

## МОНИТОРИНГОВАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ПОИСКА МЕДИКАМЕНТОВ

Гайназаров С.М., Рахманов Ж.Р. *Национальный Университет Узбекистана им.  
М.Улугбека* e-mail: gaynazarsm@mail.ru, j.rahmanov@edu.uz

Рассматриваются возможности построения и реализация распределенной информационной системы поиска медикаментов. Реализация выполнена согласно концепции сервис-ориентированной архитектуры на платформах Windows Communication Foundation, Microsoft SQL Server 2014 и Visual Studio 2015.

Ключевые слова: информационная система, поиск медикаментов, SOA, WCF, ADO NET MVC.

Предлагаемая информационная система обеспечивает удобный доступ через веб-интерфейс к каталогу медицинских препаратов, а также обычный поиск и поиск по их атрибутам [1]. Кроме того, администраторы аптек имеют возможность редактирования ресурсов аптеки. Реализация выполнена в среде Microsoft Visual Studio 2015 с использованием технологий Asp.net MVC 5, ADO.NET Entity Framework, Windows Communication Foundation (WCF). Рассмотрим эти технологии подробнее, отметив их преимущества.

### Технологии реализации

Платформа ASP.NET MVC представляет собой удобную среду создания web - приложений, интегрированную с существующими функциями ASP.NET и дополненную средствами MVC (model-view-controller) (модель-представление-контроллер):

- модель (model) предоставляет данные для представления в ответ на запросы контроллера, содержит бизнес-логику приложения;
- представление (view) отвечает за пользовательский интерфейс, отображает данные, полученные от модели;
- контроллер (controller) обрабатывает команды пользователя, определяет модель для работы и связывает ее с представлением.

Бизнес-логика, расположенная в модели, включает все правила и алгоритмы, связанные с предметной областью решаемой задачи. Проще говоря, это ядро создаваемого приложения. Рассматриваемая архитектура подразумевает, что изменения в любом из компонентов оказывают минимальные воздействия на остальные части. Несколько упрощая, работу MVC-приложения можно описать следующим образом:

- команда (уведомление о нажатии кнопки, запрос адреса сайта и т.д.) передается контроллеру;
- контроллер, исходя из полученных данных, определяет и вызывает модель;
- модель на основе заложенной в ней бизнес-логики формирует набор данных;
- контроллер выбирает представление и связывает его с данными (моделью);
- представление отображает данные пользователю.

Контроллер играет роль связующего звена между моделью и представлением. При этом он стремится как можно меньше знать о подробностях их реализаций. Его задача - определить модель для обработки полученной команды и представление, которому будут отправляться итоговые данные.

Представление зависит от модели, так как использует получаемые от нее данные. Модель не зависит ни от представления, ни от контроллера. Это позволяет вести разработку модели независимо, а также создавать для нее несколько представлений.

Платформа ASP.NET MVC имеет следующие преимущества:

- облегчает управление сложными структурами путем разделения приложения на модель, представление и контроллер;
- не использует состояние просмотра и серверные формы, что делает эту платформу идеальной для разработчиков, которым необходим полный контроль над поведением приложения;
- использует схему основного контроллера, при которой запросы web-приложения обрабатываются через один контроллер, что позволяет создавать приложения, поддерживающие расширенную инфраструктуру маршрутизации, а также обеспечивать расширенную поддержку разработки на основе тестирования;
- хорошо подходит для web-приложений, поддерживаемых крупными коллективами разработчиков, которым необходим высокий уровень контроля над поведением приложения.

Платформа ADO.NET Entity Framework (EF)

- объектно-ориентированная технология доступа к данным, являющаяся объектно-реляционным отображением (object-relational mapping, ORM) решений для .NET Framework от Microsoft. Она предоставляет возможность взаимодействовать с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL. Для облегчения построения web-решений используются и ADO.NET Data Services (Astoria), и связка из Windows Communication Foundation и Windows Presentation Foundation, позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования - MVC, MVP или MVVM [2]. Платформа ADO.NET Entity Framework дает возможность разработчикам создавать приложения для доступа к данным, работающие с концептуальной моделью приложения, а не напрямую с реляционной схемой хранения. В результате можно добиться уменьшения объема кода и снижения затрат на сопровождение приложений, ориентированных на обработку данных.

Таким образом, основной задачей ORM является установка соответствия между объектами, используемыми в приложении, и таблицами, хранящимися в реляционных БД.

Создание SQL-запросов. При запросе приложением объектов ORM-библиотека самостоятельно создает SQL-код запросов и передает его в СУБД. При необходимости разработчик может вмешаться в данный процесс с целью тонкой оптимизации производительности. При этом возможно возникновение проблемы несоответствия типов. В качестве решения ORM при записи значения может использовать свойство объекта.

Абстракция используемой БД. В процессе работы с ORM-библиотекой приложение оперирует привычными для него объектами. Но при этом для хранения

информации могут использоваться различные реляционные СУБД, в частности, SQL Server, SQL Server Express, SQL Server Compact, MySQL и т.д. Это вносит дополнительный уровень гибкости в архитектуру приложения.

Приложения Entity Framework имеют следующие преимущества:

- могут работать с концептуальной моделью в терминах предметной области, в том числе с наследуемыми типами, сложными элементами и связями;
- освобождаются от жестких зависимостей от конкретного ядра СУБД или схемы хранения;
- сопоставления между концептуальной моделью и схемой, специфичной для конкретного хранилища, могут меняться без изменения кода приложения;
- разработчики имеют возможность работать с согласованной моделью объектов приложения, которая может быть сопоставлена с разными схемами хранения, возможно, реализованными в различных СУБД;
- несколько концептуальных моделей могут быть сопоставлены с единой схемой хранения;
- поддержка запросов LINQ обеспечивает проверку синтаксиса во время компиляции для запросов к концептуальной модели.

Технология Windows Communication Foundation (WCF) [3] предоставляет единую инфраструктуру разработки, повышающую производительность и снижающую затраты на создание безопасных, надежных и транзакционных web-служб нового поколения.

WCF - это технология для построения сервис-ориентированной архитектуры приложений (SOA - service-oriented architecture), что позволяет абстрагироваться от конкретной технологии, на которой этот сервис реализован, и пользоваться им из других приложений, написанных на любых других платформах, языках, технологиях, - главное, чтобы реализация клиента отвечала определенным правилам.

Кроме того, логика самого сервиса и его реализация полностью отделены от коммуникационной составляющей и существует возможность декларативно изменять способ взаимодействия с сервисом путем изменения конфигурационного файла. Можно изменить протокол взаимодействия, адрес, настроить максимальное количество подключений, ограничить размер пакетов, тайм-аут подключения к сервису и выполнения операции и многое другое.

В основе технологии WCF лежит принцип связи с помощью обмена сообщениями, и любые объекты, моделируемые в виде сообщений (например, HTTP-запрос или сообщение очереди сообщений, MSMQ), можно представить единым образом в модели программирования. Это обеспечивает универсальный интерфейс API для разных транспортных механизмов. Сообщения можно отправлять через интрасети или через Интернет общими транспортом, такими как HTTP и TCP. С помощью встроенных точек расширения WCF добавляются дополнительные транспортные механизмы. Служба WCF поддерживает несколько шаблонов обмена сообщениями, включая запрос-ответ, одностороннюю и дуплексную связь. Разные транспорты поддерживают разные шаблоны обмена сообщениями и таким образом влияют на типы поддерживаемых взаимодействий. Интерфейсы API и среда выполнения WCF также помогают отправлять сообщения безопасно и быстро.

При создании приложений для .NET разработчики пользуются средой Visu-

al Studio. В WCF и в Visual Studio есть инструменты для реализации служб.

В WCF встроена модель размещения, позволяющая размещать службы в IIS или в Managed Services на платформе Windows. WCF поддерживает развитую модель многопоточности и ограничения пропускной способности (throttling), которая позволяет управлять созданием экземпляров с минимальными усилиями. Вне зависимости от того, обрабатываются ли поступающие запросы одновременно в одном или в нескольких потоках, модель программирования остается одинаковой, так что разработчик может не вдаваться в детали (которые, однако, остаются ему доступными). WCF поддерживает различные способы обмена сообщениями, например: запрос-ответ, односторонний и дуплексный поток. Поддерживаются также пиринговые сети, в которых клиенты могут обнаруживать друг друга и обмениваться данными при отсутствии централизованного механизма управления. Словом, технология WCF важна потому, что современные приложения немыслимы без служб, а именно это и составляет назначение и смысл WCF.

### Структура программы iTeKa

Для проведения мониторинга был разработан специальный программный продукт, который предоставляет возможность дистанционного ввода информации через Интернет каждому конкретному аптечному и стационарному лечебно-профилактическому учреждению.

Программа позволяет:

- находить аптеки где есть в наличии необходимые медикаменты;
- вывести список аптек отсортированных по близости местонахождения аптек к абоненту;
- проводить обработку вводимой информации в объемах, необходимых для получения объективной информации об уровне цен в регионах, районах и в стране в целом, а также оценивать их динамику;
- позволяет обновлять изменять и добавлять перечень лекарственных средств имеющихся в аптеке;

Электронная база данных по препаратам, включенным в перечень ЖНВЛС, дает возможность:

- позволяет получить список наиболее востребованных лекарственных препаратов;
- позволяет определить список эффективно работающих аптек;
- позволяет получить информацию о качестве обслуживания системы и аптек;
- позволяет получить информацию о количестве обращении клиентов к системе за определенный период времени. Для каждой аптеки выдается информация о количестве обращений абонентов в разрезе препаратов в течении заданного времени;
- позволяет получить информацию о динамике спроса лекарственных средств на фармацевтическом рынке.

### Литература

[1]. Таненбаум Э., Ван Стеен М. Распределенные системы. Принципы и парадигмы. СПб: Питер, 2003. 845 с.

[2]. Josuttis N.M. SOA in Practice: The Art of Distributed System Design (Theory in Practice). O'Reilly Media. 2007. 759 p.

[3]. WCF 4 для профессионалов / Сибарро П. [и др.]. М.: Диалектика, 2011. 464 с.