САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА

Факультет В и ЗО

Дипломная работа

на тему

Разработка базы данных
«Современное телекоммуникационное оборудование»
и реализация Web-интерфейса для работы с ней

Дипломник **Бажанов В.А.**
Руководитель работы **Доронин Е. М.**

Санкт-Петербург
2004 г.

РЕФЕРАТ

Дипломная работа посвящена разработке требований, структуры и реализации в сети Internet базы данных «Современное телекоммуникационное оборудование».

Обоснована необходимость и целесообразность использования в учебном процессе кафедры Обработки и передачи дискретных сообщений такой базы данных.

Для реализации базы данных выбрана СУБД MySQL. Web-интерфейс для работы с базой данных реализован с помощью серверного языка сценариев PHP Hypertext Preprocessor. Выполнено содержательное наполнение базы и показаны примеры работы с базой данных.

Пояснительная записка к дипломной работе содержит 109 листов, включая 37 рисунков.

Ключевые слова: база данных, сервер баз данных, СУБД MySQL, Web-интерфейс, язык PHP, Adobe Acrobat, PDF-формат.

СОДЕРЖАНИЕ

Введение 6

1. Обоснование целесообразности использования базы данных «Современное телекоммуникационное оборудование» в учебном процессе кафедры ОПДС 7

2. Разработка требований к базе данных и к Web-интерфейсу 14

3. Способы построения баз данных 18

3.1. Иерархические модели 20

3.2. Сетевые модели 23

3.3. Реляционные модели 26

4. Организация пользовательского интерфейса для доступа к базам данных 32

5. Выбор методов и инструментальных средств для реализации базы данных и Web-интерфейса 39

5.1. Web-сервер 40

5.2. Обзор серверных программ для различных ОС 41

5.3. Сервер баз данных MySQL 43

5.4. Выбор методов доступа к базе данных 47

5.5. PHP: Hypertext Preprocessor 48

5.6. Архитектура Web-баз данных 51

6. Разработка структуры, информационное наполнение и описание сценариев работы с базой данных 53

6.1. Проектирование базы данных 56

6.2. Выбор формата файла для описания телекоммуникационного оборудования 58

6.3. Разработка программных модулей 60

6.4. Реализация интерактивного каталога 68

6.5. Реализация интерфейса администрирования 79

7. Руководство пользователя 90

Заключение 107

Список литературы 108

ВВЕДЕНИЕ

На сегодняшний день актуальным является вопрос повышения эффективности учебного процесса при подготовке инженеров-связистов. Одним из путей повышения эффективности может быть применение современных компьютерных и телекоммуникационных технологий.

На кафедре ОПДС регулярно внедряются различные инфокоммуникационные технологии – используются электронные учебные пособия, элементы дистанционного обучения и др.

В дипломной работе решена задача создания базы данных, содержащей сведения о телекоммуникационном оборудовании, для использования в учебном процессе кафедры ОПДС.

Отмечена целесообразность наличия на кафедре ОПДС базы данных «Современное телекоммуникационное оборудование», разработаны требования для проектируемой базы данных и структура базы данных. Рассмотрены особенности построения систем управления базами данных в сети Internet.

Для реализации базы данных выбрана СУБД MySQL, а Web-интерфейса – язык PHP Hypertext Preprocessor. Выполнено начальное информационное наполнение базы данных. Приведен порядок работы с базой данных и ее администрирование.

1. Обоснование целесообразности использования базы данных «Современное телекоммуникационное оборудование» в учебном процессе кафедры ОПДС

Для обоснования целесообразности использования базы данных «Современное телекоммуникационное оборудование» на кафедре ОПДС назовем учебные и научные вопросы, которыми занимается кафедра.

Из названия кафедры (ОПДС - Обработка и передача дискретных сообщений) следует, что рассматриваются вопросы, связанные с обработкой, преобразованием и передачей информации.

Кафедра является одной из старейших выпускающих кафедр университета, была организована как кафедра телеграфии вместе с созданием Ленинградского электротехнического института связи имени проф. М. А. Бонч-Бруевича в 1930 году. Задачей кафедры было обеспечение специальной подготовки будущих специалистов в области систем документальной электросвязи.

Основными направлениями деятельности кафедры являются передача данных, документальная электросвязь, системы телематики, электронная почта и современные информационные технологии, широко используемые в глобальной компьютерной сети Internet.

Среди новых курсов можно выделить «Системы документальной электросвязи», «Компьютерные сети передачи данных», «Мультимедиа технологии в системах передачи данных», «Системы телематики и электронная почта», «Проектирование автоматизированных систем обработки информации и управления» и др.

Кафедра располагает компьютерной лабораторией по информационным технологиям, в которой студенты изучают программное телекоммуникационное обеспечение и приобретают практические навыки

работы с такими информационными технологиями как электронная почта, телефакс, видеотелеконференции, телетекст, мультимедиа.

Развернута и функционирует локальная вычислительная сеть. Имеется доступ к глобальной компьютерной сети Internet. Кафедра имеет свой Web-сайт.

Кроме того, на кафедре развернут микропроцессорный класс, где студенты изучают теоретические основы передачи дискретных сообщений.

При обучении студентов используются современные достижения науки и техники, как в области аппаратных, так и программных средств вычислительной техники.

Направлениями научной работы кафедры являются:

- исследование и построение корректирующих кодов;
- исследование эффективности систем тактовой и цикловой синхронизации, качества факсимильной связи и передачи данных по каналам связи;
- исследование телематических систем.

В настоящий момент имеющаяся на кафедре вычислительная техника стала незаменимым средством в проведении учебного процесса. Кафедра считает, что компьютеризация учебного процесса - одно из основных направлений реформы высшей школы, обеспечивающее повышение качества подготовки инженеров-связистов, повышение эффективности вузовской работы.

Вычислительная техника на кафедре используется в трех качествах:

- как объект изучения;
- как средство обучения;
- как средство для вычислений и машинного моделирования систем и сетей связи в научных исследованиях.

Студенты всех специальностей изучают персональный компьютер как терминал и его использование в качестве оконечного оборудования данных (DTE - Data Terminal Equipment), а также в качестве рабочей станции и файл-сервера локальной вычислительной сети и серверов (http, ftp, smtp, pop3, telnet и другие) сети Internet.

Совместно с соответствующим аппаратным, терминальным и сетевым программным обеспечением персональные компьютеры изучаются как средство для обработки и передачи дискретных сообщений при организации компьютерной связи на ПЭВМ.

В результате изучения дисциплин кафедры студенты приобретают знания по основам компьютерной телекоммуникации, включающим принципы построения и использования терминальных программ; состав и структуру программного обеспечения для передачи и обработки факсимильных сообщений, электронной почты, для поддержки коммуникаций между пользователями в локальных вычислительных сетях и глобальной компьютерной сети Internet; программные способы защиты информации в компьютерных сетях.

Студенты получают практические навыки в пользовании телекоммуникационными программами для передачи и приема файлов и факсимильных сообщений; в составлении, передаче, приеме и обработке сообщений в системах электронной почты; в разработке программных фрагментов для криптографической защиты информации.

Студенты разрабатывают и исследуют машинные модели сложных технических и организационных элементов систем связи, разрабатывают и отлаживают нетривиальные управляющие алгоритмы для систем синхронизации и автоматического управления многосвязных объектов.

Из программных средств на кафедре изучаются следующие:

- телекоммуникационные программные пакеты (Telix, BitFax, Bitware, Faxpro и другие);

- языковые и программные средства для реализации телекоммуникационных сценариев (SALT-язык пакета Telix);
- программные пакеты для организации E-mail в системе REX400 и сети Internet (User Agent, Internet Mail, Outlook Express, The Bat! и другие);
- криптографический пакет PGP (Pretty Good Privacy);
- программы для проведения мониторинга сети Internet;
- программные средства для поиска информации и навигации в сети Internet;
- язык машинного моделирования GPSS.

В дисциплине «Мультимедиа технологии при передаче дискретных сообщений» студенты изучают программные и аппаратные средства, позволяющие использовать компьютер для работы с неподвижными изображениями и движущимся видео, анимированной компьютерной графикой, текстом и звуком, для проведения видеотелеконференций.

Также студенты получают начальные навыки по созданию и сопровождению Web-сайтов.

Применение персональных компьютеров в качестве средства обучения позволяет эффективно решать задачу интенсификации учебного процесса. На кафедре применение компьютеров как средств обучения развивается по следующим направлениям:

- разработка моделей систем, подлежащих изучению, и, как частный случай, создание на основе моделей, так называемых программных тренажеров;
- разработка и внедрение автоматизированных обучающих систем и курсов;
- внедрение в учебный процесс типовых пакетов прикладных программ.

Сущность первого направления составляет использование математических моделей систем автоматического управления, систем синхронизации, многосвязных систем, устройств защиты от ошибок, преобразования сигналов и других элементов телекоммуникационных систем и сетей. Работа с подобными моделями позволяет ставить перед студентами различные по характеру и сложности проблемные ситуации, требующие принятия решения, качество которого оперативно оценивается и доводится до обучаемого.

Возможность многократного анализа одних и тех же или близких ситуаций с оценкой различных вариантов решений способствует развитию у студентов творческого мышления.

Наиболее эффективно используются математические модели в дисциплинах «Основы теории управления», «Теория автоматического управления», «Моделирование систем ПД», «Компьютерные сети ПД» и другие.

Вторым направлением использования компьютеров в качестве средства обучения является внедрение автоматизированных обучающих систем и курсов. Разновидностью таких курсов являются электронные пособия, которые могут размещаться на локальных и сетевых компьютерах. В последнем случае учебное пособие может иметь специальным образом организованный интерактивный режим (диалог "обучающая система - обучаемый") и выступать как элемент дистанционного обучения. В простейшем случае диалог строится в виде последовательности порций учебного материала и контрольных вопросов. Обучаемый получает на экране компьютера группу порций учебного материала, по усвоению которой он делает попытку ответить на контрольные вопросы. При этом он может обратиться к электронному учебному пособию за уточнениями новых, недостаточно усвоенных им понятий или для повторного просмотра порций учебного материала. Таким образом, реализуется индивидуальный подход в

усвоении каждым студентом нового учебного материала. Информация о качестве усвоения материала каждым студентом фиксируется в памяти электронной обучающей системы и может быть выдана преподавателю по его запросу либо на монитор, либо на печатающее устройство.

На кафедре наиболее активно используются электронные учебные пособия, в том числе для дистанционного обучения при изучении дисциплин «Программное обеспечение СПД», «Системы телематики и электронная почта», «Программное обеспечение СТД», «Мультимедиа технологии при передаче дискретных сообщений», «Основы телекоммуникаций», «Системы ПД», «Основы теории управления», «Теория автоматического управления» и другие.

Третьим направлением использования компьютеров в качестве средства обучения является использование пакетов прикладных программ (ППП). Использование ППП и отдельных расчетных программ позволяет взамен рутинной вычислительной работы сфокусировать внимание студентов на инженерном анализе получаемых результатов, поручить им выполнение значительно более глубоких по содержанию заданий, чем те, которые выполнялись бы без компьютера.

ППП и расчетные программы используются в дисциплинах «Обработка экспериментальных данных на ЭВМ», «Методы и модели передачи данных», «Моделирование систем», «Надежность АСОИУ», «Информационные технологии» и др.

Эффективность учебного процесса в большей степени зависит от информационного обеспечения учебных занятий и доступности нужной информации. Из рассмотренных выше задач, решаемых кафедрой, следует, что в большинстве дисциплин, преподаваемых на кафедре, изучается современное телекоммуникационное оборудование.

В настоящее время фундаментальная и справочная литература не успевает отслеживать быстро развивающуюся телекоммуникационную отрасль.

Преподаватели, студенты, инженерно-технический состав кафедры затрачивают много времени на поиск необходимой информации по телекоммуникационным средствам.

Поэтому была поставлена задача – обеспечить сотрудникам кафедры и студентам быстрый и достаточно результативный поиск сведений по современному телекоммуникационному оборудованию. Эта задача может быть решена различными путями, в том числе созданием специальной базы данных. Эта база данных должна регулярно пополняться новыми сведениями о телекоммуникационном оборудовании и позволить пользователям оперативное их получение и использование.

Оперативности и доступности получения данных будет способствовать то, что база данных будет размещена на Web-сервере, а для работы с ней будет реализован интуитивно понятный Web-интерфейс.

В дипломной работе такая задача решается с учетом современного уровня развития сетевых и вычислительных технологий, рассмотрены возможные варианты реализации базы данных с помощью различных инструментальных средств.

Разработанная база данных должна повысить эффективность учебного процесса кафедры. Но чтобы база данных исправно функционировала и успешно выполняла свое предназначение, необходимо разработать требования к ней.

2. Разработка требований к базе данных и к Web-интерфейсу

Разрабатываемая база данных и взаимодействие с ней (интерфейс) должны отвечать следующим основным требованиям:

- Обеспечение хранения, пополнения, редактирования, удаления и обработки данных о современном телекоммуникационном оборудовании;
- Обеспечение доступа к базе данных через сеть Internet;
- Быть удобной в использовании (не требовать от пользователя высокой квалификации для поиска данных, ее ведение и сопровождение (должна быть достаточно формализованной);
- Иметь дружественный интуитивно понятный интерфейс для взаимодействия с пользователем по сети Internet;
- Обеспечение целостности и безопасности данных.

Рассмотрим эти требования более подробно.

К современному телекоммуникационному оборудованию следует отнести: концентраторы, коммутаторы, маршрутизаторы, кабели, модемы и другие устройства.

Чтобы база данных отвечала своему предназначению, необходимо осуществлять постоянную ее актуализацию. То есть сведения в базе должны регулярно обновляться и пополняться.

База данных должна быть реализована как удаленная информационная система, построенная на технологии «клиент-сервер».

В основу реализации требований к Web-интерфейсу могут быть включены следующие действия.

После запуска главной страницы сайта, обеспечивающего доступ к базе данных, должна появляться основная рабочая форма. Она должна содержать два списка: типы устройств (например, модем, концентратор) и

производители устройств (например, 3Com, Cisco). После выбора в первом списке необходимого типа телекоммуникационного оборудования загрузится страница, содержащая устройства выбранного типа с указанием производителя и модели устройства. Далее, после выбора определенного устройства появится страница с кратким описанием и ссылкой на файл с подробной информацией об устройстве.

Соответственно, после выбора во втором списке главной страницы производителя устройства, загрузится страница, содержащая устройства выбранного производителя с указанием модели. После выбора устройства должна появиться страница с кратким описанием и ссылкой на файл с подробной информацией об устройстве.

Для изучения сведений по оборудованию может быть предназначено отдельно открывающееся окно браузера для просмотра информации, в верхней части которого будет содержаться инструментальная панель. С ее помощью могут быть осуществлены такие действия, как:

- Сохранение информации;
- Вывод на печать;
- Выбор режима просмотра;
- Масштабирование;
- Выделение текстовой и графической информации;
- Копирование;
- Навигация по страницам;
- Повороты страниц;
- Поиск текста.

Одной из страниц сайта будет форма для задания условий поиска сведений в базе. Доступ к ней должен быть обеспечен из основного меню сайта (например, кнопкой «Поиск»). Для проведения поиска необходимо ввести в специальное поле искомую строку и выбрать в списке критерий

поиска. Результатом поиска должен быть список телекоммуникационного оборудования, сведения о котором содержаться в базе данных. Из этого списка можно будет перейти к подробному описанию интересующего устройства.

Администратор базы данных должен выполнять настройку базы, а также пополнять базу информацией о новом оборудовании, удалять сведения, утратившие свою актуальность.

Для этого должен быть реализован интерфейс администратора, вход в который можно осуществить путем нажатия кнопки «Вход» в основном меню. В форме авторизации администратора необходимо будет ввести имя и пароль администратора базы данных.

После успешной авторизации в окне браузера должно появляться меню администратора, с помощью которого должно быть доступно выполнение следующих действий:

- редактирование существующих записей (тип оборудования, производители оборудования и описание устройств);
- добавление новых типов оборудования;
- добавление новых производителей устройств;
- добавление новых устройств;
- смена пароля администратора базы данных.

Так как данная разработка носит информационно-справочный характер и основана на использование описательной информации, содержащей как текст, так и графический материал, а также в связи с широким распространением формата Adobe Acrobat Reader, то для хранения описания телекоммуникационных устройств целесообразно использование файлов PDF. Такой формат обеспечит достаточно полный спектр возможностей для просмотра описаний оборудования, включая масштабирование, удобную навигацию с использованием предварительного просмотра и закладок, а

также позволит использовать поиск по текстовой информации. Кроме этого можно будет скопировать необходимый текст и изображение или распечатать документ.

3. Способы построения баз данных

Современные технологии баз данных появились как результат развития в течение нескольких десятилетий способов обработки данных и управления информацией. Обработка данных развивалась от примитивных методов пятидесятых годов к сложным интегрированным системам сегодняшнего дня.

Базы данных (БД) составляют в настоящее время основу компьютерного обеспечения информационных процессов, входящих практически во все сферы человеческой деятельности.

Действительно, процессы обработки информации имеют общую природу и опираются на описание фрагментов реальности, выраженное в виде совокупности взаимосвязанных данных. Базы данных являются эффективным средством представления структур данных и манипулирования ими. Концепция баз данных предполагает использование интегрированных средств хранения информации, позволяющих обеспечить централизованное управление данными и обслуживание ими многих пользователей. При этом БД должна поддерживаться в среде ЭВМ единым программным обеспечением, называемым системой управления базами данных (СУБД). СУБД вместе с прикладными программами называют банком данных [10].

Информационные системы, использующие базы данных, позволили преодолеть ограничения файловых систем, такие как:

- Избыточность данных;
- Слабый контроль;
- Недостаточные возможности управления данными;
- Большие затраты труда программиста.

Все вопросы, связанные с эффективным ведением и использованием базы данных, решает группа специалистов, называемых администраторами

базы данных. Новые методы обращения к данным сильно упростили процесс связывания элементов данных, что в свою очередь привело к расширению возможностей работы с данными. Все эти характеристики систем управления базами данных упрощают процесс программирования и уменьшают необходимость программной поддержки.

В настоящее время процесс создания максимально мощных систем управления базами данных идет полным ходом. За несколько десятилетий последовательно появлялись системы, основанные на трех базовых моделях данных, или концептуальных методах структурирования данных: иерархической, сетевой и реляционной.

3.1. Иерархические модели

Первая информационная система, использующая базы данных, появившаяся в середине шестидесятых годов, была основана на иерархической модели. Она возникла в отличие от остальных моделей из практики, поэтому не существует исходного документа, описывающего иерархическую модель. Поскольку иерархическая модель данных не имеет стандарта, то ее изучение требует рассмотрения СУБД, применяемых на практике. Среди реализаций иерархической модели преобладает IMS (Information Management System компании IBM). Все описания иерархической модели неизменно включают словарь и соглашения IMS [10].

Однако применяются и другие иерархические системы, включая TDMS (Time-Shared Data Management System компании Development Corporation), MARK IV (Multi-Access Retrieval System компании Control Data Corporation) и System-2000 (SAS-Institute).

IMS появилась в результате совместных усилий компаний IBM и North American Aviation (позднее Rockwell) по созданию СУБД для поддержки лунного проекта Апполон – одного из самых крупных инженерных проектов того времени. Ключевым фактором при создании IMS была необходимость управления миллионами деталей, связанных друг с другом иерархическим образом. То есть, из небольших деталей были собраны более крупные узлы, которые входили в еще большие модули и т. д.

Для больших систем с планируемыми транзакциями, требующих высокого быстродействия, IMS остается вполне конкурентоспособной системой. Дополнительная причина «живучести» IMS состоит в том, что многие структуры данных естественным образом иерархичны. Например, организация может состоять из отделов (один уровень), в отделах работают сотрудники (второй уровень), которые обладают определенными специальностями (третий уровень). Хотя эту структуру данных можно

реализовать в других моделях, более мощные возможности представления этой модели могут привести к большей сложности системы (издержкам), чем требуется.

Иерархическую структуру данных задают три объекта: деревья, сегменты и поля сегментов. Хотя любую объектно-ориентированную модель можно преобразовать в иерархическую структуру данных, требования того, чтобы все записи базы данных были деревьями, может привести к повторению сегментов. Любая ситуация, в которой естественное преобразование приводит к тому, что у сегмента имеется два разных предка, требует создания двух отдельных деревьев, которым будут принадлежать эти предки.

Хотя такое повторение позволяет избавиться от определенных трудностей реализации, оно приводит к следующим негативным результатам:

1. Объем памяти расходуется неэффективно, так как сегменты повторяются;

2. Возникает возможность противоречивости данных. Если данные изменяются в одной копии сегмента и не изменяются в другой, база данных становится противоречивой.

Эта проблема может быть разрешена при помощи виртуальных сегментов и указателей. Виртуальный сегмент не содержит данных; вместо этого он содержит указатель адреса хранения данных. Когда сегмент должен дублироваться в двух или более деревьях, то действительные данные хранятся только в одном из этих деревьев. Все остальные вхождения этого сегмента данных будут содержать указатель адреса хранения действительных данных.

Главное ограничение иерархической модели состоит в том, что для многих приложений дерево не является естественной структурой данных.

Для приложений с иерархической структурой и устойчивыми транзакциями запросов иерархическая модель может оказаться вполне удовлетворительной. Тот факт, что было установлено около 7000 IMS, подтверждает данный вывод, хотя некоторые из владельцев таких систем переходят к реляционным системам.

Языковый интерфейс, предоставляемый для иерархической модели, значительно зависит от производителя системы. Это означает, что программист должен быть осведомлен о том, какие отношения должны быть предопределены в системе, а какие – нет. Надежность работы программиста будет неудовлетворительна, если он не обладает полным знанием о том, какие интегрированные требования встроены в систему. Кроме того, если предпринимается реорганизация иерархической базы данных, то она может негативно отразиться на качестве функционирования существующих программ, поскольку структура, поддерживающая одно приложение, может оказаться не лучшей для поддержания другого приложения.

3.2. Сетевые модели

Сети - естественный способ представления отношений между объектами. Они широко применяются в математике, исследованиях операций, химии, физике, социологии и других областях знаний. Сетевая архитектура применяется также и к базам данных.

Сети могут быть представлены математической структурой, которая называется направленным графом. Граф состоит из точек или узлов, соединенных стрелками или ребрами. В контексте моделей данных узлы можно представлять как типы записей данных, а ребра представляют отношения один-к-одному или один-ко-многим. Таким образом, сетевая модель данных представляет данные сетевыми структурами типов записей, связанных отношениями общности один-к-одному или один-ко-многим. Структура графа делает возможными простые представления иерархических отношений, отношений принадлежности и многих других. Более того, после того как отношение между двумя объектами установлено, становится возможным активное выполнение извлечения данных и манипуляции ими.

Конференция по языкам систем данных (Conference on Data Systems Languages, CODASYL), в которую входят представители основных поставщиков оборудования, программного обеспечения и пользователей, в начале 60-х разработала и стандартизировала язык Кобол. В конце 60-х она поручила подгруппе, названной Database Task Group (DTBG), разработать стандарты систем управления базами данных. На DTBG оказывала сильное влияние архитектура, использованная в одной из самых первых СУБД, Integrated Data Store (IDS), созданной ранее компанией General Electric. Это влияние привело к тому, что в предварительном отчете, опубликованном в 1969 году, была рекомендована сетевая модель. Этот первый отчет породил огромное количество предложений по усовершенствованию, и в 1971 году был опубликован пересмотренный отчет. Он был представлен в

Американский институт национальных стандартов (American National Standards Institute, ANSI) для возможного принятия в качестве национального стандарта СУБД.

Модель CODASYL DTBG послужила основой для разработки сетевых систем управления базами нескольких производителей. IDS (Honeywell) и IDMS (Computer Associate) две наиболее известных коммерческих реализации. Хотя сетевая модель данных в будущем может все больше уступать позиции на рынке СУБД реляционной модели данных, сегодня она эффективно служит во многих информационных системах.

В сетевой модели существует две основные структуры данных: типы записей и наборы. Типы записей определяются обычным образом как совокупности логически связанных записей. Наборы в модели DBTG выражают отношения один-ко-многим (или один-к-одному) между двумя типами записей. После того как определены тип записи владельца и тип записи члена, все определения применяются как обычно. Действительные значения в структуре данных называются экземплярами или вхождениями.

В модели DBTG возможны только простые сети, в которых все отношения имеют мощность один-к-одному или один-ко-многим. Сложные сети, включающие одно или несколько отношение много-ко-многим, не могут быть напрямую реализованы в данной модели, однако существует метод преобразования сложной сети в простую сетевую форму.

Сетевая модель DBTG особенно хорошо подходят для информационных систем, которые характеризуются:

- большим размером;
- хорошо определенными, повторяющимися запросами;
- хорошо определенными транзакциями;
- хорошо определенными приложениями.

Если все эти факторы присутствуют, то пользователи и проектировщики системы базы данных могут сосредоточить усилия на обеспечении того, чтобы приложения были написаны наиболее эффективным образом. Негативная сторона состоит в том, что непредвиденные заранее приложения в будущем могут работать плохо; они даже могут потребовать трудоемкой реорганизации структуры базы данных.

3.3. Реляционные модели

В 1970 году Е. Ф. Кодд опубликовал статью, которая изменила устоявшиеся представления о базах данных. Он выдвинул идею, что данные нужно связывать в соответствии с их внутренними логическими взаимоотношениями, а не физическими указателями.

В своей статье Кодд предложил простую модель данных, согласно которой все данные сведены в таблицы, состоящие из строк и столбцов. Эти таблицы получили название реляции, а модель стала называться реляционной. Кодд предложил пользоваться для работы с данными в таблицах двумя языками: реляционной алгеброй и реляционным исчислением. Оба языка обеспечивают работу с данными на основе логических характеристик, а не физических указателей.

Рассматривая данные с концептуальной, а не физической точки зрения, Кодд предложил еще одну идею. В реляционных системах баз данных целые файлы данных могут обрабатываться одной командой, тогда как в традиционных системах за один раз обрабатывалась одна запись. Подход Кодда чрезвычайно повысил эффективность программирования в базах данных.

Логический подход к данным сделал также возможным создание языков запросов, более доступных для пользователей, не являющихся специалистами по компьютерам. Хотя создать язык, которым могли бы пользоваться все, независимо от опыта работы с компьютером, довольно сложно, однако реляционные языки запросов сделали базы данных доступными для более широкого круга пользователей, чем раньше.

Публикация работ Кодда в начале семидесятых активизировала работу ученых и разработчиков. Результатом этой работы явилось создание различных языков реляционных баз данных, таких как Structured Query Language (SQL, язык структурированных запросов), Query Language (Quel,

язык запросов) и Query-by-Example (QBE, запросы по образцу) и других. В 1986 году SQL был принят в качестве стандарта ANSI языков реляционных баз данных. Этот стандарт обновлялся в 1989 и 1992 годах.

Основные понятия и определения для реляционной модели данных.

Количество атрибутов реляции называется степенью реляции. Никакие два атрибута реляции не могут иметь одинаковых имен. Строки реляции называются кортежами. Предполагается, что предопределенного порядка строк или кортежей не существует, и что никакие два кортежа не имеют одинаковый набор значений.

Набор всех возможных значений, которые могут принимать атрибуты, называется областью атрибута. Две области атрибутов совпадают только в том случае, если они имеют один и тот же смысл. Два атрибута с одной и той же областью не обязательно имеют одно и то же имя.

Любой набор атрибутов, однозначно определяющий каждый кортеж реляционной таблицы, называется суперключом. Ключ реляции – это минимальный набор атрибутов. Таким образом, ключ – это минимальный суперключ.

Ключ, содержащий два или более атрибута, называется составным ключом. В любой реляционной таблице может оказаться более одного набора атрибутов, которые можно выбрать в качестве ключа. Они называются потенциальными ключами. Когда один из потенциальных ключей выбран в качестве ключа реляции, его можно назвать первичным ключом. Обычно в качестве первичного ключа выбирают потенциальный ключ, которым проще всего пользоваться при повседневной работе по вводу данных. Внешний ключ – это набор атрибутов одной таблицы, являющийся ключом другой (или той же самой) таблицы. Список, в котором даются имена реляционных таблиц с перечислением их атрибутов и определений внешних ключей, называется реляционной схемой базы данных. Она является

предварительным итогом создания этапа жизненного цикла реляционной базы данных.

В реляционной модели Кодда есть несколько ограничительных условий, используемых для поддержания целостности и проверки данных в базе данных, а также для придания данным осмысленной структуры. Ограничительные условия обеспечивают логическую основу для поддержания правильных значений данных в базе данных, предупреждая ошибки при обновлении и обработке данных.

Строки реляционной таблицы представляют в базе данных элементы конкретных объектов реального мира или категорий. Ключ реляционной таблицы однозначно определяет каждую строку и, следовательно, каждый элемент категории. Таким образом, если пользователи хотят извлекать данные конкретной строки или манипулировать ими, они должны знать значение ключа этой строки. Отсюда следует, что элемент не должен записываться в базу данных до тех пор, пока значения его ключевых атрибутов не будут полностью определены.

При построении реляционных таблиц для связывания строк одной таблицы со строками другой таблицы используются внешние ключи. База данных, в которой все непустые внешние ключи ссылаются на текущие значения ключей другой реляционной таблицы, обладает целостностью на уровне ссылок.

Нормализация – процесс приведения реляционных таблиц к стандартному виду, что позволяет исключить избыточность данных или их повторение и нарушение целостности данных (противоречивость) в базе данных. В основе нормализации лежит разбиение – процесс разделения одной таблицы на несколько. Для того чтобы сделать это, можно воспользоваться нормальными формами или правилами структурирования таблиц.

Реляционная алгебра – это процедурный язык обработки реляционных таблиц. Это означает, что в реляционной алгебре используется пошаговый подход к созданию реляционных таблиц, содержащих ответы на запросы. Реляционная алгебра имеет большое значение, так как из нее происходят многие термины для понятий обработки данных, часто встречающиеся в языках баз данных.

Реляционное исчисление – непроцедурный язык. В реляционном исчислении запрос создается путем определения таблицы запроса за один шаг. Реляционная алгебра и реляционное исчисление логически эквивалентны. Это означает, что любой запрос, который можно сформулировать при помощи логического исчисления, также можно сформулировать, пользуясь реляционной алгеброй, и наоборот.

Объединение – операция реляционной алгебры, создающая теоретико-множественное объединение двух объединительно-совместимых реляционных таблиц.

Пересечение – операция реляционной алгебры, создающая пересечение в теоретико-множественном смысле двух объединительно-совместимых реляционных таблиц.

Операция произведения, имеет большое значение в качестве составной части операции соединения. Она идентична математической операции взятия декартова произведения двух множеств.

Выборка – операция реляционной алгебры, производящая отбор строк из таблицы на основании некоторого условия.

Операция выборки используется для создания реляционной таблицы из другой реляционной таблицы путем отображения только тех строк, которые удовлетворяют заданному условию.

Если операцию выборки можно представить как исключение ненужных строк, то операцию создания проекций можно представить как исключение

ненужных столбцов. Полученная в результате операции создания проекций таблица называется проекцией исходной таблицы.

Соединение – операция реляционной алгебры, связывающая таблицы. Операция соединения используется для связывания данных между таблицами.

Простой запрос – запрос, который обращается только к одной таблице базы данных.

Подзапрос – запрос внутри запроса.

В таблице приведена сравнительная характеристика способов обращения к данным для различных структур баз данных.

Способ доступа к данным	Характеристика
Файлы последовательного доступа	Записи должны обрабатываться в последовательном порядке
Файлы произвольного доступа	Поддерживают прямой доступ к конкретной записи. Сложно обращаться к нескольким записям, связанным с одной
Иерархическая база данных	Поддерживает доступ к нескольким записям, связанным с одной. Отношения между данными ограничиваются иерархическими. Зависит от предопределенных физических указателей
Сетевая база данных	Поддерживает иерархические и неиерархические отношения между данными. Зависит от предопределенных физических указателей
Реляционная база данных	Поддерживает все логические отношения между данными. Логический доступ к данным, не зависящий от физической реализации

На основе представленных описаний различных моделей можно сделать следующее выводы.

Иерархическая модель является недостаточно гибкой и может привести к неэффективному расходу памяти, возможна противоречивость данных в базе.

Сетевая модель обеспечивает высокую эффективность создаваемых на ее основе информационных систем, но требует от пользователя спецификации структур хранения, способов доступа, логических структур данных.

При реляционном подходе к построению баз данных упрощается структура физических и логических баз данных, отпадает необходимость предсказания способов доступа, появляется возможность обработки непредсказуемых запросов.

Достоинствами этого подхода являются:

- простота представления данных (в виде двумерных таблиц);
- гибкость (операции проецирования и соединения позволяют получать таблицы данных в нужной форме);
- точность (отношения по своей природе обладают более точным смыслом, чем направленные связи, и поддаются математически точным методам манипулирования);
- простота контроля секретности;
- независимость данных от прикладных программ;
- простота ведения и модификации данных.

Сопоставляя возможности рассмотренных моделей данных, можно сделать вывод о привлекательности реляционного подхода к построению базы данных.

4. Организация пользовательского интерфейса для доступа к базам данных

Многие организации, в том числе учебные заведения, используют электронные базы данных для поддержки своих рабочих процессов. Часто это системы, выполненные с использованием dbf – ориентированных средств разработки: Clipper, DBase, FoxPro, Paradox, Access. Обычно используется ряд таких баз, независимых друг от друга. Если информация, хранимая в таких БД, представляет интерес не только для непосредственных пользователей, то для ее дальнейшего распространения используются бумажные отчеты и справки, созданные базой данных.

С появлением локальных сетей, подключением таких сетей к Internet, созданием внутрикорпоративных, сетей, появляется возможность с любого рабочего места организации получить доступ к информационному ресурсу сети. Однако, при попытке использовать существующие БД возникают проблемы связанные с требованием к однородности рабочих мест (для запуска «родных» интерфейсов), сильнейшим трафиком в сети (доступ идет напрямую к файлам БД), загрузкой файлового сервера и невозможностью удаленной работы (например, командированных сотрудников или студентов заочной формы обучения).

На кафедре ОПДС разработана и используется в учебном процессе база данных «Современное телекоммуникационное оборудование». База данных содержит сведения об образцах некоторого телекоммуникационного оборудования (модемы, адаптеры, концентраторы и др.) База данных открыта для дальнейшего наполнения, редактирования и удаления сведений о телекоммуникационном оборудовании. Для реализации базы данных была выбрана СУБД Microsoft Access. Однако для работы с такой базой данных необходимо установить ее на каждый компьютер в учебном классе или выделить один компьютер в локальной сети для такой базы данных и

работать с ней как общим сетевым ресурсом. Это не всегда удобно и эффективно. Решением проблемы могло бы стать использование унифицированного интерфейса WWW для доступа к ресурсам организации.

Технология World Wide Web получила широкое распространение из-за простоты своего пользовательского интерфейса. Принцип «жми на то, что интересно», лежащий в основе гипертекста, интуитивно понятен. В технологиях WWW все ключевые понятия просматриваемого документа: слова, картинки – имеют возможность «раскрыться» новым документом, развивающим это понятие. Такой способ представления информации называется «гипертекстом», а документы, представленные в таком виде – «гипертекстовыми документами». Для описания этих документов используется специальный язык – язык описания гипертекстовых документов или HTML (HyperText Markup Language) [11].

Из этих предпосылок возникает задача преобразования накопленных данных в гипертекстовые документы WWW, задача поддержки актуальности преобразованной структуры. Другими словами, задача предоставления WWW – доступа к существующим базам данных.

Использование технологий WWW для обеспечения доступа к каким-либо информационным ресурсам подразумевает существование следующих компонентов (рис. 4.1):

1. IP-сети с поддержкой базового набора услуг по передаче данных с единой политикой нумерации и маршрутизации, работающим сервисом имен DNS.
2. Выделенного информационного сервера – WWW-сервера, обеспечивающего предоставление гипертекстовых документов через IP – сеть в ответ на запросы WWW – клиентов.

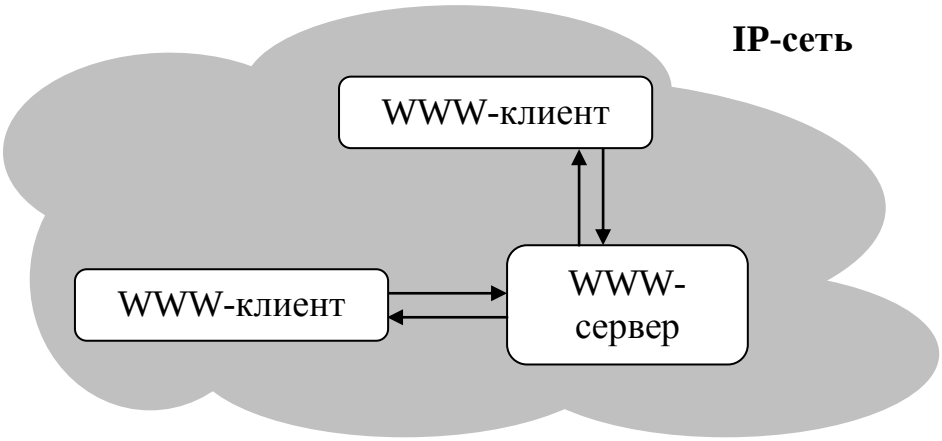

Рис. 4.1. Компоненты для использования технологи WWW

Передаваемые гипертекстовые документы оформляются в стандарте HTML – языке описания гипертекстовых документов. Эти документы могут либо храниться в статическом виде (совокупность файлов на диске), либо динамически компоноваться в зависимости от параметров запроса специальным программным обеспечением. Для динамической компоновки HTML-документов WWW-сервер использует специальным образом оформленные программы - CGI-программы (Common Gate Interface).

В состав специфики конкретной БД входят как технологические основы, такие как тип СУБД, вид интерфейсов, связи между таблицами, ограничения целостности, так и организационные решения, связанные с поддержкой актуальности баз данных и обеспечением доступа к ней.

При обеспечении WWW-доступа к существующим БД, возможен ряд путей – комплексов технологических и организационных решений. Практика использования WWW-технологии для доступа к существующим БД предоставляет широкий спектр технологических решений, по разному связанных между собой – перекрывающих, взаимодействующих и т.д. Выбор конкретных решений при обеспечении доступа зависит от специфики конкретной СУБД и от ряда других факторов, как то: наличие специалистов, способных с минимальными издержками освоить определенную ветвь

технологических решений, существование других БД, WWW-доступ к которым должен осуществляться с минимальными дополнительными затратами и т.д.

WWW–доступ к существующим базам данных может осуществляться по одному из трех основных сценариев. Ниже дается их краткое описание и основные характеристики.

1. Однократное или периодическое преобразование содержимого БД в статические документы.

В этом варианте содержимое БД просматривает специальная программа, создающая множество файлов – связных HTML-документов (рис. 4.2).

Полученные файлы могут быть перенесены на один или несколько WWW-серверов. Доступ к ним будет осуществляться как к статическим гипертекстовым документам сервера.

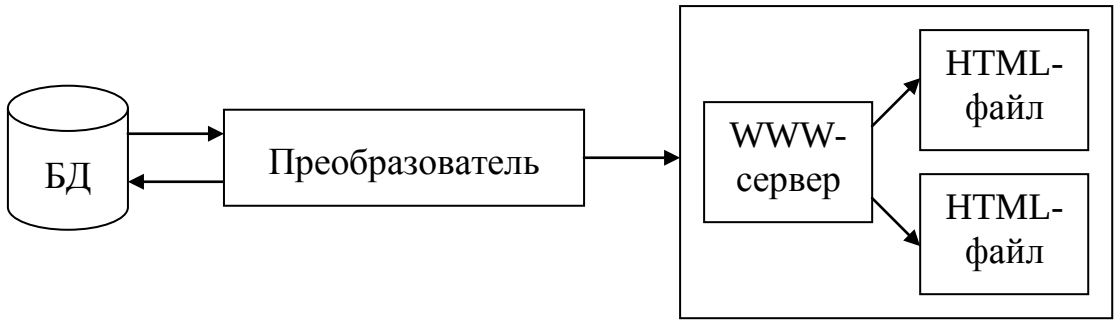

Рис. 4.2. Преобразование содержимого БД в статические документы

Этот вариант характеризуется минимальными начальными расходами. Он эффективен на небольших массивах данных простой структуры и редким обновлением, а также при пониженных требованиях к актуальности данных, предоставляемых через WWW. Кроме этого, очевидно полное отсутствие механизма поиска, хотя возможно развитое индексирование.

В качестве преобразователя может выступать программный комплекс, автоматически или полуавтоматически генерирующий статические документы. Программа-преобразователь может являться самостоятельно

разработанной программой либо быть интегрированным средством класса генераторов отчетов.

2. Динамическое создание гипертекстовых документов на основе содержимого БД.

В этом варианте доступ к БД осуществляется специальной CGI-программой, запускаемой WWW-сервером в ответ на запрос WWW–клиента. Эта программа, обрабатывая запрос, просматривает содержимое БД и создает выходной HTML-документ, возвращаемый клиенту (рис. 4.3).

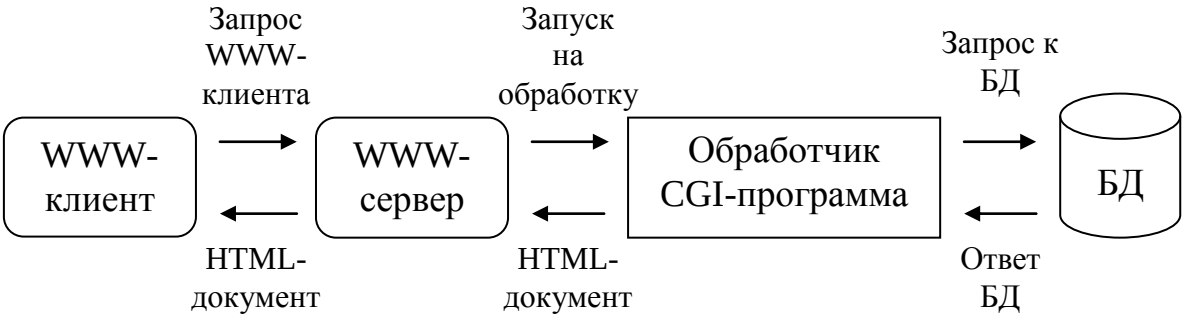

Рис. 4.3. Динамическое создание гипертекстовых документов

Это решение эффективно для больших баз данных со сложной структурой и при необходимости поддержки операций поиска. Также, если необходимо частое обновление и невозможна синхронизация преобразования БД в статические документы с обновлением содержимого. В этом варианте возможно осуществлять изменение БД из WWW-интерфейсов.

К недостаткам этого метода можно отнести большое время обработки запросов, необходимость постоянного доступа к основной базе данных, дополнительную загрузку средств поддержки БД, связанную с обработкой запросов от WWW – сервера.

Для реализации такой технологии необходимо использовать взаимодействие WWW-сервера с запускаемыми программами CGI. Выбор программных средств достаточно широк – языки программирования, интегрированные средства типа генераторов отчетов. Для СУБД со

внутренними языками программирования существуют варианты использования этого языка для генерации документов.

3. Создание информационного хранилища на основе высокопроизводительной СУБД с языком запросов SQL. Периодическая загрузка данных в хранилище из основных СУБД.

В этом варианте предлагается использование технологии, получившей название «информационного хранилища» (ИХ). Для обработки разнообразных запросов, в том числе и от WWW-сервера, используется промежуточная БД высокой производительности (рис. 4.5). Информационное наполнение промежуточной БД осуществляется специализированным программным обеспечением на основе содержимого основных баз данных (рис. 4.4).

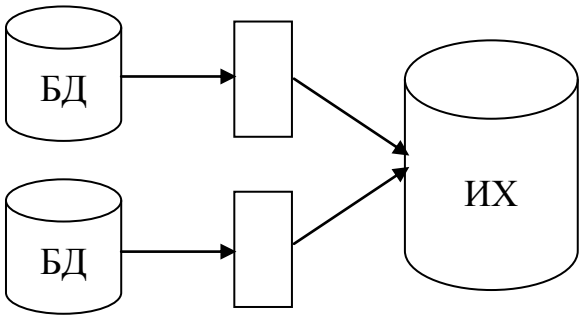

Рис. 4.4. Перегрузка данных (Этап 1)

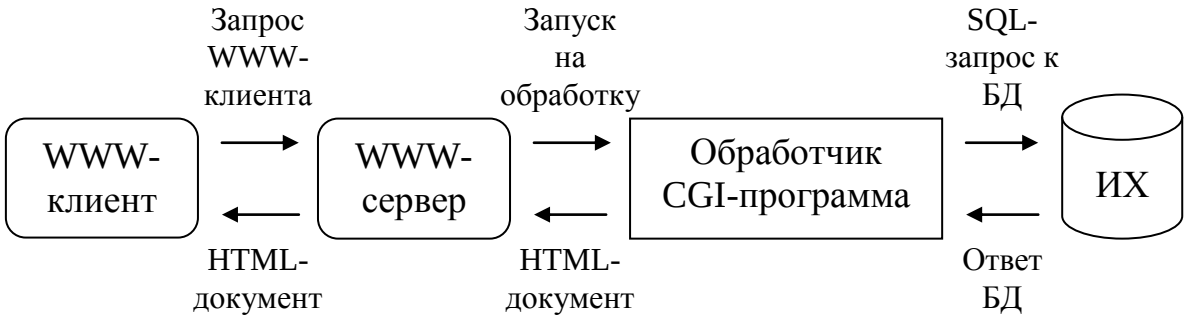

Рис. 4.5. Обработка запросов (Этап 2)

Данный вариант свободен от всех недостатков предыдущей схемы. Более того, после установления синхронизации данных информационного хранилища с основными БД возможен перенос пользовательских интерфейсов на информационное хранилище, что существенно повысит надежность и производительность, позволит организовать распределенные рабочие места.

Несмотря на кажущуюся громоздкость такой схемы, для задач обеспечения WWW-доступа к содержимому нескольких баз данных накладные расходы существенно уменьшаются.

Основой повышения производительности обработки WWW-запросов и резкого увеличения скорости разработки WWW-интерфейсов является использование внутренних языков СУБД информационного хранилища для создания гипертекстовых документов.

Для загрузки содержимого основной БД в информационное хранилище могут использоваться все перечисленные решения (языки программирования, интегрированные средства), а также специализированные средства перегрузки, поставляемые с SQL-сервером и продукты поддержки информационных хранилищ.

5. Выбор методов и инструментальных средств для реализации базы данных и Web-интерфейса

Выбор той или иной аппаратной платформы и конфигурации определяется рядом общих требований, которые предъявляются к характеристикам современных вычислительных систем [11].

К ним относятся:

- Отношение стоимость/производительность;
- Надежность и отказоустойчивость;
- Масштабируемость;
- Совместимость и мобильность программного обеспечения.

Для создания базы данных и Web-интерфейса для удаленного доступа к БД можно использовать различные инструментальные средства. Так же необходимо выбрать аппаратную платформу для Web-сервера, операционную систему, программное обеспечение Web-сервера, систему управления базами данных и язык программирования или создания сценариев.

Выбор этих компонентов взаимосвязан. Например, не все операционные системы могут работать на любом оборудовании, не все языки создания сценариев могут обеспечить подключение ко всем базам данных и т. д.

5.1. Web-сервер

Web-сервер – это программное обеспечение, отвечающее за прием запросов браузеров, поиск указанных файлов и возращение их содержимого.

В настоящее время разработано и широко применяется несколько десятков программ, реализующих эти функции. Практически для каждой операционной системы существует целый ряд таких программ. Некоторые из них являются независимыми от операционной системы и могут использоваться одновременно в разных ОС. Но в подавляющем большинстве Web-серверы ориентированы на применение только в одной операционной системе. Среди них есть как коммерческие программы, так и распространяемые бесплатно. Иногда функции Web-сервера являются только частью функций, заложенных разработчиками в программу. Кроме минимального набора выполняемых задач, определяющих основные функции Web-сервера, большинство программ содержит в себе много дополнительных возможностей. К ним относятся ограничение прав доступа к отдельным документам, возможность криптографической защиты передаваемых и принимаемых данных, создания на одном компьютере нескольких Web-серверов с разными доменными именами, использования нестандартных портов входа для сервера. Кроме этого от Web-серверов часто требуется поддержка работы с системами управления базами данных и языками Perl, Java, PHP Hypertext Preprocessor. Кроме набора функций, существенное влияние на выбор Web-сервера оказывают простота настройки и удобство в администрировании. Немаловажное значение для высоко посещаемых серверов имеет также быстрота ответа программы на запрос клиента. На сегодняшний день бесспорным лидером среди Web-серверов является бесплатно распространяемый сервер Apache. В пятерку лидеров входят также серверы Microsoft Internet Information Server, Netscape и др. [3].

5.2. Обзор серверных программ для различных ОС

Сегодня выпускаются Web-серверы для всех основных платформ, в том числе различных версий UNIX, Windows NT, IntranetWare фирмы Novell (дополнительный компонент NetWare 4.x), OS/2 Warp, Mac OS и даже Windows 95. Web-серверы стали доступными для массового пользователя.

Функции Web-серверов не ограничиваются пересылкой статических HTML-страниц. Java и сопутствующие языки составления сценариев представляют собой идеальную платформу разработки для Web. Ряд Web-серверов предусматривают собственные API (Application Programming Interface), а некоторые из них снабжены широко известным интерфейсом Netscape Server API (NSAPI).

Наибольшее распространение среди этих серверов получил метод программирования для Web с применением языков сценариев. Фирмы Microsoft и Netscape включили в свои пакеты даже объектно-ориентированные инструменты для быстрой разработки программ (Rapid Applications Development – RAD), рассчитанные на серьезных разработчиков.

Web-серверы все чаще выпускаются в виде комплексов функциональных средств, встраиваемых в ОС.

Благодаря Web-браузерам для администрирования сервера вовсе не обязательно всегда находиться в непосредственной близости от него. В большинстве продуктов имеются функции дистанционного администрирования через Web-браузер.

Чем больше число Web-узлов, участвующих в обмене конфиденциальной информацией, тем острее необходимость в надежной защите и шифровании данных. Самая распространенная форма обеспечения безопасности, применяемая на Web-серверах, - простая аутентификация, во время которой каждый пользователь должен сообщить свой идентификатор и пароль. Средствами для базовой аутентификации снабжены все

рассмотренные серверы. Разработчики некоторых серверов пошли дальше, позволив ограничивать доступ по IP-адресу и имени узла.

Для защиты от непрошеных посетителей можно подвергнуть информацию процедуре шифрования. На Web-серверах для шифрования данных служит протокол защиты на уровне гнезд – Secure Sockets Layer (SSL). Для организации защищенного, шифрованного канала связи между сервером и браузером по протоколу SSL выполняется проверка подлинности сертификата. Учреждения уполномоченные выдавать сертификаты, скажем VeriSign, за плату сертифицируют серверы при помощи протокола SSL.

5.3. Сервер баз данных MySQL

Уже довольно давно развитые СУБД основываются на архитектуре «клиент-сервер». При этой организации наиболее трудоемкие операции над базами данных выполняются на выделенном компьютере-сервере, который должен быть достаточно мощным и обладать соответствующим набором ресурсов основной и внешней памяти. Структура Internet/Intranet приложений имеет много общего с традиционной платформой «клиент-сервер». Правильнее говоря, World Wide Web (WWW) также основывается на клиент-серверной архитектуре. В самом деле, Web-браузер является типичным клиентским front-end'ом, основное отличие которого от клиентских мест, построенных с помощью Visual C++, Visual Basic, Visual FoxPro и других средств разработки, состоит в более гибко настраиваемой функциональности, которая может определяться даже во время выполнения программы. При этом не требуется ни перекомпиляции, ни переустановки модулей, что уже само по себе является нетривиальной задачей в больших и сложных клиент-серверных системах.

Первоначально браузеры использовались только как средства форматирования статического текста. Однако активно развивающиеся Web-сайты вскоре перестали довольствоваться простыми публикациями.

Типичный интерфейс клиентского приложения предполагает заполнение клиентом некоторой формы с различными элементами управления, отправку соответствующего запроса на сервер и прием результатов обработки. Таким образом, на первый план были выдвинуты принципы динамического взаимодействия браузера и Web-сервера внутри сессии, что заставило задуматься как об активной роли браузера, так и о расширении функциональности сервера по сравнению с простым хранением и пересылкой HTML-документов.

Многие фирмы-производители программных продуктов выпускают или разрабатывают средства публикации баз данных в Internet-сетях. Основные функции таких программных продуктов данного вида таковы:

- Обеспечить отображение интерфейса пользователя в формате HTML для отображения программой просмотра клиента, в частности Internet-браузерами;
- Обеспечить формирование запросов к базе данных наиболее простыми для неподготовленного пользователя средствами;
- Обеспечить аутентификацию пользователя (для разграничения доступа);
- Обеспечить обработку запроса и возврат результата в HTML формате для отображения программой просмотра пользователя.

При этом необходимо помнить о защите передаваемой по сети информации и о привлекательности и понятности интерфейса.

Для серийно выпускаемых программных продуктов данного типа характерны высокая стоимость самого программного обеспечения (ПО), аппаратной платформы, самой СУБД. К достоинствам можно отнести хорошую документированность, наличие технической поддержки, низкую стоимость дальнейших обновлений программного обеспечения.

Можно упомянуть следующие СУБД:

SQL Server фирмы Microsoft (требует мощного сервера баз данных под управлением ОС Microsoft Windows NT server).

Sybase System фирмы Sybase (требует мощного UNIX- сервера).

Informix фирмы Informix Software (также требует мощного UNIX-сервера).

Progress фирмы Progress Software (работает на той же аппаратной платформе, что и два предыдущих).

InterBase фирмы Borland (есть вариант для Windows NT и для UNIX).

Кроме вышеперечисленных достоинств можно также отметить хорошую масштабирумость (наращиваемость), устойчивость в работе, защиту от несанкционированного доступа и мощность этих программных продуктов. Всё это ПО очень хорошо подходит для работы с проектами различного масштаба в области баз данных.

Выбор реализации того или иного решения поставленной задачи сопряжен, прежде всего, с техническими показателями компьютера-сервера, а так же с операционной системой, установленной на нём.

MySQL - очень быстрая, надежная система управления реляционными базами данных (СУРБД). База данных позволяет эффективно хранить, искать, сортировать и получать данные. Сервер MySQL управляет доступом к данным, позволяя работать с ними одновременно нескольким пользователям, обеспечивает быстрый доступ к данным и гарантирует предоставление доступа только имеющим на это право пользователям. Следовательно, MySQL является многопользовательским, многопотоковым сервером. Он применяет SQL (Structured Query Language - язык структурированных запросов), используемый по всему миру стандартный язык запросов в базы данных. MySQL появился на рынке в 1996 г., но его разработка началась еще в 1979 г. [6].

В настоящее время пакет MySQL доступен как программное обеспечение с открытым исходным кодом, но в случае необходимости можно получить и коммерческие лицензии.

К конкурентам MySQL, помимо прочих, относятся PostgreSQL, Microsoft SQL Server и Oracle.

MySQL обладает многими преимуществами, в том числе:

- Высокая производительность;
- Низкая стоимость;
- Простота конфигурирования и изучения;

- Переносимость и доступность исходного кода.

1. Производительность.

MySQL работает очень быстро. Результаты сравнительных тестов производительности, выполненных фирмой-изготовителем, можно посмотреть на странице http://web.mysql.com/benchmark.html. Многие из этих сравнительных тестов показывают, что MySQL работает на порядок быстрее конкурирующих продуктов.

2. Низкая стоимость.

Пакет MySQL доступен бесплатно в соответствии с лицензией на программное обеспечение с открытым исходным кодом или, если это необходимо для приложения, за небольшую сумму можно приобрести коммерческую лицензию.

3. Простота использования.

В большинстве современных баз данных используется SQL. Если ранее программист работал с другими СУРБД, переход к этой системе не должен вызывать каких-либо затруднений. Установка MySQL столь же проста, как и установка многих аналогичных продуктов.

4. Переносимость.

MySQL может использоваться в среде многих различных систем UNIX, а также в среде Microsoft Windows.

5. Исходный код.

Исходный код MySQL можно выгружать и изменять.

5.4. Выбор методов доступа к базе данных

Сравним способы реализации доступа к базе данных. Схематически можно представить взаимодействие базы данных и Web-сервера так:

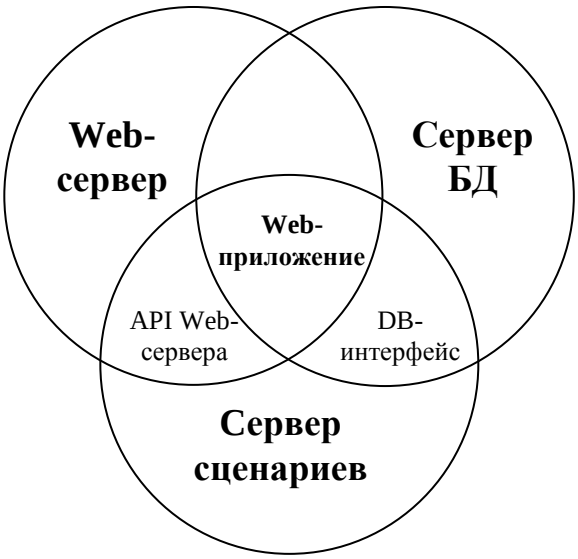

Рис. 5.1. Взаимодействие Web-приложения, базы данных и Web-сервера

Первым способом стали приложения Common Gateway Interface (CGI). Спецификация CGI позволяет браузеру вызвать тот или иной исполняемый модуль или скрипт на Web-сервере, который может обратиться с запросом к базе данных, построить в HTML-кодах страницу результатов и передать ее обратно Web-серверу. Затем Web-сервер отсылает результаты браузеру. CGI-приложения могут содержать вызовы других программных (написанных, например, на C++) или командных (.bat, .cmd) файлов. С помощью CGI-скриптов, а точнее на языке Perl (Practical Extraction and Reporting Language), построено немало интерактивных Web-приложений.

5.5. PHP: Hypertext Preprocessor

PHP — это серверный язык создания сценариев, разработанный специально для Web. В HTML-страницу можно внедрить код PHP, который будет выполняться при каждом ее посещении. Код PHP интерпретируется Web-сервером и генерирует HTML или иной вывод, наблюдаемый посетителем страницы.

Первоначально PHP являлся простой небольшой CGI оболочкой, написанной на Perl, чтобы исключить запуск Perl при каждом обращении к серверу в стандартном обращении CGI, и использовался для маленьких Internet-страниц. Позднее был встроен инструмент для включения SQL в Web-страницы. Это была CGI-оболочка, которая анализирует запросы SQL и облегчает создание форм и таблиц, основанных на этих запросах.

PHP/FI версии 2.0 - полная перезапись из этих двух пакетов, объединенных в одиночную программу. Это простой язык программирования, внедренный внутрь HTML файлов. PHP/FI сегодня используется больше для создания целых Web-серверов, чем для малых домашних страниц. Модуль устраняет потребность в многочисленных малых cgi программах на Perl, позволяя поместить простые скрипт-программы непосредственно в HTML файлы. Встроенная поддержка различных баз данных делает тривиальной разработку Web-страниц с доступом к базам данных.

Первоначально PHP являлось сокращением от Personal Home Page (Персональная начальная страница), но затем это название было изменено и теперь означает PHP Hypertext Preprocessor (Препроцессор гипертекста PHP).

PHP можно использовать как эффективный модуль сервера Apache. С появлением новой версии PHP можно устанавливать и в виде модуля ISAPI для Internet Information Server компании Microsoft.

PHP обладает множеством преимуществ по сравнению с Perl, Active Server Pages (ASP) от Microsoft, Java Server Pages (JSP) и Allaire Cold Fusion [4].

К таким приемуществам относятся:

- Высокая производительность;
- Наличие интерфейсов к различным системам баз данных;
- Встроенные библиотеки для выполнения многих общих задач, связанных с Web;
- Низкая стоимость;
- Простота изучения и использования;
- Переместимость;
- Доступность исходного кода.

Эти преимущества более подробно рассматриваются далее.

1. Производительность.

PHP исключительно эффективен. Используя единственный недорогой сервер, можно обслуживать миллионы обращений в день.

2. Интеграция с базами данных.

PHP обладает встроенной связностью со многими системами баз данных. В дополнение к MySQL, в числе прочих можно непосредственно подключаться к базам данных PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase и Sybase.

Используя ODBC - Open Database Connectivity Standard (Стандарт открытого интерфейса связи с базами данных), можно подключаться к любой базе данных, для которых существует ODBC-драйвер. Это распространяется на продукты Microsoft и многих других компаний.

3. Встроенные библиотеки.

Поскольку PHP был разработан для использования в Web, он имеет множество встроенных функций для выполнения многочисленных полезных

задач, связанных с Web. С его помощью можно «на лету» генерировать GIF-изображения, подключаться к другим сетевым службам, отправлять сообщения электронной почты и генерировать PDF-документы.

4.Стоимость.

Пакет PHP является бесплатным. Наиболее новую версию можно в любой момент совершенно бесплатно загрузить с http://www.php.net.

5.Изучение PHP.

Синтаксис PHP основывается на других языках программирования, в первую очередь на C и Perl. Если программист знаком с C, Perl или C-подобным языком, таким как C++ или Java, то он сможет эффективно использовать PHP.

6. Переносимость.

Пакет PHP можно использовать под управлением различных операционных систем. Код PHP можно создавать в среде бесплатных Unix-подобных операционных систем (Linux, FreeBSD), коммерческих версий Unix типа Solaris и IRIX и различных версий Microsoft Windows.

Как правило, программы будут работать без каких-либо изменений в различных средах с установленным PHP.

7. Исходный код.

Пользователь имеет доступ к исходному коду PHP. В отличие от коммерческих закрытых программных продуктов, если нужно что-либо изменить или добавить в этом языке, то это всегда можно сделать.

5.6. Архитектура Web-баз данных

Основная операция Web-сервера проиллюстрирована на рис. 5.2. Эта система состоит из двух объектов: Web-браузера и Web-сервера. Между ними должен существовать канал связи. Web-браузер посылает запрос на сервер, сервер отсылает обратно ответ. Для сервера, отсылающего обычные статические страницы, такая архитектура подходит. Архитектура же сайта, который включает в себя базу данных, несколько сложнее.

Рис. 5.2

Базовая архитектура Web-баз данных включает в себя Web-браузер, Web-сервер, сценарный механизм и сервер баз данных (рис. 5.3).

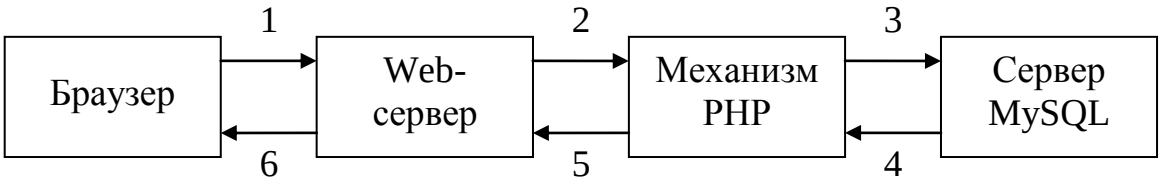

Рис. 5.3

Типичная транзакция Web-базы данных состоит из шести этапов:

1. Web-браузер пользователя отправляет HTTP-запрос определенной Web-страницы, используя HTML-форму. Пусть страница с результатами поиска называется results.php.

2. Web-сервер принимает запрос на results.php, получает файл и передает его механизму PHP на обработку.

3. Механизм PHP начинает синтаксический анализ сценария. В сценарии присутствует команда подключения к базе данных и выполнения запроса в

ней. PHP открывает соединение с сервером MySQL и отправляет необходимый запрос.

4. Сервер MySQL принимает запрос в базу данных, обрабатывает его, а затем отправляет результаты поиска обратно в механизм PHP.

5. Механизм PHP завершает выполнение сценария, форматируя результаты запроса в виде HTML, после чего отправляет результаты в HTML-формате Web-серверу.

6. Web-сервер пересылает HTML в браузер, с помощью которого пользователь просматривает результаты поиска.

Процесс этот, как правило, не зависит от того, какой сценарный механизм и какой сервер баз данных используется. Часто программное обеспечение Web-сервера, механизм PHP и сервер баз данных находятся на одной машине. Сервер базы данных может работать и на другой машине. Это делается из соображений безопасности, увеличения объема или разделения потока. С точки зрения перспектив развития оба варианта одинаковы, однако в плане производительности второй вариант может оказаться более предпочтительным.

Сравнивая возможности приведенных инструментальных средств, для реализации базы данных «Современное телекоммуникационное оборудование» была выбрана СУБД MySQL.

6. Разработка структуры, информационное наполнение и описание сценариев работы с базой данных

На первом этапе создания базы данных, прежде чем создавать таблицы и экранные формы, необходимо задать структуру базы данных. Хорошая структура базы данных является основой для создания адекватной требованиям, эффективной базы данных.

Ниже приведены основные этапы проектирования базы данных:

1. Определение цели создания базы данных;
2. Определение таблиц, которые должна содержать база данных;
3. Определение необходимых в таблице полей;
4. Задание индивидуального значения каждому полю;
5. Определение связей между таблицами;
6. Обновление структуры базы данных;
7. Добавление данных и создание других объектов базы данных.

Рассмотрим эти этапы более подробно.

1. Определение цели создания базы данных.

На первом этапе проектирования базы данных необходимо определить цель создания базы данных, основные ее функции и информацию, которую она должна содержать. То есть нужно определить основные темы таблиц базы данных и информацию, которую будут содержать поля таблиц.

База данных должна отвечать требованиям тех, кто будет непосредственно с ней работать. Для этого нужно определить темы, которые должна покрывать база данных, отчеты, которые она должна выдавать, проанализировать формы, которые в настоящий момент используются для записи данных, сравнить создаваемую базу данных с хорошо спроектированной, подобной ей базой.

2. Определение таблиц, которые должна содержать база данных.

Одним из наиболее сложных этапов в процессе проектирования базы данных является разработка таблиц, так как результаты, которые должна выдавать база данных не всегда дают полное представление о структуре таблицы.

При проектировании таблиц сначала лучше разработать структуру на бумаге. При этом рекомендуется руководствоваться следующими основными принципами:

- Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами. Когда определенная информация хранится только в одной таблице, то и изменять ее придется только в одном месте. Это делает работу более эффективной, а также исключает возможность несовпадения информации в разных таблицах;
- Каждая таблица должна содержать информацию только на одну тему. Сведения на каждую тему обрабатываются намного легче, если содержатся они в независимых друг от друга таблицах.

3. Определение необходимых в таблице полей.

Каждая таблица содержит информацию на отдельную тему, а каждое поле в таблице содержит отдельные сведения по теме таблицы. При разработке полей для каждой таблицы необходимо помнить:

- Каждое поле должно быть связано с темой таблицы;
- Не рекомендуется включать в таблицу данные, которые являются результатом выражения;
- В таблице должна присутствовать вся необходимая информация;
- Информацию следует разбивать на наименьшие логические единицы.

4. Задание индивидуального значения каждому полю.

Для связи данных из разных таблиц, каждая таблица должна содержать поле или набор полей, которые будут задавать индивидуальное значение каждой записи в таблице. Такое поле или набор полей называют первичным ключом.

5. Определение связей между таблицами.

После распределения данных по таблицам и определения ключевых полей необходимо выбрать схему для связи данных в разных таблицах. Для этого нужно определить связи между таблицами.

6. Обновление структуры базы данных.

После проектирования таблиц, полей и связей необходимо еще раз просмотреть структуру базы данных и выявить возможные недочеты. Желательно это сделать на данном этапе, пока таблицы не заполнены данными.

Для проверки необходимо создать несколько таблиц, определить связи между ними и ввести несколько записей в каждую таблицу, затем посмотреть, отвечает ли база данных поставленным требованиям.

7. Добавление данных и создание других объектов базы данных.

Если структуры таблиц отвечают поставленным требованиям, то можно вводить все данные. Затем можно создавать любые запросы и экранные формы.

6.1. Проектирование базы данных

В результате проектирования выяснилось, что в базе данных необходимо реализовать реляционную модель, реализующую, однако, некоторую иерархическую структуру, поскольку для каждого типа телекоммуникационного оборудования существуют отдельные модели оборудования. В итоге были разработаны следующие структуры таблиц.

Таблица 6.1. firms («Фирмы») – справочник наименований производителей оборудования.

Поле	Тип	Размер	Комментарии
firmid	Счетчик	10	Первичный ключ
firmname	Текстовый	50	Наименование производителя

Таблица 6.2. types («Типы») – справочник наименований типов телекоммуникационного оборудования.

Поле	Тип	Размер	Комментарии
typeid	Счетчик	10	Первичный ключ
typename	Текстовый	40	Наименование типа оборудования

Таблица 6.3. devices («Устройства») – таблица устройств.

Поле	Тип	Размер	Комментарии
devid	Счетчик	10	Первичный ключ
typeid	Целый	10	Код типа оборудования
firmid	Целый	10	Код производителя оборудования
devname	Текстовый	10	Наименование устройства
description	Текстовый	255	Краткое описание устройства
descfile	Текстовый	255	Содержимое файла описания

Таблица 6.4. admin («Сервис») – используется для хранения паролей и имен пользователей, являющихся администраторами базы данных.

Поле	Тип	Размер	Комментарии
username	Текстовый	16	Имя администратора
password	Текстовый	16	Пароль администратора

Были установлены следующие связи:

types – devices: 1-∞ (один-ко-многим), то есть для каждого типа может существовать несколько устройств.

firms – devices: 1-∞ (один-ко-многим), то есть каждый производитель может выпускать несколько моделей телекоммуникационных устройств.

С таблицей admin связи отсутствуют в связи с тем, что она является дополнительным «хранилищем» данных для сохранения всевозможных настроек базы данных.

При установлении связей была включена возможность каскадного обновления и удаления данных для обеспечения их целостности. Таким образом, при удалении записи в справочной таблице удаляются все соответствующие записи в связанных таблицах.

В результате установления необходимых связей между таблицами была разработана структура базы данных, которая представлена на рис. 6.1.

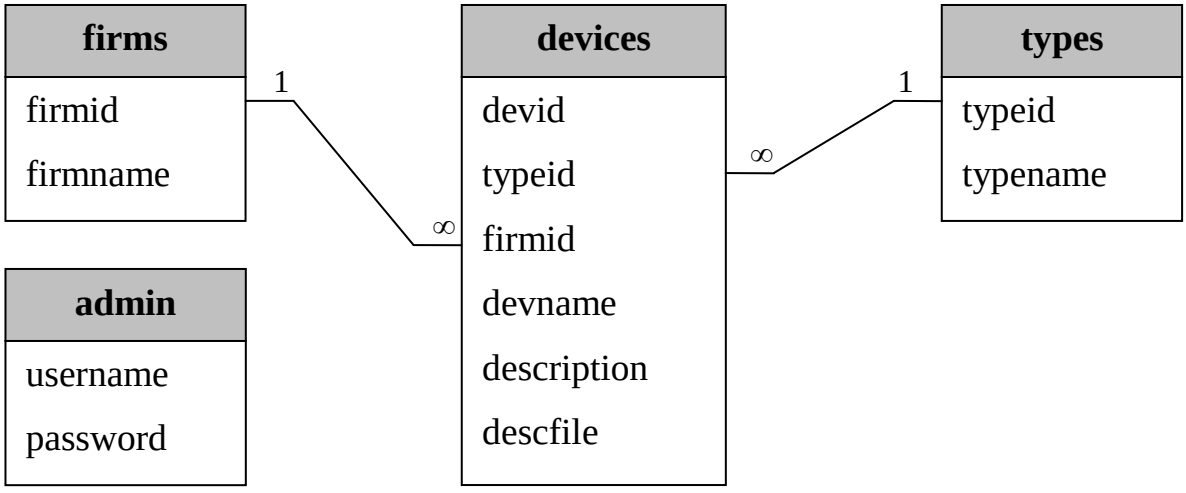

Рис. 6.1. Схема таблиц и связей базы данных

6.2. Выбор формата файла для описания телекоммуникационного оборудования

Файлы, выполненные в различных текстовых и иных процессорах (формат *.doc, *.rtf, *.ppt и др.), предназначены не для просмотра стандартными программами Internet-браузерами, а для получения их по сети и копирования на дисках пользователей. Эти форматы более привычны для значительной части пользователей и дают более широкие возможности для форматирования (за исключением, может быть, чисто текстовых форматов, где форматирование сводится к расстановке пробелов и пустых строк). Упрощены и возможности восприятия текста на русском и практически на любом из европейских языков. Однако исчезают все преимущества гипертекста, хотя возможна организация гиперссылок как внутри одного документа, так и с загрузкой других документов, а также гиперссылки на сетевые (электронные) адреса.

Целесообразно применять данные форматы для больших файлов с преимущественно линейной организацией материала (например, художественные произведения). Файлы публикаций в данных форматах могут иметь значительный объем, и поэтому рекомендуется уплотнять («сжимать») их утилитами сжатия (архиваторами), имеющими значительный коэффициент сжатия (например, pkzip или arj).

Файлы в формате программы Adobe Acrobat (PDF-формат) являются межплатформенным с полным издательским контролем отображаемого документа. PDF-файлы готовятся на основе двоичных Postscript-файлов. Следовательно, формат PDF может создаваться из документов любого приложения. Достаточно установить драйвер Postscript-принтера и вывести двоичный файл печати на этот принтер. PDF-файлы имеют те же самые преимущества, что и у предыдущей группы форматов и тот же коренной недостаток: развитого аппарата гипертекстовых ссылок. Дополнительное

преимущество PDF-файлов состоит в их полной независимости от каких бы то ни было кодировок и от наличия или отсутствия каких-либо шрифтов на конкретном компьютере. Это достигается за счет включения в них используемых шрифтов, поэтому резко возрастает необходимость в дополнительном дисковом пространстве и увеличивается время их передачи по сети из-за их значительного объема.

Достоинством данного формата является практически полная невозможность, их редактирования и копирования содержащегося в них текста. Этот формат часто применяется для создания электронных версий периодических изданий (газет и журналов), а также для естественно-научных изданий, содержащих табличный, формульный и иллюстративные материал.

Еще одним важным его достоинством является достаточная простота изготовления документов данного формата. При наличии соответствующего драйвера необходимо лишь «распечатать» информацию из любой программы на устройстве PDF writer. При этом сразу создается файл, готовый для дальнейшей работы.

Так как данная разработка носит информационно-справочный характер и основана на использование описательной информации, содержащей как текст, так и графический материал, а также в связи с широким распространением формата Acrobat Reader, то для хранения описания телекоммуникационных устройств используются данные из файлов PDF.

6.3. Разработка программных модулей

Для построения Web-сайта были реализованы следующие функциональные возможности:

- Интерактивный каталог оборудования с разбивкой по категориям и производителям;
- Механизм поиска информации в БД;
- Интерфейс администрирования.

Информационная система должна отвечать следующим требованиям:

- Необходимо отыскать способ подключения базы данных к браузеру пользователя;
- Пользователи должны иметь возможность выбирать позиции из базы данных, разбитые по категориям, с целью просмотра информации по определенному устройству, а также осуществлять быстрый поиск по ключевым словам;
- Администратор должен иметь возможность добавления, удаления и редактирования информации как о самих устройствах, так и о типах и производителях оборудования. Также должна быть реализована возможность добавления на сайт файлов описания устройств посредством Web-интерфейса.

Из перечисленных выше требований следует, что существуют два основных представления системы: пользовательское и администраторское. С учетом необходимых функциональных возможностей созданы две схемы системы - по одной для каждого представления (рис. 6.2 и 6.3).

Система в представлении пользователя дает возможность просматривать оборудование по типам, производителям и сведения о нем, а в представлении администратора - позволяет выполнять добавление, редактирование и удаление устройств, типов и производителей оборудования.

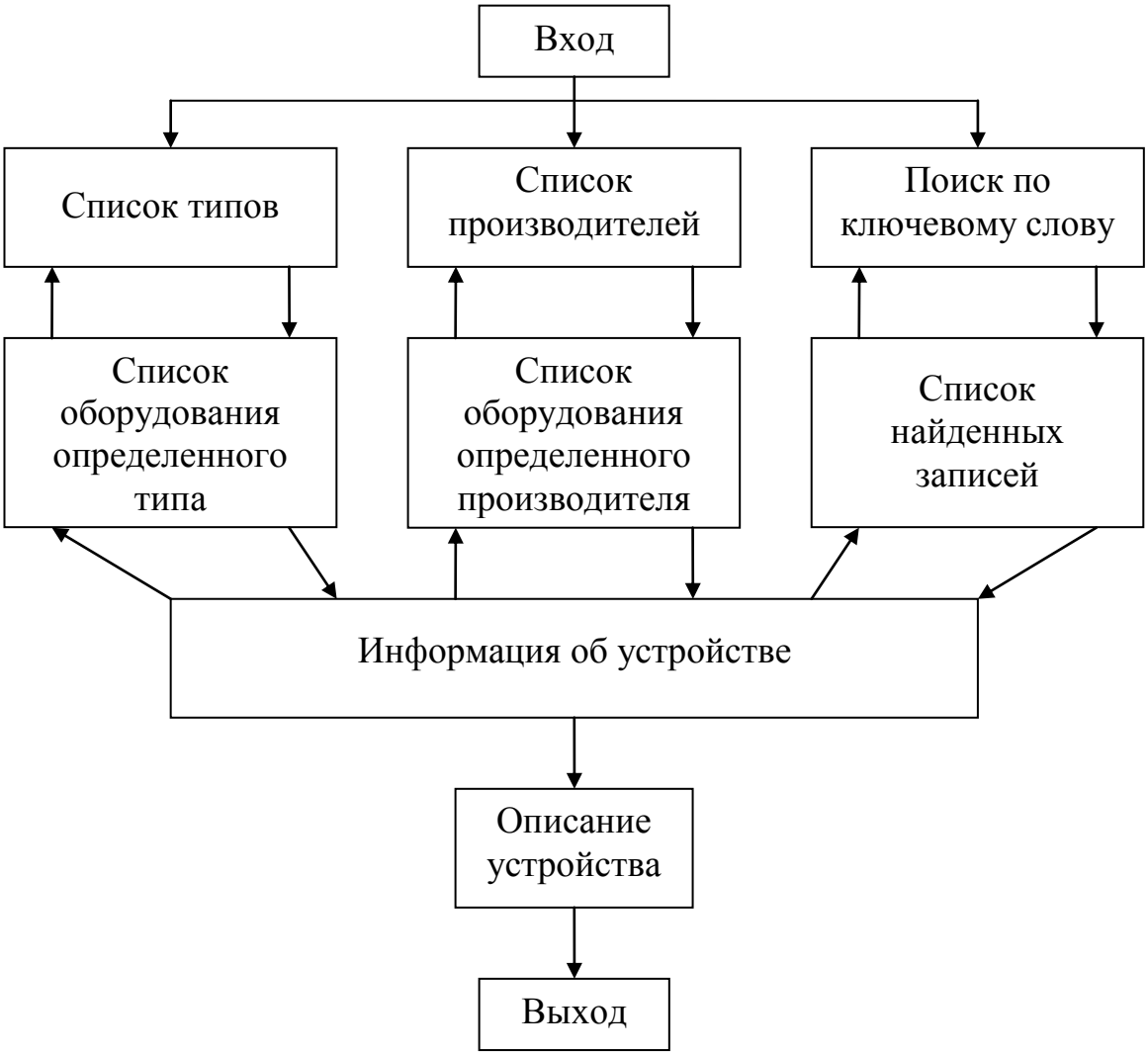

Рис. 6.2. Система в представлении пользователя

На рис. 6.2 показаны главные ссылки между сценариями в пользовательской части сайта. Пользователь сначала открывает главную страницу, в которой перечислены все типы и производители оборудования. Отсюда можно перейти к определенному типу или производителю оборудования, а затем к информации по отдельному устройству.

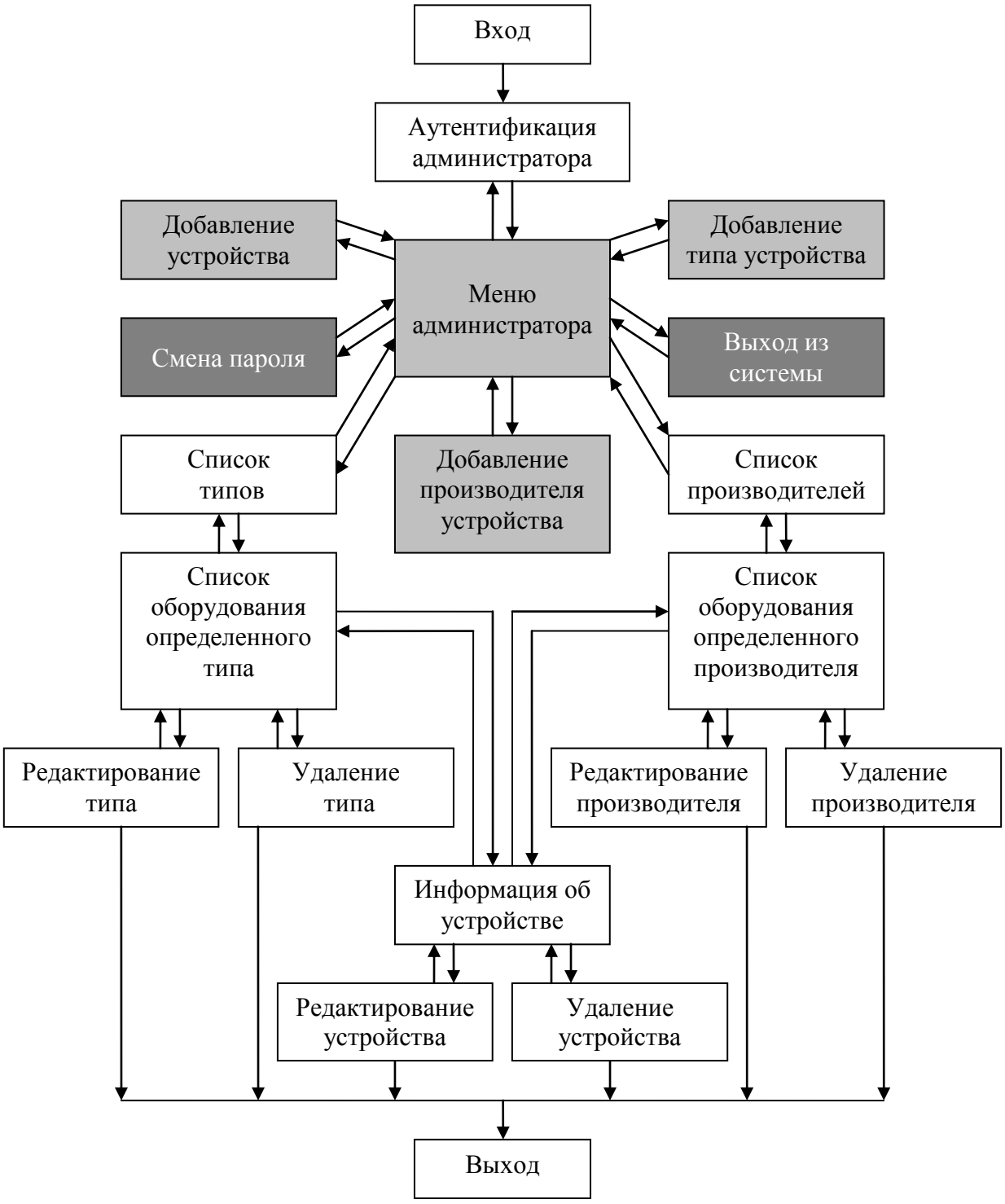

Рис. 6.3. Система в представлении администратора

На рис. 6.3 показан интерфейс администратора. В нем больше сценариев, но не особенно много нового кода. Эти сценарии позволяют администратору входить в систему и добавлять новые устройства, типы и производителей оборудования.

Простейший способ реализовать редактирование и удаление устройств, типов оборудования и производителей - отобразить для администратора несколько отличную версию интерфейса пользователя сайта. Администратор будет иметь возможность не только просматривать типы и производителей оборудования, но и переходить к определенному устройству, типу оборудования или производителю, а затем редактировать или удалять запись. Создание сценариев, одновременно пригодных для обычных пользователей и администраторов, позволяет экономить время и трудозатраты.

Ниже перечислены два основных кодовых модуля для приложения:

- Каталог;
- Администрирование.

Необходимо также создавать набор библиотек функций API-интерфейса. Объединение фрагментов кода, отвечающего за вывод HTML-содержимого в одну библиотеку, соответствует принципу разделения логики и содержимого и упрощает чтение и сопровождение кода.

Весь код данного Web-интерфейса содержится на сопровождающей дискете.

Таблица 6.5. Перечень файлов реализованного Web-приложения

Имя	**Молуль**	**Описание**
about.php	Каталог	Краткая информация о пректе.
admin.php	Администрирование	Главное меню администрирования.
admin_fns.php	Функции	Набор функций, используемых сценариями администрирования.
change_password.php	Администрирование	Изменяет пароль администратора.
change_password_form.php	Администрирование	Форма, позволяющая администратору изменять свой пароль.

Продолжение табл. 6.5.

data_valid_fns.php	Функции	Набор функций проверки допустимости вводимых данных.
db_fns.php	Функции	Набор функций для подключения к базе данных sto.
delete_device.php	Администрирование	Удаляет устройство из базы данных.
delete_firm.php	Администрирование	Удаляет производителя оборудования из базы данных.
delete_type.php	Администрирование	Удаляет тип оборудования из базы данных.
device_fns.php	Функции	Набор функций хранения и извлечения данных о устройствах.
edit_device.php	Администрирование	Обновление информации о устройстве в базе данных.
edit_device_form.php	Администрирование	Форма, позволяющая администратору редактировать информацию о устройстве.
edit_firm.php	Администрирование	Обновляет производителя оборудования в базе данных.
edit_firm_form.php	Администрирование	Форма, позволяющая администратору редактировать производителя оборудования.
edit_type.php	Администрирование	Обновляет тип оборудования в базе данных.
edit_type_form.php	Администрирование	Форма, позволяющая администратору редактировать тип оборудования.
index.php	Каталог	Титульная страница сайта. Отображает списки типов и производителей оборудования.
insert_device.php	Администрирование	Добавляет новое устройство в базу данных.

Продолжение табл. 6.5.

insert_device_form.php	Администрирование	Форма, позволяющая администратору добавлять в систему новое устройство.
insert_firm.php	Администрирование	Вставляет нового производителя оборудования в базу данных.
insert_firm_form.php	Администрирование	Форма, позволяющая администратору добавлять в базу данных нового производителя оборудования.
insert_type.php	Администрирование	Вставляет новый тип оборудования в базу данных.
insert_type_form.php	Администрирование	Форма, позволяющая администратору добавлять в базу данных новый тип оборудования.
links.php	Каталог	Отображает ссылки на другие ресурсы в сети Internet.
login.php	Администрирование	Позволяет администратору входить в систему для внесения изменений.
logout.php	Администрирование	Реализует выход администратора из системы.
output_fns.php	Функции	Набор функций вывода HTML-содержимого.
results.php	Каталог	Отображает результат поиска по ключевому слову.
search.php	Каталог	Форма, позволяющая организовать поиск по ключевому слову.
show_device.php	Каталог	Отображает данные по определенному устройству.
show_firm.php	Каталог	Отображает все устройства определенного производителя оборудования.

Продолжение табл. 6.5.

show_typ.php	Каталог	Отображает все устройства определенного типа оборудования.
sto.sql	SQL	SQL-код создания базы данных sto.
sto_fns.php	Функции	Набор подключаемых файлов.
user_auth_fns.php	Функции	Набор функций аутентификации администраторов.

Чтобы код проекта работал как задумано, необходимо задействовать режим magic quotes. Если это не выполнено, потребуется применять функцию addslashes[] ко вводимой в базу данных MySQL информации, а к извлекаемым из нее данным - функцию stripslashes[]. Упомянутый режим служит как удобное сокращение [5].

Режим magic quotes можно задействовать для отдельных каталогов с помощью следующих директив файла .htaccess:

php_value magic_quotes_gpc on (для PHP 4), либо

php3_magic_quotes_gpc on (для PHP 3).

Для практической реализации БД необходимо создать на сервере MySQL саму базу sto и таблицы, используя язык SQL.

Листинг 6.1. SQL-код создания базы данных sto с комментариями

```
/* создаем базу данных sto, если таковая отсутствует */
create database if not exists sto;

/* использовать базу данных sto */
use sto;

/* удалить таблицу admin, если она уже существует в БД */
drop table if exists admin;

/* создать таблицу admin со следующими полями и определить первичный ключ */
CREATE TABLE admin (    username char(16) NOT NULL,
                        password char(16) NOT NULL,
                        PRIMARY KEY (username) ) TYPE=MyISAM;

/* добавить в таблицу admin пользователя и пароль */
INSERT INTO admin VALUES ('system','5940bb836c4aa055')
```

Продолжение листинга 6.1.

```
/* удалить таблицу devices, если она уже существует в БД */
drop table if exists devices;

/* создать таблицу devices со следующими полями, определить первичный ключ */
CREATE TABLE devices (  devid int(10) unsigned NOT NULL auto_increment,
                        typeid int(10) unsigned NOT NULL,
                        firmid int(10) unsigned NOT NULL,
                        devname varchar(50) default NULL,
                        description varchar(255) default NULL,
                        descfile varchar(255) default NULL,
                        PRIMARY KEY (devid) ) TYPE=MyISAM;

/* удалить таблицу firms, если она уже существует в БД */
drop table if exists firms;

/* создать таблицу firms со следующими полями и определить первичный ключ */
CREATE TABLE firms (    firmid int(10) unsigned NOT NULL auto_increment,
                        firmname varchar(50) NOT NULL,
                        PRIMARY KEY (firmid) ) TYPE=MyISAM;

/* удалить таблицу types, если она уже существует в БД */
drop table if exists types;

/* создать таблицу types со следующими полями и определить первичный ключ */
CREATE TABLE types (    typeid int(10) unsigned NOT NULL auto_increment,
                        typename char(40) NOT NULL,
                        PRIMARY KEY (typeid) ) TYPE=MyISAM;
```

Чтобы создать в системе базу данных, в среде MySQL потребуется выполнить сценарий sto.sql, имея права привилегированного пользователя (root):

mysql -u root -p < sto.sql

(Следует ввести пароль привилегированного пользователя.)

Предварительно необходимо изменить пароль пользователя sto (по умолчанию он имеет значение password).

6.4. Реализация интерактивного каталога

Это приложение содержит три сценария, связанных с каталогами: главной страницы, страницы типов и производителей оборудования и страницы информации об устройстве.

Титульная страница сайта генерируется через сценарий index.php. Вывод этого сценария показан на рис. 6.4.

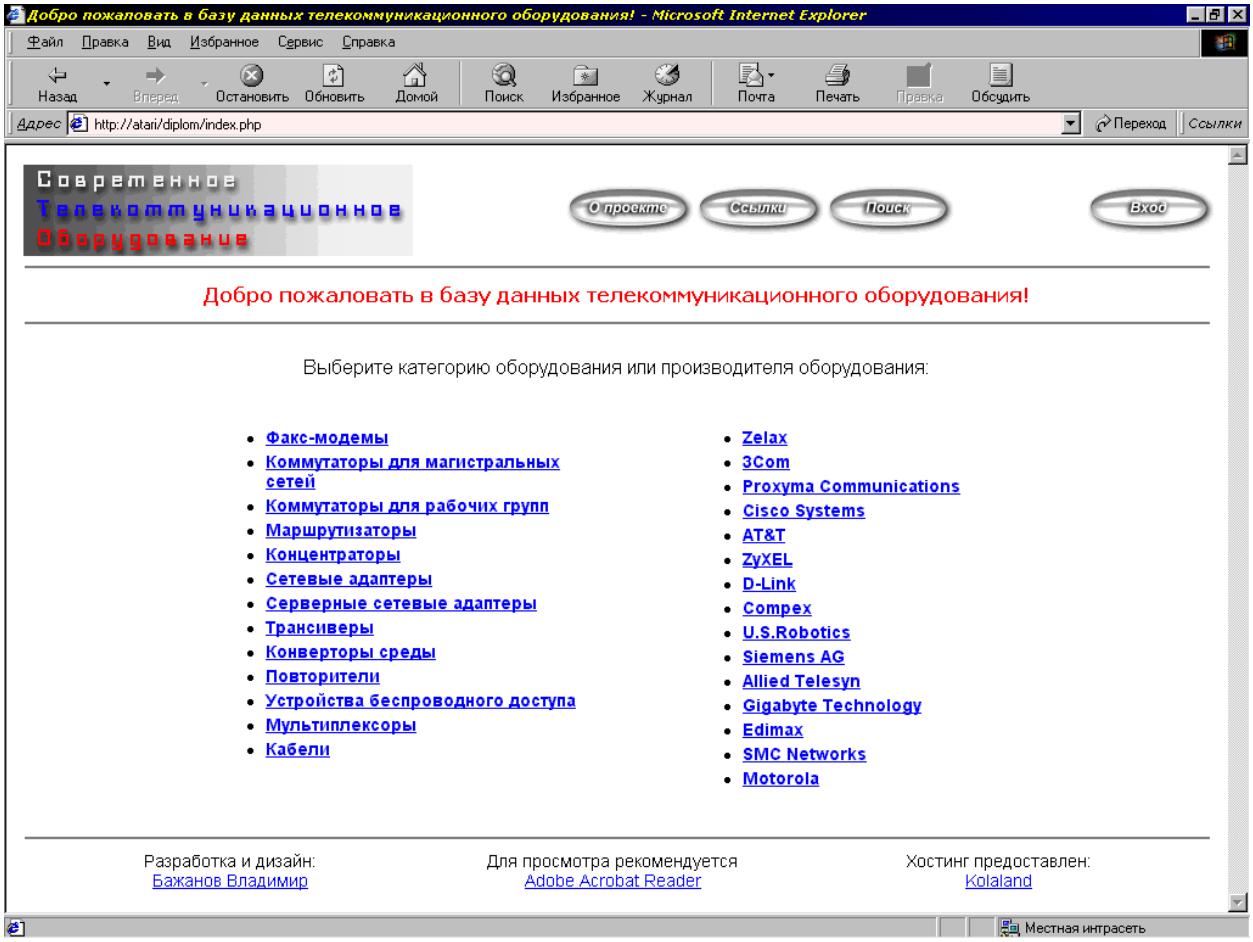

Рис. 6.4. Титульная страница сайта

Кроме списка типов оборудования и списка производителей окно содержит кнопки входа в меню администратора базы данных («Вход»), поиска по ключевому слову («Поиск»), а также кнопки, открывающие страницы со списком ссылок («Ссылки») и краткой информации о проекте

(«О проекте»). Эти элементы содержатся на каждой странице, открываемой в процессе просмотра содержимого базы данных.

Когда пользователь выполняет щелчок на одном из типов оборудования или производителе, открывается страница, содержащая список устройств выбранного типа или производителя соответственно. За это отвечают сценарии show_typ.php и show_firm.php. Страница, отображающая все устройства фирмы Zelax показана на рис. 6.5.

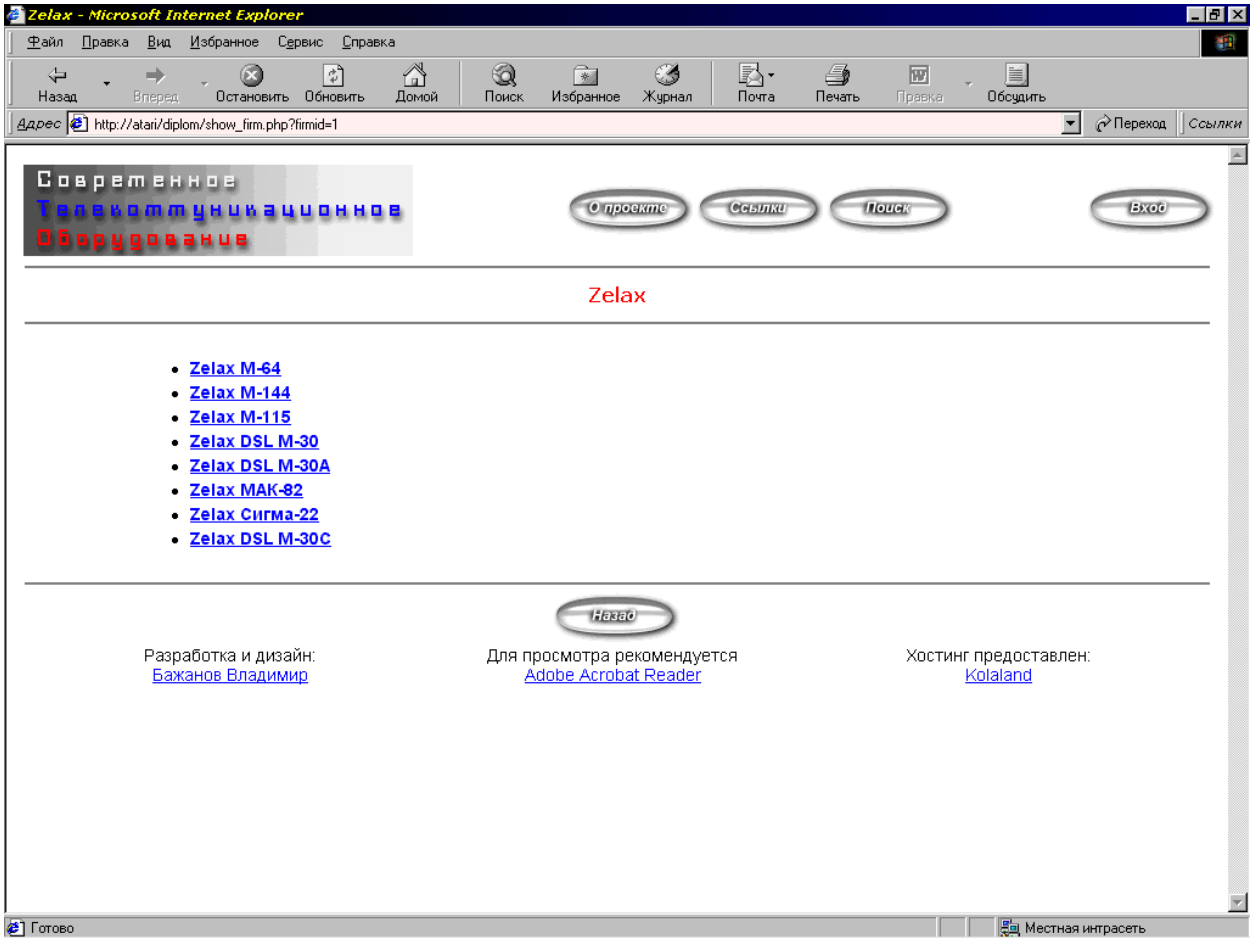

Рис. 6.5. Страница устройств определенного производителя

Все устройства данного производителя перечислены в виде ссылок. Когда пользователь выполняет щелчок на одной из них, открывается страница информации о соответствующем устройстве (рис. 6.6).

Для удобства навигации в нижней части страницы отображается кнопка «Назад», позволяющая вернуться на предыдущую страницу. Для перехода на титульную страницу сайта достаточно щелкнуть на изображении с надписью «Современное телекоммуникационное оборудование» в любой из страниц сайта.

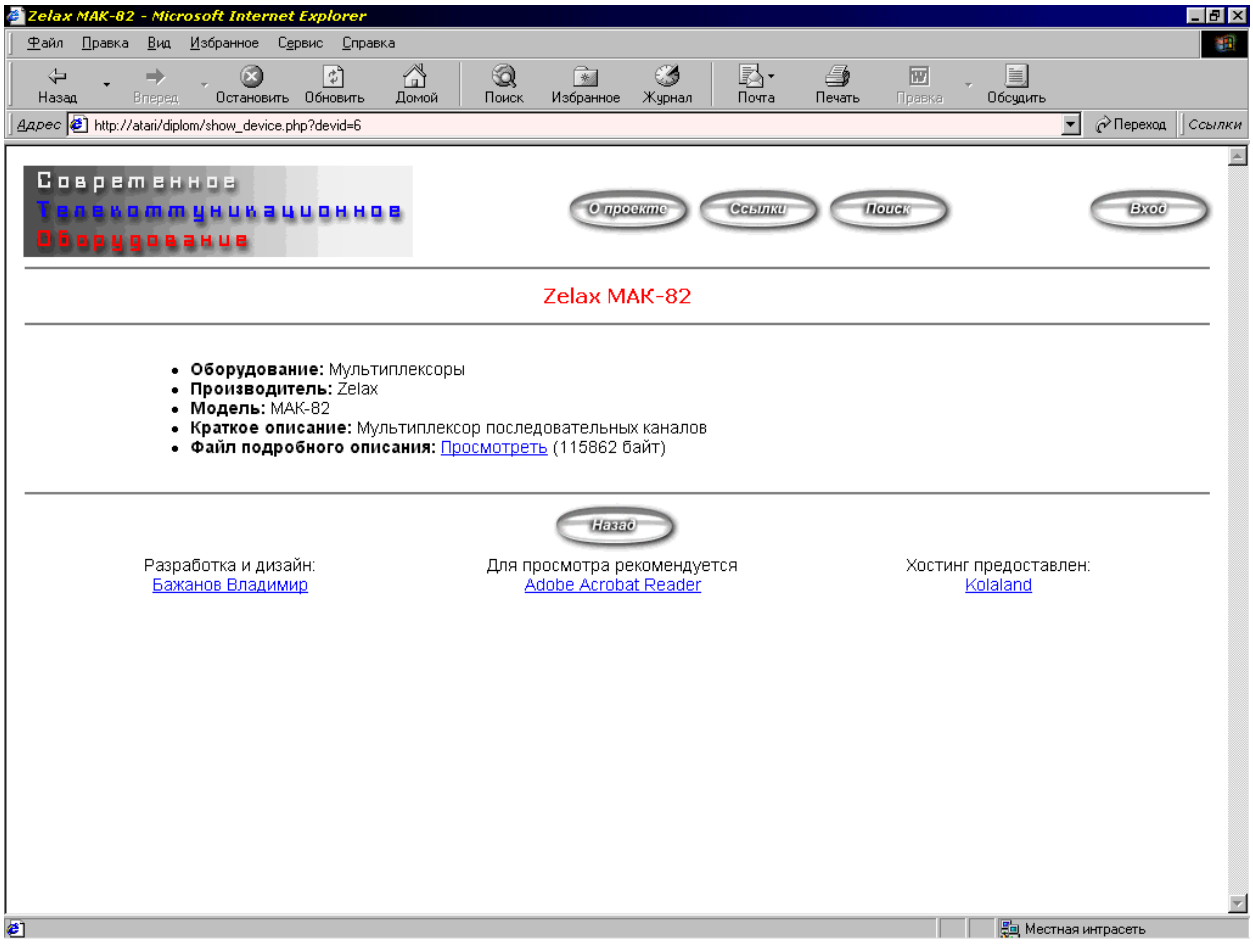

Рис. 6.6. Страница определенного устройства

При нажатии гиперссылки на файл подробного описания устройства в отдельном окне браузера загрузится файл описания в PDF формате. В этом окне также будет присутствовать инструментальная панель, позволяющая выполнять сохранение, вывод на печать и другие манипуляции с описанием (рис. 6.7). Вывод описания в отдельном окне реализован из соображений

удобства, так как такое окно всегда можно закрыть и продолжить работу с базой данных.

На случай отсутствия установленной программы просмотра PDF-файлов предусмотрена гиперссылка, ведущая на официальный сайт фирмы Adobe, где можно скачать Adobe Acrobat Reader бесплатно для любой операционной системы. Эта гиперссылка также присутствует во всех сценариях сайта.

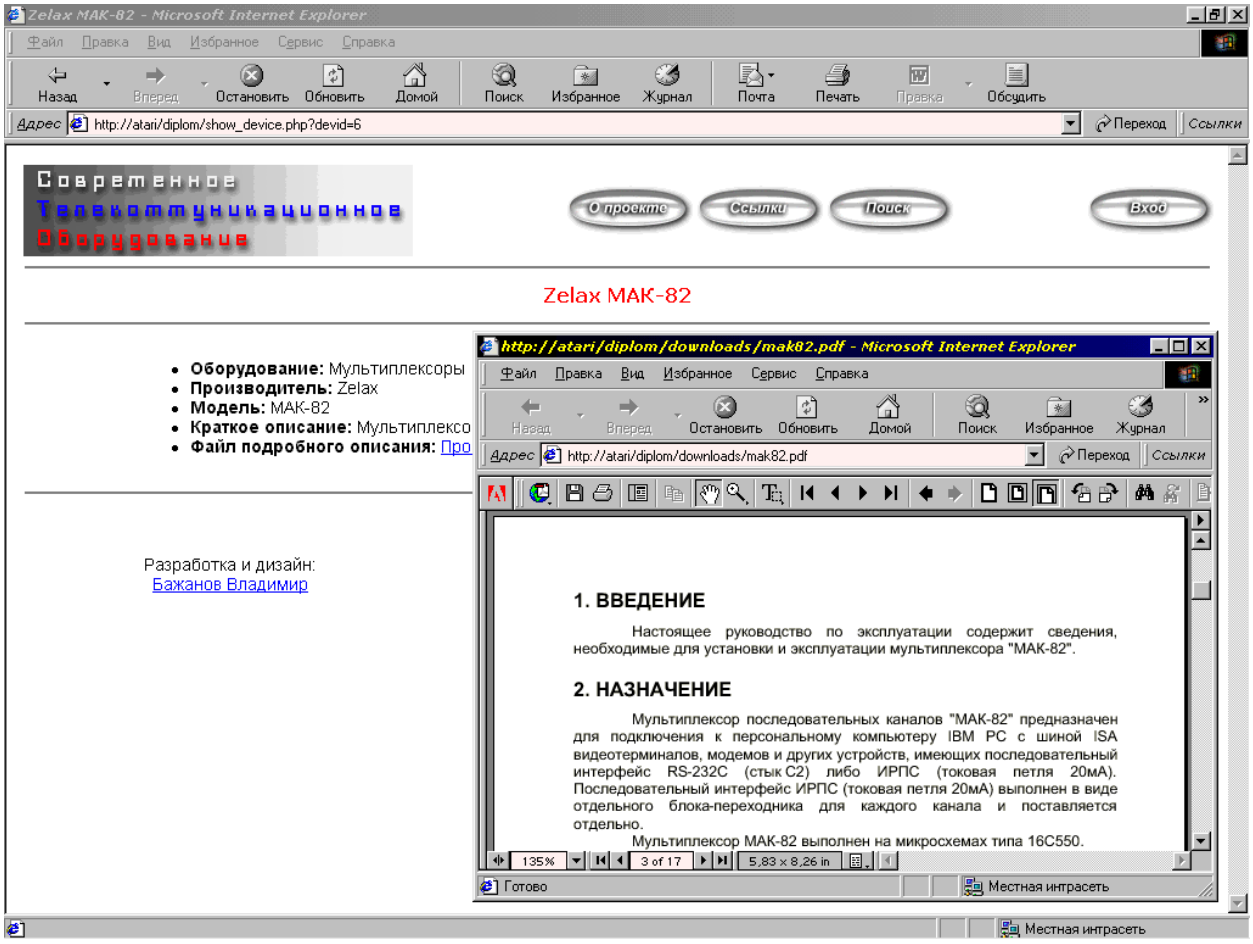

Рис. 6.7. Страница подробного описания устройства

Рассмотрим сценарии подробнее.

1. Списки типов и производителей оборудования.

Первый сценарий, index.php, выводит списки всех типов и производителей оборудования из базы данных (листинг 6.2).

Листинг 6.2. Сценарий вывода титульной страницы сайта (index.php)

```
<?
  // файл титульной страницы сайта
  // отображает списки типов оборудования, производителей оборудования
  // подключение файла функций
  require_once("sto_fns.php");

  do_html_header("Добро пожаловать в базу данных телекоммуникационного
оборудования!");

  echo "<center><p>Выберите категорию оборудования или производителя
оборудования:</p></center>";
  echo "<table width = \"80%\" align=center border = 0>";

  echo "<tr><td width = \"50%\" VALIGN=top><ul>";
  // извлечение типов оборудования из базы данных
  $typ_array = get_types();
  // отображение ссылок на страницы типов оборудования
  display_types($typ_array);
  echo "</td>";

  echo "<td VALIGN=top><ul>";
  // извлечение производителей оборудования из базы данных
  $fir_array = get_firms();
  // отображение ссылок на страницы производителей оборудования
  display_firms($fir_array);
  echo "</td></tr>";

  echo "</table>";
  echo "<hr>";

  // отобразить футер
  do_html_footer();
?>
```

Сценарий начинается с подключения файла sto_fns.php, который содержит все библиотеки функций для данного приложения.

Сценарий включает вызовы функций вывода HTML-содержимого, таких как do_html_header() и do_html_footer() (обе находятся в файле output_fns.php).

Кроме того, предусмотрен код проверки, что пользователь вошел в систему с правами администратора. В этом случае ему предоставляются другие средства навигации.

Ниже показаны наиболее важные части сценария:

```
// извлечение типов оборудования из базы данных
$typ_array = get_types();

// отображение ссылок на страницы типов оборудования
display_types($typ_array);

// извлечение производителей оборудования из базы данных
$fir_array = get_firms();

// отображение ссылок на страницы производителей оборудования
display_firms($fir_array);
```

Функции get_types(), get_firms(), display_types() и display_firms() содержатся, соответственно, в библиотеках device_fns.php и output_fns.php. Функця get_types() возвращает массив типов оборудования, который затем передается в функцию display_types(). Функция get_firms() работает аналогичным образом. Код функции get_types() приводится в листинге 6.3.

Листинг 6.3. Функция get_types() из библиотеки output_fns.php извлекает из базы данных типы оборудования

```
function get_types()
{
   // запрос к базе данных на список типов устройств
   $conn = db_connect();
   $query = "select typeid, typename from types";
   $result = @mysql_query($query);
   if (!$result)
     return false;
   $num_typs = @mysql_num_rows($result);
   if ($num_typs ==0)
     return false;
   $result = db_result_to_array($result);
   return $result;
}
```

Функция осуществляет соединение с базой данных и извлечение списка всех идентификаторов и наименований типов. Здесь используется ранее созданная функция db_result_to_array() из файла db_fns.php (листинг 6.4). Функция принимает идентификатор результата от MySQL и возвращает массив строк с числовой индексацией, где каждая строка представляет собой ассоциативный массив.

Листинг 6.4. Функция db_result_to_array() библиотеки db_fns.php преобразуетидентификатор результата MySQL в массив результатов

```
function db_result_to_array($result)
// функция заносит результат в массив
{
   $res_array = array();

   for ($count=0; $row = @mysql_fetch_array($result); $count++)
     $res_array[$count] = $row;

   return $res_array;
}
```

В нашем случае этот массив возвращается в сценарий index.php, где передается в display_types() из библиотеки output_fns.php. Эта функция отображает каждый тип оборудования в виде ссылки на страницу, отображающую список устройств определенного типа. Код функции показан в листинге 6.5.

Листинг 6.5. Функция display_types() из библиотеки output_fns.php отображает массив типов в виде списка ссылок на типы оборудования

```
function display_types($typ_array)
{
  if (!is_array($typ_array))
  {
     echo "<center><br>Типов оборудования нет.<br><br></center>";
  }
  else
  {
    echo "<br><table width = \"80%\" align=center border = 0>";

    foreach ($typ_array as $row)
    {
      $url = "show_typ.php?typid=".($row["typeid"]);
      echo "<tr><td><ul>";
      echo "<li><b>";
      $title = $row["typename"];
      do_html_url($url, $title);
      echo "</b></td></tr>";
    }
    echo "</table><br>";
  }
}
```

Функция преобразует каждый тип оборудования базы данных в ссылку. Все ссылки передаются в следующий сценарий, show_typ.php, но каждая из

них имеет свой параметр, идентификатор типа (typeid). (Это уникальное число, сгенерированное MySQL и применяемое для идентификации типа).

Упомянутый параметр определяет, какой тип оборудования будет отображать следующий сценарий.

2.Список устройств определенного типа оборудования.

Процесс вывода списка устройств определенного типа аналогичен рассмотренному выше. Вывод осуществляет сценарий show_typ.php, показанный в листинге 6.6.

Листинг 6.6. Сценарий, отображающий устройства определенного типа

```
<?
  // файл отображает все устройства определенного типа оборудования

  // подключение файла функций
  require_once("sto_fns.php");

  $name = get_type_name($typid);

  do_html_header($name);

  $device_array = get_devices($typid);

  display_devices($device_array);

  // если авторизован как администратор, то показать ссылки на редактирование
  if(session_is_registered("admin_user"))
  {
    display_button("edit_type_form.php?typid=$typid", "edit-category",
"Редактировать категорию");
    display_button("index.php", "back", "Назад");
  }
  else
    display_button("index.php", "back", "Назад");

  // отобразить футер
  do_html_footer();
?>
```

Структура этого сценария во многом подобна структуре сценария вывода титульной страницы с той лишь разницей, что вместо типов и производителей оборудования извлекаются устройства.

С использованием функции get_type_name() передаваемый идентификатор типа преобразуется в имя типа оборудования:

```
$name = get_type_name($typid);
```

Эта функция выполняет поиск имени типа оборудования в базе данных. Код функции показан в листинге 6.7.

Листинг 6.7. Функция get_type_name() библиотеки device_fns.php преобразует идентификатор типа в имя типа оборудования

```
function get_type_name($typid)
{
   // запрос к базе данных на получение имени типа по его идентификатору
   $conn = db_connect();
   $query = "select typename from types where typeid = $typid";
   $result = @mysql_query($query);
   if (!$result)
     return false;
   $num_typs = @mysql_num_rows($result);
   if ($num_typs ==0)
      return false;
   $result = mysql_result($result, 0, "typename");
   return $result;
}
```

После извлечения имени типа оборудования можно отобразить HTML-заголовок и перейти к извлечению из базы данных списка устройств, относящихся к выбранному типу:

```
$device_array = get_devices($typid);

display_devices($device_array);
```

Функции get_devices() и display_devices() во многом подобны функциям get_types(), get_firms(), display_firms() и display_types(). Единственное отличие состоит в том, что информация извлекается из таблицы устройств, а не таблицы типов или производителей.

Функция display_devices() предоставляет ссылку на каждое устройство определенного типа или производителя при помощи сценария

show_device.php. Каждая ссылка сопровождается параметром в виде суффикса, который определяет идентификатор конкретного устройства.

В заключительном фрагменте сценария show_typ.php содержится код для отображения дополнительных опций в случае, когда в систему входит администратор. Этот код рассматривается в разделе, посвященном функциям администрирования.

3. Отображение информации об устройстве.

Сценарий show_device.php принимает идентификатор типа оборудования в качестве параметра, а затем извлекает и отображает сведения о данном устройстве. Код сценария показан в листинге 6.8.

Этот сценарий также очень похож на сценарии вывода двух предыдущих страниц. С помощью строки, приведенной ниже, из базы данных извлекается информация об устройстве.

```
$device = get_device_details($devid);
```

Для вывода данных в формате HTML используется следующий код:

```
display_device_details($device);
```

Функция display_device_details() выполняет поиск файла изображения для устройства, который выглядит как images/devid.jpg. Если файл не существует, изображение не выводится.

Листинг 6.8. Сценарий отображает данные по определенному устройству

```
<?
  // файл отображает данные по определенному устройству

  // подключение файла функций
  require_once("sto_fns.php");

  $device = get_device_details($devid);
  do_html_header(get_firm_name($device["firmid"])." ".$device["devname"]);
  display_device_details($device);

  // установка ссылки на кнопку "назад"
  $target = "index.php";
  if($device["typeid"])
  {
    $target = "show_typ.php?typid=".$device["typeid"];
  }
  // если авторизован как администратор, то показать ссылки на редактирование
  if( check_admin_user() )
  {
    display_button("edit_device_form.php?devid=$devid", "edit-item",
"Редактирование записи");
    display_button($target, "back", "Назад");
  }
  else
  {
    display_button($target, "back", "Назад");
  }
  // отобразить футер
  do_html_footer();
?>
```

Оставшаяся часть сценария устанавливает средства навигации. Обычному пользователю предоставляется кнопка «Вход» (Вход для администратора), позволяющая войти в систему для администрирования, и кнопка «Назад» для возврата на предыдущую страницу. Если пользователь вошел в систему с правами администратора, ему предлагаются другие опции.

6.5. Реализация интерфейса администрирования

Реализованный в этом проекте интерфейс администрирования сводится к построению Web-интерфейса взаимодействия с базой данных, применяющем входную аутентификацию. Во многом повторяется код, используемый при реализации интерфейса пользователя. Код с краткими комментариями приводится ниже.

Интерфейс администрирования требует, чтобы пользователь вошел в систему через файл login.php, который затем выводит меню администратора, admin.php. Страница входа в систему показана на рис. 6.8.

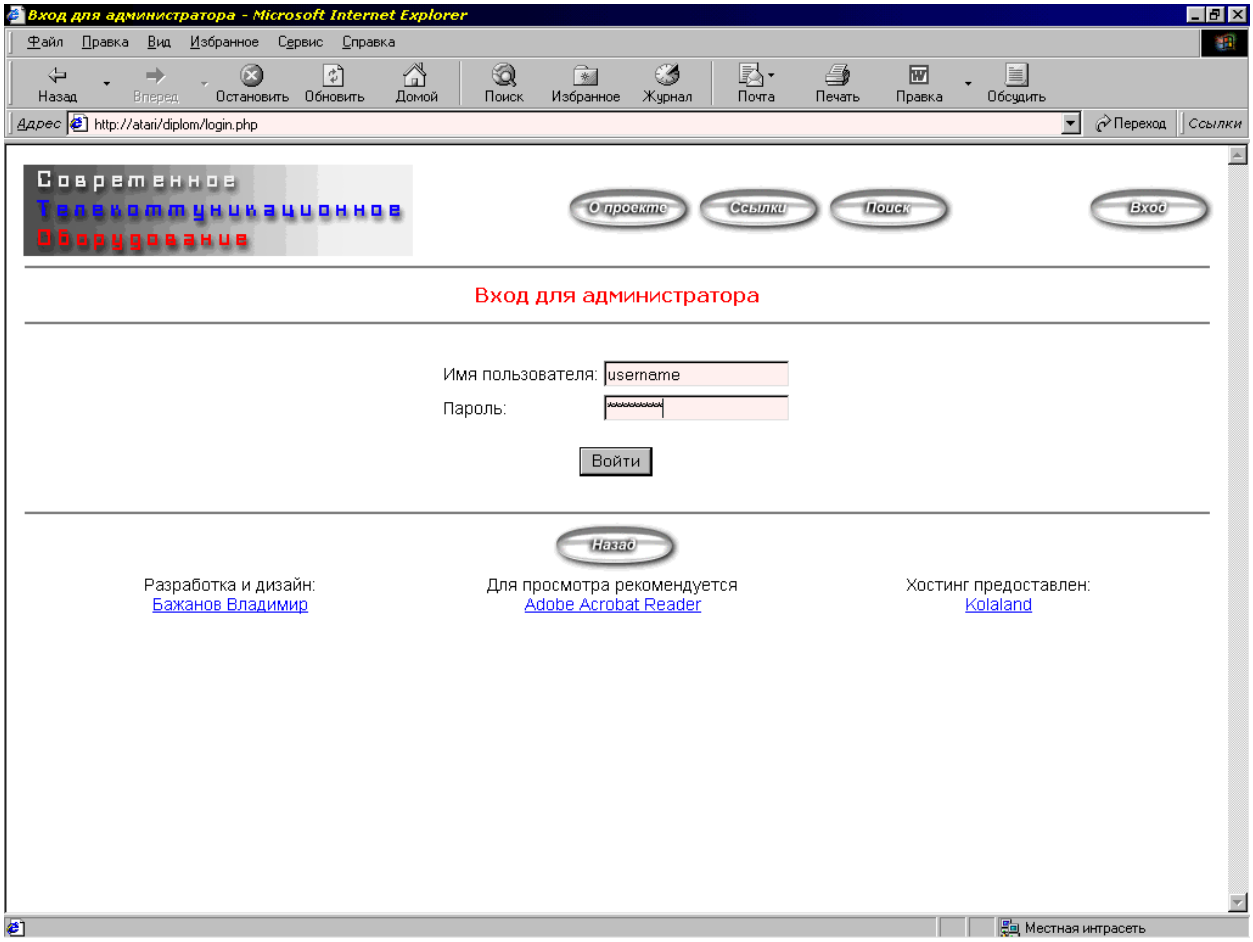

Рис. 6.8. Страница входа в систему администрирования

Листинг 6.9. Код сценария аутентификации администратора (login.php)

```
<?
  // файл позволяет администратору входить в систему для внесения изменений

  // подключение файла функций
  require_once("sto_fns.php");

  do_html_header("Вход для администратора");

  display_login_form();

  // отобразить кнопку
  display_button("index.php", "back", "Назад");

  // отобразить футер
  do_html_footer();
?>
```

Вызываемая функция display_login_form() находится в файле output_fns.php и имеет следующий вид:

Листинг 6.10. Функция, выводящая форму для аутентификации

```
function display_login_form()
{
  // Отобразить форму, которая запрашивает имя и пароль
?>
  <form method=post action="admin.php">
  <table bgcolor=#FFFFFF align=center>
   <tr>
     <td>Имя пользователя:</td>
     <td><input type=text name=username></td></tr>
   <tr>
     <td>Пароль:</td>
     <td><input type=password name=passwd></td></tr>
   <tr>
     <td colspan=2 align=center><br>
     <input type=submit value="Войти"></td></tr>
   </table></form><hr>
<?
}
```

Доступ к функциям администрирования осуществляется через меню, изображенного на рис. 6.9. Код этого меню с комментариями приведен ниже в листинге 6.11.

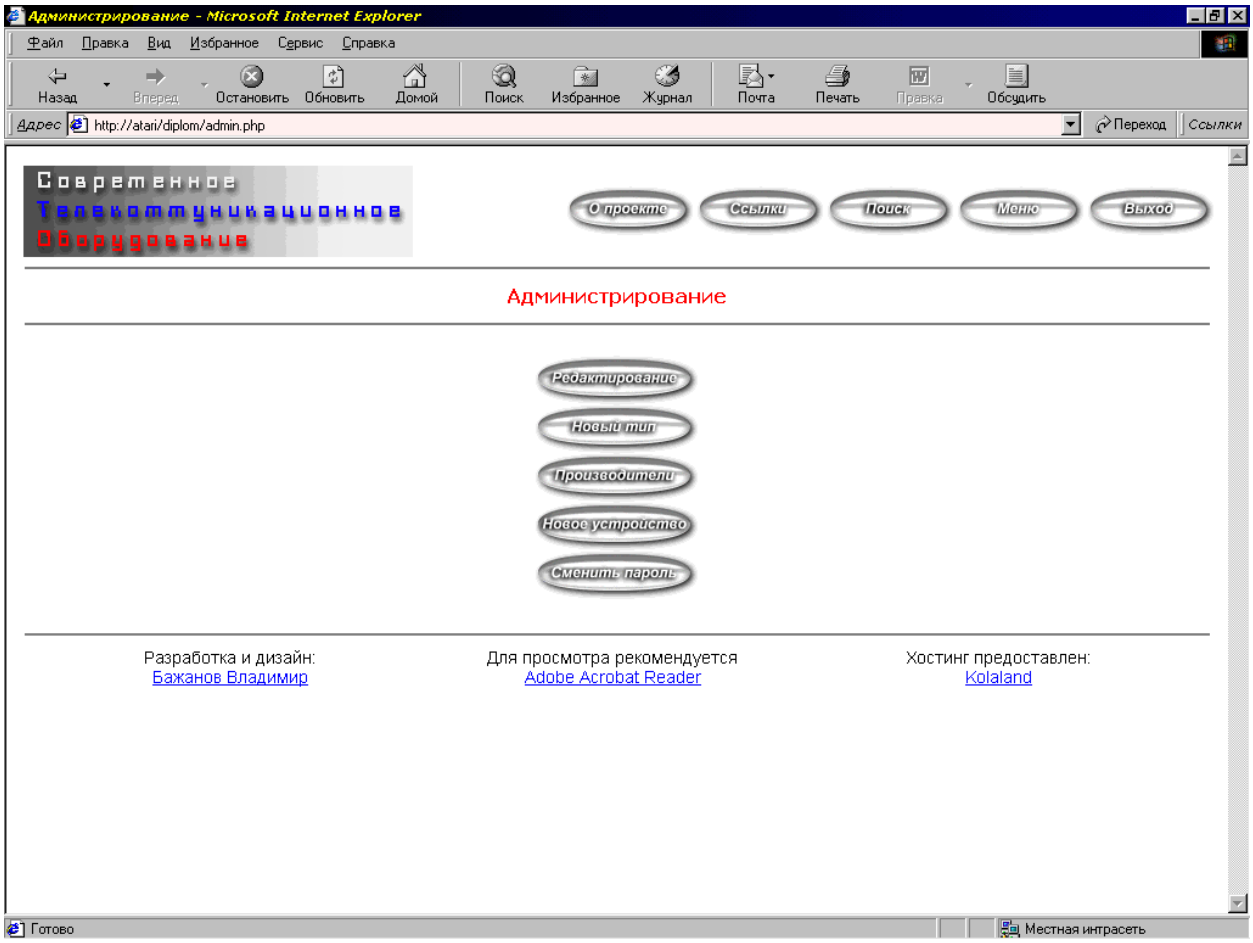

Рис. 6.9. Меню администрирования

Листинг 6.11. Сценарий, выполняющий аутентификацию и предоставляющий доступ к функциям администрирования

```
<?
  // файл меню администратора

  // подключение файлов функций
  require_once("sto_fns.php");

if ($username && $passwd)
// пользователь только что попытался войти в систему
{
    if (login($username, $passwd))
    {
      // зарегистрировать идентификатор пользователя,
      // если он содержится в базе данных
      $admin_user = $username;
      session_register("admin_user");
    }
    else
```

Продолжение листинга 6.11.

```
    {
      // неуспешная попытка входа в систему
      do_html_header("Проблема!");
      echo "<center><br>Вы не можете войти в систему.<br><br>";
      echo "Неверный пароль или имя пользователя. Попробуйте еще
раз.<br><br></center>";
      echo "<hr>";

      // отобразить кнопку
      display_button("login.php", "back", "Назад");
      // отобразить футер
      do_html_footer();
      exit;
    }
}

  do_html_header("Администрирование");
  if (check_admin_user())
    display_admin_menu();
  else
    echo "<center><br>У Вас нет прав для входа в
администрирование.<br><br></center>";
    echo "<hr>";

  // отобразить футер
  do_html_footer();
?>
```

На этом этапе администратор может изменять свой пароль либо выходить из системы.

Идентификация пользователя-администратора после входа в систему осуществляется через переменную сеанса $admin_user и функцию check_admin_user(). Эта и другие функции, используемые сценариями администрирования, содержатся в библиотеке admin_fns.php.

Когда администратору требуется добавить новый тип оборудования, производителя или устройство, вызывается сценарии insert_type_form.php, insert_firm_form.php или insert_device_form.php соответственно. Каждый сценарий предоставляет администратору форму для заполнения. Формы обрабатываются соответствующими сценариями (insert_type.php, insert_firm.php и insert_device.php), которые проверяют, что форма заполнена, и вводят новую информацию в базу данных. В качестве примера рассмотрим только сценарии, связанные с добавлением устройства, поскольку эти

сценарии мало чем отличаются от сценариев, добавляющих тип оборудования и производителя оборудования. Вывод сценария insert_device_form.php показан на рис. 6.10.

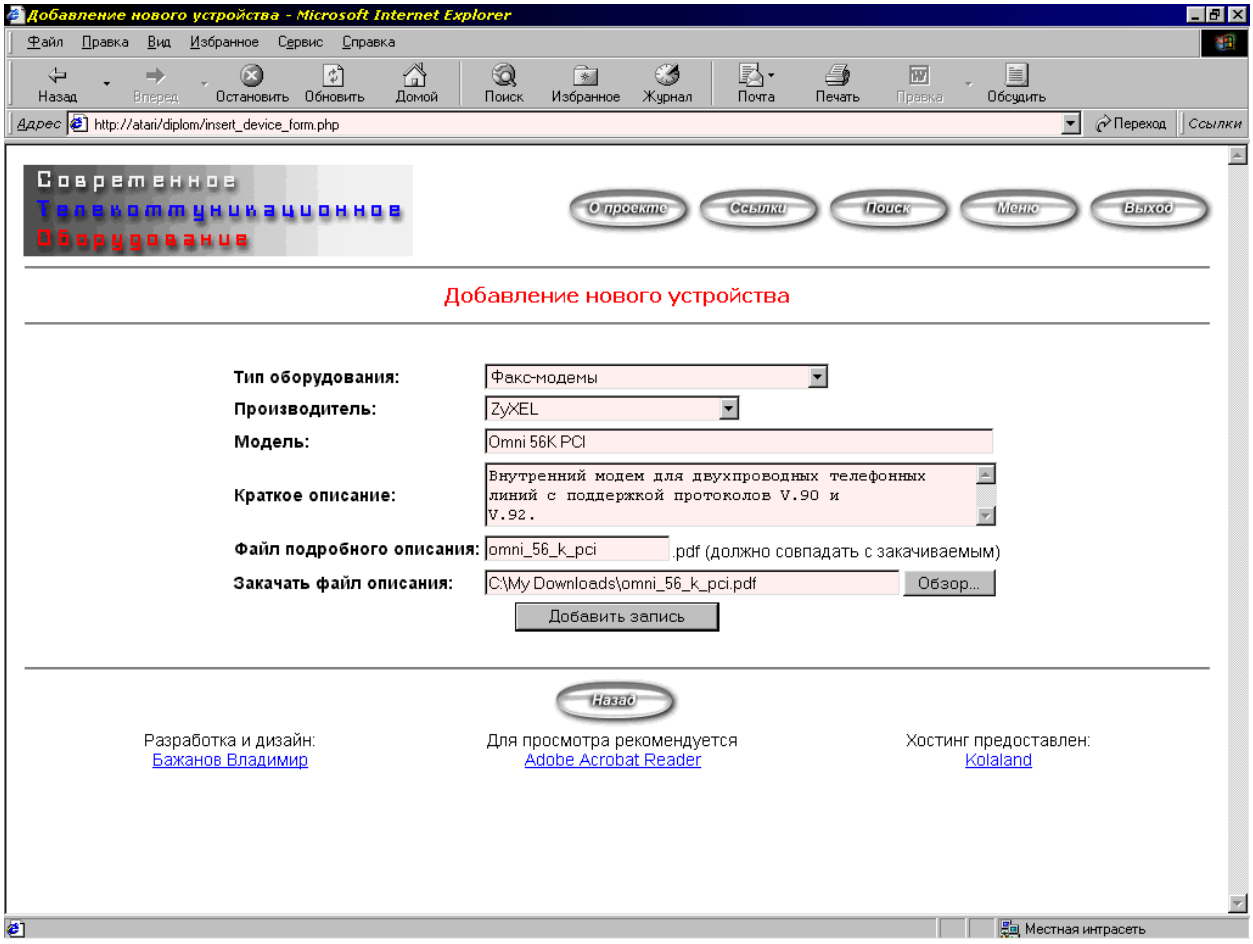

Рис. 6.10. Форма, позволяющая администратору вводить новые устройства в интерактивный каталог

Поля «Тип оборудования» и «Производитель» представляют собой HTML-элемент SELECT. Опции этих элементов генерируются за счет вызова ранее рассмотренных функций get_types() и get_firms().

После щелчка на кнопке «Добавить запись» активизируется сценарий insert_device.php. Код сценария с комментариями показан в листинге 6.12.

Листинг 6.12. Сценарий insert_device.php проверяет допустимость данных нового устройства и вводит их в базу данных

```
<?
  // файл добавляет новое устройство в базу данных

  // подключение файла функций
  require_once("sto_fns.php");

do_html_header("Добавление нового устройства");
if (check_admin_user())
{
  if (filled_out($HTTP_POST_VARS))
  {
    if(copy($zak, "downloads/$zak_name"))
      {echo("<center><br>Файл $zak_name размером $zak_size байт успешно
загружен на сайт.</center>");}
    else
      {echo("<center><br>Не удалось скопировать файл $zak_name на
сайт.</center>");}

    if(insert_device($devid, $typid, $firmid, $devname, $description,
$descfile))
      echo "<center><br>Устройство было добавлено в базу
данных.<br><br></center>";
    else
      echo "<center><br>Устройство не может быть добавлено в базу
данных.<br><br></center>";
  }
  else1
    echo "<center><br>Вы не заполнили форму или заполнили ее не правильно.
Попробуйте еще раз.<br><br></center>";
    echo "<hr>";
  display_button("admin.php", "back", "Назад в меню администратора");
}
else
  echo "<center><br>У Вас нет прав на просмотр этой
страницы.<br><br></center>";

  // отобразить футер
  do_html_footer();
?>
```

Этот сценарий вызывает функцию insert_device(). Эта и другие функции, используемые сценариями администрирования, содержатся в библиотеке admin_fhs.php.

Помимо добавления новых типов, производителей оборудования и устройств, пользователь с правами администрирования может редактировать и удалять эти элементы. При реализации упомянутых функциональных возможностей повторно использован максимально возможный объем кода.

Когда администратор в меню администрирования выполняет щелчок на изображении с надписью «Современное телекоммуникационное оборудование», являющееся ссылкой для перехода к титульной странице сайта, выводится список типов и производителей оборудования (с помощью сценария index.php). После этого администратор может выполнять навигацию по сайту так же, как и обычный пользователь. При этом используются те же сценарии.

Однако навигация администратора имеет некоторые отличия. Для него выводятся особые опции, связанные с тем, что зарегистрирована переменная сеанса $admin_user. В рассмотренной ранее странице show_device.php можно заметить отличия в меню опций (рис. 6.11).

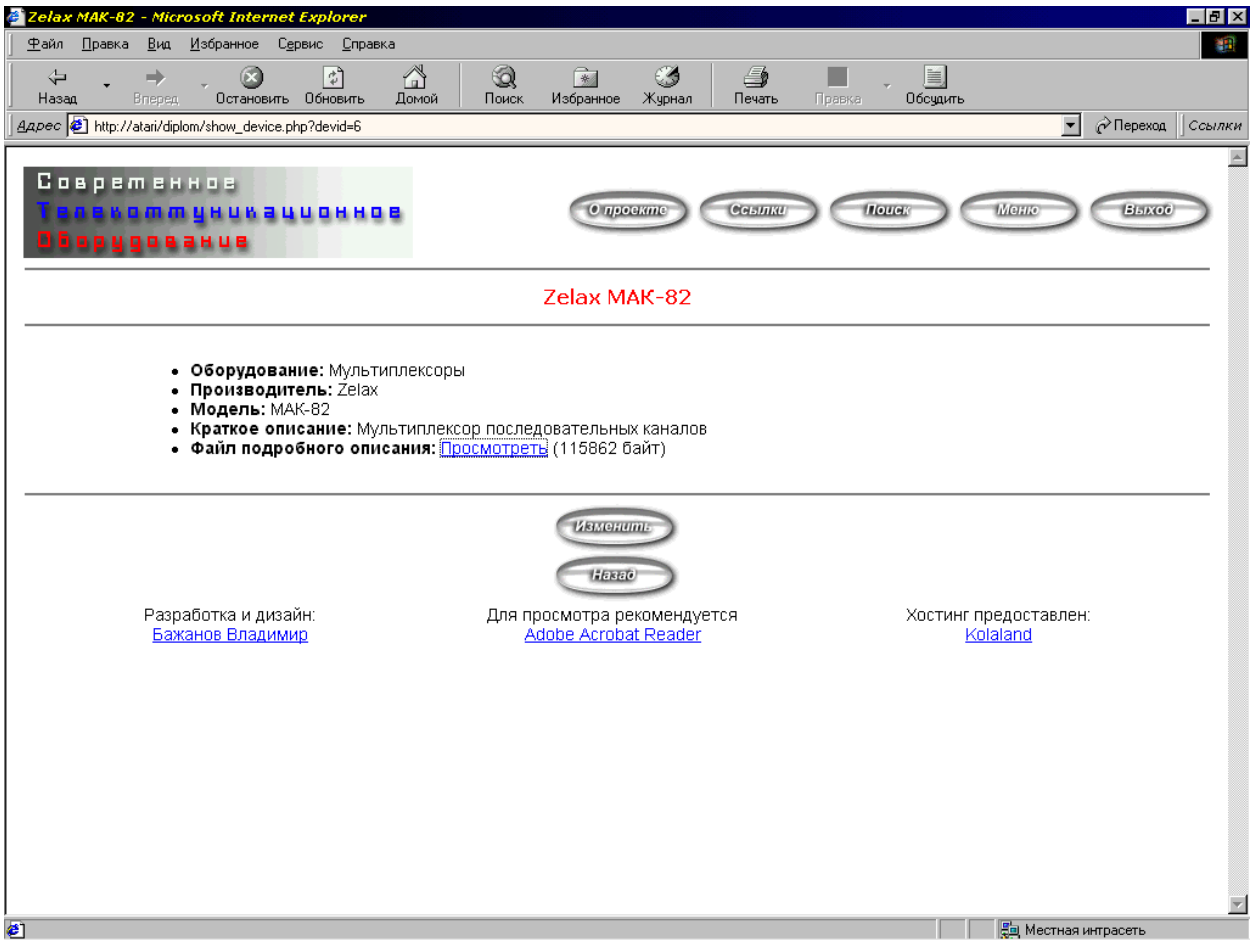

Рис. 6.11. Страница, отображающая информацию по определенному устройству в режиме администратора

На этой странице администратору предоставляются две дополнительные опции: «Изменить» (редактировать запись) и «Меню» (меню администрирования). Кроме того, в правом верхнем углу вместо кнопки «Вход» (вход для администратора) находится кнопка «Выход» (выход из системы). Это реализуется следующим фрагментом кода, представленным в листинге 6.13.

Листинг 6.13. Код отображения кнопки «Изменить»

```
  if( check_admin_user() )
  {
    display_button("edit_device_form.php?devid=$devid", "edit-item",
"Редактирование записи");
    display_button($target, "back", "Назад");
  }
```

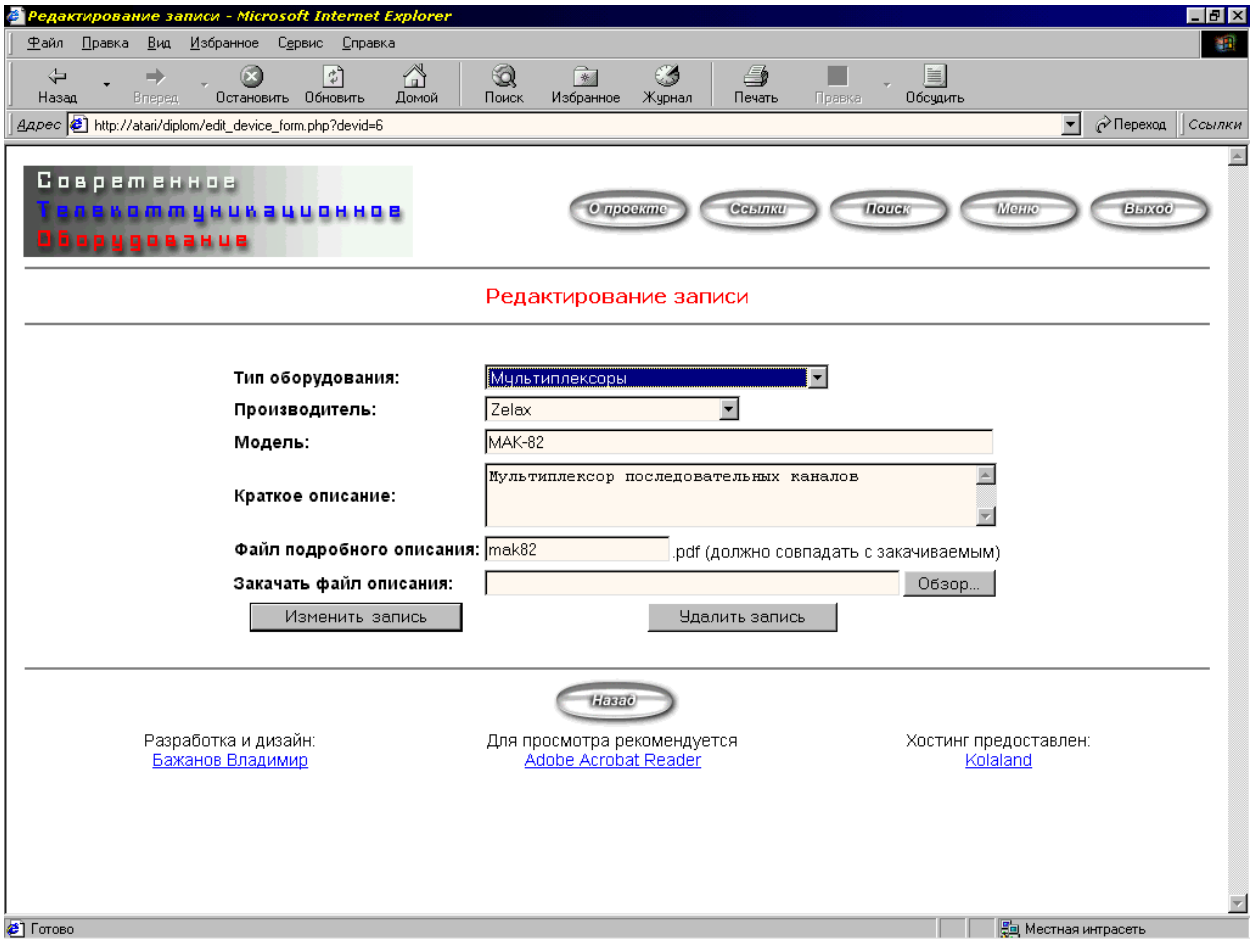

Рис. 6.12. Форма, позволяющая администратору редактировать информацию об устройстве, либо удалить устройство

Сценарии show_typ.php и show_firm.php также содержат данные опции. Когда администратор выполняет щелчок на кнопке «Изменить», запускается сценарий edit_device_form.php. (рис. 6.12).

Это та же форма, что ранее использовалась для извлечения данных о устройстве. В нее встроена опция передачи и отображения существующих сведений о устройстве. То же самое выполнено в отношении форм просмотра типов оборудования и производителей. Листинг функции display_device_form() приведен ниже.

Листинг 6.14. Функция display_device_form() библиотеки admin_fns.php выполняет двойное назначение вставки и редактирования данных

```
function display_device_form($device = "")
// отображает форму устройства
{
  // если передается существующий производитель оборудования,
  // переход к режиму редактирования
  $edit = is_array($device);
  $conn = db_connect();
  $max = mysql_query("select * from devices");
  $max_num = mysql_num_rows($max)+1;
?>
  <form enctype="multipart/form-data" method=post
        action="<?=$edit?"edit_device.php":"insert_device.php";?>">
  <table align=center border=0>
  <tr>
    <td><input type=hidden name=devid
value="<?=$edit?$device["devid"]:$max_num;?>"></td>
  </tr>
  <tr>
    <td><b>Тип оборудования:</b></td>
      <td><select name=typid>
      <?
          $typ_array=get_types();
          foreach ($typ_array as $thistyp)
          {
               echo "<option value=\"";
               echo $thistyp["typeid"];
               echo "\"";
               if ($edit && $thistyp["typeid"] == $device["typeid"])
                   echo " selected";
               echo ">";
               echo $thistyp["typename"];
               echo "\n";
          }
          ?>
  </select></td></tr>
  <tr>
  <td><b>Производитель:</b></td>
      <td><select name=firmid>
```

```
      <?  // список производителей оборудования поступает из базы данных
          $firm_array=get_firms();
          foreach ($firm_array as $thisfirm)
          {
               echo "<option value=\"";
               echo $thisfirm["firmid"];
               echo "\"";
               if ($edit && $thisfirm["firmid"] == $device["firmid"])
                   echo " selected";
               echo ">";
               echo $thisfirm["firmname"];
               echo "\n";
          }
       ?>
   </select></td></tr>
   <tr>
    <td><b>Модель:</b></td>
    <td><input type=text name=devname size=58 maxlen=40
               value="<?=$edit?$device["devname"]:""; ?>"></td></tr>
   <tr>
     <td><b>Краткое описание:</b></td>
     <td><textarea rows=3 cols=50
          name=description wrap=none>
          <?=$edit?$device["description"]:""; ?>
          </textarea></td></tr>
   <tr>
     <td><b>Файл подробного описания:</b></td>
     <td><input type=text name=descfile
          value="<?=$edit?$device["descfile"]:""; ?>">.pdf (должно совпадать
с закачиваемым)</td></tr>
   <tr>
     <td><b>Закачать файл описания:</b></td>
     <td><input name="zak" type="file" size=47></td></tr>
   <tr>
      <td <? if (!$edit) echo "colspan=2"; ?> align=center>
         <?
            if ($edit)
                  echo "<input type=hidden name=olddevid
                    value=\"".$device["devid"]."\">";
        ?>
        <input type=submit
               value="<?=$edit?"Изменить ":"Добавить"; ?> запись">
        </form></td>
        <?
           if ($edit)
           {
             echo "<td align=center>";
             echo "<form method=post action=\"delete_device.php\">";
             echo "<input type=hidden name=devid
                    value=\"".$device["devid"]."\">";
             echo "<input type=submit value=\"Удалить запись\">";
             echo "</form></td>";
           }
        ?>
   </td></tr></table></form>
<?
}
```

Если передается массив, содержащий данные о устройстве, форма переводится в режим редактирования, а поля заполняются существующими данными:

```
<input type=text name=devname value="<?=$edit?$device["devname"]:""; ?>">
```

Используется даже другая кнопка отправки формы. В форме редактирования их две - одна для обновления устройства, а вторая для его удаления. Они вызывают сценарии edit_device.php и delete_device.php, которые соответствующим образом обновляют базу данных.

Версии этих сценариев, связанные с типами и производителями оборудования, работают идентично, за исключением одного момента. Когда администратор пытается удалить тип оборудования, этого не произойдет, пока тип содержит какие-либо устройства. (Это проверяется путем запроса базы данных). В таком случае устраняются все возможности ошибочного удаления. В данном случае, если удалить тип оборудования, содержащий устройства, эти устройства станут «зависшими». Навигация к ним будет невозможна, поскольку с данными устройствами не связана ни один тип оборудования.

7. Руководство пользователя

После запуска браузера необходимо ввести адрес размещения базы данных в Internet. На существующий момент это адрес http://atari.kolaland.ru/diplom/.

В окне браузера загрузится титульная страница сайта, в центре которой содержатся списки типов оборудования и производителей оборудования, являющиеся гиперссылками на соответствующие страницы. В верхней части находится кнопка перехода на главную страницу и кнопки «О проекте», «Ссылки», «Поиск» и «Вход». Контактная информация и ссылка на сайт компании Adobe размещены в нижней части страницы. Вид титульной страницы представлен на рис. 7.1.

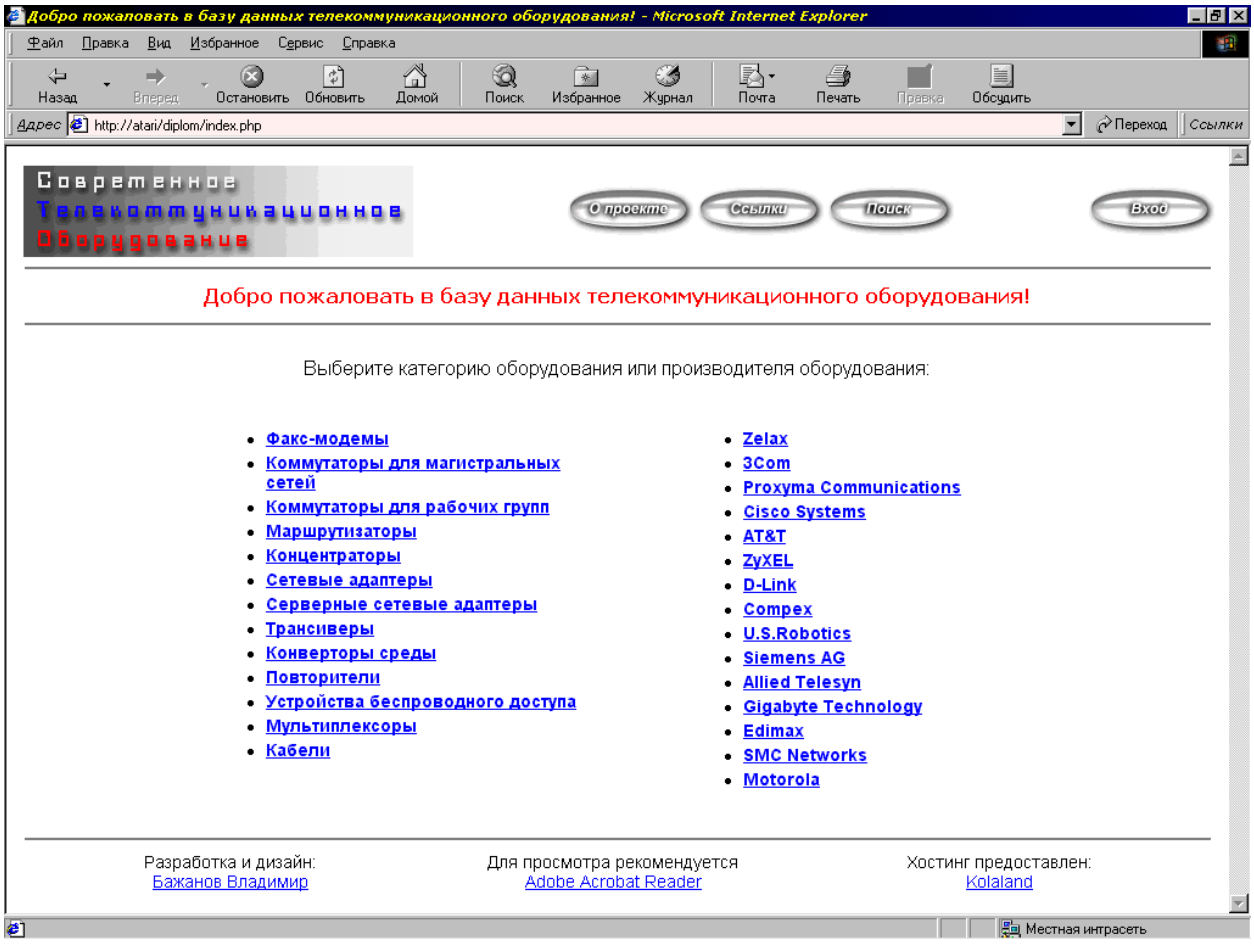

Рис. 7.1. Титульная страница сайта

После выбора в списке типов телекоммуникационного оборудования определенной категории или выбора определенного производителя оборудования, загрузится следующая страница сайта, в которой отобразится список всех устройств выбранного типа или производителя соответственно (рис. 7.2). Элемент этого списка также представляет собой гиперссылку на страницу краткого описания выбранного устройства. Верхняя часть формы остается без изменений, а в нижней части становится доступной кнопка «Назад», служащая для возврата на титульную страницу.

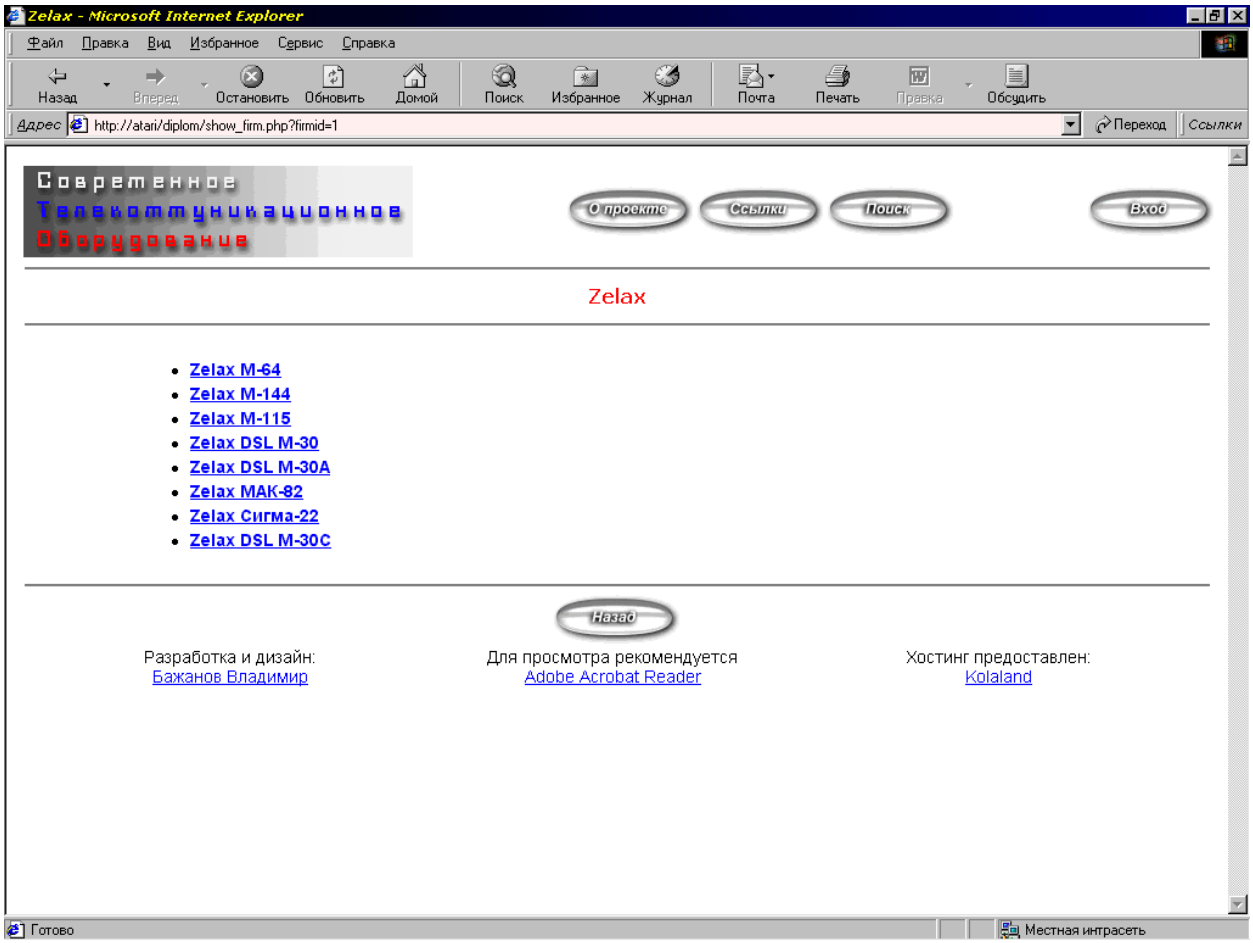

Рис. 7.2. Страница устройств определенного производителя

Далее загружается страница, отображающая следующую информацию о телекоммуникационном устройстве (рис. 7.3):

- Тип телекоммуникационного устройства;
- Производитель телекоммуникационного устройства;
- Модель устройства;
- Краткое описание устройства;
- Ссылка на подробное описание устройства с указанием размера файла описания.

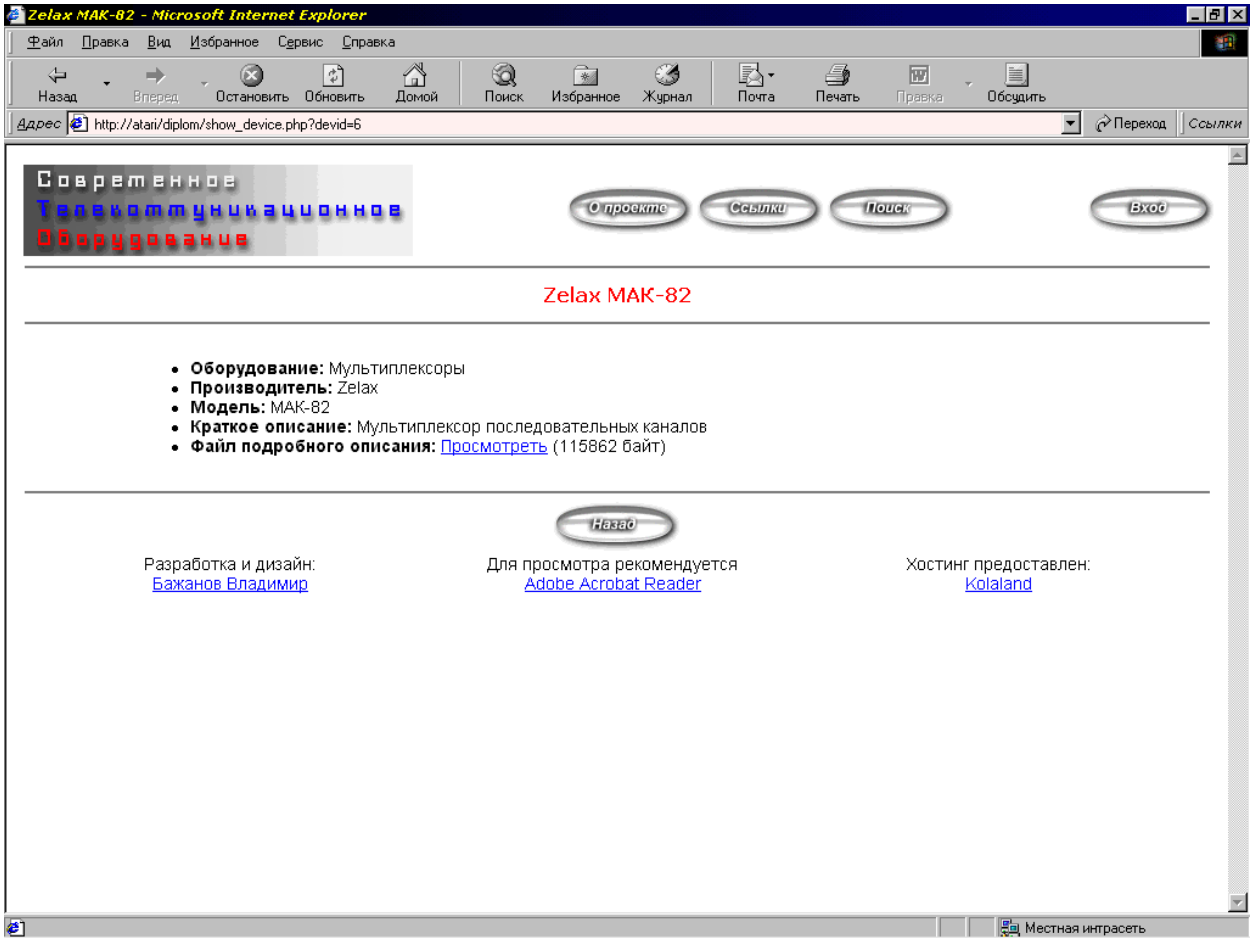

Рис. 7.3. Страница краткой информации о выбранном устройстве

Нажатие на гиперссылку «Посмотреть» позволяет просмотреть подробную информацию о выбранном устройстве, загружающуюся в новом окне браузера (рис. 7.4). Кнопка «Назад» позволяет вернуться на страницу со списком устройств определенного типа (или производителя) оборудования.

Основную часть этого окна занимает поле просмотра информации в PDF формате, в верхней части которого содержится панель. С ее помощью можно осуществить такие действия, как:

- Сохранение информации;
- Вывод информации на печать;
- Выбор режима просмотра (возможен показ закладок и др.);
- Масштабирование;
- Выделение текстовой и графической информации;
- Копирование информации;
- Переходы по страницам;
- Повороты страниц;
- Поиск текста.

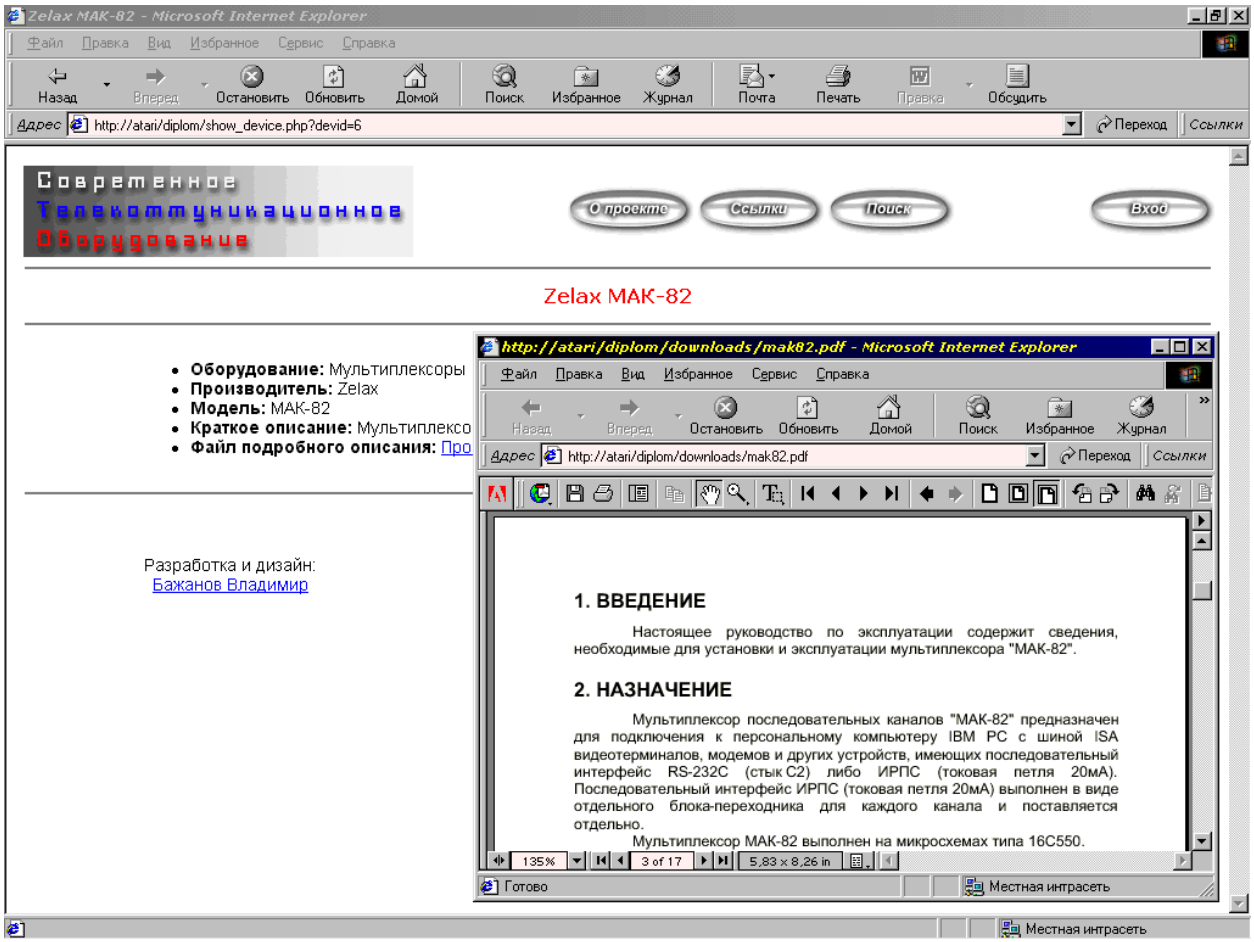

Рис. 7.4. Окно просмотра подробной информации об устройстве

Кроме приведенного выше способа просмотра информации в базе данных есть возможность осуществить поиск по ключевому слову. Для этого необходимо нажать кнопку «Поиск» в меню сайта. После этого загрузится страница с необходимыми полями для поиска (рис. 7.5).

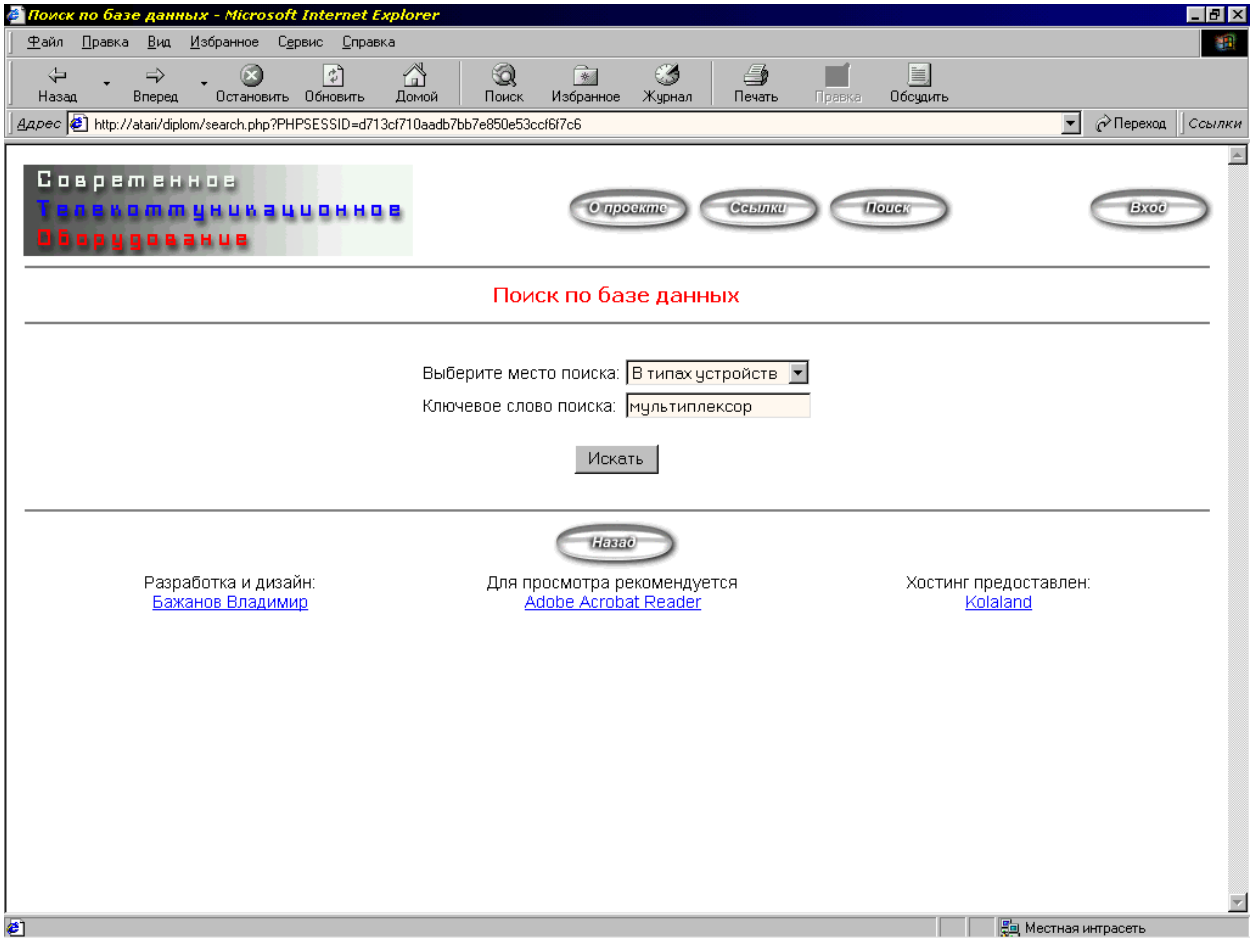

Рис. 7.5. Форма поиска информации в базе по ключевому слову

В приведенной форме необходимо выбрать критерий поиска (место поиска) и ввести искомое ключевое слово. После нажатия на кнопку «Искать» загрузится форма, отображающая список устройств, так или иначе удовлетворяющих критериям поиска (рис. 7.6).

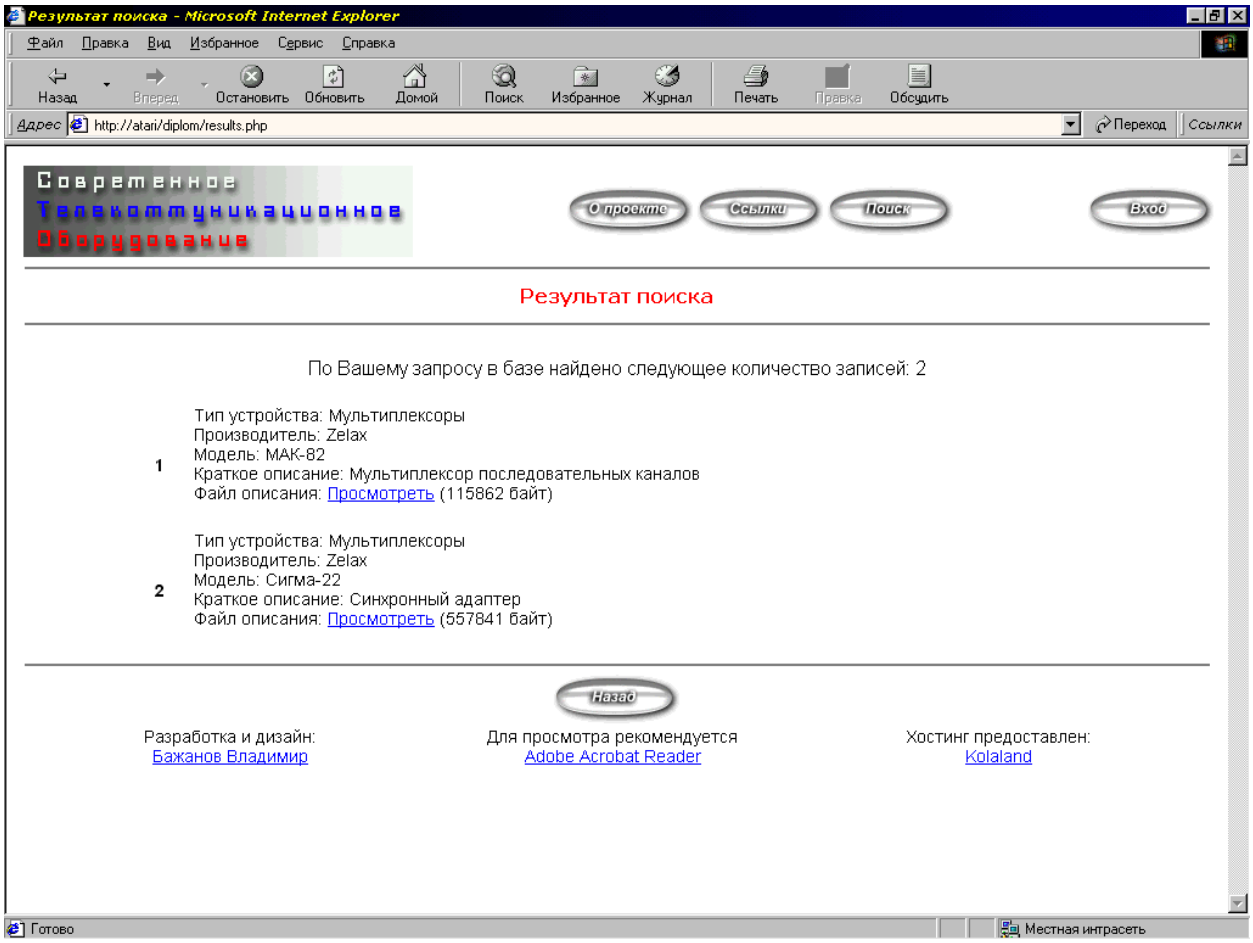

Рис. 7.6. Форма результата поиска по ключевому слову

Кроме того, как упоминалось выше, меню сайта содержит кнопку «О проекте», нажатие которой позволяет загрузить страницу, отображающую краткую информацию о дипломной работе. Вид страницы представлен на рис. 7.7.

При нажатии на кнопку «Ссылки» загрузится страница со списком ссылок на другие ресурсы сети Internet (рис. 7.8).

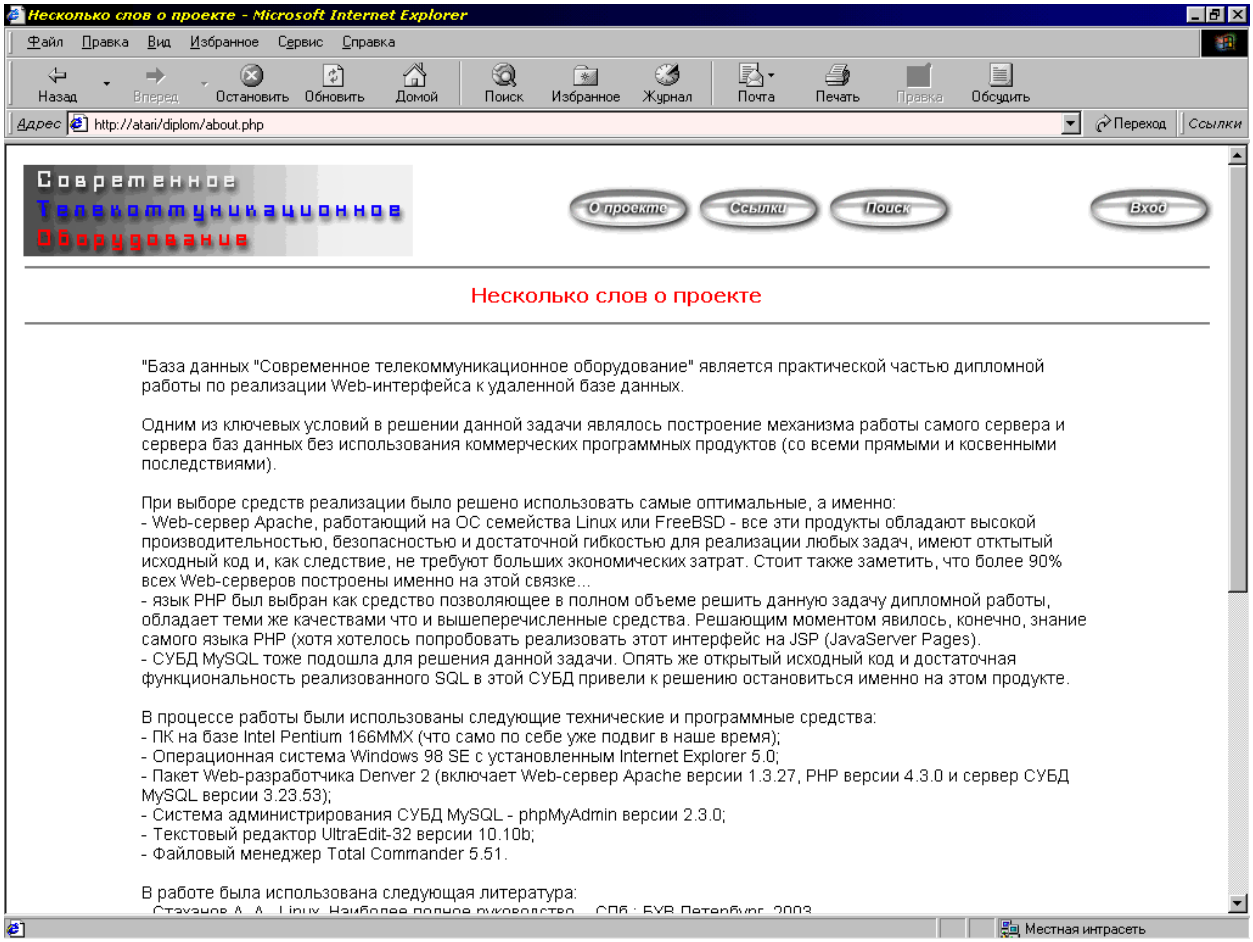

Рис. 7.7. Страница краткой информации о проекте

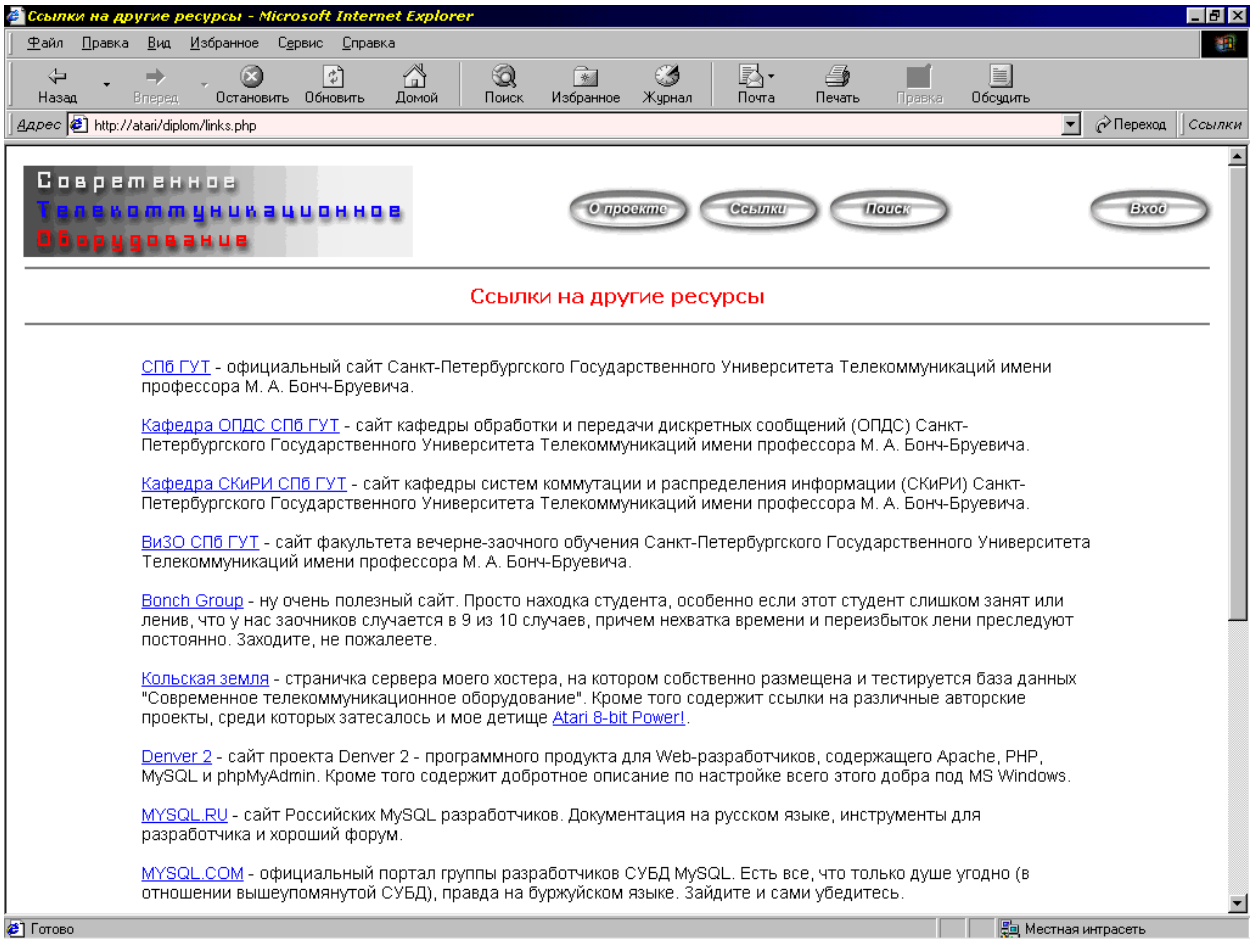

Рис. 7.8. Страница ссылок на ресурсы в сети Internet

Кнопка «Вход» в меню сайта позволяет войти в базу данных с правами администратора и осуществить редактирование, добавление и удаление информации в базе данных.

При нажатии на эту кнопку загружается страница ввода пароля и имени пользователя, имеющего права администратора (рис. 7.9). Для входа в систему необходимо правильно ввести пароль и имя пользователя в соответствующие поля нажать кнопку «Войти». В случае ввода неверного пароля или имени, система загрузит страницу с сообщением об ошибке и предложит новую попытку входа в систему.

Эта страница также содержит кнопку «Назад», нажатие на которую позволяет вернуться из формы аутентификации на титульную страницу.

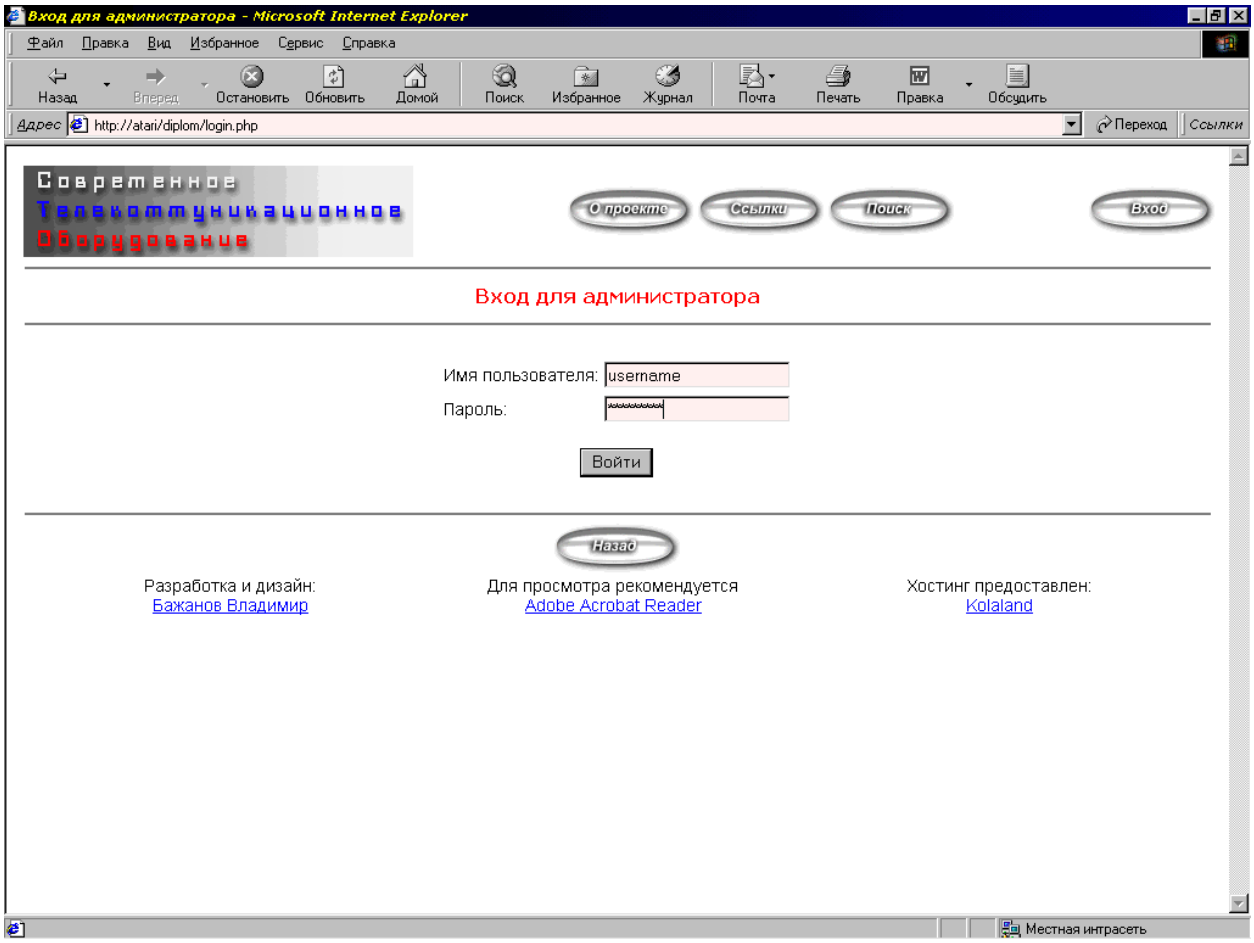

Рис. 7.9. Страница аутентификации администратора базы данных

После успешного ввода имени пользователя и пароля администратора, открывается форма, содержащая меню администратора, позволяющее осуществить следующие действия (рис. 7.10):

- Редактирование существующих записей;
- Добавление нового типа оборудования;
- Добавление нового производителя оборудования;
- Добавление нового устройства;
- Смену пароля администратора.

Кроме того, изменится вид меню сайта, расположенного в верхней части страницы. Появится кнопка «Меню», посредством которой всегда можно будет перейти в меню администрирования, кнопка «Вход» изменится на «Выход».

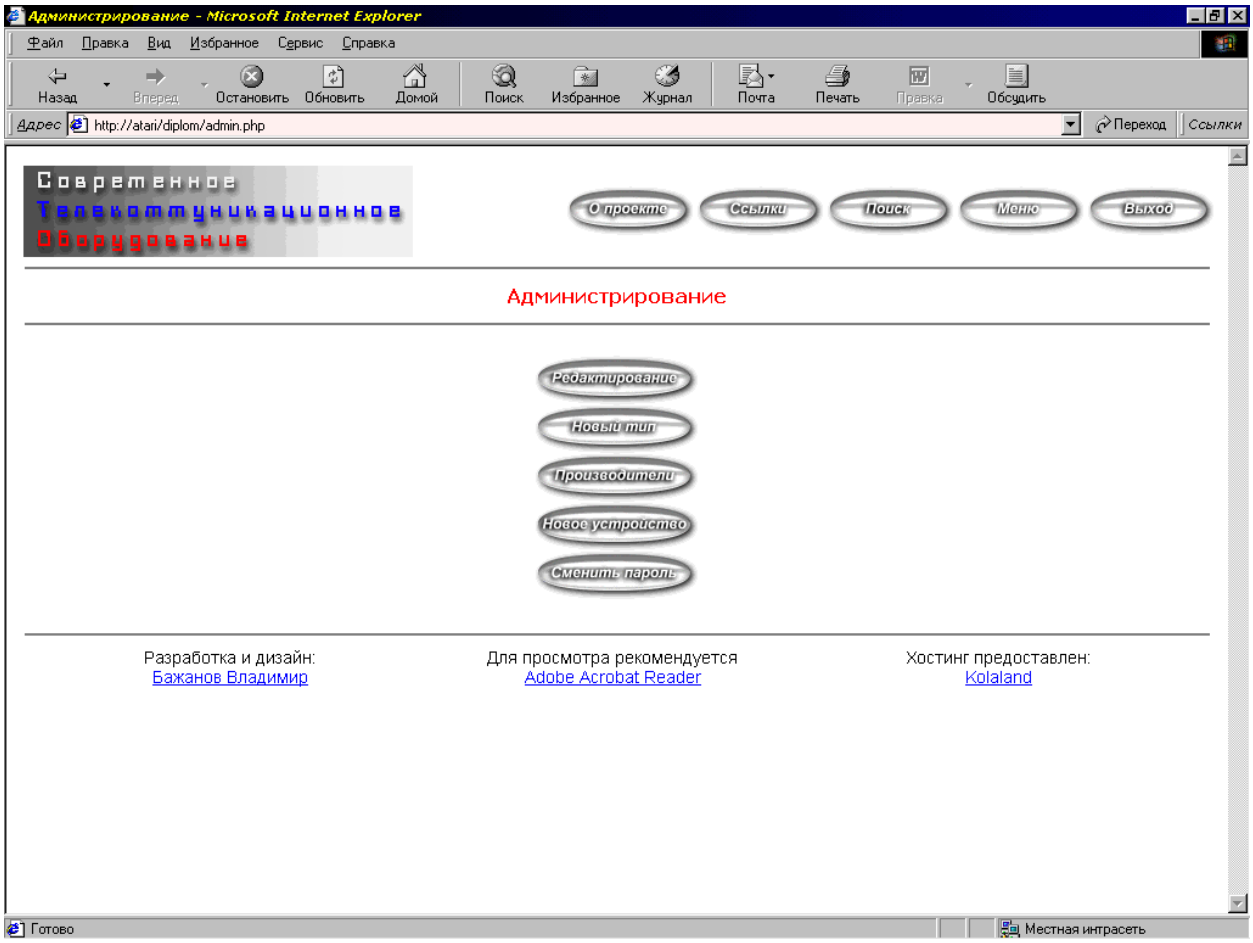

Рис. 7.10. Страница меню администрирования

Для смены пароля администратора базы данных необходимо нажать кнопку «Сменить пароль» в меню администрирования. После этого загрузится страница, позволяющая проделать данную операцию (рис. 7.11). Форма для смены пароля содержит три поля:

- Старый пароль;
- Новый пароль;
- Подтверждение.

Для правильной смены пароля необходимо в первое поле ввести существующий пароль (пароль, под которым был осуществлен вход в систему), во второе поле – новый пароль, длина которого должна быть не менее шести и не более шестнадцати символов. В третьем поле нужно ввести новый пароль еще раз на случай возможной ошибки.

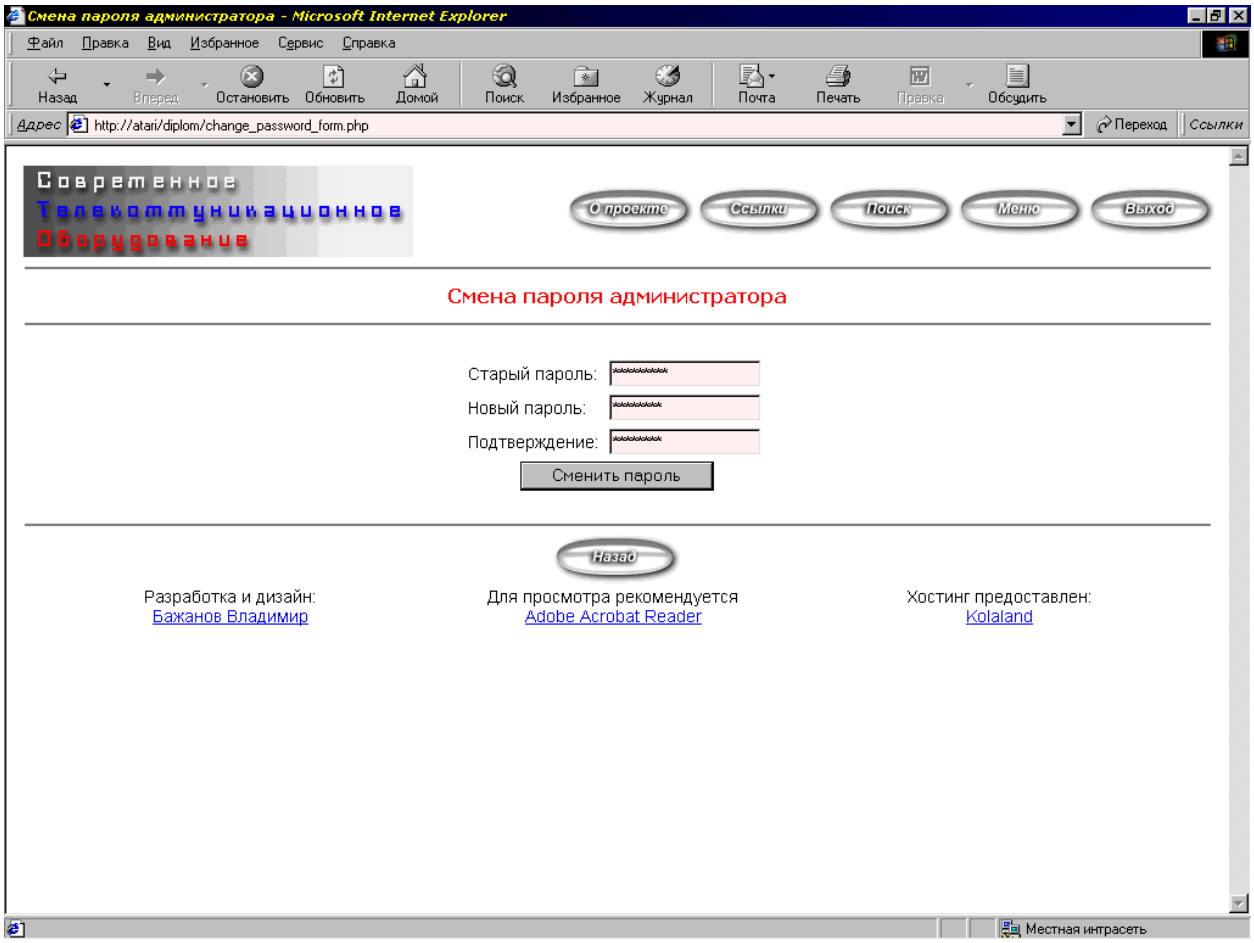

Рис. 7.11. Страница смены пароля администратора

После нажатия на кнопку «Сменить пароль», в окне браузера отобразится страница с результатом данной операции. Если смена пароля пройдет успешно, то система сообщит об этом. В противном случае, система выдаст сообщение о невозможности изменения пароля (если текущий пароль не верен), либо несовпадении нового пароля (если пароли в полях «Новый пароль» и «Подтверждение» различны). Нужно учесть и то, что для смены пароля необходимо заполнить все поля формы.

Кроме изменения пароля администратор может изменять содержимое базы данных. Для начала редактирования информации БД необходимо нажать на соответствующую кнопку «Редактирование» в меню администрирования, после чего загрузится титульная страница.

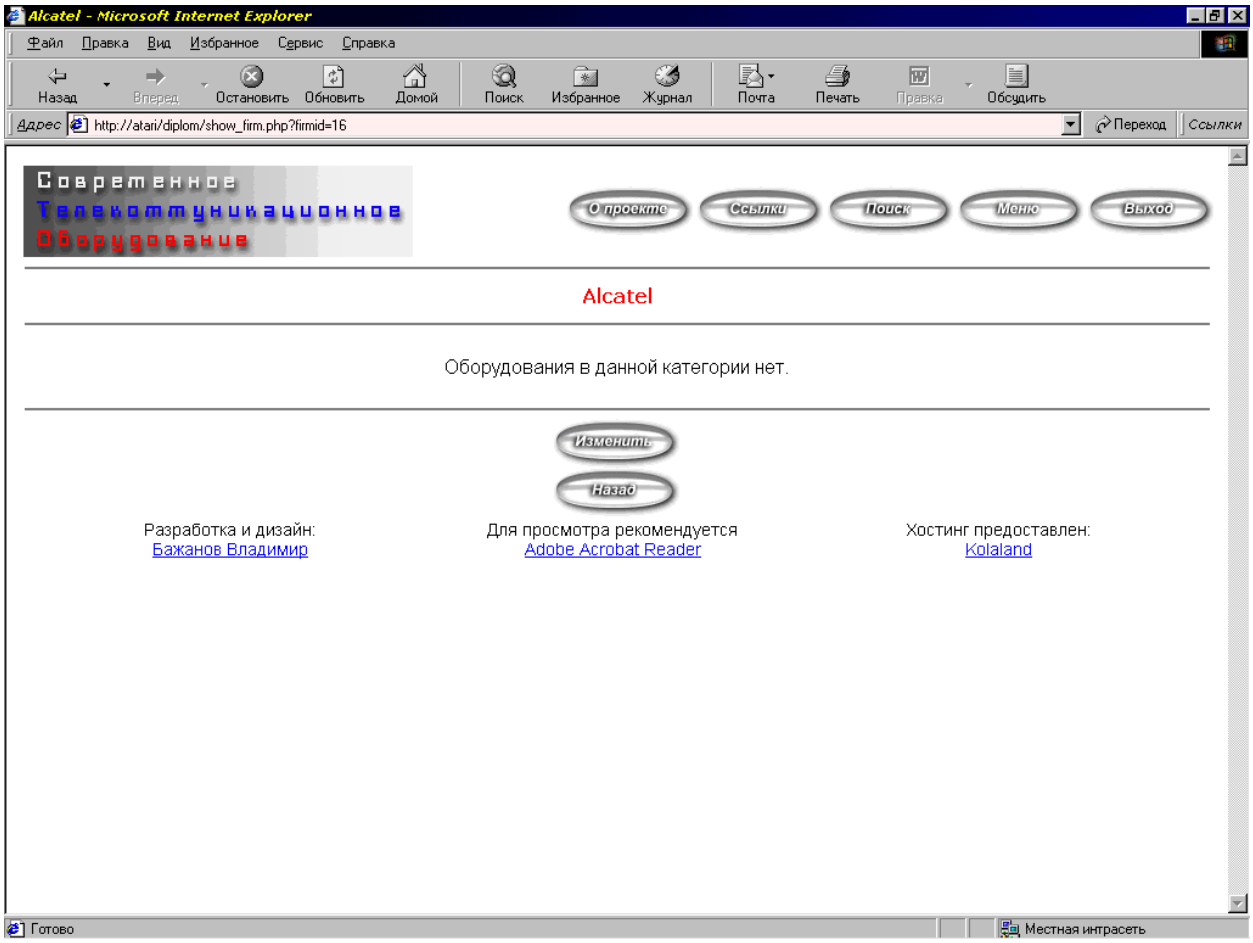

Рис. 7.12. Страница просмотра оборудования определенного производителя в режиме администратора

Навигация по страницам осуществляется таким же образом, как при работе с базой данных в режиме пользователя. Однако к просмотру информации добавится возможность ее изменения и удаления с помощью кнопки «Изменить», размещенной в нижней части страницы. В качестве примера приведены экранные формы изменения информации о производителе – фирме Alcatel (рис. 7.12).

После нажатия на кнопку «Изменить», появится страница изменения выбранного производителя (рис. 7.13). В этой форме одно поле для изменения наименования производителя и две кнопки: «Изменить производителя» и «Удалить производителя». При изменении наименования в поле и нажатии кнопки «Изменить производителя» в базе данных обновится

информация о текущем производителе. Нажатие кнопки «Удалить производителя», не изменяя наименования производителя в поле, повлечет за собой удаление текущего производителя из базы данных.

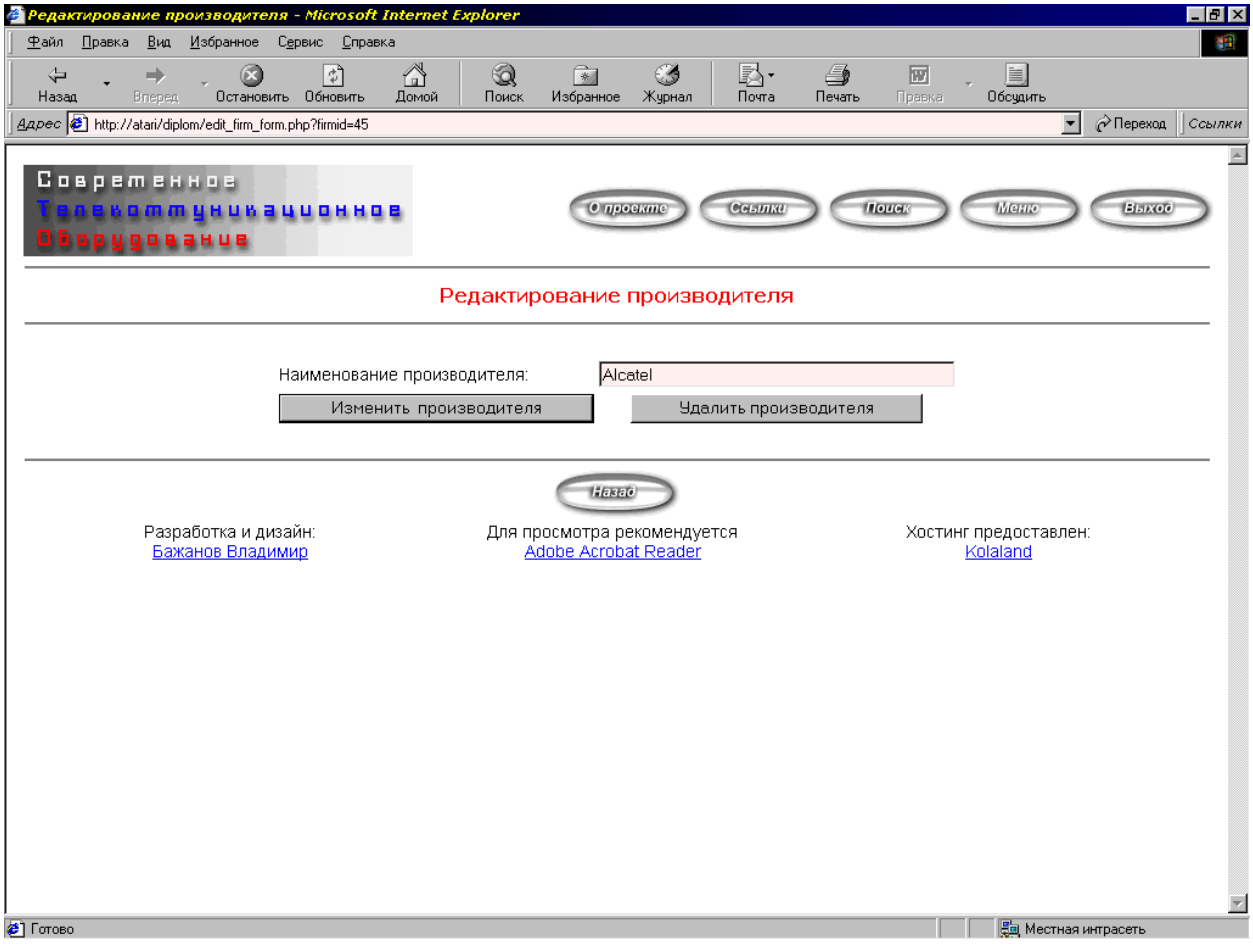

Рис. 7.13. Страница редактирования выбранного производителя

Таким же образом можно изменять и удалять в базе данных информацию о типах телекоммуникационного оборудования и о самих устройствах.

Удаление типа оборудования или производителя оборудования не сможет быть осуществлено в случае, если существует хотя бы одно устройство, принадлежащее этому типу или производителю.

Добавление нового типа телекоммуникационного оборудования осуществляется нажатием кнопки «Добавить тип» в меню

администрирования. Форма, демонстрирующая это, представлена на рис. 7.14.

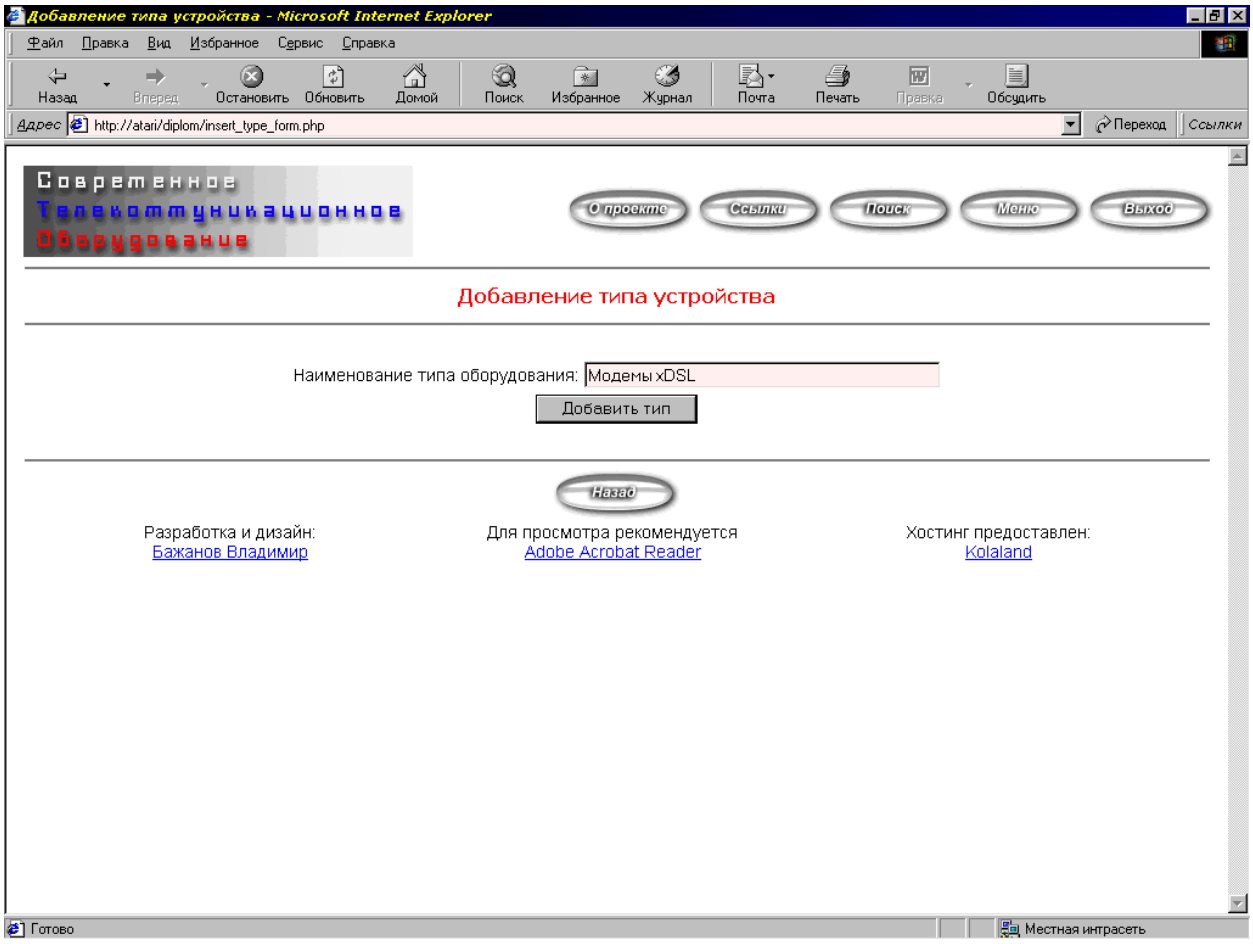

Рис. 7.14. Страница добавления нового типа оборудования в БД

В поле «Наименование типа оборудования» необходимо ввести строку с наименованием типа оборудования. При нажатии на кнопку «Добавить тип» осуществляется добавление данных в базу данных. О результате добавления в окне браузера появится сообщение.

Таким же образом происходит добавление нового производителя телекоммуникационного оборудования. При нажатии кнопки «Производители» в меню администрирования загрузится форма с полем «Наименование производителя» и кнопкой «Добавить производителя» (рис. 7.15).

Для добавления производителя в базу данных необходимо заполнить поле с именем производителя и подтвердить действие путем нажатия кнопки.

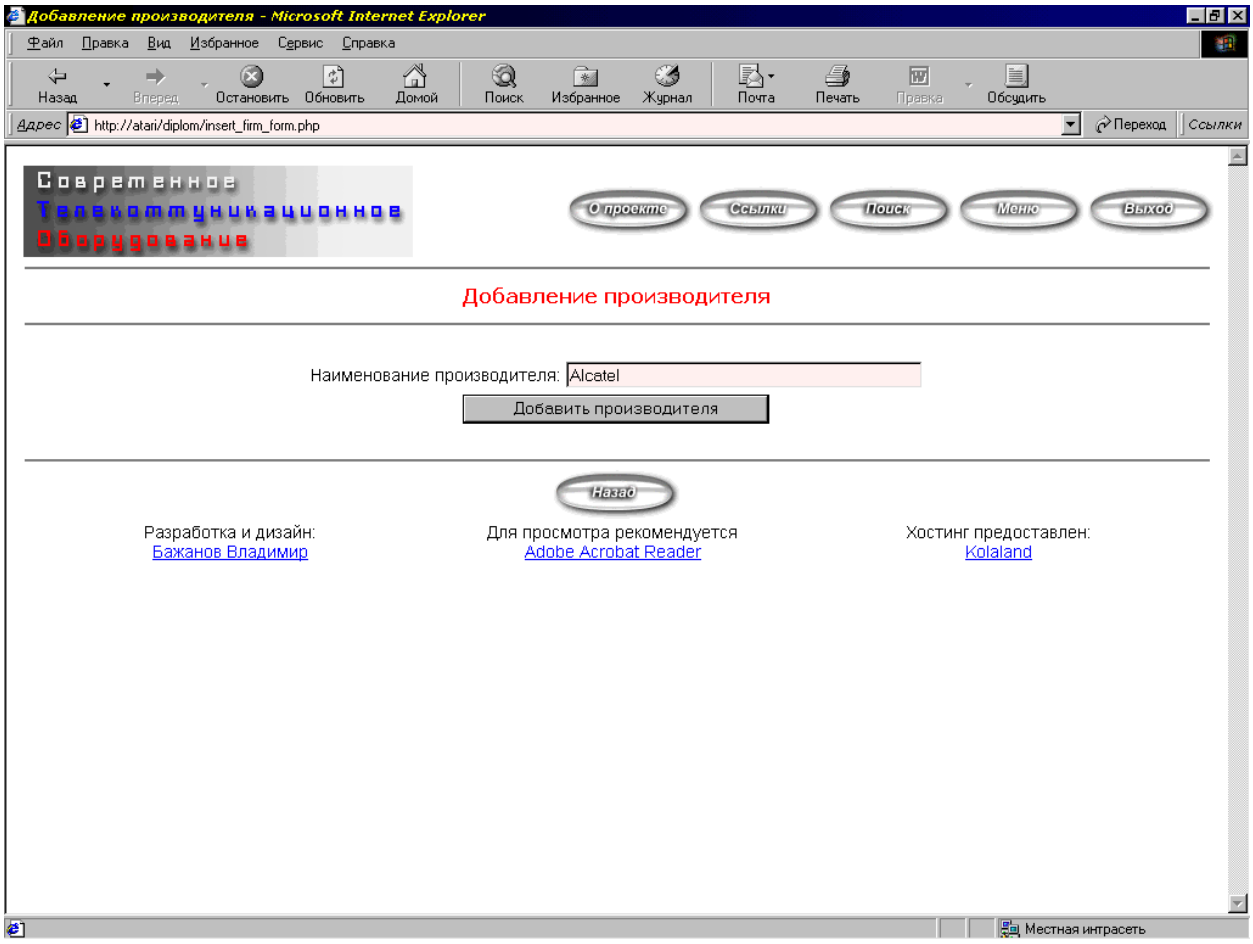

Рис. 7.15. Страница добавления нового производителя оборудования

Добавление типа оборудования или производителя оборудования не сможет быть осуществлено в случае, если в базе данных уже существуют имя типа или производителя, совпадающее с новым (добавляемым).

Меню администрирования также содержит кнопку «Новое устройство», позволяющую добавить новое телекоммуникационное устройство в базу данных. После нажатия на эту кнопку загрузится форма добавления устройства, представленная на рис. 7.16.

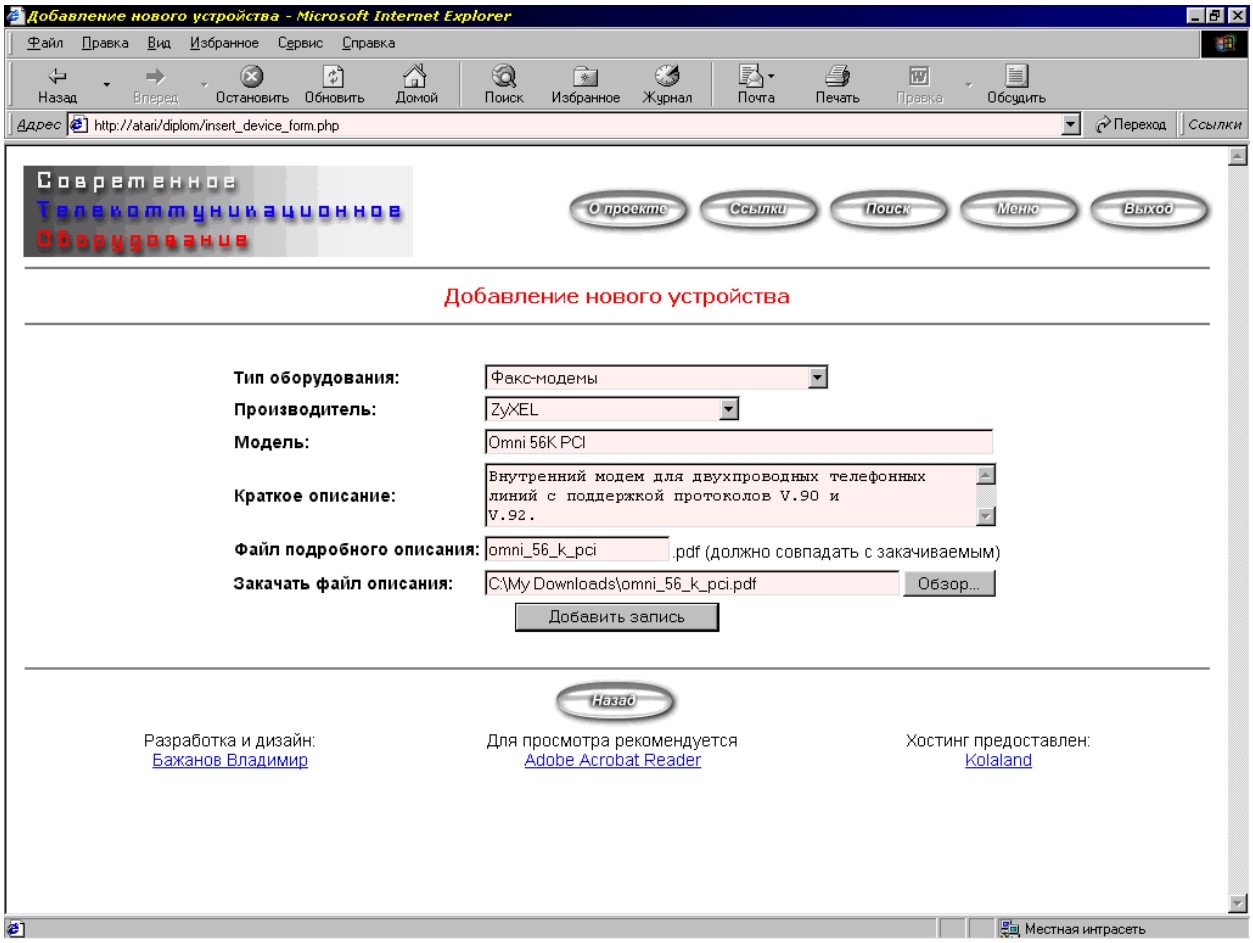

Рис. 7.16. Страница формы добавления нового устройства в базу данных

Для внесения информации форма содержит следующие поля:

- Тип оборудования;
- Производитель;
- Модель;
- Краткое описание;
- Файл подробного описания;
- Закачать файл описания.

Первые два поля представлены в виде списков, содержащих существующие записи типов оборудования и производителей оборудования.

Поле «Краткое описание» содержит краткую информацию о устройстве и служит для облегчения поиска телекоммуникационного устройства по ключевому слову.

В поле «Закачать файл описания» необходимо выбрать файл, который будет загружен с локального компьютера администратора на сайт, посредством кнопки «Обзор». В поле «Файл подробного описания» нужно ввести имя PDF-файла описания устройства. Это имя должно совпадать с именем файла, введенного в поле «Закачать файл описания», без указания дискового пути (см. рис. 7.16). Кнопка «Добавить запись» добавляет текущую информацию в базу данных, закачивает файл описания на сервер сайта и отображает страницу результатов добавления устройства.

Выход из системы администрирования осуществляется нажатием кнопки «Выход» в основном меню сайта. После появления сообщения о выходе из системы (рис. 7.17), дальнейшая работа с базой данных происходит в режиме пользователя.

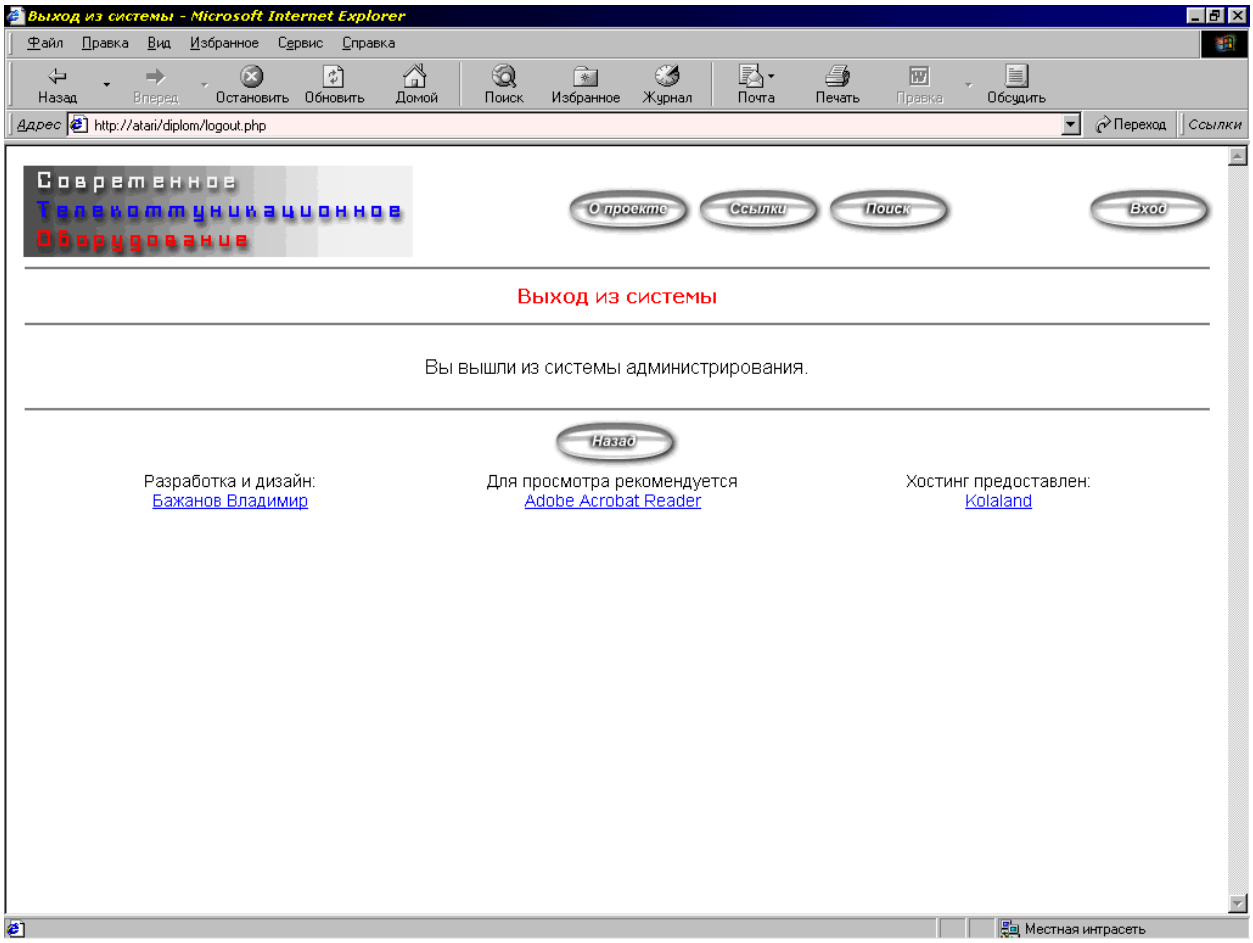

Рис. 7.17. Страница сообщения о выходе из системы администрирования

ЗАКЛЮЧЕНИЕ

В результате выполненной дипломной работы были реализованы следующие вопросы.

В дипломной работе решена задача создания и использования в учебном процессе кафедры ОПДС базы данных, содержащей сведения о современном телекоммуникационном оборудовании.

Была обоснована целесообразность наличия на кафедре ОПДС базы данных «Современное телекоммуникационное оборудование», разработаны требования к базе данных. Рассмотрены возможные способы построения базы данных и выбран наиболее рациональный способ ее построения. Разработана структура базы данных. Для реализации базы данных была выбрана СУБД MySQL и язык сценариев PHP Hypertext Preprocessor. Выполнено начальное информационное наполнение. Приведен порядок работы с базой данных.

СПИСОК ЛИТЕРАТУРЫ

1. Стаханов А. А., Linux. Наиболее полное руководство. - СПб.: БХВ-Петербург, 2003.
2. Немет Э., Снайдер Г., Хейн Т. и др., Руководство администратора Linux. - М.: Издательский дом "Вильямс", 2003.
3. Птицын К. А., Серверы Linux. Самоучитель. - М.: Издательский дом "Вильямс", 2003.
4. Колисниченко Д. Н., Самоучитель PHP 4/5. - Спб.: Наука и Техника, 2004.
5. Орлов А. А., PHP: Полезные приемы. -М.: Горячая линия-Телеком, 2004.
6. Томсон Л., Веллинг Л., Разработка Web-приложений на PHP и MySQL. - 2-е изд., испр. - СПб.: "ДиаСофтЮП", 2003.
7. Шкарина Л., Язык SQL: Учебный курс. - Спб.: Питер, 2001.
8. Бхамидипати К., SQL. Справочник программиста. - М.: Издательство ЭКОМ, 2003.
9. Стоунз Р., Мэттью Н., PostgreSQL. Основы. - Спб.: Символ-Плюс, 2002.
10. Глушаков С. В., Ломотьков Д. В., Базы данных. Учебный курс. – К.: Абрис, 2000.
11. Пригорьев Ю. А., Проблемы выбора доступа к данным при проектировании информационных систем на основе СУБД // Информационные технологии. – 1999, №5. стр. 4-10.
12. http://dklab.ru/chicken/web/ - сайт проекта Denver 2 - программного продукта для Web-разработчиков, содержащего Apache, PHP, MySQL и phpMyAdmin.
13. http://www.mysql.ru/ - MYSQL.RU - сайт Российских MySQL разработчиков.

14. http://www.apache.org/ - The Apache Software Founaion - сайт группы разработчиков HTTP-сервера Apache.
15. http://www.phpclub.ru/ - PHPClub Team - ресурс Российских PHP-разработчиков.
16. http://www.php.net/ - PHP Group - сайт разработчиков PHP.
17. http://www.citforum.ru/ - ресурс с огромным количеством документации по программированию.