

находится в непрерывном развитии, технологии автоматизации начинают использоваться в бизнес-целях, например, для сбора информации о ценах определённого продукта на различных сайтах и, следовательно, последующем ценообразовании для данного продукта.

Библиографический список:

1. Чужа В. Веб-скрейпинг и .Net / В. Чужа. – Электронный ресурс: <https://habrahabr.ru/post/101034/>, 2010.
2. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. М.: Вильямс, 2013. – 1311 с.
3. Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. Спб.: Питер, 2016. – 896 с.

РАЗРАБОТКА ИНТЕРНЕТ-ПРИЛОЖЕНИЯ НА ASP.NET MVC

Евлампов Е.Ю., Жадан М.И.

Гомельский государственный университет им. Ф.Скорины

На сегодняшний день существуют огромные проекты, которые содержат несколько сотен заданий. На таких проектах занято много людей. Для управления такими заданиями и людьми, используются системы управления времени – трэк-системы. Сегодня существует немного таких систем, и все они требуют больших денежных вложений от потребителей. Учитывая это, решено разработать трэк-систему, которая будет иметь ядро, на котором в дальнейшем можно строить приложения для различных платформ.

Целью работы является разработка системы управления временем, которая будет состоять из двух частей: общих сервисов с бизнес-логикой и веб-приложения. Эти части будут взаимодействовать друг с другом посредством связи клиент-сервер. Задачами работы являются:

- проектирование приложения с многоуровневой архитектурой;
- разработка общих переиспользуемых сервисов;
- реализация разделения пользователей на роли: администраторы, менеджера и простые пользователи;
- реализация авторизации и аутентификации на сервисах;
- реализация авторизации и аутентификации на веб-приложении;
- разработка простого и понятного интерфейса.

Для создания приложения использован стек технологий .NET. Общее ядро будет реализовано с использованием технологии WCF. Доступ к базе данных MS SQL Server будет реализован с использованием Entity Framework. Пользовательская часть приложения построена на ASP.NET MVC 5.

Разработка начинается с процесса построения архитектуры и создания приложения. Сначала разработан уровень доступа к данным и уровень логики, затем они объединены в одно приложение-хост, которое будет представлять из себя сервисы, готовые обрабатывать поступающие запросы.

Далее были написаны несколько сервисов:

- UserService – служит для управления пользователями;
- RepositoryService – служит для управления хранилищами данных;
- ProjectTaskService – управляет проектами и связанными заданиями.

После реализации сервисов и уровней доступа к данным осуществлено объединение существующих библиотек в отдельное приложение. Для хостинга использовался так называемый self-хостинг в консольном приложении.

Ядро приложения без наличия клиентов бессмысленно. Первым клиентом является веб-приложение, построенное на MVC 5. Для начала были выделены некоторые особенности приложения: мультиязычность, отображение дат в выбранном пользователем часовом поясе.

Приложение построено по паттерну Model-View-Controller. Моделью выступают ранее написанные сервисы. Далее были разработаны контроллеры. Для всех контроллеров был создан базовый класс, который содержит общую логику запросов. Для логирования всех запросов, ошибок и действий пользователей использовалась библиотека NLog [1].

После реализации определенного метода в контроллере создавалось представление – веб-страница, которую видит пользователь. Для создания представлений использовался язык разметки HTML и рендер движок Razor. Стилиевые особенности представлениям придавал bootstrap, с небольшим использованием CSS. Для удобного выбора дат пользователем был создан helper, который использовал bootstrap-datepicker с доработкой локализации.

Для авторизации и аутентификации пользователей использовалась ASP.NET Identity 2.0. Для ее использования была создана собственная реализация UserStore. Это было необходимо, так как приложение использует свои хранилища с данными, в нашем случае это WCF сервис[2].

На рисунке 96 показан вид главной страницы для авторизованного пользователя. На верхней панели пользователю доступны кнопки для входа и регистрации. По нажатию на логотип приложения происходит переход на главную страницу приложения из любой страницы.

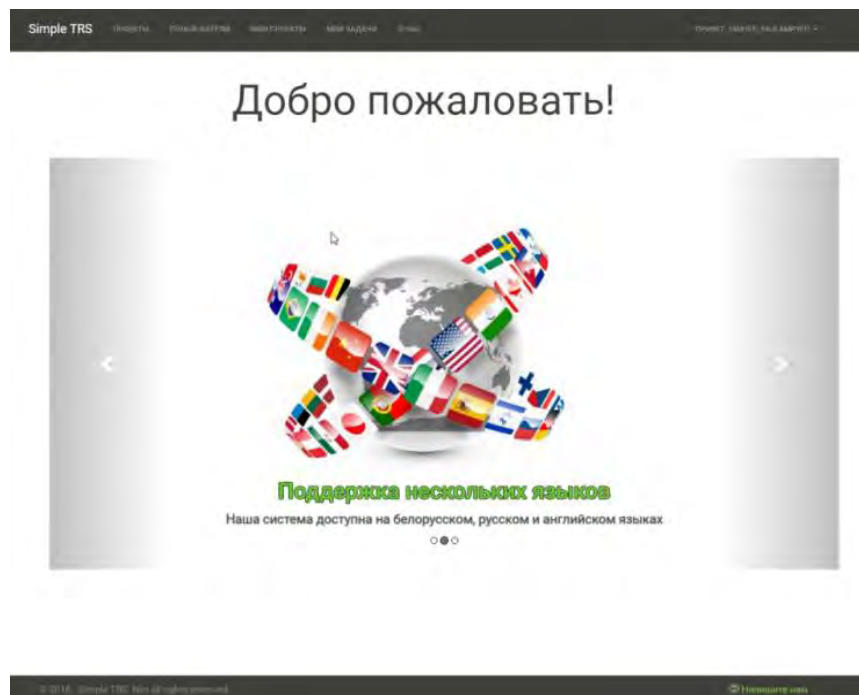


Рисунок 96 - Вид главной страницы для авторизованного пользователя

После успешной авторизации пользователя происходит перенаправление на главную страницу. Сайт доступен на трех языках: русском, белорусском, английском. На нижней панели присутствует возможность обратной связи с разработчиком.

На рисунке 97 показан вид страницы с описанием проекта. Здесь отображается его название, описание, а также список заданий. Ниже заданий находятся привязанные к проекту пользователи и их роль. Так как текущий пользователь имеет права администратора и права менеджера проекта, то он может выполнять действия по редактированию. Для пользователей, у которых нет доступа к таким действиям, соответствующие кнопки отсутствуют.

Администраторам приложения доступна страница с информацией о пользователях, на которой он может изменять данные. Для менеджеров проектов доступна страница с деталями заданий. По нажатию на кнопку «Просмотреть активность» происходит переход на страницу с отображением календаря активности для выбранного пользователя.

Результатом работы является клиент-серверное приложение для управления проектами и временем работы на них пользователей. Реализованы все поставленные задачи: спроектированы несколько хранилищ с данными, разработаны общие переиспользуемые сервисы, реализовано разделение пользователей на роли в приложении и отдельных проектах, реализована авторизация и аутентификация, разработан простой и понятный интерфейс.

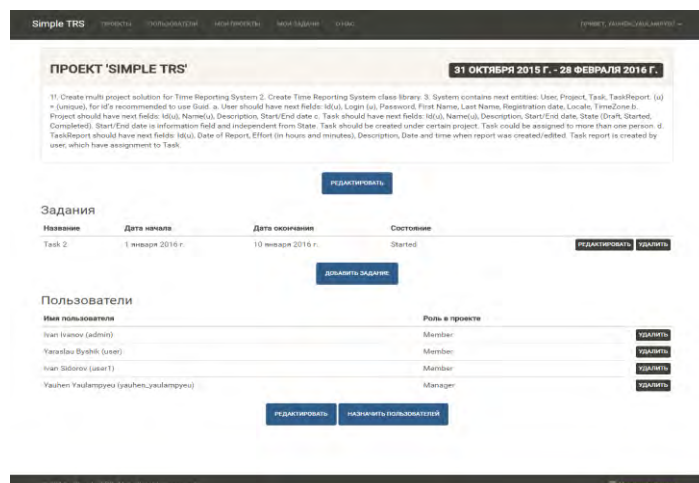


Рисунок 97 - Вид страницы с описанием проекта

Преимуществом разработанного приложения по сравнению с конкурентами является доступность в плане стоимости, также плюсом является то, что оно построено с использованием последних возможностей как WCF, так и ASP MVC 5 [3,4]. Функциональность клиентского приложения должна быть значительно расширена, поэтому, разработанное приложение можно использовать в дальнейшем, но при внесении дополнительных изменений.

Библиографический список:

1. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. – СПб.: Питер, 2013. – 896 с.
2. Lippert E., Essential C# 6.0, 5th Edition – NY: Addison-Wesley, 2015. – 1008 с
3. Löwy J. Programming WCF Services, 4th Edition – S.: O'Reilly Media, 2016. – 1018 с.
4. Freeman A. Pro ASP.NET MVC 5 Platform – NY: Apress, 2014. – 426 с.

ИССЛЕДОВАНИЕ МАГНИТНЫХ СВОЙСТВ МОНОЯДЕРНЫХ КОМПЛЕКСОВ ЖЕЛЕЗА (II) С СЕРОСОДЕРЖАЩИМИ ЛИГАНДАМИ

Ергешбаева С.Б., Еркасов Р.Ш.

Евразийский национальный университет им. Л.Н. Гумилева

В цифровую эпоху увеличилась потребность в телекоммуникациях и интернет приложениях, что ведет к увеличению спроса на устройства хранилищ данных с большой памятью. Уменьшение размера устройств хранилищ данных необходимо и может быть достигнуто на молекулярном уровне [1, 155]. Возможность адресовать одиночные молекулы или маленькие кластеры молекул могут привести к увеличению количества логических вентилях на см² на микрочипе несколькими порядками магнитуды и время отклика молекулярных устройств может составлять фемтосекунды вместо наносекунд в современных приборах. И поэтому большое внимание сфокусировано на бистабильных молекулах, которые проявляют эту фундаментально-важную способность переключаться между уровнями, что является основой в битовых расчетах [2, 221]. Спин кроссовер - явления изменения спинового состояния из низшего в высшее в молекулах под воздействием внешних факторов, таких как температура, давление, облучение светом и т.д. Комплексные соединения переходных металлов с электронной конфигурацией 3d⁴-3d⁷ способны переключаться на молекулярном уровне из высшего в низшее спиновое состояние и обладают хорошим потенциалом для использования в информационных технологиях, в дисплеях, сенсорах, пультах управления, холодильных устройствах, пищевых и медицинских складах и в агентах МРТ [3, 279].

Были синтезированы моноядерные комплексы железа (II), в котором центральный атом железа окружен двумя S и четырьмя N донорными атомами. Такое необычное для комплексов железа окружение дает новые возможности для дизайна и синтеза соединений, проявляющих спиновые переходы. Данные комплексы железа (II) были получены реакцией пер-