МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема: «Автоматизированная система управления рабочим временем»

Выполнил:

студент гр. ИВТ-16 Серышева Ирина Михайловна

Руководитель ВКР:

к.т.н., доцент кафедры ИВТ и ПМ­ Машкин Владимир Анатольевич

Чита

2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе бакалавра

по направлению подготовки 09.03.01 Информатика и вычислительная техника

профиль Вычислительные машины, комплексы, системы и сети

на тему «Автоматизированная система управления рабочим временем»

Выполнил студент группы ИВТ-16 Серышева Ирина Михайловна

Консультанты:

а) Экономическая часть: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преподаватель кафедры ЭиБУ, Т.И. Кашурникова

б) Безопасность и экологичность проекта: \_\_\_\_\_\_\_\_\_\_ доцент кафедры ВХЭиПБ, к.т.н., И.А. Бондарь

в) Специальная часть: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доцент кафедры ИВТ и ПМ, к.т.н., В.А. Машкин

Нормоконтроль: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доцент кафедры ИВТ и ПМ, к.ф.-м.н., И.Ю.Батухтина

Руководитель работы: доцент кафедры ИВТ и ПМ, к.т.н., Машкин Владимир Анатольевич

Допускаю к защите:

Зав. кафедрой ИВТ и ПМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Валова

15 июня 2020 г.

Чита

2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

УТВЕРЖДАЮ

Зав. кафедрой ИВТ и ПМ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.В. Валова

11 мая 2020 г.

ЗАДАНИЕ

на выпускную квалификационную работу

Студенту Серышевой Ирине Михайловне

по направлению 09.03.01 Информатика и вычислительная техника

профиль Вычислительные машины, комплексы, системы и сети

1 Тема выпускной квалификационной работы «Автоматизированная система управления рабочим временем»

Утверждена приказом (распоряжением) ректора по университету от 20.12.2019 № 1003-с

2 Срок подачи студентом законченной работы 15 июня 2020 г.

3 Исходные данные к работе:

– документация по языку PHP;

– стандарт ECMA-262.

4 Перечень подлежащих разработке в выпускной квалификационной работе вопросов:

a) специальная часть;

б) экономическая часть;

в) безопасность и экологичность проекта.

5 Перечень графического материала (если имеется):-

6 Консультанты по выпускной квалификационной работе (с указанием, относящихся к ним разделов):

а) экономическая часть: Кашурникова Тина Иннокентьевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

б) безопасность и экологичность проекта: Бондарь Ирина Алексеевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

в) нормоконтроль: Батухтина Ирина Юрьевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

г) специальная часть: Машкин Владимир Анатольевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания 11 мая 2020 г.

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Машкин

(подпись, расшифровка подписи)

Задание принял к исполнению

11 мая 2020 г.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Ирина Михайловна Серышева /

(имя, отчество, фамилия)

Календарный план

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Наименование раздела выпускной квалификационной работы | Неделя | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | Теоретическая часть |  |  |  |  |  |  |  |  |
| 2 | Специальная часть |  |  |  |  |  |  |  |  |
| 3 | Экономическая часть |  |  |  |  |  |  |  |  |
| 4 | Безопасность и экологичность проекта |  |  |  |  |  |  |  |  |
| 5 | Защита выпускной квалификационной работы |  |  |  |  |  |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / В.А. Машкин /

(подпись, расшифровка подписи)

РЕФЕРАТ

Пояснительная записка – 54 с, 15 рис., 5 таб., 7 источников

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, WEB-САЙТ, PHP, MYSQL, JAVASCRIPT, MVC ПАТТЕРН

Данная работа была выполнена на базе ООО «Психотерапевтический центр «Кодар». На момент обращения сайт организации нуждался в реконструкции.

Была проведена работа по проектированию нового дизайна, верстка новых страниц, разработано приложения для web-сервера, добавлена панель управления для администратора.

В результате, сайт стал динамическим, получил связь с базой данных, и стал доступным для самостоятельного управления организацией.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 8](#_Toc43742504)

[1 Постановка и анализ задачи 9](#_Toc43742505)

[1.1 Описание предметной области 9](#_Toc43742506)

[1.2 Анализ существующих систем 10](#_Toc43742507)

[1.3 Постановка задачи 11](#_Toc43742508)

[1.4 Средства реализации 12](#_Toc43742509)

[2 Анализ данных 14](#_Toc43742510)

[2.1 Входные данные 14](#_Toc43742511)

[2.2 Промежуточные данные 14](#_Toc43742512)

[2.3 Выходные данные 15](#_Toc43742513)

[3 Программная реализация 16](#_Toc43742514)

[4 Тестирование 23](#_Toc43742515)

[5 Документирование 25](#_Toc43742516)

[5.1 Техническое задание 25](#_Toc43742517)

[5.2 Руководство пользователя 27](#_Toc43742518)

[6 Экономическая часть 29](#_Toc43742519)

[6.1 Обоснование целесообразности разработки 29](#_Toc43742520)

[6.2 Определение трудоемкости разработки программного обеспечения 30](#_Toc43742521)

[6.3 Определение стоимости программного обеспечения 33](#_Toc43742522)

[6.4 Определение эффективности программного обеспечения 35](#_Toc43742523)

[7 Охрана труда 36](#_Toc43742524)

[7.1 Общие положения охраны труда 36](#_Toc43742525)

[7.2 Требования к ПЭВМ 36](#_Toc43742526)

[7.3 Требования к помещениям для эксплуатации ПЭВМ 39](#_Toc43742527)

[7.4 Требования к шуму и вибрации в помещениях с ПЭВМ 40](#_Toc43742528)

[7.5 Требования к освещению помещений и рабочих мест с ПЭВМ 41](#_Toc43742529)

[7.6 Требования к организации и оборудованию рабочих мест 43](#_Toc43742530)

[7.7 Требования к организации медицинского обслуживания 45](#_Toc43742531)

[ЗАКЛЮЧЕНИЕ 46](#_Toc43742532)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 47](#_Toc43742533)

[ПРИЛОЖЕНИЯ 48](#_Toc43742534)

ВВЕДЕНИЕ

На сегодняшний день автоматизация различных процессов в бизнесе стала обычным явлением. Становится сложно представить себе складской или бухгалтерский учет без применения специализированного программного обеспечения. Однако взаимоотношения с клиентами, особенно в малом и среднем бизнесе, редко ведутся с использованием средств автоматизации.

Автоматизированная система управления (АСУ) представляет собой сочетание комплекса программно-аппаратных средств и персонала, которые предназначены для управления различными процессами в масштабе технологического процесса, производства, предприятия.

В данной работе планируется создать систему в виде веб-приложения, включающую в себя в том числе сайт организации, позволяющую вести графики приема клиентов для сотрудников в электронном виде, осуществлять запись клиентов через сайт.

Наличие сайта значительно повышает имидж компании в глазах потенциальных клиентов и партнеров. Кроме того, сайт становиться удобным средством информирования о последних новостях, событиях, услугах, графики работы и прочее, в связи с чем отпадает необходимость в лишних звонках и визитах. Должным образом реализованный и наполненный сайт позволяет легко узнать о специфике работы предприятия, оказываемых услугах и другие подробности.

# 1 Постановка и анализ задачи

## 1.1 Описание предметной области

Сегодня практически ни одна организация не обходится без наличия собственного сайта. Наличие сайта позволяет потенциальным и действительным клиентам быстро и эффективно получать актуальную информацию в любое время суток без необходимости совершать дополнительные звонки и визиты.

Организация в свою очередь получает возможность оперативно доносить всю необходимую информацию до своих клиентов, а также, возможно, предоставлять некоторые услуги в режиме онлайн.

Многие организации также предпочитают вести страницы в социальных сетях, однако наличие сайта значительно повышает статус компании. Кроме того, сайт является более гибким инструментом для нужд компании, чем ведение аккаунтов в социальной сети, ограниченное возможностями социальной сети.

К сожалению, далеко не все сайты отвечают потребностям организации и ее потенциальных клиентов. Многие из них давно устарели, наполнены не актуальной информацией, представляют из себя статичные страницы, а организация-пользователь не имеет возможности самостоятельно вносить изменения и обновлять информацию без помощи разработчика.

Данная работы выполнена на базе ООО «Психотерапевтичский центр «Кодар».

Психотерапевтический центр - это учреждение психотерапевтической службы, предназначенное для организации психотерапевтической помощи, проведения консультативно-диагностических мероприятий в рамках психотерапевтической службы, организации и оказания специализированного психотерапевтического лечения.

Центр занимается оказанием амбулаторной помощи обращающимся пациентам. Проводятся индивидуальные и групповые консультации, тренинги и мастер-классы. Запись на прием как правило осуществляется по звонку администратору, либо напрямую у сотрудника центра.

До разработки сайт центра находился в заброшенном состоянии: многие страницы не работали, либо отображались не корректно, на сайте находилась не актуальная информация. К тому же, в силу того, что сайт представлял собой набор статичных веб-страниц и не имел связи с базой данных, администрация центра не имела возможности самостоятельно управлять содержимым сайта и своевременно обновлять информацию. Посетители сайта постоянно натыкались на устаревшую информацию или некорректную работу сайта.

Как следствие, основной целью стало создание динамического веб- сайта вместе с панелью управления сайтом, предоставляющего возможность администрации самостоятельно управлять содержимым сайта, а также возможностью дальнейшей настройки и создания дополнительных функций по согласованию с разработчиком.

## 1.2 Анализ существующих систем

Одним из наиболее быстрых и дешевых способов создания сайта является использование конструкторов, однако такой подход имеет существенные недостатки, такие как: ограничения шаблона, громоздкие html-страницы, и как следствие невозможность вручную редактировать код, отсутствие возможности дальнейшего расширения.

Посадка сайта на такие системы управления контентом как Wordpress или Joomla также не лишена некоторых из вышеперечисленных недостатков. К тому же, для них характерны такие черты как низка безопасность, высокая нагрузка на сервер и однотипность.

Как следствие, наиболее оптимальным решением в данной ситуации является разработка собственной панели управления сайтом с набором только необходимых функций, что в дальнейшем при необходимости позволит расширять этот набор.

## 1.3 Постановка задачи

Наиболее востребованной частью системы, по запросу организации на базе которой выполнялась работа, является сайт центра. Как следствие, целью работы является разработка web-приложения, обеспечивающего возможность администрации центра самостоятельно и оперативно предоставлять клиентам центра всю актуальную информацию, также вести график записей к специалистам, а клиентам записываться самостоятельно записываться на прием.

На сайте предоставлены 2 режима доступа: гость и администратор.

Режим гостя не требует авторизации и действует с момента открытия пользователям сайта. Диаграмма вариантов использования представлена на рисунке 1.

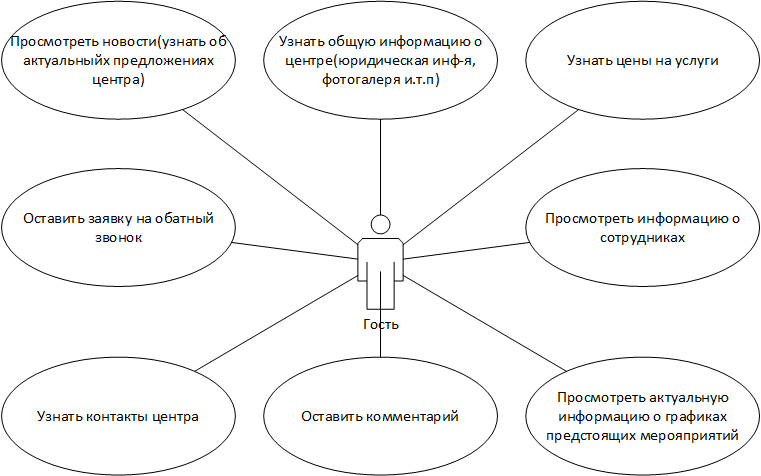


Рисунок 1 – Диаграмма вариантов использования для гостя

На сайте пользователь может получить всю актуальную и интересующую его информацию: предстоящие мероприятия, сотрудники центра, цены на услуги, контактную информацию, а также оставить заявку. Что бы заказать обратный звонок.

Режим администратора предполагает прохождение авторизации, после чего администратор получает возможность редактировать и пополнять информацию на сайте. Диаграмма вариантов использования предоставлена на рисунке 2.

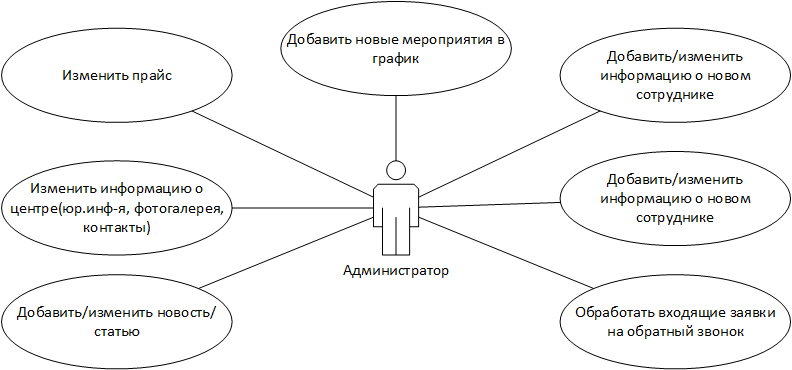


Рисунок 2 – Диаграмма вариантов использования для администратора

Таким образом были поставлены следующие задачи:

– разработка web-сайта согласно требованиям, описанным в разделе техническое задание;

– разработка панели управления сайтом обеспечивающую возможность редактировать наполнение сайта.

## 1.4 Средства реализации

Для разработки дизайна сайта был использован векторный графический онлайн-редактор Figma, ключевыми особенностями которого являются его облачность и кроссплатформенность.

В качестве средств разработки клиентской части приложения использован стандартный для этой задачи набор средств: язык гипертекстовой разметки HTML, каскадный таблицы стилей CSS и скриптовый язык программирования JavaScript c дополнительным использованием широко известной библиотеки jQuery, которая позволяет выполнять те же задачи, что с использованием чистого JavaScript, при этом значительно сокращая количество кода, притом автоматически решая проблему кроссбраузерности.

В качестве языка программирования для серверной части сайта выбран скриптовый язык PHP, главным преимуществом которого является его высокая популярность в применении для создания динамических веб-сайтов и как следствие, тот факт, что PHP поддерживается большинством хостинг-провайдеров. Кроме того, в состав языка PHP включена библиотека PDO (PHP Data Object), которая позволяет работать с базой данных независимо от используемой СУБД.

В качестве системы управления базой данных используется MySQL, являющаяся на сегодняшний день лидером среди прочих СУБД, а также поддерживающая удобные средства для визуального проектирования баз данных таких как MySQL Workbench, включающий в себя такие функции как проектирование, моделирование, создание и эксплуатацию базы данных.

# 2 Анализ данных

Данные, используемые системой делятся на 3 группы:

– входные данные – данные поступающие от администратора в систему;

– промежуточные данные – данные, используемые системой во время работы;

– выходные данные – выводимая системой информация.

## 2.1 Входные данные

В качестве входной информации в разрабатываемом приложении выступают следующие сведения:

– информация для авторизации: логин и пароль;

– новости и статьи, включающие: заголовок, текст, изображение;

– информация о сотрудниках: фото, ФИО, образование, должность и прочая дополнительная информация;

– контактная информация;

– изображения для фотогалереи;

– заявки гостей;

– комментарии гостей;

– прочая информация о центре.

## 2.2 Промежуточные данные

Все промежуточные данные содержаться в базе данных. ER-диаграмма (диаграмма сущность-связь) содержится в приложении A.

В приложении Б.1 представлено описание каждой таблицы базы данных, а приложения Б.2 – Б.10 содержат описание каждой отдельной таблицы.

## 2.3 Выходные данные

Выходными данными являются web-страницы, генерируемые приложением на сервере.

# 3 Программная реализация

В ходе работы были разработаны интерфейсы для администратора и гостя. Главные страницы представлены на рисунках ниже.

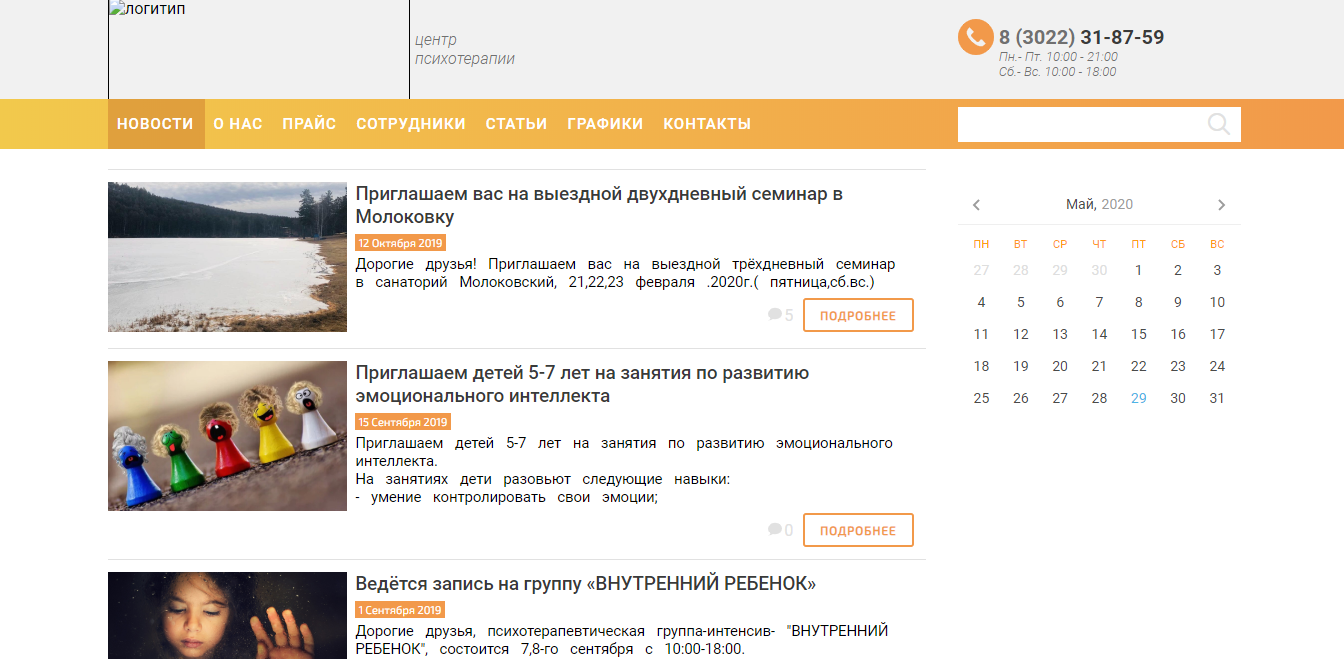


Рисунок 3 – Главная страница сайта (для гостя)

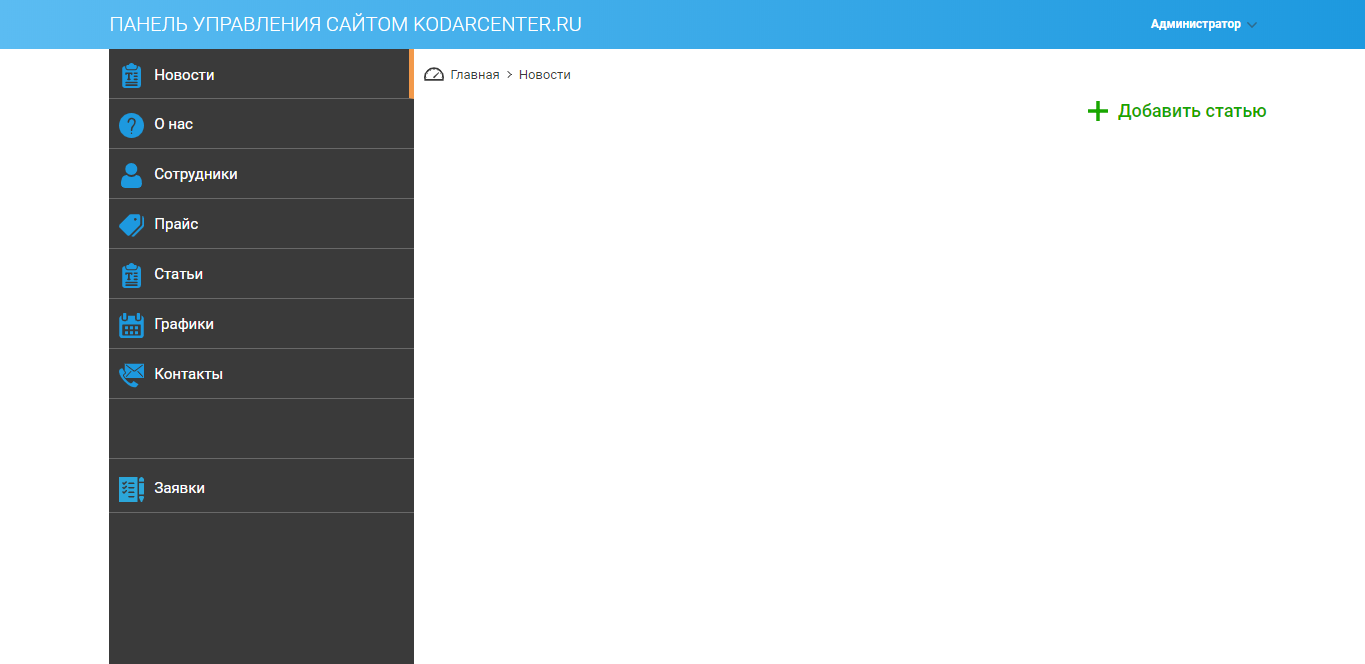


Рисунок 4 – Главная страница для администратора

Серверная часть приложения реализована с использованием MVC (Model View Controller) - паттерна. Стандартная схема этого паттерна представлена на рисунке 5.

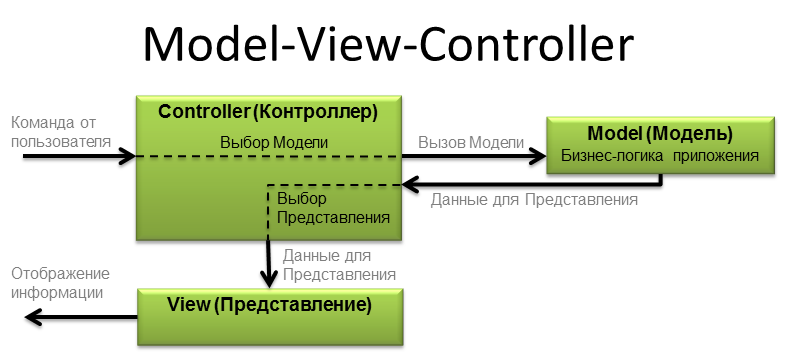


Рисунок 5 – Схема MVC-паттерна

Это архитектура построения приложения, в рамках которой оно разделяется на три компонента:

Модель (Model) – предоставляет данные для Представлений в ответ на запросы Контроллера, содержит бизнес-логику приложения.

Представление (View) – отвечает за пользовательский интерфейс, отображает данные, полученные от Модели.

Контроллер (Controller) – обрабатывает команды пользователя, определяет Модели для работы и связывает ее с Представлением.

Бизнес-логика, расположенная в Модели, включает все правила и алгоритмы, связанные с предметной областью решаемой задачи. Проще говоря – это ядро создаваемого приложения, которое может быть как банковским клиентом, так и онлайн игрой или блогом.

Рассматриваемая архитектура подразумевает, что изменения в любом из компонентов оказывают минимальные воздействия на остальные части.

Несколько упрощая, работу MVC приложения можно описать следующим образом:

– команда (уведомление о нажатии кнопки, запроса адреса сайта и т. д.) передается Контроллеру;

– Контроллер, исходя из полученных данных, определяет и вызывает Модель;

– Модель, на основе заложенной в нее бизнес-логики, формирует набор данных;

– Контролер выбирает Представление и связывает его с данными (Моделью);

– Представление отображает данные пользователю.

Диаграмма классов серверной части приложения имеет следующий вид:

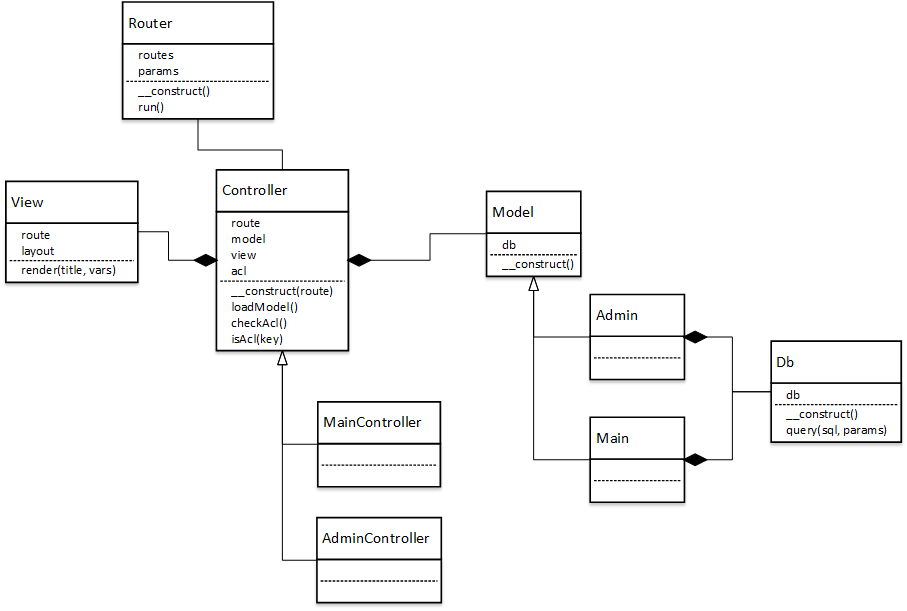


Рисунок 6 – Диаграмма классов серверной части приложения, реализующая MVC-паттерн

Все запросы к сайту (кроме обращений к файлами директориям) перехватываются и перенаправляются на страницу index.php, согласно правилам описанным в файле .htaccess (конфигурационный файл сервера Apache).

Содержимое страницы index.php

session\_start();

$router = new *Router*();

$router->run();

В начале инициализируется сессия, затем создается объект класса Router(Маршрутизатор). В процессе создания роутера, в него загружаются данные из файла о существующих маршрутах.

Далее вызывается метод run(), в котором роутер сравнивает пришедший к сайту запрос сопоставляя вызываемый адрес с доступными маршрутами, в случае успеха роутер вызывает соответствующий контроллер, в случае неудачи возвращается сообщение об ошибке.

Всего существует 2 вида контроллера:

– MainСontroller – отвечает за действия гостей на сайте;

– AdminController – отвечает за действия администратора в панели управления сайтом.

Оба контроллера наследуются от общего базового класса Controller, имеющего общие для обоих контроллеров поля и методы.

Базовый контроллер отвечает за загрузку соответствующей модели и проверку прав доступа, более специфические действия выполняются соответствующим классом-наследником.

Получив управление контроллер, сначала проверяет права доступа к запрашиваемой страницы. Например, для контроллера MainController – все страницы доступны для всех пользователей, а для контроллера AdminController изначально доступна только страница авторизации, остальные страницы не будут доступны пока авторизация не пройдена.

Контроллеры вызывают соответствующую им модель: Main для MainController и Admin для AdminController.

Базовый класс Model содержит методы общие для обеих моделей. Это преимущественно методы отвечающие, за извлечение данных из базы данных.

Модели содержат в себе бизнес-логику приложения, преимущественно отвечают за обработку данных. Модели в свою очередь включают в себя объект класса Db, содержащий в себе методы для упрощения работы с базой данных.

После обработки данных моделью и получения результата, контроллеры создает объект класса View, которому передаются результаты работы модели – данные, которые необходимо выдать пользователю. У объекта класса View вызывается метод render(), который из полученных данных и соответствующего шаблона генерирует веб-страницу, которая возвращается пользователю.

Общая структура проекта показана на рисунке 7.

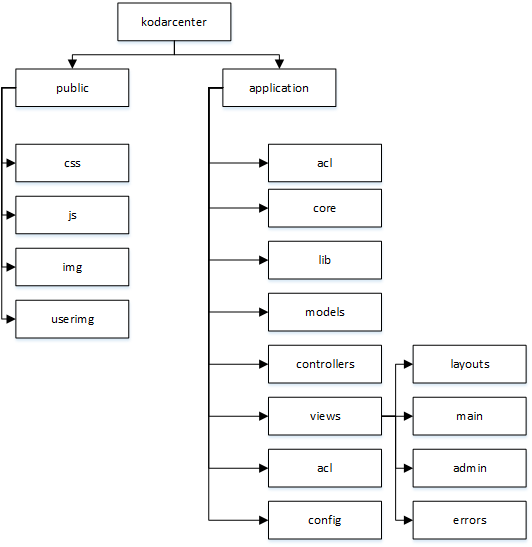


Рисунок 7 – Структура проекта

Проект делится на 2 основных папки:

Public – содержит всю клиентскую часть сайта: css – файлы, js – скрипты, дополнительные изображения (иконки, фоны и т. п.), а также изображения загруженные пользователем(администратором).

Application – серверная часть приложения.

Серверная часть состоит из папок:

– core – ядро приложения. Классы Router, Controller, View, Model;

– lib – вспомогательные классы, такие как Db;

– models – классы моделей;

– controllers – классы контроллеров;

– acl – файлы с информацией о правах доступа;

– config – конфигурационные файлы: информация о маршрутах, данные для подключения к базе данных;

– view – шаблоны для представления данных.

Папка View в свою очередь делится на:

– layouts – базовые шаблоны для гостя и для администратора;

– errors – страницы для вывода информации об ошибках;

– admin – компоненты для представления данных для администратора;

– main – компоненты для представления данных для гостя.

Многие запросы к серверу, не требующие перезагрузки страницы были реализованы с помощью технологии AJAX (аббревиатура от Asynchronous JavaScript and XML) - технология взаимодействия браузера с сервером в асинхронном режиме, т.е. без перезагрузки страницы.

Схема работы AJAX представлена на рисунке ниже:

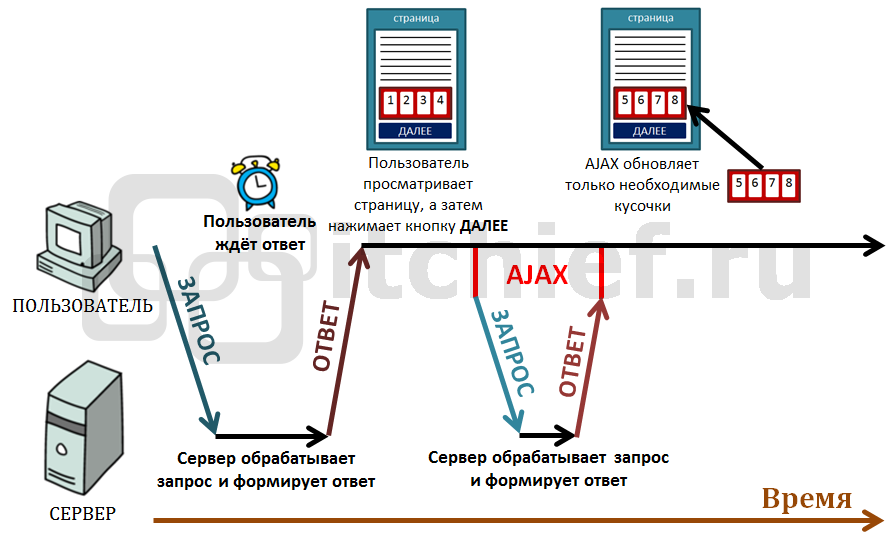


Рисунок 8 - Схема работы AJAX

Для реализации AJAX-запросов были использованы функции библиотеки jQuery, т.к. они позволяют упростить такие процессы как создание, отправка и обработка ответа AJAX-запроса, а также решают проблему совместимости с различными браузерами.

Ниже описан пример одного из таких запросов:

$('.del-button').on( 'click', *function*(){ //Обработчик на все кнопки удаления

*let* confirmation = confirm("Удалить выбранный элемент?"); //Подтверждение

        if (confirmation){ //Если подтверждено

*let* jqXHR = $.ajax({

                url:this.dataset.url, //Из какого раздела удаляем

                context:this, //Контекстом для callbacko-в, передаваемых в функции done и fail объекта jqXHR будет текущая кнопка

                data:'id='+this.dataset.id,//Идентификатор удаляемой статьи/новости/сотрудника

                method:'post'

            }).

            // Функция, выполнится в случае успешного запроса

            done(*function*(){

                if (jqXHR.responseText=='Success'){

                    $(this).parent().parent().parent().remove();//Удаление элемента из DOM, которому принадлежала кнопка

                    alert('Успешно!');

                }

                else{

                    alert('Не удалось выполнить запрос!')

                }

            })

            // Функция выполнится, если запрос не удался

            .fail(*function*(){

                alert('Не удалось выполнить запрос!')

            })

        }

    });

# 4 Тестирование

Главной целью тестирование была проверка устойчивости системы к вводу пользователем вредоносных команд способных нанести вред данным в базе, таким как SQL-инъекции.

Для устойчивости системы к подобным атакам было решено использовать обращение к базе данных с помощью подготовленных запросов.

В языке библиотеке PDO языка PHP предоставлены методы для осуществления подготовленных запросов. Это PDO::prepare() для подготовки запроса, который в случае успешного выполнения возвращает объект PDOStatement готовый к выполнению запроса и PDOStatement::execute() для выполнения запроса.

Вызов PDO::prepare() и PDOStatement::execute()помогает избежать SQL-инъекций, так как нет необходимости экранировать передаваемые параметры.

Ниже приведен код метода класса Db, используемого для работы с базой данных, который использует методы библиотеки PDO для создания и выполнения подготовленных запросов.

    public *function* query($sql, $params=[]){

        $stmt = $this->db->prepare($sql);

        if (!empty($params)){

            foreach($params as $key => $value){

                $stmt->bindValue(':'.$key, $value);

            }

        }

        $stmt->execute();

        return $stmt;

    }

Для тестирования в форму для заказа обратного звонка был введен следующий набор:

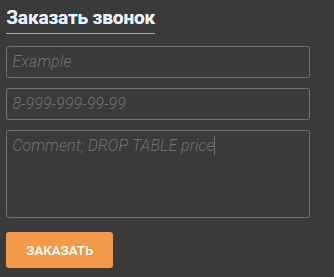


Рисунок 9 - Пример данных в форме для осуществления SQL-инъекции

Результат: данные были вставлены в таблицу, SQL-команда после точки с запятой не была выполнена. Таблица price не была удалена.

# 5 Документирование

## 5.1 Техническое задание

## 5.1.1 Назначение разработки

Сайт психотерапевтического центра «Кодар» предназначен для предоставления ознакомительной информации о центре.

## 5.1.2 Требования к функциональным характеристикам

Для гостя на сайте должны быть предоставленные следующие разделы:

– главная страница: новости о предстоящих событиях, группах, интенсивах, и т.п;

– прайс-лист: информация о ценах на услуги центра;

– сотрудники: информация о действующих сотрудниках$

– график групп: актуальная информация о группах;

– статьи: тематические статьи, информация о методиках, и.т.п;

– контакты: адрес, телефон, группа «В контакте» e-mail, схема проезда, карта;

– о центре: юридическая информация, история центра.

Также гости должны иметь возможность:

1. оставлять комментарии к новостям и статьям на сайте;
2. скачивать прайс-лист в pdf-формате;
3. заказывать обратный звонок.

Администратор должен иметь возможность:

1) авторизироваться;

2) редактировать информацию во всех разделах сайта, добавлять новую;

3) просматривать и обрабатывать поступившие заявки.

## 5.1.3 Требование к надёжности

В случае возникновения ошибок приложение должно возвращать страницу отображения ошибки с кодом ошибки (например 404).

В случае несерьезных ошибок в работе сайта, пользователю должно выводится сообщение о том, что именно необходимо исправить.

## 5.1.4 Требования к составу и параметрам технических средств

Минимальные требования для корректной работы сайта: необходимо иметь персональный компьютер любой операционной системой, с оперативной памятью 512 Мб, также свободного места на диске должно быть 350 Мб, процессор с частотой 500 Мгц.

Оптимальные требования для корректной работы сайта: необходимо иметь персональный компьютер любой операционной системой, с оперативной памятью не менее 512 Мб, также свободного места на диске должно быть не менее 700 Мб, процессор с частотой 600 Мгц.

## 5.1.5 Требования к программной документации

Программная документация должна содержать руководство пользователя.

## 5.1.6 Требования к информационной и программной совместимости

Хостинг-провайдер должен обеспечивать поддержку языка PHP. На устройстве клиента должен быть установлен браузер.

5.2 Руководство пользователя

Для управления сайтом администратору необходимо пройти авторизацию введя в форму входа логин и пароль. Форма входа представлена на рисунке ниже.

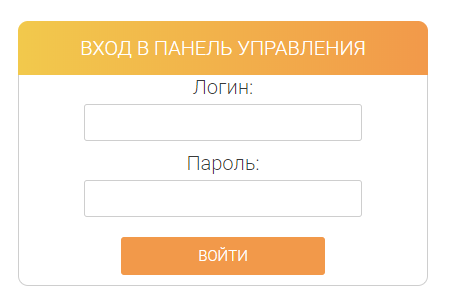


Рисунок 10 – Форма входа для администратора

Добавление и редактирование новостей/статей.

Для добавления новости или редактирование уже существующей, добавления или редактирования информации о сотрудниках, а также информации о центре необходимо перейти в соответствующий раздел в меню (рисунок 11).

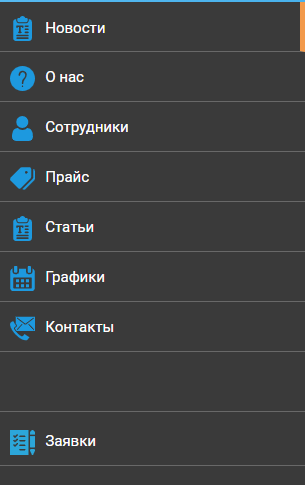


Рисунок 11 – Меню панели управления сайтом

По щелчку на кнопке 1 откроется форма для добавления новой записи, а на кнопке 2 – для редактирования уже существующей. Кнопка 3 предназначена для удаления записи после подтверждения.

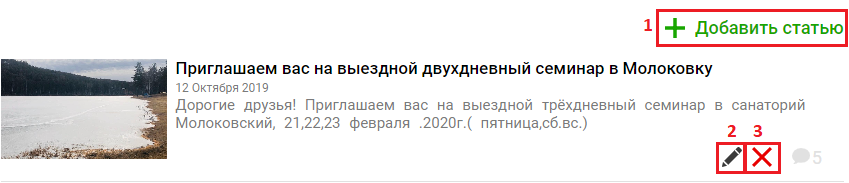


Рисунок 12 – Кнопки добавления/удаления/редактирования новостей

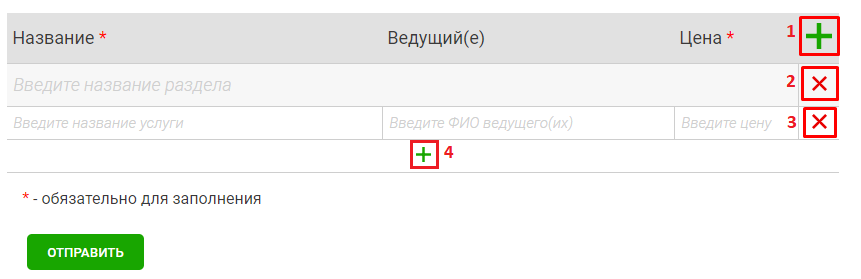
Для редактирования прайса, после перехода в раздел Прайс, кнопка 1 добавляет новый раздел, кнопка 2 удаляет выбранный раздел, кнопка 3 удаляет выбранную запись из раздела, а кнопка 4 добавляет новую запись в раздел. Для конструктора графиков в разделе Графики инструкция аналогична.

Рисунок 13 – Форма редактирования прайса

Заявки на обратный звонок можно просмотреть в разделе Заявки. Не обработанным заявкам можно изменить статус обработав их нажатием на выделенные кнопки.

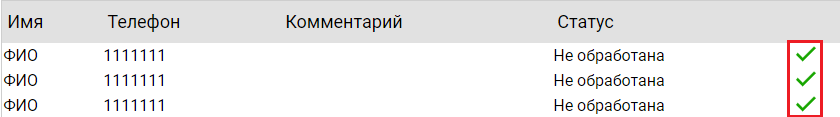


Рисунок 14 – Форма обработки и просмотра заявок

# 6 Экономическая часть

## 6.1 Обоснование целесообразности разработки

В век информационных технологий внедрение информационной системы необходимо каждой организации. У ООО «Психотерапевтический центр «Кодар» обнаружен существенный недостаток – устаревший сайт с неактуальной информацией, а также без возможности управления содержимым данного сайта.

Сайт для организации – лучший инструмент, используемый для таких целей как реклама, поиск новых клиентов и партнеров, предоставление подробной информации об услугах и поддержка имиджа организации.

Действующий сайт центра не удовлетворяет вышеописанным требованиям и требованиям самой организации, следовательно потенциальные клиенты не информированы должным образом о предоставляемых компанией услугах и ценах на них, о способах связи с организацией, ее местоположении и так далее. Разработанное программное решение необходимо для устранения данных недочетов.

Цель написания экономической части – определение трудоёмкости разработки, стоимости и эффективности использования программного продукта. Стоимость программного обеспечения может рассматриваться с двух точек зрения:

– с точки зрения формирования объектов нематериальных активов, как исключительное право для использования непосредственно на предприятии;

– с точки зрения формирования объектов нематериальных активов, как исключительное право для непосредственного коммерческого использования, продажи и продвижения программного обеспечения на рынке.

Разрабатываемое программное решение является нематериальным активом, на который распространяется исключительное нематериальное право для использования непосредственно на предприятии. Коммерческое использование не предусмотрено.

В данном разделе будет определена экономическая эффективность расчетом стоимости программного продукта и, затем, расчетом экономической эффективности использования.

## 6.2 Определение трудоемкости разработки программного обеспечения

Трудоемкость — это затраты рабочего времени на производство единицы продукции. Трудоёмкость разработки программного обеспечения рассчитывается на основе типовых норм в определённой сфере деятельности. Для разработки компьютерных информационных технологий разработаны типовые нормы времени. Они предназначены для разработки нормирования труда специалистов занятых созданием программного установления для ЭВМ, установление численности исполнителей, а так же обоснование трудности разработки проекта.

Для расчета трудозатрат программиста при разработке данного программного решения необходимо определить степень новизны. У разрабатываемых комплексов задач существует четыре степени новизны. Степень А – это разработка комплекса задач, включающих применение совершенно новых методов разработки, а также осуществление научно-исследовательских работ. Степень Б – это разработка принципиально новых проектных решений, задач, методов и систем, не имеющих аналогов. Степень В – это разработка проекта с использованием уже существующих аналогичных проектных решений с условием их изменения. Степень Г – привязка готовых проектных решений к собственному проекту.

Разрабатываемое программное решение соответствует степени новизны В.

Разработка программного продукта включает пять стадий:

1. техническое задание (ТЗ). На этой стадии заказчик формирует требования к ПП и консультирует разработчиков по данному вопросу. Далее идет обоснование принципиальной возможности решения данной задачи, разрабатывается концепция, определяются и согласовываются сроки разработки ПП;
2. эскизный проект (ЭП). На данной стадии прорабатывается ТЗ и по итогам выбирается и разрабатывается математическая модель, алгоритм разработки ПП;
3. технический проект (ТП). На стадии ТП разрабатывается программная документация и определяется конфигурация технических средств для реализации ПП;
4. рабочий проект (РП) является наиболее трудоемким этапом, на котором происходит непосредственная разработка ПП, его испытание и отладка. Результатом данной стадии является готовый ПП с руководством пользователя и рабочей документацией;
5. внедрение (ВН). На этой стадии проводится проверка правильности работы ПП на практике в процессе подготовки соответствующей документации, а затем ПП сдается в эксплуатацию.

На стадии «Техническое задание» расход времени на выполнение задачи обозначается за Т1. В свою очередь Т1 имеет значение равное 24 чел./дн., которое определяется в соответствии с таблицей 5.4 [1].

На стадии «Эскизный проект» расход времени на выполнение задачи обозначается за Т2. В свою очередь Т2 имеет значение равное 67 чел./дн., которое определяется в соответствии с таблицей 5.5 [1].

Для остальных этапов («Технический проект», «Рабочий проект» и «Внедрение») предварительные нормы затрат времени определяются по формуле:

где r1, r2, r3 – коэффициенты, приведенные в таблицах 5.6 – 5.8;

Ф1 – количество макетов входной информации;

Ф2 – количество разновидностей форм выходной информации.

Коэффициенты r1, r2, r3 принимают следующие значения в соответствии с таблицей 1.

Таблица 6.1– Значения коэффициентов r1, r2, r3

|  |  |  |  |
| --- | --- | --- | --- |
| Стадия разработки | r1 | r2 | r3 |
| Технический проект (Т3) | 7.79 | 0.63 | 0.19 |
| Рабочий проект (Т4) | 41.69 | 0.41 | 0.42 |
| Внедрение (Т5) | 7.83 | 0.44 | 0.86 |

Так как разработчик комплекса участвовал в подготовке информационного обеспечения, на стадиях «Технический проект» и «Рабочий проект» коэффициенты r1, r2, r3 были увеличены в 1,1 раз (повышающий коэффициент).

Для разрабатываемого проекта Ф1 = 2, а Ф2 = 2.

Ниже представлен расчет норм временных затрат Т3, Т4, Т5.

Т3 = 7,79 ∙ 20,63 ∙ 20,19 = 13,75 (чел./дн.);

Т4 = 41,69 ∙ 20,41 ∙ 20,42 = 74,11 (чел./дн.);

Т5 = 7,83 ∙ 20,44 ∙ 20,86 = 19,28 (чел./дн.).

Тогда общие трудозатраты будут равны:

ТΣ=T1+T2+T3+T4+T5= 24 + 67 + 13,75 + 74,11 + 19,28 = 198,14 (чел./дн.).

Численность исполнителей, необходимая для выполнения работ по созданию платформы вычисляется по формуле:

где Фпл – фонд рабочего времени, планируемый на одного разработчика;

Твр – временные трудозатраты на реализацию программы.

Платформа разрабатывается в течении девяти месяцев с сентября по май, что составляет 182 рабочих дня, соответственно рекомендуемая численность разработчиков проекта составляет:

Ч = 198,14 / 182 = 1,09 ≈ 1 (чел.).

## 6.3 Определение стоимости программного обеспечения

Себестоимость – это стоимостная оценка используемых в процессе производства материальных, трудовых и других затрат, а также затрат на реализацию продукции. Стоимость платформы определена на основе укрупненного метода учета затрат по материальным затратам, расходам на оплату труда, отчислений на социальное страхование, амортизации основных фондов и прочих расходов. Стоимость рассчитывается по формуле:

где МЗ– материальные затраты;

ФЗП – фонд заработной платы разработчика программного продукта;

Ао– амортизационные отчисления;

П – прочие расходы.

Количество рабочих часов, затраченных на разработку данной программной платформы, при учете того, что проект разрабатывался в течение 182 рабочих дней, а длительность рабочего дня составляет 8 часов, получим общее количество рабочих часов:

КЧ = TΣок ∙ 8 = 182 ∙ 8 = 1456 (час.).

Материальные затраты включают в себя:

а) расходные материалы:

* оплата услуг интернет-провайдера за 9 месяцев - 6300р.

б) расходы на электроэнергию.

Определение затрат на электроэнергию ведется из расчета того, что потребление энергии компьютером составляет 0,5 кВт⋅ч, а стоимость 1 кВт/ч равна 4,20 р., тогда затраты на электроэнергию составят:

1456 ∙ 0,5 ∙ 4,20 = 3057,6 (р.).

Всего материальные затраты составляют:

МЗ = 6400 + 3057,6 = 9 457,6(‬р.)

Заработная плата определяется на основе тарифной формы оплаты труда. Согласно Постановлению Министерство Труда Российской Федерации от 06 июня 1996 г. №-32 «Об утверждении разрядов оплаты труда и тарифно - квалификационных характеристик (требований) по общеотраслевым должностям служащих» программист 12 - 13 разряда должен иметь высшее профессиональное образование и стаж работы в должности программиста I категории не менее 3 лет.

В соответствии с единой тарифной сетке по оплате труда работников бюджетных организаций, финансируемых из бюджетов регионального и муниципального уровней с 1 января 2020 г., должностной оклад инженера-программиста 13 разряда составляет 11336 р. Исходя из оклада, производится расчет заработной платы.

Законом Забайкальского края от 26.09.2008 № 39-ЗЗК «О районном коэффициенте и процентной надбавке к заработной плате работников бюджетных организаций» установлен районный коэффициент в размере 20 %; 30 % процентную надбавку за стаж работы в районах Крайнего Севера и приравненных к ним местностях, а также в остальных районах Севера.

Получаем, что месячная заработная плата составляет:

11336\*1,5 = 17004 р.

Так как разработка ведется 9 месяцев, то фонд заработной платы составляет: ФЗП = 17004 \* 9 = 153036 р.

Начиная с 01.01.2018 года, взносы на социальное страхование составляют 30 % от общей заработной платы. Законом установлены тарифы страховых взносов, описанных ниже:

– Пенсионный фонд Российской Федерации – 22,00 %;

– Фонд социального страхования Российской Федерации – 2,9 %;

– Федеральный фонд обязательного медицинского страхования – 5,1 %.

Таким образом, отчисления составили:

153036 ∙ 0,3 = 45910,8‬ (р.).

Основное средство, подлежащее амортизации при разработке программной системы – это компьютер. Первоначальная стоимость одного компьютера 25000 р.

Вычислительная техника входит в амортизационную группу 2. По установленным нормам, постановление Правительства Российской Федерации от 18.11.2006 № 697, время службы компьютера берется от 2 до 3 лет включительно. Установим норму службы компьютера равную 3 годам. Из этого следует, что амортизация за год составляет 1/3 ∙ 100 % = 33 % от первоначальной стоимости компьютера. Следовательно, амортизация за девять месяцев составит:

A0 = (25000 + 3057,6) ∙ 0,33 ∙ 9 /12 = 6944,26 (р.).

Величина прочих расходов составляет 15 % от основной заработной платы. В прочие расходы входит амортизация нематериальных активов, а также накладные расходы:

П = 17004 ∙ 0,15 ∙ 9 = 22995,4 (р.).

Таким образом, общая стоимость затрат на создание программной системы составляет:

СПО = 9457,6 + 153036 + 45910,8 + 6944,26 + 22995,4 = 238344,06 (р.)

## 6.4 Определение эффективности программного обеспечения

Эффективность является общим экономическим понятием. За эффективность принимается характеристика системы с точки зрения результатов ее функционирования. Использование разработанной системы позволяет управлять содержимым сайта с любого устройства, имеющего браузер и получать выходные данные с сайта в удобном для восприятия виде;

разграничивать доступ различных групп пользователей к управлению.

# 7 Охрана труда

## 7.1 Общие положения охраны труда

Гигиенические требования к персональным электронно-вычислительным машинам (ПЭВМ) и организации работы с ними контролируются санитарно-эпидемиологическими правилами и нормами СанПиН 2.2.2/2.4.1340-03 (далее – санитарные правила) [1]. Настоящие санитарные правила разработаны в соответствии с Федеральным законом «О санитарно-эпидемиологическом благополучии населения» от 30.03.1999 г. № 52-Ф3 и Положении о государственном санитарно-эпидемиологическом нормировании, утвержденным постановлением Правительства Российской Федерации от 24.07.2000 г. №554.

Санитарные правила действуют на всей территории Российской Федерации и устанавливают требования к ПЭВМ и условиям труда. Требования санитарных правил направлены на предотвращение неблагоприятного влияния вредных факторов производства и трудового процесса при работе с ПЭВМ на здоровье человека.

Требования санитарных правил распространяются на: организацию и условия работы с ПЭВМ, персональные и портативные ЭВМ, устройства отображения информации, периферийные устройства и игровые комплексы на базе ПЭВМ.

## 7.2 Требования к ПЭВМ

ПЭВМ должны соответствовать требованиям настоящих санитарных правил. Каждый их тип подлежит обязательной санитарно-эпидемиологической экспертизе. Данная экспертиза должна проводиться в специальных аккредитованных испытательных лабораториях.

Список контролируемых гигиенических параметров опасных и вредных факторов представлен в таблице 7.1.

Таблица 7.1 – Контролируемые гигиенические параметры

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вид продукции | Код ОКП | Контролируемые гигиенические параметры |
| 1 | Персональные и портативные ЭВМ | 40 1300,  40 1350,  40 1370 | Уровни электромагнитных полей (ЭМП), акустического шума, концентрации вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение |
| 2 | Периферийные устройства | 40 3000 | Уровни ЭМП, акустического шума, концентрация вредных веществ в воздухе |
| 3 | Устройства отображения информации | 40 3200 | Уровни ЭМП, визуальные показатели, концентрация вредных веществ в воздухе, мягкое рентгеновское излучение |
| 4 | Игровые автоматы на базе ПЭВМ | 96 8575 | Уровни ЭМП, акустического шума, концентрация вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение |

Звуковое давление – давление дополнительно возникающее при прохождении звуковой волны в среде. Оно является избыточным над средним давлением среды.

Таблица 7.2 – Допустимые значения уровней звукового давления

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Уровни звукового давления в октановых полосах со среднегеометрическими частотами | | | | | | | | | Уровни звука в дБА |
| 31,5  Гц | 63  Гц | 125  Гц | 250  Гц | 500  Гц | 1000  Гц | 2000  Гц | 4000  Гц | 8000  Гц |
| 86 дБ | 71 дБ | 61 дБ | 54 дБ | 49 дБ | 45 дБ | 42 дБ | 40 дБ | 38 дБ | 50 |

Допустимые уровни звукового давления и звука, создаваемые ПЭВМ, не должны превышать значений, указанных в таблице 7.2. Все измерения проводятся на расстоянии 50 см от поверхности оборудования.

Временные допустимые уровни (ВДУ) электромагнитных полей (ЭМП), создаваемые ПЭВМ, не должны превышать значений, указанных в таблице 7.3.

Таблица 7.3 – Допустимые уровни ЭМП

|  |  |  |
| --- | --- | --- |
| Наименование параметров | | ВДУ ЭМП |
| Напряженность электрического поля | в диапазоне частот с 5 Гц до 2 кГц | 25 В/м |
| в диапазоне частот с 2 кГц до 400 кГц | 2,5 В/м |
| Плотность магнитного потока | в диапазоне частот с 5 Гц до 2 кГц | 250 нТл |
| в диапазоне частот с 2 кГц до 400 кГц | 25 нТл |
| Электростатический потенциал экрана видеомонитора | | 500 В |

Допустимые параметры устройств отображения информации указаны в таблице 7.4.

Таблица 7.4 – Допустимые параметры устройств отображения информации

|  |  |  |
| --- | --- | --- |
| № | Параметр | Допустимые значения |
| 1 | Яркость белого поля | Не менее 35 кд/м2 |
| 2 | Неравномерность яркости рабочего поля | Не более ± 20 % |
| 3 | Контрастность (для монохромного режима) | Не менее 3:1 |
| 4 | Временная нестабильность изображения | Не должна фиксироваться |
| 5 | Пространственная нестабильность изображения | Не более 2 · 10-41, где L – проектное расстояние наблюдения, мм |

Для дисплеев на электронно-лучевых трубках частота обновления экрана не должна быть ниже 75 Гц при любом разрешении. Для остальных типов дисплеев частота обновления экрана должна быть не менее 60 Гц.

Концентрация вредных веществ, выделяемых ПЭВМ, не должна превышать предельно допустимую концентрацию, установленную для атмосферного воздуха. Мощность мягкого рентгеновского излучения в любой точке на расстоянии 0,05  м от экрана ПЭВМ и его корпуса не должна превышать 1 мкЗв/ч.

Корпус ПЭВМ должен быть окрашен в мягких и спокойных тонах с диффузным рассеиванием света. ПЭВМ и периферийные устройства должны иметь матовую поверхность с отсутствием деталей способных создавать блики. При этом коэффициент отражения поверхности должен варьироваться от 0,4 до 0,6 условных единиц.

## 7.3 Требования к помещениям для эксплуатации ПЭВМ

Эксплуатация ПЭВМ разрешена в помещениях с естественным и искусственным освещением. В случае отсутствия естественного освещения, использование ПЭВМ возможна только при соответствующем обосновании и наличии положительного заключения санитарно-эпидемиологической экспертизы, которое выдается в установленном порядке.

Уровни естественного и искусственного освещения должны соответствовать действующим требованиям нормативной документации. Эксплуатация ПЭВМ наиболее рекомендована в помещениях с расположением окон на северной или северо-восточной стороне.

Оконные проемы при этом должны быть оборудованы регулируемыми устройствами: жалюзи, занавесы, внешние козырьки, которые позволят полностью закрыть оконные проемы.

На рабочих местах дисплеи следует располагать так, чтобы они были ориентированы боковой стороной к световым проемам, так что бы свет падал с левой стороны от экрана.

Площадь одного рабочего места пользователя ПЭВМ, включающего в себя жидкокристаллический или плазменный дисплей, должна быть не менее 4,5 м2.

Для внутренней отделки интерьера помещения применяются диффузно отражающие материалы с коэффициентом отражения для потолка от 0,7 до 0,8; для стен от 0,5 до 0,6; для пола от 0,3 до 0,5. Полимерные материалы используются при наличии санитарно-эпидемиологического заключения.

Помещения с размещенными в них рабочими местами ПЭВМ, должны быть оборудованы защитным заземлением, удовлетворяющим техническим требованиям по эксплуатации. Не рекомендуется схема размещения рабочих мест ПЭВМ вблизи силовых кабелей, высоковольтных трансформаторов или технологического оборудования, которое способно создать помехи в работе ПЭВМ.

## 7.4 Требования к шуму и вибрации в помещениях с ПЭВМ

Уровни шума на рабочих местах должны не должны превышать предельно допустимых значений, которые установлены действующими санитарно-эпидемиологическими нормативами.

В образовательных и культурно-развлекательных учреждениях для детей и подростков, а также в производственных помещениях, где расположены и эксплуатируются ПЭВМ, уровень шума и вибрации должен быть ниже предельно допустимых значений, установленных в соответствии с существующими санитарно-эпидемиологическими нормативами.

Оборудование, которое при работе имеет высокий уровень шума (печатающие устройства, серверы и т.д.), превышающий допустимые значения, установленные нормативами, должно располагаться вне помещения с рабочим местом ПЭВМ.

## 7.5 Требования к освещению помещений и рабочих мест с ПЭВМ

В помещениях для работы с ПЭВМ искусственное освещение должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, при преимущественной работе с документами, следует применять системы комбинированного освещения. В этом устанавливаются дополнительно светильники местного освещения для зоны работы с документами.

В помещениях для работы с ПЭВМ искусственное освещение должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, при преимущественной работе с документами, следует применять системы комбинированного освещения. В этом устанавливаются дополнительно светильники местного освещения для зоны работы с документами.

Искусственное освещение должно обеспечивать требуемую освещённость на рабочих местах не ниже нормируемых значений: экран – не более 300 лк; клавиатура и рабочий стол от 300 до 500  лк.

Рекомендовано минимизировать прямую блёскость от источников освещения и отраженную блёскость на поверхностях таких как экран, стол или клавиатура. Уменьшение отраженной блёскости достигается за счет верного выбора типов электроосветительных приборов и схемы расположения рабочих мест по отношению к источникам освещения. При этом яркость потолка, а также светящихся поверхностей, таких как окна или светильники, находящиеся в поле зрения пользователя при работе, не должны быть выше 200 кд/м2, а яркость бликов на дисплее не должно превышать 40 кд/м2.

Неравномерное распределение яркости в поле зрения пользователя ПЭВМ следует свести к минимуму. В областях между рабочими поверхностями соотношение яркости должно быть меньше, чем три к одному, а между рабочими поверхностями, стенами и оборудованием ниже, чем десять к одному.

Для искусственного освещения следует применять преимущественно люминесцентные лампы белого света и компактные люминесцентные лампы. При устройстве отражённого освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные.

Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с ЭПРА, состоящих из равного числа опережающих и отстающих ветвей. Применение светильников без рассеивателей и экранирующих решеток не допускается. При отсутствии светильников с ЭПРА лампы многоламповых светильников или рядом расположенные светильники общего освещения следует включать на разные фазы трёхфазной сети.

В случае использования люминесцентных светильников общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении дисплеев.

При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращённому к оператору.

Коэффициент запаса для осветительных установок общего освещения должен приниматься равным 1,4 условных единиц.

Коэффициент пульсации не должен превышать 5 %.

Для обеспечения нормируемых значений освещённости в помещениях для использования ПЭВМ следует проводить чистку стёкол, оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

**7.6 Требования к организации и оборудованию рабочих мест**

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с мониторами, должно быть не менее 2,0 м, а расстояние между боковыми поверхностями мониторов – не менее 1,2 м.

Расстояние между экраном монитора и глазами пользователя должно быть в пределах от 600 до 700 мм, но не ближе 500 мм.

Рабочие места с ПЭВМ в помещениях с наличием источников вредных производственных факторов обязаны размещаться в изолированных кабинах с организованным воздухообменом.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования, поддержание рациональной рабочей позы при эксплуатации ПЭВМ, а также позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

Поверхность рабочего стола должна иметь коэффициент отражения от 0,5 до 0,7.

Тип рабочего кресла следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ. Рабочее кресло должно иметь подъёмно-поворотный механизм, позволяющий производить регулировку по высоте и углам наклона сиденья и спинки, а также расстояния спинки от переднего края сиденья. При этом настройка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Поверхность сиденья, спинки и других элементов кресла должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах от 680 до 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Конструкция рабочего стула должна обеспечивать:

1. ширину и глубину поверхности сиденья не менее 400 мм;
2. плоскую поверхность сиденья с закругленным передним краем;
3. регулировку высоты поверхности сиденья в пределах от 400 до 550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов;
4. высоту опорной поверхности спинки от 280 до 320 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм;
5. угол наклона спинки в вертикальной плоскости в пределах от минус 30 до плюс 30 градусов;
6. регулировку расстояния спинки от переднего края сиденья в пределах от 260 до 400 мм;
7. стационарные или съемные подлокотники длиной не менее 250 мм и шириной от 50 до 70 мм;
8. регулировку подлокотников по высоте над сиденьем в пределах от 230 до 30 мм и внутреннего расстояния между подлокотниками в пределах от 350 до 500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Обязательно должна проводиться влажная ежедневная уборка помещений и систематическое проветривание после каждого часа работы с ПЭВМ.

## 7.7 Требования к организации медицинского обслуживания

Все сотрудники, работающие с ПЭВМ более половины рабочего времени, обязаны проходить предварительные при поступлении на работу и периодические медицинские осмотры в установленном порядке.

Медицинский осмотр студентов высших учебных заведений, учащихся средних специальных учебных заведений, детей дошкольного и школьного возраста на предмет установления противопоказаний к работе с ПЭВМ проводится в установленном порядке учреждения.

## 7.8 Требования электробезопасности

Во избежание поражения электрическим током необходимо оборудовать помещение, в котором установлены ПЭВМ, розетками с заземлением, либо отдельным контуром заземления, на который должны быть заземлены ПЭВМ и все устройства, подключаемые к ним.

Крайне нежелательна схема расположения ПЭВМ рядом с источниками влаги или воды.

Необходимо производить уборку от пыли для предотвращения поражения ПЭВМ статическим электричеством.

# ЗАКЛЮЧЕНИЕ

В ходе работы не удалось создать полноценную автоматизированную систему управления рабочим временем.

Однако была разработана ее основа и наиболее востребованная организацией часть системы: новый web-сайт организации и панель управления сайтом.

В дальнейшем запланировано расширение системы и добавление таких функций как возможность записи на прием через сайт, ведение администратором графиков сотрудников, новые режимы пользователей на сайте такие как «сотрудник» и «клиент».

Вышеописанные функции можно реализовать путем добавления новых таблиц в базу данных, написания классов новых контроллеров, создания новых моделей и добавления новых ролей в списки доступа.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Андреев Г.И. Практикум по оценке интеллектуальной собственности: учебное пособие / Г.И. Андреев, В.В. Витчинка. – Москва: Финансы и статистика, 2003. – 176 с

2. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Санитарно-эпидемиологические правила и нормативы СанПиН 2.2.2/2.4.1340-03.

3. Автоматизированные системы управления [Электронный ресурс] – Режим доступа: https://spravochnick.ru/informacionnye\_tehnologii/setevye\_informacionnye\_sistemy/avtomatizirovannye\_sistemy\_upravleniya/;

4. Шаблон MVC [Электронный ресурс] – Режим доступа: https://andrey.moveax.ru/post/mvc3-in-depth-basics-01-mvc-template;

5. PHP: Hypertext Preprocessor [Электронный ресурс] – Режим доступа: https://www.php.net/

6. Назначение и применение AJAX при создании сайтов [Электронный ресурс] – Режим доступа: <https://itchief.ru/lessons/javascript/what-is-ajax>

7. Русская документация по API jQuery [Электронный ресурс] – Режим доступа: https://jquery-docs.ru/

# ПРИЛОЖЕНИЕ А

(обязательное)

ER-диаграмма базы данных

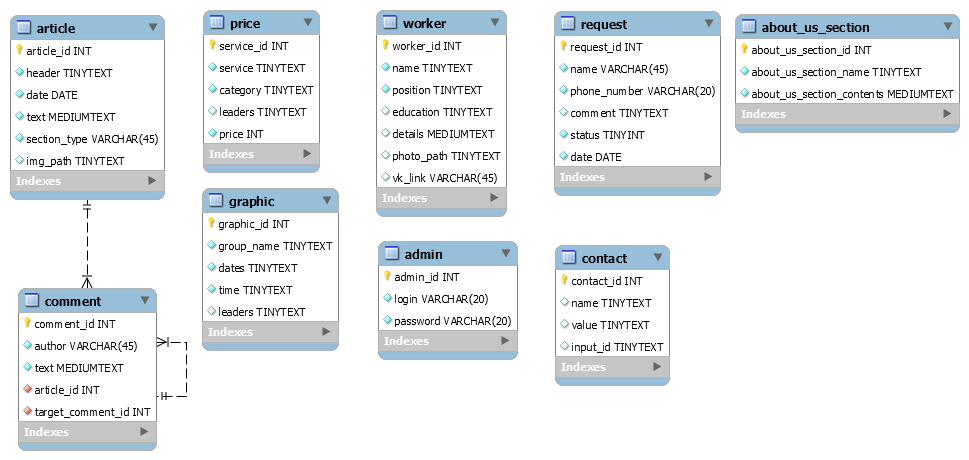


Рисунок А.1 – ER-диаграмма базы данных

# ПРИЛОЖЕНИЕ Б

(обязательное)

Описание таблиц базы данных

Таблица Б.1 – Описание таблиц базы данных

|  |  |
| --- | --- |
| Название таблицы | Описание |
| Article | Статьи и новости |
| Price | Прайс на услуги |
| Worker | Информация о сотрудниках |
| Request | Заявки на обратный звонок |
| About\_us\_section | Раздел “О центре” |
| Comment | Комментарии к новостям и статьям |
| Graphic | Графики групп |
| Admin | Информация для авторизации администратора |
| Contact | Контактная информация |

Таблица Б.2 – Статьи и новости

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Article\_id | Int | Идентификатор статьи |
| Header | Tinytext | Ее заголовок |
| Date | Date | Дата публикации |
| Text | Mediumtext | Текст статьи |
| Section\_type | Varchar(45) | Раздел: новость или статья |
| Img\_path | Tinytext | Путь к изображению к статье |

Таблица Б.3 – Прайс на услуги

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Service\_id | Int | Идентификатор услуги |
| Service | Tinytext | Название услуги |
| Category | Tinytext | Категория, к которой относится услуга |
| Leaders | Tinytext | Ведущие |
| Price | Int | Цена услуги |

Продолжение приложения Б

Таблица Б.4 – Информация о сотрудниках

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Worker\_id | Int | Идентификатор сотрудника |
| Name | Tinytext | ФИО сотрудника |
| Position | Tinytext | Должность |
| Education | Tinytext | Образование |
| Details | Mediumtext | Дополнительная информация |
| Photo\_path | Tinytext | Путь к расположению фото сотрудника |
| Vk\_link | Varchar(45) | Ссылка на ВК сотрудника |

Таблица Б.5 – Заявки на обратный звонок

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Request\_id | Int | Идентификатор заявки |
| Name | Varchar(45) | Имя автора заявки |
| Phone\_number | Varchar(20) | Телефон автора заявки |
| Comment | Tinytext | Комментарий |
| Status | Tinyint | Статус заявки |

Таблица Б.6 – Раздел “О центре”

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Aboutus\_section\_id | Int | Идентификатор раздела |
| Aboutus\_section\_name | Tinytext | Заголовок раздела |
| Aboutus\_section\_content | Mediumtext | Текст раздела |

Таблица Б7 – Комментарии к новостям и статьям

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Comment\_id | Int | Идентификатор комментария |
| Author | Varchar(45) | Автор комментария |
| Text | Mediumtext | Текст комментария |
| Article\_id | Int | Идентификатор статьи к которой оставлен комментарий |
| Target\_comment\_id | Int | Идентификатор комментария, в ответ к которому оставлен комментарий |

Продолжение приложения Б

Таблица Б.8 – Графики групп

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Graphic\_id | Int | Идентификатор услуги |
| Group\_name | Tinytext | Название услуги |
| Dates | Tinytext | Период проведения группы |
| Time | Tinytext | Время начала |
| Leaders | Tinytext | Ведушие группы |

Таблица Б.9 – Информация для авторизации администратора

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Admin\_id | Int | Идентификатор администратор |
| Login | Varchar(20) | Логин админитратора |
| Password | Varchar(20) | Пароль администратора |

Таблица Б.10 – Информация для авторизации администратора

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Описание |
| Contact\_id | Int | Идентификатор контакта |
| Name | Tinytext | Название контакта |
| Value | Tinytext | Значение контакта |
| Input\_id | Tinytext | Идентификатор для элемента input |

# ПРИЛОЖЕНИЕ В

(обязательное)

Класс Router (отвечает за маршрутизацию)

*class* Router{

    protected $routes = [];

    protected $params = [];

*function* \_\_construct(){

        $arr = require 'application/config/routes.php';

        foreach ($arr as $key => $val){

            $this->add($key, $val);

        }

        // debug($arr);

    }

    public *function* add($route, $params){

        $route = '#^'.$route.'$#';

        $this->routes[$route] = $params;

    }

    public *function* match(){

        $uri = $\_SERVER['REQUEST\_URI'];

        // Вырезаем имя домена

        $uri = preg\_replace('%/kodarcenter-final/%', '', $uri);

        // Вырезаем параметры, если они есть

        $uri = preg\_replace('%\?(.\*)%', '', $uri);

        foreach ($this->routes as $route =>$params){

            // Eсли, найденно совпадение с каким-либо маршрутом в формате controller/action

            if (preg\_match($route, $uri, $matches)){

                // То заносим его controller и action в поле Router->params

                $this->params = $params;

                return true;

            }

        }

        return false;

    }

    public *function* run(){

        if ($this->match()){

            $path = "application\controllers\\".ucfirst($this->params['controller'])."Controller";

            if (class\_exists($path)){

                $action = $this->params['action']."Action";

                if (method\_exists($path, $action)){

Продолжение приложения В

                    $controller= new $path($this->params);

                    $controller->$action();

                }

                else{

*View*::errorCode(404);

                }

            }

            else{

*View*::errorCode(404);

            }

        }

        else{

*View*::errorCode(404);

        }

    }

}

# ПРИЛОЖЕНИЕ Г

(обязательное)

Класс Controller (отвечает за привязку данных к их представлению)

abstract *class* Controller{

    public $route;// Содержит массив с ключами 'controller' и 'action'

    public $view;

    public $model;

    public $acl;

    public *function* \_\_construct($route){

        $this->route = $route;

        if (!$this->checkAcl()) {

*View*::errorCode(403);

        }

        $this->view = new *View*($route);

        $this->model = $this->loadModel($route['controller']);

    }

    public *function* loadModel($name){

        $path = 'application\models\\'.ucfirst($name);

        if (class\_exists($path))

            return new $path;

    }

    public *function* checkAcl(){

        $this->acl = require 'application/acl/'.$this->route['controller'].'.php';

        if ($this->isAcl('all')) {

            return true;

        }

        elseif (!isset($\_SESSION['admin']['id']) and $this->isAcl('guest')) {

            return true;

        }

        elseif (isset($\_SESSION['admin']) and $this->isAcl('admin')) {

            return true;

        }

        return false;

    }

    public *function* isAcl($key){

        return in\_array($this->route['action'], $this->acl[$key]);

    }

}