

**Министерство образования и науки Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Факультет математики и информатики  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.02 – Информационные системы и технологии  
Профиль: Информационные системы и технологии

ДОПУСТИТЬ К ЗАЩИТЕ  
Зав. кафедрой  
\_\_\_\_\_ А.В. Бушманов  
« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Разработка CMS-системы для создания сайта-визитки

Исполнитель  
студент группы 255-об

\_\_\_\_\_

(подпись, дата)

К.О. Шишковский

Руководитель  
доцент, канд. техн. наук

\_\_\_\_\_

(подпись, дата)

Л.А. Соловцова

Нормоконтроль  
инженер кафедры

\_\_\_\_\_

(подпись, дата)

В.В. Романико

Благовещенск 2016

**Министерство образования и науки Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Факультет математики и информатики  
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ  
Зав.кафедрой  
\_\_\_\_\_ А.В.Бушманов  
« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

**З А Д А Н И Е**

К выпускной квалификационной работе студента Шишковского Кирилла Олеговича

1. Тема выпускной квалификационной работы: Разработка CMS-системы для создания сайта-визитки  
(утверждена приказом от 03.06.2016 № 1215-уч )
2. Срок сдачи студеном законченной работы: 29.06.2016 г.
3. Исходные данные к выпускной квалификационной работе: отчет по преддипломной практику, ГОСТы, дополнительная литература.
4. Содержание выпускной квалификационной работы: анализ деятельности предприятия, проектирование информационной системы, разработка программного обеспечения.
5. Перечень материалов приложения: 5 рисунков.
6. Консультанты по выпускной квалификационной работе: по проектированию – Соловцова Любовь Александровна, канд. техн. наук, доцент.
7. Дата выдачи задания: сентябрь 2015 г.

Руководитель выпускной квалификационной работы: Соловцова Любовь Александровна, канд. техн. наук, доцент.

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению: \_\_\_\_\_  
(подпись студента)

## РЕФЕРАТ

Бакалаврская работа содержит 73 с., 52 рисунка, 11 таблиц, 4 приложения, 34 источника.

### CMS-СИСТЕМА, ШАБЛОН ПРОЕКТИРОВАНИЯ MVC, АЛГОРИТМ, РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Объектом исследования является предприятие ООО «Зебра», которое занимается созданием рекламы для организаций, а так же разработкой рекламных сайтов.

Целью данной работы является проектирование и разработка CMS-системы для производственного отдела в соответствии с требованиями.

Методологией разработки являются шаблон проектирования MVC и основные методы разработки веб-приложений.

В результате проектирования и разработки была получена CMS-система основанная на шаблоне проектирования MVC.

*ВКР.125047.09.03.02.ПЗ*

Изм	Лист	№ докум	Подп.	Дата				
Разраб		Шишковский К.О			<b>РАЗРАБОТКА CMS-СИСТЕМЫ ДЛЯ СОЗДАНИЯ САЙТА-ВИЗИТКИ</b>	Лит.	Лист	Листов
Пров.		Соловцова Л.А.				У	3	78
Консульт.						<b>АмГУ кафедра ИУС</b>		
Н.контр.		Романико В.В.						
Зав.каф.		Бушманов А.В.						

## СОДЕРЖАНИЕ

Введение	8
1 Анализ деятельности предприятия	10
1.1 Техничко-экономическая характеристика объекта	10
1.1.1 Организационная структура	10
1.1.2 Анализ внешнего и внутреннего документооборота	12
1.2 Анализ локальной вычислительной сети	14
1.3 Анализ аппаратного обеспечения	15
1.4 Анализ программного обеспечения	16
2 Проектирование информационной системы	18
2.1 Обоснование необходимости создания системы	18
2.2 Обоснование выбора среды разработки	18
2.3 Характеристика функциональных подсистем проектируемой ИС	20
2.4 Характеристика обеспечивающих подсистем проектируемой ИС	22
2.4.1 Подсистема организационного обеспечения	22
2.4.2 Лингвистическое обеспечение	22
2.5 Информационное обеспечение	22
2.5.1 Инфологическое проектирование	23
2.5.2 Логическое проектирование	24
2.5.3 Физическое проектирование	24
2.6 Программное обеспечение	25
2.6.1 Структура программного обеспечения	25
2.6.2 Проектирование программного обеспечения	26
3 Разработка программного обеспечения	29
3.1 Описание программы	29
3.1.1 Логическая структура программы	30
3.1.2 Описание модулей, функций, обработчиков событий	31
3.2 Описание интерфейса	61
Заключение	69

					<i>ВКР. 125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		4

Библиографический список	70
Приложение А DFD диаграмма внешнего документооборота предприятия	74
Приложение Б DFD Диаграмма внутреннего документооборота предприятия	75
Приложение В IDEF0 диаграммы CMS-системы	76
Приложение Г Диаграмма классов CMS-системы	78

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		5

## НОРМАТИВНЫЕ ССЫЛКИ

В настоящей выпускной квалификационной работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 2.104-68 ЕСКД Основные надписи

ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам

ГОСТ 2.106-96 ЕСКД Текстовые документы

ГОСТ 2.111-68 ЕСКД Нормоконтроль

ГОСТ 2.306-68 ЕСКД Обозначение графических материалов и правила нанесения их на чертежах

ГОСТ 2.605-68 ЕСКД Плакаты учебно-технические. Общие технические требования

ГОСТ 19.001-77 ЕСПД Общие положения

ГОСТ 19.004-80 ЕСПД Термины и определения

ГОСТ 19.101-77 ЕСПД Виды программ и программных документов

ГОСТ 19.102-77 ЕСПД Стадии разработки

ГОСТ 19.103-77 ЕСПД Обозначение программ и программных документов

ГОСТ 19.104-78 ЕСПД Основные надписи

ГОСТ 19.105-78 ЕСПД Общие требования к программным документам

ГОСТ 19.106-78 ЕСПД Требования к программным документам, выполненным печатным способом

ГОСТ 19.401-78 ЕСПД Текст программы. Требования к содержанию и оформлению

ГОСТ 19.402-78 ЕСПД Описание программы

ГОСТ 19.502-78 ЕСПД Описание применения. Требования к содержанию и оформлению

ГОСТ 19.505-79 ЕСПД Руководство оператора. Требования к содержанию и оформлению

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		6

## ОПРЕДЕЛЕНИЯ ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

БД – база данных;

CMS – Control Management System;

HTML – HyperText Markup Language;

PHP – Hypertext Preprocessor;

СУБД – система управления базой данных;

ТЗ – техническое задание;

DFD – Data Flow Diagram;

IDEF0 – Integrated computer aided manufacturing Definition;

MVC – Model View Controller.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		7

## ВВЕДЕНИЕ

Глобальные сети в данный момент занимают довольно большую часть информационного рынка, и играют главную роль в оперировании информацией в современном мире. Главным объектом в глобальных сетях являются сайты и их производные, с помощью них реализуется обмен и обработка информации.

Сайты используются во всех сферах человеческой деятельности, начиная от обычных блогов, заканчивая ресурсами для проектирования и расчетов. Поэтому, приобретение сайта является довольно важной задачей для любого предприятия.

Метод разработки сайта зачастую зависит от сложности его функционала. Более сложные виды сайтов не ограничиваются написанием HTML кода и используют другие виды языков, способные расширить функционал.

Большинство современных видов сайтов имеют большой объем информации, который порой необходимо редактировать, и в случае обычного HTML кода эта задача может стать очень долгой.

Поэтому сайту необходима CMS система, которая позволит в удобном виде редактировать информацию.

В качестве объекта исследования выбрано рекламное агентство ООО «Зебра», которое помимо создания дизайна для рекламы занимается созданием рекламных сайтов, сайтов-визиток для организаций. Предмет исследования – CMS системы.

Целью работы является разработка CMS системы для сайта-визитки, необходимой для нужд производственного отдела, в частности конструирование шаблонных сайтов.

Решаемые задачи CMS системы:

- оптимизация механизма вывода сайта в браузере;
- обеспечение удобного редактирования содержимого сайта;
- создание возможности редактирования внешнего вида страниц;

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		8



– создание панели администрирования обеспечивающего функционал CMS системы.

Для реализации задач необходимо знать, для каких целей предназначен сайт, какой функционал должен быть реализован, какие данные должны содержаться в базе данных и какого вида должен быть интерфейс.

Инструментами разработки являются: HTML, PHP, CSS JS языки и СУБД MySQL.

В результате создания CMS системы упростится редактирование страниц сайта, а так же оптимизируется механизм вывода.

					<i>ВКР. 125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		9

# 1 АНАЛИЗ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ

## 1.1 Техничко-экономическая характеристика объекта

ООО «Зебра» является организацией, которая занимается изготовлением рекламных конструкций, размещению наружной рекламы, разработке, поддержке, продвижению веб-сайтов.

Данное предприятие является юридическим лицом, имеет самостоятельный баланс, счета, открываемые в соответствии законодательством Российской Федерации, печать со своим наименованием и изображением.

Основными задачами организации являются:

- консультация и помощь клиентам в сфере рекламы;
- разработка и поддержка рекламных кампаний путем анализа сферы деятельности организации, которой необходима реклама и составлением плана работ;
- разработка и поддержка рекламных сайтов;
- изготовление и размещение рекламных конструкций.

В своей деятельности предприятие руководствуется Конституцией Российской Федерации, федеральными конституционными законами, федеральными законами, Указами и распоряжениями Президента Российской Федерации, постановлениями и распоряжениями Правительства Российской Федерации, Уставом (основным Законом) Амурской области, законами Амурской области, постановлениями и распоряжениями губернатора области, приказами и инструкциями министерства труда и социального развития Российской Федерации.

### 1.1.1 Организационная структура предприятия

Предприятие является обществом с ограниченной ответственностью, во главе стоит генеральный директор занимающийся вопросами управления предприятием и заключением сделок с поставщиками и партнерами. Организационная структура предприятия представлена на рисунке 1.

					<i>ВКР. 125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		10



Рисунок 1 – Организационная структура предприятия ООО «Зебра»

Клиентский отдел занимается работой с клиентами, такой как:

- сбор сведений о клиентах;
- прием заказов на изготовление рекламных баннеров, разработку рекламных сайтов, путем составления ТЗ, разработку рекламных кампаний по продвижению организаций и предприятий;
- подписание договоров о предоставлении услуг;
- консультация клиентов в сфере рекламы.

Коммерческий отдел занимается продвижением и развитием предприятия:

- определение долговременной стратегии коммерческой деятельности предприятия;
- изучение рынка и влияний на продажу продукции;
- контроль над реализацией продукции, материально-техническим обеспечением предприятия, финансовыми и экономическими показателями деятельности предприятия;
- продвижением и рекламой продуктов на рынке.

Бухгалтерия занимается подсчетом средств и ресурсов предприятия:

- бухгалтерский учет средств предприятия;
- бухгалтерский учет ресурсов предприятия;
- составление налоговой отчетности;
- подсчет выплат зарплат сотрудникам;
- банковские операции и операции с финансовыми документами;
- составление договора о предоставлении услуг.

Производственный отдел занимается выпуском продукции и проектированием:

- проектирование и верстка шаблона сайтов по ТЗ;
- проектирование и разработка движка сайта;
- печать рекламных баннеров;
- анализ рынка для разработки стратегий рекламирования продуктов;
- проведение работ по продвижению продуктов предприятий или же самих предприятий.

Отдел кадров занимается учетом персонала предприятия:

- привлечение новых кадров для работы на предприятии;
- составление таблиц рабочего времени;
- составление отчетов о сотрудниках для пенсионного фонда.

#### 1.1.2 Анализ внешнего и внутреннего документооборотов

Предприятие имеет внешние связи документооборота с клиентами, а так же с другими организациями, такими как: налоговая, банк, пенсионный фонд.

Клиент предоставляет личную информацию для организации, а так же формирует ТЗ, либо делает заказы на иные услуги, такие как печать баннеров. В свою очередь клиент получает на руки копию договора, счет на оплату, а так же технический рабочий проект или акт о выполненных работах.

Налоговая дает распоряжения предприятию о составлении для нее налоговой декларации за отчетный период, который она и получает.

Банк проводит обмен финансовыми документами с предприятием, такими как реквизиты, состояния счетов, статусы оплаты или переводов.

Пенсионный фонд дает распоряжения предприятию на составление списка сотрудников и сведений об их занятости передает страховые свидетельства сотрудников, предприятие отправляет отчеты о переводах пенсионному фонду, а так же сведения о сотрудниках.

DFD диаграмма внешнего документооборота предоставлена на рисунке А.1 в приложении А.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		12

Внутренний документооборот распределен между такими отделами как: клиентский отдел, коммерческий отдел, бухгалтерия, отдел кадров, отдел производства.

Клиентский отдел получает заказы, ТЗ и сведения о клиентах от клиентов, перенаправляет полученные ТЗ и заказы в производственный отдел, в то же время отправляет сведения о заказах для составления договора. Затем получает оформленные договора и счет на оплату от бухгалтерии, которые передает клиенту, а так же акт о выполненной работе от производственного отдела, который тоже передается клиенту по окончанию работ по заказу.

Производственный отдел составляет получает ТЗ и заказы от клиентского отдела. По окончанию работ отправляет технический рабочий проект клиенту, если это сайт, или составляет акт о выполненной работе и передает его клиентскому отделу. Так же получает рабочий план в котором содержатся сведения о необходимых поправках в планах производства.

Бухгалтерия ведет обмен финансовыми документами с банком, получает распоряжения от налоговой о составлении налоговой отчетности за указанный период и передает налоговые декларации. Так же получает сведения о заказах от клиентского отдела, составляет на их основе договор, и передает обратно счет на оплату и договора. Еще принимает таблицу рабочего времени от отдела кадров, которая необходима для вычисления зарплаты сотрудников и ее последующего начисления.

Отдел кадров ведет обмен с пенсионным фондом, принимая распоряжения на составление списка сотрудников и страховые свидетельства, в свою очередь отправляя сведения о сотрудниках и отчеты о переводах. Так же составляет и передает таблицу рабочего времени в бухгалтерию и коммерческий отдел.

Коммерческий отдел принимает отчеты о деятельности от всех остальных отделов, и дополнительно таблицу рабочего времени от отдела кадров. На основе всего этого отдаются распоряжения во все отделы, а так же составляется рабочий план для производственного отдела.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		13

DFD диаграмма внутреннего документооборота представлена на рисунке Б в приложении Б.

## 1.2 Анализ локальной вычислительной сети

Анализ локальной сети предприятия показал что, для передачи данных используется кабель категории UTP Category 5 с поддержкой интерфейса 100BASE-TX. Все компьютеры организации подключены к общей локальной сети посредством использования сетевых коммутаторов. Выход в глобальную сеть осуществляется через модем, маршрутизатор служит сетевым экраном сети.

Схема локальной вычислительной сети представлена на рисунке 2.

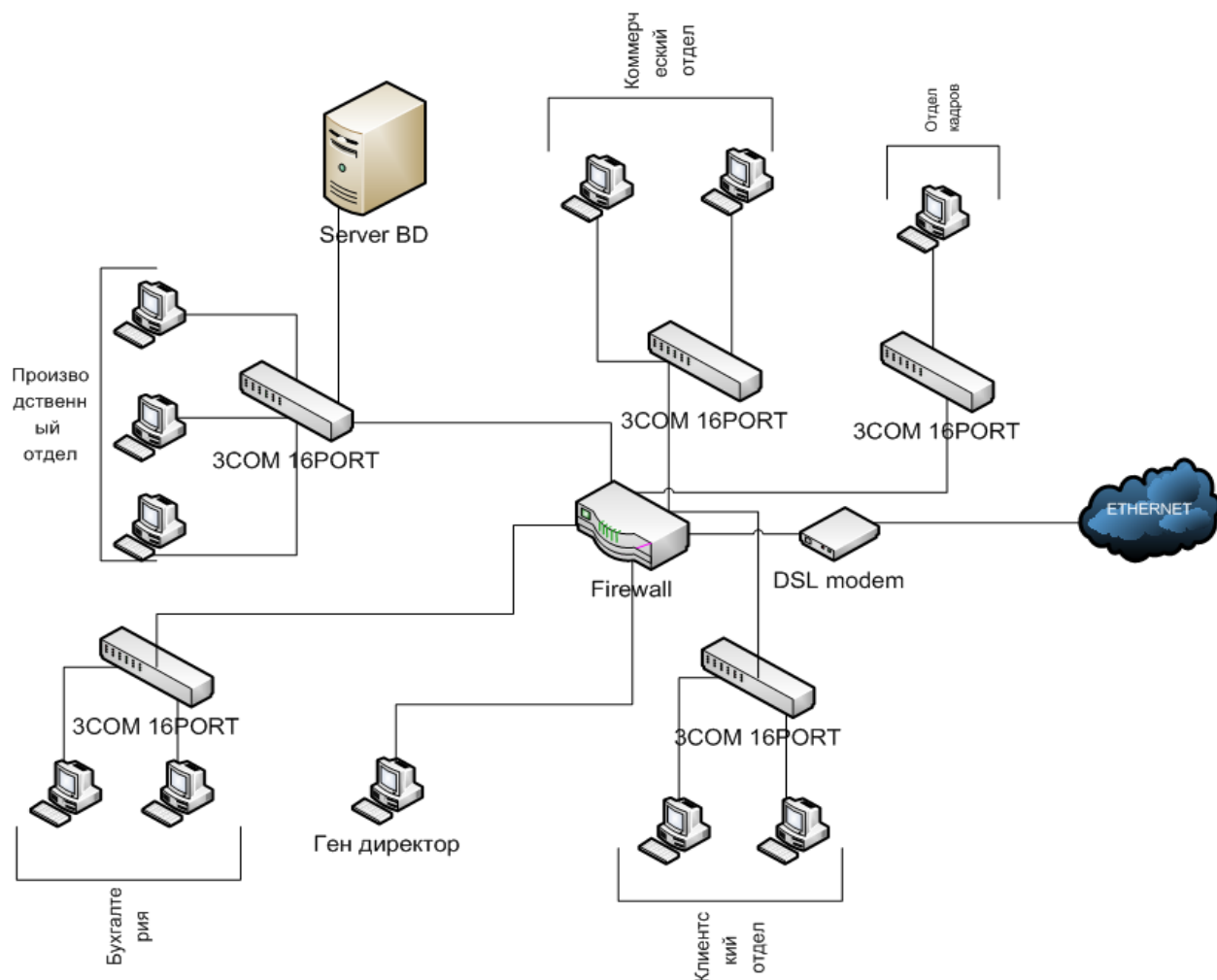


Рисунок 2 – Схема локальной вычислительной сети ООО «Зebra»

Сводная характеристика устройств подключенных к сети представлена в таблице 1.

					<i>ВКР.125047.09.03.02.ПЗ</i>	Лист
Изм	Лист	№ докум.	Подп.	Дата		14

Таблица 1 – Сетевые устройства

Наименование	Краткая характеристика	Кол-во
Рабочая станция	ОС W7.	11 шт
Файловый сервер	Intel core i5,4x3.5 ГГц, 4 Гб, 3 Тб, WinS 3	1 шт
Коммутатор	Нас,10/100/1000 м/с,16 port, MAC 8000	4 шт
ИПБ CyberPower UT450E	450Ва,8мин,4порт,65Гц	8 шт
ИПБ <u>DEXP Home 650VA</u>	650Ва,5 мин, 6 порт, 60Гц	3 шт
Кабель AirTone	5 е, дв.жил, алюм	150 м
Модем TP-LINK	4 port, ieee 802.3 300м/с, wps	1 шт

На рабочих станциях сети так же установлено ПО для работы в локальной сети и сети интернет.

Список сетевого ПО представлен в таблице 2.

Таблица 2 – Список сетевого ПО

Наименование	Тип	Назначение
Google Chrome	ПО	Обеспечивает просмотр веб-страниц,
The Dude	ПО	Сетевой монитор для управления лвс предприятия.
Сбербанк Бизнес Онлайн	ПО	Система дистанционного банковского обслуживания

### 1.3 Анализ аппаратного обеспечения

На предприятии аппаратное обеспечение это: компьютеры, сервер БД, печатный станок для печати баннеров.

Сервер БД – представляет центральный ресурс в сети для хранения и обеспечения совместного доступа к файлам пользователям сети. Содержит следующий набор аппаратных устройств:

- процессор Intel Core i5-4690;
- блок питания Aerocool VX-750;
- ИБП DEXP Home 650VA;

- оперативная память Hynix [H5TC4G83AFR-PBA] 8 Гб. 2 шт;
- жесткий диск Seagate SV35 3 терабайт;
- материнская память Gigabyte GA-85M.

В производственном отделе находится печатной станок и три компьютера со схожими характеристиками:

- материнская плата ASUS B85M-G;
- процессор Intel Core i5-4440 OEM;
- видеокарта ASUS GeForce GT 740;
- ИБП DEXP Home 650VA;
- блок питания Aerocool VX-750
- оперативная память Hynix [H5TC4G83AFR-PBA] 8 Гб. 2 шт;
- 1 Тб Жесткий диск WD Blue [WD10EZRZ].

Подобные высокие характеристики объясняются необходимостью работы с мультимедиа, компьютерной графикой и рендерингом.

В остальных отделах находятся компьютеры со стандартными для офисных компьютеров характеристиками:

- материнская плата Asrock H81M-VG4 R2.0;
- процессор Intel Celeron G1820 OEM;
- ИБП CyberPower UT450E;
- блок питания DEXP DTS-400;
- оперативная память Qumo [QUM3U-4G1600C11] 4 Гб. 1 шт;
- 500 Гб Жесткий диск Toshiba P300 [HDWD105UZSVA].

#### **1.4 Анализ программного обеспечения**

Все компьютеры используют ОС Windows 7 Professional x64. Так же везде установлен пакет Microsoft Office 2013 и антивирус Касперского Internet Security 2011.

В производственном отделе присутствует большой пакет программ для работы с графикой и мультимедиа, а так же для разработки и верстки сайтов, таких как:

- Blender 2.77;

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		16



- CorelDRAW X8;
- Sony Vegas pro 13.0;
- Adobe After Effects 13.7;
- Adobe – Audition CC 2014 7.0.0.118;
- Anime Studio Pro 9.2;
- Codelobster PHP Edition 5.8.1;
- NetBeans 8.1.

Бухгалтерия, коммерческий отдел, отдел кадров и клиентский отдел имеют одинаковый набор правовых программ и программ для учета:

- 1С: Предприятие 8.3;
- КонсультантПлюс: Версия ПРОФ 2012.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		17

## 2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

### 2.1 Обоснование необходимости создания системы

Чем больше становится страниц в сайте, тем больше HTML страниц становится в вашем доступе, и тем сложнее производить редактирование контента. Редактирование HTML страниц можно считать достаточно быстрым, если страниц насчитывается около двух-трех. При дальнейшем расширении сайта, редактирование контента станет очень долгим и рутинным. Нужно будет помнить расположение необходимых элементов, которые надо отредактировать, заходить в необходимую страницу и редактировать код непосредственно в HTML странице. Поэтому для сайтов с достаточно большим объемом информации используют CMS системы.

CMS системы являются неотъемлемой частью современных сайтов. В настоящее время операции редактирования контента производятся непосредственно с помощью CMS систем.

Данная будет обладать следующими возможностями:

- редактирование баз данных связанных непосредственно с сайтом;
- возможность редактирования отображения страниц сайта путем редактирования файла CSS стилей.

Частями CMS системы будут являться: само ядро CMS системы, шаблонный сайт, построенный по MVC технологии, для демонстрации возможностей CMS систем.

### 2.2 Обоснование выбора среды разработки

Для реализации возможностей CMS системы и технологии MVC был выбран язык PHP. Как альтернативы, существуют несколько других языков, таких как: perl, python, ruby.

Основной выбор был между PHP и Perl, так как они более схожи в функционале относительно web программирования. PHP выиграл у Perl, исходя из моих представлений, по нескольким пунктам:

					ВКР.125047.09.03.02.ПЗ	Лист
Изм	Лист	№ докум.	Подп.	Дата		18

- более простой в освоении язык, нежели Perl, так как до этого изучался язык C++, который по синтаксису родственен с PHP;
- PHP является более популярным языком для веб-программирования;
- PHP работает с базами данных быстрее, чем Perl, что является необходимым показателем для функционирования сайта;
- PHP изначально является языком для веб-программирования, в отличие от Perl, который является языком программирования общего назначения.

Для разработки был использован программный пакет Denwer, который включает в себя:

- веб-сервер Apache с поддержкой SSI, SSL, mod\_rewrite, mod\_php;
- интерпретатор PHP с поддержкой GD, MySQL, SQLite;
- СУБД MySQL с поддержкой транзакций (mysqld-max);
- система управления виртуальными хостами, основанная на шаблонах;
- система управления запуском и завершением;
- панель phpMyAdmin для администрирования СУБД;
- ядро интерпретатора Perl без стандартных библиотек (поставляются отдельно);
- эмулятор sendmail и сервера SMTP с поддержкой работы совместно с PHP, Perl, Parser и др.

Denwer – набор дистрибутивов (локальный сервер WAMP) и программная оболочка, предназначенные для создания и отладки веб-приложений и прочего динамического содержимого интернет-страниц на локальном ПК без необходимости подключения к сети Интернет, под управлением ОС Windows.

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д.

MySQL – свободная реляционная система управления базами данных. MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты.

### **2.3 Характеристика функциональных подсистем проектируемой системы**

Для выполнения своих задач, CMS система использует следующие подсистемы:

- подсистема администрирования данных;
- подсистема запроса модулей;
- подсистема запроса данных;
- подсистема вывода модулей.

IDEF 0 диаграмма CMS системы представлена на рисунке В.1 в приложении В.

Декомпозиция IDEF0 диаграммы CMS системы представлена на рисунке В.2 в приложении В.

Подсистема администрирования данных выполняет функции редактирования данных. Сама подсистема разбита на модули и имеет визуальное представление. Поэтому, работоспособность системы напрямую зависит от работоспособности других подсистем.

Подсистема должна обладать следующими возможностями:

- удобный вывод информации о данных содержащихся в БД;
- удобное редактирование и дополнение БД;
- исключение ошибок записи и соблюдение целостности данных;
- взаимодействие с файлами конфигурации CMS системы;
- соблюдение условий конфиденциальности, в плане авторизации перед входом в панель администрации.

Подсистема запроса модулей выполняет функции по составлению визуального представления сайта. Подсистема имеет модульную структуру и не имеет визуального представления.

Подсистема должна обладать следующими возможностями:

- создание ссылочных запросов с помощью языка PHP;
- обработка данных от пользователя для составления запросов;
- предопределения правил вывода страниц;
- компоновка данных модулей с их визуальным представлением;
- формирование страниц.

Подсистема вывода модулей выполняет функции формирования визуального представления модулей. Подсистема имеет модульную структуру и имеет визуальное представление в виде блоков, которые формируют сайт.

Подсистема должна обладать следующими возможностями:

- формирование визуального представления модуля с помощью языков разметки HTML;
- представление логической части страниц с помощью языков PHP, JavaScript;
- возможность форматирования вывода блоков модуля с помощью формального языка CSS;
- поддержка дополнительных функций с помощью интерпретаторов других серверных языков.

Подсистема запросов данных выполняет функцию выборки данных для модулей. Подсистема имеет модульную структуру и не имеет визуального представления.

Подсистема должна обладать следующими возможностями:

- создания подключения к БД необходимым для выборки данных;
- формирование запросов необходимых данных из БД;
- формирование списка полученных из БД данных;
- проверка ошибок подключения и обработки данных.

## **2.4 Характеристика обеспечивающих подсистем проектируемой системы**

### **2.4.1 Подсистема организационного обеспечения**

Подсистема «Организационное обеспечение» является одной из важнейших подсистем, от которой зависит успешная реализация целей и функций системы. В ее составе можно выделить три группы компонентов.

Совокупность средств, необходимых для эффективного проектирования и функционирования системы.

Техническая документация, получаемая в процессе обследования, проектирования и внедрения системы: техническое задание на разработку системы и первичные формы входных документов.

Группа «Пользователи», которые будут иметь доступ к базе данных, имеет одну категорию:

– специалист, осуществляющий обслуживание и настройку системы, обеспечивающий ее работоспособность. Квалификация – администратор системы, программист.

### **2.4.2 Лингвистическое обеспечение**

Требования к лингвистическому обеспечению предполагают использование единого логического интерфейса для пользователей. Пользовательский интерфейс должен обеспечивать единство представления данных с учетом ограничений, налагаемых операционными средами, осуществлять взаимодействие с пользователями на русском языке, а также предоставлять различного вида отчеты на русском языке. Должны быть предусмотрены простые, легкие и удобные в использовании, методы выбора операций для ввода данных, формирования отчетов, выполнения запросов.

## **2.5 Информационное обеспечение**

Информационным обеспечением для CMS-системы является набором таблиц данных реализованных в среде СУБД MySQL.

Набор таблиц может меняться в зависимости от необходимости заполнения страниц контентом или создания дополнительных модулей сайта.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		22

### 2.5.1 Инфологическое проектирование

Для работы CMS-системы выделены таблицы данных представленные в таблице 3.

Таблица 3 – Таблицы данных

Название таблицы	Описание таблицы
category	Содержит информацию о категориях, на которые разбиты статьи представленные на сайте
menu	Содержит информацию о разделах сайта ссылки, на которые представлены в верхней части сайта
topics	Содержит информацию о статьях сайта
users	Содержит информацию о пользователях для авторизации в панель администрирования

Атрибуты таблицы данных category представлены в таблице 4.

Таблица 4 – Атрибуты таблицы «category»

Название атрибута	Описание атрибута	Диапазон значений	Единица измерения	Пример
id_category	Уникальный номер категории	>0	-	1
name_category	Наименование категории	Текст	-	Category1

Атрибуты таблицы данных «menu» представлены в таблице 5.

Таблица 5 – Атрибуты таблицы «menu»

Название атрибута	Описание атрибута	Диапазон значений	Единица измерения	Пример
id_menu	Уникальный номер пункта меню	>0	-	1
name_menu	Наименование пункта меню	Текст	-	Контакты
text_menu	Текст для вывода относящийся к пункту меню	Текст	-	Краткая контактная информация

Атрибуты таблицы данных «topics» представлены в таблице 6.

Таблица 6 – Атрибуты таблицы «topics»

Название атрибута	Описание атрибута	Диапазон значений	Единица измерения	Пример
id	Уникальный номер статьи	>0	-	1
title	Наименование статьи	Текст	-	Статья1
description	Краткое описание статьи	Текст	-	Кратко-кратко
text	Полный текст статьи	Текст	-	Полно-полно
date	Дата добавления статьи	Дата	-	24-05-2016
cat	Номер категории в которую входит статья	>0	-	1

Атрибуты таблицы данных «users» представлены в таблице 7.

Таблица 7 – Атрибуты таблицы «users»

Название атрибута	Описание атрибута	Диапазон значений	Единица измерения	Пример
id	Уникальный номер пользователя	>0	-	1
login	Логин пользователя	Текст	-	admin
password	Пароль пользователя	Текст	-	admin1234

### 2.5.2 Логическое проектирование

Таблицы «category» и «topics» имеют связь один ко многим, порожденной сущностью в данной связи является таблица «topics».

Внешним ключом в таблице является атрибут «cat».

### 2.5.3 Физическое проектирование

Физическое представление таблицы «category» представлено в таблице 8.



Таблица 8 – Физическое представление таблицы «category»

Название поля	Тип данных	Длина	Индексация	Допустимость NULL	Ограничения
<u>id_category</u>	tinyint	3	Да	Нет	-
name_category	varchar	100	Нет	Нет	-

Физическое представление таблицы «menu» представлено в таблице 9.

Таблица 9 – Физическое представление таблицы «menu»

Название поля	Тип данных	Длина	Индексация	Допустимость NULL	Ограничения
<u>id_menu</u>	tinyint	3	Да	Нет	-
name_menu	varchar	100	Нет	Нет	-
text_menu	text	-	Нет	Нет	-

Физическое представление таблицы «topics» представлено в таблице 10.

Таблица 10 – Физическое представление таблицы «topics»

Название поля	Тип данных	Длина	Индексация	Допустимость NULL	Ограничения
<u>id</u>	int	10	Да	Нет	-
title	varchar	100	Нет	Нет	-
description	text	-	Нет	Нет	-
text	text	-	Нет	Нет	-
date	date	-	Нет	Нет	-
cat	tinyint	3	Нет	Нет	-

Физическое представление таблицы «users» представлено в таблице 11.

Таблица 11 – Физическое представление таблицы «users»

Название поля	Тип данных	Длина	Индексация	Допустимость NULL	Ограничения
<u>id</u>	int	11	Да	Нет	-
login	text	-	Нет	Нет	-
password	text	-	Нет	Нет	-

## 2.6 Программное обеспечение

### 2.6.1 Структура программного обеспечения

Сайт и CMS система работают по шаблону проектирования MVC.

MVC (model-view-controller) – это шаблон проектирования приложений, при котором управляющая логика таким образом, что модифицирование одного из них дает минимальное влияние на остальные.

MVC шаблон разделяет логику программы от ее визуального представления, и делит всю систему на три компонента:

- модель – предоставляет данные для вывода;
- представление – отвечает за отображение и представление информации в нужном виде;
- поведение – принимает данные и запросы от пользователя, интерпретирует их в запросы для модели и представления.

Применительно к PHP, MVC модель выглядит следующим образом.

В модели формируются подключения к БД, и выборки данных для представления. С помощью разделения на отдельные классы, можно делать выборку из различных БД и таблиц, группировать, сортировать, делать выборку данных.

Контроллер делится на несколько модулей, в зависимости от количества функций, и эти модули контролируют работу приложения, в зависимости от запросов пользователя.

Представление так же делится на несколько модулей в зависимости от количества различных частей приложения. Отображает внешний вид модулей приложения в зависимости от запросов пользователя и данных полученных из модели.

Преимуществом данного шаблона является более удобное модифицирование приложения, его логики и визуального представления, т.к. они отделены друг от друга и имеют пересечение лишь посредством функций.

Структура шаблона проектирования MVC представлена на рисунке 3.

### 2.6.2 Проектирование программного обеспечения

CMS-системы имеют несколько классификаций по способу работы.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		26

CMS-система генерирующая страницы по запросу это система которая работает на основе связки «Модуль редактирования → База данных → Модуль представления».

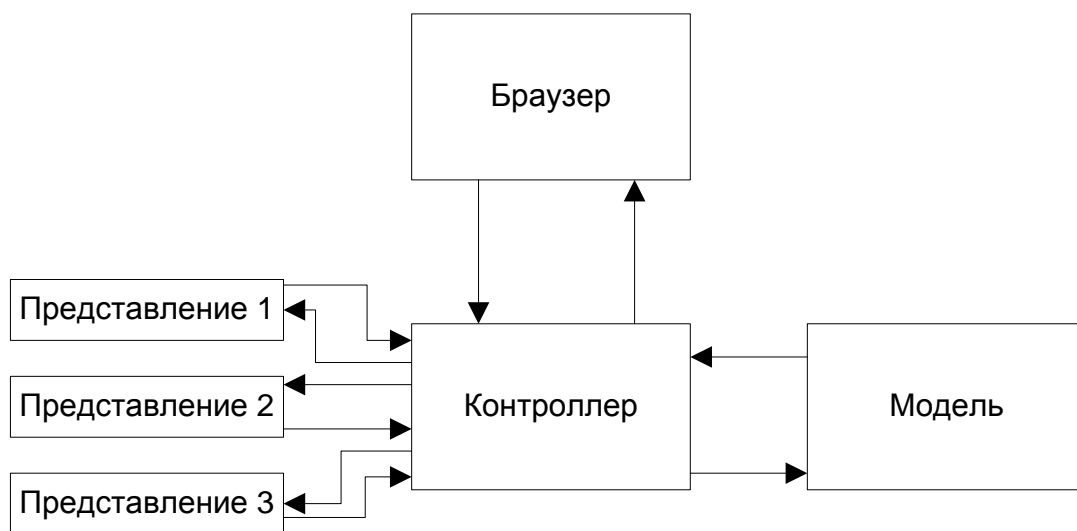


Рисунок 3 – Структура шаблона MVC

Модуль представления генерирует страницу с содержанием при запросе на него, на основе информации из базы данных. Информация в базе данных изменяется с помощью модуля редактирования. Страницы заново создаются сервером при каждом запросе, что в свою очередь создаёт дополнительную нагрузку на системные ресурсы. Нагрузка может быть многократно снижена при использовании средств кэширования, которые имеются в современных веб-серверах;

CMS-система генерирующая страницы при редактировании. Система этого типа суть программы для редактирования страниц, которые при внесении изменений в содержание сайта создают набор статических страниц. При таком способе в жертву приносится интерактивность между посетителем и содержимым сайта;

CMS-система смешанного типа это системы которые сочетают в себе преимущества первых двух. Может быть реализована путём кэширования – модуль представления генерирует страницу один раз, в дальнейшем она в несколько раз быстрее подгружается из кэша. Кэш может обновляться как автоматически, по истечении некоторого срока времени или при внесении измене-

ний в определённые разделы сайта, так и вручную по команде администратора. Другой подход – сохранение определённых информационных блоков на этапе редактирования сайта и сборка страницы из этих блоков при запросе соответствующей страницы пользователем.

Так же, их можно разделить на веб-приложения и обычные приложения.

Разрабатываемая CMS система будет использовать генерацию страниц по запросу и будет представлена веб-приложением. Данный выбор объясняется тем, что сайт будет представлен в виде двух динамически изменяющихся страниц, собирающихся по запросу. Поэтому использование кэширования страниц для вывода не будет иметь смысла.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		28

## 3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 3.1 Описание программы

CMS системы это информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления контентом.

Их существование можно объяснить тем, что с расширением сайта и появлением не статичного наполнения сайта, для редактирования этого наполнения будет затрачено много времени и усилий, так как все данные неорганизованны и хранятся в десятках страниц.

Для статичных страниц или для довольно небольшого объема данных которые нужно редактировать, реализация CMS системы будет нерациональной.

CMS системы позволяют структурировать вывод данных необходимых для редактирования. Для работы с данными теперь используются не файлы, а таблицы реляционных СУБД, что упрощает их вывод, а так же позволяет делать группировку данных.

CMS так же является, по сути, «движком» сайта, т.е. набором функций которые формируют структуру сайта по своим правилам. В таком случае сайты переходят от обычных HTML файлов к более сложным и динамически генерируемым модульным страницам. Зачастую, для этого используются языки программирования PHP и Perl.

Принцип работы всех CMS основан на разделении контента (содержания) и дизайна (оформления) сайта. Обычно дизайн сайта меняется редко, тогда, как изменения контента могут происходить не только каждый день, но и даже каждый час. Поэтому в своей работе CMS используют так называемые шаблоны – специальные «пустые» заготовки страниц, в которых дизайн сайта уже прописан и осталось лишь наполнить их информацией. Пользователю достаточно воспользоваться специальным WYSIWYG-редактором. Этот редактор по внешнему виду очень похож на привычные текстовые редакторы офисных приложе-

					ВКР.125047.09.03.02.ПЗ	Лист
Изм	Лист	№ докум.	Подп.	Дата		29

ний, поэтому пользователю не составляет особого труда освоить его. Данные редакторы написаны на языке JavaScript.

Приложение представляет собой объектно-ориентированное приложение, диаграмма классов для которого представлена на рисунке Г приложения Г.

### 3.1.1 Логическая структура программы

Логика программы построена по MVC шаблону. Логическая структура представлена на рисунке 4.

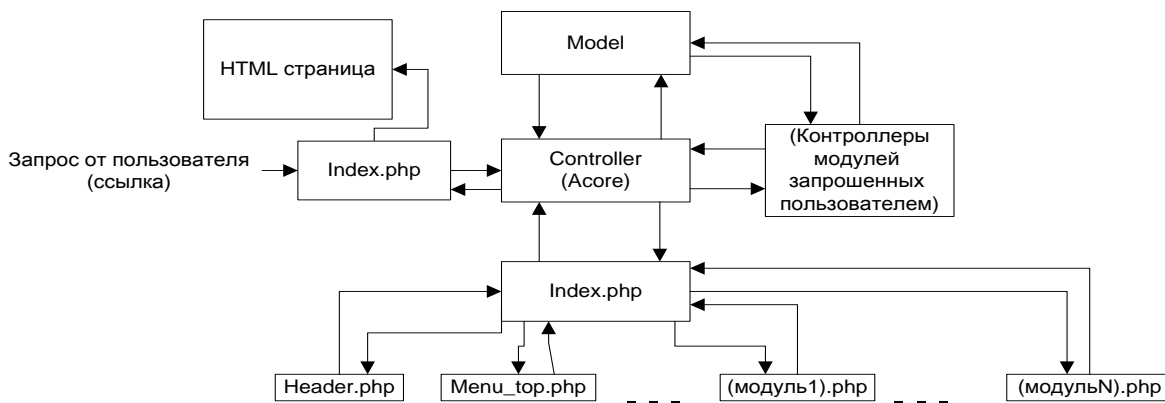


Рисунок 4 – Логическая структура программы

Логика программы представлена на примере отображения страницы сайта.

Существует контроллер index в котором заложена логика обработки ссылок от пользователя и загрузки контроллеров для АCore.

Первым делом, в браузере формируется запрос пользователя в виде ссылки на страницу. Главным указателем в ссылке служит надпись option. Данный запрос обрабатывается index.php и передает переменную для подключения нужных модулей. АCore, в свою очередь, опрашивает другие стандартные контроллеры и дополнительные контроллеры, которые должны быть задействованы в зависимости от полученной переменной. Список контроллеров для вывода основных модулей страницы описан в модуле подсистемы представления index.php. Так же, там описан вывод дополнительных модулей. Все контроллеры, включая АCore, формируют запросы к Model, где происходит выборка нужных данных и их передача обратно контроллерам. Стоит отметить, что кон-

троллеры необязательно могут запрашивать данные из модуля, контроллеры могут реализовывать функции не связанные с БД.

В зависимости от набора контроллеров, полученных ими данных или выполненных ими функций, с помощью списка модулей в `index.php (tpl)` и переданных данных главному контроллеру `ACore`, а затем `index.php` формируется страница сайта.

Некоторые контроллеры способны сами создавать запросы для главного контроллера с помощью переадресации ссылки в адресной строке, т.е. меняя адрес они, тем самым, запускают формирование нового набора модулей для страницы.

### 3.1.2 Описание модулей, функций, обработчиков событий

Контроллер `Index.php` описывает логику подключения модулей. Представляет собой неклассовый PHP код, к которому обращается веб-приложение.

Входными данными для него являются ссылки.

Выходными данными для него является страница сайта.

В начале контроллер объявляет открытие сессионных переменных, затем происходит подключение дополнительных конфигурационных файлов. Потом идет проверка правильности написания запроса в адресной строке и попытка вывода модулей на страницу, при этом происходит вызов функции `__autoload` через вызов `class_exists`.

Функция `__autoload` подключает контроллеры и модели к приложению.

Входные данные для функции это наименование контроллера.

Алгоритм функции `__autoload` представлен на рисунке 5.

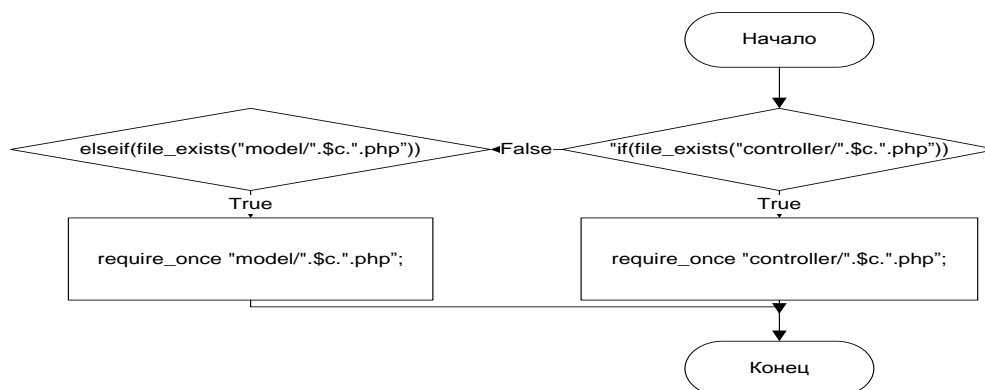


Рисунок 5 – Алгоритм функции `__autoload`

На рисунке 6 изображен алгоритм работы контроллера index.php

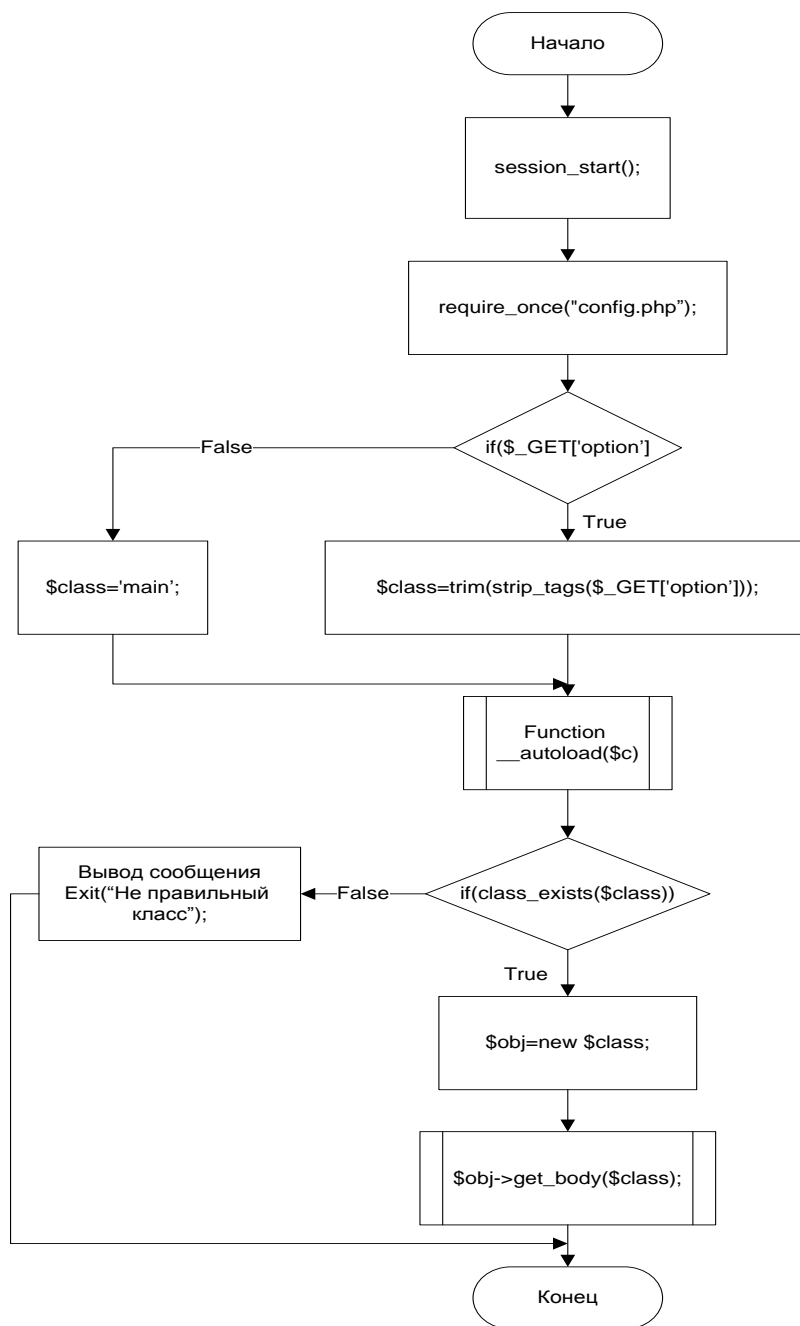


Рисунок 6 – Алгоритм работы «index.php»

Контроллер ACore.php является одним из основных контроллеров формирующих вывод страницы сайта.

Входными данными для него являются переменная для подключения классов \$tpl.

Выходными данными являются набор представлений формирующих страницу сайта.

Алгоритм контроллера показан на рисунке 7.



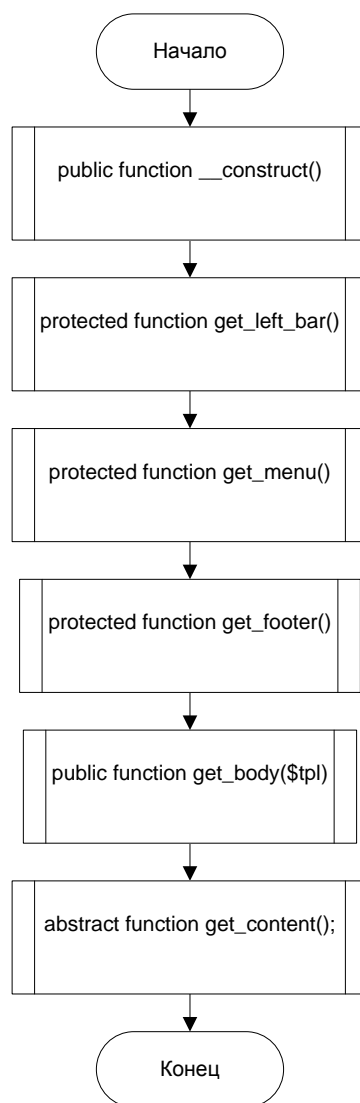


Рисунок 7 – Алгоритм работы контроллера ACore.php

В контроллере происходит поочередный вызов функций необходимых для опроса других контроллеров и составления списка данных.

Функция `__construct` подключает к переменной `m` класс `model`.

Функции `get_left_bar`, `get_menu`, `get_footer`, `get_body`, производят запрос данных из `model`, с помощью обращения к `model`.

Выходными данными являются набор данных из таблицы БД.

Функция `get_body` выполняет две функции:

- принимает результаты выполнения обработчиков отправки формы `obr`;
- производит запросы к своим внутренним функциям и внешним контроллерам.

Входные данные для функции это наименование контроллера и данные из таблиц БД.

Выходные данные это набор представлений.

Алгоритм функции `get_body` представлен на рисунке 8.

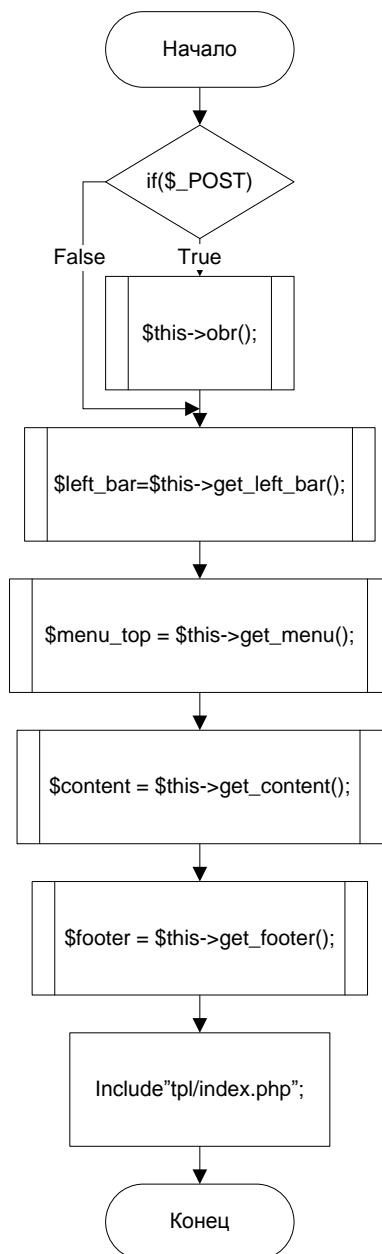


Рисунок 8 – Алгоритм функции `get_body`

Контроллер `ACore_admin` является одним из основных контроллеров формирующих вывод страницы сайта. Данный контроллер отвечает за выведение страницы редактирования контента.

Входными данными для него являются сессионная переменная «user» и переменная для подключения классов `$tpl`.

Выходными данными являются наборы модулей формирующих страницу редактирования контента.

Алгоритм контроллера показан на рисунке 9.

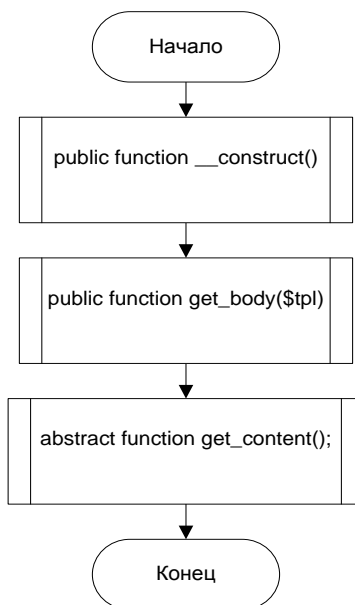


Рисунок 9 – Алгоритм контроллера Acore\_Admin

В контроллере происходит вызов функций `__construct` и `get_body`.

Функция `__construct` выполняет следующие функции:

- выполняет проверку сессионной переменной 'user' которая служит идентификатором авторизации пользователя;
- производит подключение к переменной `m` класса `model`.

Входной переменной служит переменная 'user'.

Алгоритм функции `__construct` представлен на рисунке 10.

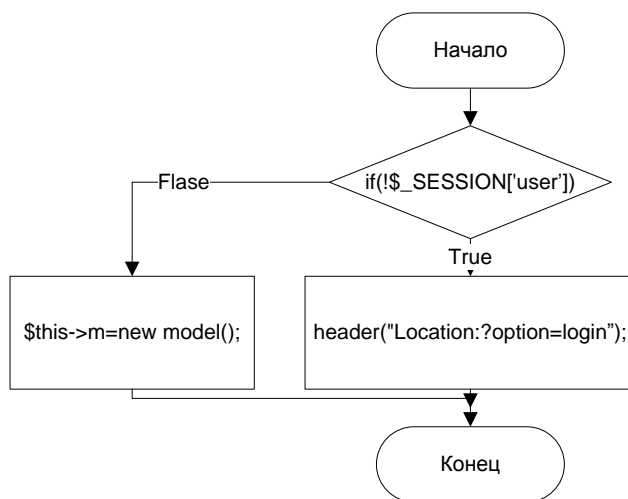


Рисунок 10 – Алгоритм функции `__construct`

Функция `get_body` выполняет три функции:

- принимает результаты выполнения обработчиков отправки формы `obr` ;
- принимает результаты выполнения обработчика `obr` при получении в суперглобальной переменной `$_GET` значения `'del'`;
- производит запросы к внешним контроллерам.

Входными переменными являются наименование контроллера, данные из таблиц БД и значение переменной `$_GET`;

Выходными данные это набор представлений.

Алгоритм функции `get_body` представлен на рисунке 11.

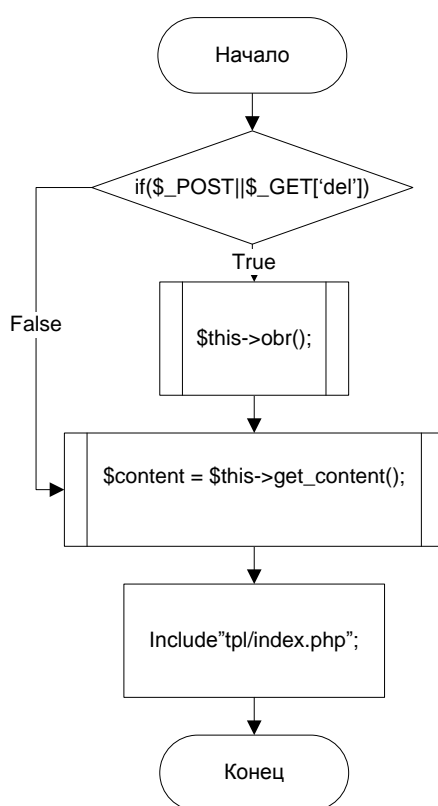


Рисунок 11 – Алгоритм функции `get_body`

Модуль `model.php` является моделью и выполняет функции выборки данных для контроллеров.

Входными данными являются различные переменные для выборки данных и данные для подключения к БД.

Выходными данными являются результаты выборки.

Алгоритм модели `model` представлен на рисунке 12.

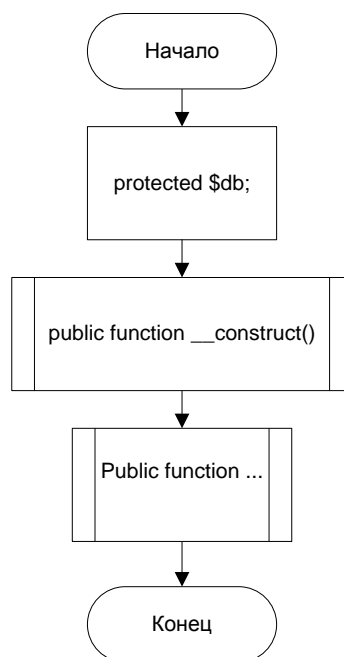


Рисунок 12 – Алгоритм модели model

Сначала объявляется переменная \$db для записи в нее сессии подключения к БД, затем выполняется функция \_\_construct и та функция, которая была запрошена контроллером.

Функция \_\_construct выполняет функцию инициализации подключения и проверки его на наличие ошибок.

Входными переменными являются данные для подключения к БД.

Выходными данными является подключение к БД.

Алгоритм функции \_\_construct представлен на рисунке 13.

Имеют одно предназначение – это формирование запроса и выборка данных.

Сначала происходит формирование запроса в переменной \$query, затем запись результата выборки в переменную \$result, затем происходит построчная запись данных из переменной \$result в переменную \$row. Может происходить как выборка одной строки данных, так и массива данных, в зависимости от этого может использоваться цикл for.

Входными данными иногда могут являться переменные для выборки.

Выходные данные это данные из таблиц БД.

Алгоритм функции get\_left\_bar представлен на рисунке 14.

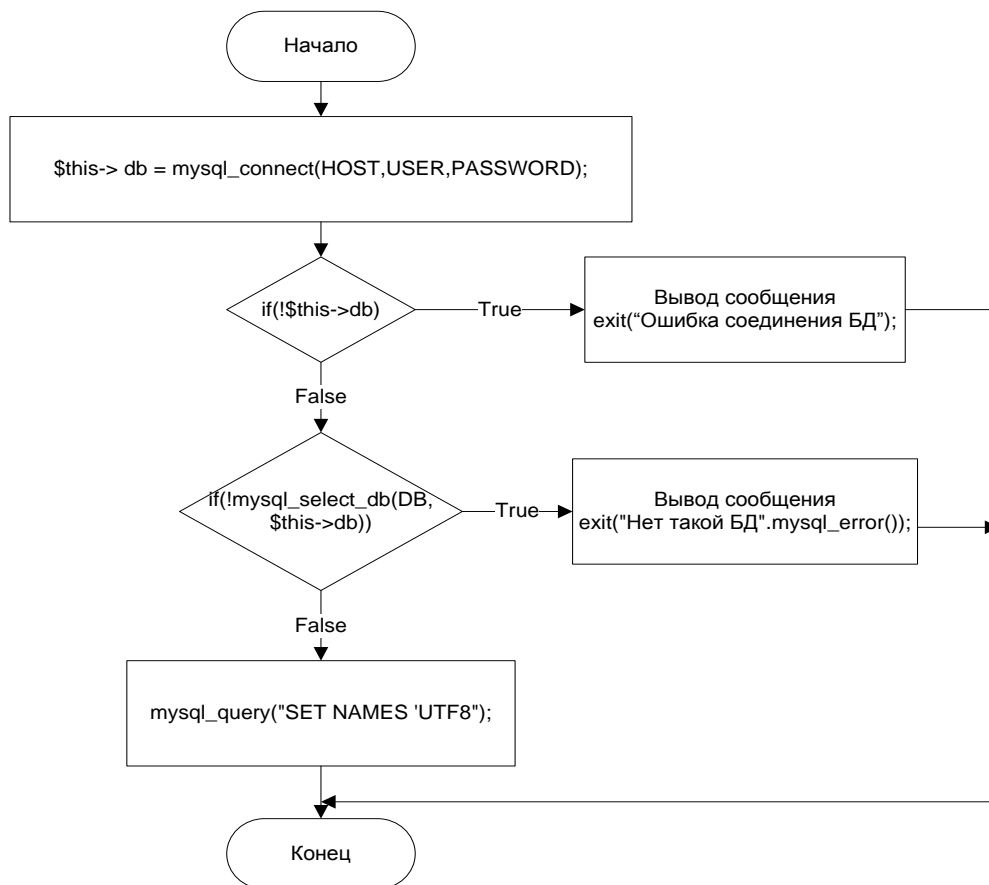


Рисунок 13 – Алгоритм функции \_\_construct

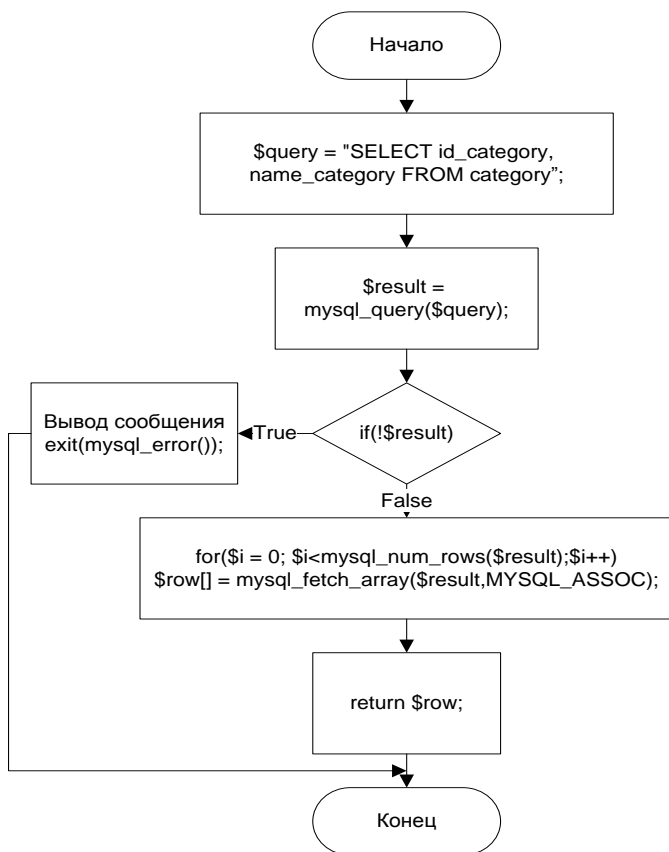


Рисунок 14 – Алгоритм функции get\_left\_bar

Алгоритм функции menu\_array представлен на рисунке 15.

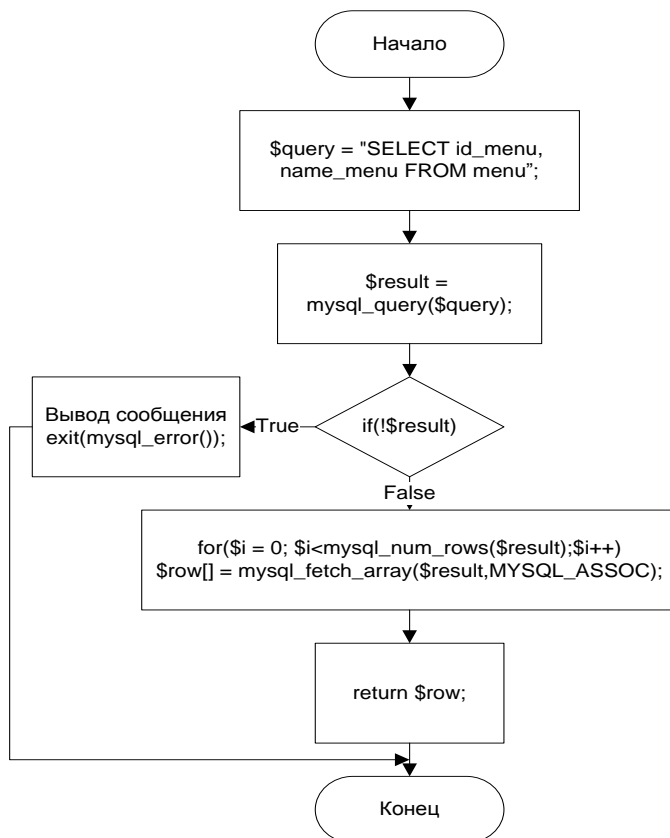


Рисунок 15 – Алгоритм функции menu\_array

Алгоритм функции get\_main\_content представлен на рисунке 16.

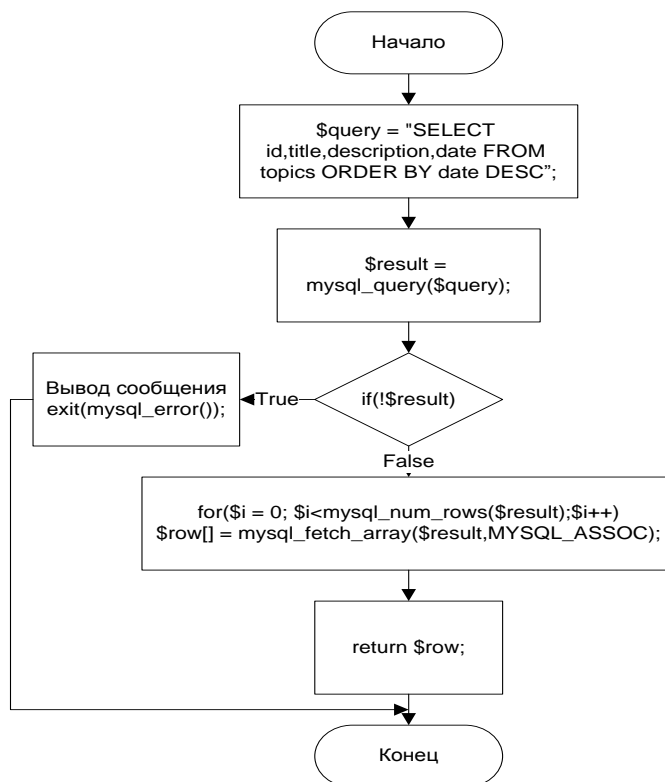


Рисунок 16 – Алгоритм функции get\_main\_content

Алгоритм функции get\_cat представлен на рисунке 17.

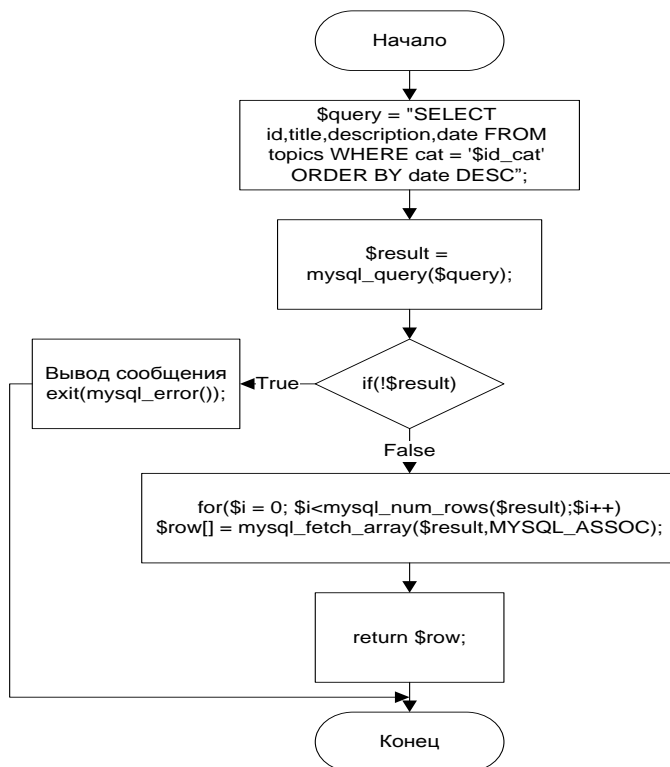


Рисунок 17 – Алгоритм функции get\_cat

Алгоритм функции get\_menu представлен на рисунке 18.

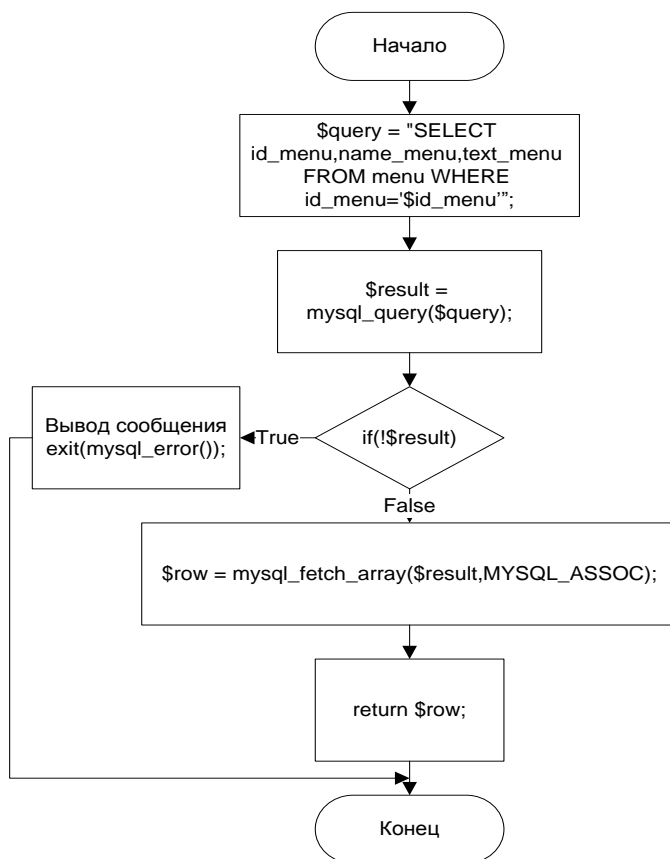


Рисунок 18 – Алгоритм функции get\_menu

Изм.	Лист	№ докум.	Подп.	Дата

ВКР.125047.09.03.02.ПЗ

Лист

40



Алгоритм функции `get_topic` представлен на рисунке 19.

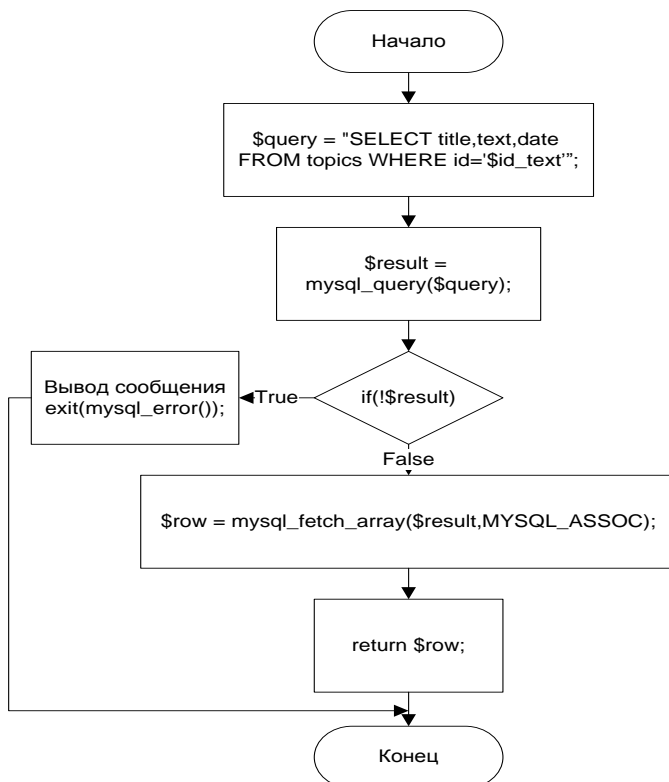


Рисунок 19 – Алгоритм функции `get_topic`

Алгоритм функции `get_admin_menu` представлен на рисунке 20.

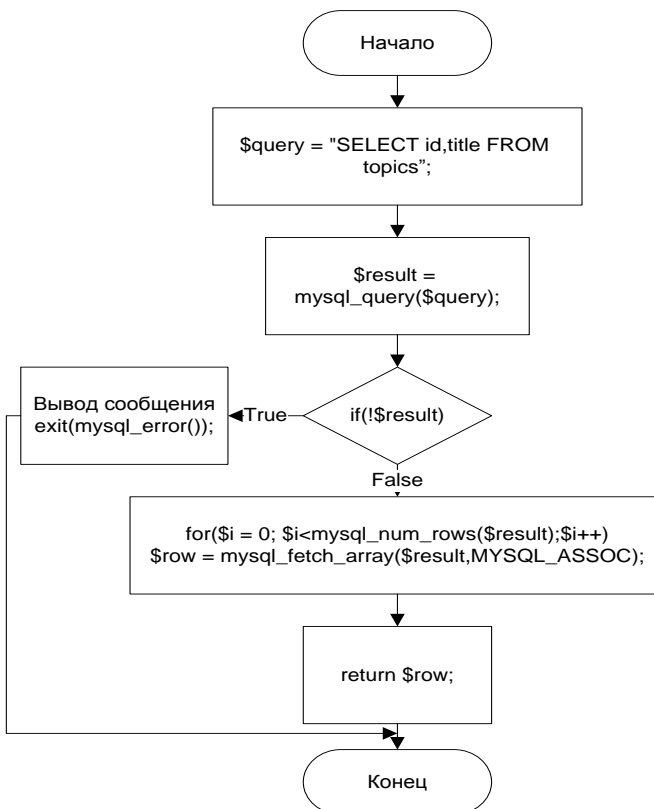


Рисунок 20 – Алгоритм функции `get_admin_menu`

Изм.	Лист	№ докум.	Подп.	Дата

Алгоритм функции get\_edit\_menu представлен на рисунке 21.

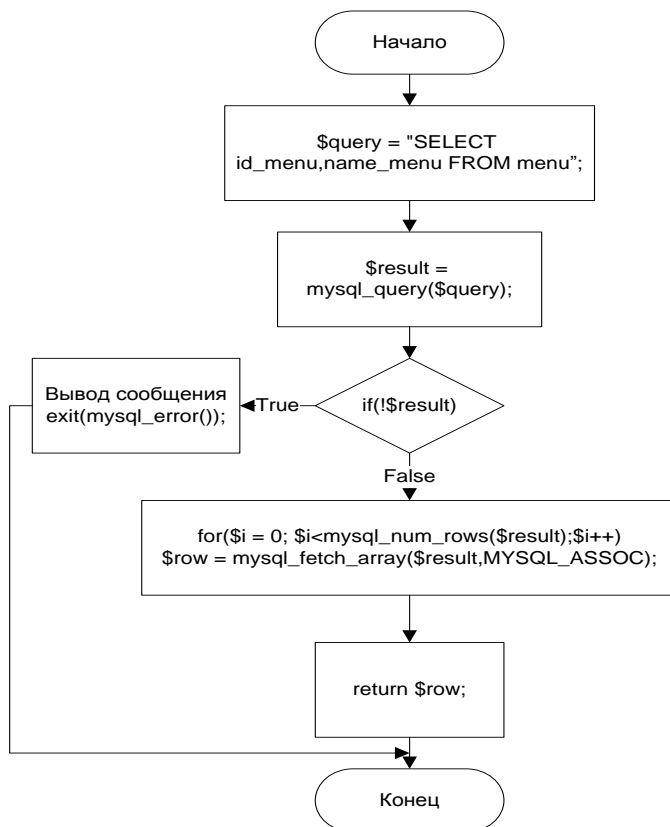


Рисунок 21 – Алгоритм функции get\_edit\_menu

Алгоритм функции get\_text\_menu представлен на рисунке 22.

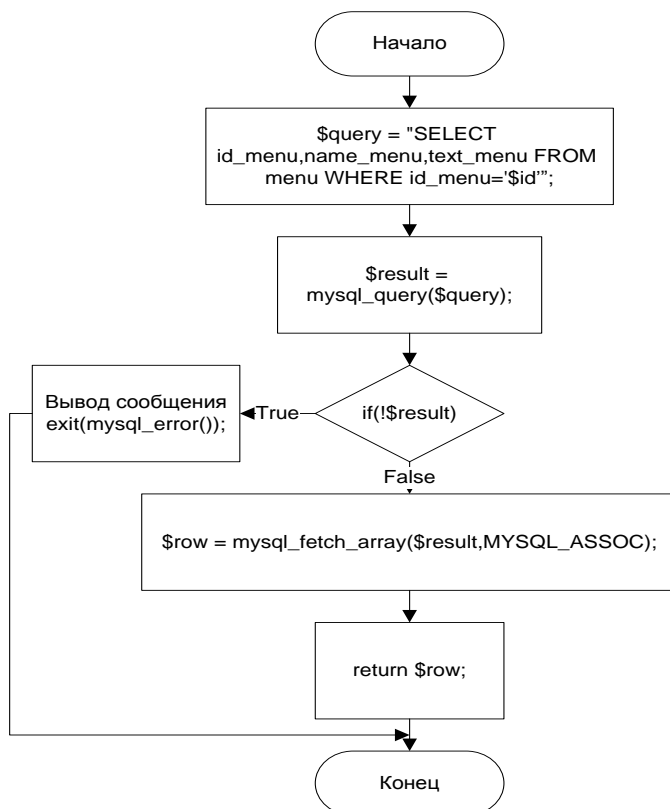


Рисунок 22 – Алгоритм функции get\_text\_menu

Алгоритм функции `get_categories` представлен на рисунке 23.

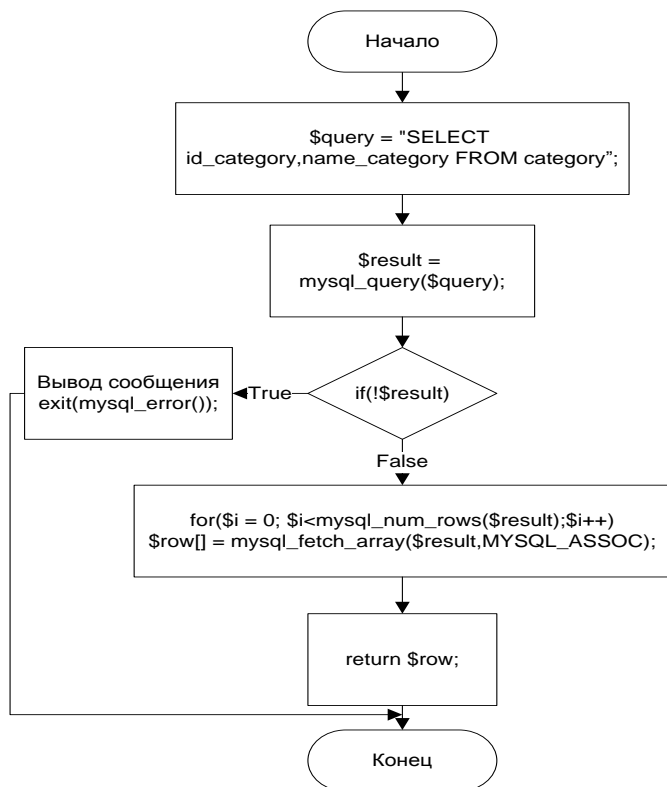


Рисунок 23 – Алгоритм функции `get_categories`

Алгоритм функции `get_text_topic` представлен на рисунке 24.

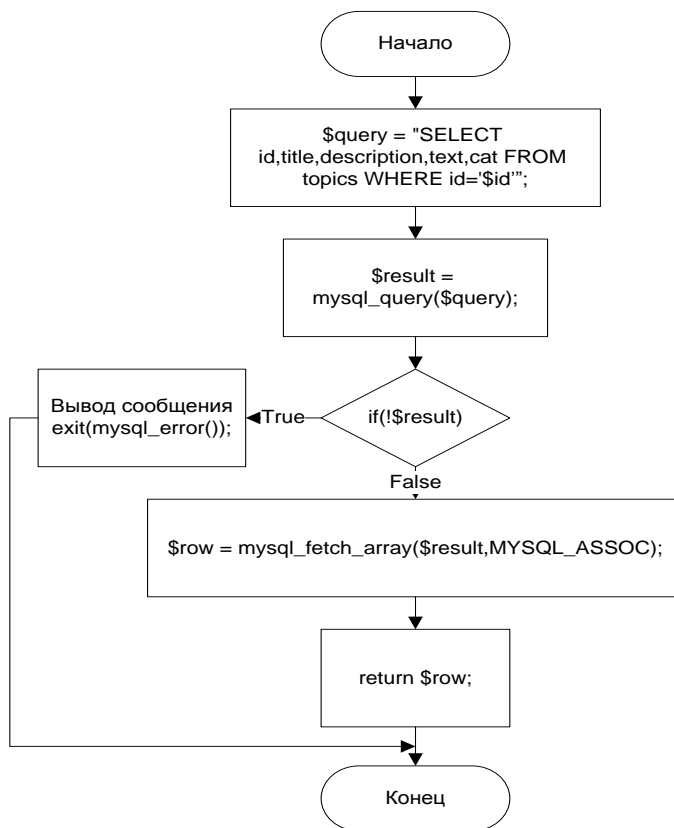


Рисунок 24 – Алгоритм функции `get_text_topic`

Алгоритм функции `get_categories2` представлен на рисунке 25.

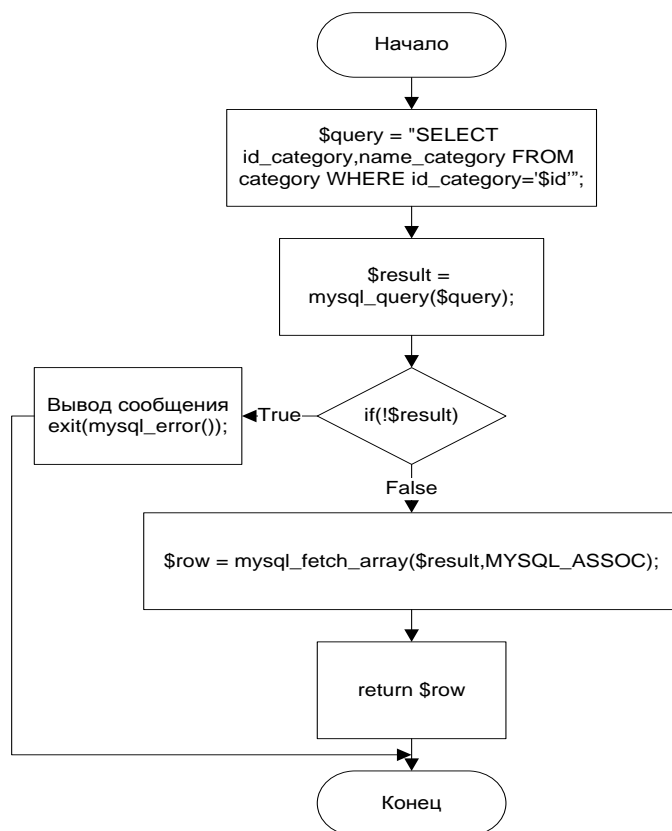


Рисунок 25 – Алгоритм функции `get_categories2`

Контроллер `add_category` является дочерним классом `ACore_admin` и выполняет функцию добавления категории для статей в БД, для последующего отображения их на странице.

Входными данными для него являются данные записанные в форму.

Выходными данными является запрос в БД для создания новой строки в таблице `category`.

Алгоритм контроллера `add_category` представлен на рисунке 26.

Сначала инициализируется функция `get_content`, которая является обработчиком контроллера и его основной функцией, затем, при отправке формы, инициализируется функция `obr`.

Функция `get_content` в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция `obr` получает данные от переменной `$_POST` и записывает данные в таблицу БД.

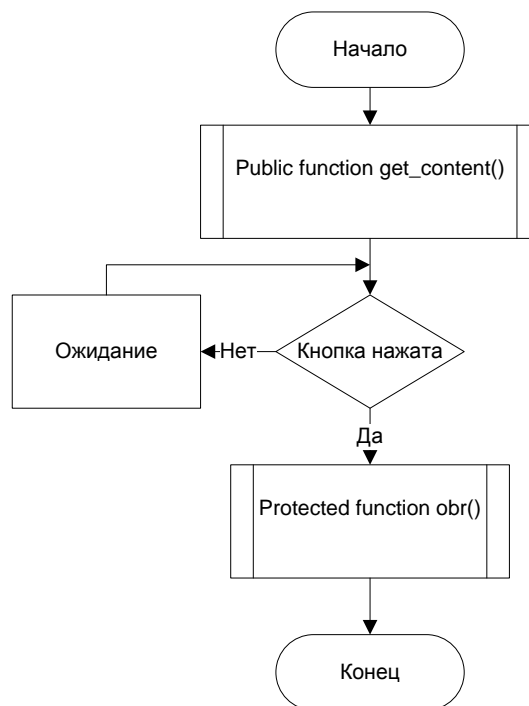


Рисунок 26 – Алгоритм контроллера add\_category

Входные данные это переменная \$\_POST.

Выходные данные это сессионная переменная 'res' и запрос на запись данных.

Алгоритм функции obr представлен на рисунке 27.

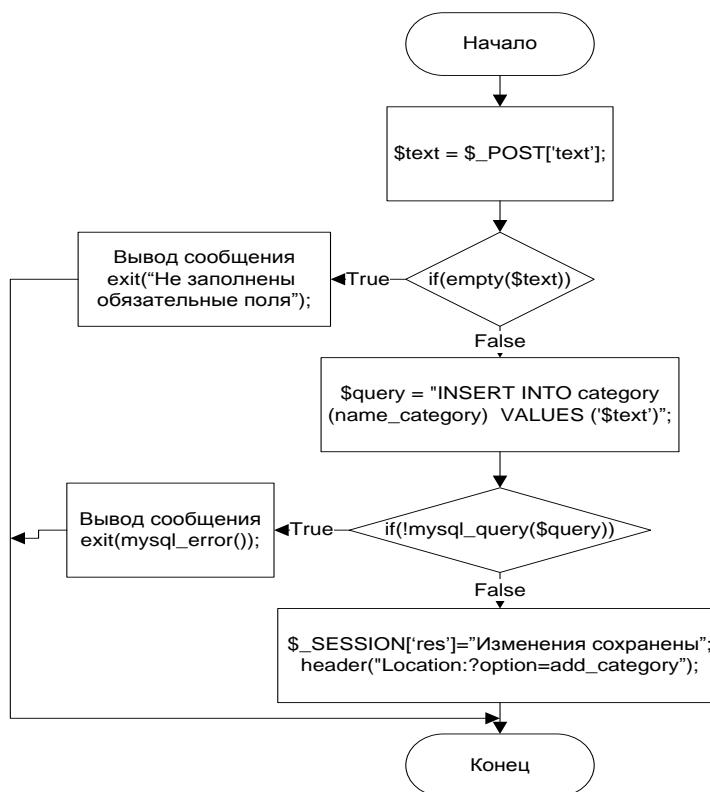


Рисунок 27 – Алгоритм функции obr

Изм.	Лист	№ докум.	Подп.	Дата

У контроллера `add_category` имеется представление `add_category`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении содержится форма в которой содержится поле для ввода строки. Отправка формы происходит по нажатию кнопки.

Контроллер `add_menu` является дочерним классом `ACore_admin` и выполняет функцию добавления пункта меню в БД, для последующего отображения их на странице.

Входными данными для него являются данные записанные в форму.

Выходными данными является запрос в БД для создания новой строки в таблице `menu`.

Алгоритм контроллера `add_menu` идентичен алгоритму контроллера `add_category`. Контроллер `add_menu` так же содержит две функции `obr` и `get_content`.

Функция `get_content` в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция `obr` получает данные от переменной `$_POST` и записывает данные в таблицу БД.

Входные данные это переменная `$_POST`.

Выходные данные это сессионная переменная `'res'` и запрос на запись данных.

Алгоритм функции `obr` представлен на рисунке 28.

У контроллера `add_menu` имеется представление `add_menu`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении содержится форма в которой содержится поле для ввода строки. Отправка формы происходит по нажатию кнопки.

Контроллер `add_topic` является дочерним классом `ACore_admin` и выполняет функцию добавления статей в БД, для последующего отображения их на странице.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		46



Рисунок 28 – Алгоритм функции obr

Входными данными для него являются данные записанные в форму и данные о категориях.

Выходными данными является запрос в БД для создания новой строки в таблице topics.

Алгоритм контроллера add\_topic идентичен алгоритму контроллера add\_category. Контроллер add\_topic так же содержит две функции obr и get\_content.

Функция get\_content в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция obr получает данные от переменной \$\_POST и записывает данные в таблицу БД.

Входные данные это переменная \$\_POST.

Выходные данные это сессионная переменная 'res' и запрос на запись данных.

Алгоритм функции `obr` представлен на рисунке 29.

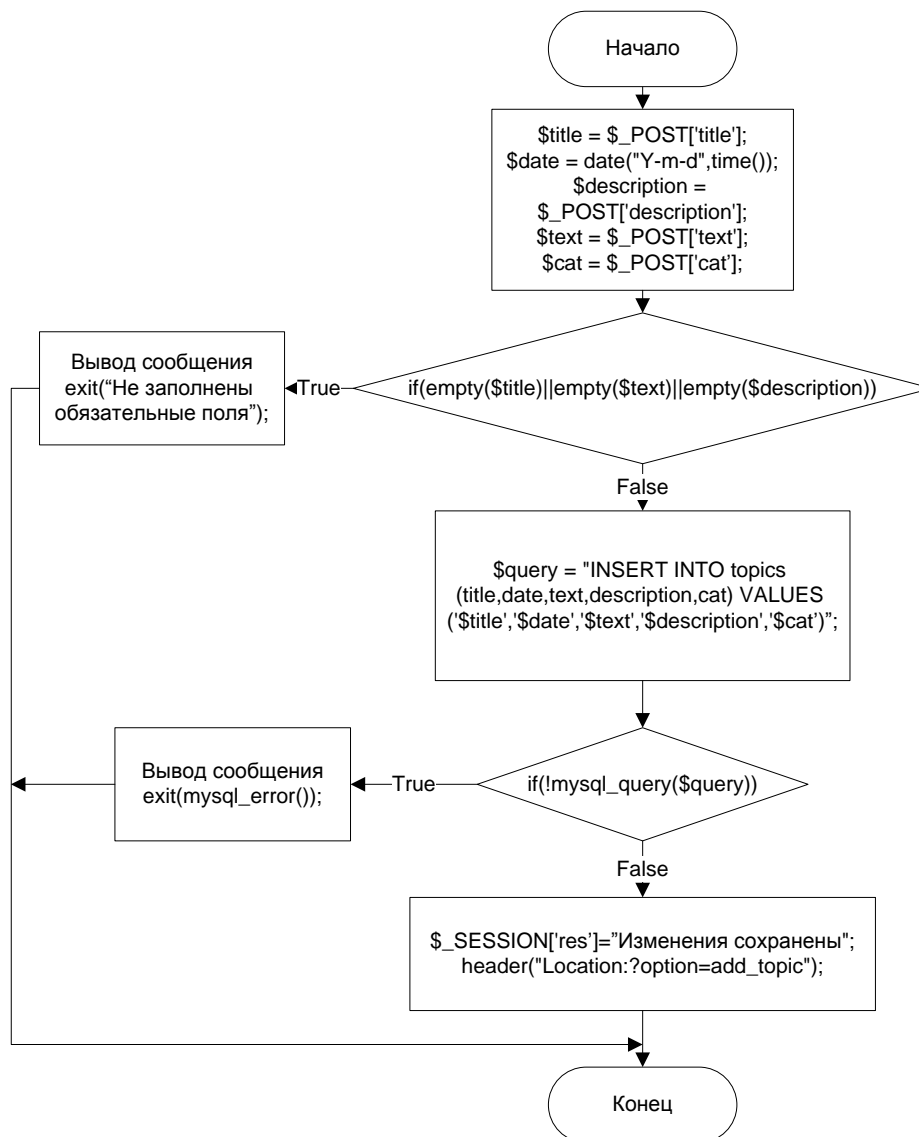


Рисунок 29 – Алгоритм функции `obr`

У контроллера `add_topic` имеется представление `add_topic`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении содержится форма в которой содержится поля для ввода данных. Отправка формы происходит по нажатию кнопки. В представлении происходит обращение к `model`.

Контроллер `admin` является дочерним классом `ACore_Admin` и выполняет функции вывода ссылок на изменение статей содержащихся в БД.

В контроллере выполняется одна функция, `get_content` которая делает обращение к `model`.



Выходными данными для нее являются данные полученные из таблиц БД.

У контроллера `admin` имеется представление `admin`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображен вывод строк обозначающих ссылки на статью, а так же ссылки на удаление и ссылка на создание новой статьи.

Контроллер `category` является дочерним классом `ACore` и выполняет функции вывода ссылок на статьи содержащихся в БД, отобранных по значению нужной категории.

В контроллере содержится одна функция `get_content`, где происходит проверка, и определение переменной `$_GET` в которой содержится номер категории, а затем происходит обращение в `model`.

Входные данные это значение переменной `$_GET`.

Выходные данные это данные полученные из таблиц БД.

У контроллера `category` имеется представление `category`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображен вывод сведений о статье, отсортированных по категории.

Контроллер `delete_category` является дочерним классом `ACore_Admin` и выполняет функцию удаления категорий из таблиц БД.

Входными данными для него является значение переменной `$_GET`.

Выходными данными является запрос в БД для удаления строки.

Алгоритм контроллера `delete_category` представлен на рисунке 30.

Все функции в контроллере выполняются последовательно.

Функция `get_content` в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция `obr` получает данные от переменной `$_DEL` и удаляет данные в зависимости от значения переменной.

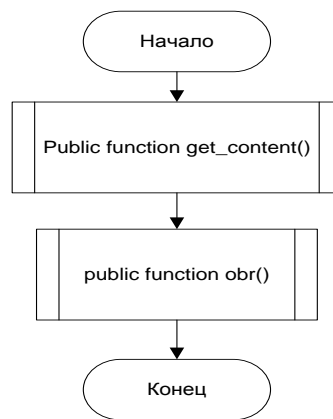


Рисунок 30 – Алгоритм контроллера delete\_category

Входные данные переменная \$\_DEL.

Выходные запрос в БД на удаление строки из таблицы.

Алгоритм функции obr представлен на рисунке 31.

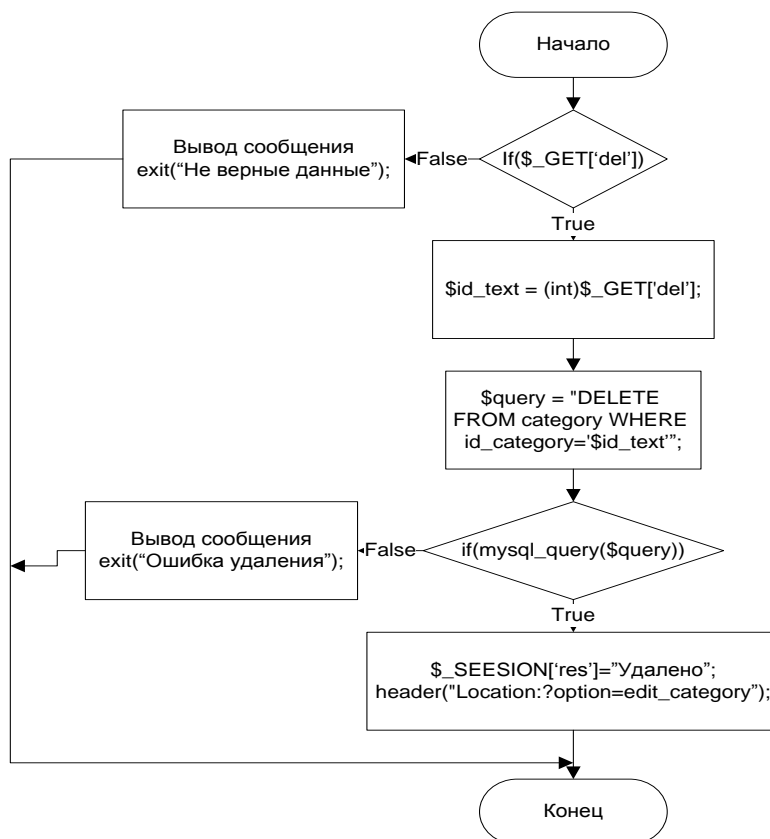


Рисунок 31 – Алгоритм функции obr

У контроллера delete\_category не имеется представления т.к. контроллер лишь выполняет функцию удаления, и затем перенаправляет пользователя на страницу редактирования категорий.

Контроллер delete\_menu является дочерним классом ACore\_Admin и выполняет функцию удаления пунктов меню из таблиц БД.

Входными данными для него является значение переменной \$\_GET.

Выходными данными является запрос в БД для удаления строки.

Алгоритм контроллера delete\_menu идентичен алгоритму контроллера delete\_category.

Функция get\_content в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция obr получает данные от переменной \$\_DEL и удаляет данные в зависимости от значения переменной.

Входные данные переменная \$\_DEL.

Выходные запрос в БД на удаление строки из таблицы.

Алгоритм функции obr представлен на рисунке 32.

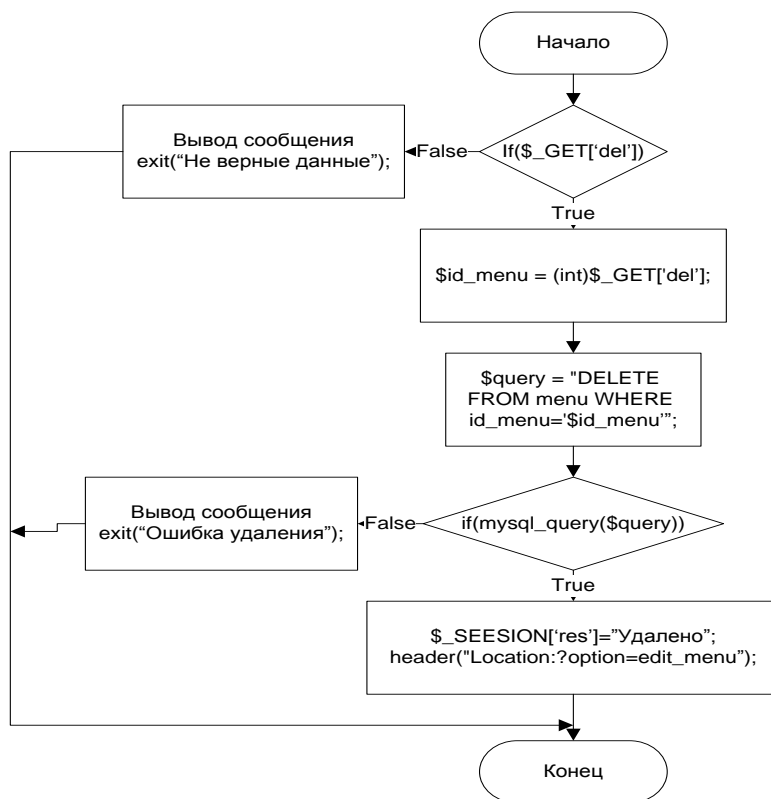


Рисунок 32 – Алгоритм функции obr

У контроллера delete\_menu не имеется представления, т.к. контроллер лишь выполняет функцию удаления, и затем перенаправляет пользователя на страницу редактирования меню.

Контроллер delete\_topic является дочерним классом ACore\_Admin и выполняет функцию удаления пунктов меню из таблиц БД.

Входными данными для него является значение переменной \$\_GET.

Выходными данными является запрос в БД для удаления строки.

Алгоритм контроллера delete\_topic идентичен алгоритму контроллера delete\_category.

Функция get\_content в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция obr получает данные от переменной \$\_DEL и удаляет данные в зависимости от значения переменной.

Входные данные переменная \$\_DEL.

Выходные запрос в БД на удаление строки из таблицы.

Алгоритм функции obr представлен на рисунке 33.

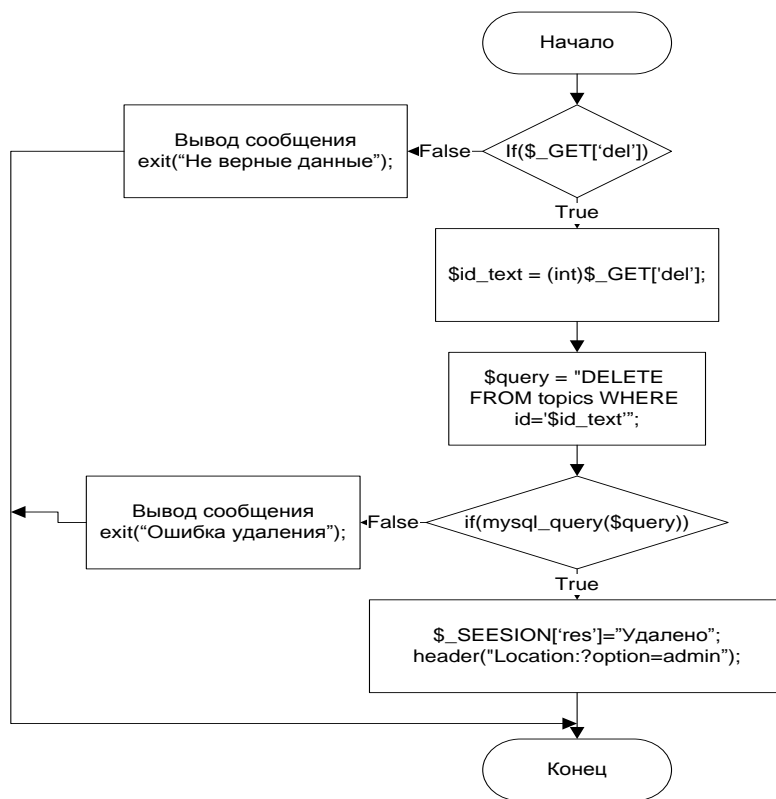


Рисунок 33 – Алгоритм функции obr

У контроллера delete\_topics не имеется представления т.к. контроллер лишь выполняет функцию удаления, и затем перенаправляет пользователя на страницу редактирования статей.

Контроллер edit\_category является дочерним классом ACore\_Admin и выполняет функции вывода ссылок на изменение категорий содержащихся в БД.

В контроллере выполняется одна функция, get\_content которая делает обращение к model.

Выходными данными для нее являются данные полученные из таблиц БД.

У контроллера edit\_category имеется представление edit\_category.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображен вывод строк обозначающих ссылки на категории, а так же ссылки на удаление и ссылка на создание новой категории.

Контроллер edit\_css является дочерним классом ACore\_Admin и выполняет функцию удаления пунктов меню из таблиц БД.

Алгоритм контроллера delete\_topic идентичен алгоритму контроллера add\_category.

Функция get\_content в данном контроллере остается пустой, т.к. контроллер не запрашивает данных из модели.

Функция obr получает данные при отправке формы, открывает поток записи в файл и записывает туда новые данные.

Входные данные переменная \$\_POST.

Выходные данные поток данных в файл.

Алгоритм функции obr представлен на рисунке 34.

У контроллера edit\_css имеется представление edit\_css.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображена форма, в которой расположено поле для редактирования CSS файла. Отправка формы происходит по нажатию на кнопку.

Контроллер edit\_menu является дочерним классом ACore\_Admin и выполняет функции вывода ссылок на изменение пунктов меню содержащихся в БД.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		53

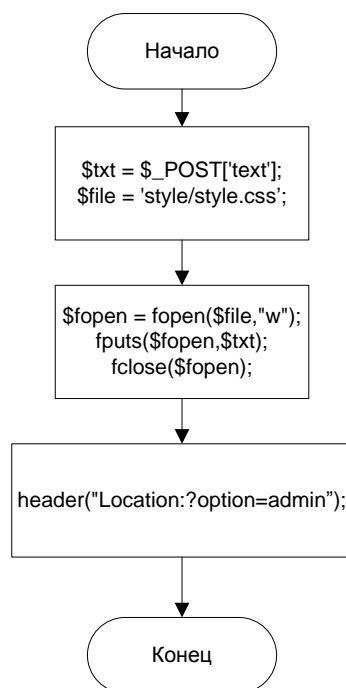


Рисунок 34 – Алгоритм функции obr

В контроллере выполняется одна функция, `get_content` которая делает обращение к `model`.

Выходными данными для нее являются данные полученные из таблиц БД.

У контроллера `edit_menu` имеется представление `edit_menu`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображен вывод строк обозначающих ссылки на пункты меню, а так же ссылки на удаление и ссылка на создание нового пункта меню.

Контроллер `exit_admin` является дочерним классом `ACore_Admin` и выполняет функции выхода из панели администрации.

В контроллере выполняется одна функция, `get_content` которая производит перенаправление пользователя на главную страницу сайта и очищение сессионной переменной `'user'`.

У контроллера `exit_admin` не имеется представления т.к. контроллер лишь выполняет функцию выхода.

Контроллер login является дочерним классом ACore и выполняет функцию авторизации пользователя для последующего перенаправления в панель администрирования.

Входными данными для него является значение сессионной переменной 'user'.

Алгоритм контроллера login идентичен алгоритму контроллера add\_category.

Функция get\_content проверяет наличие переменной 'user' у пользователя и при ее присутствии перенаправляет пользователя в панель администрации.

Входные переменные сессионная переменная 'user'.

Функция obr получает данные от переменной \$\_POST и производит ряд запросов и проверок.

Входные данные это переменная \$\_POST и данные из таблиц БД.

Выходные данные это сессионная переменная 'user'.

Алгоритм функции obr представлен на рисунке 35.

У контроллера login имеется представление login.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода.

В представлении отображена форма, в которой расположено поле для ввода данных пользователя. Отправка формы происходит по нажатию на кнопку.

Контроллер main является дочерним классом ACore и выполняет функции вывода ссылок на изменение категорий содержащихся в БД.

В контроллере выполняется одна функция, get\_content которая делает обращение к model.

Выходными данными для нее являются данные полученные из таблиц БД.

У контроллера main имеется представление main.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		55

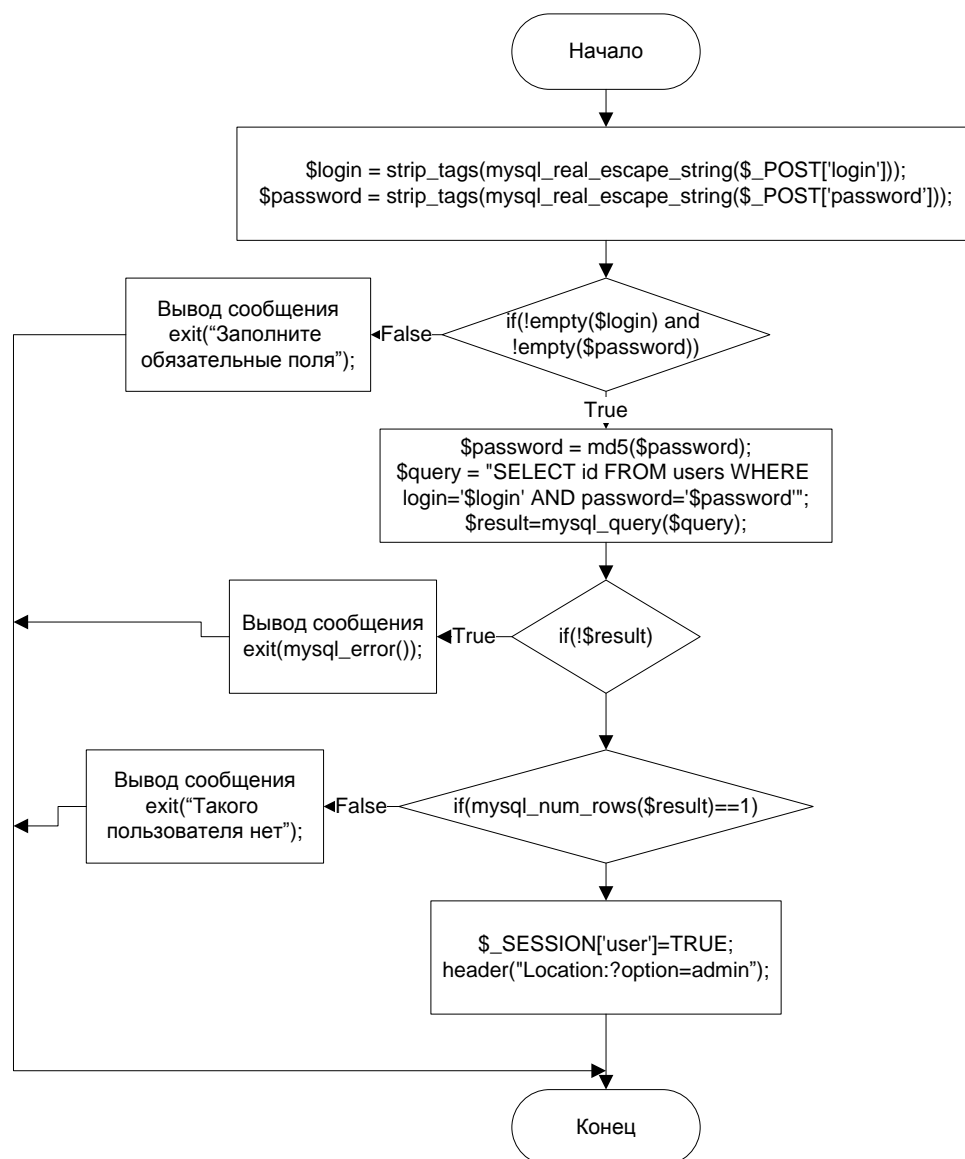


Рисунок 35 – Алгоритм функции `obr`

В представлении отображен вывод данных о статьях содержащихся в таблицах БД, и ссылки на их полное отображение. В представлении выполняется обращение к `model`.

Контроллер `menu` является дочерним классом `ACore` и выполняет функции вывода ссылок на изменение категорий содержащихся в БД.

В контроллере выполняется одна функция, `get_content` которая делает обращение к `model`.

Выходными данными для нее являются данные полученные из таблиц БД.

У контроллера `menu` имеется представление `menu`.



Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображен вывод данных о пунктах меню содержащихся в таблицах БД. В представлении выполняется обращение к model.

Контроллер `update_category` является дочерним классом `ACore_admin` и выполняет функцию изменения названия категорий.

Входными данными для него являются данные записанные в форму.

Выходными данными является запрос в БД на изменение строки в таблице `category`.

Алгоритм контроллера `update_category` идентичен алгоритму контроллера `add_category`.

Функция `get_content` выполняет функцию определения номера категории, для создания запроса в БД.

Функция `obr` получает данные от переменной `$_POST` и обновляет данные в таблице БД.

Входные данные это переменная `$_POST`.

Выходные данные это сессионная переменная `'res'` и запрос на изменение данных.

Алгоритм функции `obr` представлен на рисунке 36.

У контроллера `update_category` имеется представление `update_category`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML, а так же встроенных функций PHP кода.

В представлении отображена форма, содержащая текстовые поля для заполнения. Отправка формы происходит по нажатию на кнопку. В представлении выполняется обращение к model.

Контроллер `update_menu` является дочерним классом `ACore_admin` и выполняет функцию изменения данных пунктов меню.

Входными данными для него являются данные записанные в форму.

Выходными данными является запрос в БД на изменение строки в таблице `category`.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		57

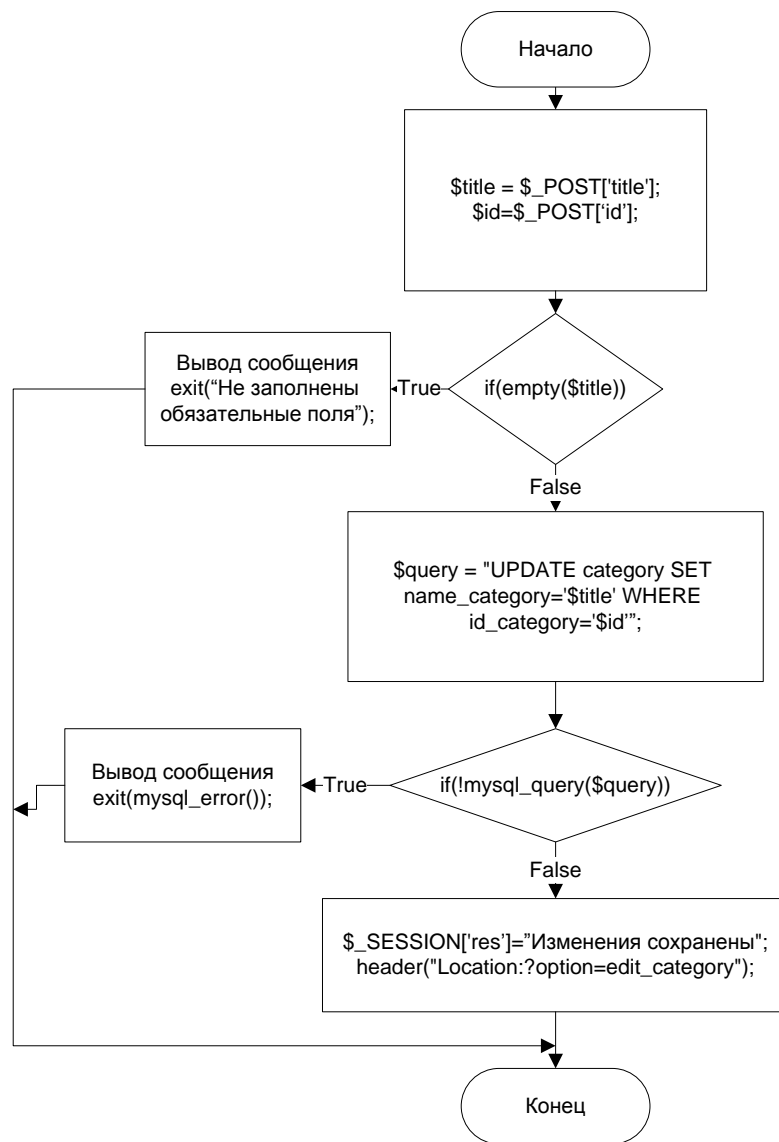


Рисунок 36 – Алгоритм функции obr

Алгоритм контроллера update\_menu идентичен алгоритму контроллера add\_category.

Функция get\_content выполняет функцию определения номера пункта меню, для создания запроса в БД.

Функция obr получает данные от переменной \$\_POST и обновляет данные в таблице БД.

Входные данные это переменная \$\_POST.

Выходные данные это сессионная переменная 'res' и запрос на изменение данных.

Алгоритм функции obr представлен на рисунке 37.

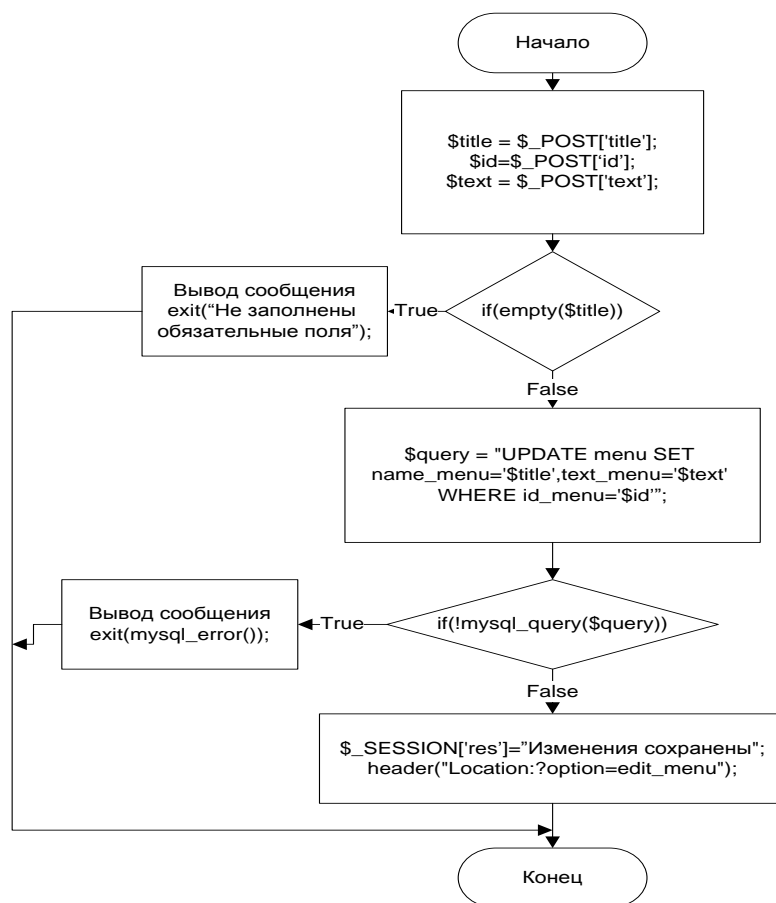


Рисунок 37 – Алгоритм функции `obr`

У контроллера `update_menu` имеется представление `update_menu`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML, а так же встроенных функций PHP кода.

В представлении отображена форма, содержащая текстовые поля для заполнения. Отправка формы происходит по нажатию на кнопку. В представлении выполняется обращение к `model`.

Контроллер `update_topic` является дочерним классом `ACore_admin` и выполняет функцию изменения статей.

Входными данными для него являются данные записанные в форму.

Выходными данными является запрос в БД на изменение строки в таблице `category`.

Алгоритм контроллера `update_topic` идентичен алгоритму контроллера `add_category`.

Функция `get_content` выполняет функцию определения номера статьи, для создания запроса в БД.

Функция `obr` получает данные от переменной `$_POST` и обновляет данные в таблице БД.

Входные данные это переменная `$_POST`.

Выходные данные это сессионная переменная `'res'` и запрос на изменение данных.

Алгоритм функции `obr` представлен на рисунке 38.

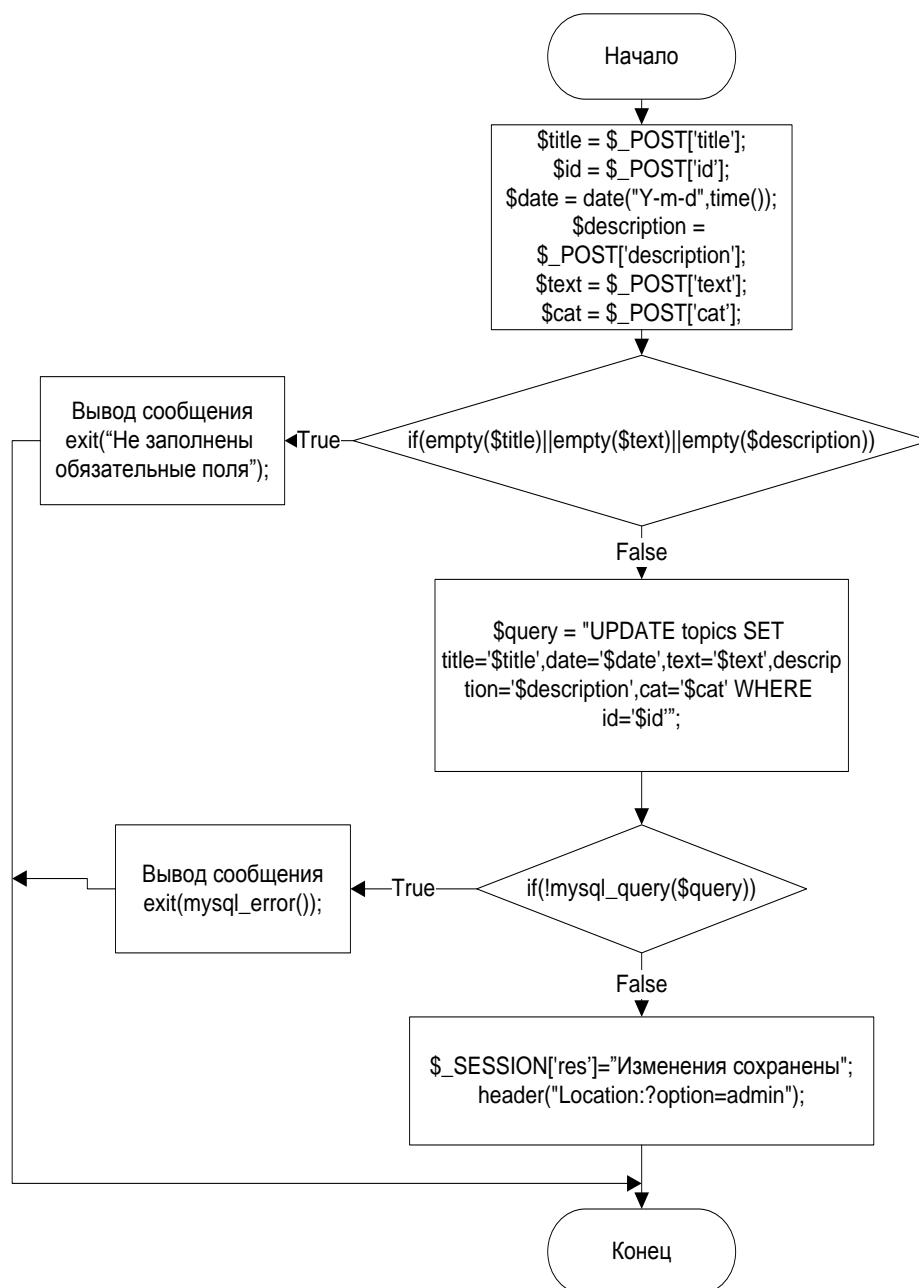


Рисунок 38 – Алгоритм функции `obr`

У контроллера `update_topic` имеется представление `update_topic`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML, а так же встроенных функций PHP кода.

В представлении отображена форма, содержащая текстовые поля для заполнения. Отправка формы происходит по нажатию на кнопку. В представлении выполняется обращение к `model`.

Контроллер `view` является дочерним классом `ACore` и выполняет функции отображения статьи.

В контроллере выполняется одна функция, `get_content` которая делает проверки о правильности запроса и обращение к `model`.

Входными данными является значение переменной `$_GET`.

Выходными данными для нее являются данные полученные из таблиц БД.

У контроллера `view` имеется представление `view`.

Представление содержит в себе визуальное отображение модуля, которое выражено в виде HTML кода, а так же встроенных функций PHP кода.

В представлении отображен вывод данных о статье содержащейся в таблицах БД. В представлении выполняется обращение к `model`.

### **3.2 Описание интерфейса**

Система состоит из двух основных страниц: главная страница и панель администрирования.

Главная страница сайта представлена на рисунке 39.

Статьи на главной странице могут быть отфильтрованы с помощью пунктов на левой боковой панели.

Каждая статья сайта имеет полный текст, который отображается по нажатию на ссылку «Читать далее».

Все пункты на верхней панели меню, кроме «Главная» и «Администрирование», имеют свой полный текст. Пункт «Главная» осуществляет переход на главную страницу сайта.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		61

Меню

>> [О CMS](#)

>> [Новости](#)

>> [Новости обновлений](#)

### Современные CMS

2016-05-20

В настоящее время существует множество CMS на любой вкус и цвет и особо из них выделяются следующие кандидаты

- Joomla
- WordPress
- Drupal
- MODX

[Читать далее...](#)

### Ver 0.01

2016-05-20

Особенности версии:

- Наличие редактирования содержимого статей.
- Обновление интерфейса.

[Читать далее...](#)

Рисунок 39 – Главная страница сайта

Пример показа полного текста статьи представлен на рисунке 40.

Меню

>> [О CMS](#)

>> [Новости](#)

>> [Новости обновлений](#)

### Современные CMS

2016-05-20

В настоящее время существует множество CMS на любой вкус и цвет и особо из них выделяются следующие кандидаты

- Joomla
- WordPress
- Drupal
- MODX

### Wordpress

WordPress — система управления содержимым сайта с открытым исходным кодом; написана на PHP; сервер базы данных — MySQL; выпущена под лицензией GNU GPL версии 2. Сфера применения — от блогов до достаточно сложных новостных ресурсов и интернет-магазинов. Встроенная система «тем» и «плагинов» вместе с удачной архитектурой позволяет конструировать проекты широкой функциональной сложности.

### Joomla

Рисунок 40 – Полный текст статьи

Полный текст пункта меню представлен на рисунке 41.

					<i>ВКР. 125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		62



Рисунок 41 – Полный текст пункта меню

При нажатии на пункт «Администрирование» открывается форма входа, или же, если вход был выполнен раньше, то пользователь будет перенаправлен в панель администрирования.

Форма входа в панель администрирования представлена на рисунке 42.

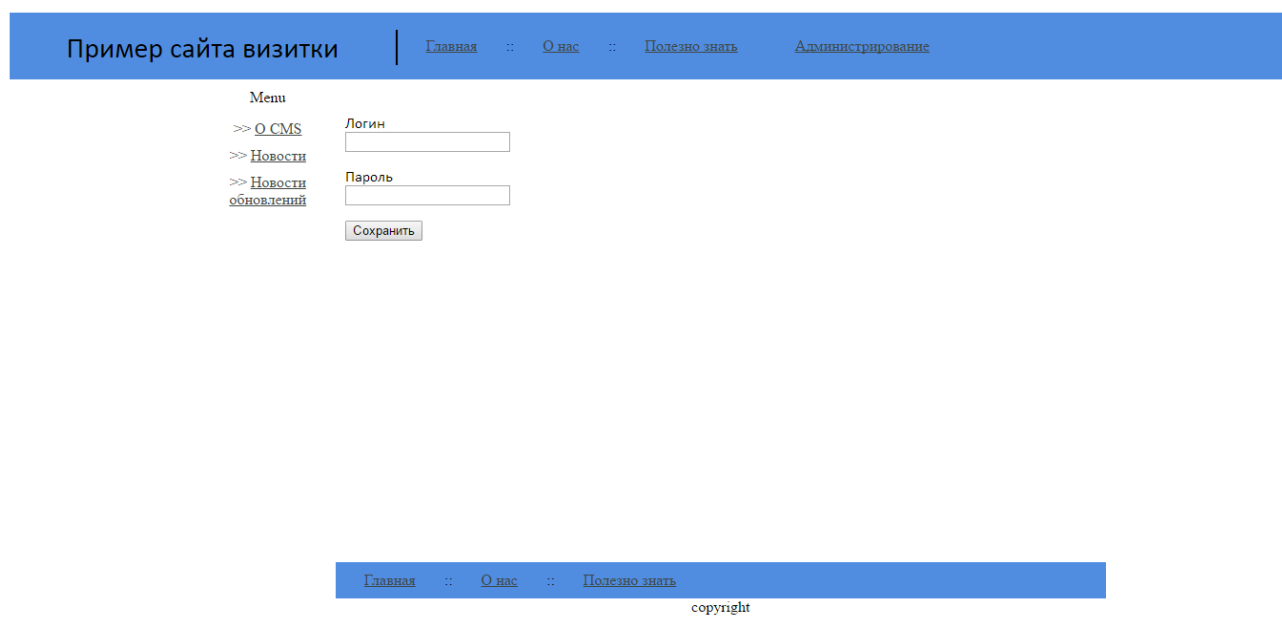


Рисунок 42 – Форма входа в панель администрирования

После введения верных данных пользователь попадает в панель администрирования.

Главная страница панели администрирования представлена на рисунке 43.

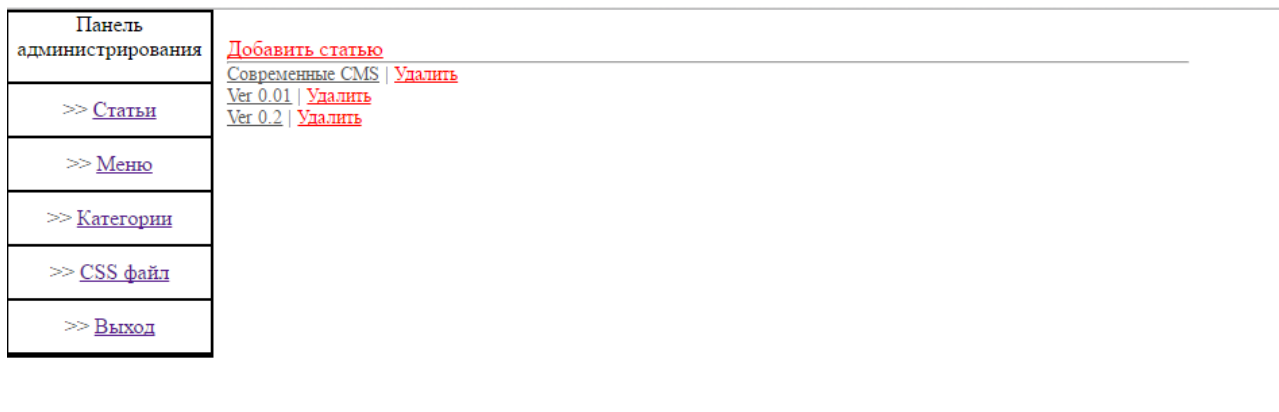


Рисунок 43 – Главная страница панели администрирования

Главная страница статьи имеет одинаковое представление с пунктом меню «Статьи» в которой представлены статьи сайта. По нажатию на ссылку «Добавить статью» откроется окно добавления новой статьи. По нажатию на пункт статьи откроется окно редактирования статьи. По нажатию на ссылку «Удалить» рядом со статьей, статья удалится из таблицы БД.

Окно добавления новой статьи представлено на рисунке 44.



Рисунок 44 – Окно добавления статьи

По нажатию на кнопку сохранить, статья записывается в таблицу БД, открывается новое окно добавления статьи. Все поля обязательны к заполнению, возможно использование тегов HTML и языка стилей CSS для оформления.

Окно редактирования статьи представлено на рисунке 45.



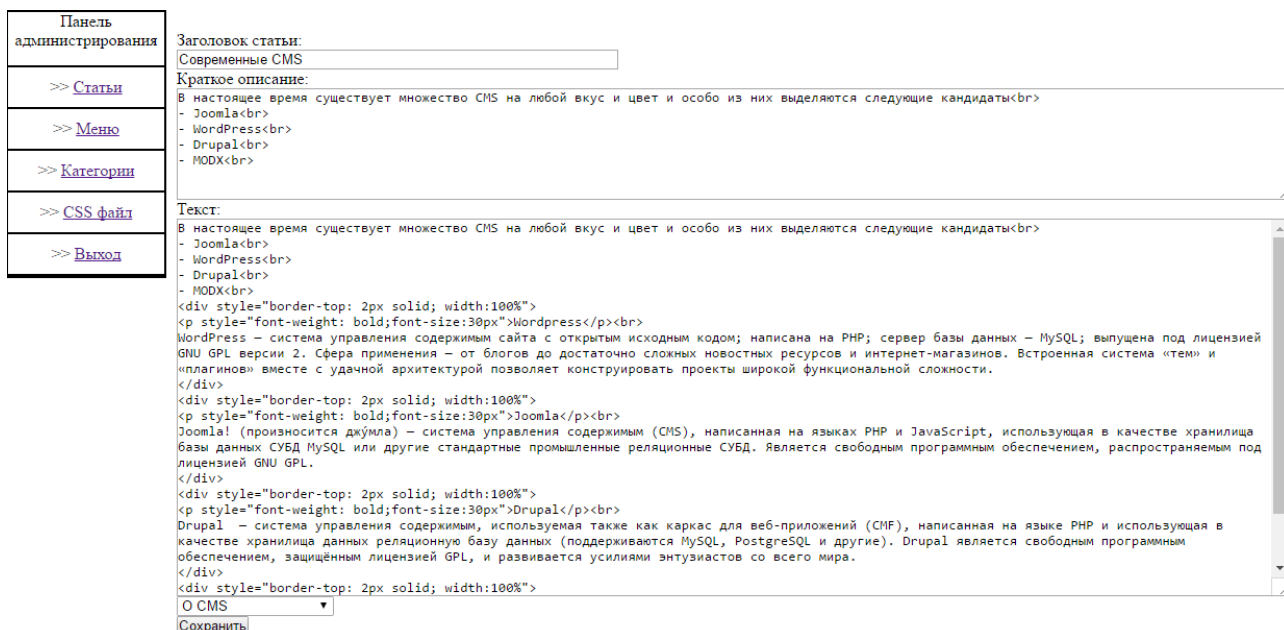


Рисунок 45 – Окно редактирования статьи

По нажатию на кнопку сохранить данные в таблице БД обновляются, происходит перенаправление на главную страницу панели администрирования.

Все поля обязательны к заполнению, возможно использование тегов HTML и языка стилей CSS для оформления.

По нажатию на пункт меню «Меню» открывается окно где представлены пункты меню сайта.

Окно пункта меню «Меню» представлено на рисунке 46.

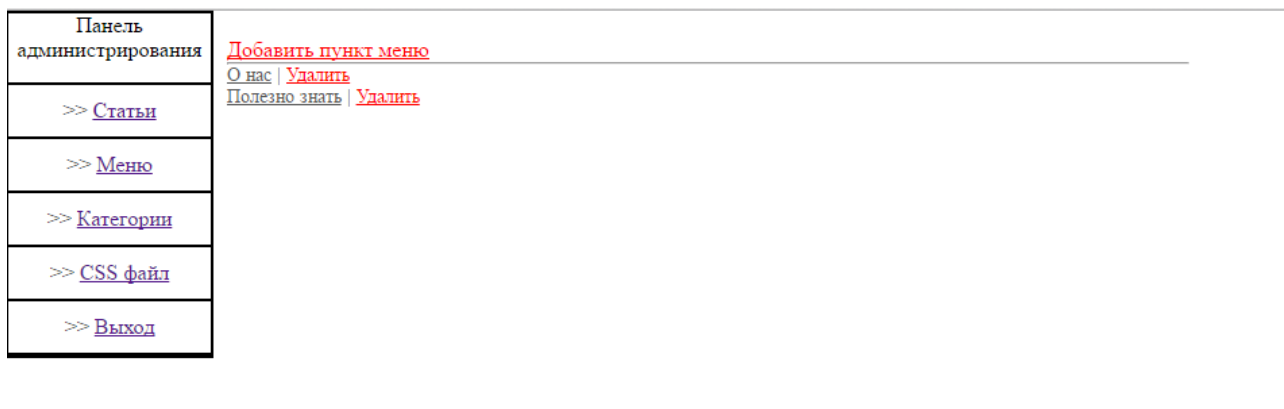


Рисунок 46 – Окно пункта меню «Меню»

По нажатию на ссылку «Добавить пункт меню» откроется окно добавления нового пункта меню. По нажатию на пункт меню откроется окно редакти-

рования пункта меню. По нажатию на ссылку «Удалить» рядом с пунктом меню, пункт меню удалится из таблицы БД.

Окно добавления нового пункта меню представлено на рисунке 47.

Панель администрирования  <a href="#">&gt;&gt; Статьи</a>  <a href="#">&gt;&gt; Меню</a>  <a href="#">&gt;&gt; Категории</a>  <a href="#">&gt;&gt; CSS файл</a>  <a href="#">&gt;&gt; Выход</a>	Заголовок меню:
	Пункт меню
	Текст:
	Текст пункта меню
	Сохранить

Рисунок 47 – Окно добавления нового пункта меню

По нажатию на кнопку сохранить, пункт меню записывается в таблицу БД, открывается новое окно добавления пункта меню. Все поля обязательны к заполнению, возможно использование тегов HTML и языка стилей CSS для оформления.

Окно редактирования пункта меню представлено на рисунке 48.

Панель администрирования  <a href="#">&gt;&gt; Статьи</a>  <a href="#">&gt;&gt; Меню</a>  <a href="#">&gt;&gt; Категории</a>  <a href="#">&gt;&gt; CSS файл</a>  <a href="#">&gt;&gt; Выход</a>	Заголовок меню:
	О нас
	Текст:
	Данная работа выполнена для защиты диплома на тему: Разработка CMS-системы для сайта-визитки Исполнитель Шишковский Кирилл Олегович студент группы 25506
	Сохранить

Рисунок 48 – Окно редактирования пункта меню

По нажатию на кнопку сохранить данные в таблице БД обновляются, происходит перенаправление на окно пункта меню «Меню».

Все поля обязательны к заполнению, возможно использование тегов HTML и языка стилей CSS для оформления.

По нажатию на пункт меню «Категории» открывается окно, где представлены категории статей сайта.

Окно пункта меню «Категории» представлено на рисунке 49.

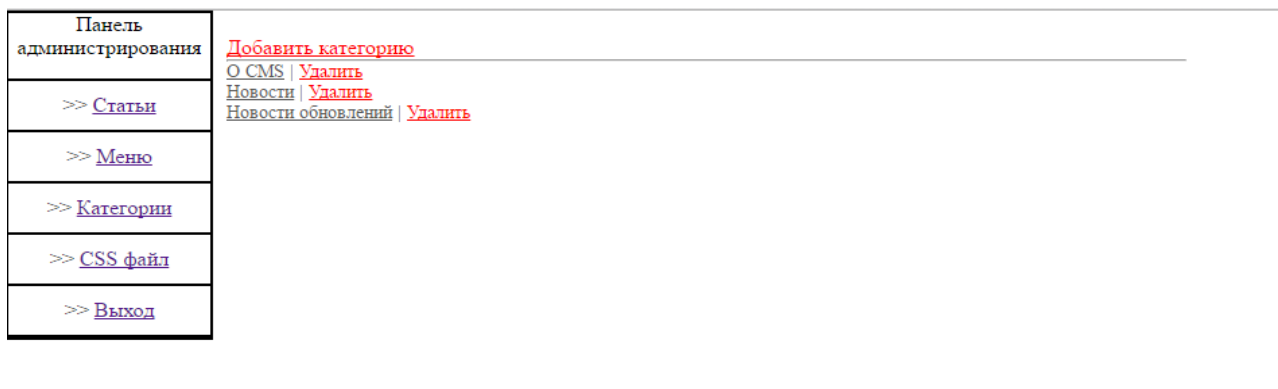


Рисунок 49 – Окно пункта меню «Категории»

По нажатию на ссылку «Добавить категорию» откроется окно добавления новой категории. По нажатию на пункт категории откроется окно редактирования наименования категории. По нажатию на ссылку «Удалить» рядом с категорией, категория удалится из таблицы БД.

Окно добавления новой категории представлено на рисунке 50.

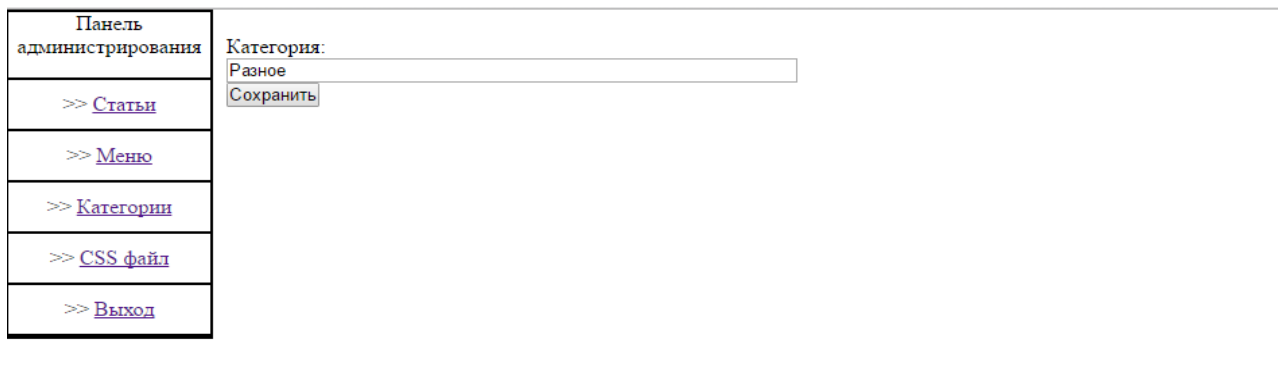


Рисунок 50 – Окно добавления новой категории

По нажатию на кнопку сохранить, категория записывается в таблицу БД, открывается новое окно добавления категории.

Окно редактирования категории представлено на рисунке 51.

По нажатию на кнопку сохранить данные в таблице БД обновляются, происходит перенаправление на окно пункта меню «Категории».

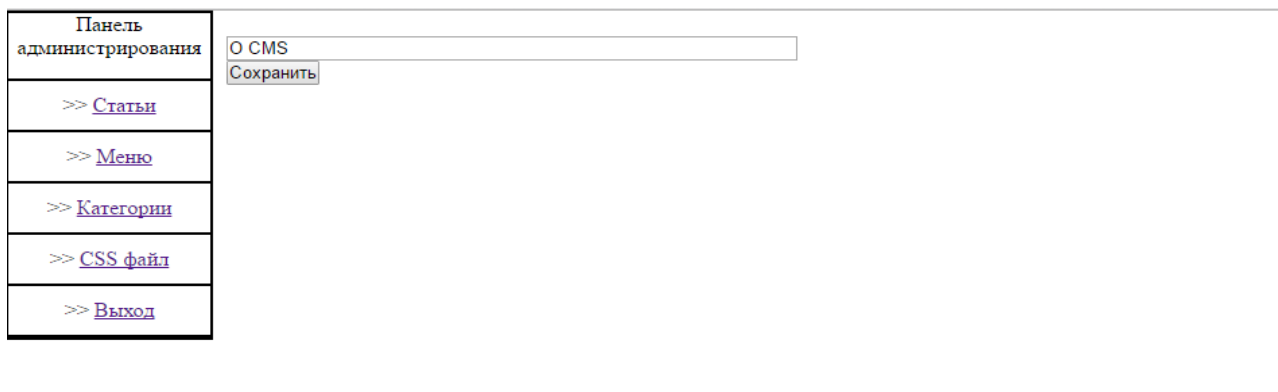


Рисунок 51 – Окно редактирования категории

По нажатию на пункт меню «CSS файл» открывается окно редактирования файла стилей CSS для страницы сайта.

Окно редактирования файла стилей CSS для страницы сайта отображено на рисунке 52.

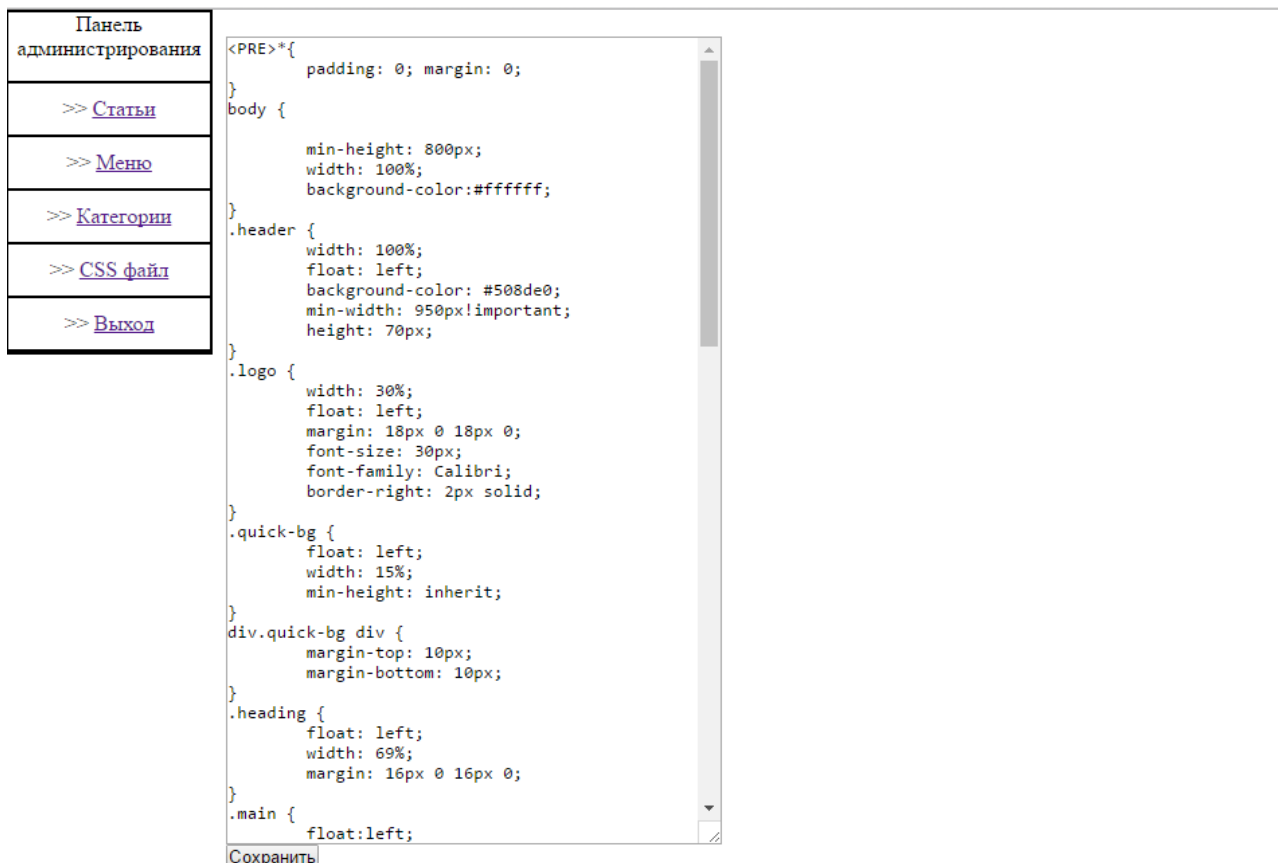


Рисунок 52 – Окно редактирования файла стилей CSS для страницы сайта

По нажатию на кнопку сохранить, данные перезаписываются в файл, открывается главная страница панели администрирования.

По нажатию на пункт меню «Выход» происходит выход из панели администратии и перенаправление пользователя на главную страницу сайта.

## ЗАКЛЮЧЕНИЕ

В итоге проведенной работы, было исследовано предприятия ООО «Зебра», проведено исследование его внешнего и внутреннего документооборота, аппаратного и программного обеспечения, топология сети.

Так же, была спроектирована CMS-система для производственного отдела предприятия, выделены ее подсистемы, функции, спроектировано информационное и программное обеспечения.

Были рассмотрены особенности алгоритмов функций и модулей системы, выделены основные моменты их взаимодействия.

В завершении работы была разработана CMS-система, в которой для решения поставленных задач были спроектированы и разработаны следующие модули и механизмы:

- оптимизирован механизм вывода страниц сайта путем применения шаблона проектирования MVC;
- разработаны модули редактирования контента сайта в панели администрации;
- реализована возможность вывода в поле редактирования CSS файла для его изменения;
- разработана панель администрации для реализации возможностей редактирования.

CMS-система будет использоваться для создания рекламных сайтов-визиток, страницы сайта будут динамически генерируемыми с помощью модулей CMS-системы. Система имеет дальнейший потенциал на развитие, добавления новых модулей, расширения набора функций сайта и усовершенствования интерфейса.

					ВКР.125047.09.03.02.ПЗ	Лист
Изм	Лист	№ докум.	Подп.	Дата		69

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Анин, Б.Ю. Защита компьютерной информации / Б.Ю. Анин. – СПб.: БХВ – Санкт-Петербург, 2000. – 384 с.

2 Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. – 2-е изд., испр. – М. : Национальный Открытый Университет «ИНТУИТ», 2016. – 265 с.

3 Буренин, С.Н. Web-программирование и базы данных: учебный практикум/ Буренин С.Н. – Электрон. текстовые данные. – М.: Московский гуманитарный университет, 2014. – 120 с.

4 Горизонтальное меню на CSS[Электронный ресурс]. – Режим доступа: <http://sitear.ru/material/horizontal-menu-css>. – 5.04.2016.

5 Диков, А.В. Веб-технологии HTML и CSS : учебное пособие / А.В. Диков. – 2-е изд. – М. : Директ-Медиа, 2012. – 78 с.

6 Сидерхолм, Д. CSS3 для веб-дизайнеров / Дэн Сидерхолм – Электрон. текстовые данные. – М.: Манн, Иванов и Фербер, 2013. – 135 с.

7 Заика, А.А. Локальные сети и интернет / Заика А.А. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 323 с.

8 Маркотт, И. Отзывчивый веб-дизайн / Итан Маркотт – Электрон. текстовые данные. – М.: Манн, Иванов и Фербер, 2012. – 173 с.

9 Кудряшев, А.В. Введение в современные веб-технологии / Кудряшев А.В., Светашков П.А. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2010. – 208 с.

10 Кузнецов, М.В. PHP5. Практика разработки WEB-сайтов / И.В. Симдянов, С.В. Голышев – СПб.: БХВ – Петербург, 2005. – 960 с.

11 Лыткина, Е.А. Основы языка HTML : учебное пособие / Е.А. Лыткина, А.Г. Глотова ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северный (Арктический)

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		70

федеральный университет имени М. В. Ломоносова», Министерство образования и науки Российской Федерации. – Архангельск : САФУ, 2014. – 104 с.

12 Маркин, А.В. Основы web-программирования на PHP : учебное пособие / А.В. Маркин, С.С. Шкарин. – М. : Диалог-МИФИ, 2012. – 252 с.

13 Некоторые концепции для новичков: Как работают селекторы в CSS | CSS [Электронный ресурс]. – Режим доступа: [http://www.internet-technologies.ru/articles/article\\_2065.html](http://www.internet-technologies.ru/articles/article_2065.html). – 5.04.2016.

14 Панфилов, К.С. Создание веб-сайта от замысла до реализации [Электронный ресурс] / Панфилов К.С.– Электрон. текстовые данные. – М.: ДМК Пресс, 2009. – 440 с.

15 Пархимович, М.Н. Основы интернет-технологий : учебное пособие / М.Н. Пархимович, А.А. Липницкий, В.А. Некрасова ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северный (Арктический) федеральный университет имени М. В. Ломоносова», Министерство образования и науки Российской Федерации. – Архангельск : ИПЦ САФУ, 2013. – 366 с.

16 Полное руководство по центрированию DIV – элемента | CSS [Электронный ресурс]. – Режим доступа: [http://www.internet-technologies.ru/articles/article\\_2795.html](http://www.internet-technologies.ru/articles/article_2795.html). – 30.03.2016.

17 Применение каскадных таблиц стилей (CSS) : курс / П. Храмцов, С. Брик, А. Русак, А. Сурин. – 2-е изд., исправ. – М. : Национальный Открытый Университет «ИНТУИТ», 2016. – 82 с.

18 Савельев, А.О. HTML 5. Основы клиентской разработки / Савельев А.О., Алексеев А.А. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2012. – 166 с.

19 Савельева, Н.В. Язык программирования PHP / Н.В. Савельева. – 2-е изд., испр. – М. : Национальный Открытый Университет «ИНТУИТ», 2016. – 330 с.

20 Скляр, Д. PHP. Сборник рецептов / Д. Скляр, А. Трахтенберг. – СПб: БХВ-Петербург, 2007. – 250с.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		71

21 Сотникова, О.П. Интернет-издание от А до Я: руководство для веб-редактора. Учебное пособие для студентов вузов / Сотникова О.П. – Электрон. текстовые данные. – М.: Аспект Пресс, 2014. – 160 с.

22 Справочник по языку SQL (DML) [www.sql-ex.ru](http://www.sql-ex.ru) [Электронный ресурс]. – Режим доступа : <http://www.sql-ex.ru/help/?Lang=0>. – 10.04.2016.

23 Справочник CSS | [htmlbook.ru](http://htmlbook.ru) [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/css>. – 20.03.2016.

24 Стасышин, В.М. Проектирование информационных систем и баз данных: учебное пособие / Стасышин В.М. – Электрон. текстовые данные. – Новосибирск: Новосибирский государственный технический университет, 2012. – 100 с.

25 Сычев, А.В. Перспективные технологии и языки веб-разработки / Сычев А.В. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 493 с.

26 Тузовский, А.Ф. Проектирование и разработка web-приложений: учебное пособие / Тузовский А.Ф. – Электрон. текстовые данные. – Томск: Томский политехнический университет, 2014. – 219 с.

27 Филиппов, С.А. Основы современного веб-программирования : учебное пособие / С.А. Филиппов. – М. : МИФИ, 2011. – 160 с.

28 Хачатрян, К.Г. Гипертекстовая технология / К.Г. Хачатрян. – М. : Лаборатория книги, 2012. – 121 с.

29 PHP.su – Оптимизация запросов в MySQL [Электронный ресурс]. – Режим доступа: <http://www.php.su/articles/?cat=phpdb&page=005>. – 26.03.2016.

30 PHP.su – Пишем код, устойчивый к ошибкам [Электронный ресурс]. – Режим доступа: <http://www.php.su/articles/?cat=common&page=003>. – 21.03.2016.

31 PHP.su – Проектирование интернет приложений [Электронный ресурс]. – Режим доступа: <http://www.php.su/articles/?cat=common&page=002>. – 20.03.2016.

					<i>ВКР.125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		72



32 PHP.su – Работа с MySQL. Новостная лента для страницы [Электронный ресурс]. – Режим доступа: <http://www.php.su/articles/?cat=phpdb&page=003>. – 25.03.2016.

33 PHP.su – Сессии в PHP [Электронный ресурс]. – Режим доступа: <http://www.php.su/articles/?cat=examples&page=070>. – 26.03.2016.

34 PHP: Руководство по PHP – Manual [Электронный ресурс]: офиц.сайт. – Режим доступа: <http://php.net/manual/ru/index.php>. – 15.03.2016.

					<i>ВКР. 125047.09.03.02.ПЗ</i>	<i>Лист</i>
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>		73

## ПРИЛОЖЕНИЕ А

### DFD диаграмма внешнего документооборота предприятия

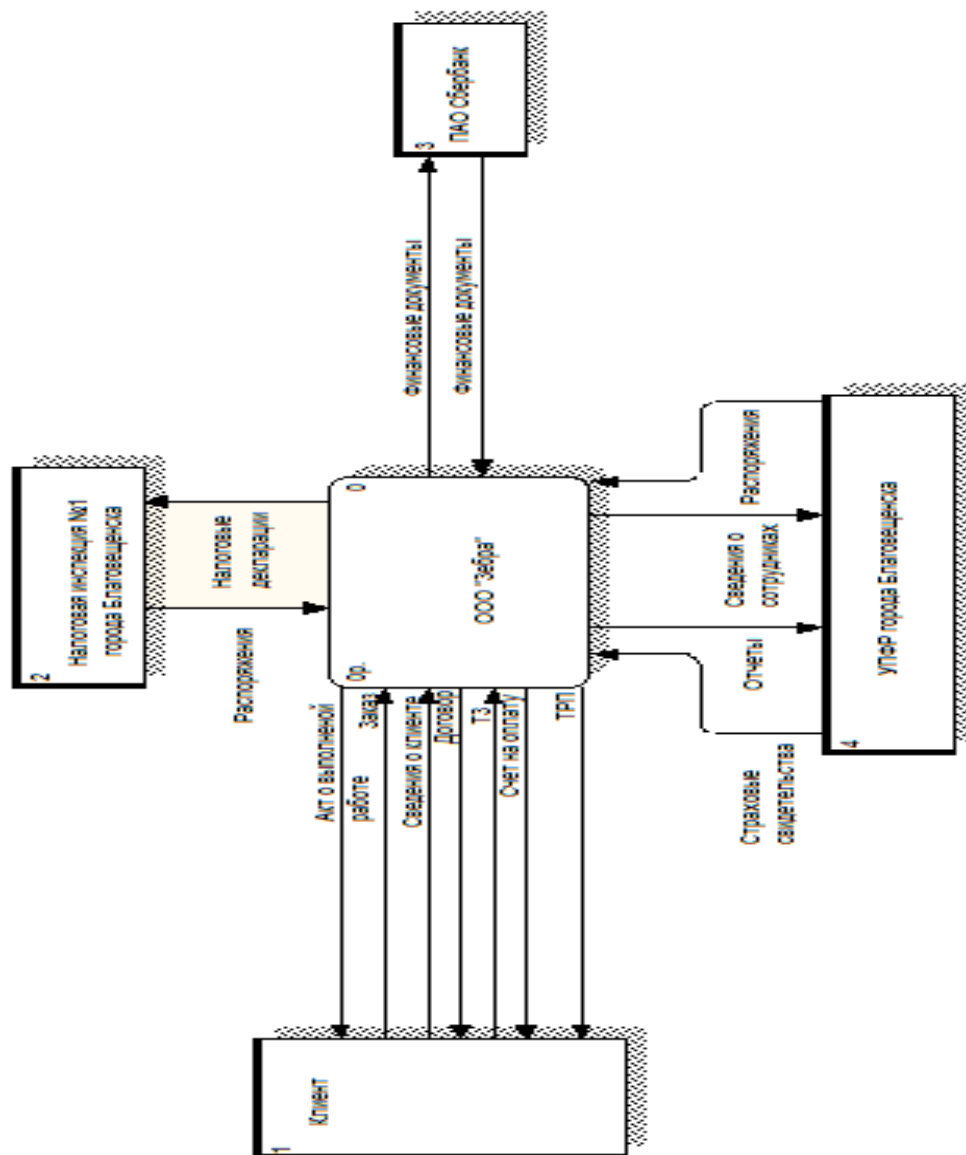


Рисунок А.1 – DFD диаграмма внешнего документооборота предприятия  
ООО «Зебра»

## ПРИЛОЖЕНИЕ Б

### DFD Диаграмма внутреннего документооборота предприятия

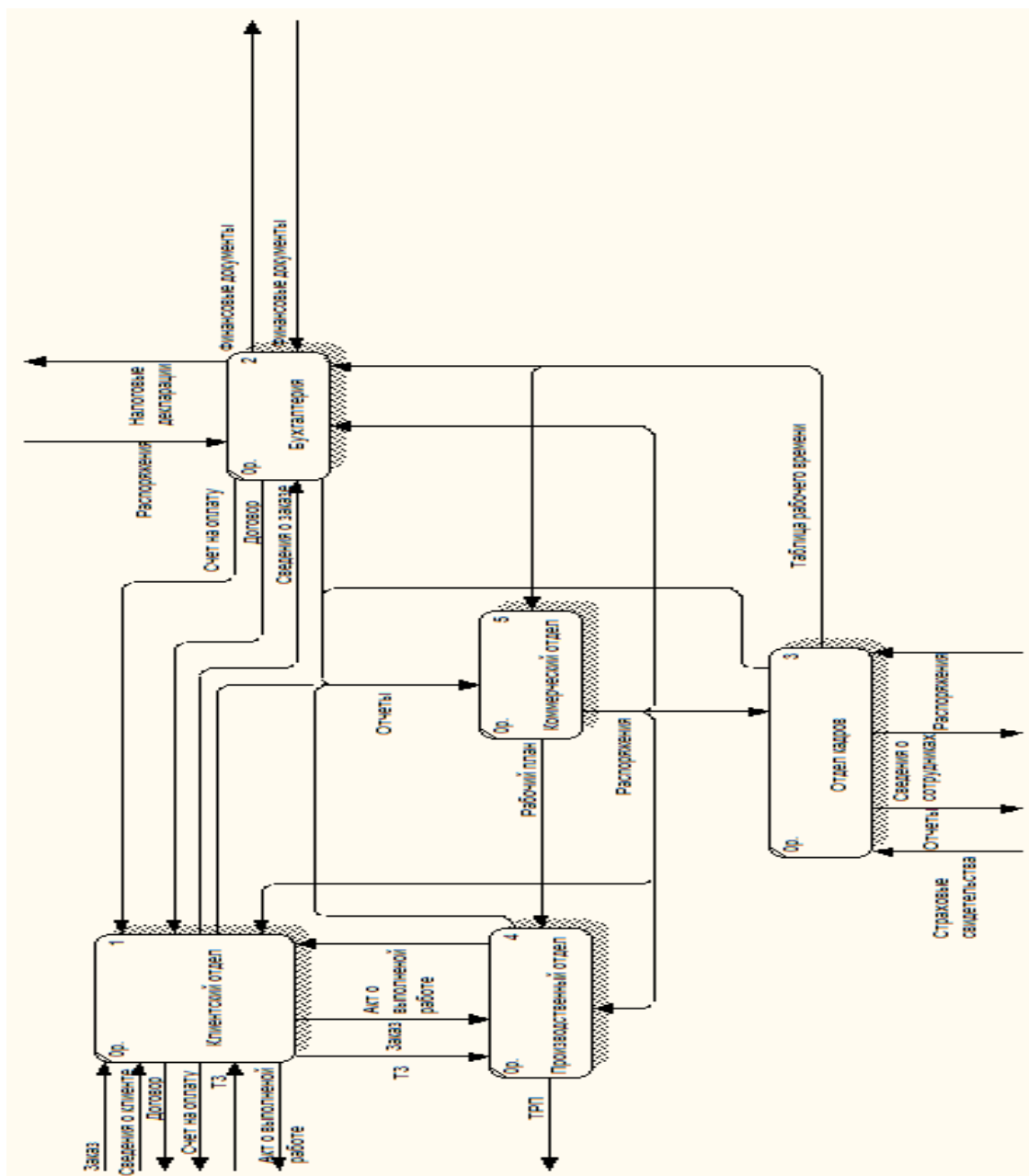


Рисунок Б.1 – DFD диаграмма внутреннего документооборота предприятия ООО «Зebra»

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

ВКР.125047.09.03.02.ПЗ

Лист

75

ПРИЛОЖЕНИЕ В  
 IDEF0 диаграммы CMS-системы

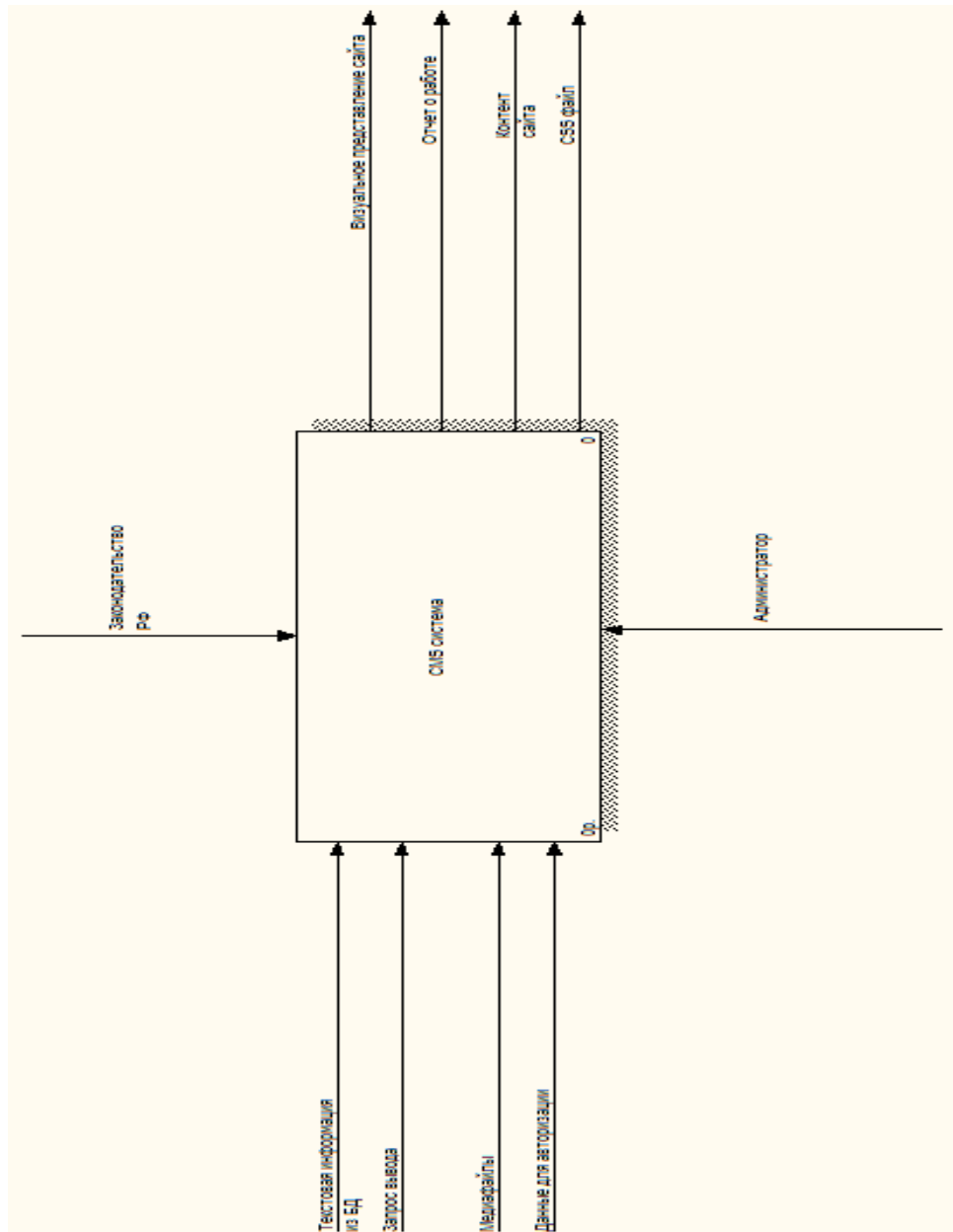


Рисунок В.1 – IDEF0 диаграмма CMS системы

Изм	Лист	№ докум.	Подп.	Дата

ВКР.125047.09.03.02.ПЗ

Лист

76

## Продолжение ПРИЛОЖЕНИЯ В

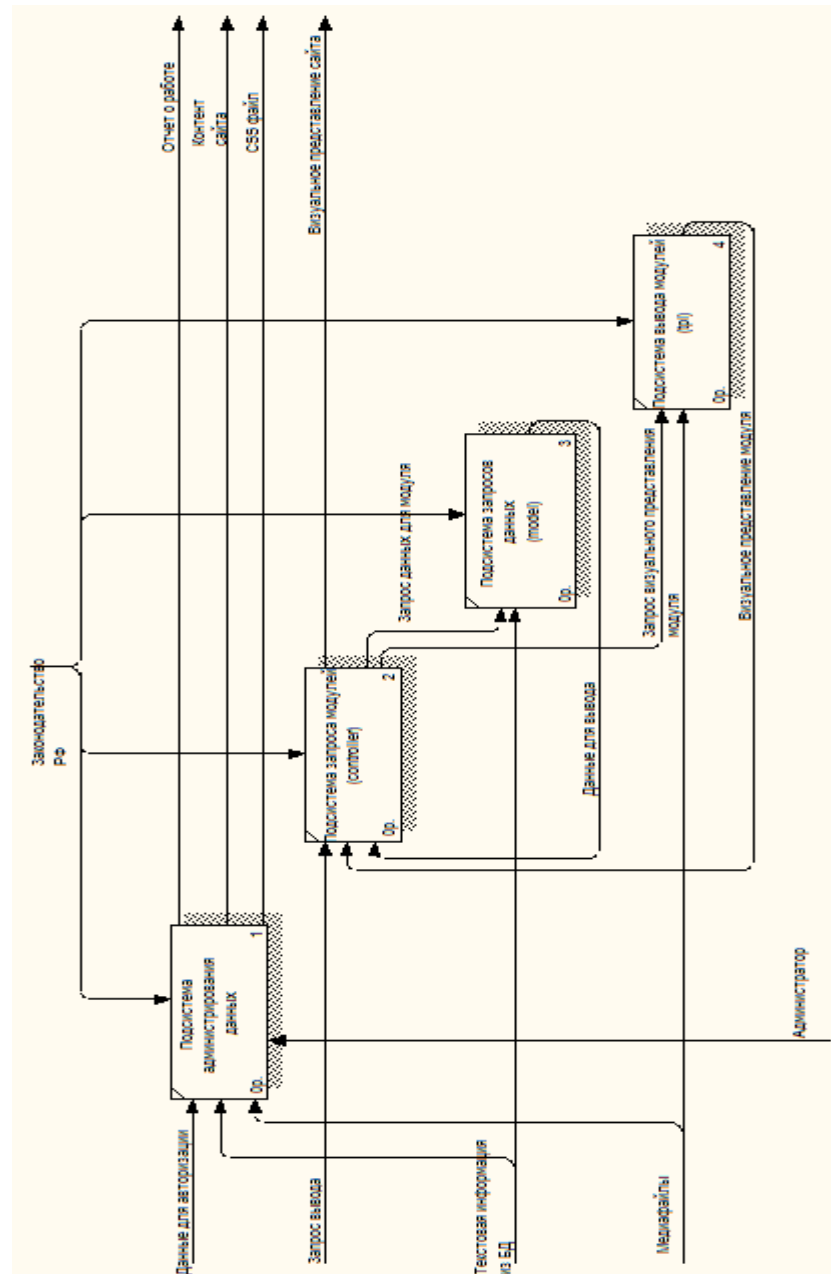


Рисунок В.2 – Декомпозиция IDEF0 диаграммы

Изм.	Лист	№ докум.	Подп.	Дата	

# ПРИЛОЖЕНИЕ Г

## Диаграмма классов CMS-системы

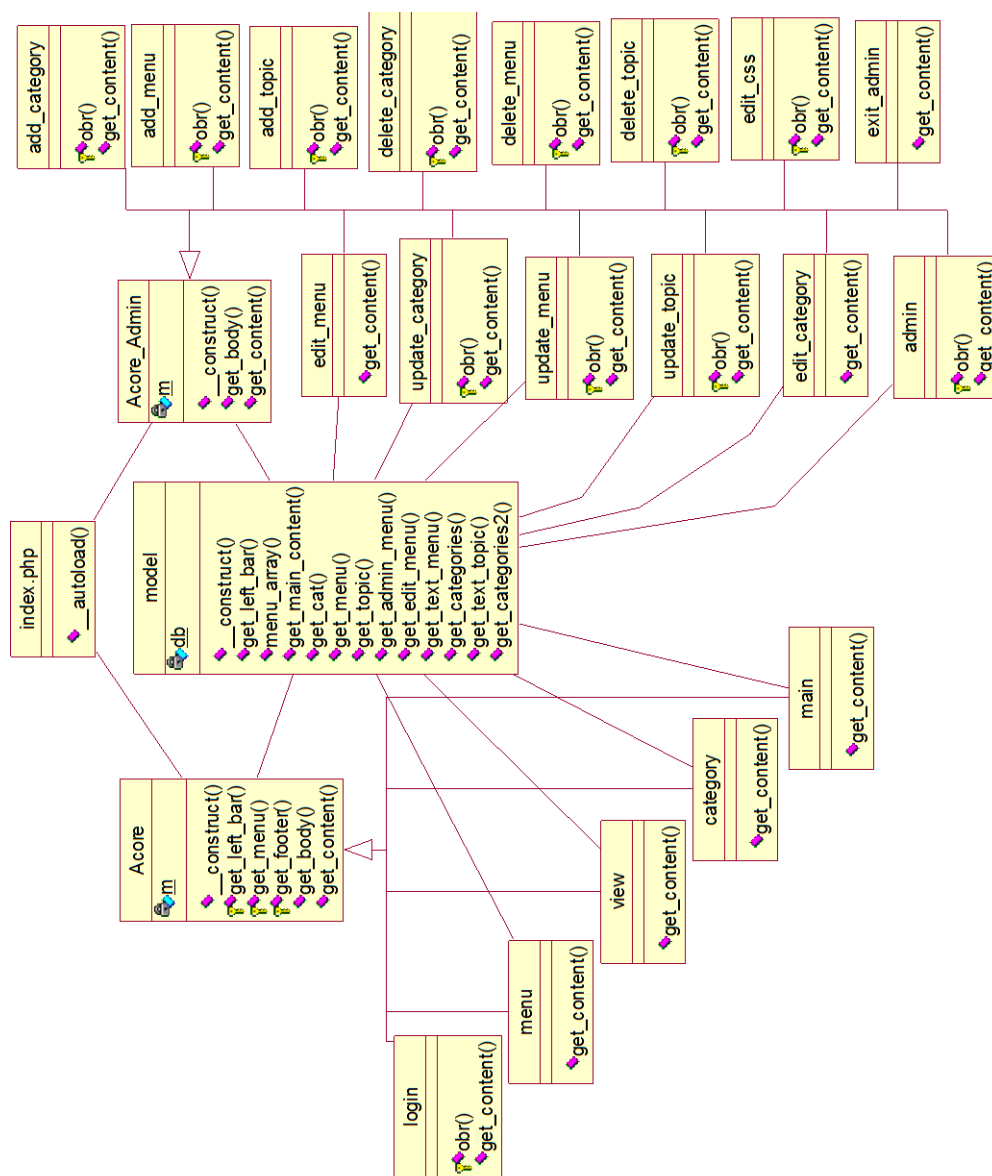


Рисунок Г.1 – Диаграмма классов CMS-системы