

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ВЕБ-ПОРТАЛОВ.....	5
1.1 Изучение проблем создания информационных систем.....	5
1.2 Обзор существующего программного обеспечения.....	9
1.3 Требования к программному обеспечению.....	12
1.4 Роль человека в информационной системе.....	19
Выводы по главе.....	20
2 ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА.....	23
2.1 Архитектура программного продукта.....	23
2.2 Модульный состав пользовательской подсистемы.....	23
2.3 Концептуальная схема базы данных.....	24
2.4 Руководство программиста.....	26
2.5 Руководство пользователя.....	35
2.6 Тестирование.....	40
Выводы по главе.....	42
ЗАКЛЮЧЕНИЕ.....	43
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	45

ВВЕДЕНИЕ

В наше время существует большое количество специализированных автомобильных порталов, созданных для облегчения жизни автолюбителей, главным образом позволяющих сократить затрачиваемые силы и время пользователя на поиск необходимой ему информации в специализированной области.

Но, несмотря на большое количество существующих автомобильных сайтов, далеко не каждый из них является достаточно большой структурированной информационной системой, удовлетворяющей потребности пользователя в интересующей его информации. К тому же возможны случаи, когда ресурс, обладая большой информационной системой, не всегда выдает уместную и актуальную информацию, вследствие чего автолюбитель тратит большое количество времени на ее поиск. Это может быть обусловлено плохой структурированностью системы, вследствие чего она перестает справляться со своей главной задачей, а именно с хранением информации в удобном виде для конечного пользователя.

Данная работа позволит не только упростить процесс поиска информации в области технического обслуживания транспортного средства за счет создания большой структурированной информационной системы, но и уменьшить время, затрачиваемое на данную операцию. Систематизация большого количества данных в информационной системе предоставит автовладельцам ряд возможностей: узнать историю создания любимой марки/модели автомобиля; поделиться впечатлениями об эксплуатации технического средства передвижения; узнать технические особенности конкретной модели; получить помощь при выборе транспортного средства и др.

Таким образом, **целью** бакалаврской работы явилось повышение качества удовлетворения информационных потребностей автолюбителей за счет разработки информационного портала для автовладельцев.

Для достижения данной цели необходимо решить ряд **задач**:

- изучение предметной области;
- рассмотрение существующего программного обеспечения;

- выбор программных средств;
- разработка архитектуры программного продукта;
- проектирование базы данных;
- практическая реализация программного продукта;
- тестирование программного продукта.

Выпускная квалификационная работа состоит из введения, двух глав, заключения, списка использованных источников, и изложена на 46 страницах, содержащих 4 таблицы и 13 рисунков. В первой главе представлены теоретические сведения о принципах построения информационных систем, где дополнительно рассмотрено существующее программное обеспечение. Во второй главе изложен процесс разработки программного продукта, представлена его структура, руководство программиста, руководство пользователя, а также результат его тестирования.

1 ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ВЕБ-ПОРТАЛОВ

1.1 Изучение проблем создания информационных систем

1.1.1 Основные положения информационных систем

В зависимости от конкретной области применения информационные системы могут очень сильно различаться по своим функциям, архитектуре, реализации. Однако можно выделить, по крайней мере, два свойства, которые являются общими для всех информационных систем. Во-первых, любая информационная система предназначена для сбора, хранения и обработки информации. Поэтому в основе любой информационной системы лежит среда хранения и доступа к данным. Среда должна обеспечивать уровень надежности хранения и эффективность доступа, которые соответствуют области применения информационной системы [3, с. 12].

Во-вторых, информационные системы ориентируются на конечного пользователя, например, офисного работника. Такие пользователи могут быть очень далеки от мира компьютеров. Для них терминал, персональный компьютер или рабочая станция представляют собой всего лишь орудие их собственной профессиональной деятельности. Поэтому информационная система обязана обладать простым, удобным, легко осваиваемым интерфейсом, который должен предоставить конечному пользователю все необходимые для его работы функции, но в то же время не дать ему возможность выполнять какие-либо лишние действия. Иногда этот интерфейс может быть графическим: с меню, кнопками, подсказками и т.д. В любом случае наличие развитых интерфейсных средств является обязательным для любой современной информационной системы.

Архитектура системы клиент/сервер

Клиент-серверное приложение состоит из ПО сервера (в дальнейшем – сервер) и ПО клиентов (в дальнейшем – клиент).

Совокупность сервера и клиента называется домен (Рисунок 1). Клиент и сервер, по сути, являются равноправными компонентами (сервисами) работы с объектами системы. Они используют отдельную библиотеку, подключаемую для работы сервера и клиентского ПО. Объекты системы создаются на основании ранее описанных классов и хранятся в объектной базе данных (далее – ОБД). Методы объектов могут быть описаны на любом языке, который может быть транслирован в байт-код, в котором эти методы хранятся и используются в ОБД.

Взаимодействие сервера и клиентов осуществляется по определенному протоколу поверх протокола *TCP/IP*. По этому протоколу осуществляется получение объектов, вызов методов объектов на сервере или же скачивание байт-кода методов.

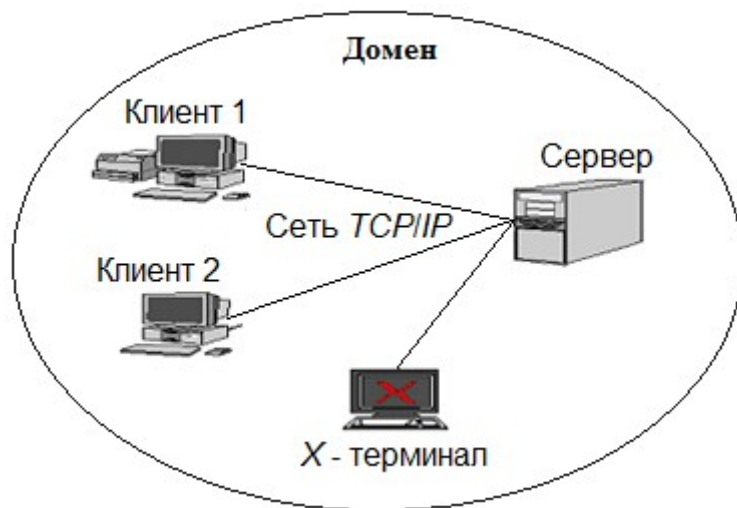


Рисунок 1 – Домен клиент/сервер

Принципы работы клиента с сервером

Основные принципы работы клиента с сервером заключаются в следующем:

- клиент после аутентификации получает форму своей рабочей области, которая содержит все необходимые инструменты для работы пользователя в соответствии с его правами доступа и привилегиями;
- при вызове определенной формы, она запрашивается на сервере. После получения формы (и необходимых методов для работы с ней локально), она адресовывается через драйвер форм;

- оператор работает с данными в форме, и события работы с формой обрабатываются через байт-машину и локальные или серверные методы работы с формой и объектом;

- помимо указанной работы с формой, оператор может получить печатную форму необходимого документа или отчет (посредством вызова определенного метода формирования печатных форм). Этот документ может быть распечатан или экспортирован в другую программу.

Клиент/серверная архитектура предполагает распределение задач и вычислительной нагрузки по нескольким компьютерам, объединенным в сеть. Программное обеспечение информационной системы, реализованной в этой архитектуре, логически разделяется на серверное ПО, ПО среднего уровня и ПО клиентских рабочих мест (клиентов). При этом программное обеспечение клиентов отвечает за презентационные функции и взаимодействие с пользователем, средний уровень – за выполнение логических правил и обработку информации, идущей к клиенту или от клиента, а сервер обеспечивает безопасное и надежное хранение данных и осуществление выбора данных по запросам [7, с. 21].

В логической трехуровневой клиент/серверной архитектуре каждый уровень представлен отдельным программным модулем. Модули могут устанавливаться как на три и более отдельные машины, так и на две, в зависимости от обстоятельств.

Реализация информационных систем в архитектуре клиент/сервер обеспечивает:

- надежность хранения информации – аппаратные сбои или ошибочные действия персонала не приведут к потере данных;
- целостность информации – встроенные механизмы существенно снижают вероятность возникновения коллизий при вводе и редактировании данных;
- регламентированный доступ к информации – каждый пользователь может работать только с теми данными, к которым он получил доступ, и только в том режиме, который соответствует данным ему правам;
- возможность безопасного подключения к глобальным сетям;
- многопользовательский режим работы с информацией, в том числе и удаленных пользователей;

- высокую производительность и снижение нагрузки на сеть за счет распределения процессов между серверами и рабочими станциями;
- минимизацию сетевого трафика;
- масштабируемость – производительность системы растет пропорционально числу процессоров и оперативной памяти серверов;
- оптимизацию распределения вычислительной нагрузки между сервером и клиентом.

1.1.2 Обзор проблем, возникающих при проектировании информационных систем

Тенденции развития современных информационных технологий приводят к постоянному возрастанию сложности информационных систем (ИС), создаваемых в различных областях экономики. Современные крупные проекты ИС характеризуются, как правило, следующими особенностями:

- сложность описания (достаточно большое количество функций, процессов, элементов данных и сложные взаимосвязи между ними), требующая тщательного моделирования и анализа данных и процессов;
- наличие совокупности тесно взаимодействующих компонентов (подсистем), имеющих свои локальные задачи и цели функционирования (например, традиционных приложений, связанных с обработкой транзакций и решением регламентных задач, и приложений аналитической обработки (поддержки принятия решений), использующих нерегламентированные запросы к данным большого объема);
- отсутствие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений и прикладных систем;
- необходимость интеграции существующих и вновь разрабатываемых приложений;
- функционирование в неоднородной среде на нескольких аппаратных платформах;
- разобщенность и разнородность отдельных групп разработчиков по уровню квалификации и сложившимся традициям использования тех или иных инструментальных средств;

- существенная временная протяженность проекта, обусловленная, с одной стороны, ограниченными возможностями коллектива разработчиков, и, с другой стороны, масштабами организации-заказчика и различной степенью готовности отдельных ее подразделений к внедрению ИС [2, с. 14].

Для успешной реализации проекта объект проектирования (ИС) должен быть, прежде всего, адекватно описан, должны быть построены полные и непротиворечивые функциональные и информационные модели ИС. Накопленный к настоящему времени опыт проектирования ИС показывает, что это является логически сложной, трудоемкой и длительной по времени работой, требующей высокой квалификации участвующих в ней специалистов. Однако до недавнего времени проектирование ИС выполнялось в основном на интуитивном уровне с применением неформализованных методов, основанных на искусстве, практическом опыте, экспертных оценках и дорогостоящих экспериментальных проверках качества функционирования ИС. Кроме того, в процессе создания и функционирования ИС информационные потребности пользователей могут изменяться или уточняться, что еще более усложняет разработку и сопровождение таких систем.

1.2 Обзор существующего программного обеспечения

На сегодняшний день информационные технологии широко используются во всем мире и позволяют найти необходимую информацию в разы быстрее. Также происходит постоянный рост количества автомобилистов и транспортных средств. Автомобиль уже является не роскошью, которой являлся буквально 60 лет назад, а средством передвижения. Он усложняется в техническом плане, и обслуживать его становится гораздо тяжелее, так как количество запчастей, которое входит в его состав, увеличивается. Для упрощения жизни автомобилистов, ищущих запчасти, вышедшие из строя, были созданы специализированные интернет-магазины, которые позволяют найти и заказать необходимые детали, не выходя из дома. Владелец остается лишь оплатить и забрать запчасти в определенном месте.

Большое количество моделей транспортных средств породило их огромное разнообразие, которое позволяет человеку выбрать удобный в использовании и привлекательный автомобиль за разумные для него денежные средства. Также, благодаря данному разнообразию, становится возможным менять автомобиль гораздо чаще, чем это было раньше, продавая старое транспортное средство новому хозяину. Для продажи таких автомобилей создаются специализированные порталы. Среди таких порталов можно выделить www.24auto.ru, www.drom.ru, www.auto.ru. Из «большой тройки» первый и второй веб-порталы пользуются большей популярностью в Красноярском крае.

Сравнительная характеристика автомобильных порталов представлена в таблице 1.

Таблица 1 – Сравнительная характеристика автомобильных порталов

Функциональные возможности	<i>www.24auto.ru</i>	<i>www.drom.ru</i>	<i>www.auto.ru</i>
Вариант технологии взаимодействия	сетевой	сетевой	сетевой
Операционная система	<i>Mac OS, UNIX-подобные системы, Windows 95 и выше, Android, iOS</i>	<i>Mac OS, UNIX-подобные системы, Windows 95 и выше, Android, iOS</i>	<i>Mac OS, UNIX-подобные системы, Windows 95 и выше, Android, iOS</i>
Минимальные требования к ПО	веб-браузер: <i>Internet Explorer 6, Mozilla Firefox 14.0, Opera 8.0, Google Chrome 9.0, Safari 6.0</i>	веб-браузер: <i>Internet Explorer 6, Mozilla Firefox 14.0, Opera 8.0, Google Chrome 9.0, Safari 6.0</i>	веб-браузер: <i>Internet Explorer 6, Mozilla Firefox 14.0, Opera 8.0, Google Chrome 9.0, Safari 6.0</i>
Дополнительные требования	наличие подключения к интернету	наличие подключения к интернету	наличие подключения к интернету
Обновление системы	интернет	интернет	интернет
Территория охвата	Красноярский край	Россия	Россия
Среднее количество посетителей в день (тыс. человек)	85	1 200	600
Среднее количество просмотров в день (млн.)	2,66	35	-
Позиция в рейтинге <i>top.mail.ru</i> по количеству посетителей в Красноярском крае на апрель 2014 г.	333	8 470	135 910
Индекс популярности по версии <i>Rambler</i>	9 182	100 893	97 871
Количество объявлений	33 284	-	336 561
Наличие собственного телеканала	-	-	<i>www.autoru.tv</i>

Проанализировав статистические данные трех самых популярных автомобильных порталов, можно сделать вывод, что самым популярным в Красноярском крае является информационный портал www.24auto.ru, так как он непосредственно на него и ориентирован (Рисунок 2).



Рисунок 2 – Автомобильный портал «www.24auto.ru»

Безоговорочным лидером по количеству посещений, просмотров и зарегистрированных пользователей в стране является автомобильный портал www.drom.ru, имеющий огромный функционал для упрощения жизни автомобилиста (Рисунок 3).

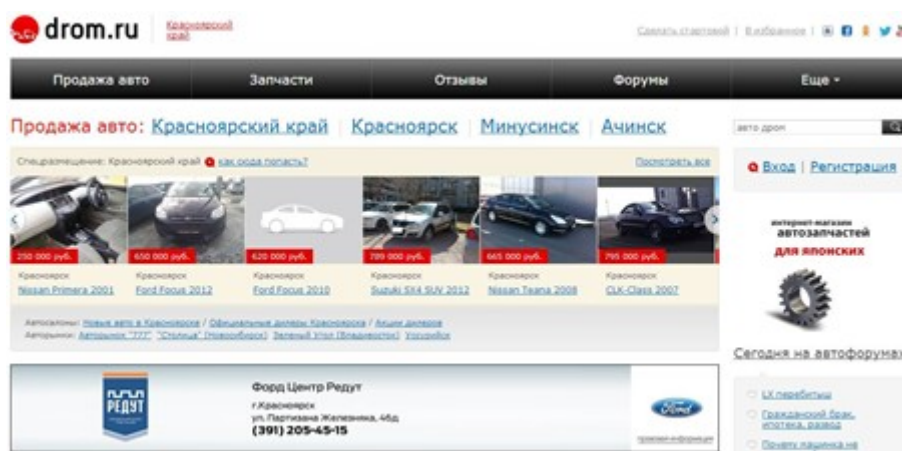


Рисунок 3 – Автомобильный портал «www.drom.ru»

Веб-ресурс www.auto.ru был ведущим лидером в данной области, являясь одним из первых ресурсов такого плана, но из-за быстрой тенденции к развитию автомобильных порталов, его лидирующая позиция была утеряна (Рисунок 4).

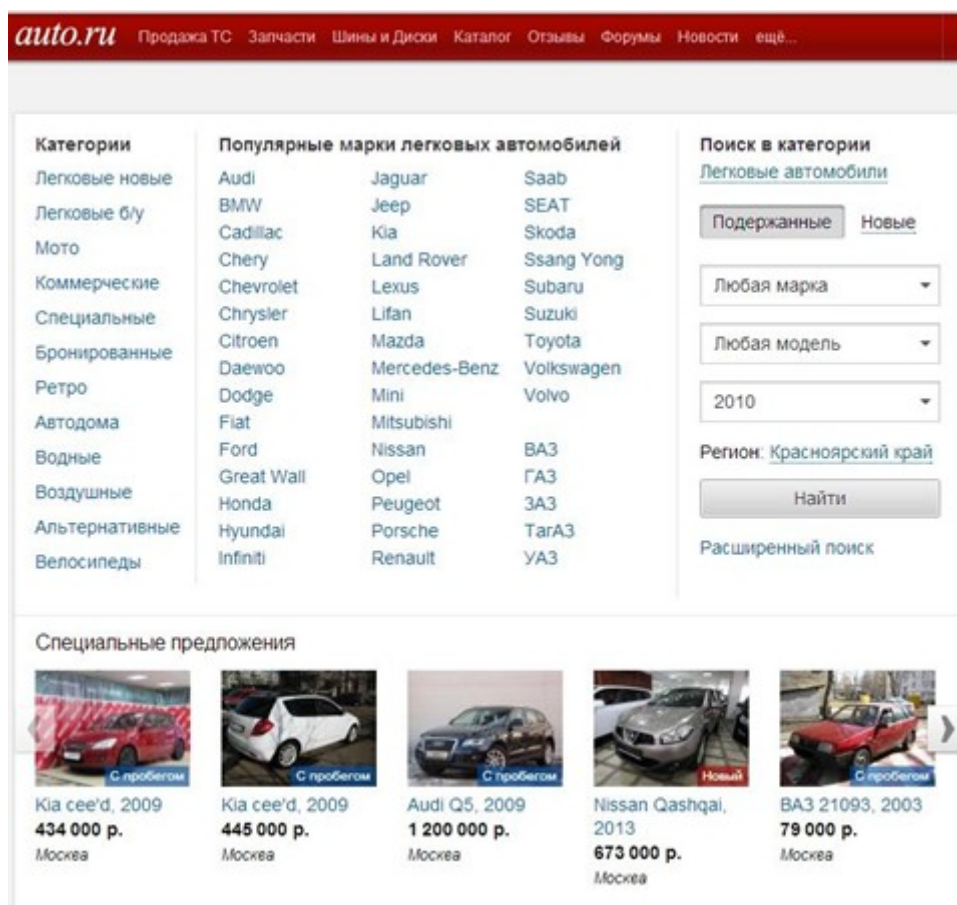


Рисунок 4 – Автомобильный портал «www.auto.ru»

1.3 Требования к программному обеспечению

1.3.1 Выбор целевой СУБД

Выбор целевой СУБД – это выбор СУБД (система управления базами данных) подходящего типа, предназначенной для поддержки создаваемого приложения базы данных.

Наиболее подходящим моментом для выбора целевой СУБД является промежуточное положение между концептуальной и логической фазами проектирования базы данных. Однако, этот выбор можно осуществить в любой другой момент времени до начала логического проектирования, при условии, что имеется вся необходимая информация о таких общих требованиях к системе, как производительность, простота реорганизации, уровень защищенности и ограничения целостности данных [1, с. 68].

Существует несколько вариантов целевых СУБД, основными из которых на сегодняшний день являются *Oracle Database*, *Microsoft Access*, *MySQL*, *PostgreSQL*.

Oracle Database

Oracle Database – платная объектно-реляционная СУБД от компании *Oracle*, которая имеет следующие особенности:

- многоверсионность данных для управления параллельными транзакциями – механизм обеспечения одновременного конкурентного доступа к БД, заключающийся в предоставлении каждому пользователю «снимка» БД, обладающего тем свойством, что вносимые пользователем изменения в БД невидимы другим пользователям до момента фиксации транзакции. Таким образом, происходит защита от блокирования взаимно противоположных транзакций;
- секционирование – разделение хранимых объектов БД на отдельные части с раздельными параметрами физического хранения. Данный метод повышает управляемость и производительность больших БД;
- возможность автономных транзакций – метод управления транзакциями, когда позволено создание новых подтранзакций, сохраняющих или отменяющих изменения вне зависимости от родительской транзакции;
- автоматическое управление хранением файлов БД;
- наличие аналитических функций в *SQL* – расширение *SQL*;
- возможность ретроспективных запросов, позволяющих запрашивать данные так, как если бы это было сделано в заданный момент времени в прошлом;
- наличие ПО для кластеризации и повышения доступности;
- наличие объектно-ориентированных свойств;
- автоматический мониторинг и диагностика БД для выявления проблем производительности и, возможно, автоматической корректировки.

Microsoft Access

Microsoft Access – проприетарная интерактивная реляционная СУБД для *Windows* от компании *Microsoft*, которая имеет следующие особенности:

- наличие широкого спектра функций (связанные запросы, связь с внешними таблицами и БД);

- наличие встроенного языка *Visual Basic for Applications (VBA)*, что позволяет в самом *Access* писать приложения, работающие с БД;
- наличие построителя таблиц, экранных форм, *SQL*-запросов, отчетов, выводимых на печать;
- отсутствие ряда механизмов, необходимых в многопользовательских БД, таких, например, как триггеры.

PostgreSQL

PostgreSQL – свободная объектно-реляционная СУБД для множества *Unix*-подобных операционных систем, а также для *Windows*, которая следующие достоинства:

- поддержка БД практически неограниченного размера;
- мощные и надежные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования;
- наследование;
- надежность;
- легкая расширяемость.

Также данная СУБД имеет описанные ниже основные возможности:

- наличие функций – блоков кода, исполняемых на сервере, а не на клиенте БД;
- наличие триггеров – функций особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено действием по модификации данных (добавление, удаление, изменение);
- наличие правил и представлений;
- возможность поддержки разных типов индексов (*B*-дерево, хэш, *R*-дерево и т.п.), а также создания новых;
- многоверсионность данных для управления параллельными транзакциями;
- поддержка большого набора встроенных типов, данных;
- возможность добавления пользовательских объектов (функций, типов данных, операторов, процедурных языков и т.п.);
- контроль целостности;

- репликация – механизм синхронизации содержимого нескольких копий объекта;

- наличие аналитических функций;
- поддержка регулярных выражений *Perl*.

MySQL

MySQL – свободная реляционная СУБД, разработку и поддержку которой осуществляет корпорация *Oracle* и которая имеет следующие особенности:

- является решением для малых и средних приложений;
- входит в состав серверов *WAMP*, *AppServ*, *LAMP* и портативные сборки серверов *Денвер*, *XAMPP*;

- является гибкой за счет поддержки большого количества типов таблиц;
- ориентирована на большое количество *Unix*-подобных платформ, а также на *Windows*;

- имеет интерфейс программирования приложений (*API* – *Application Programming Interface*) для наиболее популярных языков программирования (*Delphi*, *C*, *C++*, *Java*, *Perl*, *PHP*, *Python*, *Ruby*, *Smalltalk* и др.), библиотеки для языков платформы *.NET*;

- отличается быстротой и стабильностью;
- отличается простотой использования, а также исключительно скромными требованиями в отношении дискового пространства и памяти;

- доступна как в версии бесплатного ПО, так и в коммерческом исполнении;
- поддерживает следующие важные возможности: транзакции, блокирование на уровне строк, внешние ключи, подзапросы и полнотекстовый поиск, использование сохраненных процедур;

- хорошо проверена и надежна;
- используется целым рядом очень требовательных заказчиков, таких как *Yahoo!*, *Finance*, *Slashdot* и Бюро переписи США (*U.S. Census Bureau*).

Таким образом, после детального изучения всевозможных вариантов для разрабатываемой БД была выбрана СУБД *MySQL*, так как она достаточно проста в использовании, имеет скромные требования в отношении дискового пространства и памяти и удовлетворяет следующим требованиям: физическим параметрам, доступности, обработки транзакций, утилитам и средствам разработки.

1.3.2 Выбор языка программирования

PHP (англ. *PHP: Hypertext Preprocessor* – «*PHP*: препроцессор гипертекста», англ. *Personal Home Page Tools* (устар.) – «Инструменты для создания персональных веб-страниц») – язык программирования, созданный для генерирования *HTML*-страниц на веб-сервере и работы с базами данных. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров. Входит в *LAMP* – распространенный набор для создания веб-сайтов (*Linux, Apache, MySQL, PHP (Python)*) [4, с. 132].

Одним из главных достоинств *PHP* является тот факт, что он внедряется прямо в *HTML*-код, поэтому программисту не приходится писать программу с множеством команд для простого вывода *HTML*. Код *HTML* и *PHP* можно чередовать по мере необходимости.

Характеристики PHP

Главным фактором при проектировании языка *PHP* является практичность. *PHP* должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер *PHP* обусловлен пятью важными характеристиками:

- классическим *C*-подобным синтаксисом;
- простотой;
- эффективностью;
- безопасностью;
- гибкостью.

Существует еще одна «характеристика», которая делает *PHP* особенно привлекательным: он распространяется свободно.

Традиционность

Язык *PHP* кажется знакомым программистам, работающим в разных областях. Многие конструкции языка позаимствованы из *C* и *Perl*, а нередко код *PHP* практически неотличим от того, что встречается в типичных программах *C* или *Pascal*. Это заметно снижает начальные усилия при изучении *PHP*.

Простота

Сценарий *PHP* может состоять из 10 000 строк или из одной строки – все зависит от специфики вашей задачи. Не придется подгружать библиотеки, указывать специальные параметры компиляции или что-нибудь в этом роде. Механизм *PHP* просто начинает выполнять код после первой экранирующей последовательности (`<?`) и продолжает выполнение до того момента, когда он встретит парную экранирующую последовательность (`?>`). Если код имеет правильный синтаксис, он исполняется в точности так, как указал программист.

Эффективность

Эффективность является исключительно важным фактором при программировании для многопользовательских сред, к числу которых относится и *www*. В *PHP* 5.3 был реализован механизм выделения ресурсов и обеспечена улучшенная поддержка объектно-ориентированного программирования, а также средства управления сеансом [6, с. 129].

Безопасность

PHP предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории:

- средства безопасности системного уровня. В *PHP* реализованы механизмы безопасности, находящиеся под управлением администраторов; при правильной настройке *PHP* это обеспечивает максимальную свободу действий и безопасность. *PHP* может работать в так называемом безопасном режиме (*safe mode*), который ограничивает возможности применения *PHP* пользователями по ряду важных показателей. Например, можно ограничить максимальное время выполнения и использование памяти (неконтролируемый расход памяти отрицательно влияет на быстродействие сервера). По аналогии с *cgi-bin* администратор также может устанавливать ограничения на каталоги, в которых пользователь может просматривать и исполнять сценарии *PHP*, а также использовать сценарии *PHP* для просмотра конфиденциальной информации на сервере (например, файла *passwd*);

- средства безопасности уровня приложения. В стандартный набор функций *PHP* входит ряд надежных механизмов шифрования. *PHP* также совместим со многими приложениями независимых фирм, что позволяет легко интегрировать его с защищенными технологиями электронной коммерции (*e-commerce*). Другое преимущество заключается в том, что исходный текст сценариев *PHP* нельзя просмотреть в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя. Реализация *PHP* на стороне сервера предотвращает похищение нетривиальных сценариев пользователями, знаний которых хватает хотя бы для выполнения команды *View Source* [6, с. 130].

Гибкость

Поскольку *PHP* является встраиваемым (*embedded*) языком, он отличается исключительной гибкостью по отношению к потребностям разработчика. Хотя *PHP* обычно рекомендуется использовать в сочетании с *HTML*, он с таким же успехом интегрируется и в *JavaScript*, *WML*, *XML* и другие языки. Кроме того, хорошо структурированные приложения *PHP* легко расширяются по мере необходимости (впрочем, это относится ко всем основным языкам программирования).

Нет проблем и с зависимостью от браузеров, поскольку перед отправкой клиенту сценарии *PHP* полностью компилируются на стороне сервера. В сущности, сценарии *PHP* могут передаваться любым устройствам с браузерами, включая сотовые телефоны, электронные записные книжки, пейджеры и портативные компьютеры. Программисты, занимающиеся вспомогательными утилитами, могут запускать *PHP* в режиме командной строки [6, с. 131].

Поскольку *PHP* не содержит кода, ориентированного на конкретный *web*-сервер, пользователи не ограничиваются определенными серверами (возможно, неизвестными для них). *Apache*, *Microsoft IIS*, *Netscape Enterprise Server*, *Stronghold* и *Zeus* – *PHP* работает на всех перечисленных серверах. Поскольку эти серверы работают на разных платформах, *PHP* в целом является платформенно-независимым языком и существует на таких платформах, как *UNIX*, *Linux*, *Mac*, *Solaris*, *FreeBSD* и *Windows XP/7* [6, с. 132].

Наконец, средства *PHP* позволяют программисту работать с внешними компонентами, такими как *Enterprise Java Beans* или *COM*-объекты *Win32*. Благодаря этим новым возможностям *PHP* занимает достойное место среди современных технологий и обеспечивает масштабирование проектов до необходимых пределов.

Свободное распространение исходных текстов программ в массах оказало, несомненно, благотворное влияние на многие проекты, в первую очередь – *Linux*, хотя и успех проекта *Apache* сильно подкрепил позиции сторонников *Open Source*. Сказанное относится и к истории создания *PHP*, поскольку поддержка пользователей со всего мира оказалась очень важным фактором в развитии проекта *PHP*.

Принятие стратегии *Open Source* и бесплатное распространение исходных текстов *PHP* оказало неоценимую услугу пользователям. Вдобавок, отзывчивое сообщество пользователей *PHP* является своего рода «коллективной службой поддержки», и в популярных электронных конференциях можно найти ответы даже на самые сложные вопросы.

Таким образом, основываясь на вышесказанном, можно сделать вывод, что *PHP* является хорошим выбором для создания *web*-приложений.

1.4 Роль человека в информационной системе

Пользователями данной информационной системы могут быть следующие группы лиц:

- автолюбитель;
- страховой агент;
- компании, занимающиеся продажей транспортных средств;
- лица, занимающиеся продажей транспортных средств.

Каждая группа может преследовать свои собственные интересы в процессе использования информационной системы.

Автолюбитель

Данный тип пользователей может быть заинтересован в автомобильном портале, как источнике знаний, откуда он может почерпнуть новую информацию о тонкостях эксплуатации транспортных средств; поделиться своими эмоциями, знаниями и рассуждениями с другими пользователями; узнать что-то из последних новостей автомобильного мира и т.п.

Страховой агент

Тип пользователей, который может воспользоваться упрощенной системой расчета ОСАГО для автоматизации процесса подсчета стоимости страховки автомобильного средства. Данная система позволяет совершить расчет без знания необходимых коэффициентов и формул.

Компании, занимающиеся продажей транспортных средств

Компания, желающая увеличить свою прибыль и реализовать свою продукцию, может воспользоваться системой продажи транспортных средств, либо автозапчастей.

Лица, занимающиеся продажей транспортных средств

Группа лиц, которые могут воспользоваться системой для поиска нового автомобиля с целью покупки, либо продажи своего старого транспортного средства. Также данный тип пользователей может быть заинтересован в поиске запчастей для своего средства передвижения, либо в поиске ответа на вопрос, связанного с эксплуатацией технического средства передвижения.

Таким образом, человек играет очень важную роль в функционировании информационной системы. Можно сказать, что без человека, как источника и потребителя информации, существование информационной системы бессмысленно.

Выводы по главе

Данная глава раскрыла основные принципы построения информационных систем и проблемы, которые могут появиться в процессе их разработки.

Также на основе рассмотрения достоинств и недостатков существующего ПО были выбраны следующие программные средства, необходимые для создания автомобильного портала:

- СУБД – *MySQL*;
- основной язык программирования – *PHP*.

К тому же были рассмотрены возможные группы пользователей информационной системы, а также сделан вывод о роли человека в ее работе.

2 ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА

2.1 Архитектура программного продукта

Архитектура программного продукта представлена на рисунке 5. Основной программной подсистемой является пользовательская подсистема. Состав модулей и управляющие связи пользовательской подсистемы (т. е. иерархия вызовов модулей) показаны на рисунке 6. В качестве модели управления выбрана модель централизованного управления (модель «вызов–возврат»). Административная подсистема предназначена для управления полномочиями доступа различных пользователей к базе данных из программных модулей. Эта подсистема опирается на группу так называемых административных таблиц базы данных. В этих таблицах хранятся метаданные о пользователях программного продукта. При запуске каждого модуля пользовательской подсистемы выполняется проверка полномочий доступа конкретного пользователя именно на основе этих таблиц.

В базе данных создаются таблицы, которые условно можно разделить на три группы. Кроме административных таблиц есть также справочные таблицы, предназначенные для хранения справочных данных (история создания марки транспортного средства, эксплуатационные показатели и др.), и основные таблицы, предназначенные для хранения общей информации. Все три группы таблиц базы данных имеют свои рабочие таблицы. В рабочих таблицах хранится текущая версия данных, с этими таблицами работает пользователь, он получает данные именно из этих таблиц.

Для облегчения и ускорения процесса ввода информации используется скрипт, совершающий сбор данных в автоматическом режиме.

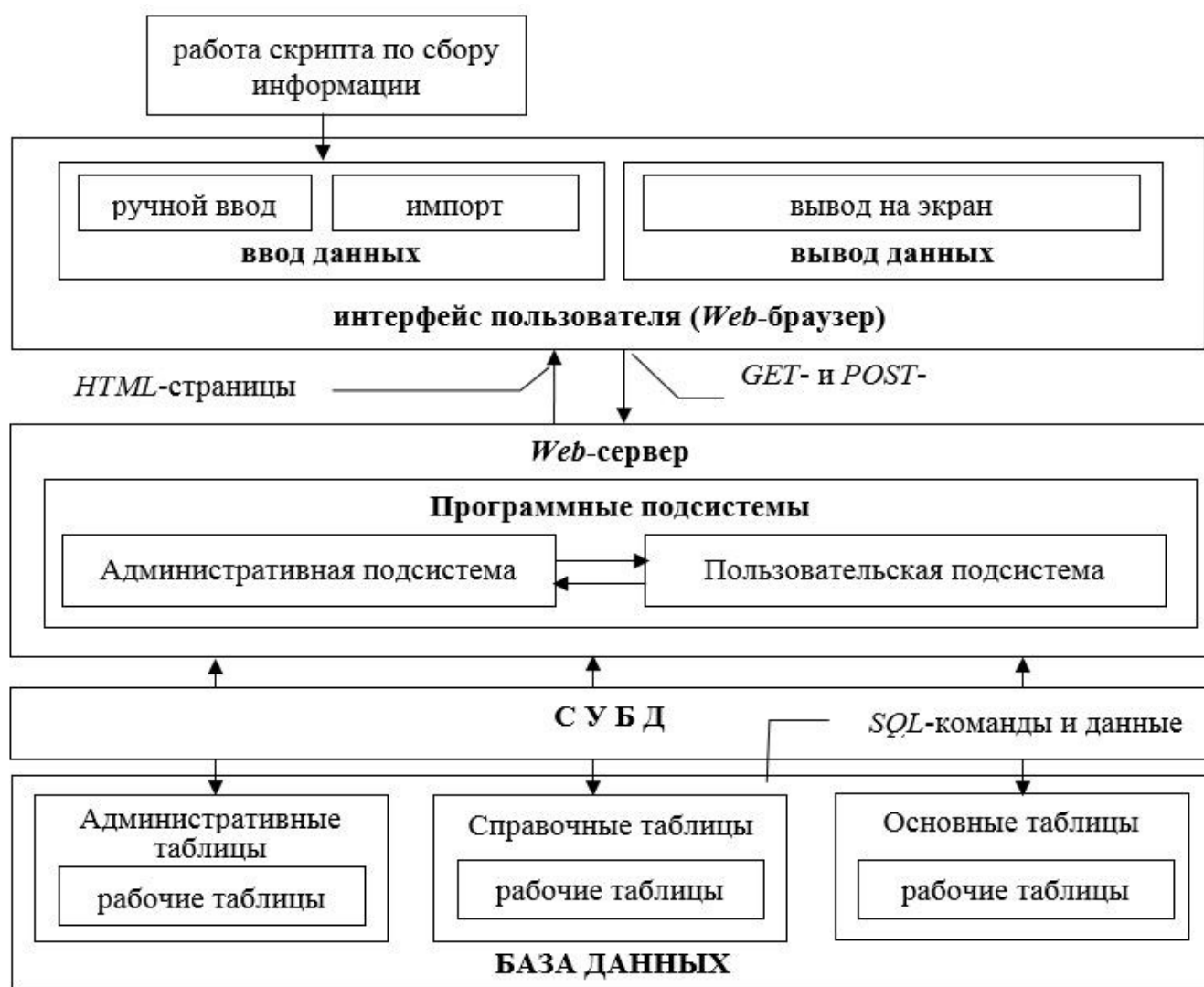


Рисунок 5 – Архитектура программного продукта

2.2 Модульный состав пользовательской подсистемы

На рисунке 6 стрелками указаны направления вызовов модулей (от вызывающего модуля к вызываемому). Справочники вызываются не только из главного модуля, но также и из модуля, отвечающего за размещения объявления о продаже транспортного средства. Кроме того, справочники могут вызываться один из другого.

В состав справочников на входят (Рисунок 6):

- города;
- виды марок транспортных средств;
- виды моделей транспортных средств;
- виды поколений модели транспортного средства;

- виды модификаций модели транспортного средства;
- виды эксплуатационных показателей для модели транспортного средства;
- виды коэффициентов для расчета стоимости полиса ОСАГО;
- виды коэффициентов для расчет стоимости автокредита.

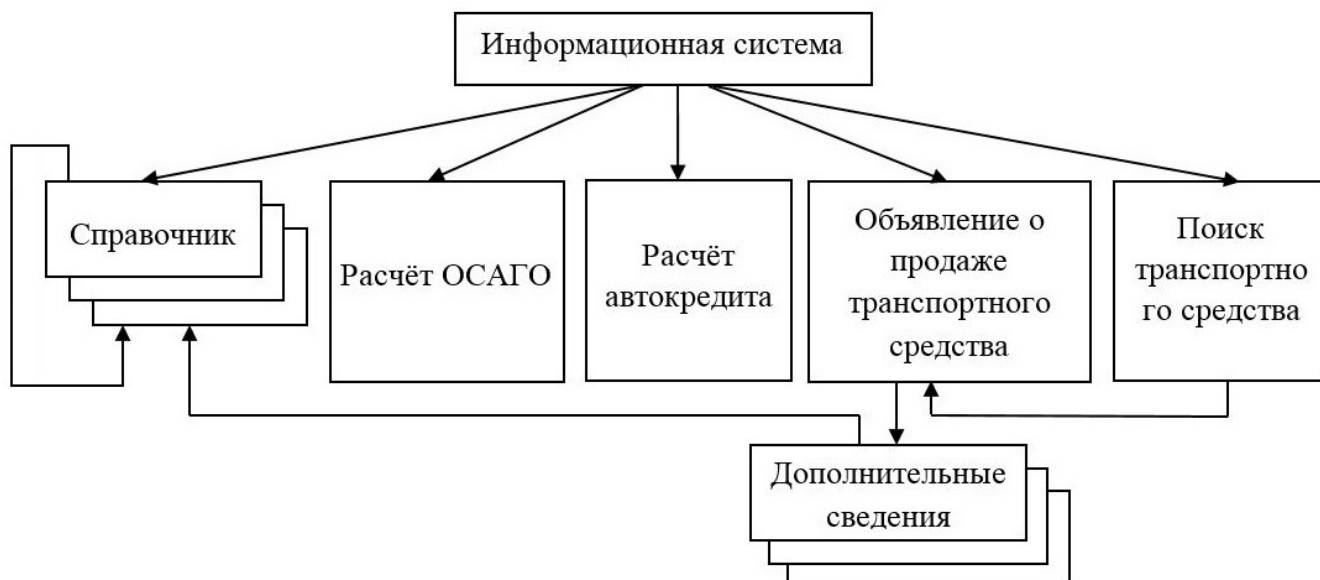


Рисунок 6 – Структура вызовов модулей пользовательской подсистемы

2.3 Концептуальная схема базы данных

Описания типов сущностей программного продукта представлены в таблице 2.

Таблица 2 – Сведения о типах сущностей информационной системы

№	Название сущности	Описание	Тип
1	Портал	-	-
2	Администратор	содержит информацию о администраторе (логин, пароль)	сильный
3	Покупатель	содержит сведения о покупателе (логин, пароль, дата регистрации, дата рождения, город и т.д.)	сильный
4	Продавец	содержит сведения о продавце (логин, пароль, дата регистрации, список продаваемых автомобилей, дата рождения, город и т.д.)	сильный
5	Заказ	содержит сведения о заказываемой запчасти (марка и модель автомобиля, кузов, двигатель, группа запчастей, наименование запчасти)	слабый
6	Заказы из запасов	содержит сведения о наличии запасов на складе (количество единиц запчастей на складе)	слабый
7	Интернет-магазин	содержит информацию о запчасти (стоимость, наличие, сроки доставки)	слабый

Окончание таблицы 2

8	Поиск модели	содержит информацию о найденном автомобиле (год выпуска, город продажи, цена, пробег, тип кузова, расположение руля, объем двигателя, тип привода, тип КПП)	слабый
9	Описание автомобиля	содержит информацию о продаваемом автомобиле (марка, модель, год выпуска, стоимость, пробег, объем двигателя, тип КПП и т.д.)	слабый
10	Расчет ОСАГО	содержит информацию о стоимости полиса ОСАГО (коэффициенты: возраст водителя, стаж, тип транспортного средства, город проживания)	слабый

11	Расчет автокредита	содержит информацию о сумме кредите и размере ежемесячных выплат (информация о процентной ставке в зависимости от выбранного срока кредита, ежемесячный платеж, итоговая сумма кредита)	слабый
----	--------------------	---	--------

На основе данных типов сущностей была разработана *ER*-диаграмма, представляющая их взаимодействие между собой (Рисунок 7).

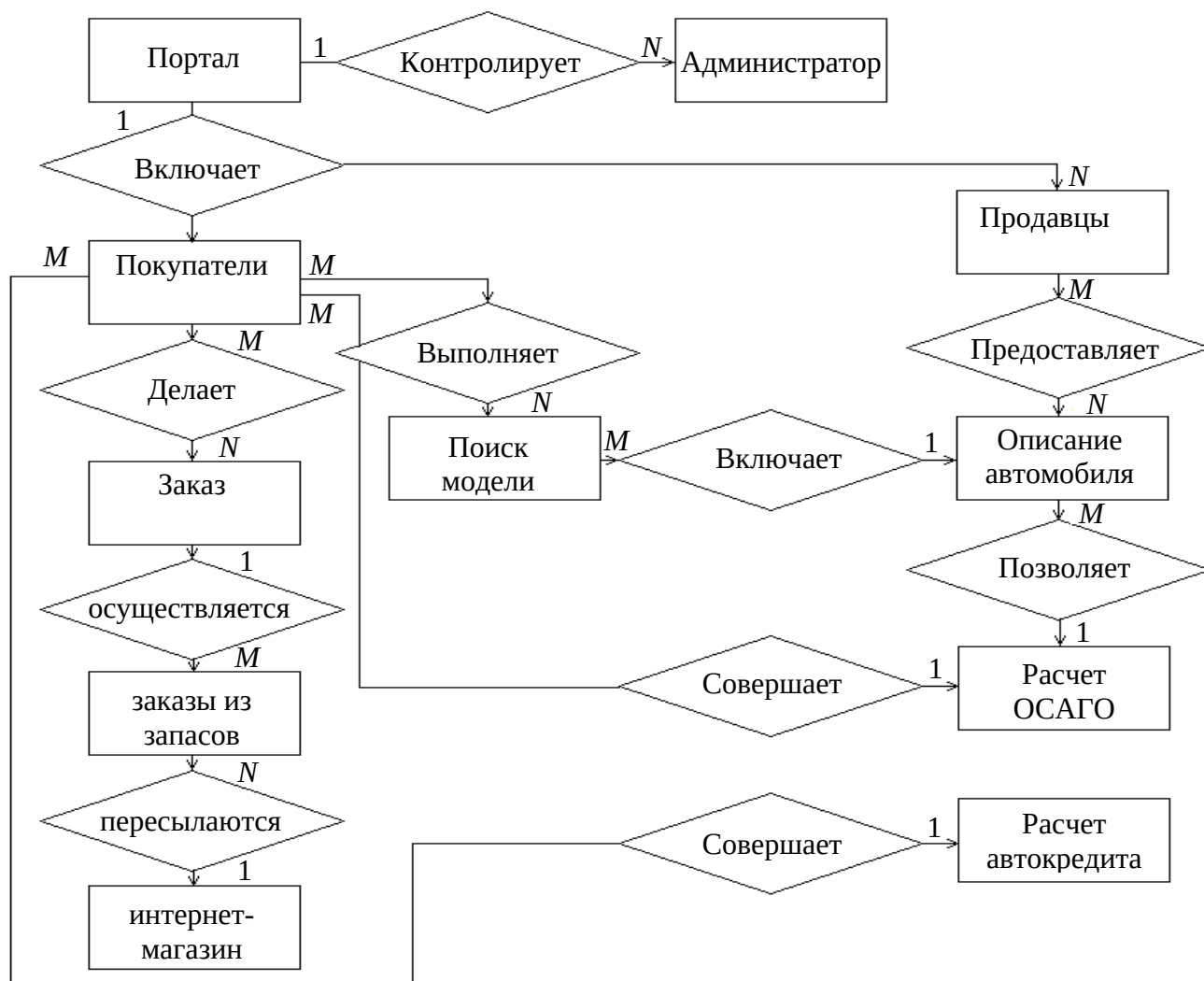


Рисунок 7 – *ER*-диаграмма информационного портала

Из полученной *ER*-диаграммы были определены типы связей между сущностями, которые описаны в таблице 3.

Таблица 3 – Сведения о сущностях автомобильного портала

№	Тип сущности	Тип связи	Тип сущности	Кардинальность
1	Портал	контролирует	администратор	1:N
2	Портал	включает	покупатели	1:N
3	Портал	включает	продавцы	1:N
4	Покупатели	выполняет	поиск модели	M:N
5	Покупатели	велаает	заказ	M:N
6	Покупатели	вовершает	расчет ОСАГО	M:1
7	Заказ	осуществляется	заказы из запасов	1:N
8	Заказы из запасов	пересылаются	интернет-магазин	N:1
9	Поиск модели	включает	описание автомобиля	M:1
10	Описание автомобиля	позволяет	расчет ОСАГО	M:1
11	Продавцы	предоставляют	описание автомобиля	M:N

12	Покупатели	совершает	расчет автокредита	<i>М:1</i>
----	------------	-----------	--------------------	------------

2.4 Руководство программиста

2.4.1 Общие сведения

В данном разделе освещается следующая информация: назначение и функции, выполняемые программой; условия, необходимые для выполнения программы (объем оперативной памяти, требование к составу и параметрам периферийных устройств, требования к программному обеспечению и т.п.); описание основных характеристик и особенностей программы; организация используемой входной и выходной информации. Продукт поддерживает интерфейс пользователя на основе меню, ориентированный на использование как мыши, так и клавиатуры.

Назначение программы

Данный программный продукт предназначен для помощи автомобилистам в продаже/покупке транспорта и поиске запчастей для него.

Минимальные требования, необходимые для выполнения программы

- стандартная клавиатура;
- манипулятор «мышь»;
- наличие подключения к Интернету;
- наличие web-браузера: *Internet Explorer, Mozilla Firefox, Google Chrome, Opera* или *Safari*;
- ОЗУ 128 Мб;
- процессор: *Intel Pentium 4* и более поздней версии;
- свободное место на диске: 100 Мб.

Характеристики программы

Данный программный продукт имеет следующие характеристики:

1) Языки программирования:

- *PHP* (версия 5.3.x);
- *JavaScript*:
 - библиотека *jQuery*;
 - технология *AJAX*;

- *CSS*.

2) Система администрирования:

- *phpMyAdmin* (версия 3.5.x).

3) СУБД:

- *MySQL* (версия 5.x).

4) Веб-сервер:

- *Apache* (версия 2.2.22).

Операционная система может быть *Windows* или *Linux*. Версии могут быть не самые современные.

В качестве клиентского средства используется браузер:

- *Mozilla Firefox, Chrome, Iceweasel, Konqueror* в среде операционной системы *Linux*;

- *Chrome, Opera, Mozilla Firefox, Internet Explorer, Safari* в среде операционной системы *Windows*.

Программный продукт «Разработка информационного портала автовладельца» имеет следующие количественные характеристики:

- 1) объем исходных текстов – 9400 строк;
- 2) число таблиц базы данных – 32;
- 3) число файлов проекта – около 90.

Обращение к программе

Для запуска программы на локальном компьютере необходимо установить веб-сервер *WAMP (Windows Apache MySQL PHP)*, который можно скачать на сайте <http://www.denwer.ru> [10].

После инсталляции сервер *WAMP* установится в каталог *C:/WebServers/home/localhost/www/*.

Управлять работой web-сервера можно с помощью ярлыков в меню «Пуск» либо на рабочем столе.

После этого следует обратиться к поисковой строке браузера и набрать в ней адрес: *http://localhost/*(название каталога, в который были помещены файлы проекта).

Входные и выходные данные

- входные данные программы хранятся в виде защищенных паролем таблиц базы данных;
- выходные данные создаются по мере надобности.

2.4.2 Структура каталогов программного продукта

Проект программного продукта должен быть установлен в каталогах веб-сервера *Apache*:

- *site* (*\WebServers\home\localhost\www*).

Каталог *site* содержит следующие подкаталоги, в которых хранятся исходные коды информационной системы, представленные в кодировке *UTF-8* и написанные на языках *PHP*, *JavaScript*, *HTML* и *CSS*:

- *admin* – содержит файлы, организующие работу администратора;
- *blocks* – содержит файлы, описывающие основные блоки портала;
- *CSS* – содержит файлы каскадных таблиц стилей;
- *images* – содержит изображения, добавленные пользователем;
- *img* – содержит системные изображения;
- *js* – содержит файлы вспомогательных скриптов;
- *media* – содержит дополнительный медиа файлы.

2.4.3 Структура базы данных

Определения справочных и основных таблиц БД находятся в файле *db.sql*.

Проект содержит 32 таблицы. Структура наиболее важных из них представлена в таблице 4.

Таблица 4 – Основные таблицы БД

users (пользователи)					
Название	Тип	UNIQUE	NOT NULL	Индекс	Описание
<i>id_user</i>	<i>int(11)</i>	+	+	<i>key</i>	<i>id</i> пользователя
<i>login</i>	<i>varchar(20)</i>	+	+	-	логин пользователя
<i>password</i>	<i>varchar(35)</i>	-	+	-	пароль пользователя
<i>email</i>	<i>varchar(50)</i>	-	-	-	электронный адрес пользователя
<i>reg_date</i>	<i>varchar(32)</i>	-	+	-	дата регистрации
<i>name_user</i>	<i>varchar(32)</i>	-	-	-	имя пользователя
<i>lastname</i>	<i>varchar(32)</i>	-	-	-	фамилия пользователя
<i>skype</i>	<i>varchar(25)</i>	-	-	-	логин <i>skype</i> пользователя
<i>phonenumner</i>	<i>varchar(20)</i>	-	-	-	номер
<i>male</i>	<i>varchar(10)</i>	-	-	-	пол пользователя
<i>gorod</i>	<i>varchar(48)</i>	-	-	-	город проживания пользователя
<i>birthday</i>	<i>varchar(11)</i>	-	-	-	дата рождения пользователя
<i>img_href</i>	<i>varchar(150)</i>	-	-	-	ссылка на изображения пользователя
tbl_auto (марка автомобиля)					
<i>id_auto</i>	<i>tinyint(3)</i>	+	+	<i>key</i>	<i>id</i> марки автомобиля
<i>name_auto</i>	<i>varchar(50)</i>	+	+	-	название марки автомобиля
<i>href_img</i>	<i>varchar(150)</i>	-	-	-	ссылка на изображение марки автомобиля
<i>producer</i>	<i>tinyint(1)</i>	-	+	-	производитель (0 – зарубежный, 1 – отечественный)
<i>href_auto</i>	<i>varchar(60)</i>	-	+	-	ссылка на автомобиль
tbl_auto_model (модель автомобиля)					
<i>id_auto_model</i>	<i>smallint(3)</i>	+	+	<i>key</i>	<i>id</i> модели автомобиля
<i>id_auto</i>	<i>tinyint(3)</i>	+	+	*	<i>id</i> марки автомобиля

Продолжение таблицы 4

<i>name_auto_model</i>	<i>varchar(55)</i>	-	+	-	название модели автомобиля
<i>href_model</i>	<i>varchar(50)</i>				ссылка на модель автомобиля
<i>generation</i>	<i>tinyint(2)</i>				количество поколений
<i>img_href_model</i>	<i>varchar(150)</i>				ссылка на изображение модели автомобиля
tbl_auto_description (описание марки автомобиля)					
<i>id_description</i>	<i>int(11)</i>	+	+	<i>key</i>	<i>id</i> описания марки автомобиля

<i>id_auto</i>	<i>int(3)</i>	+	+	*	<i>id</i> марки автомобиля
<i>fullname</i>	<i>varchar(150)</i>	-	-	-	полное название марки автомобиля
<i>date</i>	<i>varchar(100)</i>	-	-	-	дата создания марки
<i>oldname</i>	<i>varchar(150)</i>	-	-	-	старые названия марки
<i>ancestors</i>	<i>varchar(200)</i>	-	-	-	имена создателей марки
<i>location</i>	<i>varchar(90)</i>	-	-	-	местоположение
<i>parentcompany</i>	<i>varchar(150)</i>	-	-	-	родительская компания
<i>website</i>	<i>varchar(90)</i>	-	-	-	адрес веб-сайта марки
<i>reiting</i>	<i>double</i>	-	-	-	рейтинг марки

***tbl_model_legacy* (поколения моделей транспортного средства)**

<i>id_model_legacy</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> поколения модели
<i>id_auto_model</i>	<i>smallint(4)</i>	-	+	*	<i>id</i> модели автомобиля
<i>model_legacy_name</i>	<i>varchar(55)</i>	-	+	-	название поколения
<i>date production</i>	<i>varchar(15)</i>	-	-	-	дата производства

***tbl_model_legacy_type* (тип поколения моделей автомобиля)**

<i>id_legacy_type</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> тип поколение
<i>id_model_legacy</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> поколения модели
<i>model_legacy_type</i>	<i>varchar(55)</i>	-	+	-	название типа поколения
<i>href_img_legacy</i>	<i>varchar(150)</i>	-	-	-	ссылка на изображение

***tbl_model_legacy_mc* (характеристики модификаций модели автомобиля)**

<i>id_legacy_type_mc</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> модификации модели
<i>id_legacy_type</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> тип поколение
<i>name_modific</i>	<i>varchar(50)</i>	-	+	-	название модификации
<i>name_engine_type</i>	<i>varchar(10)</i>	-	+	-	тип топлива
<i>name_drive</i>	<i>varchar(8)</i>	-	+	-	тип привода
<i>name_acccleration</i>	<i>varchar(4)</i>	-	+	-	разгон до 100км/ч
<i>name_speed_max</i>	<i>varchar(10)</i>	-	-	-	максимальная скорость
<i>name_consumtion_mixed</i>	<i>varchar(10)</i>	-	-	-	расход топлива
<i>name_releaseperiod</i>	<i>varchar(20)</i>	-	-	-	года производства
<i>href_legacy_mc</i>	<i>varchar(100)</i>	-	-	-	ссылка на модификацию

Продолжение таблицы 4

<i>begin_year</i>	<i>year(4)</i>	-	+	-	начало производства
<i>end_year</i>	<i>year(4)</i>	-	+	-	конец производства

***tbl_model_suspension* (тип подвески модели автомобиля)**

<i>id_model_suspension</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> типа подвески
<i>id_legacy_type_mc</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> модификации модели
<i>front_suspension</i>	<i>varchar(50)</i>	-	-	-	задняя подвеска
<i>rear_suspension</i>	<i>varchar(50)</i>	-	-	-	передняя подвеска

***tbl_model_steering* (тип рулевого управления у модели автомобиля)**

<i>id_model_steering</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> типа рулевого управления
<i>id_legacy_type</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> модификации модели
<i>type_steering</i>	<i>varchar(50)</i>	-	-	-	тип рулевого управления
<i>tbl_model_transmission</i> (тип трансмиссии у модели автомобиля)					
<i>id_model_transmission</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> типа трансмиссии
<i>id_legacy_type_mc</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> модификации модели
<i>type_kpp</i>	<i>varchar(15)</i>	-	-	-	тип КПП
<i>number_gears</i>	<i>tinyint(1)</i>	-	+	-	количество передач
<i>drive</i>	<i>varchar(15)</i>	-	+	-	тип привода
<i>tbl_model_brake_system</i> (тип тормозной системы у модели автомобиля)					
<i>id_model_brake_system</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> типа тормозной системы
<i>id_legacy_type_mc</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> модификации модели
<i>front_brake</i>	<i>varchar(25)</i>	-	-	-	передние тормоза
<i>rear_brake</i>	<i>varchar(25)</i>	-	-	-	задние тормоза
<i>tbl_model_performance</i> (эксплуатационные показатели модели автомобиля)					
<i>id_model_performance</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> показателя
<i>id_legacy_type_mc</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> модификации модели
<i>max_speed</i>	<i>smallint(3)</i>	-	-	-	передние тормоза
<i>acceleration_time</i>	<i>decimal(2,1)</i>	-	-	-	задние тормоза
<i>fuel_city</i>	<i>decimal(2,1)</i>	-	-	-	расход в городе
<i>fuel_highway</i>	<i>decimal(2,1)</i>	-	-	-	расход на трассе
<i>fuel_mixed</i>	<i>decimal(2,1)</i>	-	-	-	расход смешанный
<i>weight</i>	<i>smallint(4)</i>	-	-	-	собственный вес
<i>max_weight</i>	<i>smallint(4)</i>	-	-	-	максимальный вес
<i>tank_capacity</i>	<i>tinyint(4)</i>	-	-	-	размер топливного бака
<i>tire_size</i>	<i>smallint(4)</i>	-	-	-	размер шин
<i>environmental_standart</i>	<i>varchar(25)</i>	-	-	-	экологический стандарт
<i>tbl_model_body</i> (тип кузова модели автомобиля)					
<i>id_body</i>	<i>mediumint(5)</i>	+	+	<i>key</i>	<i>id</i> типа кузова
<i>id_legacy_type_mc</i>	<i>mediumint(5)</i>	+	+	*	<i>id</i> модификации модели

Окончание таблицы 4

<i>type_body</i>	<i>varchar(15)</i>	-	-	-	тип кузова
<i>number_doors</i>	<i>tinyint(1)</i>	-	-	-	количество дверей
<i>seats</i>	<i>tinyint(2)</i>	-	-	-	количество мест
<i>length</i>	<i>smallint(4)</i>	-	-	-	длина
<i>width</i>	<i>smallint(4)</i>	-	-	-	ширина
<i>height</i>	<i>smallint(4)</i>	-	-	-	высота
<i>wheelbase</i>	<i>smallint(4)</i>	-	-	-	колесная база
<i>front_track</i>	<i>smallint(4)</i>	-	-	-	колея передняя

<i>rear_track</i>	<i>smallint</i> (4)	-	-	-	колея задняя
<i>clearance</i>	<i>smallint</i> (3)	-	-	-	клиренс
<i>cargo_volume</i>	<i>smallint</i> (4)	-	-	-	объем багажника
<i>tbl_model_engine</i> (тип двигателя модели автомобиля)					
<i>id_model_engine</i>	<i>mediumint</i> (5)	+	+	<i>key</i>	<i>id</i> типа двигателя
<i>id_legacy_type_mc</i>	<i>mediumint</i> (5)	+	+	*	<i>id</i> модификации модели
<i>type_engine</i>	<i>varchar</i> (15)	-	-	-	тип двигателя
<i>model_engine</i>	<i>varchar</i> (20)	-	-	-	модель двигателя
<i>volume_engine</i>	<i>smallint</i> (4)	-	-	-	объем двигателя
<i>power_engine</i>	<i>smallint</i> (3)	-	-	-	мощность двигателя
<i>revs_engine</i>	<i>smallint</i> (4)	-	-	-	обороты двигателя
<i>torgue</i>	<i>varchar</i> (20)	-	-	-	крутящий момент
<i>power_system</i>	<i>varchar</i> (40)	-	-	-	система питания
<i>gas_distribution</i>	<i>varchar</i> (20)	-	-	-	газораспределительный механизм
<i>position_cylinders</i>	<i>varchar</i> (20)	-	-	-	позиция цилиндров
<i>number_cylinders</i>	<i>tinyint</i> (2)	-	-	-	количество цилиндров
<i>diametr_cylinders</i>	<i>decimal</i> (3,1)	-	-	-	диаметр цилиндра
<i>stroke</i>	<i>decimal</i> (3,1)	-	-	-	ход поршня
<i>compression_ratio</i>	<i>decimal</i> (3,1)	-	-	-	степень сжатия
<i>number_valves</i>	<i>tinyint</i> (2)	-	-	-	количество клапанов
<i>fuel</i>	<i>varchar</i> (15)	-	-	-	топливо

На рисунке 8 представлено взаимодействие основных таблиц базы данных.

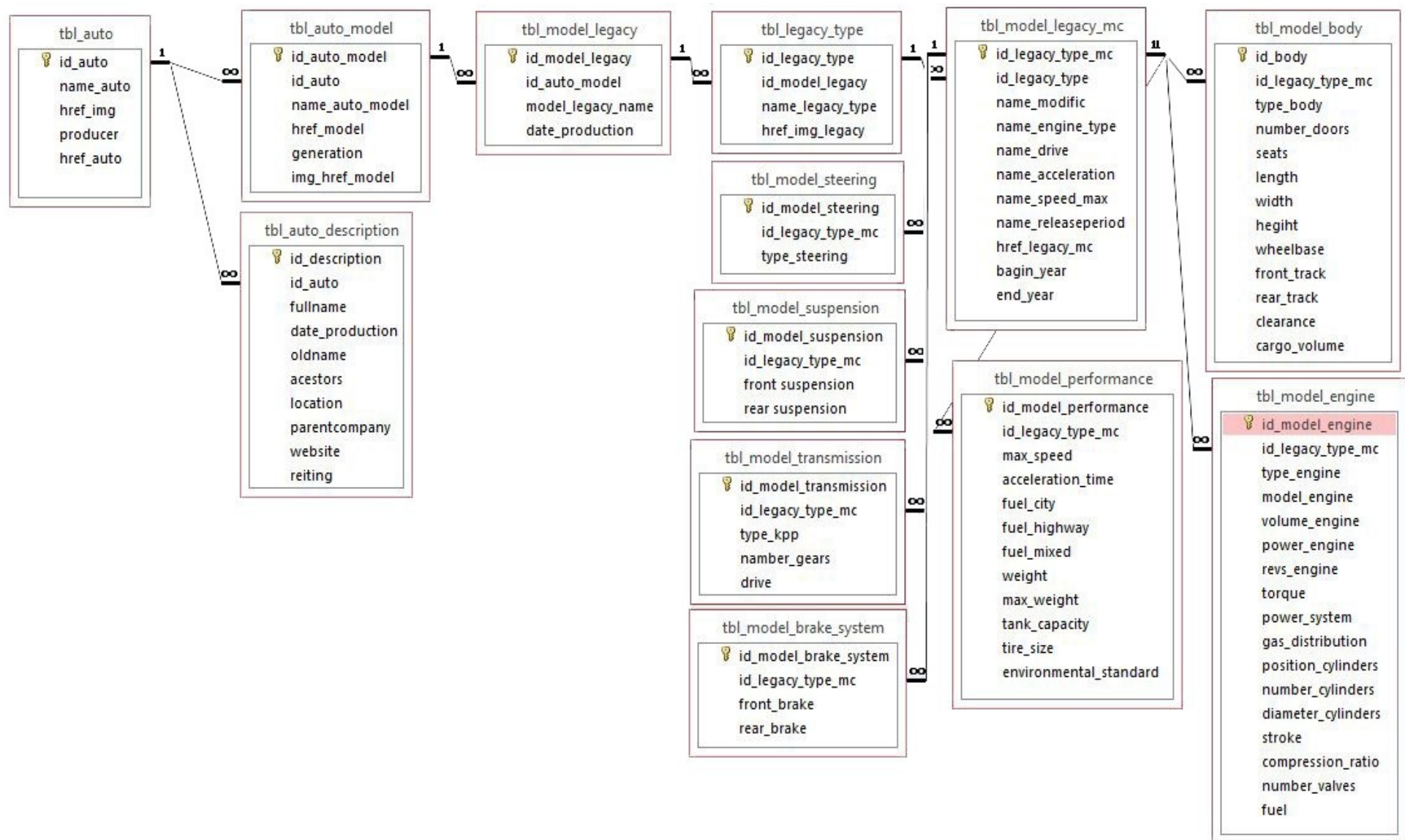


Рисунок 8 – Основные таблицы базы данных

2.4.4 Основные файлы проекта

Данный проект представлен рядом основных файлов:

- *about.php* – файл, позволяющий совершать общение между пользователем и администратором ресурса;
- *add_auto.php* – файл, отвечающий за внесение данных о продаваемом транспортном средстве в БД;
- *autocredit.php* – файл, позволяющий рассчитать кредитную ставку;
- *cat_marka.php* – файл, отображающий категории транспортных средств;
- *catalog_auto.php* – файл, отображающий категории марок транспортных средств;
- *catalog_auto_model.php* – файл, отображающий описание моделей транспортных средств;
- *catalog_model.php* – файл, отображающий категории моделей транспортных средств;
- *enter_login.php* – файл, отвечающий за вход пользователя в свой профиль на автопортале;
- *exit.php* – файл, отвечающий за выход пользователя из своего профиля на автопортале;
- *index.php* – файл, отображающий главную страницу веб-ресурса;
- *login.php* – файл, выполняющий вход пользователя в свой профиль;
- *registration.php* – файл, выполняющий регистрацию на портале;
- *send_message_admin.php* – файл, выполняющий отправку сообщения администратору ресурса;
- *update_user.php* – файл, выполняющий обновление информации в профиле пользователя;
- *update_user_password.php* – файл, выполняющий изменение пароля пользователя;
- *upload.php* – файл, выполняющий загрузку изображений на сервер;
- *user.php* – файл, отображающий описание профиля пользователя;

- *verification.php* – файл, выполняющий проверку правильности введенного пароля и логина для входа в профиль.

Также помимо данных файлов в проекте насчитывается ряд вспомогательных модулей.

2.5 Руководство пользователя

Данный портал имеет достаточно простой и удобный в использовании интуитивно понятный интерфейс.

В данном программном продукте существует только два вида пользователей: обычный человек и администратор, возможности и пользовательский интерфейс которых несколько отличаются.

Порядок запуска программы

Для запуска программы необходимо включить веб-браузер и ввести в нем адрес данного программного продукта: <http://www.autoselling.php>. В результате выполнения этих действий откроется главная страница сайта информационного портала (Рисунок 9).

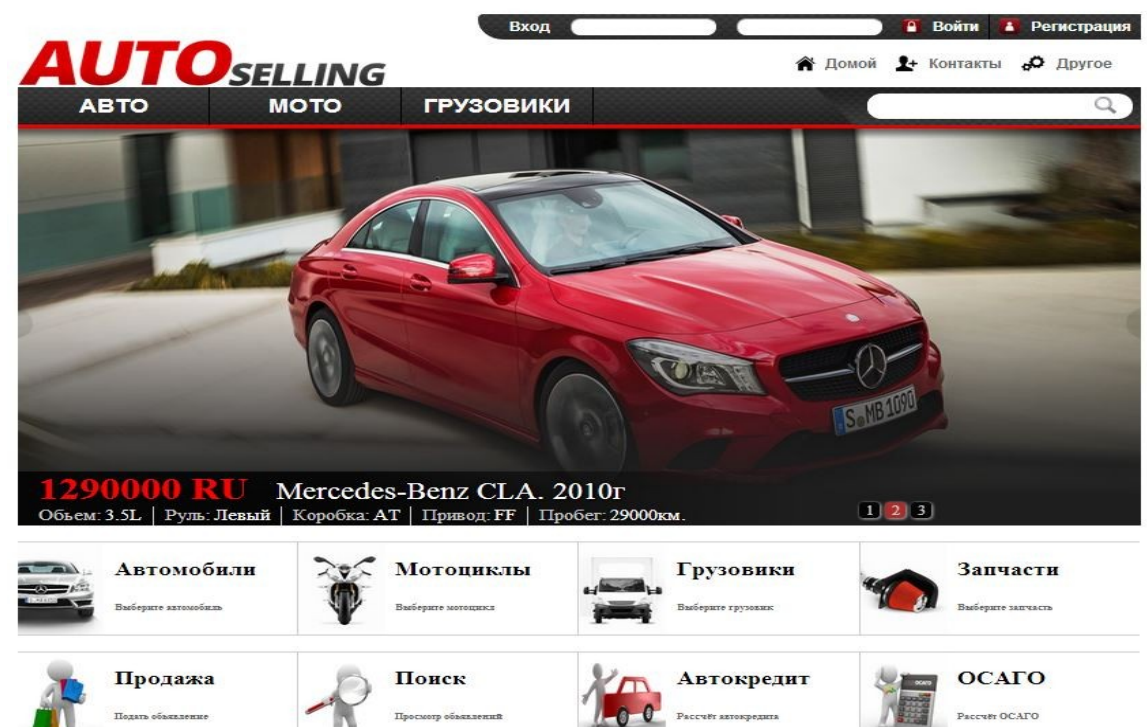


Рисунок 9 – Главная страница сайта

Регистрация на сайте

Каждый пользователь, желающий иметь свой личный кабинет, может создать его, выполнив регистрацию на сайте. Для осуществления регистрации на сайте пользователю необходимо нажать кнопку «Зарегистрироваться», после чего появится форма регистрации (Рисунок 10).

AUTO SELLING

Вход | Войти | Регистрация

Домой | Контакты | Другое

АВТО | МОТО | ГРУЗОВИКИ

РЕГИСТРАЦИЯ

РЕГИСТРАЦИЯ НОВОГО ПОЛЬЗОВАТЕЛЯ

Логин:
Введите логин

Пароль:
Введите пароль

Подтверждение пароля:
подтвердите пароль

E-mail:
example@gmail.com

Имя:
Введите имя

Фамилия:
Введите фамилию

Дата рождения:
день.месяц.год

Город:
Красноярск

Пол:
Мужской

Номер телефона:
777-777-7777

Skype:
Введите skype

Загрузить

SPECIAL

Lamborghini Gallardo

5.900.000 руб.

Категории

- Автомобили
- Мотоциклы
- Грузовики
- Запчасти

Информация

- Специальный предложения
- Лучшие продавцы
- Контакты
- Карта сайта

Мой профиль

- Мой профиль
- Избранное

facebook

© 2014 Powered by Shaldurov™

Рисунок 10 – Форма для регистрации пользователя на сайте

При своей загрузке форма регистрации не имеет никаких введенных данных, и кнопка «Регистрация» является неактивной как на рисунке 10. Таким образом, чтобы сделать кнопку активной, пользователю необходимо ввести всю требующуюся информацию в соответствующие поля. Система проверяет вводимые пользователем данные, выдавая сообщения в случае ошибки.

Существует четыре возможных ошибки пользователя при регистрации:

1) Ошибка логина. На логин существует единственное ограничение – он должен быть длиннее 3 символов.

2) Ошибка паролей. Эта ошибка возникает тогда, когда поля «Пароль» и «Подтверждение пароля» не совпадают. Пользователь должен быть внимательным и следить за тем, чтобы оба пароля были одинаковыми. Также длина пароля должна быть более 6 символов.

3) Ошибка ввода электронной почты.

4) Наличие незаполненных полей, обязательных для заполнения.

О каждой такой ошибке система информирует пользователя в виде сообщений, а также выделяет поле с неверным введенным значением красным цветом (Рисунок 11).

На рисунке 11 рассматривается ситуация, когда пользователь неправильно заполнил все поля. В таком случае он получает сообщение об ошибке.

Ошибка: Недопустимые символы в логине

РЕГИСТРАЦИЯ НОВОГО ПОЛЬЗОВАТЕЛЯ

Логин:*

Пароль:*

Подтверждение пароля:*

E-mail:*

Имя:

Фамилия:

Дата рождения:

Город

Пол

Номер телефона

Skype

Изображение:

Загрузить

Регистрация

Рисунок 11 – Ошибочное заполнение формы регистрации

В случае, если пользователь не допустил никаких ошибок при вводе информации, поля выделяются зеленым цветом, и кнопка «Регистрация» становится активной.

После регистрации необходимо авторизоваться на сайте.

Авторизация на сайте

Для того чтобы зайти в личный кабинет необходимо прежде всего зарегистрироваться на сайте. Только после этого возможно выполнение авторизации.

Чтобы авторизоваться необходимо нажать кнопку «Войти». После чего появится форма авторизации (Рисунок 12). В этой форме необходимо ввести требующиеся данные, а именно «Логин» и «Пароль».

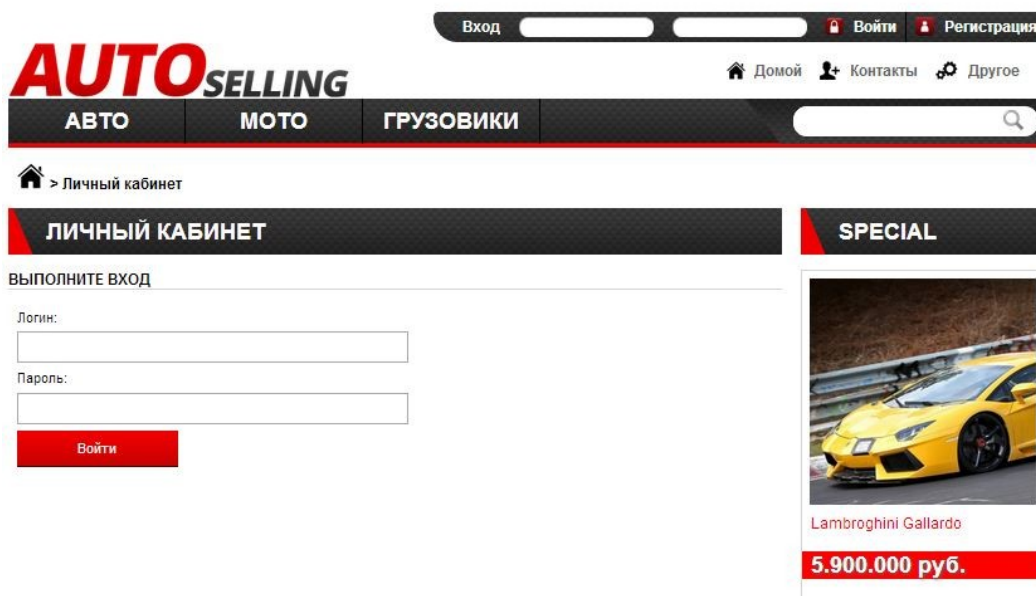


Рисунок 12 – Форма входа в личный кабинет

Если все введенные данные верны, то появится личный кабинет пользователя (Рисунок 13). В случае ошибки в отдельном окне появится сообщение «Неверный логин или пароль!».

В личном кабинете пользователь может просмотреть личные данные, введенные при регистрации. Также пользователь может изменить свои личные данные или пароль нажатием на соответствующие кнопки.

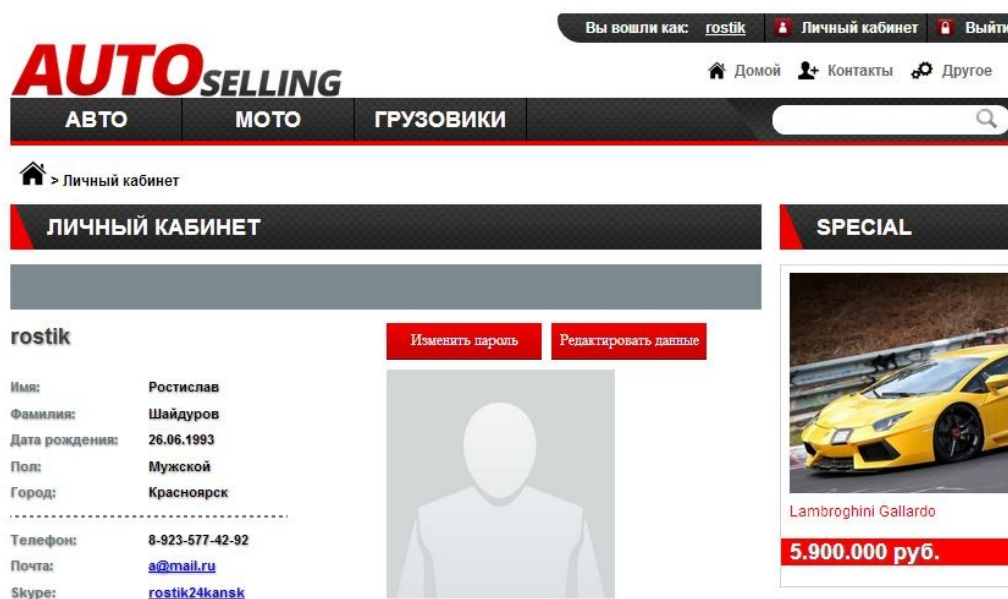


Рисунок 13 – Личный кабинет пользователя

2.6 Тестирование

Тестирование – это процесс выполнения прикладных программ с целью поиска ошибок. Как и при проектировании базы данных, пользователи новой системы должны быть вовлечены и в процесс ее тестирования. В идеале тестирование системы должно проводиться на отдельном комплекте оборудования. При использовании реальных данных важно предварительно создать их резервные копии на случай их повреждения в результате ошибок. По завершении тестирования процесс создания прикладной системы считается законченным, и она может быть передана пользователям в промышленную эксплуатацию.

Методы тестирования программного продукта, используемые в работе при отладке приложения:

- метод нисходящего/восходящего тестирования. Нисходящее тестирование начинается на уровне подсистем с модулями, которые представлены заглушками, т.е. простыми компонентами, имеющими такой же интерфейс, как и модуль, но без функционального кода. В конечном итоге все программные компоненты заменяются фактическим кодом и снова тестируются. Преимуществом такого подхода является то, что ошибки проектирования могут быть обнаружены еще на ранней стадии тестирования. Кроме того, уже на ранней стадии создания можно получить работающую систему, хотя и с ограниченной функциональностью. Недостатком этой стратегии тестирования является необходимость создания многочисленных заглушек модулей для моделирования низкоуровневых компонентов системы, а также затруднена проверка выходных данных (если не организовать их выдачу искусственным путем). Восходящее тестирование выполняется в противоположном направлении;
- метод тестирования приложения другим пользователем. Этот метод тестирования – один из самых простых. Исполняемый файл был передан человеку, не имеющему отношения к нашей работе, с целью выявления возможных ошибок;

- Методы черного и белого ящика для тестирования приложения. Как известно, тестирование продукта методом черного ящика производится путем внесения в поля данных непредусмотренного типа. В данном программном продукте для защиты от случайного ввода используются как маски ввода, так и анализ введенного значения.

Вторая стратегия – это тестирование программы методом белого ящика, т.е. тестирования, управляемого логикой программы. Эта стратегия позволяет исследовать внутреннюю структуру программы. В этом случае тестовые данные получают путем анализа логики программы. Тестирование проходит не весь программный продукт в целом, а какая-либо функция или действие, т.е. на вход в заранее выбранную функцию подаются различные данные в виде случайно выбранных символов. И наблюдается, как на это реагирует функция.

Тестирование данного программного продукта происходило по методике черного ящика. В результате чего были подготовлены тестовые наборы данных для проверки модуля регистрации пользователя на сайте. Процедура ручного тестирования представлена ниже.

1) Ввод логина длиной меньше четырех символов.

Ожидаемые результаты: система должна выдать ошибку.

Фактические результаты: система выдает ошибку.

2) Ввод пароля длиной меньше шести символов, которые не будут совпадать с дублирующим паролем.

Ожидаемые результаты: система выдаст ошибку, которая будет содержать информацию о том, что пароль слишком короткий и не совпадает с дублирующим паролем.

Фактические результаты: система выдает ошибку, которая содержит информацию о том, что пароль слишком короткий и не совпадает с дублирующим паролем.

3) Оставить поле пароль и логин пустыми.

Ожидаемые результаты: система выдаст сообщение о том, что поле логин и пароль являются пустыми.

Фактические результаты: система выдает сообщение о том, что поле логин и пароль являются пустыми.

4) Ввод в поле *e-mail* адрес электронной почты. Например, «1234567».

Ожидаемые результаты: система выдаст ошибку о том, что адрес электронной почты имеет неверный формат.

Фактические результаты: система выдает ошибку о том, что адрес электронной почты имеет неверный формат.

Таким образом, ожидаемые и фактические результаты поведения системы регистрации пользователя совпали.

Выводы по главе

В данной главе рассмотрено проектирование рабочего информационного портала со всеми его основными функциями.

Программные средства (СУБД *MySQL*, язык *PHP*), выбранные для реализации информационного портала, проявили себя с хорошей стороны, позволив добиться достаточного быстродействия и надежности программного продукта.

Разделение функций пользователя и администратора, а также создание отдельного интерфейса для каждого из них позволило повысить удобство управления порталом и надежность его функционирования.

Также по результатам тестирования никаких ошибок выявлено не было. Соответственно, можно сказать, что в итоге получился программный продукт, готовый к использованию.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута основная цель проекта, а именно: было улучшено качество удовлетворения информационных потребностей пользователя за счет разработки информационного портала для автолюбителей. Данный портал позволяет сократить время поиска требующейся пользователю информации и является более простым и удобным в использовании.

Были изучены основы данной предметной области, а именно: принципы построения информационных систем и проблемы, возникновение которых возможно в процессе их разработки.

В результате рассмотрения существующих автомобильных сайтов было установлено, что далеко не все из них обладают достаточно большой структурированной информационной системой для удовлетворения потребностей пользователя в интересующей его информации. К тому же возможны случаи, когда ресурс, обладая большой информационной системой, не всегда выдает уместную и актуальную информацию, вследствие чего автолюбитель тратит большое количество времени на ее поиск. Это может быть обусловлено плохой структурированностью системы, вследствие чего она перестает справляться со своей главной задачей, а именно с хранением информации в удобном виде для конечного пользователя.

По результатам изучения всевозможных программных средств в качестве основного языка программирования был выбран *PHP* за счет своей универсальности, удобства и простоты использования. А в качестве СУБД была выбрана *MySQL*, так как она достаточно проста в использовании, имеет скромные требования в отношении дискового пространства и памяти и удовлетворяет следующим требованиям: физическим параметрам, доступности, обработки транзакций, утилитам и средствам разработки.

В ходе разработки архитектуры программного продукта была определена основная схема функционирования информационной системы, а также ее основные модули и их взаимосвязи.

В результате проектирования БД, был выделен ряд основных сущностей информационной системы с их взаимосвязями, после чего была составлена концептуальная схема.

Все выделенные сущности и модули информационной системы были реализованы на практике.

Данный программный продукт был протестирован по методике черного ящика на уровне пользователя. По результатам тестирования никаких отклонений обнаружено не было. Все фактические результаты совпали с ожидаемыми.

У полученного программного продукта есть дальнейшие пути развития. Например, возможно запустить функционирование информационного портала во Всемирной сети. Также возможна дальнейшая интеллектуализация процесса добавления новой информации в БД и реализация механизма принятия решений для пользователя, который помогал бы покупателю машины выбирать транспортное средство в зависимости от значимости для него тех или иных критериев.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дронов, В. А. *PHP, MySQL и Dreamweaver MX 2004*. Разработка интерактивных Web-сайтов [Текст] / В. А. Дронов. – М. : БХВ-Петербург, 2005. – 448 с.
2. Зольников, Д. С. *PHP5* [Текст] : учебный курс / Д. С. Зольников – М.: НТ Пресс, 2007. – 256 с.
3. Колисниченко, Д. Н. Самоучитель *PHP 5* [Текст] / Д. Н. Колисниченко – М.: Наука и Техника, 2007. – 640 с.
4. Колисниченко, Д. Н. *PHP 5 на примерах* [Текст] / Д. Н. Колисниченко, И. В. Симдянов, С. В. Голышев. – М. : БХВ-Петербург, 2005. – 576 с.
5. Леонтьев, Б. В. *Web-Дизайн: Тонкости, хитрости и секреты* [Текст] / Б. В. Леонтьев. – М. : Майор, 2010. – 170 с.
6. Мазуркевич, А. С. *PHP: настольная книга программиста* [Текст] : учебный курс / А. С. Мазуркевич, Д. А. Еловой. – М. : Новое знание, 2003. – 480 с.
7. Шлосснейгл, Д. Профессиональное программирование на *PHP* [Текст] : пер. с англ. / Д. Шлосснейгл. – М. : Вильямс, 2006. – 624с.
8. Курсы по созданию сайта для бизнеса [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://ruseller.com/>. – Загл. с экрана.
9. Учебник *PHP* – Оглавление – *HTML.net* [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://ru.html.net/>. – Загл. с экрана.
10. *Denwer* [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.denwer.ru/>. – Загл. с экрана.
11. *jQuery – Javascript* [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.javascript.ru/jquery/>. – Загл. с экрана.
12. *PHP для начинающих* [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.cyberforum.ru/php-beginners/>. – Загл. с экрана.
13. *PHP и базы данных* [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.cyberforum.ru/php-database/>. – Загл. с экрана.

14. *PHP: Простой учебник – Manual* бизнеса [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://php.net/manual/ru/tutorial.php/>. – Загл. с экрана.
15. *PHP, MySQL и другие веб-технологии* [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.php.su/>. – Загл. с экрана.