

РАЗРАБОТКА КОМПЛЕКСА ПРОГРАММ ДЛЯ ХРАНЕНИЯ И ПОИСКА ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ

А.В. Савкина

Мордовский государственный университет им. Н. П. Огарева

Аннотация. В статье представлен алгоритм и соответствующий ему комплекс программ для хранения и поиска изображений с помощью вейвлет-преобразований с описанием интерфейса и структуры базы данных.

Интерфейс автоматизированной системы основан на технологии Visual Studio.NET, которая обеспечивает возможность создания приложений и веб-служб XML и может использовать для разработки любой язык программирования. Для хранения изображений комплекс программ использует технологию SQL Sever Enterprise Manager, что обосновано ее высокой эффективностью и развитыми средствами работы с данными. Для работы с базой данных применяется технология ADO, которая позволяет клиентским приложениям получать доступ и работать с информацией из различных баз данных через SQL Server провайдера.

Разработанный комплекс программ для хранения и поиска изображений дает возможность быстрого поиска изображений, отвечающих определенным критериям с возможностью навигации по категориям. Данный комплекс программ помогает грамотно организовывать базы данных и управлять коллекциями изображений.

Ключевые понятия: вейвлет-преобразование, вейвлет-коэффициенты, хранение изображения, поиск по образцу, база данных.

В зависимости от конкретной области применения информационные системы могут очень сильно различаться по своим функциям, архитектуре, реализации. Однако можно выделить свойства, которые являются общими для всех информационных систем.

Информационные системы ориентируются на конечного пользователя. Поэтому информационная система обязана обладать простым, удобным, легко осваиваемым интерфейсом, который должен предоставить пользователю все необходимые для его работы функции, но в то же время не дать ему возможность выполнять какие-либо лишние действия.

Средства взаимодействия с коллекциями изображений или данных довольно сложны. Для организации и поиска в коллекции в настоящее время используются новые информационные технологии, базирующиеся на мощном математическом аппарате.

Разработка комплекса программ для хранения и поиска изображений дает возможность нахождения изображения, отвечающего определенным критериям. Кроме того, разработанный комплекс программ организует изображения с возможностью навигации по категориям: изображения со сходным содержанием располагаются рядом. Разработанная система хранения растровых изображений удовлетворяет всем этим требованиям, помогает создавать базы данных и управлять коллекциями изображений.

Разработанный комплекс программ позволяет решать поставленную задачу с помощью вейвлет-преобразований.

Система Visual Studio.NET упрощает разработку эффективных и надежных корпоративных решений, помогающих коммерческим организациям действовать более эффективно в изменяющихся рыночных условиях. Ключевым фактором для этого является применение полнофункциональных средств разработки и масштабируемых многократно используемых серверных компонентов.

Из пакета Microsoft Visual Studio .NET Enterprise Architect используется средство Microsoft Visual C# .NET, которое предоставляет удобную среду разработки для создания Windows-приложений и .NET-приложений, Web-приложений и XML Web-сервисов. Visual C# .NET включает ставшую стандартом отрасли библиотеку ATL (Active Template Library) и MFC (Microsoft Foundation Class), расширения языка C#, а также мощную интегрированную среду разработки (IDE), позволяющую разработчикам эффективно создавать и отлаживать код.

В сочетании с .NET Common Language Runtime C# может применяться для разработки объектно-ориентированных программ с привлечением всего опыта C, C++ или COM, что позволит создавать быстрые и мощные приложения. Поддерживая шаблоны и имея оптимизирующий компилятор, Visual C# .NET генерирует высокоэффективные приложения и компоненты. Данный продукт позволяет разрабатывать целый комплекс Windows-приложений, включая Web-приложения, приложения для мобильных устройств, а также решения для «тонкого клиента» [1].

При добавлении нового изображения в базу данных оно масштабируется до размера 128 x 128 пикселей. Такой размер выбран в качестве оптимального, при котором происходят небольшие потери данных и время преобразования изображения такого размера в вейвлет-коэффициенты займет не слишком много времени. Также изображение для преобразования должно быть обязательно квадратным, размером по степеням двойки.

Затем изображение раскладывается на компоненты – R (red – красный), G – (green – зеленый), B – (blue – синий).

Перейдя от цветовой модели RGB (красный, зеленый, синий) к цветовой

модели YIQ, можно разграничить информацию о яркости (Y) и информацию о цвете (I и Q) и вычислять вейвлет-преобразование для каждой компоненты изображения по отдельности. Поскольку восприятие человека более чувствительно к изменениям Y и менее чувствительно к изменениям Q, мы можем расширить допуск на погрешность по компоненте Q изображения. Цветовые модели RGB и YIQ являются аппаратно-ориентированными. В отличие от них предложенная Смитом модель HSV (Hue, Saturation, Value) ориентированна на человека. Она позволяет задавать цвета, опираясь на интуитивные понятия тона, насыщенности и яркости.

Потом выполняется вейвлет-преобразование для каждого цветового канала. В качестве вейвлета выбран вейвлет Хаара, позволяющий максимально быстро вычислять искомые коэффициенты и в качестве двумерного вейвлет-преобразования реализовано нестандартное разложение изображения, так как оно является более эффективным. Для изображения $m \times m$ стандартное разложение требует $4(m^2 - m)$ операций, тогда как нестандартное разложение требует только $8/3(m^2 - 1)$ операций.

Нестандартное разложение представляет чередование операций над строками и столбцами. Сначала выполняется один этап горизонтального попарного усреднения и нахождения разности значений пикселей в каждой строке изображения, затем – к каждому получившемуся столбцу. Этот процесс рекурсивно повторяется только на квадрантах, содержащих средние значения в обоих направлениях [2].

Алгоритм нестандартного разложения приведен ниже:

```

procedure DecompositionStep (c: array [1...2j] of reals)
  for i ← 1 to 2j/2 do
    c'[i] ← (c[2i - 1] + c[2i])/√2
    c'[2j/2 + i] ← (c[2i - 1] - (c[2i - 1] + c[2i])/2)/√2
  end for
  c ← c'
end procedure.

```

```

procedure NonstandardDecomposition(c: array [1... 2j, 1...2k] of reals)
  c ← c/2j      {нормализуем исходные коэффициенты}
  g ← 2j
  while g ≥ 2 do
    for row ← 1 to g do
      DecompositionStep(c[row,1...g])
    end for
    for col ← 1 to g do
      DecompositionStep(c[1.. .g,col])
    end for
    g ← g/2
  end while
end procedure

```

Из полученного массива коэффициентов сохраняются только 60 максимальных коэффициентов для каждого цветового канала, необходимых для оптимального запроса изображения. Если оставить большее количество коэффициентов, то поиск в базе данных будет осуществляться очень долго. В то же время если коэффициентов взять меньше, то точность поиска окажется слишком низкой.

При сравнении изображений изображение-запрос масштабируется до размера 128 x 128 пикселей. Затем оно раскладывается на компоненты RGB.

Затем выполняется такое же вейвлет-разложение изображения-запроса, как и для помещаемого в базу данных, и вновь отбрасывается вся информация, за исключением среднего значения цвета и 60 самых больших коэффициентов.

Затем поочередно из базы данных извлекаются хранящиеся там изображения и анализируются коэффициенты. Используя эти наборы коэффициентов, относящихся к изображению-запросу Q и изображению, хранимому в базе данных J , по выбранной линейной метрике сравнения изображений

$$|Q - J|_1 = \sum_{i,j} |Q[i, j] - J[i, j]| \quad (1)$$

определяем количество и точность совпадений коэффициентов. Если $|Q - J|_1$ меньше заданной точности ε у всех 60 коэффициентов по какому-либо каналу, то считается, что изображение-запрос полностью соответствует хранящемуся в базе данных изображению. Если совпало более 5 метрик по какому-либо каналу, и у них $|Q - J|_1$ меньше заданной точности ε , то изображение считается похожим на запрашиваемое [3].

Схема алгоритма сравнения изображений представлена на рисунке 1.

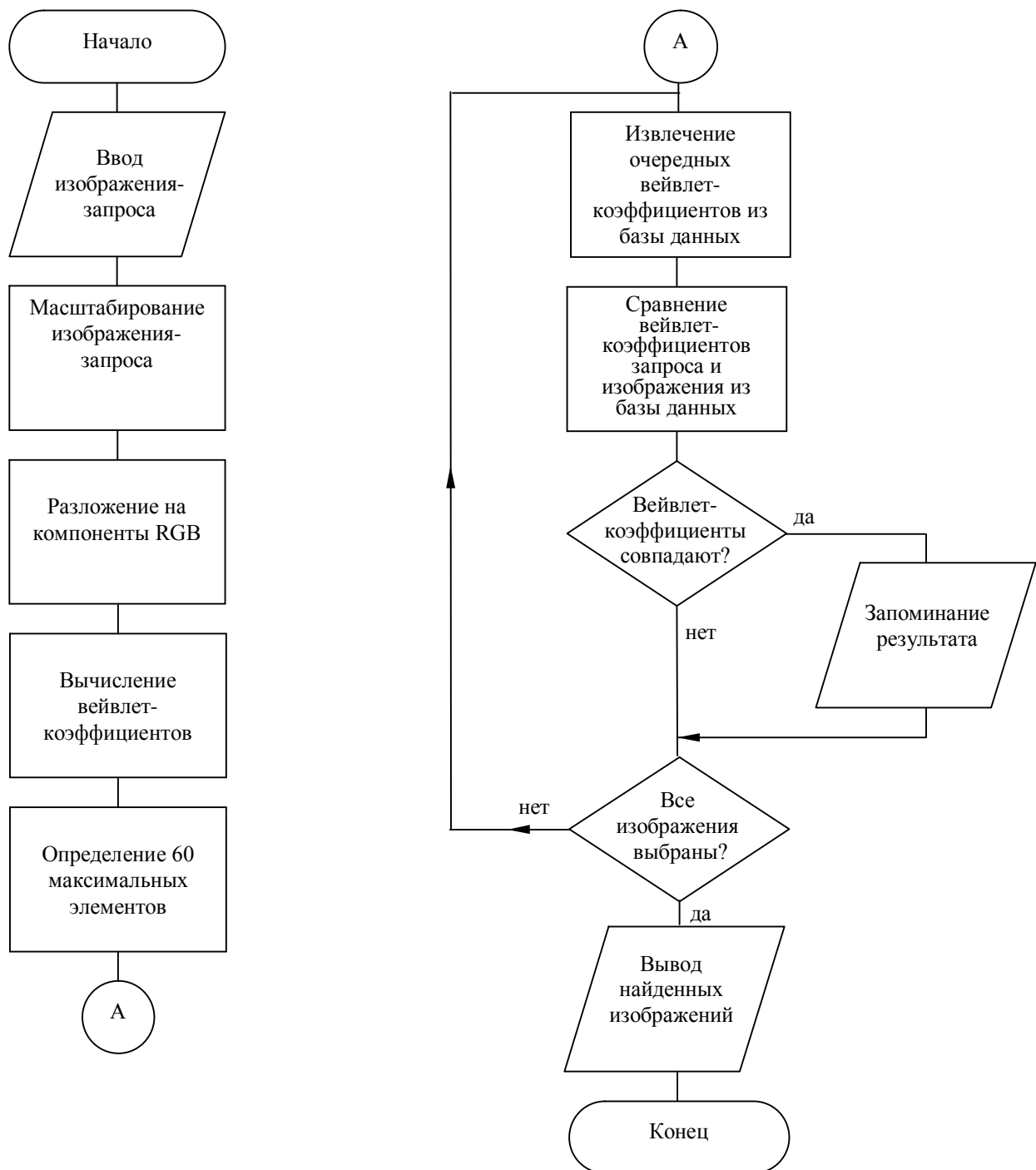


Рис.1. Схема алгоритма поиска изображения по изображению-запросу

В результате создания системы была разработана структура базы данных (рис. 2), которая содержит два класса: Images и Categories, каждый из которых отвечает за одну таблицу в базе данных.

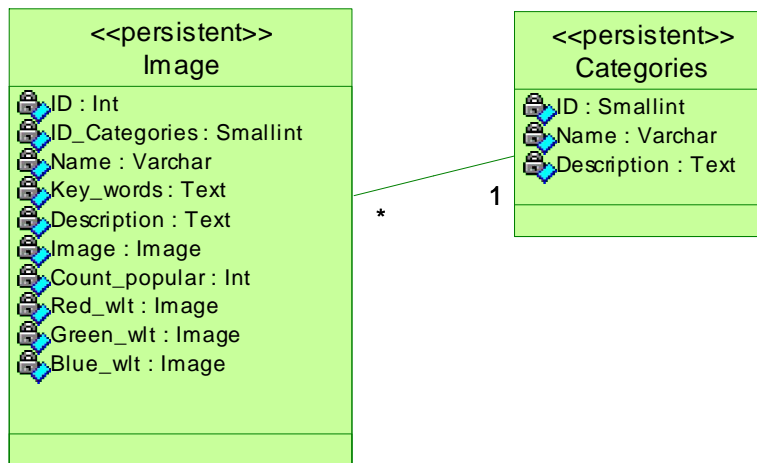


Рис.2. Схема базы данных

Создание базы данных осуществляется по спроектированной схеме базы данных. База данных создана в приложении SQL Sever Enterprise Manager. Использование именно этой СУБД обоснованно ее высокой эффективностью и развитыми средствами работы с данными.

В приложении SQL Sever Enterprise Manager создана новая база данных BD_Images, которая состоит из связанных между собой двух таблиц. Основной является таблица с изображениями и данными о них – Images. Эта таблица связана с таблицей категорий – Categories. Создание таблиц осуществляется в окне Design Table путем ввода наименований столбцов, выбора типа для каждого столбца и изменения некоторых других параметров (рис. 3).

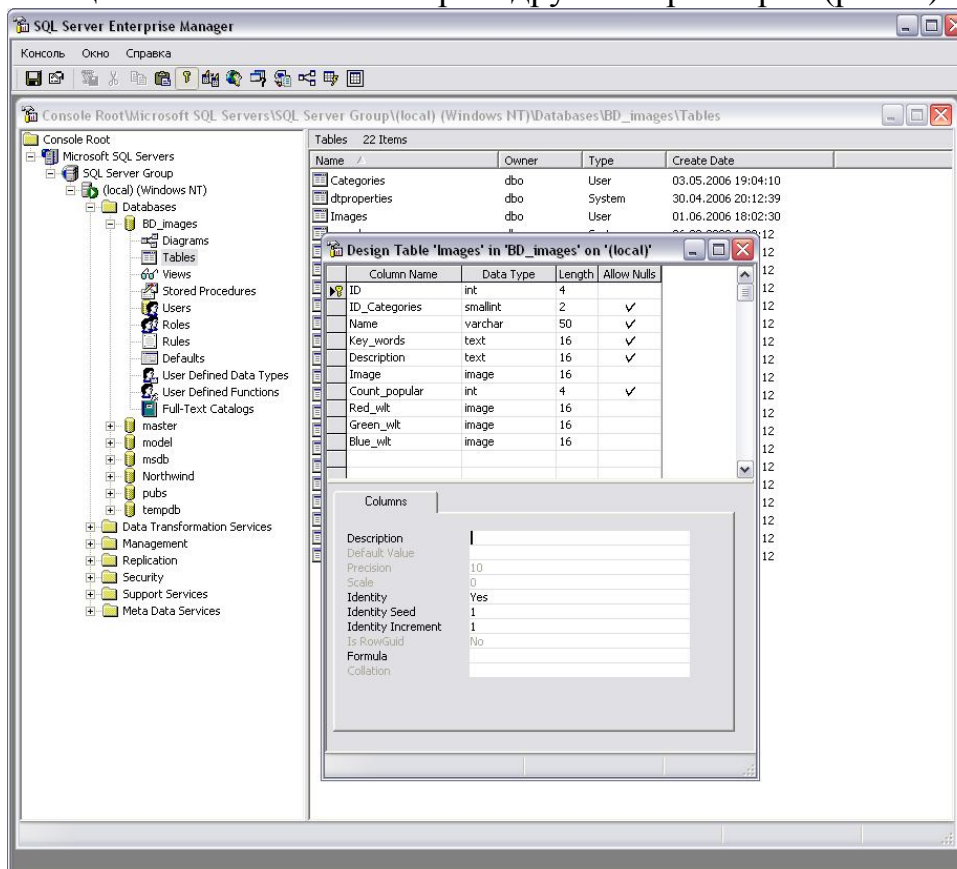


Рис.3. Окно создания таблицы Images

Работа с базой данных осуществляется с помощью технологии ADO (ActiveX Data Objects). Технология ADO позволяет клиентским приложениям получать доступ и манипулировать данными из различных баз данных через SQL Server провайдера. Основными преимуществами ADO являются легкость в использовании, высокая скорость и незначительное использование памяти. ADO также поддерживает ключевые возможности для построения клиент/серверных и Web-приложений [4].

Для разработки клиентского приложения и обработки запросов изображений используется технология ASP.NET, которая является надежной и простой для разработки Web-ориентированных корпоративных решений. ASP.NET позволяет создавать сложные, высокофункциональные, интерактивные и хорошо защищенные приложения Internet.

Приложения Web-форм – это приложения, графический интерфейс пользователя которых основан на динамическом HTML. Как правило, представление HTML создается в ответ на достаточно сложное обращение к серверу баз данных или какому-либо другому источнику.

Важное место в создании системы занимает создание удобного пользовательского интерфейса.

Пользовательский интерфейс обеспечивает взаимодействие программы с пользователем системы. Общая эффективность системы зависит не только от быстроты действия и надежности программной части, но и от безошибочной работы человека. Поэтому интерфейс должен быть не только функционален (то есть предоставлять всю необходимую информацию пользователю), но и эргономичен. При работе с программой с удобным, «дружественным», интерфейсом, пользователь делает меньше ошибок, меньше устает. Удобное расположение элементов управления, таких как кнопки, всплывающие меню и т.п., сокращает количество движений «мышью», что не только сокращает время работы, но и уменьшает нагрузку на кисть оператора.

Руководствуясь всем вышеперечисленным, разработаем интерфейсы для тех форм, которые будут использовать при своей работе с системой хранения изображений пользователи. Также предусмотрим обработку ошибок в тех частях кода, исполнение которых зависит от правильности вводимой пользователем информации (рис. 4).

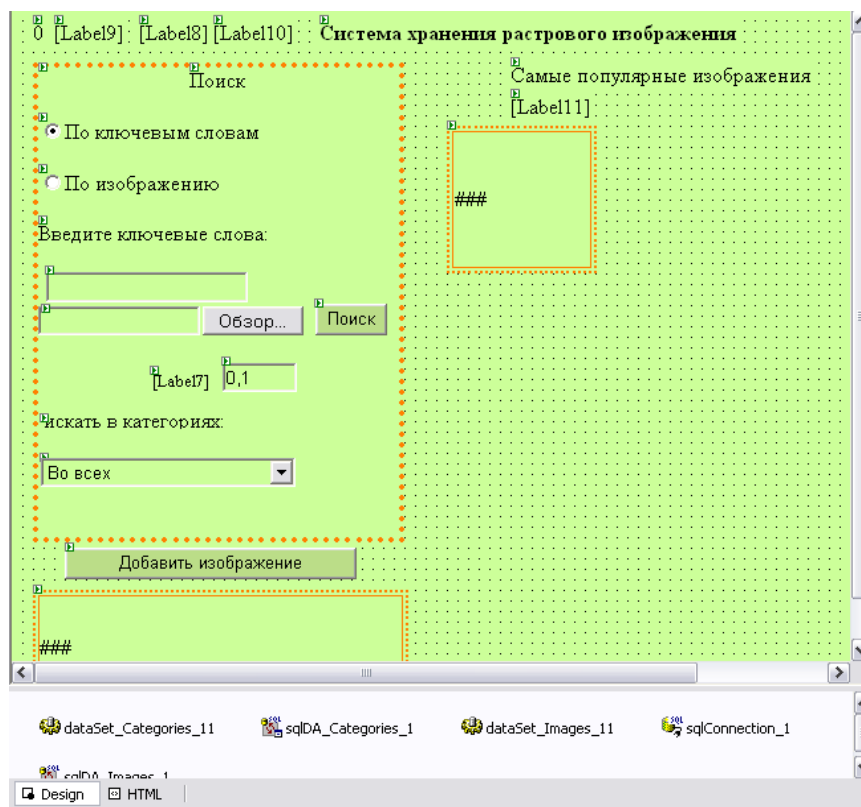


Рис.4. Окно создания главной формы MainForm.aspx

Разработанная система состоит из Web-форм, которые предназначены для вывода информации пользователю, добавления изображения в базу данных, добавления новой категории в базу данных, просмотра конкретного изображения и данных о нем, просмотра изображения в натуральную величину.

Пользователь имеет возможность поиска изображения по ключевым словам или по изображению запросу. Также имеется возможность просмотра изображений из базы данных по категориям. Результат выводится в таблицу. Пользователь может добавить изображение в базу данных. Поля «Путь к изображению» и «Название изображения» являются обязательными. И если пользователь попытается нажать кнопку «Добавить» не введя хотя бы одного из этих полей или выбрав файл, не содержащий растровое изображение, то система предупредит его об этом.

Рабочие окна программы приведены на рисунках 5-7.

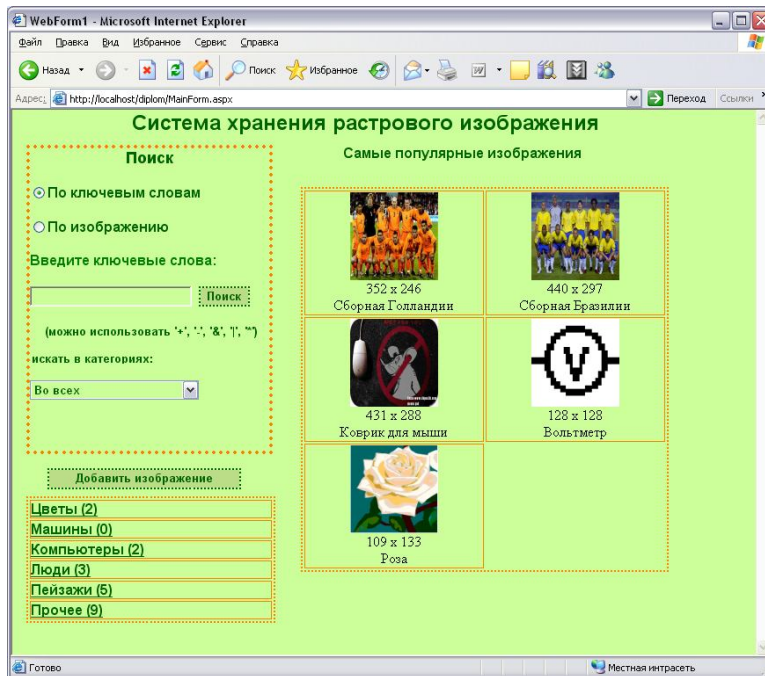


Рис.5. Начальный вид главной формы приложения MainForm.aspx

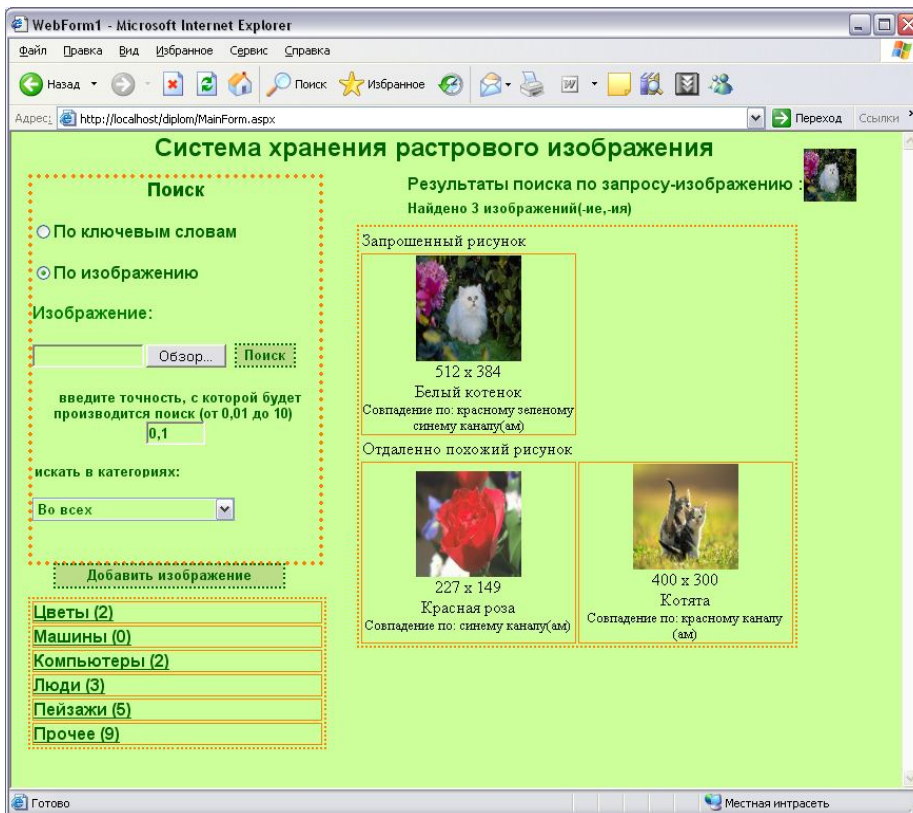


Рис.6. Вид главной формы после поиска изображения по изображению-запросу

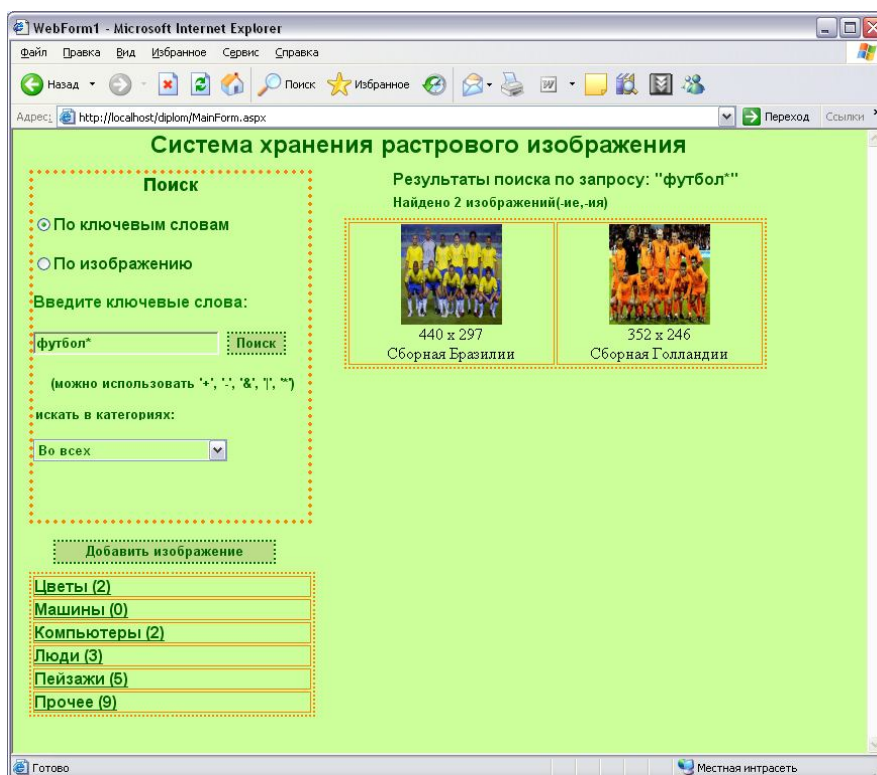


Рис.7. Вид главной формы после поиска изображения по ключевым словам

Список использованной литературы

1. Плат, Д.С. Знакомство с Microsoft.NET / Пер. с англ. / Д.С. Платт. – М.: Издательско-торговый дом «Русская редакция», 2001. – С. 240.
2. Столниц, Э. Вейвлеты в компьютерной графике /Пер. с англ. / Э. Столниц, Т. ДеРоуз, Д. Салезин. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2002. – С.272.
3. Уэлстид, С. Фракталы и вейвлеты для сжатия изображения в действии. Учебное пособ. / С. Уэлстид. – М.: Издательство «Триумф», 2003. – С.320.
4. Сеппа, Д. Microsoft ADO.NET / Пер. с англ. / Давид Сеппа. – М.: Издательско-торговый дом «Русская редакция». – 2003. – С.640.

Сведения об авторе

Савкина Анастасия Владимировна, аспирантка Мордовского государственного университета им. Н.П.Огарева, преподаватель кафедры автоматизированных систем обработки информации и управления

Научный руководитель – Федосин Сергей Алексеевич, к.т.н., профессор, зав. кафедрой «Автоматизированные системы обработки информации и управления», (8-342)290603, fedosinsa@mrsu.ru