Образовательное учреждение высшего образования

«Южно-Уральский институт управления и экономики»

Разработка интернет-магазина для компании ООО «Кузовчел»

НАУЧНАЯ РАБОТА

Руководитель работы

Т.Н. Лебедева

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Автор работы

В.С. Финогенов

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Челябинск

2017

АННОТАЦИЯ

Финогенов Вячеслав Сергеевич

«Разработка интернет-магазина для ООО «Кузовчел».

Научная работа на тему: «Разработка интернет-магазина для общества с ограниченной ответственностью «Кузовчел» (ООО «Кузовчел»).

В работе раскрывается актуальность темы, сформулированы цели и задачи исследования.

Научная работа состоит из введения, трех глав и заключения.

Во введении определены актуальность работы, объект, предмет, цели и задачи, практическая значимость работы.

В первой главе представлен анализ предметной области, требования к разрабатываемой информационной системе, инструменты для решения поставленной задачи.

Во второй главе описана деятельность ООО «Кузовчел»: представлены организационная структура и бизнес-процессы компании, рассмотрены способы и средства реализации информационных процессов на предприятии, а также сформулировано техническое задание на разработку сайта.

Третья глава содержит общее описание разработанной системы, оценку качества разрабатываемой системы в соответствии с ГОСТ, особенности внедрения, расчёт экономической эффективности проекта на разработку информационного сайта.

В заключении представлены общие выводы по работе.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc486416300)

[ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc486416301)

[1.1 Понятия и функции интернет-магазина 7](#_Toc486416302)

[1.2 Формирование требований к проекту на разработку web-приложения 10](#_Toc486416303)

[1.2.1 Цели проекта 10](#_Toc486416304)

[1.2.2 Результаты проекта 10](#_Toc486416305)

[1.2.3 Допущения и ограничения проекта 11](#_Toc486416306)

[1.2.4 Ключевые участники и заинтересованные стороны проекта 12](#_Toc486416307)

[1.3 Анализ способ разработки сайтов и обзор систем управления контентом сайта 12](#_Toc486416308)

[1.4 Выбор средств проектирования и средств создания web-приложения 17](#_Toc486416309)

[1.4.1 Microsoft Visio 17](#_Toc486416310)

[1.4.2 PHP 18](#_Toc486416311)

[1.4.3 MySQL 19](#_Toc486416312)

[1.4.4 PhpMyAdmin 19](#_Toc486416313)

[1.4.5 JavaScript 20](#_Toc486416314)

[1.4.6 AJAX 21](#_Toc486416315)

[1.4.7 Notepad++ 21](#_Toc486416316)

[ВЫВОД ПО ГЛАВЕ 1 22](#_Toc486416317)

[ГЛАВА 2 ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К WEB-ПРИЛОЖЕНИЮ 23](#_Toc486416318)

[2.1 Описание деятельности компании ООО «Кузовчел» 23](#_Toc486416319)

[2.2 Описание информационной системы компании ООО «Кузовчел» 23](#_Toc486416320)

[2.3 Описание модели деятельности по организации 24](#_Toc486416321)

[2.4 Формирование технического задания 27](#_Toc486416322)

[ВЫВОД ПО ГЛАВЕ 2 32](#_Toc486416323)

[ГЛАВА 3 РАЗРАБОТКА И ВНЕДРЕНИЕ WEB-ПРИЛОЖЕНИЯ 33](#_Toc486416324)

[3.1 Разработка Интернет-магазина 33](#_Toc486416325)

[3.2 Оценка качества информационной системы 48](#_Toc486416326)

[3.3 Внедрение информационной системы 49](#_Toc486416327)

[3.4 Оценка экономических затрат на проект 55](#_Toc486416328)

[ВЫВОД ПО ГЛАВЕ 3 71](#_Toc486416329)

[ЗАКЛЮЧЕНИЕ 72](#_Toc486416330)

[СПИСОК ЛИТЕРАТУРЫ 73](#_Toc486416331)

# ВВЕДЕНИЕ

Интернет-технологии открыли совершенно новые возможности для предприятий торговли, которые, благодаря своим неоспоримым преимуществам, настолько прочно вошли в практику, что в течение десятилетия стали не только привычным, но и необходимым явлением. Интернет сегодня выступает универсальной деловой средой, обеспечивающей эффективную коммуникацию компаний и потребителей. В той или иной степени, методы электронного бизнеса используют практически все предприятия сферы торговли, при этом отмечается непрерывное совершенствование схем коммуникации.

Розничная торговля представляет собой наиболее благоприятную почву для внедрения инноваций, прежде всего, в сфере системы отношений с клиентами, от которых в наибольшей степени зависит эффективность и успешность работы предприятия. Высокая конкуренция в данной сфере определяет необходимость поиска эффективных сочетаний различных факторов, создания и внедрения новых эффективных инструментов.

Применение информационных технологий, выступающих сегодня основными направлениями инноваций в сфере торговли, обеспечивает повышение эффективности управления предприятием и способствует увеличению ключевого показателя – оборачиваемости товара, чем повышают его конкурентоспособность на рынке.

Актуальность разработки заключается в возможности применения разработанного интернет-магазина в практической деятельности предприятия. Внедрение интернет-магазина позволит расширить рынки сбыта продукции компании, увеличить клиентскую базу, наладить коммуникацию с потенциальными клиентами компании.

Цель научной работы: разработка интернет-магазина для компании ООО «Кузовчел».

Поставленная цель, объект и предмет исследования определяют следующие задачи исследования:

* проанализировать специфику использования web-технологий в бизнесе;
* выделить основные особенности интернет-магазина, рассмотреть его структуру;
* изучить основные способы разработки сайтов, средства проектирования и создания интернет-магазина;
* спроектировать, разработать и внедрить интернет-магазин;
* произвести оценку качества интернет-магазина и оценку экономических затрат на проект.

# ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Понятия и функции интернет-магазина

Интернет-магазин – это реализованное в сети Интернет представительство путем создания Web-сервера для продажи товаров и услуг другим пользователям сети Интернет. Электронный магазин называют также Интернет-магазином. К нему полностью подходит определение виртуального предприятия. Иначе говоря, Электронный магазин — это сообщество территориально разобщенных сотрудников магазина (продавцов, кассиров) и покупателей, которые могут общаться и обмениваться информацией через электронные средства связи при полном (или минимальном) отсутствии личного прямого контакта.

Электронная торговля в Интернет- магазине основывается на той же структуре, что и традиционная торговля. Сравнительная характеристика традиционной и электронной торговли приведена в таблице 1.

Таблица 1 – Сравнительная характеристика традиционной и электронной работы

|  |  |  |
| --- | --- | --- |
|  | Традиционный магазин | Интернет-магазин |
| Тип торговой площади | Торговый зал | Виртуальный магазин |
| Способ навигации в магазине | Ходьба покупателя по торговому залу и осмотр товаров на полках магазинов | Просмотр покупателем страниц сервера |
| Предоставление консультационных услуг для покупателя | Личный контакт покупателя с продавцом (консультация) | Консультация у продавца (при необходимости) по компьютерной сети или по телефону |
| Выбор покупателем товара | Выбор покупателем товара | Выбор покупателем товара |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Способ заказа товара | Заказ товара напрямую у продавца | Заказ товара через сервер |
| Отчёт о  выполненном заказе | Выписка продавцом и вручение покупателю счета на оплату заказанного товара | Пересылка продавцом по компьютерным сетям покупателю счета на оплату заказанного товара |
| Способ оплаты заказа | Оплата покупателем счета на товар в кассе магазина наличными деньгами или банковской картой | Оплата покупателем счета по какой-нибудь системе электронных платежей (банковская карта, электронный чек, цифровые деньги, электронные деньги) |

Перевод традиционной торговли в сеть Интернет делает ее более гибкой, так как электронная торговля, оперируя цифровой информацией в компьютерных сетях, облегчает сотрудничество людей.

С технической точки зрения интернет-магазин – это совокупность интернет-витрины и торговой системы – фронт-офиса и бэк-офиса. Интернет-витрина предоставляет интерфейс к базе данных продаваемых товаров (в виде каталога, прайс-листа), работает с виртуальной торговой тележкой, оформляет заказы и регистрирует покупателя, предоставляет помощь покупателю в онлайновом режиме, передает информацию в торговую систему и обеспечивает безопасность личной информации покупателя. Далее торговая система осуществляет автоматическую обработку поступающих заказов – резервирует товар на складе, контролирует оплату и доставку товара.

Основные функции Интернет-магазина – это информационное обслуживание покупателя, обработка заказов, проведение платежей, а также сбор и анализ различной статистической информации.

В зависимости от функциональных возможностей, электронный магазин может быть отнесен к одной из трех категорий:

* интернет-витрина;
* торговый автомат;
* автоматический магазин.

В случае интернет-витрины электронного магазина представляет собой логическое расширение обычного web-сайта, когда на него просто выкладывается вся необходимая информация о товарах, которая по мере надобности обновляется. По существу, интернет-витрина реализует только рекламную функцию. Для осуществления покупки после посещения такого электронного магазина пользователь проходит обычный цикл: звонок или визит в компанию, оплата и доставка. Затраты на создание и администрирование интернет-витрины невелики. Электронный магазин такой категории может быть оправдан, прежде всего, при представлении продукции, изучение которой в торговом зале обычного магазина требует слишком много времени.

Торговый автомат, в отличие от интернет-витрины, осуществляет реальную торговлю. Такой электронный магазин может не только выполнять функции витрины, но и принимать заказы и передавать их менеджеру, то есть оформлять заказы и выписывать счета на оплату без присутствия покупателя. Однако его создание и администрирование требует навыков и определенной квалификации.

Автоматический магазин – эффективное и комплексное решение в торговом бизнесе. Он не только выписывает счета, но и отслеживает заказы, принимает электронные платежи и формирует заявки на доставку товаров покупателям. Электронные магазины этого типа довольно сложны, требуют немалых затрат на разработку, но вместе с тем, могут существенно увеличить объем продаж.

Существенно отличает друг от друга варианты построения электронных магазинов используемая модель бизнеса:

* онлайновый магазин (отсутствует традиционная торговая сеть);
* совмещение офлайнового бизнеса с онлайновым (когда электронный магазин создается на основе действующей реальной торговой структуры).

Несомненным преимуществом обладают магазины второго типа:

* электронный магазин пользуется преимуществами доставки с существующей сети розничных магазинов, он может предлагать вариант получения товара в выбранном магазине, в отличии от чисто онлайнового магазина, для него нет проблем при возврате товара;
* офлайновые покупатели могут предварительно ознакомиться с товарным ассортиментом и характеристиками на сайте, а затем прийти в ближайший реальный магазин.

## 1.2 Формирование требований к проекту на разработку web-приложения

### 1.2.1 Цели проекта

Цели создания интернет-магазина для компании ООО «Кузовчел» представлены в таблице 2.

Таблица 2 – Основные цели создания интернет-магазина для предприятия

|  |  |  |  |
| --- | --- | --- | --- |
| № | Цель | Показатель | Тренд показателя |
| 1 | Оптимизация процесса торговли продукцией и привлечение новых клиентов | Время, потраченное на поиск необходимого товара | Уменьшение |
| Количество сотрудников в отделе продаж | Уменьшение |
| Количество ошибок при составлении и оформлении заказа | Уменьшение |
| 2 | Расширение географии клиентов | Количество клиентов | Увеличение |

### 1.2.2 Результаты проекта

Результатом реализации проекта станет интернет-магазин, который будет удовлетворять информационные потребности различных категорий пользователей:

* информационные потребности пользователей:

1. просмотр каталога товаров;
2. просмотр раздела услуг, предоставляемых компанией;
3. заказ товара из каталога;

* информационные потребности администрации сайта:

1. авторизация в системе по логину и паролю (со своими правами доступа).
2. возможность запуска email-рассылки клиентам;
3. просмотр данных о клиентах магазина;
4. добавление, редактирование и обновление информации о товарах и услугах;
5. модерация отзывов и новостной ленты;
6. просмотр отчетов;
7. автоматическое формирование отчетов.

* внутрисистемные требования:

1. сохранения клиентской базы;

### 1.2.3 Допущения и ограничения проекта

* контент не должен противоречить закону о сертификации товаров и услуг;
* заказчик должен предоставить хостинг сайта с характеристиками:

1. операционная система Microsoft Windows или Unix (Linux, FreeBSD и пр.);
2. веб-сервер Apache 1.3.30 и выше или nginx;
3. PHP 5.3 и выше;
4. СУБД MySQL 4.1 и выше;

* специфические требования к защите информации. Защита информации должна обеспечиваться:
* функционалом хостинга;
* функционалом CMS:

1. от ннесанкционированного уничтожения или изменения данных на сайте;
2. от ннесанкционированного получения доступа к информации, представляющей коммерческую тайну или персональные данные;
3. от изменения злоумышленниками паролей администраторов, блокировки доступа в систему администрирования.

### 1.2.4 Ключевые участники и заинтересованные стороны проекта

Ключевыми участниками проекта являются заказчик системы, пользователи системы и разработчик.

Заказчик системы – директор компании ООО «Кузовчел».

Пользователи системы:

* клиенты и посетители сайта – люди, которые находятся в поиске необходимых им автозапчастей, а также услуг, предоставляемых компанией;
* модераторы сайта – лица, ответственные за модерирование контента

Разработчик системы, который будет выполнять все этапы разработки системы: анализ требований, проектирование системы, дизайн и верстку сайта, разработку и тестирование кода, обучение сотрудников института.

## 1.3 Анализ способ разработки сайтов и обзор систем управления контентом сайта

В наше время, время бурного роста интернет-технологий, существуют различные способы разработки сайтов. От мгновенных запусков в несколько часов, до продолжительных длящихся месяцами или даже годами. Рассмотрим наиболее популярные из них.

Самым простым способом, не требующим знаний языков программирования это использование визуальных редакторов, где путем преставления блоков заполняется вся структура и получается простейшие странички сайта. Так же есть огромное множество сервисов по созданию блогов, в которых сайт собирается как конструктор. Самыми распространёнными из них являются uCoz, Wix, uKit и т.д.

Если есть знания языка гипертекстовой разметки HTML, а также CSS (каскадные таблицы стилей), то можно сделать простой статический сайт. С помощью HTML строится разметка содержимого будущего сайта, а через CSS настраивается его внешний вид. Для создания более сложных динамических сайтов потребуется знания скриптового языка программирования PHP. Что бы сайт обладал большей функциональностью потребуются знания JavaScript и различных технологий, позволяющих сделать сайт более красивым, динамичным и быстродействующими. Такое создание с нуля займет уйму времени, как показывает практика, возникает множество ошибок и работы. Для устранения этих ошибок существуют такое программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта, называемое framework. Это каркас программной системы (или подсистемы), который может включать вспомогательные программы, библиотеки кода, язык сценариев и другое ПО, облегчающее разработку и объединение разных компонентов большого программного проекта. Обычно объединение происходит за счёт использования единого API. Framework определяется как множество конкретных и абстрактных классов, а также определений способов их взаимоотношения. Конкретные классы обычно реализуют взаимные отношения между классами. Абстрактные классы представляют собой точки расширения, в которых каркасы могут быть использованы или адаптированы.

Третьим более простым способом создания сайта является использование системы для обеспечения и организации совместного процесса создания, редактирования и управления содержимым страницы (CMS). В основном CMS имеют модульную архитектуру, и управление контентом происходит благодаря отдельным модулям. В большинстве CMS встроено множество готовых моделей, а также есть уже готовый шаблонный сайт, который можно начать редактировать под себя.

В данный момент существует большое множество CMS, как платных, так и бесплатных. CMS позволяют решать задачи различного уровня сложности в гораздо меньшие сроки, чем написание всего кода с нуля.

Нами был проведен анализ CMS. Были выбраны наиболее популярные из систем и произведен анализ по важных характеристикам для разрабатываемого web-приложения. Результат данного анализа представлен в таблице 3.

Таблица 3 – Сравнение характеристик различных CMS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Характеристика | WordPress | Joomla | Modx | 1С-Битрикс | [NetCat](http://www.ratingruneta.ru/netcat/) |
| Функционал | большой набор плагинов | огромное количество расширений, способных удовлетворить любые пользовательские потребности, но некоторые из них платные | средний набор плагинов | хороший выбор готовых модулей, а также сразу готовых решений с возможностью быстрой настройки (кроме редакции «Первого сайта») | небольшой набор стандартных модулей, модули партнеров |

Продолжение таблицы 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Особенности использования | будет работать хорошо, если правильно настроить систему.  Огромный набор функционала излишне загружает систему | имеет более богатую и гибкую структуру, чем WordPress | SEO ориентированность | повышенные требования к хостингу.  Система ориентирована на крупных заказчиков.  Возможность интеграции с большим числом сервисами | простой и понятный интерфейс для администраторов, отсутствие нагрузок на сервер.  Большие возможности для разработки. |
| Цена за минимальную редакцию | Open source | Open source | Open source | 4 900 руб. | 5 900 руб. |
| Количество Сайтов на 1 системе | Один | Один | Несколько | 1 шт, за каждый последующий необходимо доплачивать (1 470 руб) | Неограниченно |
| Поддержка мультимедийного контента | да | нет | да | да | да |
| Надежность | низкая | низкая | низкая | высокая | высокая |

Исходя из сравнительного анализа систем и личного опыта использования некоторых из них, для реализации проекта была выбрана система управления контентом сайта NetCat.

Компания NetCat (ООО «НетКэт») - одна из ведущих платных систем на российском рынке CMS. Первая версия CMS была выпущена в 1999 году. И на сегодняшний день предлагается 5 различных редакций, каждая из которых имеет свой набор готовых модулей и возможностей:

* Standart;
* Business;
* Corporate;
* E-commerce;
* Extra.

Система управления контентом сайтом NetCat написана с помощью языка программирования PHP.

NetCat система довольно функциональна, что позволяет без особого труда реализовать сложные задачи довольно быстро, а при необходимости с легкостью можно написать свои компоненты (это функционал сайта, который управляет содержательной частью страницы), которые внедряются автоматически и без проблем.

У NetCat очень понятный и удобный интерфейс, очень гибкая система управления данными нужных типов. Система имеет свои особенности из-за различия структуры модуля и доступа к данным. Управление структурой сайта позволяет создавать любой уровень вложенности разделов.

При помощи макетов дизайна можно реализовать любой дизайн.

Модуль «пользователи и права» дает достаточное управление пользователями системы, разграничением их прав доступа. Пользователей, которые имеют доступ к административной панели сайта, может быть неограниченное множество. При этом каждый из них может обладать различным уровнем прав, которые будут определять возможность их действий.

Для поставленной задачи NetCat дает возможность управлять настройками web-приложения пользователю с низким уровнем знаний в области ИТ, который просто умеет работать с компьютером на уровне офисных программ без знания необходимых языков программирования и языков разметки.

Для разработчика предоставляются широчайшие возможности по созданию и модификации компонентов и макетов стандартными средствами системы. Еще одна из возможностей позволяет в считанные секунды обратиться напрямую к базе данных MySQL на этапе отладки и быстро получить необходимые данные. Система также поддерживает интерфейс вызова задач по расписанию (cron). Для оптимизации работы проектов с большой посещаемостью во все редакции NetCat включен модуль «Кэширование», позволяющий уменьшить нагрузку на сервер.

Что не мало важно, система устанавливается практически на любой современный хостинг без установки дополнительного программного обеспечения.

Все версии NetCat проходят постоянный аудит у специалистов по безопасности веб-приложений, это помогает быть уверенным в безопасности от взломов и вирусов. В работе используются собственные и международные методики поиска уязвимостей.

Так же у NetCat есть бесплатная партнёрская лицензия их продукта (http://netcat.ru/products/charity), что позволяет использовать бесплатно систему NetCat для реализации данного проекта.

## 1.4 Выбор средств проектирования и средств создания web-приложения

Для проектирования системы была выбрана система бизнес-моделирования Microsoft Visio. В качестве CMS – система NetCat, которая работает на базе языков JavaScript, PHP и системы управления базами данных MySQL. Для разработки собственных компонентов использовались языки JavaScript, PHP, системы управления базами данных MySQL, технология AJAX.

### 1.4.1 Microsoft Visio

Microsoft Visio — программный продукт для бизнес-моделирования, позволяющая построить как комплексную иерархическую модель, так и описать ряд отдельных процессов. Для этого система предлагает наиболее популярные и удобные нотации моделирования: IDEF0, Процесс, Процедура, BPMN 2.0, EPC.

Основные решаемые задачи:

* формализация стратегии предприятия и контроль ее достижения;
* проектирование и оптимизация бизнес-процессов;
* проектирование организационной структуры и штатного расписания;
* формирование и распространение среди сотрудников регламентирующей документации;
* внедрение системы менеджмента качества в соответствии со стандартами ISO;
* формирование технических заданий и поддержка внедрения информационных систем.

Microsoft Visio обладает широкими функциональными возможностями для максимально легкого создания бизнес-архитектуры, начиная с постановки целей и проектирования бизнес-процессов, обеспечения сотрудников базой знаний и удобным доступом ко всей необходимой информации, и заканчивая контролем состояния компании, а также анализом отклонений и предложений сотрудников для совершенствования системы управления.

В нашем проекте система Microsoft Visio используется для анализа и проектирования бизнес-процессов, а также формирования на их основе технического задания к web-приложению.

### 1.4.2 PHP

Netcat – выбранная система управления контентом сайта, обоснование выбора представлено в пункте 1.3. В основе системы лежит язык программирования PHP, поэтому всю разработку необходимо вести на этом языке. На данный момент последней версией является PHP 7.0.7.

В области программирования для сети Интернет PHP – один из популярных скриптовых языков (наряду с JavaScript, Perl и языками, используемыми в ASP.NET) благодаря богатой функциональности, простоте, скорости выполнения, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

Благодаря встроенным средствам для разработки веб-приложений, язык имеет большую популярность. Основными из них являются:

* автоматическое извлечение POST и GET-параметров, и переменных окружения веб-сервера в предопределённые массивы;
* взаимодействие с большим количеством различных систем управления базами данных;
* автоматизированная отправка HTTP-заголовков;
* работа с HTTP-авторизацией;
* работа с cookies и сессиями;
* работа с локальными и удалёнными файлами, сокетами;
* обработка файлов, загружаемых на сервер;
* работа с XForms.

### 1.4.3 MySQL

NetCat использует реляционную систему управления базами данных MySQL, которая является открытым программным обеспечением.

MySQL отличатся хорошей скоростью работы, надежностью, гибкостью. Является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

MySQL система, основанная на привилегиях и паролях, за счет чего обеспечивается гибкость и безопасность, и с возможностью верификации с удаленного компьютера. Пароли защищены, т.к. они при передаче по сети при соединении с сервером шифруются.

### 1.4.4 PhpMyAdmin

Для удобства работы с базами данных MySQL через браузер будет использоваться phpMyAdmin.

PhpMyAdmin – веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования системы управления базами данных MySQL. Данное приложение позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять системой управления базами данных MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

На сегодняшний день phpMyAdmin широко применяется на практике. Последнее связано с тем, что разработчики интенсивно развивают свой продукт, учитывая все нововведения системы управления базами данных MySQL. Подавляющее большинство российских провайдеров используют это приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

Приложение распространяется под лицензией GNU General Public License и поэтому многие другие разработчики интегрируют его в свои разработки, например, XAMPP, Denwer, AppServ, Open Server.

### 1.4.5 JavaScript

JavaScript – прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

### 1.4.6 AJAX

AJAX – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

AJAX – не самостоятельная технология, а концепция использования нескольких смежных технологий. AJAX базируется на двух основных принципах:

* использование технологии динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью;
* использование динамического языка гипертекстовой разметки (DHTML) для динамического изменения содержания страницы.

Действия с интерфейсом преобразуются в операции с элементами называемыми Document Object Model (DOM), с помощью которых обрабатываются данные, доступные пользователю, в результате чего представление их изменяется. Производится обработка перемещений и щелчков мышью, а также нажатий клавиш.

В качестве формата передачи данных могут использоваться фрагменты простого текста, HTML-кода, JSON или XML.

### 1.4.7 Notepad++

Для написания кода использовалась программа Notepad++.

Notepad ++ - это бесплатный редактор исходного кода, который поддерживает несколько языков. Программа используется в операционной системы Windows.

На основе мощного компонента редактирования Scintilla, Notepad++ написан на C ++ и использует чистый Win32 API и STL, который обеспечивает более высокую скорость выполнения при минимальном размере программы. Базовая функциональность программы может быть расширена как за счёт плагинов, так и сторонних модулей, таких как компиляторы и препроцессоры.

# ВЫВОД ПО ГЛАВЕ 1

В первой главе был проведен анализ предметной области. На основе этого анализа были определены основные требования к разработке интернет-магазина для компании ООО «Кузовчел».

В главе представлен также представлен обзор инструментальных средств для разработки web-приложений. На основе личного опыта и обзора в качестве основного инструментального средства разработки системы была выбрана система NetCat.

Исходя из выбранной CMS были выбраны средства проектирования и разработки:

* языки разметки HTML и CSS;
* основной язык программирования PHP и как дополнительный JavaScript;
* система управления базами данных MySQL и приложение для управление СУБД PhpMyAdmin;
* технология Ajax;
* редактор кода Notpad++.

# ГЛАВА 2 ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К WEB-ПРИЛОЖЕНИЮ

### 2.1 Описание деятельности компании ООО «Кузовчел»

ООО «Кузовчел» была зарегистрирована в 2010 году. Компания занимается продажей автозапчастей, а также услуг по авторазбору.

Ниже представлена организационная структура компании (рисунок 4).

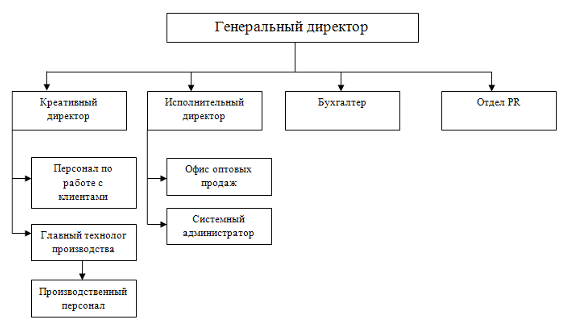


Рисунок 4 – Организационная структура компании ООО «Кузовчел»

### 2.2 Описание информационной системы компании ООО «Кузовчел»

В компании ООО «Кузовчел» установлено 9 автоматизированных рабочих мест (АРМ), принтеры, сканеры.

Все компьютеры компании соединены по локальной сети. Для установки локальной сети между компьютерами используются сетевые адаптеры, витая пара, маршрутизаторы с беспроводной точкой доступа, а также коммутаторы.

Тип сети: сеть с выделенным сервером. Сеть построена на основе топологии «звезда».

Распределение ресурсов: совместный доступ к данным, общие принтеры, другие совместно используемые периферийные устройства, организуется путем предоставления локальных ресурсов и периферийных устройств в общее пользование, хотя не исключено и использование сетевых устройств, главным образом принтеров.

На рисунке 5 представлен отдельный фрагмент схемы локальной вычислительной сети.

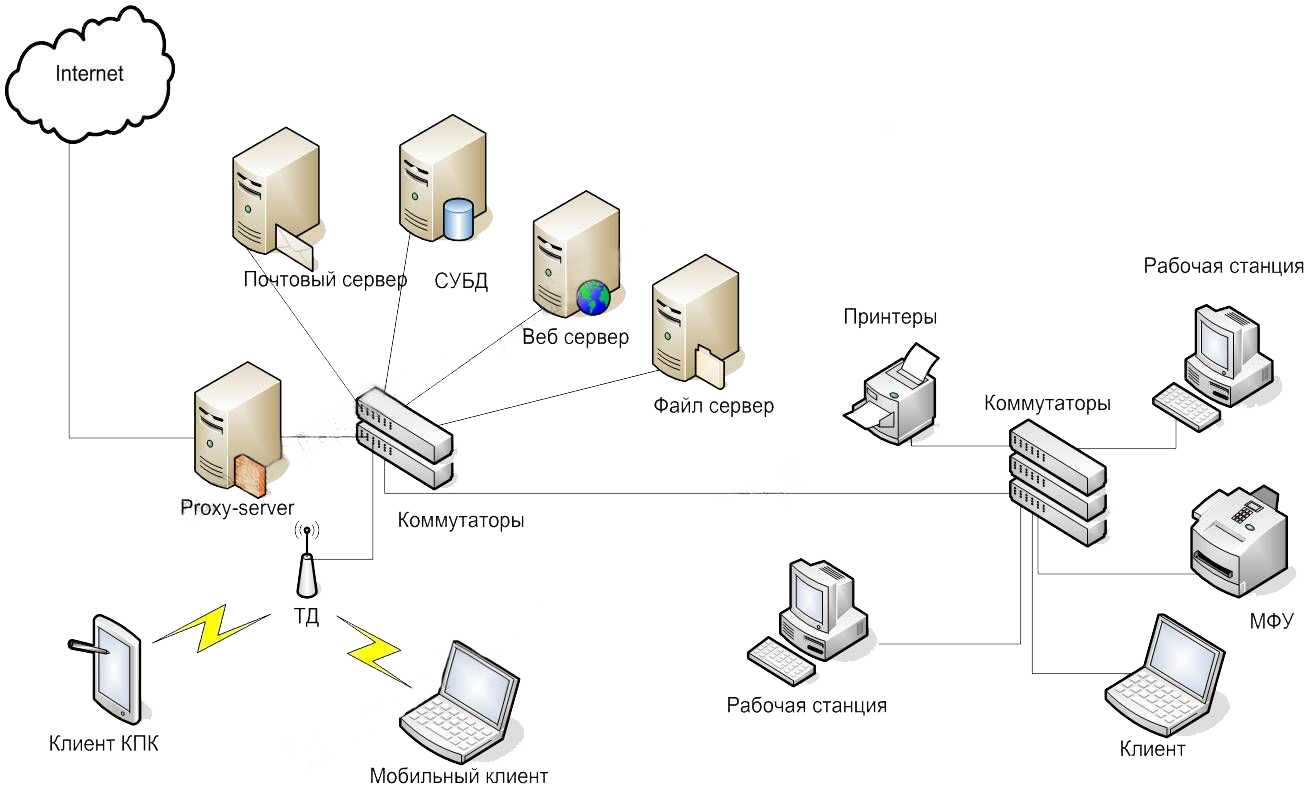


Рисунок 5 – Схема локальной вычислительной сети компании «Кузовчел»

### 2.3 Описание модели деятельности по организации

Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Вершина этой древовидной структуры, представляющая собой самое общее описание системы и ее взаимодействия с внешней средой, называется контекстной диаграммой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее до достижения нужного уровня подробности описания.

В IDEF0 различают пять типов стрелок:

* вход (Input) - материал или информация, которая используется или преобразовывается работой;
* управление (Control) - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления;
* выход (Output) - материал или информация, которая производится работой. Каждая работа должна иметь хотя бы одну стрелку выхода;
* механизм (Mechanism)- ресурсы, которые выполняют работу, например, персонал предприятия станки, механизмы и т.д.;
* вызов - специальная стрелка, указывающая на другую модель работы.

Уровень А0 самый важный в модели, это есть специальный вид (контекстной) диаграммы IDEF0, состоящей из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управления, и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.

На рисунке 6 представлена контекстная диаграмма процесса работы интернет-магазина компании в нотации IDEF0.

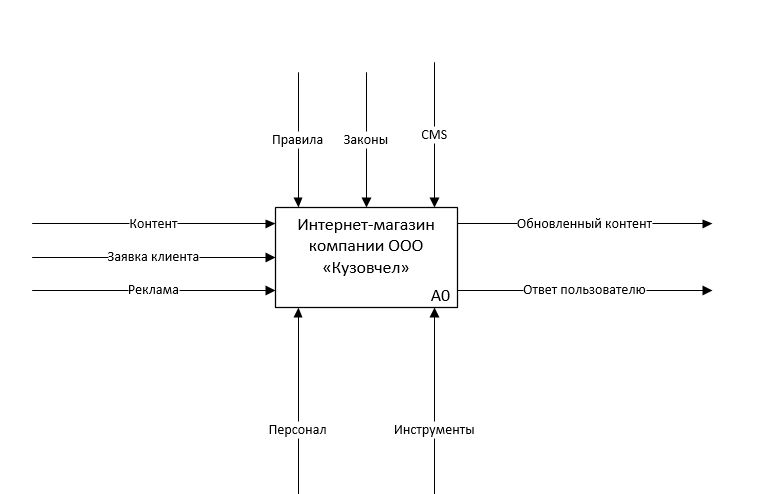


Рисунок 6 – Контекстная диаграмма бизнес-процесса IDEF0

После декомпозиции была получена следующая диаграмма, которая изображена в соответствии с рисунком 7.

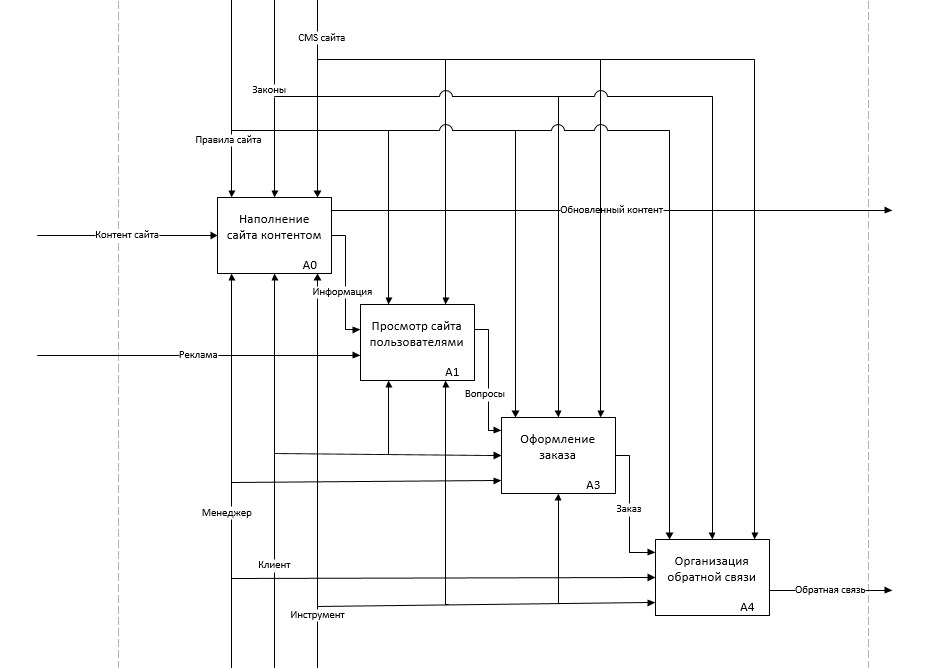


Рисунок 7 – Декомпозиция процесса

Технология описания бизнес-процесса делает все процессы прозрачными и понятными, позволяет анализировать и находить в них проблемы, приводящие к сбоям. Главное, что бизнес-процессы позволяют понимать взаимодействие что, кому и для чего они передают или принимают на каждом этапе. Как следствие, процессный подход значительно упрощает адаптацию.

### 2.4 Формирование технического задания

Полное наименование информационной системы – Интернет-магазин для компании.

Краткая характеристика области применения: сайт разрабатывается на CMS Netcat и предназначен для разработки Интернет-магазина компании.

Функциональное назначение: автоматизация процесса продажи товаров и услуг.

Эксплуатационное назначение: система предназначена для эксплуатации модерацией сайта и его клиентами.

Требования к информационной системе.

Автоматизированные процессы представлены в таблице 4

Таблица 4 – Автоматизируемые процессы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Действие | Исполнители | Используемые документы | | Модуль ИС |
| Входы | Выходы |
| 1 | Формирование и редактирование контента | Модератор | Статьи | - | Положение, задания, контакты |
| 2 | Формирование и редактирование каталога | Модератор | Текст, изображения. | Каталог товаров |  |
| 3 | Просмотр контента | Клиент | Анкета | Обратная связь |  |

Перечень формируемых отчетов.

Сайт должен обеспечить автоматическое формирование следующего отчета сообщение на почту модератору для обратной связи.

Требования к производительности.

Работа любого скрипта не должна превышать 60 секунд. При условии нагрузки на сервер не более 500.000 обращений к страницам сайта в сутки.

Требования к надежности.

Требования к надежности определяются возможностями системы прав доступа, заложенными разработчиком в системе управления сайтом NetCat, средствами администрирования используемого хостинга.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени на перезагрузку задействованных технических и программных средств при условии соблюдения условий эксплуатации самих технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановку программных средств.

Отказы веб-приложения возможны вследствие некорректных действий пользователя. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

Требования к численности и квалификации персонала.

Модератор должен обладать практическими навыками работы с графическим пользовательским интерфейсом и изучить руководство для пользователя.

Требования к составу и параметрам технических средств.

Требования к составу и параметрам технических средств определяются разработчиками CMS NetCat, приведенными в официальной документации к платформе и на сайте разработчика в Интернете http://netcat.ru/.

Требования к исходным кодам и языкам программирования

Требования к сайту определяются возможностями и рекомендациями, заложенными разработчиком системы управления контентом NetCat, приведенными в официальной документации к системе на сайте разработчика в Интернете http://netcat.ru/support/documentation/.

Требования к программным средствам, используемым веб-приложением.

Требования к программным средствам веб-приложения определяется возможностями, заявленными разработчиками «CMS NetCat», приведенными в официальной документации к платформе и на сайте разработчика в Интернете http://netcat.ru/developers/docs/install-and-settings/tech-requirements/:

* ооперационная система Microsoft Windows;
* PHP 5.3 и выше;
* СУБД MySQL 4.1 и выше.

Клиентская часть системы требует только установки веб-браузера на компьютере пользователя. Во всех современных операционных системах браузеры входят в стандартную комплектацию системы [3].

Требования к защите информации веб-приложения.

Требования к защите информации определяются возможностями системы прав доступа, заложенными разработчиком в системе управления сайтом NetCat, средствами администрирования используемого хостинга.

Стадии и этапы разработки.

Стадии и этапы разработки сайта приведены в таблице 5.

Таблица 5 – Стадии и этапы разработки сайта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сроки проекта | | | | |
| № п/п | Название этапа | Дата начала | Длительность | Дата окончания |
| 1 | Сбора данных | 30.01.2017 | 15 | 15.02.2017 |
| 2 | Освоение платформы | 16.02.2017 | 20 | 08.03.2017 |
| 3 | Разработка приложения | 09.03.2017 | 20 | 29.03.2017 |
| 4 | Продвижение | 30.03.2017 | 10 | 10.04.2017 |
| 5 | Завершение | 11.04.2017 | 5 | 16.05.2017 |

Ниже на рисунке 8 представлена диаграмма Ганта, которая позволяет боле конкретно понять распределение этапов и срока их выполнения.

Рисунок 8 – Диаграмма Ганта

# ВЫВОД ПО ГЛАВЕ 2

Компания ООО «Кузовчел» является современным предприятием, внедряющий передовые образовательные технологии с учетом актуальных требований общества. В соответствии с основными задачами своей деятельности, направленными на подготовку квалифицированных специалистов, а также на формирование у учащихся необходимых компетенций.

В рамках текущей главы была изучена деятельность организации, рассмотрена организационная структура и информационная система, основное используемое программное обеспечение.

В целях оптимизации торгового процесса и улучшения качества торгового сервиса компания успешно применяет возможности сетевого маркетинга. По этой причине появилась необходимость в разработке интернет-магазина для успешного выхода компании на рынок сетевой торговли.

Для формирования требований к системе была построена модель бизнес-процесса организации и проведения анкетирования пользователей, которая включает в себя следующие под процессы:

* заполнение контента;
* формирование корзины товаров;
* обратная связь.

На основе представленной модели сформировано техническое задание к Интернет-магазину для компании ООО «Кузовчел».

# ГЛАВА 3 РАЗРАБОТКА И ВНЕДРЕНИЕ WEB-ПРИЛОЖЕНИЯ

### 3.1 Разработка Интернет-магазина

3.1.1 Проектирование Интернет-магазина

Последовательность этапов от выявления потребностей до разработки тестовых сценариев, т.е. сверху вниз, является процессом проектирования системы. Движение по пирамиде от тестовых сценариев и удовлетворения потребностей пользователя, т.е. снизу-вверх, демонстрирует работу системы.

Рассмотрим процесс формирования сценариев использования с помощью диаграммы прецедентов языка UML.

В нотации языка UML данный вид диаграмм называется по-разному: диаграммы сценариев использования, вариантов использования или диаграммы прецедентов (английский вариант – Use Cases). Это диаграмма взаимодействия пользователя (актора, эктора, актёра, англ. actor) с системой для получения результата.

Здесь эктор – находящееся вне системы нечто (некто), взаимодействующее с системой. Это моет быть, как пользователь, так и другая программа. При взаимодействии пользователя с системой первый ожидает определенного, предсказуемого действия последней. В свою очередь прецедент (сценарий использования) выступает одним из способов определения реакции системы. С помощью прецедентов описывают поведение разрабатываемой программы, не определяя ее реализацию.

С помощью наглядного изображения диаграмм можно достичь взаимопонимания между заинтересованными лицами проекта: разработчиками, аналитиками, пользователями и др., – а также проверить архитектуру системы во время ее разработки.

Составление диаграммы прецедентов – это способ перейти от функциональных особенностей к конкретным сценариям. Каждый отдельный прецедент представляет функциональные требования в целом и отражает выполнение некоторого объема работ. Для конечного пользователя – эктора – прецедент реализует нечто ценное, например, формирует отчет, вычисляет, обрабатывает данные и проч. Прецеденты в свою очередь не должны быть слишком общими или чересчур специфичными.

Изображение системы на диаграмме осуществляется в виде прямоугольника. Прецедент – в виде эллипса, находящегося в границах системы. Для каждого прецедента должно быть определено уникальное имя – текстовая строка внутри эллипса. Эктор – на рисунке 9.

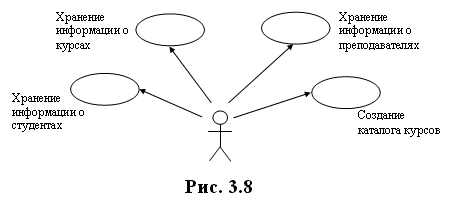


Рисунок 9 – Нотация эктора на диаграмме

Между экторами и прецедентами определяются отношения – ассоциации в виде соединяющих их линий.

Прецеденты делятся на два вида: высокого уровня и развернутые.

Прецеденты высокого уровня(high-level use case) – это очень краткие описания процессов, обычно состоящие из двух-трех предложений. Их используют начальном этапе при формулировании требований к системе. Это способ осознать степень сложности разрабатываемой системы, которые имеет слабое отношение к конкретному проектному решению.

Развернутые прецеденты(expanded use case) представляют собой более подробное описание, чем прецеденты первого вида. Они позволяют углубить понимание требований и процессов. Обычно оформляются в виде диалога между пользователем и программой.

При этом всегда важно помнить: любой прецедент описывает, что делает система, но не определяет, каким образом она это делает.

На рисунке 10 представлена UML диаграмма прецедентов, которая иллюстрирует структуру работы Интернет-магазина.

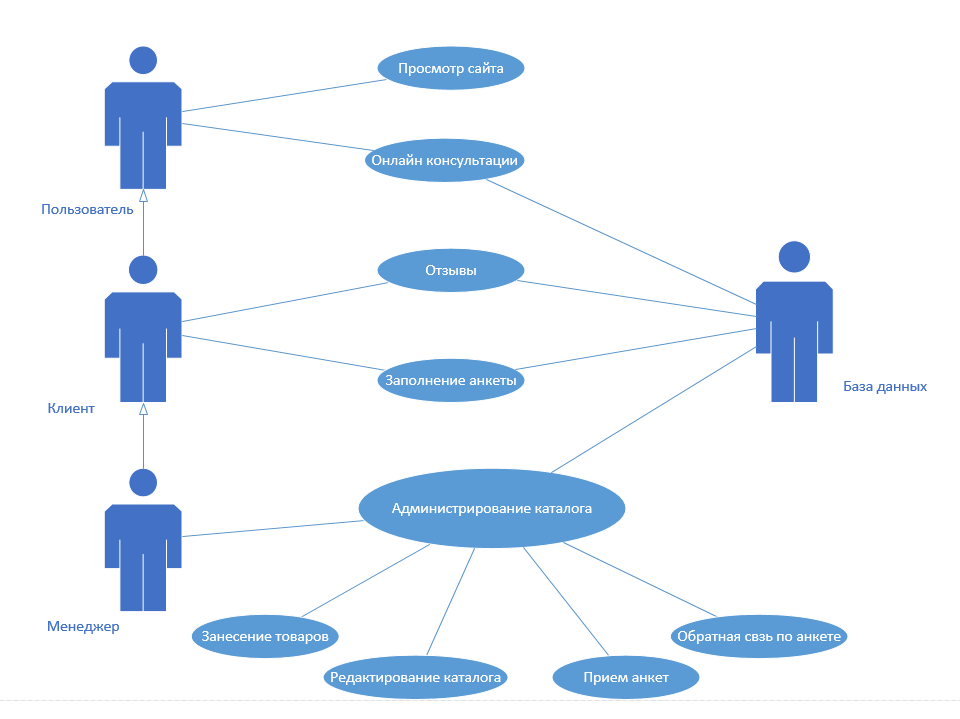


Рисунок 10 – UML диаграмма прецедентов.

Проектировать схему базы данных нет необходимости т.к. при установки системы управления сайтом NetCat база данных создается автоматически. Рассматривать структуру всей базы данных и описывать в отдельности каждую из таблиц нет смысла потому, что при разработке, а также при дальнейшей работе с сайтом используется всего несколько таблиц: Classificator, Subdivision, Sub\_Class и MessageXX (где XX индивидуальный номер).

Таблица Classificator

Свойства данных таблиц представлены в таблицах 1 – 6. Все эти таблицы имеют одинаковую структуру. Их назначение представлено ниже:

* поле – наименование таблицы;
* тип – тип данных, хранящихся в поле;
* ключ - признак ключевого поля (символ "\*" обозначает, что поле входит в состав первичного ключа);
* ai – признак авто заполнения поля (символ "\*" обозначает, что поле заполняется автоматически путем прибавления единицы к предыдущему полю);
* null - признак разрешение хранения значений типа NULL (символ "\*" обозначает, что в поле разрешено хранение значений типа NULL).

Subdivision - таблица разделов. В ней хранится структура разделов сайтов. Связанные таблицы: Catalogue, Template и классификаторы. Свойства указаны в таблице 5. Таблица Subdivision связана с таблицей Catalogue по типу связи «много:1», т.е. для одного сайта может быть определено несколько разделов.

Таблица 5 – Таблица разделов сайта

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поле | Тип | | Ключ | a\_i | Null | Описание |
| Subdivision\_ID | int(11) | | \* | \* |  | Уникальный номер страницы |
| Catalogue\_ID | int(11) | |  |  |  | Уникальный номер сайта |
| Parent\_Sub\_ID | int(11) | |  |  |  | Уникальный номер страницы родителя |
| Subdivision\_Name | varchar(255) | |  |  |  | Имя страницы |
| Template\_ID | int(11) | |  |  | \* | Уникальный номер макета дизайна |
| ExternalURL | varchar(255) | |  |  | \* | Внешняя ссылка |
| EnglishName | varchar(255) | |  |  |  | Псевдоним на английском языке |
| LastUpdated | | timestamp |  |  |  | Дата последнего изменения |

Продолжение таблицы 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Created | datetime |  |  |  | Дата создания |
| LastModified | datetime |  |  | \* | Дата последнего редактирования |
| LastModifiedType | int(11) |  |  |  | Тип последнего редактирования |
| Hidden\_URL | varchar(255) |  |  |  | Ссылка |
| Read\_Access\_ID | int(11) |  |  |  | Уникальный номер типа прав на запись |
| Write\_Access\_ID | int(11) |  |  |  | Уникальный номер типа прав на чтение |
| Priority | int(11) |  |  | \* | Приоритет |
| Checked | smallint(6) |  |  |  | Отображать / скрывать раздел |
| Edit\_Access\_ID | int(11) |  |  |  | Уникальный номер типа прав на редактирование |
| Checked\_Access\_ID | int(11) |  |  |  | Уникальный номер типа прав на просмотр |
| Delete\_Access\_ID | int(11) |  |  |  | Уникальный номер типа прав на удаление |
| Subscribe\_Access\_ID | int(11) |  |  |  | Уникальный номер типа прав на подписку |

Продолжение таблицы 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Moderation\_ID | int(11) |  |  |  | Уникальный номер модератора |
| Favorite | smallint(6) |  |  |  | Избранное |
| TemplateSettings | text |  |  | \* | Настройки макета дизайна |
| UseMultiSubClass | tinyint(1) |  |  |  | Использование более одного компонента на странице |
| UseEditDesignTemplate | tinyint(1) |  |  |  | Использование макета дизайна для редактирование контента |
| DisallowIndexing | int(11) |  |  |  | Запретить индексацию в поисковых системах |
| img | varchar(255) |  |  | \* | Изображение раздела |
| Pic | varchar(255) |  |  | \* | Изображение раздела когда активен |
| Description | text |  |  | \* | Блок Description |
| Keywords | text |  |  | \* | Блок Keywords |
| Title | varchar(255) |  |  | \* | Заголовок страницы |
| Language | varchar(255) |  |  | \* | Кодировка страницы |
| AlterTitle | char(255) |  |  | \* | Альтернативный заголовок страницы |
| DisplayType | enum |  |  |  | Тип отображения страницы |

Продолжение таблицы 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cache\_Access\_ID | int(11) |  |  |  | Уникальный номер типа доступа к КЭШ информации |
| Cache\_Lifetime | int(11) |  |  |  | Время хранения КЭШ информации |
| Comment\_Rule\_ID | int(11) |  |  |  | Уникальный номер типа комментирования страницы |
| SitemapPriority | float |  |  | \* | Приоритет в карте сайта |
| SitemapChangefreq | enum |  |  |  | Частота обновления страницы в карте сайта |
| IncludeInSitemap | int(11) |  |  |  | Включить страницу в карту сайта |

Таблица Template хранит информацию о макетах дизайна. Структура расширяется при помощи интерфейса управления системными таблицами. Свойства указаны в таблице 6.

Таблица 6 – Макеты дизайна

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поле | Тип | Ключ | a\_i | Null | Описание |
| Template\_ID | int(11) | \* | \* |  | Уникальный номер макета дизайна |
| Description | varchar(64) |  |  |  | Описание макета дизайна |

Продолжение таблицы 6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parent\_Template\_ID | int(11) |  |  |  | Уникальный номер родителя |
| Settings | text |  |  |  | Настройки по умолчанию макета |
| CustomSettings | text |  |  | \* | Дополнительные настройки компонента |
| Header | text |  |  |  | HTML код шапки сайта |
| Footer | text |  |  |  | HTML код подвала сайта |
| CSS | text |  |  | \* | CSS код |
| File\_Mode | tinyint(1) |  |  |  |  |
| File\_Path | varchar(255) |  |  | \* |  |
| File\_Hash | char(32) |  |  |  |  |

Таблица Sub\_Class – это связка между разделом (Subdivision) и компонентов (Class). Используется для составления соответствия между компонентами и разделами (связь между ними «много: много»). Структура таблицы расширяется при помощи интерфейса управления системными таблицами. Связанные таблицы: Subdivision, Class. Свойства указаны в таблице 7.

Таблица 7 – Связи раздел – компонент

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поле | Тип | Ключ | a\_i | Null | Описание |
| Sub\_Class\_ID | int(11) | \* | \* |  | Уникальный номер связки |
| Subdivision\_ID | int(11) |  |  |  | Уникальный номер страницы |

Продолжение таблицы 7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class\_ID | int(11) | |  |  |  | Уникальный номер компонента |
| Sub\_Class\_Name | varchar(255) | |  |  |  | Название компонента |
| Priority | int(11) | |  |  |  | Приоритет |
| Read\_Access\_ID | int(11) | |  |  |  | Идентификатор правил записи |
| Write\_Access\_ID | int(11) | |  |  |  | Идентификатор правил чтения |
| EnglishName | | varchar(64) |  |  | \* | Псевдоним на английском языке |
| Checked | | smallint(6) |  |  |  | Отображать / Скрыть |
| Catalogue\_ID | | int(11) |  |  |  | Уникальный номер каталога |
| Edit\_Access\_ID | | int(11) |  |  |  | Идентификатор правил доступа |
| Checked\_Access\_ID | | int(11) |  |  |  | Идентификатор правил отображения |
| Delete\_Access\_ID | | int(11) |  |  |  | Идентификатор правил удаления |
| Subscribe\_Access\_ID | | int(11) |  |  |  | Идентификатор правил подписки |
| Moderation\_ID | | int(11) |  |  |  | Идентификатор правил модерирования |
| DaysToHold | | int(11) |  |  | \* | Приоритет |
| AllowTags | | int(11) |  |  |  | Разрешить тэги |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RecordsPerPage | int(11) |  |  |  | Количество записей на странице |
| SortBy | varchar(255) |  |  |  | Сортировка |
| Created | datetime |  |  |  | Дата создания |
| LastUpdated | timestamp |  |  |  | Дата последнего изменения |
| DefaultAction | enum |  |  |  | Действие по умолчанию |
| NL2BR | tinyint(4) |  |  |  | Системное поле |
| UseCaptcha | tinyint(4) |  |  |  | Защищать компонент капчей |
| CustomSettings | text |  |  | \* | Настройки разработчика |
| Class\_Template\_ID | int(10) |  |  |  | Макет отображения компонента |
| isNaked | tinyint(4) |  |  |  | Вывод объекта без дизайна |
| AllowRSS | tinyint(4) |  |  |  | Разрешить RSS |
| AllowXML | tinyint(4) |  |  | \* | Разрешить XML |
| SrcMirror | int(11) |  |  | \* | Изображение раздела когда активен |
| Edit\_Class\_Template | int(11) |  |  | \* | Макет дизайна для редактирования |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cache\_Access\_ID | int(11) |  |  | \* | Идентификатор правил доступа к КЭШ информации |
| Cache\_Lifetime | int(11) |  |  | \* | Время хранения КЭШ информации |
| CacheForUser | tinyint(4) |  |  | \* | Разрешить КЭШ для пользователей |
| Comment\_Rule\_ID | int(11) |  |  | \* | Идентификатор правил комментирования |

Таблицы типа MessageXX предназначены для хранения данных. Названия таблиц имеют формат MessageXX, где XX – номер компонента (Class), который соответствует таблице. Структура таблиц расширяется при помощи интерфейса управления компонентами через административную панель сайта. Связанные таблицы: Sub\_Class, User, классификаторы. Свойства указаны в таблице 8.

Таблица 8 – Таблица компонента

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поле | Тип | Ключ | a\_i | Null | Описание |
| Message\_ID | int(11) | \* | \* |  | Уникальный номер единицы компонента |
| User\_ID | int(11) |  |  |  | Уникальный номер создателя |
| Subdivision\_ID | int(11) |  |  |  | Уникальный номер раздела |
| Sub\_Class\_ID | int(11) |  |  |  | Уникальный номер компонента |
| Priority | int(11) |  |  |  | Приоритет |
| Checked | tinyint(4) |  |  |  | Отображать / скрывать |

Продолжение таблицы 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IP | varchar(15) |  |  | \* | IP адрес создателя |
| UserAgent | varchar(255) |  |  | \* | Название браузера создателя |
| Parent\_Message\_ID | int(11) |  |  |  | Уникальный номер родителя |
| Created | datetime |  |  |  | Дата создания |
| LastUpdated | timestamp |  |  |  | Дата и время последнего обновления |
| LastUser\_ID | int(11) |  |  |  | Уникальный номер последнего редактора |
| LastIP | varchar(255) |  |  | \* | Последний IP редактора |
| LastUserAgent | varchar(255) |  |  | \* | Название браузера создателя |
| Keyword | varchar(255) |  |  |  | Описание |
| ncTitle | varchar(255) |  |  | \* | Заголовок компонента |
| ncKeywords | varchar(255) |  |  | \* | Блок Keywords |
| ncDescription | text |  |  | \* | Блок Description |

Таблица Classificator или по-другому списки хранит значения которые будут использоваться для множественного выбора или для выбора определенного значения из выпадающего списка select. Система предполагает создание неограниченного числа списков разной размерности и различного содержания. Такие списки очень удобны в случае, когда на сайте предполагается выбор связанный с тематикой сайта. Свойства указаны в таблице 9.

Таблица 9 – Список

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поле | Тип | Ключ | a\_i | Null | Описание |
| Classificator\_ID | int(11) | \* | \* |  | Уникальный номер списка |
| Classificator\_Name | char(32) |  |  |  | Название списка |
| Table\_Name | char(32) |  |  |  | Имя таблицы |
| System | smallint(6) |  |  |  | Относить список к системным |
| Sort\_Type | tinyint(3) |  |  |  | Тип сортировки |
| Sort\_Direction | tinyint(3) |  |  |  | Отображать / скрывать |

Таблица Class предназначена для хранения экземпляров сущностей, называемых «Компонент». Поля компонентов хранятся в таблице Field. Связанные таблицы: Module, системные списки (системные классификаторы). Свойства указаны в таблице 10.

Таблица 10 – Экземпляры сущностей «Компонент»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поле | Тип | Ключ | a\_i | null | Описание |
| Class\_ID | int(11) | \* | \* |  | Уникальный номер компонента |
| Class\_Name | varchar(255) |  |  |  | Название компонента |
| Class\_Group | varchar(64) |  |  |  | Название группы |
| DaysToHold | int(11) |  |  |  |  |
| AllowTags | smallint(6) |  |  |  | Разрешить теги |
| FormPrefix | text |  |  | \* | HTML код шапки компонента |
| FormSuffix | text |  |  | \* | HTML код подвала компонента |
| RecordTemplate | text |  |  | \* | HTML код экземпляра компонента |

Продолжение таблицы 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RecordsPerPage | int(11) |  |  |  | Количество записей на одной странице |
| SortBy | varchar(255) |  |  |  | Сортировка по полю |
| RecordTemplateFull | text |  |  | \* | HTML код полного отображения экземпляра |
| TitleTemplate | varchar(255) |  |  | \* | Макет заголовка |
| TitleList | varchar(255) |  |  | \* | Список заголовков |
| UseAltTitle | tinyint(4) |  |  | \* | Использовать альтернативный заголовок |
| AddTemplate | text |  |  |  | HTML код добавления экземпляра |
| EditTemplate | text |  |  |  | HTML код редактирования экземпляра |
| AddActionTemplate | text |  |  |  | Исполняемый PHP код после добавления экземпляра |
| EditActionTemplate | text |  |  |  | Исполняемый PHP код после редактирования экземпляра |
| SearchTemplate | text |  |  |  | HTML код поиска |
| FullSearchTemplate | text |  |  |  | HTML код расширенного поиска |
| SubscribeTemplate | text |  |  |  | Макет подписок |
| System\_Table\_ID | int(11) |  |  |  | Уникальный номер системного списка |

Продолжение таблицы 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Settings | text |  |  |  | Настройки |
| AddCond | text |  |  |  | Условия добавления |
| EditCond | text |  |  |  | Условия изменения |
| SubscribeCond | text |  |  |  | Условия подписки |
| DeleteCond | text |  |  |  | Условия удаления |
| NL2BR | tinyint(4) |  |  |  | Системное поле |
| UseCaptcha | tinyint(4) |  |  |  | Использовать капчу |
| CheckActionTemplate | text |  |  |  | Исполняемый PHP код после включения экземпляра |
| DeleteActionTemplate | text |  |  |  | Исполняемый PHP код после удаления экземпляра |
| CustomSettingsTemplate | text |  |  | \* | Макет настроек разработчика |
| ClassDescription | text |  |  | \* |  |
| DeleteTemplate | text |  |  |  | Макет удаления |
| ClassTemplate | int(10) |  |  |  | Макет дизайна класса |
| Type | enum |  |  | \* | Тип |
| File\_Mode | tinyint(1) |  |  |  | Доступ к папке |
| File\_Path | varchar(255) |  |  | \* | Папка на сервере |
| File\_Hash | char(32) |  |  |  | ХЭШ значение папки |
| CacheForUser | tinyint(3) |  |  |  | КЭШ для пользователей |

Связи между таблицами в базе данных использованы в двух типах таких как: один к одному и многие ко многим. Связи таблиц показаны на рисунке 11.

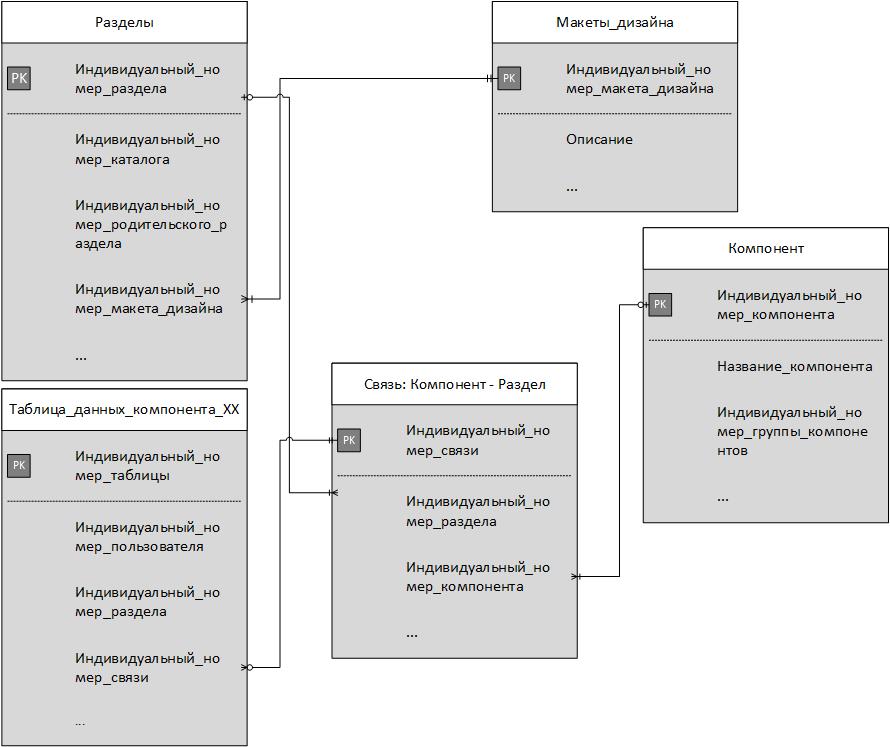


Рисунок 11 – Связи таблиц в базе данных

### 3.2 Оценка качества информационной системы

Разработка Интернет-магазина включает в себя создание большого количества функций, которые в обязательном порядке требуют тестирования и отладки после разработки. Необходимо в первую очередь проверять работоспособность - корректность ссылок, удобство, орфографию и корректное выполнение функций. Важно, чтобы приложение корректно отображался в разных браузерах и различных устройствах.

Заказчик проверяет соответствие структуры сайта техническому заданию, правильность работы функциональных модулей сайта, полноту материала представленного на страницах, правильность отображения в различных браузерах, соответствие кода сайта стандартам кодирования и т.д. При выявлении ошибок производится их немедленное устранение.

Разработанный web-сайт обладает следующими функциями:

* просмотр контента сайта;
* добавление выбранных товаров в корзину для последующей организации покупки;
* заполнение анкеты по запросу отсутствующих запчастей;
* информирование клиентов о статусе покупки(email-рассылка).

Для администраторов и модераторов сайта:

* добавление и редактирование контента;
* обновление каталога товаров, а также перечня предоставляемых услуг.

Полученный web-сайт полностью удовлетворяет бизнес-процессам организации, следовательно, полностью соответствует поставленным задачам

### 3.3 Внедрение информационной системы

3.3.1 Публикация web-сайта в сети Интернет

Обычно под понятием услуги хостинга подразумевают как минимум услугу размещения файлов сайта на сервере, на котором запущено ПО, необходимое для обработки запросов к этим файлам (веб-сервер).

Хостинг базы данных, размещение файлов, хостинг электронной почты, услуги DNS могут предоставляться отдельно как самостоятельные услуги, либо входить в комплексную услугу. Для размещение web-приложения в сети Интернет был выбран платный хостинг «IHC» расположенный по адресу ресурса в сети интернет www.ihc.ru, который полностью удовлетворяет минимальным техническим требованиям к хостингу.

Короткая информация о выбранном хостинге:

* панель управления;
* PHP 7 / 5.6 / 5.5 / 5.4 / 5.3;
* MySQL 5.6;
* почтовые ящики;
* ежедневное резервное копирование;
* антиспам и антивирус для почты.

3.3.2 Руководства пользователей для администрации сайта

Для начала редактирования сайта необходимо зайти в панель администрирования. Для этого нужно дописать к адресу сайта /netcat/admin/. В нашем случае получится http://kuzovchel.ru/netcat/admin/ и в соответствующие поля ввести выданные «Логин» и «Пароль» и нажать кнопку «Авторизоваться».

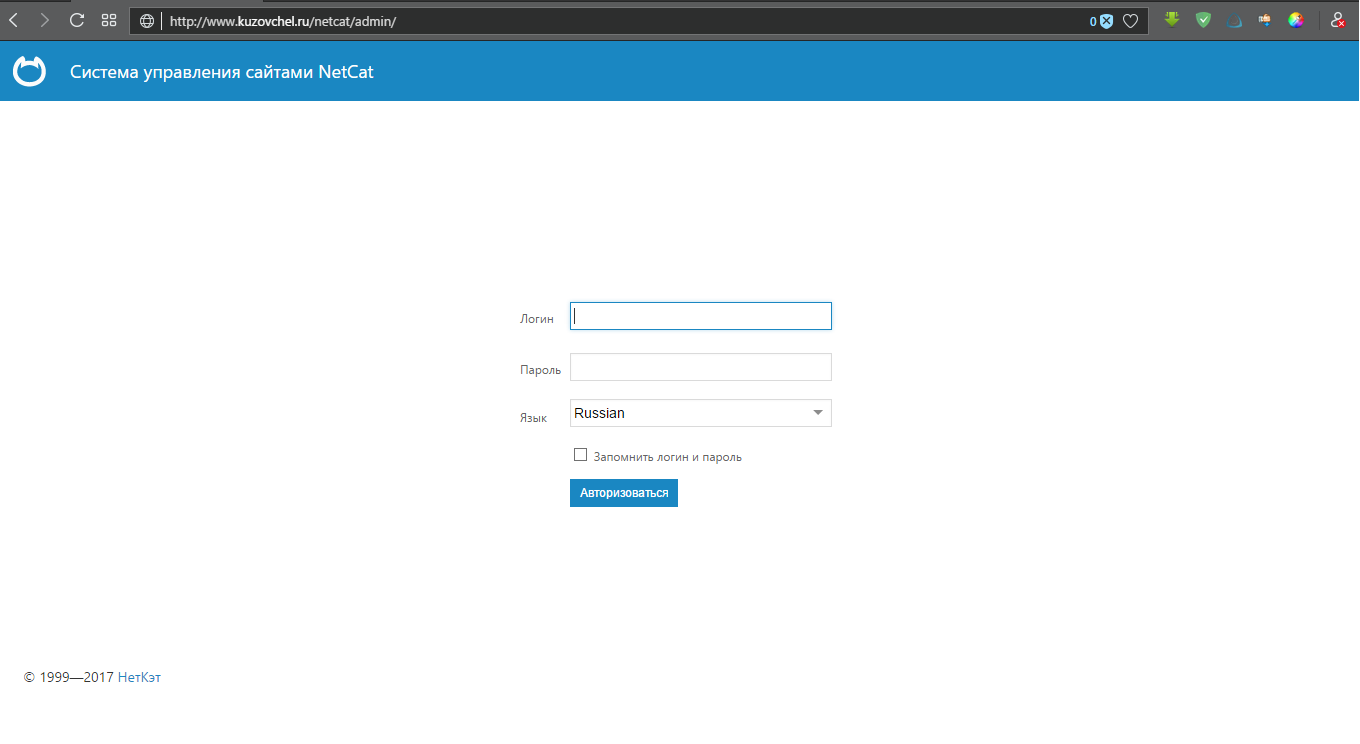


Рисунок 12 – Страница авторизации в CMS

При выполнении успешной авторизации в системе происходит переход в административную панель сайта, изображенную на рисунке 13.

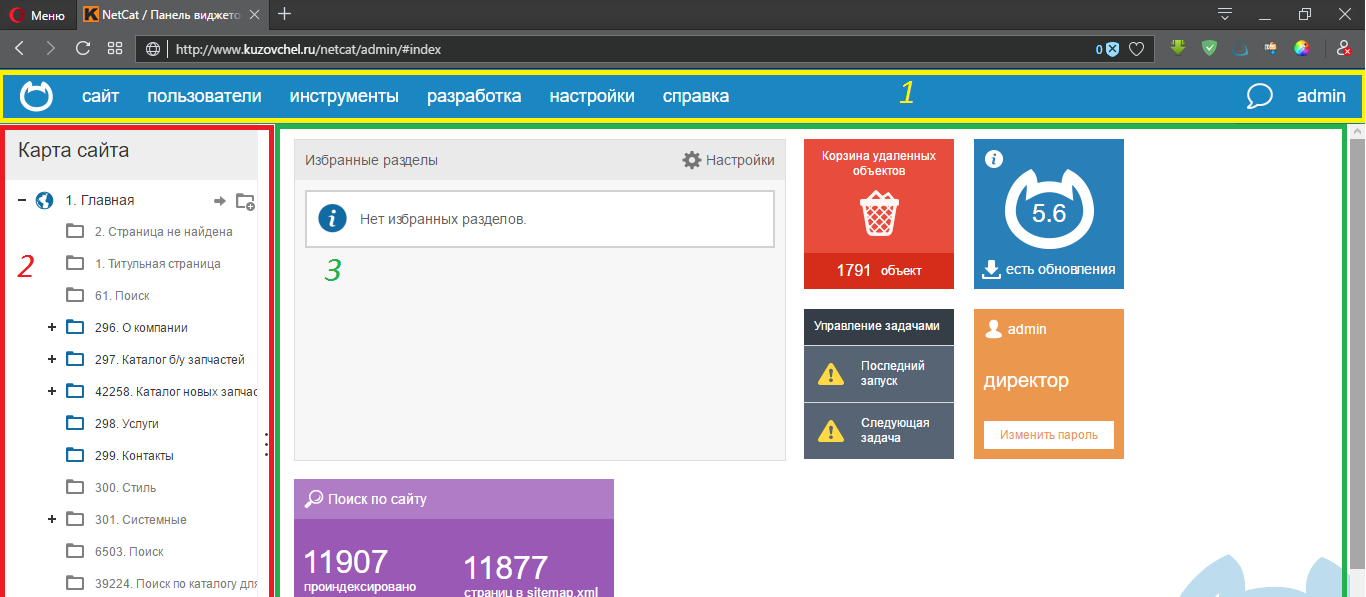


Рисунок 13 – Интерфейс CMS NetCat

Интерфейс системы (рисунок 13) условно поделен на 3 основных блока.

Блок 1 – навигационное меню системы (рисунок 14). Данный блок отвечает за навигацию по разделам CMS. Для того чтобы перейти в желаемый раздел необходимо кликнуть левой кнопкой мыши на один из пунктов меню, а затем из выпадающего списка выбрать требуемый раздел.

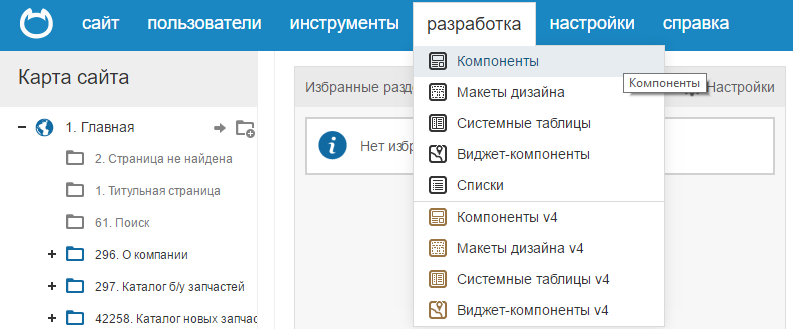


Рисунок 14 – Блок навигации в системе NetCat

Блок 2 – «дерево» текущего раздела (рисунок 15). Данный блок будет изменяться в зависимости от раздела в котором находится пользователь. При клике на «ветвь» дерева справа от дерева, в рабочей области (рисунок 16), будет подгружена информация из выбранной ветви. Если «ветвь» имеет слева от своего названия значок «+», то это означает что данный элемент является родительским и содержит в себе подразделы. Для их отображения необходимо кликнуть на значок «+», либо на саму ветвь.

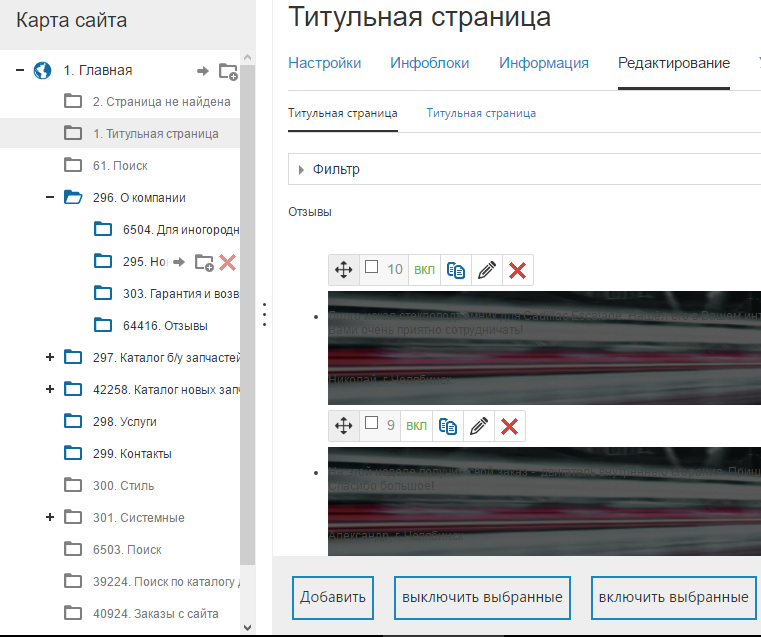


Рисунок 15 – «Дерево» текущего раздела

Блок 3 – рабочая область CMS (рисунок 16). В этом блоке будет происходить взаимодействие с сущностью, выбранной в «дереве» текущего раздела. В нижней части блока расположена панель управления рабочей областью, элементы которой изменяются в зависимости от текущего раздела.

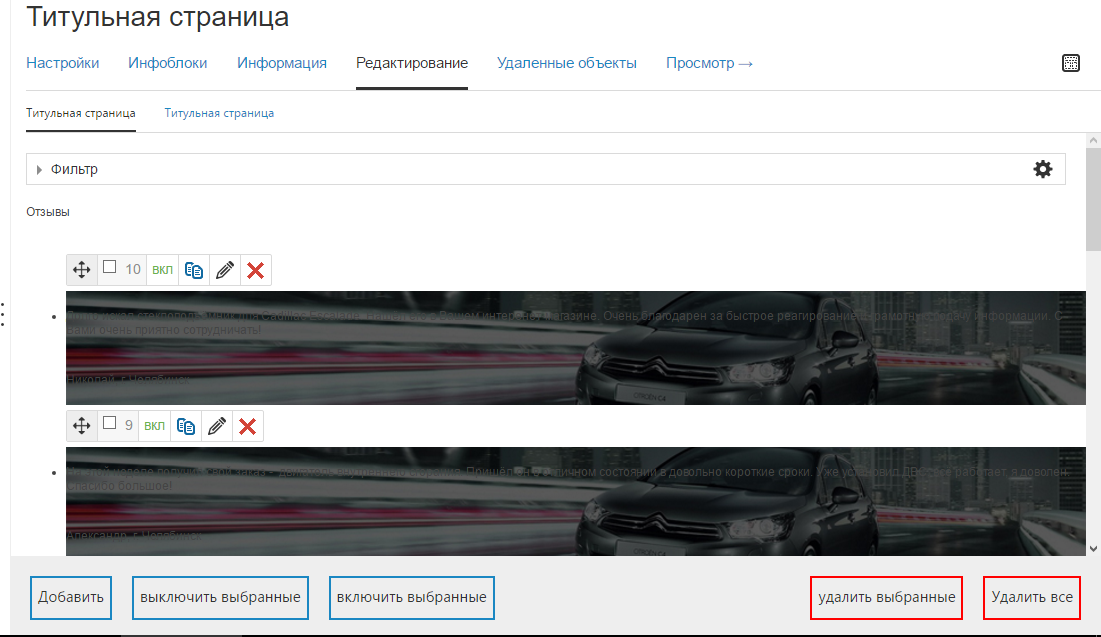
-

Рисунок 16 – Рабочая область CMS

Для выполнения основной функции сайта – продажи товара необходимо добавить необходимый товар в каталог. Для этого в «дереве» карты сайта необходимо выбрать раздел, в который мы будем добавлять товары (рисунок 17). Раздел зависит от новизны товара, так раздел с ID 297 является каталогом б/у запчастей, а раздел с ID 42258 – каталог новых запчастей. После перехода в необходимый категорию товаров при помощи кнопки добавления подразделов создать раздел с инфоблоком «Марки автомобиля», который отвечает за вывод элемента в каталоге. Для того чтобы добавить товар нужно добавить в раздел с маркой авто ещё один уровень с инфоблоком «товар», который отвечает непосредственно за вывод продаваемой продукции. Если не добавлять в товар объект, то раздел будет считаться корневым – это необходимо если требуется создать дополнительные подкатегории товара.

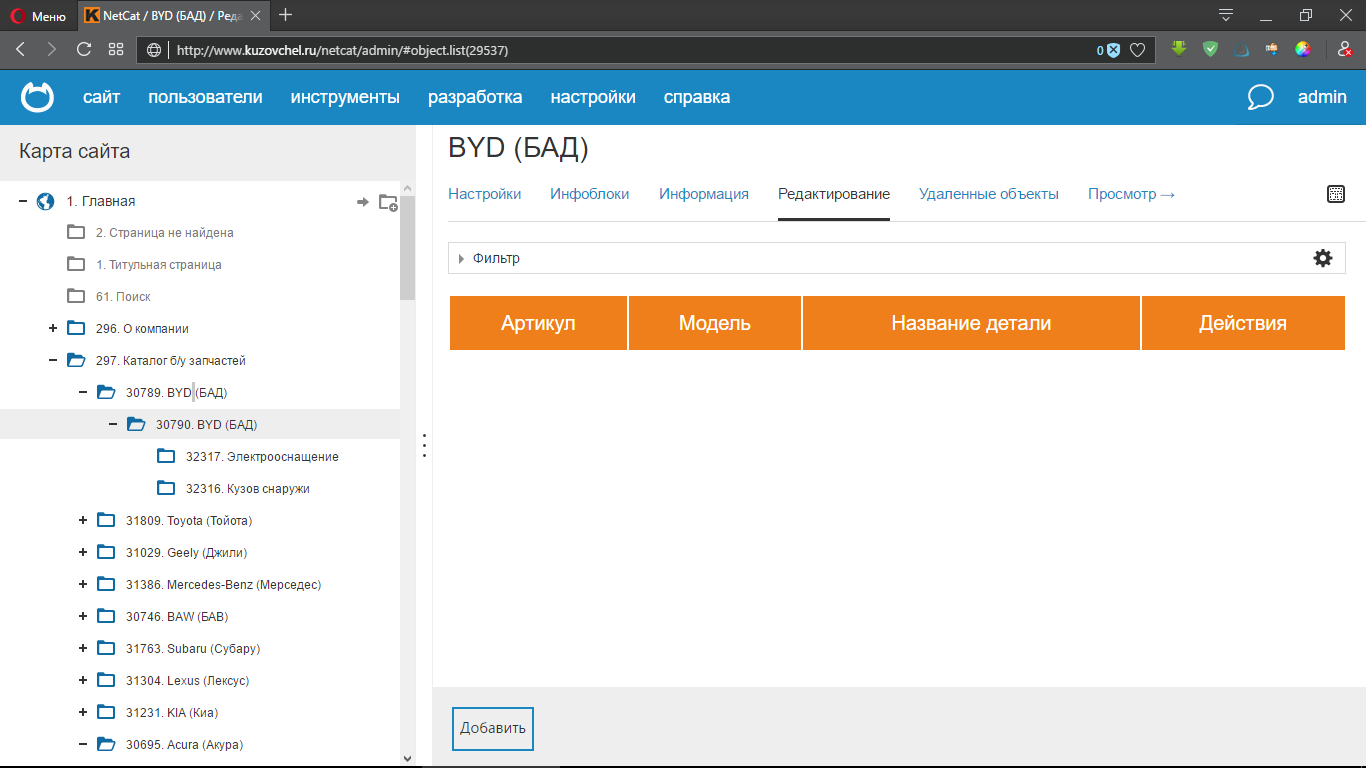


Рисунок 17 – Интерфейс добавления раздела для товара

При нажатии на кнопку «добавить» в панели управления в нижней части окна откроется интерфейс добавления товара (рисунок 18). После заполнения всех указанных в интерфейсе полей товар создастся в качестве объекта в текущем инфоблоке и автоматически отобразится в каталоге товаров на сайте, где пользователь может взаимодействовать с ним.

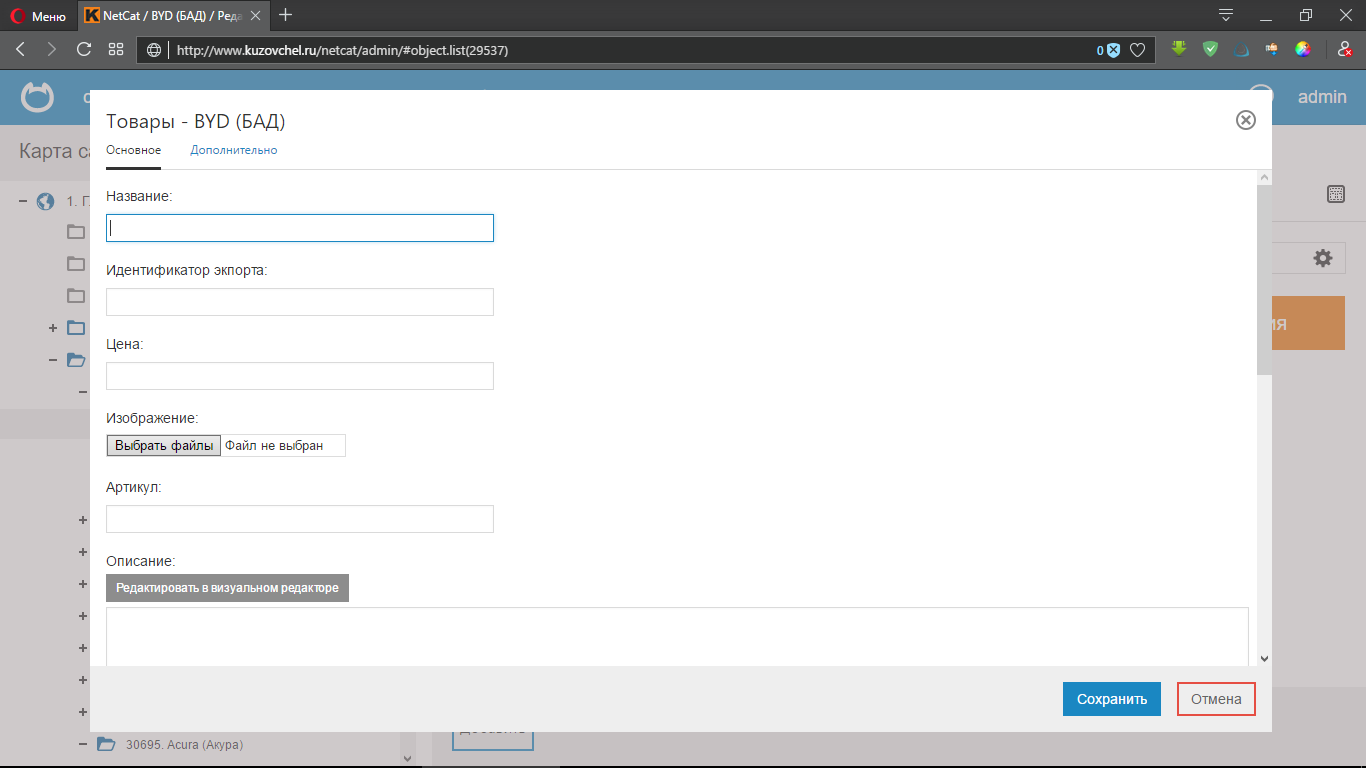


Рисунок 18 – Интерфейс добавления товара

### 3.4 Оценка экономических затрат на проект

Чтобы оценить экономический эффект от внедрения нового программного продукта, следует рассчитать затраты на разработку проекта и эксплуатационные затраты, связанные с обеспечением его нормального функционирования.

Расчет затрат на разработку проекта.

Величину затрат на создание программы определяют на основе метода калькуляций. В этом случае затраты определяются расчетом по отдельным статьям расходов и их последующим суммированием.

Рассмотрим затраты по отдельным статьям расходов в каждой категории: постоянные и переменные издержки.

Постоянные издержки:

− амортизационные отчисления на компьютер и программное обеспечение;

− затраты на текущий ремонт.

Амортизационные отчисления на компьютер и программное обеспечение производятся ускоренным методом с тем условием, что срок морального старения происходит через четыре года.

Балансовая стоимость ЭВМ включает отпускную цену, расходы на транспортировку, монтаж оборудования и его наладку и вычисляется по формуле:

, (1)

где  *−* балансовая стоимость ЭВМ, р.;

*−* рыночная стоимость компьютера, р./шт.;

−затраты на доставку и установку компьютера, %.

Компьютер, на котором будет разработан проект, был приобретен ранее по цене 22500 р., затраты на установку и наладку включены в стоимость компьютера.

Отсюда:

р./шт.

Сумму годовых амортизационных отчислений определяем по формуле:

, (2)

где − сумма годовых амортизационных отчислений, р.;

− балансовая стоимость компьютера, р./шт.;

− норма амортизации, %.

р.

Следовательно, сумма амортизационных отчислений за период создания проекта будет равняться произведению амортизационных отчислений в день на количество дней эксплуатации компьютера и программного обеспечения при создании программы.

р.

Общая стоимость программного обеспечения, которое будет использоваться для разработки программного продукта, составляет 9000 р. На программное обеспечение, как и на компьютеры, производятся амортизационные отчисления.

р.

 р.

Амортизация за время эксплуатации компьютера и программного обеспечения при создании программы вычисляется по формуле:

,(3)

где *−* амортизационные отчисления на компьютер за время его эксплуатации;

*–* амортизационные отчисления на программное обеспечение за время его эксплуатации.

Таким образом, амортизационные отчисления за период разработки составят:

АП= 924,67 + 369,86 = 1294,53 р.

Затраты на текущий и профилактический ремонт принимаются 5 % от стоимости ЭВМ, вычисляются по формуле:

,(4)

где − балансовая стоимость компьютера, р./шт.;

− процент на текущий ремонт, %;

− срок исполнения работ.

**р.

Таким образом, полученные данные составляют постоянные издержки и приведены в таблице 11. Большую долю в постоянных издержках занимают амортизационные отчисления на используемый компьютер и программное обеспечение.

Таблица 11 – Постоянные издержки

|  |  |  |
| --- | --- | --- |
| Вид постоянных издержек | Денежная оценка, р. | Удельный вес, % |
| Амортизационные отчисления | 1294,53 | 87,6 |
| Текущий ремонт | 184,93 | 12,4 |
| Итого: | 1479,46 | 100 |

Переменные издержки:

− затраты на приобретение материалов;

− затраты на потребляемую электроэнергию;

− затраты на оплату труда разработчиков программы;

− отчисления по налогам.

К статье «Приобретение материалов» относятся стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредствен­но в процессе изготовления программного продукта. В стоимость материальных затрат включаются транспортные расходы (5 % от стоимости материалов). Расчет статьи «Материалы» приводится в таблице 12.

Таблица 12 – Материалы и покупные изделия

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Количество | Цена за еди­ницу, р. | Стоимость, р. |
| Бумага для принтера, пачка | 1 | 110 | 110 |
| Заправка тонера для картриджа, шт. | 1 | 226 | 226 |
| Флеш-карта (1 Гб), шт. | 1 | 450 | 450 |
| Итого: | | | 786 |

Затраты на материалы и покупные изделия с учетом транспортных расходов равны:

** р.

К статье «Затраты на потребляемую электроэнергию» относится стоимость потребляемой электроэнергии компьютером за время разработки программы.

Стоимость электроэнергии, потребляемой за год, определяется по формуле:

, (5)

где − суммарная мощность ЭВМ, кВт;

− время работы компьютера, часов;

− стоимость 1 кВт /ч. электроэнергии, р.

Согласно техническому паспорту ЭВМ, кВт.

Стоимость 1 кВт/ч электроэнергии  р.

Рабочий день равен восьми часам, следовательно, стоимость электроэнергии за период работы компьютера во время создания программы будет равна:

р.

К статье «Затраты на оплату труда разработчиков программы» относятся:

− основная заработная плата разработчиков программы;

− дополнительная заработная плата разработчиков, входящая в фонд заработной платы.

Расчет основной заработной платы выполняется от доли выполнения работы и величины месячного должностного оклада исполнителя. Среднее количество рабочих дней в месяце равно 21.

Произведение срока исполнения на сумму дневной заработной платы определяет затраты по основной зарплате для каждого работника на все время разработки.

При расчете основной заработной платы за период разработки программы учитываем, что руководитель участвует в разработке проекта только на этапах системного анализа и анализа требований, которые занимают 14 % всего времени (таблице 13).

В дополнительную заработную плату входит районный коэффициент, равный 30 %, и северный коэффициент, также равный 30 %.

Затраты на фонд заработной платы за время разработки программного продукта приведены в таблице 14.

Таблица 13 – Расчет оклада сотрудников за один рабочий день

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Должность | Оклад, р. | Стоимость 1 рабочего дня, р. | Дни | Сумма, р. |
| Руководитель | 5700,00 | 271,43 | 9 | 2442,87 |
| Программист | 5000,00 | 238,10 | 60 | 14286,00 |
| Web дизайнер | 5000,00 | 238,10 | 9 | 2142,90 |
| Итого: | | | | 18871,77 |

Таблица 14 – Расчет фонда заработной платы за период разработки про­граммы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вид заработной платы | Руководитель, р. | Программист, р. | Web дизайнер, р. | Сумма, р. | Удельный вес, % |
| Основная заработная плата | 2442,87 | 14286,00 | 14286,00 | 31014,87 | 62,5 |
| Дополнительная оплата | 1465,72 | 8571,60 | 8571,60 | 18608,92 | 37,5 |
| Итого: | 3908,59 | 22857,60 | 22857,60 | 49623,79 | 100 |

Отчисления по налогам включают отчисления по единому социальному налогу (ЕСН), который составляет 26,2 %. Затраты по данной статье определяются ус­тановленным нормативом от расходов на оплату труда работников, непосредст­венно занятых созданием научно-технической продукции, приведены в таблице 15 и вычисляются по формуле:

*,* (6)

где − затраты по ЕСН, р.;

−фонд заработной платы, р.;

− налоговые отчисления, %.

Таблица 15 - Отчисления по ЕСН

|  |  |  |
| --- | --- | --- |
| Исполнитель | Фонд заработной платы, р. | Общая сумма ЕСН, р. |
| Руководитель | 3 908,59 | 1 024,05 |
| Программист | 22857,60 | 5988,69 |
| Web дизайнер | 22857,60 | 5988,69 |
| Итого: | 49623,79 | 13001,43 |

Список переменных издержек и их содержание приведены в таблице 16.

Таблица 16 – Переменные издержки

|  |  |  |
| --- | --- | --- |
| Вид переменных издержек | Величина, р. | Удельный вес, % |
| Затраты на приобретение материалов | 798,00 | 2,30 |
| Затраты на потребляемую электроэнергию | 157,44 | 0,45 |
| Затраты на оплату труда разработчиков | 49623,79 | 77,06 |
| Отчисления по налогам | 7012,74 | 20,19 |
| Итого: | 57591,98 | 100 |

Рассчитаем общие затраты. На статью «Общие затраты» относятся все издержки, которые были произведены при создании программного продукта.

Полная себестоимость разработки определяется суммированием постоянных и переменных издержек и вычисляется по формуле:

, (7)

где − себестоимость программного проекта;

− постоянные затраты;

*−* переменные затраты.

Структура полных затрат приведена в таблице 17, из которой видно, что при создании программного продукта наибольший удельный вес занимают переменные издержки.

Таблица 17 – Структура полных издержек

|  |  |  |
| --- | --- | --- |
| Вид издержек | Величина, р. | Удельный вес, % |
| Постоянные | 2455,9 | 4,09 |
| Переменные | 57591,98 | 95,91 |
| Итого | 60047,94 | 100 |

Таким образом, себестоимость создаваемого программного продукта равна р.

Расчет эксплуатационных затрат.

К эксплуатационным затратам относятся затраты, связанные с обеспече­нием нормального функционирования проекта. Это могут быть затраты на ве­дение информационной базы, эксплуатацию комплекса технических средств, эксплуатацию систем программно-математического обеспечения, реализацию технологического процесса обработки информации по задачам, эксплуатацию системы в целом.

Например, благодаря внедрению проекта, отдел сможет не увеличивать штат в связи с увеличением объема работ, следовательно произойдет экономия затрат на заработную плату сотрудников. В базовом варианте должны работать 2 сотрудника, в новом − 1. Должностной оклад сотрудника 5 000 р.

Сумма амортизационных отчислений рассчитывается следующим образом:

, (8)

где − балансовая стоимость j-ого вида оборудования, р.;

− норма годовых амортизационных отчислений для j-ого вида оборудо­вания, %;

− количество единиц оборудования j-ого вида;

− время работы j-ого вида оборудования, ч.;

− эффективный фонд времени работы оборудования в год, ч.

Эффективный фонд времени работы оборудования можно вычислить:

, (9)

где – количество рабочих дней в году (Др = 264 дн.);

− норматив среднесуточной загрузки (Нэ = 8 ч.).

Таким образом, эффективный фонд времени работы единицы оборудования равен:

ч.

Рассчитаем амортизационные отчисления по применению оборудования.

Для базового варианта: *g* ***=*** 2; а = 25 %; С = 22 500 р.

Сумма амортизационных отчислений составит:



Для проекта: *g* ***=*** 1; а = 25 %; С = 22 500 р.

Сумма амортизационных отчислений составит:



Затраты на силовую энергию рассчитываются по формуле:

*,* (10)

где − установленная мощность j-го вида технических средств, кВт;

− время работы j-го вида технических средств, ч.;

− коэффициент использования установленной мощности оборудования;

− стоимость 1 кВт/ч. электроэнергии, р.

P = 0,4 кВт; Сэл = 0,82 р.

Для базового варианта:



Для проекта:



Затраты на текущий ремонт оборудования рассчитываются по формуле:

, (11)

где − норматив затрат на ремонт j-го вида технических средств (= 0,05);

− балансовая стоимость j-го компьютера, р./шт.;

− время работы j-го вида технических средств, ч.

Для базового варианта:



Для проекта:



Расчет затрат на заработную плату приведен в таблице 18.

Таблица 18 – Затраты на заработную плату сотрудника

|  |  |  |
| --- | --- | --- |
| Вид затрат в месяц | Сумма, р. | |
| Базовый  вариант | Разрабатываемый проект |
| Основная заработная плата | 10000,00 | 5000,00 |
| Дополнительная оплата (30 % + 30 %) | 6000,00 | 3000,00 |
| Отчисления по ЕСН (26,2 %) | 2 620,00 | 1310,00 |
| Итого в месяц: | 18620,00 | 9310,00 |
| Итого в год: | 223440,00 | 111720,00 |

Расчет эксплуатационных затрат в год для разрабатываемого проекта и базового по статьям представлен в таблице 19.

Таблица 19 – Расчет эксплуатационных затрат

|  |  |  |
| --- | --- | --- |
| Статьи затрат | Базовый проект | Разрабатываемый проект |
| Основная и дополнительная зарпла­та с отчислениями по ЕСН, р. | 223440,00 | 11720,00 |
| Амортизационные отчисления, р. | 11250,00 | 5625,00 |
| Затраты на электроэнергию, р. | 1385,47 | 692,74 |
| Затраты на текущий ремонт, р. | 2250,00 | 1125,00 |
| Итого: | 238325,47 | 119162,74 |

Таким образом, эксплуатационные затраты в год составят:

− для разрабатываемого проекта 119162,74 р.;

− для базового проекта 238325,47 р.

Расчет показателя экономического эффекта.

Внедрение информационных технологий сопряжено скапитальными вложениями как на приобретение техники, так и на разработку проектов, выполнение подготовительных работ и подготовку кадров.

Под эффективностью автоматизированного преобразования экономической информациипонимают целесообразность приме­нения средств вычислительной и организационной техники при формировании, передаче и обработке данных. Различают расчет­ную и фактическую эффективность. Первую (расчетную) опреде­ляют на стадии проектирования автоматизации информационных работ, т. е. разработки техно рабочего проекта, вторую (фактическую) − по результатам внедрения техно рабочего проекта.

Обобщенным критерием экономической эффективности явля­ется минимум затрат живого и овеществленного труда. При этом установлено, что чем больше участков управленческих работ автоматизировано, тем эффективнее используется техническое и программное обеспечение.

Экономический эффект от внедрения вычислительной и организационной техники подразделяют на прямой и косвенный.

Под прямой экономической эффективностью программного продуктапонимают эко­номию материально-трудовых ресурсов и денежных средств, полученную в результате сокращения численности управленческого персонала, фонда заработной платы, расхода основных и вспомогательных материалов вследствие автоматизации конкретных видов планово-учетных и аналитических работ.

Не исключено, что внедрение программного продукта на первом этапе не приведет к уменьшению числа работников планово-учетных служб. В этом случае учитывают косвенную эффективность,проявляющуюся в конечных результатах хозяйственной деятельности предприятия. Ее локальными критериями могут быть: сокращение сроков со­ставления сводок, повышение качества планово-учетных и аналитических работ, сокращение документооборота, повышение культуры и производительности труда и т. д. Основным же показателем является повышение качества управления, которое, как и при прямой экономической эффективности, ведет к экономии живого и овеществленного труда. Оба вида рассмотренной эко­номической эффективности взаимоувязаны.

Экономическую эффективностьопределяют с помощью трудовых и стоимостных показателей. Основным при расчетах является метод сопоставления данных базисного и отчетного периодов. В качестве базисного периода при переводе отдельных работ на автоматизацию принимают затраты на обработку информации до внедрения программного продукта (при ручной обработке), а при со­вершенствовании действующей системы автоматизации экономических работ − затраты на обработку информации при достигнутом уровне автоматизации.

Оценка экономической эффективности при создании автоматизированной системы основывается на расчете показателей сравнительной экономической эффективности капитальных вложений.

Для обобщающей характеристики эксплуатационно-технического уровня системы (ЭТУ) используем аддитивно-мультипликативный показатель «значимость технического решения» (ЗТР) для решенной технической задачи,в общем виде рассчитываемый по формуле:

, (12)

где − коэффициент актуальности;

− коэффициент соответствия програм­мам важнейших работ научно-технического прогресса;

− коэффициент сложности;

− коэффициент места использования;

− коэффициент объема использования;

− коэффициент широты охвата охранными мероприятиями.

Расчет коэффициентов приведен в таблице 20.

Таблица 20 – Определение коэффициентов ЭТУ

|  |  |  |
| --- | --- | --- |
| Коэффициенты | Базовый вариант | Разрабатываемый вариант |
|  | 1 | 1,2 |
|  | 1 | 1 |
|  | 1 | 3 |
|  | 1 | 1 |
|  | 1 | 2 |
|  | 1 | 1 |
| ЗТР | 2 | 5,6 |

Таким образом, из данной таблицы видно, что разрабатываемый проект имеет более высокий показатель эксплуатационно-технического уровня по сравнению с базовым вариантом.

Вычисляем коэффициент эксплуатационно-технического уровня  по формуле:

,(13)

где  и − значимость технического решения для проекта и для базового варианта соответственно.

,

*,* следовательно,разработка проекта является оправданной с технической точки зрения.

Кроме того, для большей уверенности в обоснованности автоматизации можно использовать обобщающий индекс эксплуатационно-технического уровня  (комплексный показатель качества проекта по группе показателей), который рассчитывается по формуле:

,(14)

где  *−* коэффициент весомости *i* -го показателя;

*−* относительный показатель качества, устанавливаемый экспертным путем по выбранной шкале оценивания.

Для оценки рекомендуется пятибалльная шкала оценивания. В таблице 21 представлены результаты расчета балльно-индексным методом.

Коэффициент технического уровня  рассчитываетсяпо формуле:

.(15)

Вычислим коэффициент технического уровня:

*.*

Таблица 21 – Расчет показателя качества

|  |  |  |  |
| --- | --- | --- | --- |
| Показатель качества | Весовой  коэффициент, *bi* | Оценка, *Xi* | |
| Разрабатываемый проект | Базовый проект |
| Удобство работы (пользовательский) | 0,2 | 4 | 2 |
| Надежность (защита данных) | 0,2 | 3 | 3 |
| Функциональные возможно­сти | 0,3 | 4 | 3 |
| Временная экономичность | 0,2 | 5 | 2 |
| Время обучения персонала | 0,1 | 3 | 3 |
| Комплексный показатель качества | | 3,9 | 2,6 |

Для расчета экономического эффекта рассчитаем приведенные затраты  на единицу работ, выполняемых по базовому и разрабатываемому вариантам, по формуле:

,(16)

где − текущие эксплуатационные затраты единицы i-го вида работ, р.;

− суммарные затраты, связанные с внедрением проекта;

− нормативный коэффициент экономической эффективности.

Для базового варианта:



Для проекта:



Экономический эффект от использования разрабатываемой системы определяется по формуле:

,(17)

где, − приведенные затраты на единицу работ, выполняемых с по­мощью базового и проектируемого вариантов процесса обработки информации, р.;

− коэффициент эксплуатационно-технической эквивалентности (формула 3);

− объем работ, выполняемых с помощью разрабатываемого проекта, натуральные единицы.

Экономический эффект от использования разрабатываемой системы:

Э = (238325,47⋅1,5 – 131113,30) ⋅1 = 226374,91 р.

Также необходимо рассчитать срок окупаемости затрат на разработку проекта по формуле:

, (18)

где − единовременные затраты на разработку проекта, р.;

− годовая эффективность, р.

Рассчитываем срок окупаемости затрат на разработку продукта:

Таким образом, срок окупаемости проекта составляет примерно 3 месяца. Малый срок объясняется тем, что проект не включает сложных программных разработок.

Фактический коэффициент экономической эффективности разработки ():

*.* (19)

Нормативное значение коэффициента эффективности капитальных вложений если то делается вывод об эффективности капитальных вложений.

Рассчитаем фактический коэффициент экономической эффективности разработки ():

.

Так как то разработка и внедрение разрабатываемого продукта являются эффективными, т. е. эффект от использования данной системы окупает все затраты, связанные с проектированием и эксплуатацией. В таблице 22 приведены сводные данные экономического обоснования разработки и внедрения проекта.

Таблица 22 – Сводные данные экономического обоснования

|  |  |
| --- | --- |
| Показатель | Величина |
| Затраты на разработку проекта, р. | 60047,94 |
| Общие эксплуатационные затраты, р. | 119162,74 |
| Экономический эффект, р. | 226374,91 |
| Коэффициент экономической эффективности | 6,25 |
| Срок окупаемости, мес. | 3 |

# ВЫВОД ПО ГЛАВЕ 3

В третьей главе работы описаны этапы работы над проектом. В ходе проектирования интернет-магазина были спроектированы диаграмма вариантов использования, определяющая будущий функционал системы, и логическая схема базы данных.

В соответствии с особенностями CMS NetCat разработка приложения имеет модульную структуру. Для каждого модуля можно определить дизайн (параметры отображения), данные, хранимые в базе, и алгоритм выполнения операций.

Интернет-магазин был опубликован в сети по адресу kuzovchel.ru на хостинге «IHC».

Также было написано руководство пользователя, где полностью описываются и иллюстрируются последовательность действий для интернет продаж.

Произведена оценка экономической эффективности проекта, из которой следует, что разработка и внедрение разрабатываемого интернет-магазина являются эффективным. Срок окупаемости составит 3 месяца.

# ЗАКЛЮЧЕНИЕ

В ходе научной работы был разработан Web-сайт, интернет-магазин для компании ООО «Кузовчел». Web-сайт был опубликован в сети по адресу kuzovchel.ru на хостинге «IHC».

В ходе выполнения научной работы был проведен анализ уже разработанных интернет магазинов по продаже автозапчастей, а именно центр «Emix» - Интернет магазин по продаже автозапчастей, сайт автосалона «Автолайн» и сервис по авторазбору «УралАВТО». Для разработки Web-приложения были выбраны следующие инструменты: Microsoft Visio, NetCat, PHP, JavaScript, AJAX, HTML, CSS, MySQL, PhpMyAdmin.

На основе проведенного анализа деятельности компании была спроектирована модель бизнес-процесса «Интернет магазина по продаже автозапчастей». Данная модель легла в основу формирования требований к web-приложению.

В ходе разработки web-приложения созданы компоненты такие как: каталог товаров, корзина товаров, рассылка, форма запроса отсутствующих на складе товаров и формирование отчетов.

Для организации работы модераторов сайта с web-приложением разработано исчерпывающее руководства пользователя для администратора сайта.

Web-приложение успешно внедрено в деятельность компании.

# СПИСОК ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. Федеральный закон РФ 152-ФЗ «О персональных данных».
2. ГОСТ 19.201-1978. Техническое задание. Требования к содержанию и оформлению.; Введ. 01.01.1990. — М. : Издательство стандартов, 1989. — 13 с.
3. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы: Гос. стандарт. – Введ. 01.01.1990 // утверждён приказом роспрома 16.09.2004 №95.

Основная литература

1. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Джен­ нифер Роббинс; [пер. с англ. М. А. Райтман]. — 4-е издание. — М. : Эксмо, 2014. — 528 с.
2. HTML5. Разработка приложений для мобильных устройств. / Вейл Э. — СПб.: Питер, 2015. — 480 с.: ил.
3. MySQL разработка Web-приложений. – 4-е изд., перераб. И доп. — МПб.: БХВ-Петербург, 2013. — 560 с. : Ил.
4. PHP и MySQL. Исчерпывающее руководство. 2-е изд. / Маклафлин Б. — СПб.: Питер, 2014. — 544 с.: ил.
5. UML: Первое знакомство. / Бабич А.В. – М.: Национальный Открытый Университет «Интуит» — 2016. – 209 с. : ил. ; То жэ [Электронный ресурс]. – URL: http://biblioclub.ru/index.php?page=book\_view\_red&book\_id=233305 (18.04.2016).
6. Дженнифер Роббинс [пер. с англ. М. А. Райтман]. HTML5, CSS3 и JavaScript: Исчерпывающее руководство. - М.: Эксмо, 2014. — 528 с.
7. Моделирование на UML. / Ф. Новиков, Д. Иванов – М.: ДМК, 2013. – 281с.
8. Моделирование систем управления : практикум / Д. О. Глухов, И. В. Петухов; под ред. Д. О. Глухова – Йошкар-Ола : Паволжский государственный технологический университет, 2015. – 84 с. ; То жэ [Электронный ресурс]. – URL: http://biblioclub.ru/index.php?page=book\_view\_red&book\_id=437061 (18.04.2016).
9. Моделирование бизнес-процессов: учеб. пособие / В. Г. Мамонова, Н.Д. Ганелина. – Новосибирск: Изд-во НГТУ, 2012. – 43с. ; То жэ [Электронный ресурс]. – URL: http://biblioclub.ru/index.php?page=book\_view\_red&book\_id=228975 (24.05.2016).
10. Моделирование систем. Объектно-ориентированный подход. Учебное пособие / Ю. Б. Колесов, Ю. Б. Сениченков. — СПб.: БХВ-Петербург, 2012. — 192 с.: ил.
11. Моделирование систем и процессов : учебник для академического бакалавриата / В. Н. Волкова, Г. В. Горелова, В. Н. Козлов [и др.] ; под ред. В. Н. Волковой, В. Н. Коз- лова. — М. : Издательство Юрайт, 2015. — 449 с.
12. Управление процессами : учебное-практическое пособие / М. В. Самсонова. – Ульяновск : УлГТУ, 2014. -187 с. ; То жэ [Электронный ресурс]. – URL: http://biblioclub.ru/index?page=book\_view\_red&book\_id=363491 (18.04.2016).
13. Методология и инструментарий моделирования бизнес-процессов: учебное пособие / Цуканова О. А. – СПб.: Университет ИТМО, 2015. – 100 с.
14. Применение UML 2.0 и шаблонов проектирования / Кпег Ларман 3-е изд. — М.: Вильямс, 2013. — 736 с.
15. Разработка и управление требеваниями. Практическое руководство пользователя/Элизабет Халл, Кен Джексон, Джемир Дик. — СПб.: БХВ-Петербург, 2012. — 229 с.: ил.
16. Экономическая оценка новационных технических решений: монография: в 2 ч. / Э.М. Гайнутдинов, Л.И. Поддерегина. – Минск: БНТУ, 2013. – Ч. 1. – 332 с.
17. Язык UML: руководство пользователя. / Г. Буч, Д. Рамбо, А.Джекобсон – М.: ДМК, 2012. – 257с.

Дополнительная литература

1. PHP и JQuery для профиссионалов. : Пер. с англ. – М. : ООО “И.Д. Вильямс”, 2011. – 352 с. : ил.
2. Бенкен, Е. PHP, MySQL, XML: программирование для Интернета (+CD). / Е. Бенкен, Е. Кондукова. – М.: BHV, 2011 г. – 304 c.
3. Бизнес-процессы. – Электронный ресурс. – Режим доступа: http://www.betec.ru (27.05.2016).
4. Битрикс Разработчикам - Документация по CMS «1С-Битрикс: Управление сайтом» // Документации на русском языке 1c-bitri. – Электронный ресурс. – Режим доступа: http://www.1c-bitrix.ru (27.05.2016).
5. Документации на русском языке WordPress // Русский. – Электронный ресурс. – Режим доступа: https://ru.wordpress.org (27.05.2016).
6. Документация разработчика // Сайт поддержки российского сообщества Joomla. – Электронный ресурс. – Режим доступа: http://joomla.ru (27.05.2016).
7. Модели бизнес-процессов предприятния // Business Studio официальный сайт. – Электронный ресурс. – Режим доступа: http://www.businessstudio.ru. – (27.05.2015).
8. Официальная документация к системе NetCat 5 для разработчика // CMS NetCat - система управления сайтами. – Электронный ресурс. – Режим доступа: http://netcat.ru/developers/docs/ (27.05.2016).
9. Проектирование и исследование бизнес-процессов : [электронный ресурс] учеб. пособие / А.С. Козлов. – 4-е изд , стер. – М : Флинта, 2011 – 272с. – http://biblioclub.ru/index.php?page=book\_view\_red&book\_id=103541 (25.05.2016).
10. Сайт денвер. – Электронный ресурс.–Режим доступа: http://www.denwer.ru/ (27.05.2016).