

77-51038/495780 Автоматизированный показ презентаций

Молодежный научно-технический вестник # 10, октябрь 2012

авторы: Фенске А. В., Большакова Д. О.

УДК 004.415.28

fenix.iu8@gmail.com, boldol@mail.ru

МГТУ им. Н.Э. Баумана

Введение

Те, кто занимается бизнесом или наукой сталкиваются с презентациями регулярно: сегодня нужно представить новому клиенту свой проект, а завтра ознакомить коллег со свежими планами и стратегиями или сделать доклад на конференции или профильной выставке. Кроме того, все большую популярность набирают системы дистанционного обучения. Такие методы обучения накладывают определенные ограничения на процесс доставки материала слушателя посредством презентации. Поэтому задача разработки программного обеспечения для такого типа обучения является актуальной [3].

1. Описание системы

На сегодняшний день существуют множество программ для создания презентаций, например, *MS Power Point* или ресурс *Prezi.com*, позволяющий создавать презентацию в режиме онлайн. Кроме того, некоторые программы имеют набор встроенных инструментов для создания презентаций.

Однако некоторые системы ДО предъявляют к средствам показа презентаций следующее требование: переключение слайдов должно осуществляться одним пользователем, который показывает презентацию, пользователи, просматривающие презентацию, не должны иметь возможности управлять показом [3].

Для реализации поставленной задачи можно использовать следующие технологии: *Silverlight, HTML5, CSS3, jQuery, AJAX, ASP.NET* и т. п.

Рассмотрим программный продукт, созданный для реализации поставленной задачи. Данная программа позволяет осуществлять показ презентаций и просмотр презентаций в браузере. Для организации просмотра никакого специального ПО не требуется. Система разработана с использованием следующих программных решений: *Visual Studio 2010, VS2010 СУБД MS SQL Server Express*.

2. Бизнес-логика системы

В системе существуют две роли пользователей: учитель и ученик. Выбор роли осуществляется на главной странице системы. После выбора роли осуществляется переход к странице, содержащей список презентаций. В зависимости от выбранной роли страница имеет определенную функциональность.

Для учителя доступны функции добавления и удаление презентации, темы, добавление изображений в презентацию, показ презентации. Для ученика доступен просмотр презентации. При переходе во время показа учителя к следующему изображению, у ученика переход осуществляется автоматически.

3. Подготовка среды разработки

Перед началом разработки была подготовлена среда разработки, настроено необходимое программное обеспечение. Для этого был выполнен следующий набор действий:

- 1) Установлен *VS 2010*
- 2) Подключен *MS SQL Server Express*

4. Архитектура проекта

Программа поделена на уровни:

BaseAccessor – базовый класс для доступа к БД

ProjectAccessor – класс для доступа к данным.

ProjectModel – класс описывающий модель проекта. В этом классе описаны методы работы с БД системы.

В качестве структурных единиц БД выступают следующие таблицы: *Images, Presentations, Themes*.

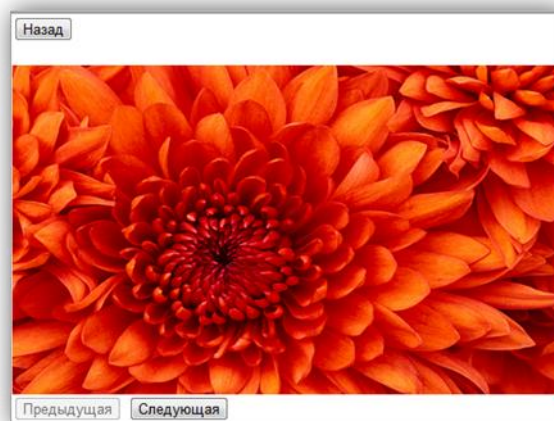


Рис.1. Показ презентации

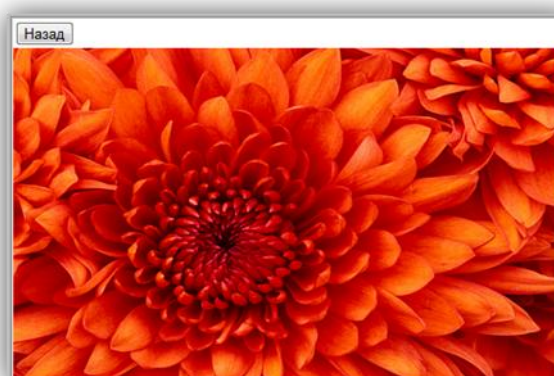


Рис.2. Просмотр презентации

5. Механизм взаимодействия компонентов

При входе в систему пользователь попадает на страницу *Default*. На странице из веб-элемента управления *DropDownList* выбирается нужная роль. Нажатие кнопки «ОК» порождает событие, обработчик которого переводит пользователя к соответствующей странице, содержащей список презентаций.

Для отображения в списке презентаций набора презентаций вызывается процедура *GetPresentation*, настраивается *GridView*, осуществляется привязка данных. Для загрузки в *DropDownList* списка тем вызывается процедура *GetThemes*.

На странице списка презентаций для пользователя в роли учителя доступны следующие действия:

- 1) Удаление презентации
- 2) Редактирование наименования темы
- 3) Показ или просмотр презентации, переход к странице показа

При переходе во время показа учителя к следующему изображению, в БД текущая картинка помечается флагом *True*. Ученик обращается к БД каждую секунду по таймеру и проверяет наличие флага *True*. Если флаг сменился на *False*, то осуществляется поиск пути картинки с флагом *True* в рамках данной сессии для данной темы и презентации. Затем происходит обновление картинки в области *UpdatePanel*. Нажатие кнопки «Добавить тему» приведет к странице добавления темы *ThemeCard*.

«Добавить презентацию» - переход к странице добавления презентации *PresentationCard*. Добавленная с локальной машины на сервер картинка сохраняется в файловой системе, путь заносится в БД, определяется принадлежность картинки к соответствующей презентации[2,3].

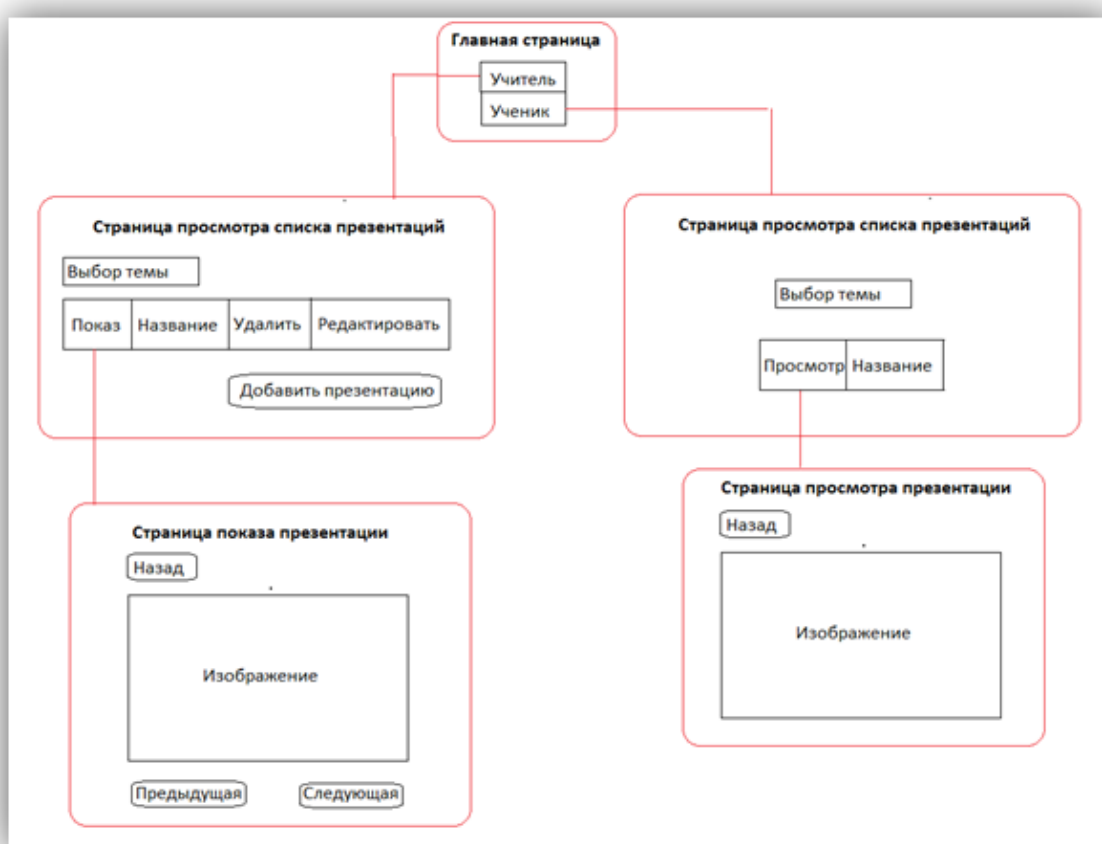


Рис.3. Схема переходов

6. Используемые технологии

При разработке данного проекта были использованы компоненты библиотеки ASP.NET Ajax Control Toolkit, такие как Timer, UpdatePanel.

Использование серверных элементов управления из библиотеки *ASP.NET AJAX Control Toolkit* — это простой способ реализовать *AJAX*-возможности в *ASP.NET*-приложении[2].

6.1. Привязка данных *Data binding*

Позволяет осуществить привязку полученных из БД данных к компоненту *ASP.NET Grid*. Для этого необходимо настроить *ObjectDataSource* и привязать его к *GridView*[1].

6.2. Элемент управления *UpdatePanel*

В приложениях *ASP.NET* при выполнении обратной отправки страницы на сервер (*postback*) вся страница визуализируется заново. Это приводит к “мельканию” в окне браузера или другой клиентской программы. Сервер обрабатывает запрос от клиента и произведенные изменения вступают в силу. В конце концов выполняется код заданного обработчика события для элемента, который вызвал *postback*-событие[2,3].

Использование элемента *UpdatePanel* позволяет устранить необходимость в полном обновлении страницы. Элемент *UpdatePanel* отмечает на веб-странице область, которая будет автоматически обновляться при возникновении *postback*-события (но без упомянутого выше *postback*-поведения в клиенте). Обмен данными в таком случае происходит через *XMLHttpRequest*-объект. Страница на сервере все равно, как обычно, обрабатывает *postback*-событие, вызывает обработчики событий и так далее, но окончательная визуализация страницы означает, что обновляются только заданные в *UpdatePanel* области. Кроме того, на одной странице можно иметь несколько элементов *UpdatePanel* и даже вкладывать элементы *UpdatePanel* друг в друга. Чтобы использовать элемент управления *UpdatePanel*, необходимо просто перетащить его на веб-форму в окне конструктора.

Однако *UpdatePanel* не может функционировать без элемента *ScriptManager* на странице. Элемент *ScriptManager* является сердцем платформы *ASP.NET AJAX*. Он берет на себя управление рабочим циклом *ASP.NET AJAX*, а также управление связанными сценариями. Элемент *ScriptManager* должен быть размещен на странице до элемента *UpdatePanel*. Иначе говоря, если просматривать исходный код страницы сверху вниз, ссылка на *ScriptManager* должна появиться раньше ссылки на *UpdatePanel*. Использование “помощника” (*Tasks Assistant*) гарантирует, что элементы размещаются правильно. Если элемент *ScriptManager* отсутствует или размещен на странице неправильно, то при попытке открыть страницу в браузере возникает ошибка.

В элементе управления *UpdatePanel* имеется область, в которую можно поместить *HTML*-код. После *postback*-события будет обновляться только этот код, если *ScriptManager* настроен на частичное обновление страницы[2].

Разметка для элемента *UpdatePanel* в данной системе выглядит так.

```
<asp:UpdatePanel ID="UpdatePanel" runat="server">
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer" EventName="Tick" />
</Triggers>
<ContentTemplate>
<asp:Image ID="CurrentImage" runat="server" Height="300" Width="500"/>
</ContentTemplate>
</asp:UpdatePanel>
```

6.3. Элемент управления *Timer*

Серверный элемент управления *Timer* предоставляет простую в использовании функциональность таймера, которую можно настроить на многократное выполнение операций в зависимости от истекшего времени. Можно запускать определенные операции с регулярным интервалом синхронно или асинхронно, не обновляя страницу вручную. Эта возможность может оказаться полезной во многих веб-приложениях.

Чтобы использовать *Timer*, на странице должен присутствовать элемент *ScriptManager*. Добавить элемент *Timer* можно путем его перетаскивания в окне конструктора.

В данном проекте таймер применяется для обновления содержимого элемента *UpdatePanel* по истечении определенного времени. Чтобы увидеть элемент *Timer* в работе, можно добавить на пустую страницу элементы *UpdatePanel* и *ScriptManager*. Затем на страницу следует установить элемент *Timer*. Кроме этого, внутри элемента *UpdatePanel* следует поместить надпись (*Label*).

```
<asp:Timer ID="Timer" runat="server" Interval="1000" ontick="Timer_Tick">
</asp:Timer>
```

У элемента *Timer* возникает событие *Timer_Tick*, внутри которого в базовом классе страницы можно выполнять какую-либо операцию каждый раз при срабатывании таймера, например, обновление картинки.

Триггер *AsyncPostBackTrigger* используется внутри *UpdatePanel* для инициирования обновлений по событию *Tick* таймера. Как выглядит разметка для этого, показано ниже.

```

<div>
    <asp:ScriptManager ID="ScriptManager" runat="server">
    </asp:ScriptManager>
</div>
<asp:Timer ID="Timer" runat="server" Interval="1000" ontick="Timer_Tick">
</asp:Timer>
<asp:UpdatePanel ID="UpdatePanel" runat="server">
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="Timer" EventName="Tick" />
    </Triggers>
    <ContentTemplate>
        <asp:Image ID="CurrentImage" runat="server" Height="300" Width="500"/>
    </ContentTemplate>
</asp:UpdatePanel>

```

В браузере в элементе *CurrentImage* каждую секунду будет появляться новая картинка. Это будет происходить без полного обновления страницы[2].

Заключение

В рамках данной работы были проанализированы подходы для решения данной задачи, включающие в себя как анализ инструментов создания системы, так и варианты ее проектирования.

Исследования показало, что подобный механизм не реализован в системах ДО. Однако, *Microsoft PowerPoint* предоставляет возможность организации широковещательного показа слайдов. Реализуется широковещание при помощи *PowerPoint Broadcast Service*. Функциональность показа доступна пользователем, имеющим установленный *MS PowerPoint* на компьютере. Просмотр доступен всем пользователям, подключенным к интернету и имеющим в распоряжении браузер.

Разработанная система корректно обрабатывает неверные действия пользователя и полностью соответствует предъявляемым требованиям[3].

Литература

1. Дино Эспозито Программирование с использованием Microsoft ASP.NET 3.5 // Русская Редакция, Питер, 2009 - с. 358-429.
2. Дуглас Дж. Рейли Создание приложений Microsoft ASP.NET // Русская Редакция, 2002 – с. 79-132.
3. Microsoft Developer Network (MSDN) // <http://msdn.microsoft.com/ru-ru/> (дата обращения 18.10.11)