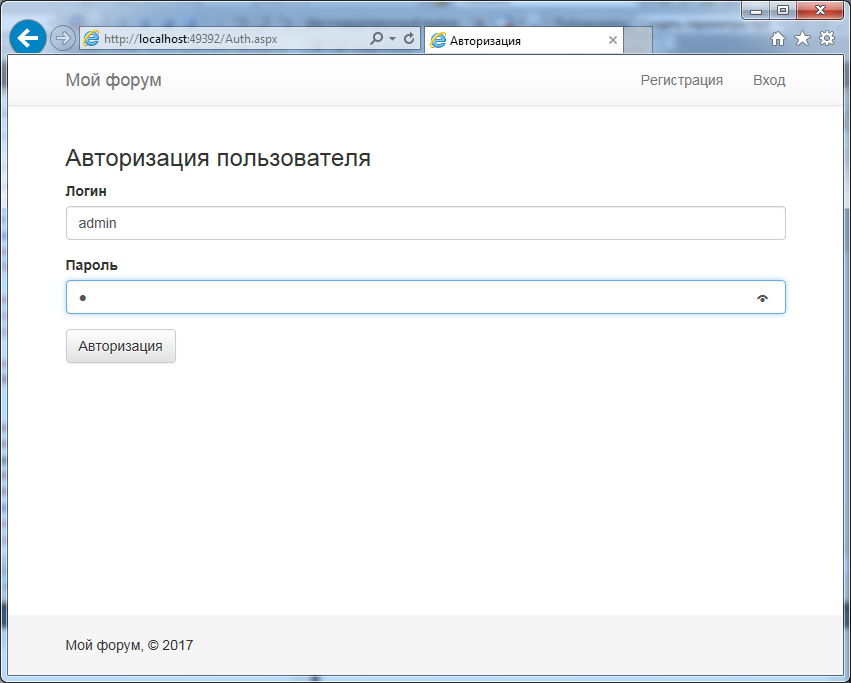


Рисунок . – Форма авторизации

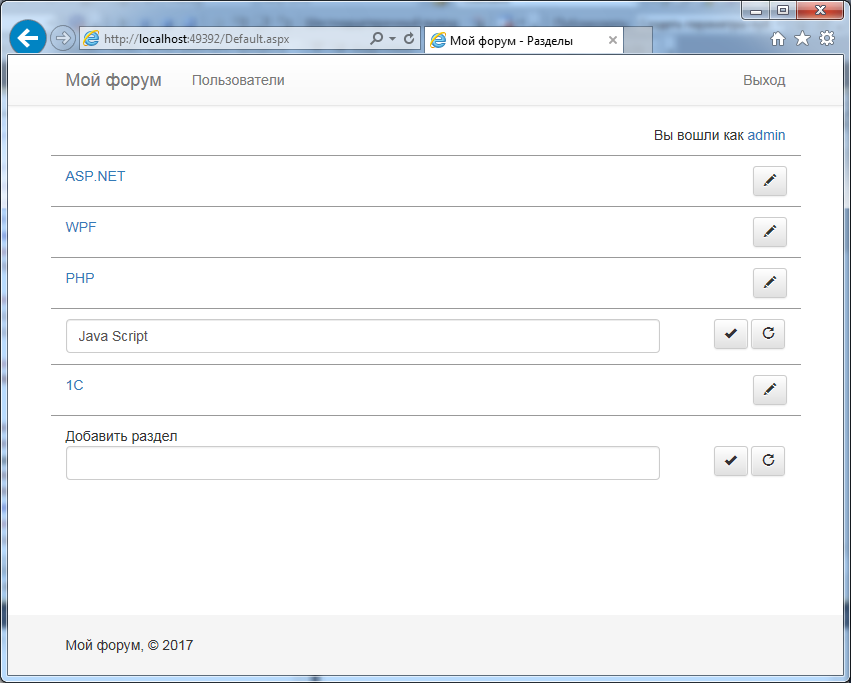


Рисунок . – Главная страница в режиме редактирования

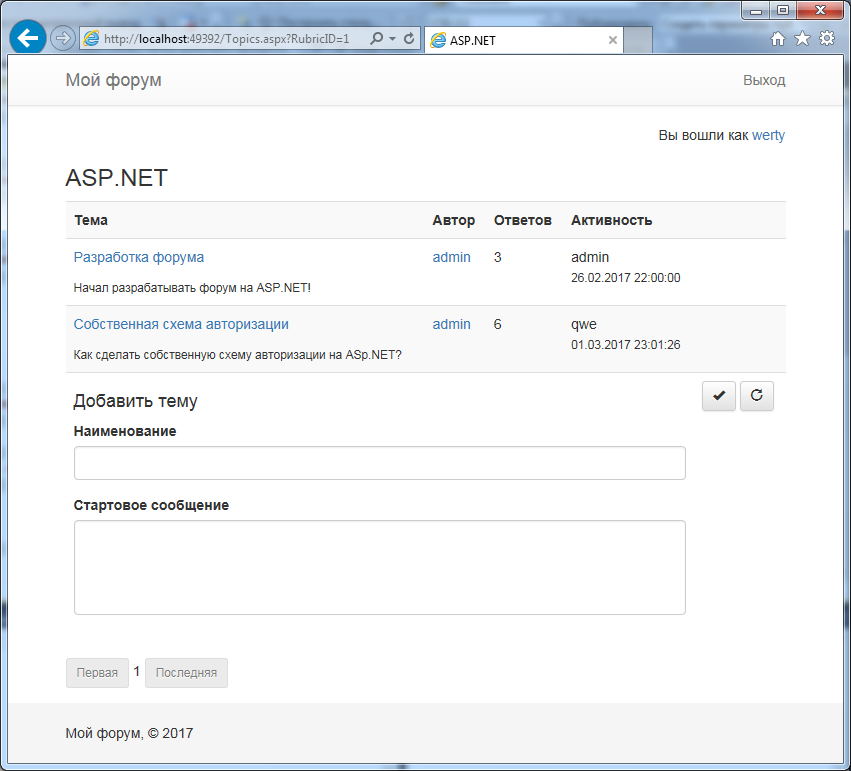


Рисунок . – Страница списка тем

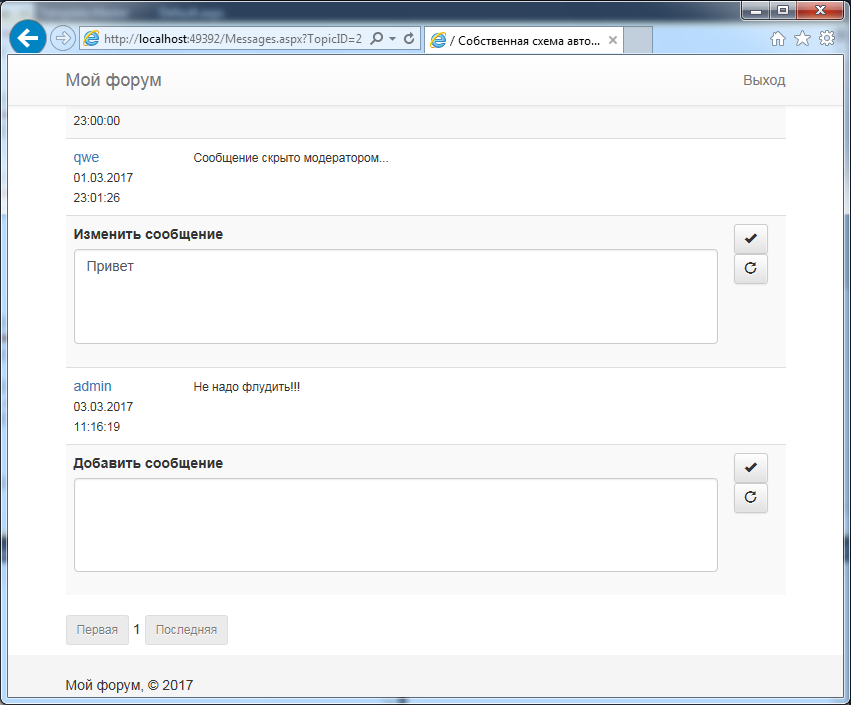


Рисунок . – Список сообщений

Страница списка пользователей приведена на рисунке 1.12.

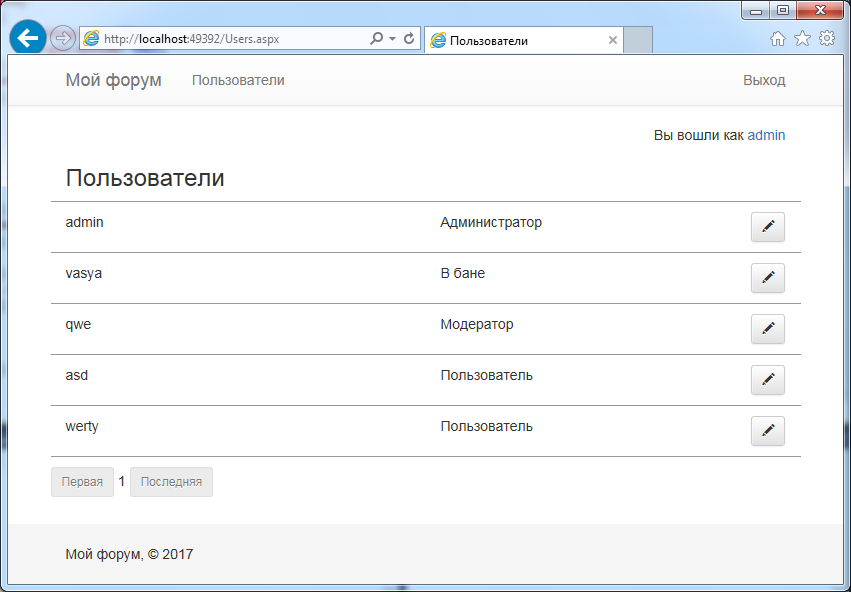


Рисунок . – Список пользователей

Страница профиля пользователя при изменении роли приведена на рисунке 1.13.

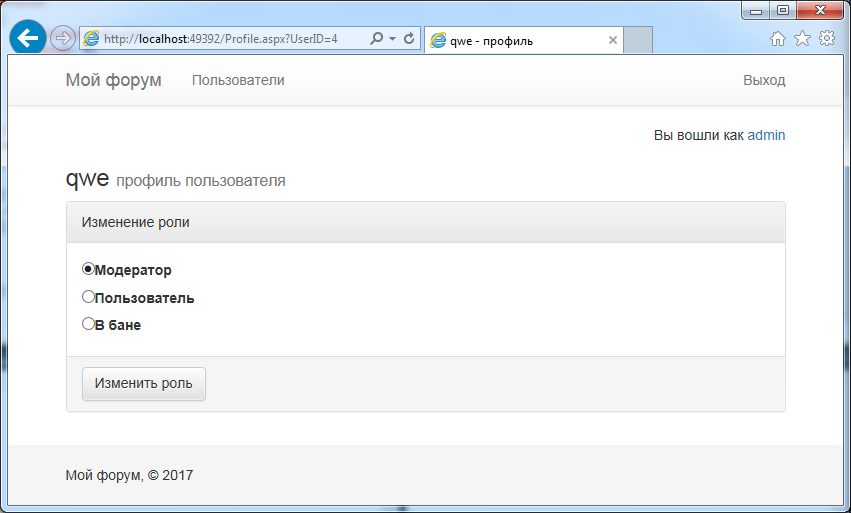


Рисунок . – Изменение роли пользователя

Страница профиля пользователя при изменении пароля приведена на рисунке 1.14.

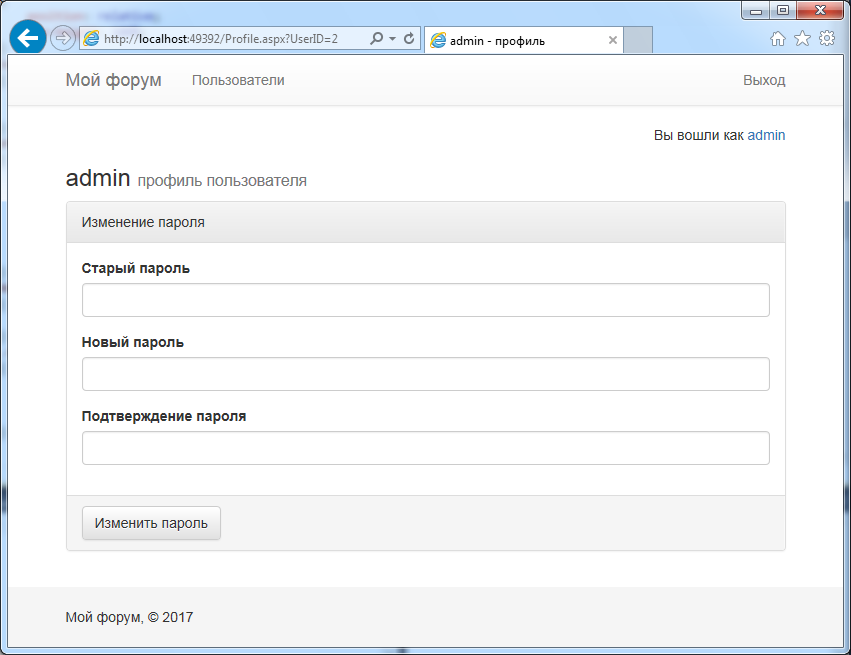


Рисунок . – Изменение пароля пользователя

# ASP.NET MVC

## Постановка задачи

Необходимо создать сайт на MVC, взяв в качестве темы и основы веб-проект, созданный в предыдущей работе. Можно использовать другую тему веб-проекта (по желанию). В данной работе необходимо продемонстрировать понимание структуры ASP.NET MVC-приложения, работу с маршрутами, контроллерами и представлениями.

## Разработка программного кода

Для доступа к данным используется Entity Framework по шаблону Code First. Таким образом, таблицы базы данных создаются автоматически на основании разработанных классов модели данных.

В соответствии с требованиями разработаны модели (классы объектов данных), приведенные в таблице 2.1.

Таблица . – Классы моделей

|  |  |
| --- | --- |
| Основные классы, с помощью которых реализован интерфейс к БД | Вспомогательные классы для реализации интерфейса с представлениями |
| Message | MessageEdit |
| Role |  |
| Rubric |  |
| Topic | TopicEdit |
| User | LogOnModel  RegisterModel  ChangePasswordModel  ChangeRoleModel |

Интерфейс с БД реализован с помощью класса ForumContext – расширение базового класса DbContext. Класс контекста определяет перечень моделей, соответствующих набору данных в БД, а также связи между моделями.

Для авторизации используется класс LoginHelper, аналогичный по функциям классу, разработанному для обычного приложения ASP.NET (см. раздел 1), адаптированный к использованию Entity Framework.

Для управления данными разработаны следующие классы контроллеров:

* AccountController – контроллер, реализующий авторизацию, регистрацию и управление данными пользователей;
* DefaultController – контроллер, реализующий управление данными разделов форума;
* TopicsController – контроллер, реализующий управление данными тем разделов форума;
* MessagesController – контроллер, реализующий управление данными сообщений тем форума.

Для реализации пользовательского интерфейса разработаны представления, приведенные в таблице 2.2.

Таблица . – Перечень представлений

|  |  |  |
| --- | --- | --- |
| Наименование контроллера | Представление | |
| Наименование | Назначение |
| AccountController | ChangePassword | Форма изменения пароля |
| ChangeRole | Форма изменения роли |
| List | Форма списка пользователей |
| Login | Форма авторизации |
| Profile | Форма профиля пользователя |
| Register | Форма регистрации |
| DefaultController | Create | Форма добавления раздела |
| Edit | Форма редактирования раздела |
| Index | Форма списка разделов (главная) |
| MessagesController | Create | Форма добавления сообщения |
| Delete | Форма удаления сообщения |
| Edit | Форма редактирования сообщения |
| List | Форма списка сообщений |
| TopicsController | Create | Форма добавления темы |
| Delete | Форма удаления темы |
| Edit | Форма редактирования темы |
| List | Форма списка тем |

Перечисленные представления используют общий макет \_Layout. Также разработано представление страницы ошибки Error.

## Демонстрация работы

Главная страница со списком разделов приведена на рисунке 2.1.

Страница профиля пользователя приведена на рисунке 2.2.

Форма добавления раздела приведена на рисунке 2.3.

Страница со списком тем раздела приведена на рисунке 2.4.

Форма добавления темы приведена на рисунке 2.5.

Страница со списком сообщений темы приведена на рисунке 2.6.

Форма редактирования сообщения приведена на рисунке 2.7.

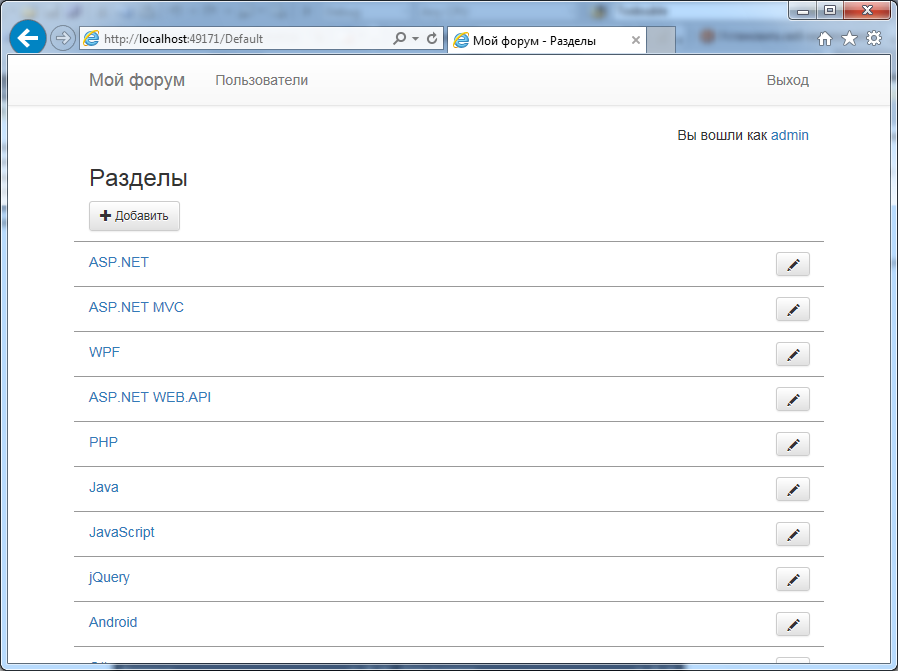


Рисунок . – Список разделов

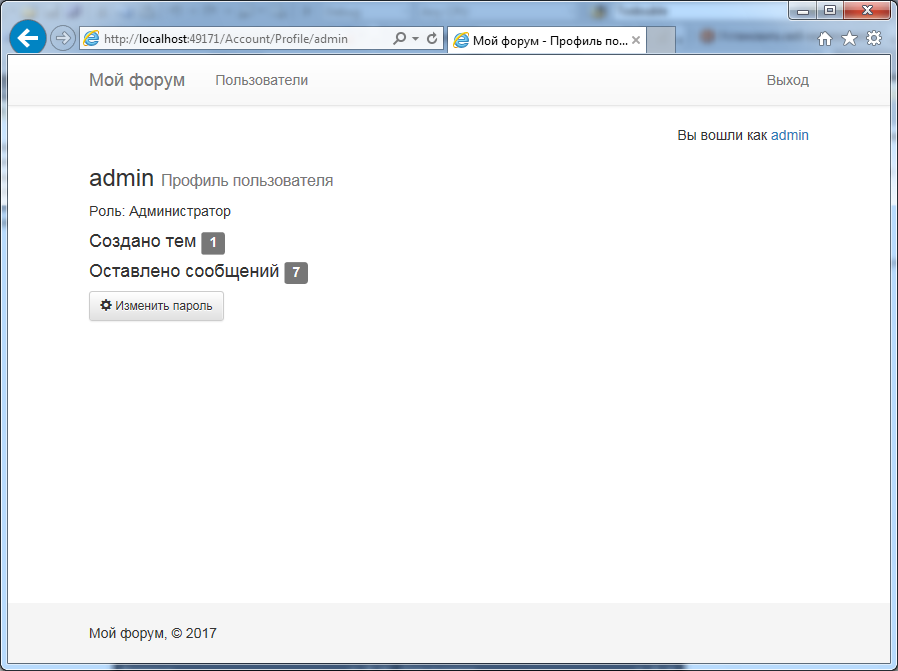


Рисунок . – Страница собственного профиля пользователя

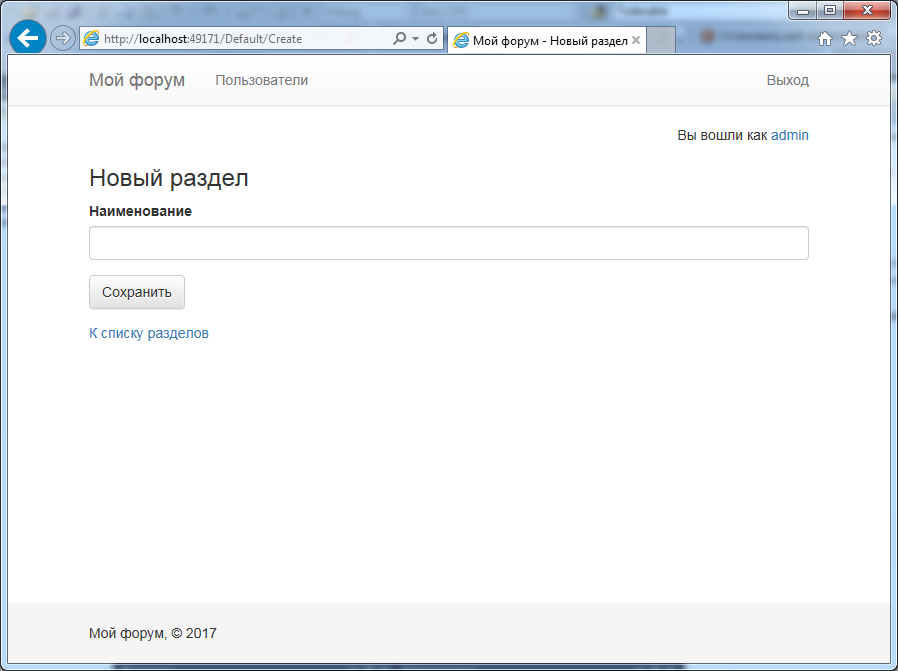


Рисунок . – Форма добавления раздела

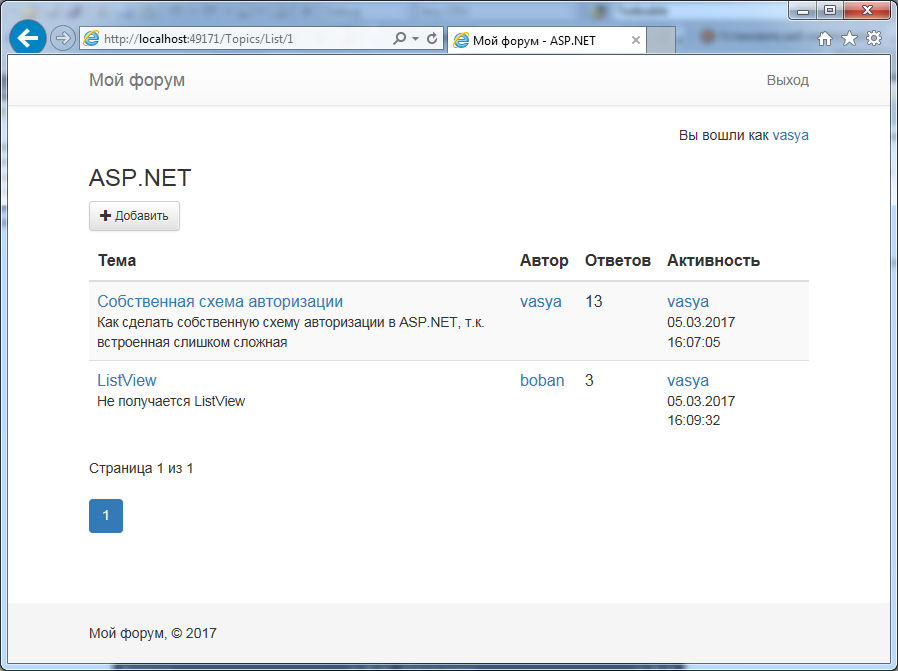


Рисунок . – Список тем раздела

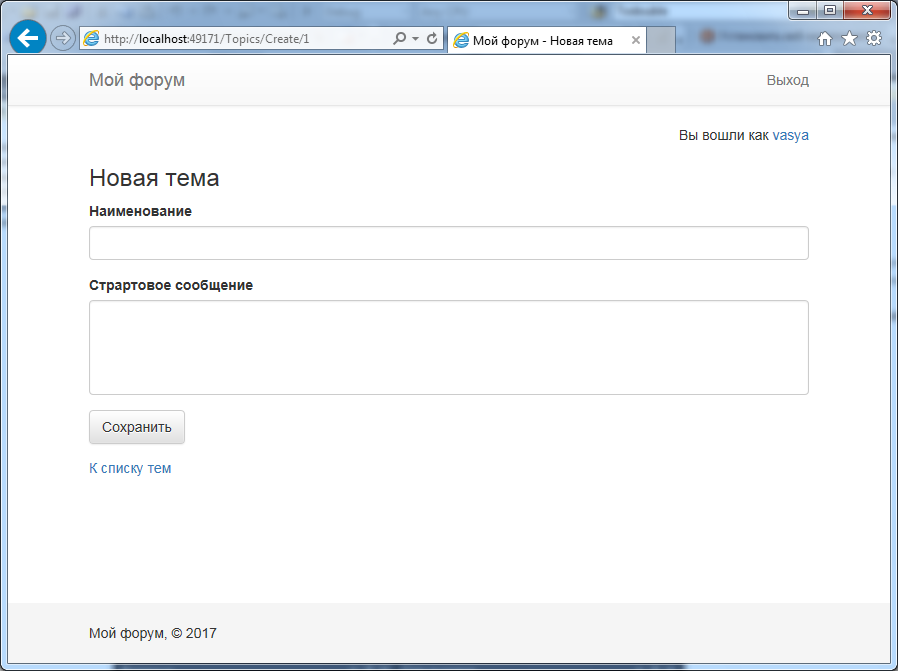


Рисунок . – Форма добавления темы

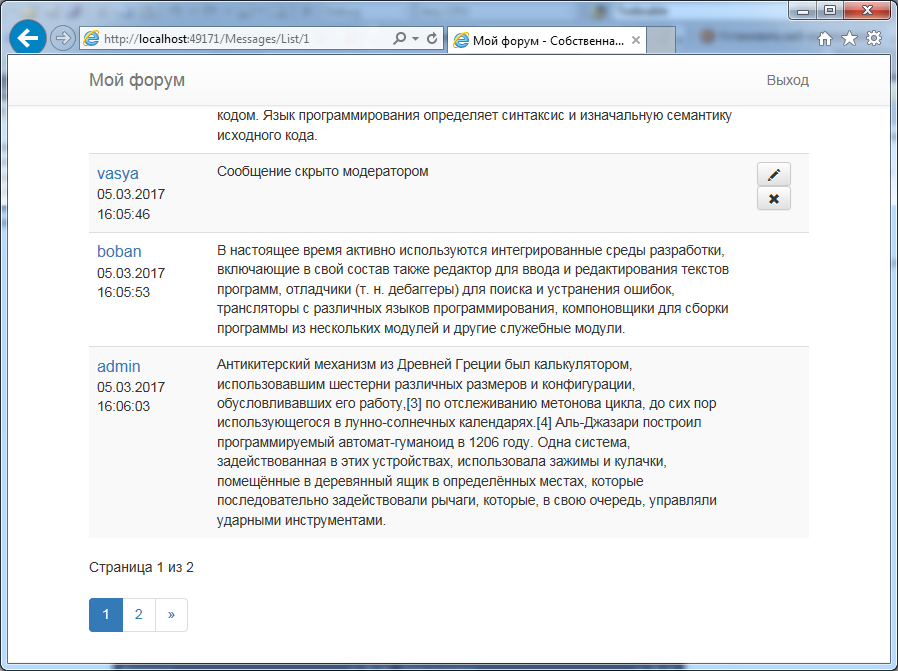


Рисунок . – Список сообщений темы

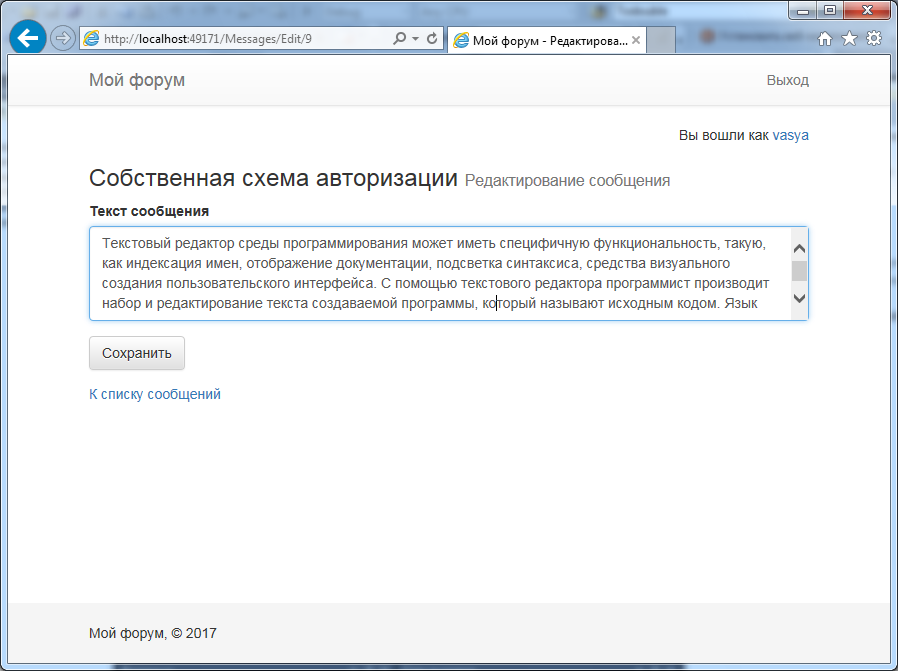


Рисунок . – Форма редактирования сообщения