**Введение**

На сегодняшний день применение баз данных и бизнес-приложений приобрело весьма важное значение для многих организаций, которые для упрощения своей работы применяют компьютерные технологии.

Большинство, если не все бизнес-приложения предназначены для обработки бизнес-данных. Самые первые бизнес-решения так и назывались — программы обработки данных*.* Эффективное хранение, обработка и взаимодействие с данными — только важная составляющая управления предприятием, в тоже время компании инвестируют значительные средства в разработку компьютеризированных системы для эффективного решения этих задач.

В настоящее время, несмотря на повышение компьютеризации общества, в сфере бизнеса и торговли до сих пор нет средств, позволяющих в достаточной мере автоматизировать процесс пополнения склада товарами.

Одной из основных задач можно рассматривать проблему предсказания предложения, а также оперативную корректировку данных при возникновении необходимости в этом.

О своевременности и актуальности рассматриваемой проблемы говорит тот факт, что большую часть своего времени администрация магазина или завода тратит на анализ спроса предыдущих месяцев. Огромное количество магазинов и отсутствие предложений в данной сфере гарантирует высокую потребность в данном продукте.

Объектом для создания бизнес-приложения я решил взять бетонный завод. Данная программа предлагает хранение данных о покупателях и их покупках.

Базу данных будут использовать не только администрация бетонного завода, а также клиенты, совершая свои покупки используя отдельное веб-приложение.

Целью данного курсового проекта является создание приложения, которое может хранить данные о покупках и анализировать их, помогая администрации предсказывать какое количество товаров необходимо завезти на склад.

В данной курсовой работе будет использована специальная объектно-ориентированная технология на базе фреймворка .NET для работы с данными — Entity Framework, которая будет взаимодействовать с SQL Server 2014. Приложение для администрирования написано на Windows Application Forms C#, а веб-приложение для клиентов предприятия реализовано на технологии ASP .NET MVC 5.

**1. ASP .NET MVC 5**

Платформа ASP.NET MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

Модель (model) представляет класс, описывающий логику используемых данных.

1. **Entity Framework**

Entity Framework представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища.

Отличительной чертой Entity Framework является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем не только извлекать определенные строки, хранящие объекты, из бд, но и получать объекты, связанные различными ассоциативными связями.

1. **Windows Forms**

Windows Forms представляет простейшую технологию для создания графических настольных программ на платформе .NET. И в этом разделе выкладываются все материалы, посвященные созданию оконных приложений с использованием технологии Windows Forms, в том числе по работе различными элементами управления, двухмерной и трехмерной графикой.

**4. SQL Server 2014**

SQL Server 2014 представляет собой мощный полнофункциональный сервер баз данных, отличающийся высокой производительностью быстротой освоения и удобным интерфейсом администрирования. Под его управлением могут работать базы данных в широком диапазоне от уровня среднего звена предприятия до распределённых баз масштаба корпорации.

**Основная часть**

1. **Анализ проблемы**

Моделирование деятельности бетонного предприятия нужно начать с клиентского приложения. Для заказа пользователь будет вводить свои данные для своей идентификации и выбирать понравившийся товар, а также его количество. В соответствие с количеством и ценой товара за штуку будет формироваться сумма заказа. Это приложение необходимо будет распространять, так что веб-приложение здесь подходит больше всего.

Руководству нужны функции

1. **Анализ функциональных требований**

Для реализации поставленной задачи мне необходимы два приложения. Первое это приложение для клиентов бетонного завода, а второе для администрирования.

Здесь должна быть ЮМЛ схема 1.

Для клиента доступны функции:

1. Покупка товаров.
2. Выбор из ассортимента товаров.
3. Возможность просмотра цены в Долларах.

Здесь должна быть ЮМЛ схема 2.

Для администратора доступны функции:

1. Создание, редактирование и удаление производимых товаров. В случае, когда завод начнёт производить новые товары не нужно будет вносить в SQL Server, а можно это будет сделать прямо из приложения.
2. Пополнение склада товаров, а также просмотр статистики и истории продаж
3. Создание, редактирование и удаление пользователей программы. Это дополнительная возможность завести несколько учётных записей на нескольких сотрудников.
4. Просмотр предсказаний продаж на следующий месяц.
5. **Предсказание на основе экстраполяции значений продаж за предыдущие месяцы**

Экстраполяция — это особый тип аппроксимации, при котором функция аппроксимируется вне заданного интервала, а не между заданными значениями.

Иными словами, экстраполяция — приближённое определение значений функции в точках , лежащих вне отрезка , по её значениям в точках .

Для экстраполирования значений я использовал MATLAB R2016a, а если конкретнее использовал функцию interp1 со значением extrap. Для использования функций MATLAB в приложение на C# нам потребуется интегрировать написанную функцию в C#.

Чтобы собрать dll-библиотеку MATLAB'а для интеграции в C#.NET, нужно настроить линкер, т.е. какой средой мы будем собирать проект. Для начала нужно установить среду выполнения MCR (MATLAB Component Runtime). Это набор dll-библиотек для полной поддержки языка MATLAB. Установочный файл можно найти:

...\MATLAB\R2016a\toolbox\compiler\deploy\win32\MCRInstaller

Для настройки линкера в командном окне MATLAB'а набираем mbuild -setup. Со всем соглашаемся и выбираем нужную нам среду, в нашем случае это Microsoft Visual Studio 2015. Получаем что-то похожее:

Please choose your compiler for building standalone MATLAB applications:

Would you like mbuild to locate installed compilers [y]/n? y

Select a compiler:

[1] Lcc-win32 C 2.4.1 in C:\PROGRA~1\MATLAB\R2016a\sys\lcc

[2] Microsoft Visual C++ 2008 SP1 in C:\Program Files\Microsoft Visual Studio 15

[0] None

Compiler: 2

Please verify your choices:

Compiler: Microsoft Visual C++ 2008 SP1

Location: C:\Program Files\Microsoft Visual Studio 15

Are these correct [y]/n? y

После подобного сообщения — линкер настроен.

Следующий шаг — это написание самой функции, компилятор MATLAB'а понимает только функции т.е., каждый сценарий должен начинаться с function (желательно заканчиваться end) и быть отдельным m-файлом.

function vq3 = hint(x,v)

xq = length(x)+1;

vq3 = interp1(x,v,xq,'pchip','extrap');

end

После этого мы уже можем получить динамическую библиотеку. Набираем в командном окне MATLAB'a deploytool. Создаем новый .NET Assembly проект, указываем размещение. Далее создаем класс, в моём случае он называется MLHint, добавляем в него m-файл и нажимаем кнопку build.

После компиляции создается интересующая нас папка с установочным файлом для нашей библиотеки. На эту библиотеку в дальнейшем и будет ссылаться наша программа.

1. **Реализация**

**Заключение**

1 —

2 —

3 —