

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Анализ существующих программных средств для управления продажами автомобилей

Перед началом разработки нового программного средства для управления продажами автомобилей было принято решение провести анализ существующих аналогов веб-сайтов. В ходе анализа, было изучено большое количество веб-сайтов и учтены их основные недостатки и достоинства, чтобы в дальнейшем при разработке избавиться от отрицательных сторон.

Веб-сайт «Mazda Cars, Sedans, SUVs & Crossovers» позволяет пользователю произвести поиск по одному конкретному производителю машин [1]. Это является недостатком, так как в нашей системе должна быть возможность хранения и вывода машин от нескольких производителей. На сайте отсутствует фильтрация, по каким либо параметрам машины, что также должно быть реализовано в нашей системе. Кроме того на сайте отсутствует функциональность, связанная с размещением фото- и видеогалерей. Таким образом, был сделан вывод, что данный веб-сайт обладает недостаточной функциональностью, и при разработке нашего программного средства необходимо это учитывать и добавить необходимые модули.

Следующим типичным сайтом является «AutoNation. New & Used Auto Dealerships» [2]. Помимо того, что он также обладает недостаточным количеством функций, он повсеместно использует технологию Flash. В настоящее время при разработке веб-сайтов от данной технологии практически отказались. Во-первых, она очень ограниченно поддерживается или совсем не поддерживается на большинстве мобильных устройств. Во-вторых, данная технология считается устаревшей, и существуют аналогичные современные технологии, посредством которых можно реализовать те же функции. Таким образом, принято решение не использовать в нашем программном средстве технологию Flash.

Также был проанализирован сайт «York New Toyota Inventory. Toyota of York» [3]. Данный сайт обладает большим количеством возможностей, по сравнению с его предыдущими аналогами. На сайте присутствует фильтрация по определенным параметрам и отображения списка найденных машин. Однако каждая страница данного списка при переключении между ними приводит к полному обновлению сайта. Это осуществляет излишнюю нагрузку на сервер, а также неудобно с точки зрения пользовательского интерфейса.

В настоящее время существует технология асинхронных запросов, называемая AJAX, которая позволяет обновлять часть содержимого страницы без перезагрузки всей страницы полностью. Это позволяет оптимизировать работу, так как сервер оперирует с меньшим количеством данных и нагрузка на него уменьшается.

Также это удобно пользователю сайта, так как он не видит обновления

полностью при переходе между страницами. Поэтому при разработке нашего программного средства, было принято решение использовать технологию асинхронных запросов Ajax.

Таким образом, проанализировав достаточное количество существующих аналогичных программных средств для управления продажами автомобилей, были выявлены их основные недостатки. При разработке нашей системы данные недостатки должны быть учтены и исправлены, для того чтобы этот продукт обладал большим преимуществом по сравнению со своими аналогами.

1.2 Обзор используемых технологий

На этапе начала проектирования программного модуля для системы были учтены требования того, что она должна быть запущена и функционировать под управлением операционной системы Windows Server 2008.

Для реализации поставленных требований могла быть выбрана одна из платформ для разработки web-приложений: Microsoft .NET, Ruby on Rails или стек технологий Java и Spring.

Учитывая, что программное средство будет запущено на операционной системе Windows, сразу было принято решение отказаться от платформы Ruby on Rails, так как её запуск и работа на данной операционной системе поддерживается ограниченно.

Оставшиеся две платформы облают схожей функциональностью. Подробное их сравнение приведено в статье «ASP.NET MVC vs. Spring» [4]. Однако, не смотря на все это необходимо учитывать условия, что в дальнейшем система может быть интегрирована с иными компонентами, которые также функционируют на Windows. Таким образом, было принято решение остановить выбор на платформе Microsoft .NET.

На данной платформе имеется возможность вести разработку на большом количестве языков программирования. Однако было принято решение использовать язык программирования C#, как наиболее современный и распространённый на данный момент. Данный язык имеет синтаксически более выразительные конструкции, которые позволяют описывать действия в коде более лаконично и ёмко. C# сочетает в себе объектно-ориентированные и контекстно-ориентированные концепции.

При разработке некоторые аспекты работы самой платформы уточнялись с использованием книги «Troelsen A. Pro C# 5.0 and the .NET 4.5 Framework» [5]. При возникновении спорных вопросов непосредственно по самом языку программирования C#, можно обращаться напрямую к его спецификации [6].

Для разработки web-приложений на платформе Microsoft .NET существуют несколько технологий. Однако было принято решение остановить свой выбор на технологии ASP .NET MVC. Она обладает рядом

преимуществ по сравнению с более старой технологией ASP .NET WebForms.

Технология ASP .NET MVC реализует в себе шаблон проектирования MVC (см. рисунок 1.1) [7].

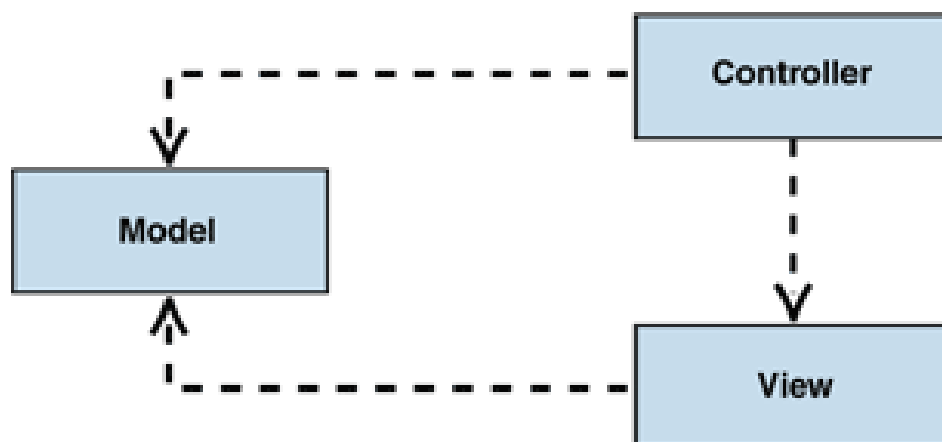


Рисунок 1.1 – Шаблон проектирования MVC

Данный шаблон позволяет реализовать концепцию разделения ответственности между частями системы.

Контроллеры призваны обрабатывать все запросы, которые посылает браузер приложению. Он обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции

Представление призвано визуализировать данные бизнес-модели в системе. Оно отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма с графическими элементами.

Модель хранит в себе данные для отображения, а также содержит всю бизнес-логику доступа к данным в базе. Таким образом, она предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать

Модель взаимодействия MVC оповещает представление о том, что в ней произошли изменения, а представления, которые заинтересованы в оповещении, подписываются на эти сообщения. Это позволяет сохранить независимость модели как от контроллера, так и от представления (см. рисунок 1.2).

При этом такое программное средство начинает обладать хорошими возможностями для тестирования.

Необходимые данные об автомобилях, некоторые системные данные, а также вся информация, необходимая для реализации бизнес-логики должна храниться в базе данных.

В нашей системе мы будем использовать базу данных Microsoft SQL

Server 2012. Это одна из наиболее мощных СУБД архитектуры клиент – сервер.

Эта СУБД позволяет удовлетворять такие требования, предъявляемые к системам распределенной обработке данных, как тиражирование данных, параллельная обработка, поддержка больших баз данных на относительно не дорогих аппаратных платформах при сохранении несмежного управления [8].

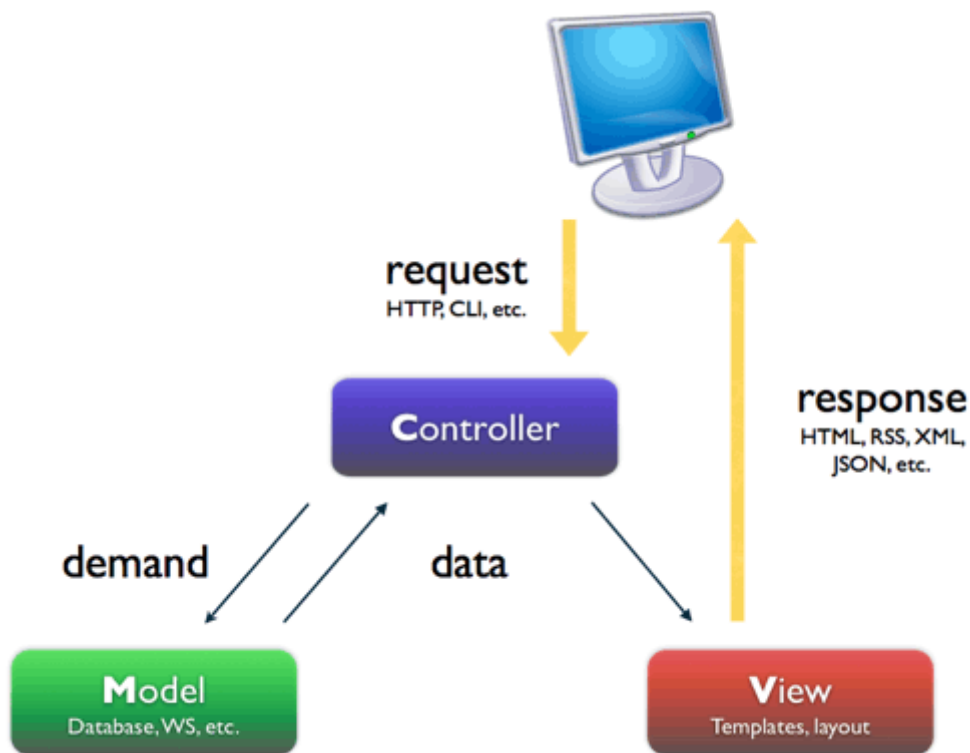


Рисунок 1.2 – Поведение модели взаимодействия MVC

Microsoft SQL Server не предназначен непосредственно для разработки пользовательских приложений, а выполняет функции управления базой данных. Для пользовательского приложения SQL Server является мощным источником генерации и управления нужными данными.

Для доступа к данным в базе данных из нашего приложения мы будем применять технологию ORM. Данная технология позволяет связать существующую базу данных с объектами, которые создаются в объектно-ориентированном языке программирования, в данном случае С#. Таким образом, создается виртуальная объектная база данных, которая позволяет более удобно представлять данные для манипулирования ими. Кроме того ORM предоставляет удобный механизм создания, изменения и удаления информации базы данных. При этом нам не нужно писать запросы на языке запросов SQL, а мы можем применять встроенные синтаксические средства языка программирования.

В нашем программном средстве мы используем ORM под названием

NHibernate. Это ORM-решение для платформы Microsoft .NET, портированное с Java. Данная библиотека является бесплатной.

NHibernate позволяет отображать объекты бизнес-логики на базу данных. По заданному XML-описанию сущностей и связей NHibernate автоматически создает SQL-запросы для загрузки и сохранения объектов.

Таким образом, даже имея большое количество таблиц в базе данных, применяя NHibernate, мы без особого труда сможем манипулировать всеми данными. Для получения дополнительных сведений можно воспользоваться официальным сайтом данной ORM [9].

При первичном анализе программных средств и технологий была попытка найти готовое решение, подходящее нам по требованиям, которое распространяется бесплатно. Никакой подходящей системы для управления продажами автомобилей найти не удалось, поэтому было принято решения разрабатывать собственное программное средство.

При этом для упрощения и ускорения разработки можно использовать систему управления контентом. Для данного выбранного стека технологий существует обладающая достаточным количеством функций, расширяемая и удобная в использовании система Orchard CMS.

Orchard – это бесплатная система управления контентом с открытым исходным кодом, построенная на платформе ASP.NET MVC. Она обладает следующими особенностями:

- открытый бесплатный и свободный проект, зависящий от запросов сообщества;
- быстрый движок с модульной архитектурой и всеми необходимыми средствами CMS;
- большие возможности в расширяемости компонентов системы;
- общедоступная онлайн-галерея модулей, тем и других компонентов расширения от сообщества;
- высокое качество типографики, внимание к компоновке и разметке страниц;
- упор на создание удобной и функциональной панели администрирования;
- быстрое развертывание системы на рабочем месте и легкая публикация на сервер.

Таким образом, данная система управления контентом хорошо подходит для реализации нашего программного средства из-за своей модульности, большого количества функций и бесплатного распространения.

Для удобной и быстрой разработки на платформе .NET использовалась в основном интегрированная среда разработки Visual Studio. Она обладает полным набором инструментов для комфортного написания кода, в том числе подсветку синтаксиса и автодополнение.

Одним из преимуществ использования данной среды разработки является встроенный пакетный менеджер NuGet [10].

Данный пакетный менеджер имеет собственную онлайн хранилище доступных библиотек. Он позволяет быстро устанавливать необходимые библиотеки, а также поддерживать их версии в актуальном состоянии.

Однако среда разработки Visual Studio, обладая большим количеством функций, требует довольно продолжительное время на загрузку проекта. Зачастую необходимо быстро изменить код и посмотреть на результат. Для этого можно использовать текстовый редактор Sublime Text 3. Он обладает базовой функциональностью по подсветке кода, которая может быть расширена с помощью соответствующих плагинов [11]. Таким образом, данным редактором удобно пользоваться при редактировании верстки веб-сайта, стилей для него, а также скриптов, написанных на языке Javascript.

Основой разработки на стороне клиента в данный момент является язык программирования Javascript. Он позволяет выполнять некоторые действия над веб-страницей прямо в браузере, не используя при этом сервер.

Таки образом, можно использовать Javascript для реализации анимации на странице, валидации пользовательского ввода или написания логики, для отображения различных элементов на странице в зависимости от изменяемых параметров.

Это все позволяет существенно снизить нагрузку на сервер, так как все действия в системе выполняются на стороне клиента без участия веб-сервера.

При разработке некоторые аспекты использования языка Javascript уточнялись с помощью книги «JavaScript: Подробное руководство» [12].

Продолжением развития концепции разработки на клиенте является технология асинхронного обмена данными с сервером Ajax. Данная технология позволяет при выполнении определенных действий на странице пользователем не обновлять страницу полностью, а лишь её часть. Таким образом, веб-сервер, получив запрос такого типа, не формирует полностью новую страницу, а отдает клиенту лишь ту часть, которая необходима. Браузер, получая обновленную разметку, встраивает её содержимое в просматриваемую пользователем веб-страницу.

Таким образом, значительно уменьшается нагрузка на веб-сервер, так как ему нужно формировать лишь часть данных. Также пользование таким сайтом более удобно для пользователя, потому как не происходит частое полное обновление странице при запросе даже незначительных данных.

Для передачи данных с помощью Ajax запросов чаще всего используются форматы данных XML и JSON. Формат JSON является более компактным для передачи по сети.

В нашем программном модуле очень удобно использовать Ajax запросы, так как нам необходимо выводить на страницу списки машин постранично.

При переходе на следующую страницу само содержимое сайта остается прежним, поэтому обновлять его будет лишним. Необходимо получить новую обновленную информацию лишь для следующих или предыдущих машин из списка. Затем, получив её, встроить новые машины на

страницу, при этом оставив все остальное содержание нетронутым.

Именно в таких типичных случаях удобно и целесообразно применять Ajax запросы. Это значительно снизит нагрузку на веб-сайт и увеличит его быстродействие, а также с сайтом станет приятнее работать пользователям, так как страница не будет полностью обновляться.

Сравнительный анализ классического запроса на веб-сервер и запроса выполняемого с помощью технологии Ajax показан на рисунке 1.3

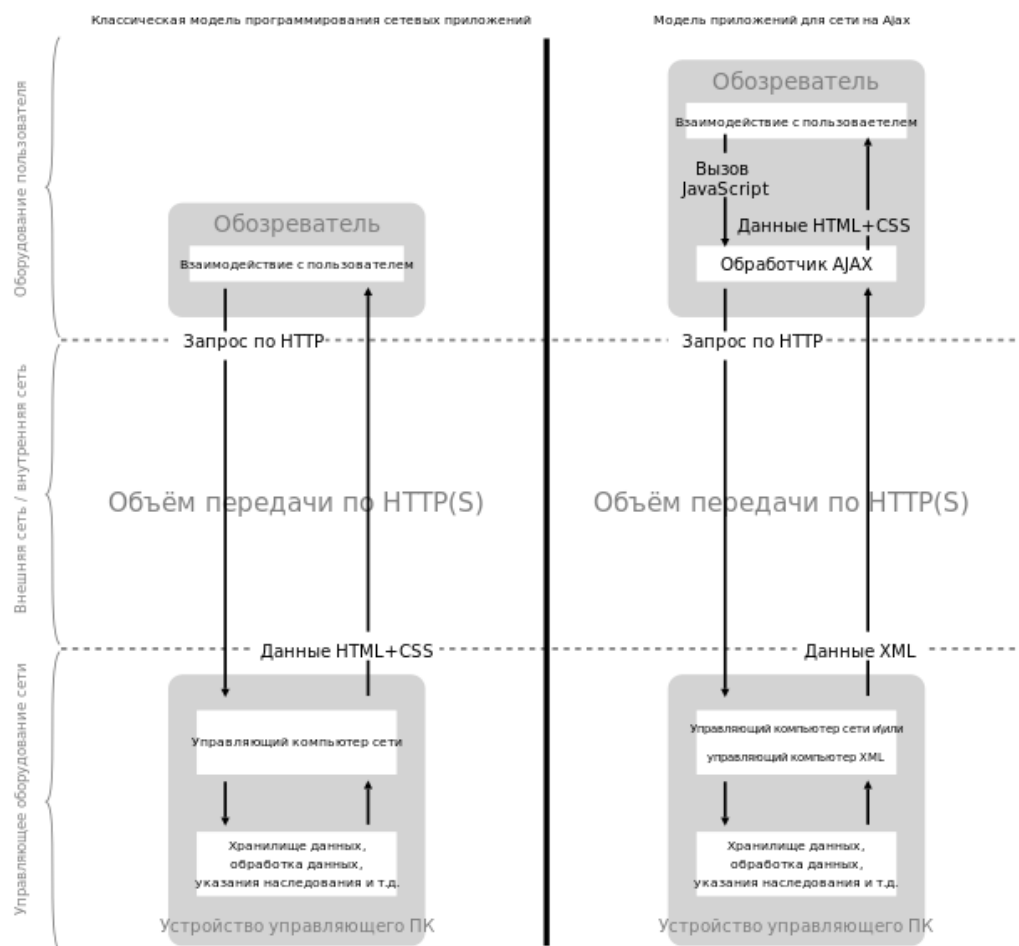


Рисунок 1.3 – Сравнительный пример классического и Ajax запроса

Помимо применения чистого языка программирования Javascript для реализации некоторой логики на клиентской стороне, есть возможность применения различных уже написанных библиотек.

Наиболее известной и популярной Javascript библиотекой является jQuery. Она имеет собственный синтаксис и позволяет удобно манипулировать с объектами на странице, получая или устанавливая им необходимые свойства, параметры или классы.

Преимуществом данной библиотеки является её простота и прозрачность в использовании, а также гарантированная одинаковая работа во всех наиболее популярных браузерах.

jQuery имеет также удобную функциональность по созданию Ajax запросов. Так как в нашем проекте было решено их применять, то использование этой библиотеки упростит и ускорит разработку.

При использовании данной библиотеке, по всем спорным вопросам можно обращаться непосредственно к её документации [13].

На основе языка программирования Javascript также создано несколько библиотек, позволяющих очень удобно передавать на страницу данные, визуализировать их и в дальнейшем изменять.

Большинство данных библиотек построены на основе шаблона MVVM (см. рисунок 1.4).

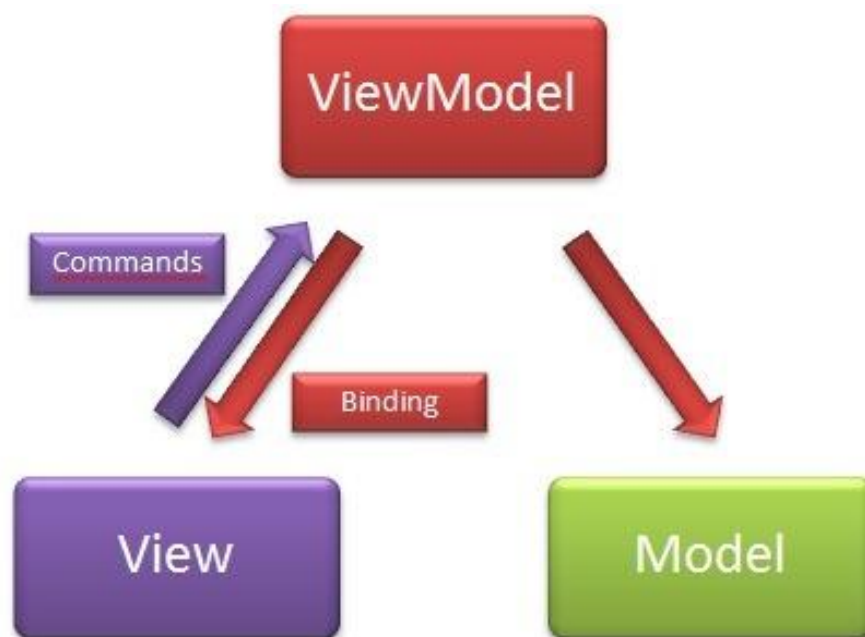


Рисунок 1.4 – Структура шаблона MVVM

Данный шаблон очень похож на ранее описанный MVC. Отличие заключается в том, что на стороне клиента реализуется View Model, которая позволяет хранить в себе состояние всех данных на странице. При этом данная модель привязывается к элементам интерфейса на странице, что позволяет удобно манипулировать ими, сокращая количество обращений у серверной модели.

Наиболее распространенной реализацией данного шаблона проектирования является Javascript библиотека KnockoutJS. При разработке проекта планируется её частое и повсеместное использование.

При возникновении, каких либо вопросов необходимо обращаться к её спецификации, в которой есть большое количество примеров ее применения и использования [14].