

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)
КИТП**

**ВЫПУСКНАЯ
КВАЛИФИКАЦИОННАЯ РАБОТА**

Студент Раков Антон Николаевич

Колледж информационных технологий и предпринимательства

Направление 09.02.04 – Информационные системы (по отраслям)

Тема выпускной квалификационной работы

Информационная система парикмахерской

Руководитель _____
подпись

Д. Д. Короткова
инициалы, фамилия

Студент _____
подпись

А. Н. Раков
инициалы, фамилия

**Допустить выпускную квалификационную работу к защите
в государственной экзаменационной комиссии**

Заведующий кафедрой ИСПИ _____
подпись

И.Е. Жигалов
инициалы, фамилия

«_____» _____ 20____ г.

АННОТАЦИЯ

Тема: Разработка информационной системы «Барбершоп».

Ключевые слова: информационная система, барбершоп, расписание, услуга.

В данной работе представлена реализация взаимодействия расписания мастеров барбершопа с конечным клиентом, написанная с использованием WEB технологий.

Выпускная квалификационная работа представлена на 71 страницах, рисунков – 23, использованных источников – 15, приложений – 2, таблиц - 12.

ABSTRACT

Subject: Development of information system «Barbershop».

Keywords: information system, barbershop, schedule, service.

This paper presents the implementation of the interaction of the barbershop master schedule with the end client, written using WEB technologies.

The course project is presented on 71 pages, figures - 23, used sources - 15, applications - 2, tables - 12.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1 Описание предметной области	8
1.2 Цели и задачи разработки	8
1.3 Процессы предприятия.....	9
1.4 Сравнение аналогов	13
1.5 Требования к информационной системе	15
2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	19
2.1 Технологии для разработки.....	19
2.2 Архитектура информационной системы	22
2.3 Логика и последовательность	25
2.4 Проектирование базы данных.....	27
3. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ	30
3.1 Разработка серверной части.....	30
3.2 Разработка клиентской части.....	31
3.3 Тестирование	33
3.4 Развёртывание	36
4. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ	39
4.1 Расчет стоимости разработки системы.....	39
4.2 Расчет стоимости выполнения процесса до автоматизации.....	43
4.3 Расчет стоимости выполнения системы после автоматизации	47
4.4 Расчет показателей эффективности системы.....	50

					ВлГУ.09.02.04.16.3.00 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дат	Информационная система парикмахерской Пояснительная записка	Лит.	Лист	Листов
Разраб.		Раков А. Н.				У	4	71
Провер.		Короткова Д. Д.				ИСсп-117		
Провер.		Шамышева О. Н.						
Н. Контр.		Маркова В. Б.						
Утверд.		Жигалов И. Е.						

ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	54
ПРИЛОЖЕНИЕ А.	56

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

ВВЕДЕНИЕ

Барбершоп — это предприятие, занимающееся предоставлением услуг для населения по уходу за волосами (стрижка, завивка, создание причёски, окраска, мелирование и другие виды работ с красителями, стрижка огнём, бритьё и стрижка бород и усов и др.) в оборудованном специально для этого помещении. Как правило, в парикмахерских дополнительно оказываются следующие виды услуг: маникюр, педикюр, косметические услуги и услуги визажиста.

В барбершопе существует потребность в системах, обеспечивающих ведение учёта работы мастеров, информации о клиентах, хранение информации обо всех услугах, а также возможность осуществлять запись без участия персонала. Особенность барбершопа подразумевает хранение информации об услугах и клиентах, а также о графике работы самих мастеров – барберах, ведь от этого зависит сможет ли конечный клиент получить необходимую услугу в необходимое время. Создание системы, которая позволила бы все это реализовать, обеспечило бы более качественную учетную деятельность барбершопа.

Эти спецификации применимы не только к барбершопу, но и к другим заведениям, где оказывают подобные услуги. Делается вывод, что данная предметная область актуальна, так как растет спрос на программное обеспечение для таких предприятий. Однако создание и содержание таких систем может быть очень дорогостоящим, что будет не выгодно для конечных предприятий.

Для удешевления и упрощения процесса управления барбершопом разумно создать расширяемую систему на базе WEB технологий.

В рамках данной работы рассмотрена разработка серверной и клиентской

					<i>ВлГУ.09.02.04.16.3.00 ПЗ</i>	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

части для возможности взаимодействовать с системой не только за компьютером, но и с использованием любого другого устройства, которое поддерживает выход в интернет и имеет на своём борту браузер.

					<i>ВлГУ.09.02.04.16.3.00 ПЗ</i>	<i>Лист</i>
						<i>7</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области

Барбершоп в своём классическом варианте — это парикмахерская для мужчин. Так как барбершоп является предприятием, предоставляющее людям определённые услуги, то предприятие ведет расписание мастеров и что-нибудь дописать.

На предприятии происходят следующие процессы:

- ведение расписания мастеров для получения услуг;
- ведение клиентской базы;
- ведение базы сотрудников;
- ответ на обратный звонок.

Исходя из рассмотренных процессов предприятия, в рамках выполнения работы реализован следующий функционал:

- учет клиентов – персональная информация клиента для связи с ним (ФИО и телефон);
- учёт мастеров и персонала;
- учёт расписания.

1.2 Цели и задачи разработки

Цель данного проекта – автоматизация основополагающих процессов работы предприятия для сокращения затрат времени на выполнение

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

посредством внедрения расширяемой информационной системы, включающую серверную и клиентскую конфигурации системы.

Основная задача реализации проекта – удобный процесс записи клиента на необходимую услугу по средствам динамического расписания. Процесс подготовки расписания заканчивается генерацией HTML-таблицы с временными ячейками и условно разделяется на следующие этапы:

- создание записей мастеров с ФИО и личной информацией;
- создание необходимых объектов сущностей для реализации проекта (календарные дни, временные промежутки, услуги и клиенты);
- создание определённого SQL запроса, который выполняет функцию сбора данных в один результирующий массив данных;
- генерация данных в HTML-таблицу с удобным пользовательским интерфейсом и стилизация для более удобного восприятия.

В итоге, необходимые данные должны быть синхронизированы между серверным и клиентскими приложениями, для упрощения некоторых процессов предприятия.

1.3 Процессы предприятия

Рассмотрим некоторые процессы предприятия, функционал которых необходимо реализовать. В данный перечень входит:

- ведение расписания мастеров предприятия;
- ведение учёта текущих записей;
- ведение учёта запросов на обратный звонок.

Автоматизация данных процессов в информационной системе позволит не только упростить сам процесс записи клиента на услугу, но и обеспечить снижение появления некорректных данных о свободных временных периодах. Также с помощью обратного звонка появляется возможность оперативно реагировать на проблемы и пожелания пользователей и клиентов. Рассмотрим

перечисленные процессы в виде диаграмм последовательности до и после внедрения информационной системы на предприятие.

На рисунках 1-2 представлен процесс записи на получение услуги до внедрения информационной системы на предприятие.

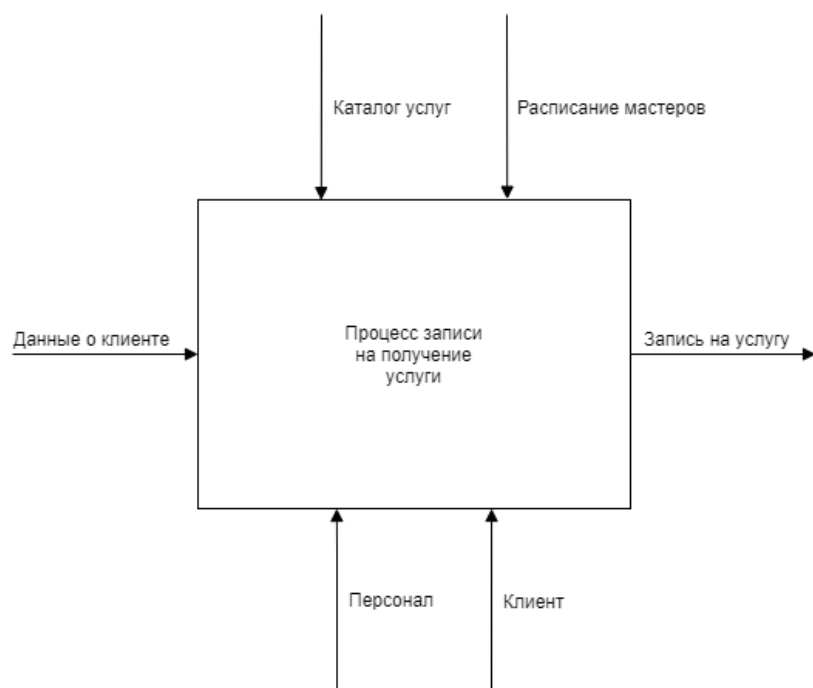


Рисунок 1 - Процесс записи на получение услуги до внедрения информационной системы

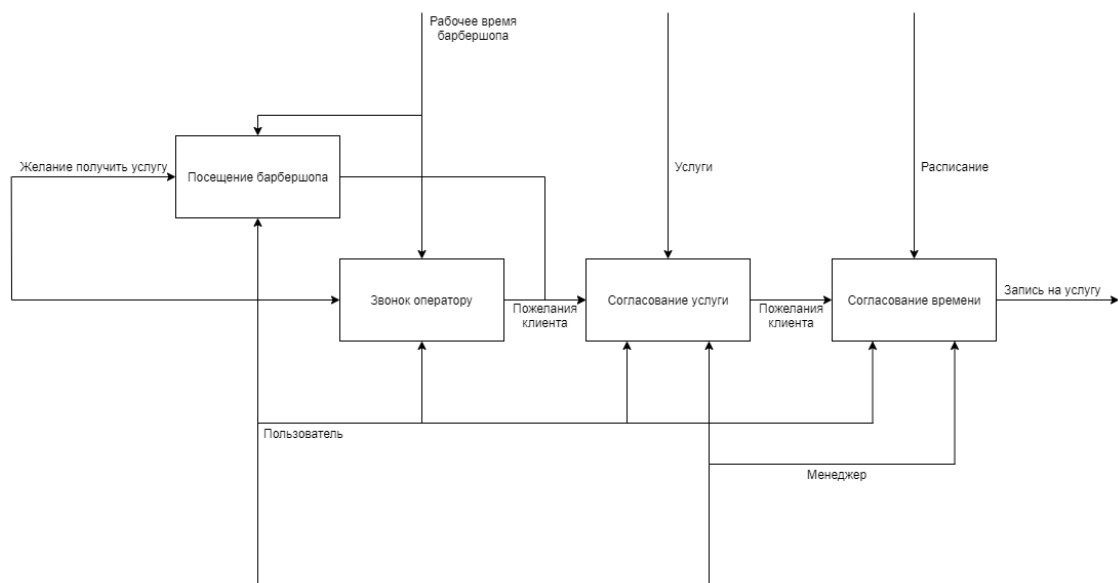


Рисунок 2 - Детализация процесса записи на получение услуги до внедрения информационной системы

Изм.	Лист	№ докум.	Подпись	Дата

ВлГУ.09.02.04.16.3.00 ПЗ

Лист

10

На рисунках 3-6 представлен процесс записи на получение услуги после внедрения информационной системы на предприятие.

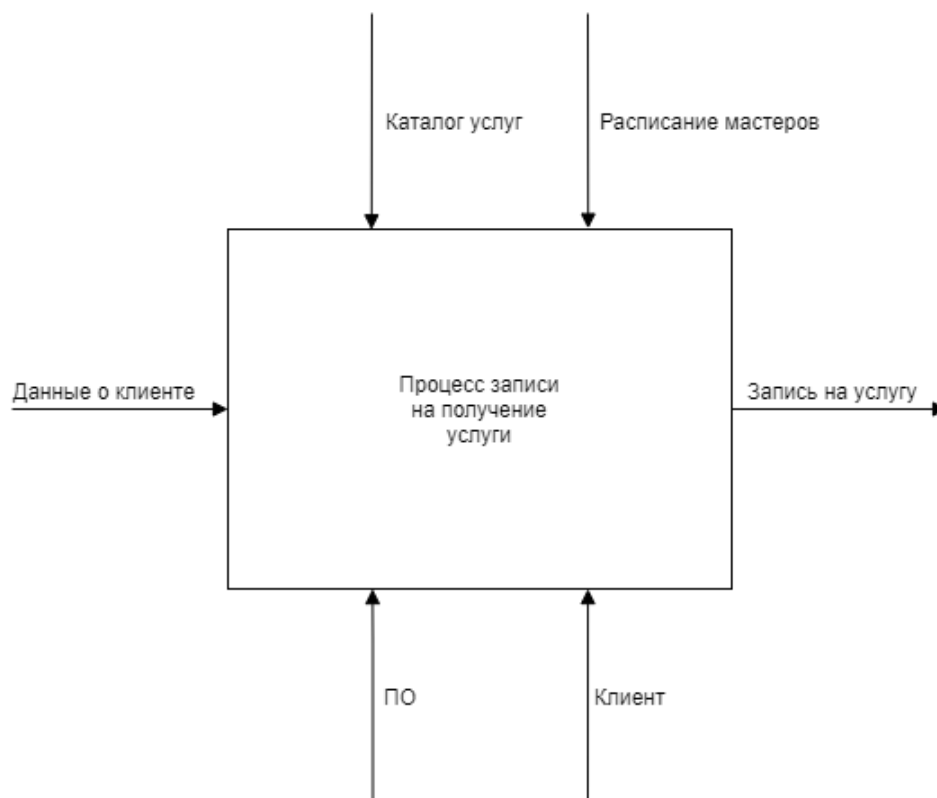


Рисунок 3 - Процесс записи на получение услуги после внедрения информационной системы

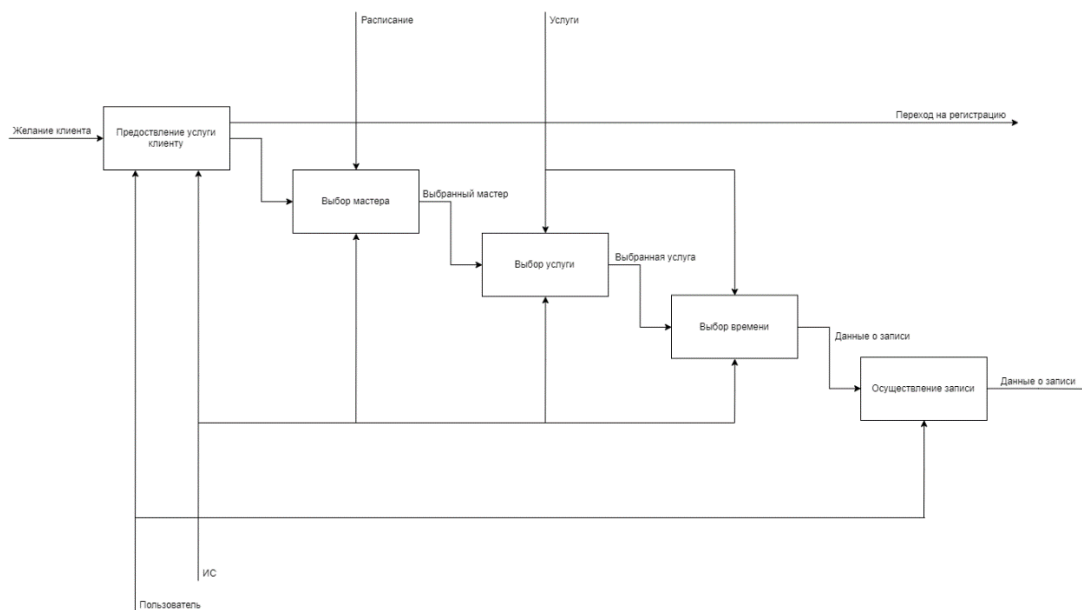


Рисунок 4 - Детализация процесса записи на получение услуги после внедрения информационной системы

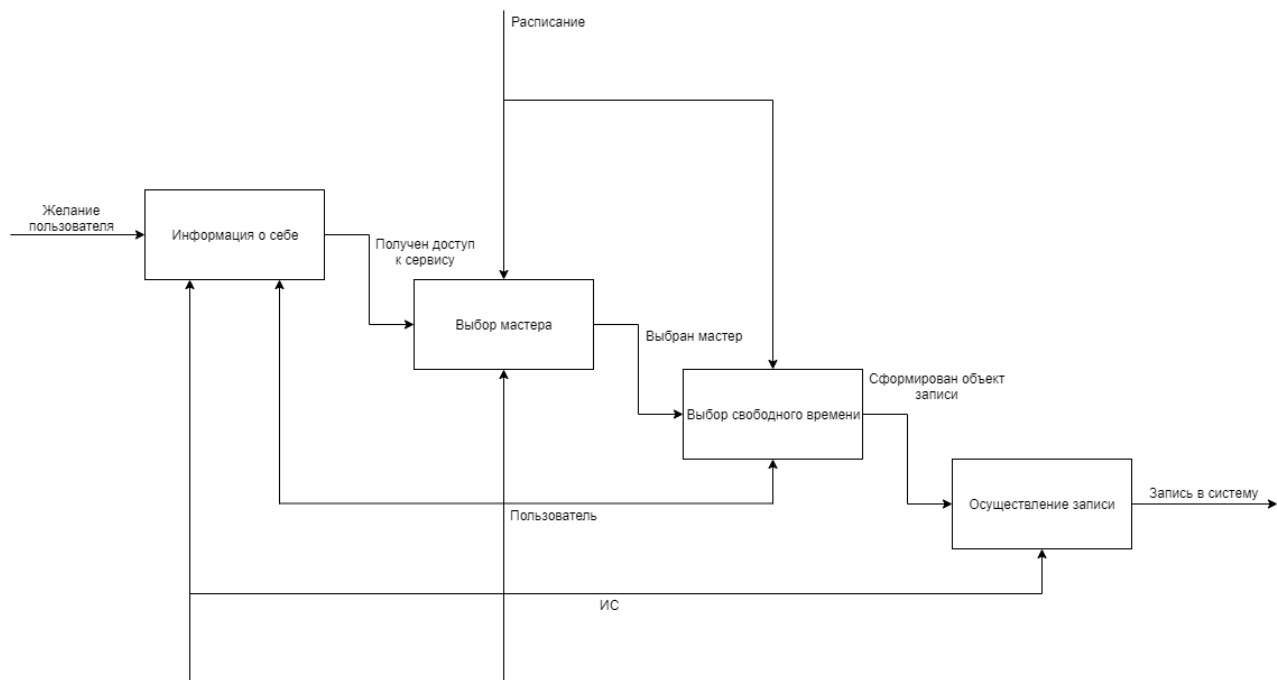


Рисунок 5 - Детализация процесса записи на получение услуги после внедрения информационной системы

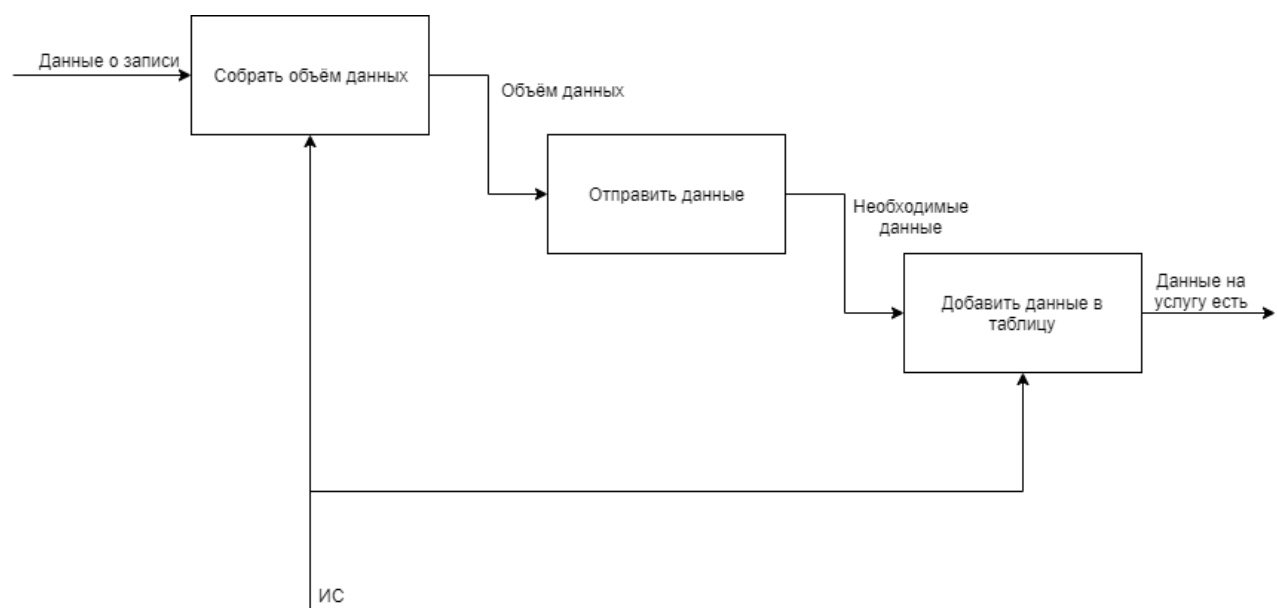


Рисунок 6 - Детализация процесса записи на получение услуги после внедрения информационной системы

1.4 Сравнение аналогов

Важно требование при проектировании информационной системы состоит в том, чтобы создаваемая система была лучше и эффективней существующей, то есть аналога. ИС должна превосходить по функционированию, качеству и экономической целесообразности. В результате проведённого поиска аналогов и анализа типовых технических решений рассмотрен ряд существующих ИС:

- YCLIENTS;
- Bookform;
- модуль «Онлайн-запись» UMI.CMS.

YCLIENTS — это сервис онлайн-записи и платформа автоматизации для сферы услуг. Компания предоставляет CRM которая позволяет вести базу клиентов и запись на услуги, а также интеграцию со многими системами учёта и аналитики. Стоимость лицензии начинается от 1200 руб\месяц. Предоставляет возможность встраивать функциональный виджет на собственный веб-сайт либо пользоваться виджетом на платформе YCLIENTS.

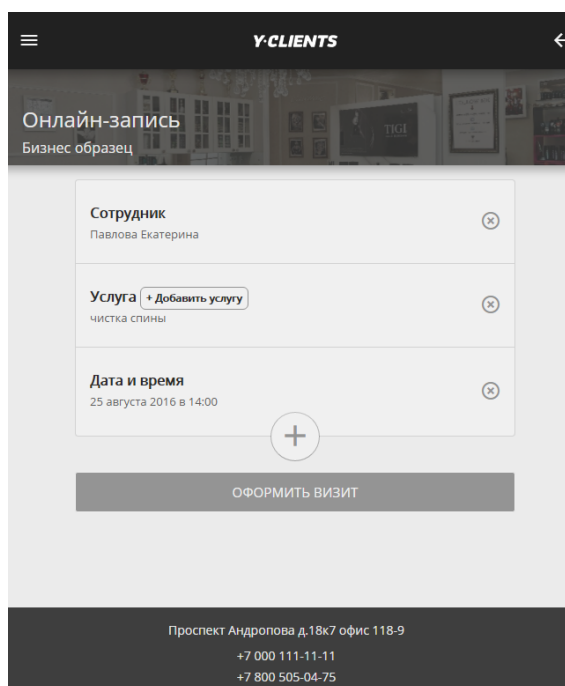


Рисунок 7 - Скриншот окна записи сервиса YCLIENTS

					ВЛГУ.09.02.04.16.3.00 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

Bookform – онлайн-запись и работа с клиентами. Это бесплатная CRM-система от Яндекса. Позволяет настроить запись клиентов, продажу билетов, аренду площадок и оборудования. Лицензия бесплатная. Предоставляет возможность встраивать функциональный виджет на собственный веб-сайт либо пользоваться виджетом на платформе Яндекса.

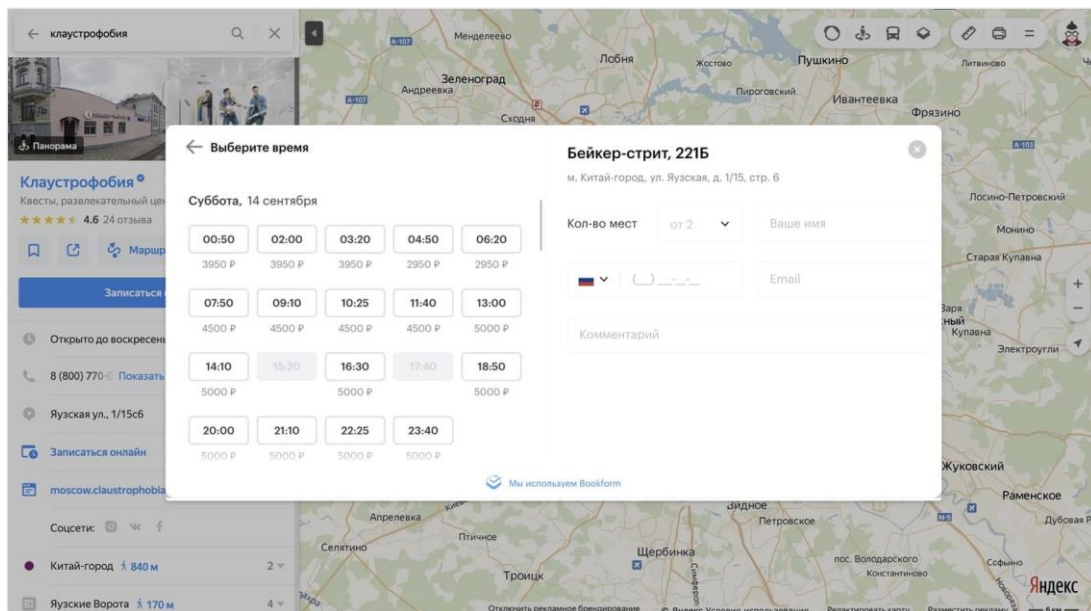


Рисунок 8 - Скриншот окна записи сервиса Bookform

UMI.CMS – это система управления контентом. Предоставляет модуль «Онлайн-запись». Стоимость лицензии 6900 рублей на бессрочный период. Предоставляет встроить данный модуль на собственный веб-сайт, которой находится под управлением системы управления контентом UMI.CMS.

1 Выберите услугу:

СЕРВИСНОЕ ОБСЛУЖИВАНИЕ

→ Ремонт техники 250

→ Замена комплектующих 150

→ Установка и настройки 100

Доступное время записи: Свободно

Сегодня ЧТ утро день вечер

Завтра ПТ утро день вечер

25 июля ПН утро день вечер

26 июля ВТ утро день вечер

В другие дни - свободно*

Мастера, выполняющие услугу:

→ Василий Зайцев

2 Специалист (пожалуйста, укажите)

3 Дата визита (пожалуйста, укажите)

4 Подтверждение заявки на визит (пожалуйста, заполните)

Рисунок 9 - Скриншот окна записи модуля «Онлайн-запись» сервиса UMI.CMS

Исходя из анализа приведённых аналогов, можно сделать вывод, что разработка системы является целесообразной, потому что:

- аналоги ограничивают в действиях. Нет возможности добавить новый функционал или доработать старый. Только готовые решения;
- два из трёх сервисов платные, Bookform условно бесплатный (комиссия за платежи). Данные решения не являются экономически выгодными;
- зависимость от компании и её политики. Стоимость продукта или его правила работы могут измениться в любое время. Это достаточно опасный фактор из-за которого работа предприятия может встать, а в следствии потеря денег компанией.

1.5 Требования к информационной системе

До начала проектирования информационной системы нужно определиться с требованиями к ней. Допускается выделение следующей группы

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

требований: к функциональности, к эргономичности, к надежности, к производительности, к обслуживанию до момента завершения жизненного цикла ИС.[1]

1.5.1 Функциональные требования

Немаловажными требованиями, которые предъявляются к разработке информационных систем, являются функциональные требования. В разрабатываемой информационной системе должны присутствовать следующие свойства:

а) распределённая система доступа к информации. У пользователя должно быть три уровня, а именно:

- 1) пользователь;
- 2) менеджер;
- 3) администратор;

б) гость имеет возможность зарегистрироваться на сайте и создать 2 записи на получение услуги в неделю. Расписать про регистрацию и авторизацию;

в) менеджер имеет возможность просматривать информацию об обратных звонках и удалять её, просматривать информацию о мастерах, а также удалять и создавать её, просматривать информацию о пользователях уровня пользователь, просматривать записи на получение услуг;

г) администратор имеет полный доступ к системе, а именно все возможности менеджера, а также может просматривать информацию о пользователях любого уровня, а также создавать и удалять её.

1.5.2 Нефункциональные требования

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

Следующие не менее значимые требования, относящиеся к информационной системе, это нефункциональные требования:

- система не должна быть ориентирована на опытного пользователя ПК, так как пользователями являются сотрудники организации, чей профиль деятельности, далек от информационных технологий. Данным сотрудникам совсем не обязательно знать техническую базу по работе с компьютером в совершенстве;

- разрабатываемая информационная система должна максимально ёмко упрощать процессы обработки массивов информации и взаимодействие с ней не должно вызывать затруднений. Интерфейс разрабатываемой системы должен быть сделан максимально удобным и понятным на интуитивном уровне любому конечному пользователю;

- информационная система должна быть стабильной и надежной на протяжении всего времени использования. Нужно достичь серьёзных результатов, чтобы падения системы и сбои случались очень редко при этом не приводя к потере или повреждению базы данных, а также отсутствию работоспособности сторонних клиентских приложений;

- разрабатываемая информационная система должна быть довольно производительной и не требовать большой объём ресурсов системы;

- использование ИС не должно приводить к модернизации имеющихся аппаратных устройств на данный момент, а точнее системного блока персонального компьютера или другого девайса на котором будет развёрнута система;

- время отклика работающего системного комплекса должно быть минимальным на основе возможностей имеющейся компьютерной техники, так как работа с сайтом и таблицами производиться непрерывно и является основным условием функционирования данного предприятия. Отклик не более 30мс;

- в виду того, что разрабатывается информационная система работа, кой предполагается на стороне клиента и предприятия, ресурсы ИС должны быть

доступны ежесекундно, но несмотря на это надёжно защищены от не санкционированного доступа;

- разрабатываемая информационная система должна быть адаптивной и кроссплатформенной, потому что на предприятии возможно использование некоторых программных решений на одном ПК для выполнения различных обязанностей персонала. Поэтому информационная система должна быть бесконфликтной по отношению с программными продуктами сторонних разработчиков;

- также одним из важных свойств информационной системы должна быть ее модернизация и добавление нового функционала. Должен быть вариант дальнейшего расширения разрабатываемой системы, поскольку реализация полноформатного программного продукта решающего все задачи в рамках ВКР практически невозможно;

- адаптивность — значит приспособленность разрабатываемой ИС к необходимым условиям. Сайт должен корректно отображаться на любом из современных устройств: планшеты, смартфоны, ноутбуки, настольные компьютеры. Сайт должен корректно отображаться в интернет-браузерах актуальных версий: Microsoft Edge 17, Mozilla Firefox 62 и новее, Opera 56, Safari 9, Google Chrome 68 и новее, YaBrowser 18.9 и новее;

- целостность – данный аспект значит следующее - все элементы ИС выполняют свои функции как одно целое;

- разрабатываемая ИС должна иметь защиту от несанкционированного доступа.

2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Технологии для разработки

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты (потребители этих функций).

Практические реализации такой архитектуры называются клиент-серверными технологиями.[2]

Для реализации серверной части информационной системы выбран следующий стек технологий:

- язык программирования PHP [12];
- фреймворк RedBeanPHP [13];
- MySQL [14];
- OpenServer [15].

PHP - это широко используемый язык сценариев общего назначения с открытым исходным кодом. Говоря проще, PHP это язык программирования, специально разработанный для написания web-приложений, исполняющихся на Web-сервере. Аббревиатура PHP означает «Hypertext Preprocessor (Препроцессор Гипертекста)». Данный язык был выбран исходя из следующих преимуществ:

- PHP является бесплатным. Исходный код находится в открытом доступе;
- для PHP создано немалое количество фреймворков, библиотек и БД;
- PHP гибкий и совместим с большим количеством серверов.

RedBeanPHP - это простой, легкий в использовании картограф объектов «на лету», особенно подходящий для RAD, прототипирование и люди с дедлайнами. RedBeanPHP автоматически создает таблицы, столбцы, ограничения и индексы, поэтому вам не нужно постоянно переключаться между клиентом базы данных (phpMyAdmin) и редактором (это не значит, что вам никогда не придется использовать phpMyAdmin или SQL). Поскольку RedBeanPHP экономит много времени, вы можете потратить больше времени на разработку остальной части приложения.

Данный фреймворк был выбран исходя из следующих преимуществ:

- нет конфигурации - Большинство ORM используют конфигурационные файлы (XML, INI или YAML) или какую-то систему аннотаций для определения отображений. Вместо того чтобы использовать конфигурацию, он использует соглашения; очень маленький набор правил. RedBeanPHP использует эти соглашения для вывода взаимосвязей и автоматизации сопоставлений. RedBeanPHP также помогает вам следовать этим соглашениям, автоматически создавая исходные таблицы и столбцы, что также экономит много времени. Это означает, что нет никакой конфигурации, меньше шаблонного кода и больше времени осталось сосредоточиться на бизнес-логике, тестировании и документации, тем самым повышая производительность разработки и качество кода;

- качество кода - RedBeanPHP был тщательно спроектирован, чтобы быть кратким и гибким. Кодовая база содержит много встроенной документации, полностью объектно-ориентирована и повышает безопасность за счет продвижения подготовленных операторов на основе PDO и привязки параметров;

MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Несмотря на отсутствие некоторого функционала, имеющегося у других СУБД, MySQL обладает достаточно обширным разнообразием доступных инструментов для

создания приложений[3]. Выбор данной СУБД можно аргументировать следующими преимуществами:

- Масштабируемость – система легко расширяется для работы с большим количеством данных
- Функционал – данный продукт может похвастаться большим функционалом, инструментов которого хватит для разработки проекта любого уровня

OpenServer - это портативная программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов.

Open Server Panel широко используется с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях.[4]

Для разработки клиентской части приложения выбран следующий набор технологий:

- HTML – это язык гипертекстовой разметки. Определяет содержание и структуру, но не внешний вид. Элементы языка имеют структуру дерева. Узлы дерева представляют собой либо текст либо структурные элементы, которые называются теги имеют различные атрибуты.

- CSS – формальный язык описания внешнего вида документа, написанного с помощью языка гипертекстовой разметки HTML. CSS работает со шрифтами, полями, таблицами, отступами, картинками и остальными элементами, которые предоставляет язык гипертекстовой разметки.[7] Также данный язык предоставляет возможность создания анимации и различных визуальных эффектов.

- Google Fonts - позволяет легко привнести индивидуальность и производительность в веб-сайты и продукты. Надежный каталог шрифтов и иконок с открытым исходным кодом позволяет легко интегрировать

выразительный шрифт и иконки - независимо от того, где вы находитесь в мире. Google Fonts - это библиотека из 1052 бесплатных лицензионных семейств шрифтов и API для удобного использования шрифтов через CSS и Android. Библиотека предоставляет восхитительные, красиво сделанные иконки для общих действий и предметов.

- JavaScript – объектно-ориентированный скриптовый язык программирования. JS встраиваемый язык. Имеет широкое применение при создании веб-страниц. Данный язык не требует компиляции, а подключается к странице и работает «как есть». Является высокоуровневым, динамическим, нетипизированным и интерпретируемым языком программирования.

2.2 Архитектура информационной системы

ИС «Барбершоп» – это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер (в широком смысле).

Основная часть приложения, находится на стороне веб-сервера, который обрабатывает полученные запросы в соответствии с бизнес-логикой продукта и формирует ответ, отправляемый пользователю. На этом этапе в работу включается браузер, именно он преобразовывает полученный ответ от сервера в графический интерфейс, понятный пользователю. На рисунке 10 изображена архитектурная схема информационной системы.

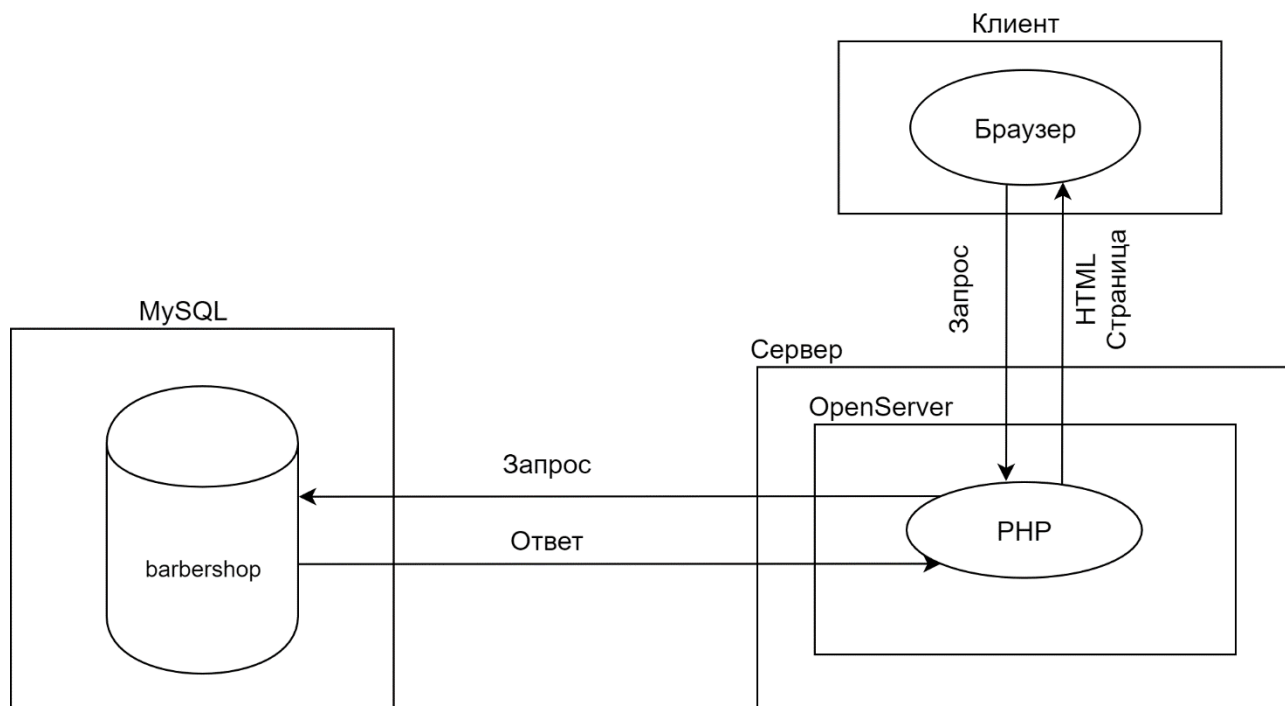


Рисунок 10 - Архитектурная схема ИС

2.2.1 Клиентская часть

Клиент — из названия становится понятно, что это пользователь сервиса (веб-приложения), который обращается к серверу для получения какой-то информации. На рисунке 11 представлена диаграмма навигации по клиентской части информационной системы.

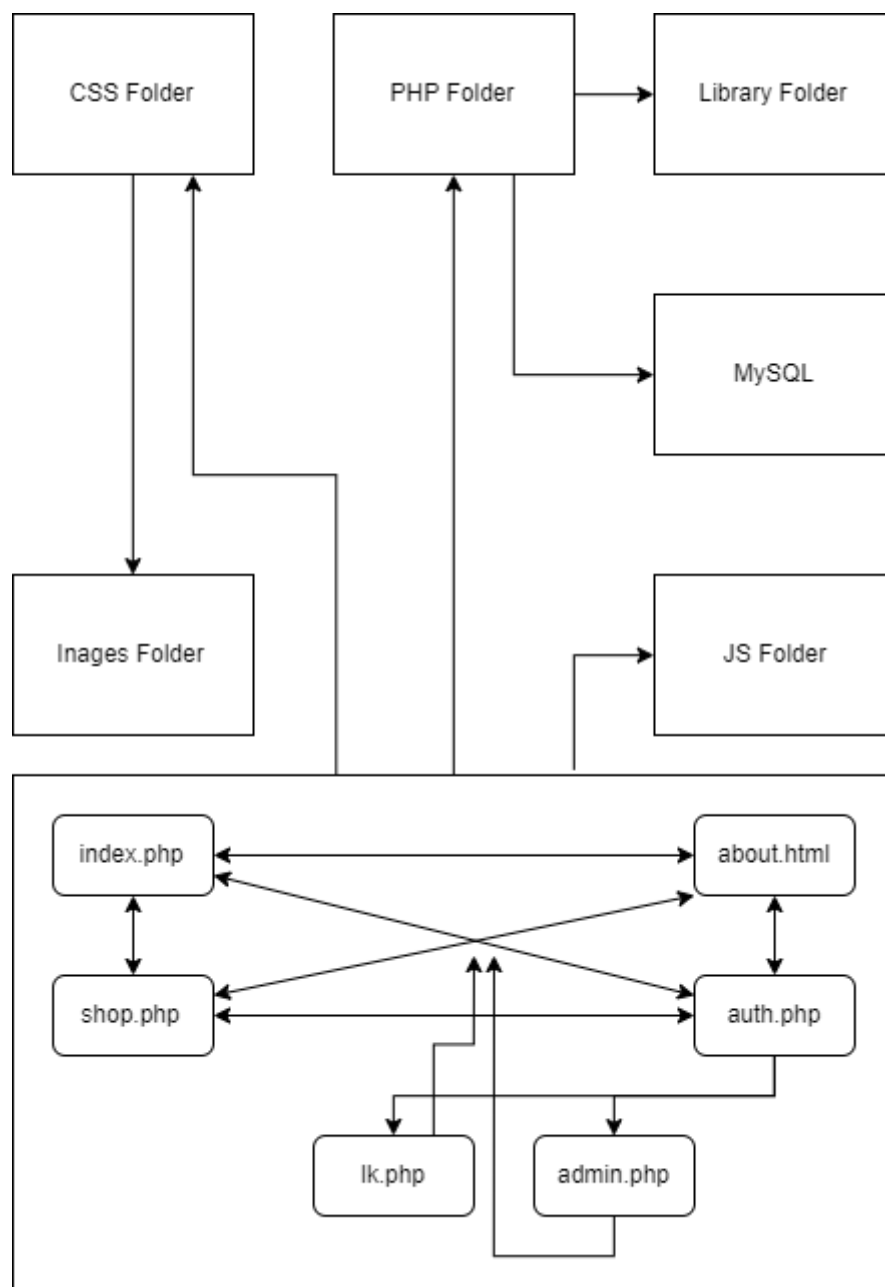


Рисунок 11 - Диаграмма навигации

2.2.2 Серверная часть

Сервер — место, где располагается твоё веб-приложение или его серверная часть. Он владеет необходимой информацией о пользователях или может ее запрашивать. Также при обращении клиента сервер возвращает ему запрашиваемую информацию.

Сеть — все просто: обеспечивает обмен информацией между клиентом и сервером.

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов и выдающий им HTTP-ответы. Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает. Наиболее распространенными видами ПО веб-серверов являются Apache, IIS и NGINX. На веб-сервере функционирует тестируемое приложение, которое может быть реализовано с применением самых разнообразных языков программирования: PHP, Python, Ruby, Java, Perl и пр.

База данных фактически не является частью веб-сервера, но большинство приложений просто не могут выполнять все возложенные на них функции без нее, так как именно в базе данных хранится вся динамическая информация приложения (учетные, пользовательские данные и пр).

База данных - это информационная модель, позволяющая упорядоченно хранить данные об объекте или группе объектов, обладающих набором свойств, которые можно категоризировать. Базы данных функционируют под управлением так называемых систем управления базами данных (далее – СУБД). Самыми популярными СУБД являются MySQL, MS SQL Server, PostgreSQL, Oracle (все – клиент-серверные).

2.3 Логика и последовательность

Основная цель данной информационной системы – это предоставление клиенту удобного веб-приложения для записи на получение услуги, а для предприятия – это экономия ресурсов и удобное отслеживание всего процесса начиная от записи клиента до получения услуги. Рассмотрим подробно процесс записи на услугу со стороны ИС. На рисунке 13 представлена диаграмма последовательности записи на получение услуги.

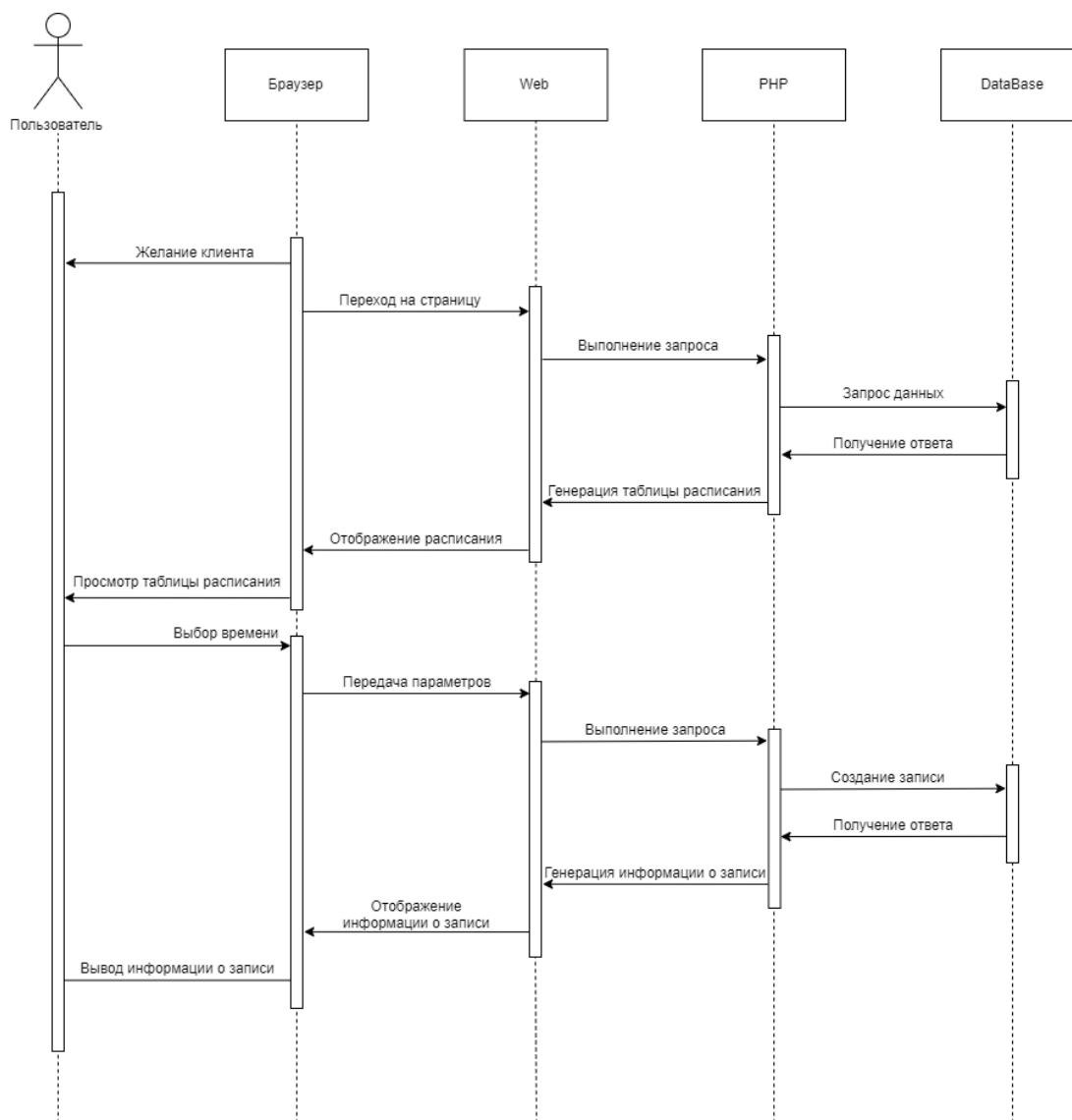


Рисунок 12 - Диаграмма последовательности записи на получение услуги

На рисунке 13 представлена диаграмма алгоритмов основного процесса.

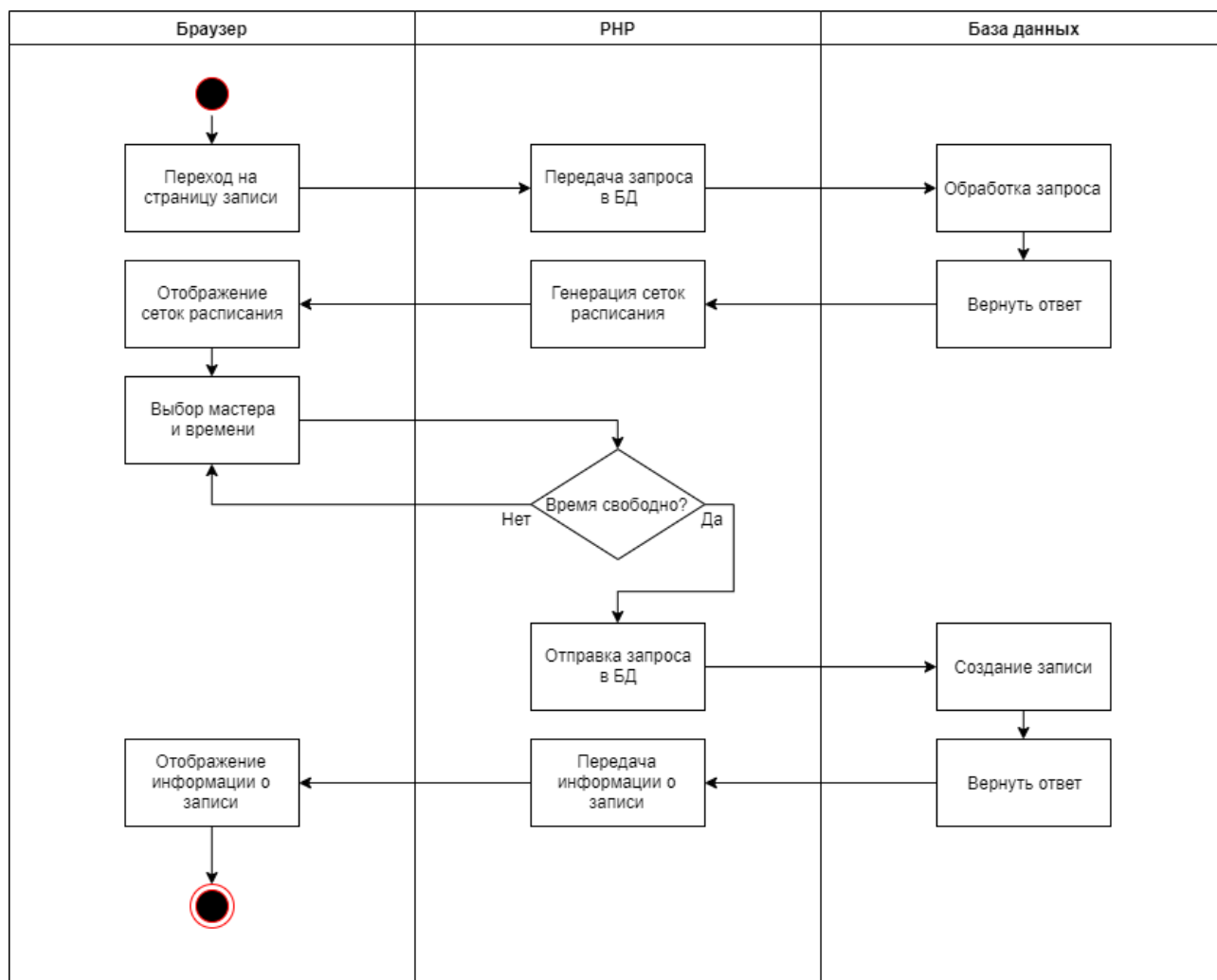


Рисунок 13 - Диаграмма алгоритмов основного процесса

2.4 Проектирование базы данных

База данных – именованная, структурированная совокупность взаимосвязанных данных, характеризующих отдельную предметную область и находятся под управлением СУБД (система управления базами данных).

Достоинство системы с БД по сравнению с классическим методом ведения учёта:

- компактность;
- актуальность;
- скорость;
- малые трудозатраты;

- централизованное управление данными и информацией;
- независимость данных и информации.

Система БД включает в себя четыре главных компонента: данные, аппаратное обеспечение, программное обеспечение (в конкретном случае это система управления базой данных, или СУБД) а также пользователи.

На основании анализа предметной области следует выделить следующие сущности:

- Пользователи – данная сущность используется для хранения информации о пользователях системы. Пользователя характеризует имя, номер телефона, пароль и уровень доступа. В этой сущности хранятся как обычные пользователи, так менеджеры и администраторы. Разделение реализовано благодаря полю «уровень доступа», но при этом таблица одна, что позволяет повысить производительность системы по средствам минимизации таблиц. Пользователь может внести только имя или ФИО полностью (100 символов для такого случая). Номер телефона храниться в десятизначном формате. Пароль по длине не должен превышать 100 символов.

- Мастера – данная сущность используется для хранения информации о мастерах предприятия. Мастера характеризует ФИО, номер телефона, а также биография (например стаж, опыт работы и возможные достижения в профессиональной сфере, мастер-классы и конкурсы). В данной сущности храниться информация только о мастерах, они не считаются за пользователей.

- Записи – данная сущность используется для хранения информации о записях на получение услуги. Данную сущность характеризует идентификатор, идентификатор клиента, для указания ссылки на сущность «пользователи», идентификатор даты записи для указания ссылки на сущность «дни», идентификатор времени записи для указания ссылки на сущность «время», идентификатор мастера, который будет предоставлять услугу для указания ссылки на сущность «мастера» и описание предоставляемой услуги. С помощью данного решения отсутствует логическая перегрузка.

- Дни – данная сущность используется для хранения информации о днях, которые используются для записи на получение услуги. Данную сущность характеризует идентификатор и день. Под словом «день» подразумевается конкретная дата (например 20.05.2021). На данную сущность ссылается сущность записи для указания даты.

- Время – данная сущность используется для хранения информации о времени, которое используется для записи на получение услуги. Данную сущность характеризует идентификатор, а также время. Под словом «время» подразумевается конкретное время (например 13:00). На данную сущность ссылается сущность записи для указание временного периода.

- Запросы (идентификатор, фамилия имя отчество, номер телефона) – данная сущность используется для хранения информации о запросах на обратный звонок. Данную сущность характеризует идентификатор, а также фамилия имя отчество человека, который оставил заявку на обратный звонок и его номер телефона на который будет произведён звонок. Так как сущность используется только для регистрации запросов на обратный звонок и конечный клиент неизвестен системе, то необходимость ссылаться на сущность пользователи отсутствует.

На рисунке 14 представлена схема базы данных.

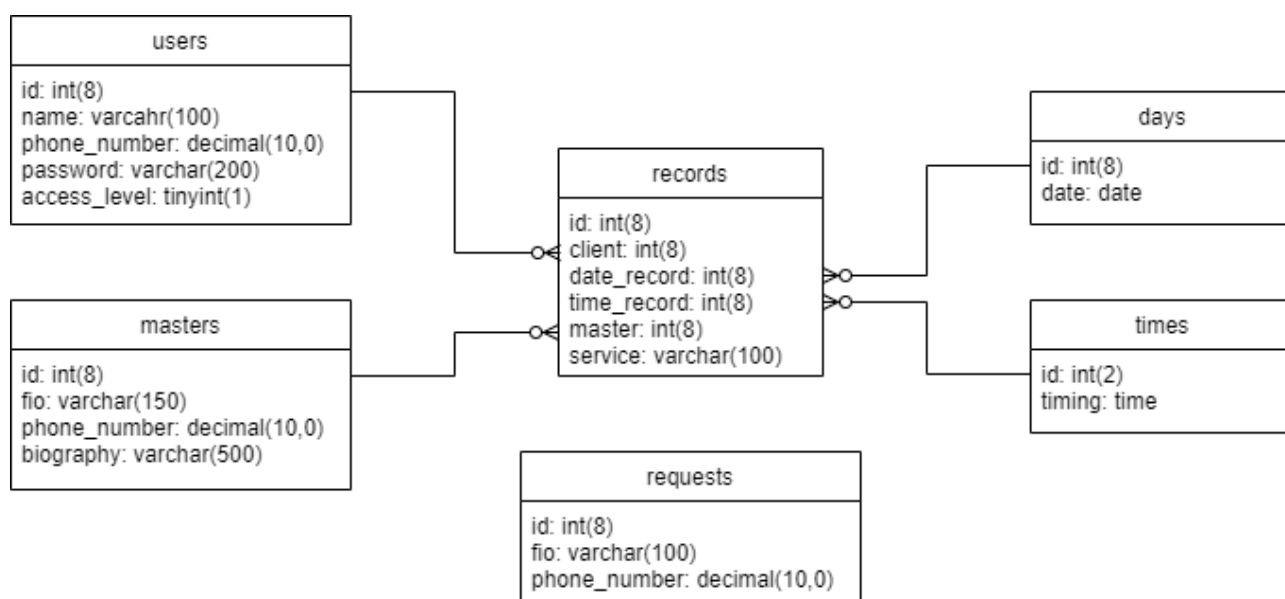


Рисунок 14 - Схема базы данных

3. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Разработка серверной части

Важной частью информационной системы является безопасность, а именно защита от несанкционированного доступа. Так как HTTP-запросы передаются с помощью метода GET, то необходима защита от такой уязвимости как SQL-инъекция.

SQL-инъекция - это атака, смысл которой заключается во вставке или “инъекции” SQL-запроса через входные данные от клиента в приложение. Успешный эксплойт SQL-инъекции может считывать конфиденциальные данные из базы данных, изменять данные базы данных (Вставлять/обновлять/Удалять), выполнять операции администрирования базы данных (например, выключать СУБД), восстанавливать содержимое данного файла, присутствующего в файловой системе СУБД, и в некоторых случаях выдавать команды операционной системе.[5] Атаки SQL-инъекций-это тип инъекционных атак, при которых SQL-команды вводятся во входные данные плоскости данных для того, чтобы повлиять на выполнение predetermined команды SQL.

Серьезность атак SQL-инъекций ограничена умением и воображением атакующего и, в меньшей степени, глубокими контрмерами защиты, такими как соединения с низкими привилегиями к серверу базы данных и т.д. В общем, SQL-инъекция признано считать за высокий уровень воздействия из вне.

Внедрение SQL-кода стало распространенной проблемой для веб-сайтов, управляемых базами данных. Этот недостаток легко обнаруживается и легко

эксплуатируется, и поэтому любой сайт или программный пакет даже с минимальной базой пользователей, скорее всего, подвергнется атаке такого рода.

По сути, атака выполняется путем помещения метасимвола во входные данные, чтобы затем поместить SQL-команды в плоскость управления, которой там раньше не было. Этот недостаток связан с тем, что SQL не делает реального различия между плоскостями управления и данными.[6]

Для устранения данной уязвимости применен следующий комплекс решений:

- проверка авторизации пользователя перед осуществлением действий предназначенных только для авторизованных пользователей. Например запись на получение услуги или её удаление в личном кабинете пользователя;
- проверка уровня доступа пользователя перед осуществлением действий предназначенных только для пользователей с определённым уровнем доступа. Например удаление записей о персонале предприятия или создание записи о новом мастере.

3.2 Разработка клиентской части

Интернет-технологии сейчас стремительно развиваются. Невозможно представить какую-либо компанию или предприятие без собственного веб-сайта. Основное требование к веб-приложению – оно должно работать быстро, надёжно, качественно и иметь удобный и понятный для конечного пользователя интерфейс. Однако разработка качественной ИС является не лёгкой задачей. Разработка клиентской части приложения разделена на следующие этапы:

- разработка графического макета страниц;
- выбор стека технологий для разработки;
- разработка страниц веб-сайта;
- связь с серверной частью системы;
- тестирование системы в целом.

На рисунках 15-18 представлены скриншоты реализации страниц ИС.

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

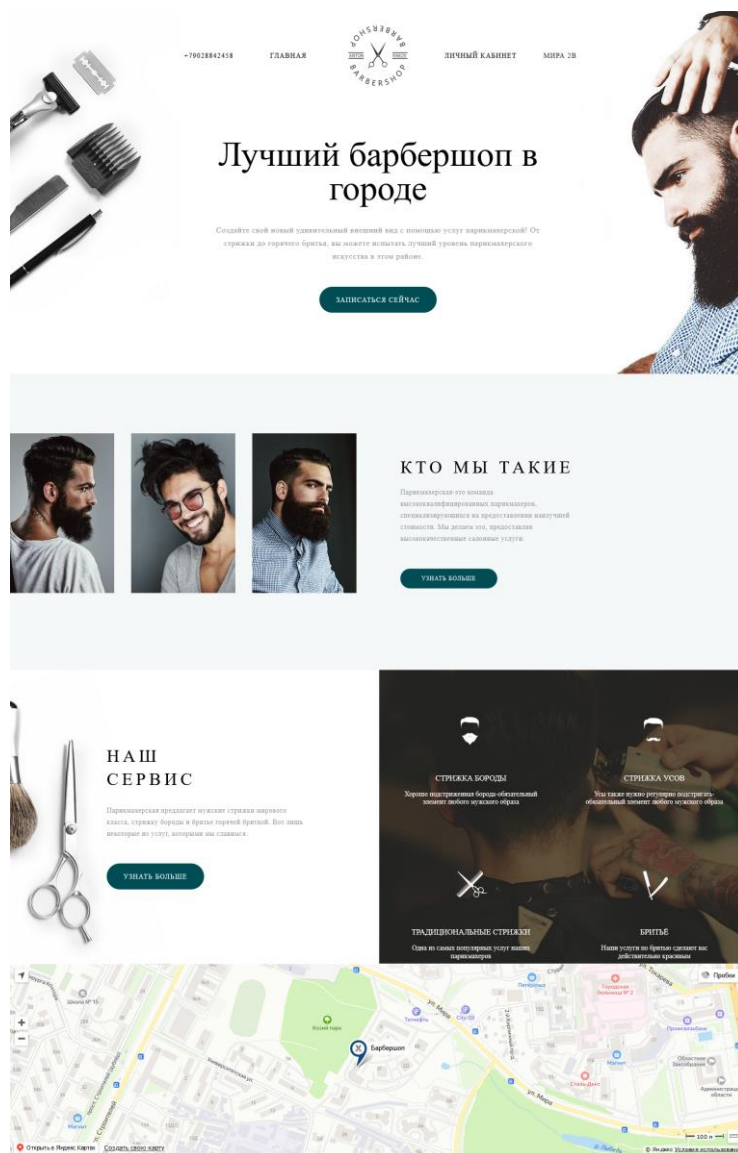


Рисунок 15 - Реализация главной страницы ИС

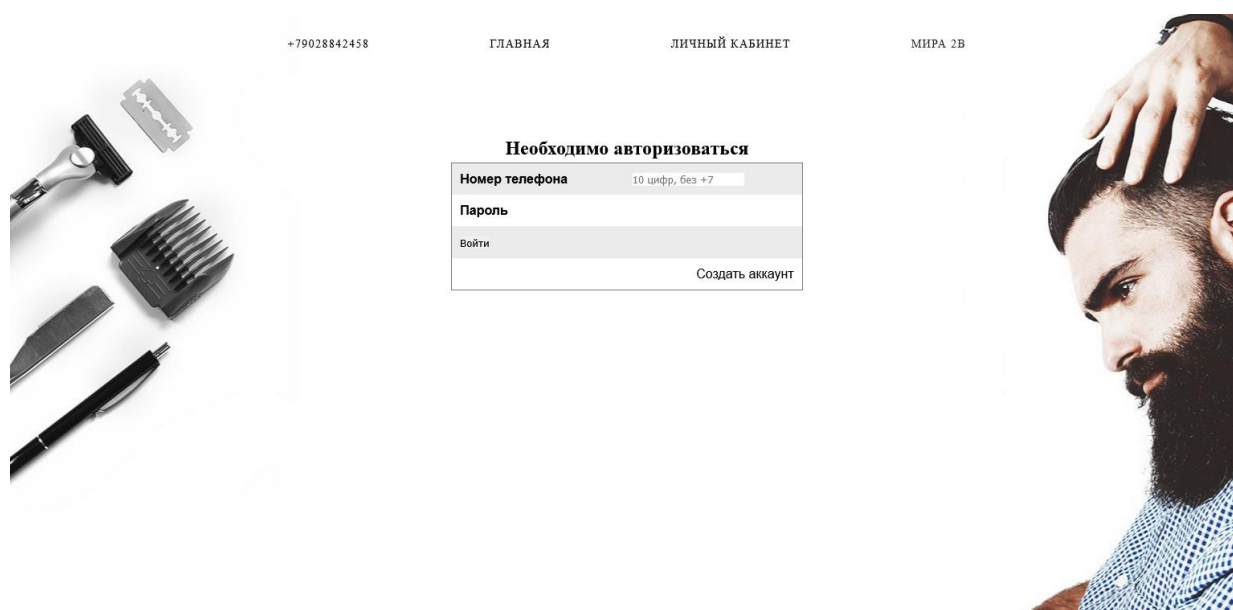


Рисунок 16 - Реализация страницы авторизации.

Изм.	Лист	№ докум.	Подпись	Дата

ВЛГУ.09.02.04.16.3.00 ПЗ

Лист

32



Панель управления

Выйти Главная

Обратный звонок

#	ФИО	Номер телефона	Действия
11	Виктор	9895432847	Удалить
12	Афанасий	9548871457	Удалить

Записи

Клиенты

Мастера

Управляющий персонал (admin)

Рисунок 17 - Реализация страницы панели управления



+79028842458 Главная Личный кабинет МИРА 2В

Текущие записи

Вы записаны:
2021-05-13 11:00:00 к барберу Большаков Андрей Юрьевич

Капин Александр Юрьевич - Опыт 3 года

2021-05-24	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
2021-05-25	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
2021-05-26	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
2021-05-27	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
2021-05-28	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
2021-05-29	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
2021-05-30	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00

Большаков Андрей Юрьевич - Опыт работы 5 лет

Степанов Михаил Кириллович - Опыт работы 6 лет

Рисунок 18 - Реализация страницы личного кабинета пользователя.

3.3 Тестирование

В ходе разработки выбран метод неформального приёмочного тестирования.

Приемочное тестирование — это последний тест перед развертыванием программного обеспечения. Приемочное тестирование делается для проверки готовности программного обеспечения выполнять задачи и функции, поставленные при разработке. Существуют три базовые стратегии выполнить приемочное тестирование, а именно:

- формальная приемка;
- неформальная приемка или альфа-тестирование;
- бета-тестирование.

При неформальном приемочном тестировании процедуры тестирования не планируются так тщательно, как при формальном приемочном тестировании. Хотя тестируемые функции и свойства определены, нет жестко определенных тестовых наборов. Тестирующий определяет метод тестирования. Этот подход менее контролируем, чем формальное тестирование, и более субъективен.

Неформальное приемочное тестирование — это наиболее распространенный подход к тестированию в организациях-заказчиках.

Ниже приведены преимущества этой формы тестирования:

- тестируемые функции и свойства известны;
- выполнение теста можно отслеживать и измерять;
- известны критерии приемлемости;
- будет обнаружено большее количество недостатков, зависящих от пользователя, чем при формальном приемочном тестировании.

Недостатки перечислены ниже:

- требуются ресурсы, планирование и управление ими;
- нельзя задавать определенные тестовые наборы;
- пользователи могут привыкнуть к системе и не увидеть недостатков;
- пользователи могут увлечься поиском новшеств в новой версии, а не недостатков;
- При тестировании не контролируются ресурсы.

Для тестирования данной системы выбрана следующая связка ПО:

- PHP Фреймворк «CODECEPTION_»;

- WebDriver (Chrome Driver);

- Selenium.

В процессе протестированы следующие функции:

- проверка наличия текста;

- проверка наличия изображений;

- проверка функционирования записи.

Конфигурационный файл:

```
actor: AcceptanceTester
modules:
  enabled:
    - WebDriver
  config:
    WebDriver:
      url: 'http://vkr/'
      window_size: 1920x1080
      browser: 'chrome'
```

На рисунках 20-22 представлены скриншоты успешного прохождения тестирования.



Рисунок 19 - Результат успешного прохождения теста на проверку текста

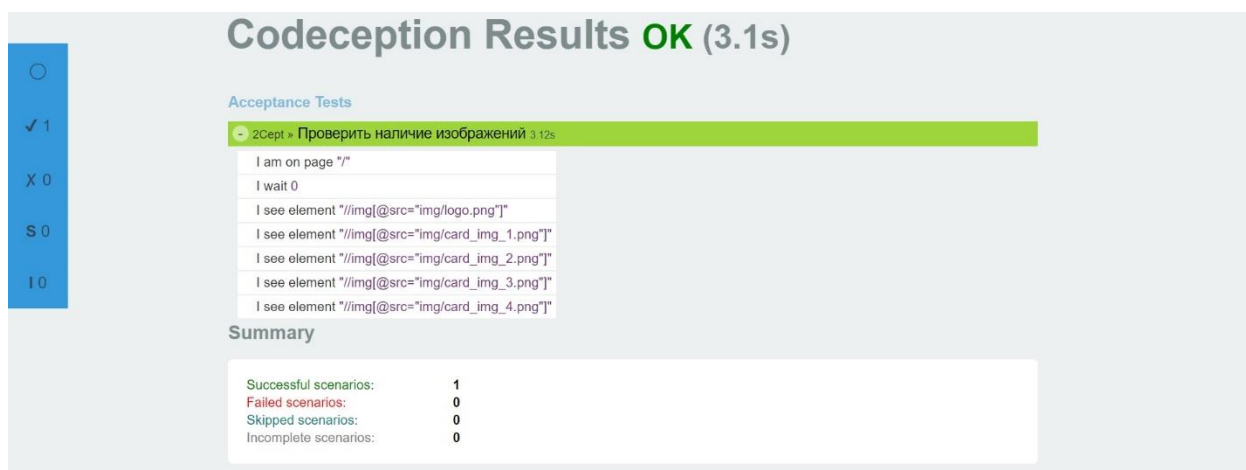


Рисунок 20 - Результат успешного прохождения теста на проверку изображений



Рисунок 21 - Результат успешного прохождения теста на проверку функционирования записи

3.4 Развёртывание

Когда разработка ИС завершена, то её нужно где-то разместить. Для этого нужно окружение развёртывания – это среда, которая в себе имеет сервер и необходимые модули. Информационная система «Барбершоп» развёрнута на локальном сервере с помощью бесплатного программного продукта OpenServer. Выбран данный продукт исходя из ключевых преимуществ:

- стоимость. Программное решение является бесплатным;
- удобство. OpenServer предоставляет все необходимые инструменты и модули для работы информационной системы;
- OpenServer является локальным сервером. Исходя из этого работа ИС зависит от ПК, на котором развёрнута система на базе данного решения, поэтому обеспечив бесперебойную работу аппаратной и программной части машины будет обеспечена бесперебойная работа ИС предприятия вне зависимости от сторонних факторов. Например если система развёрнута на стороннем VDS-сервере или хостинге, то работа ИС зависит от состояния удалённой машины, бесперебойность и стабильность которой мы гарантировать не можем;
- автономная работа без доступа к сети интернет.

Развёртывание происходит в несколько шагов которые описаны ниже:

- загрузка OpenServer по ссылке <https://ospanel.io/download/> в базовом пакете;

- установка OpenServer;
- запуск OpenServer;
- настройка OpenServer под следующие параметры.

На рисунках 23-24 представлены скриншоты параметров настройки OpenServer.

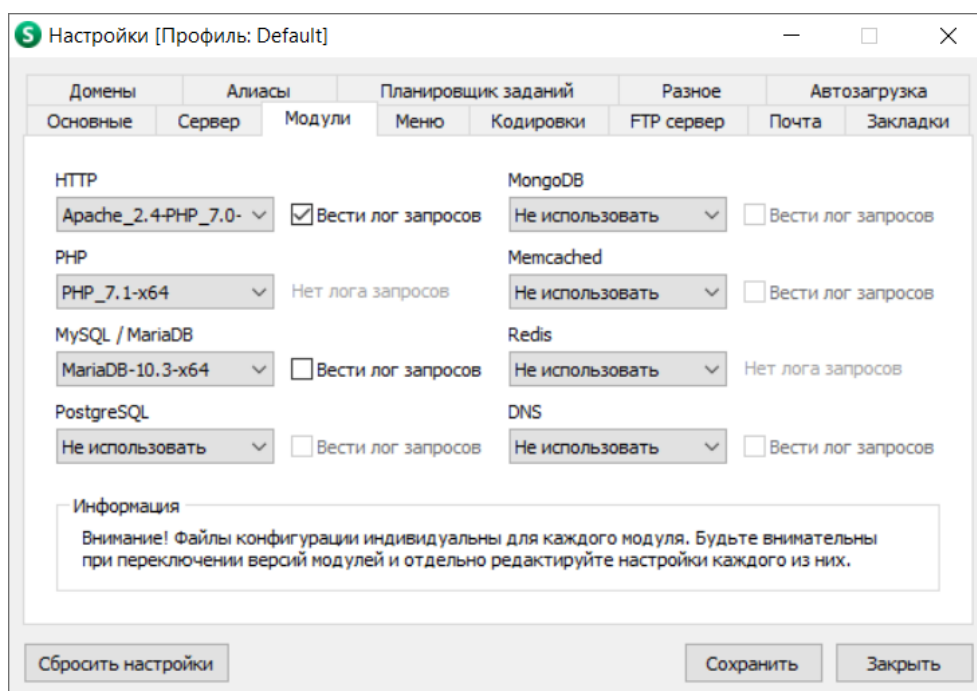


Рисунок 22 - Параметры вкладки «Модули»

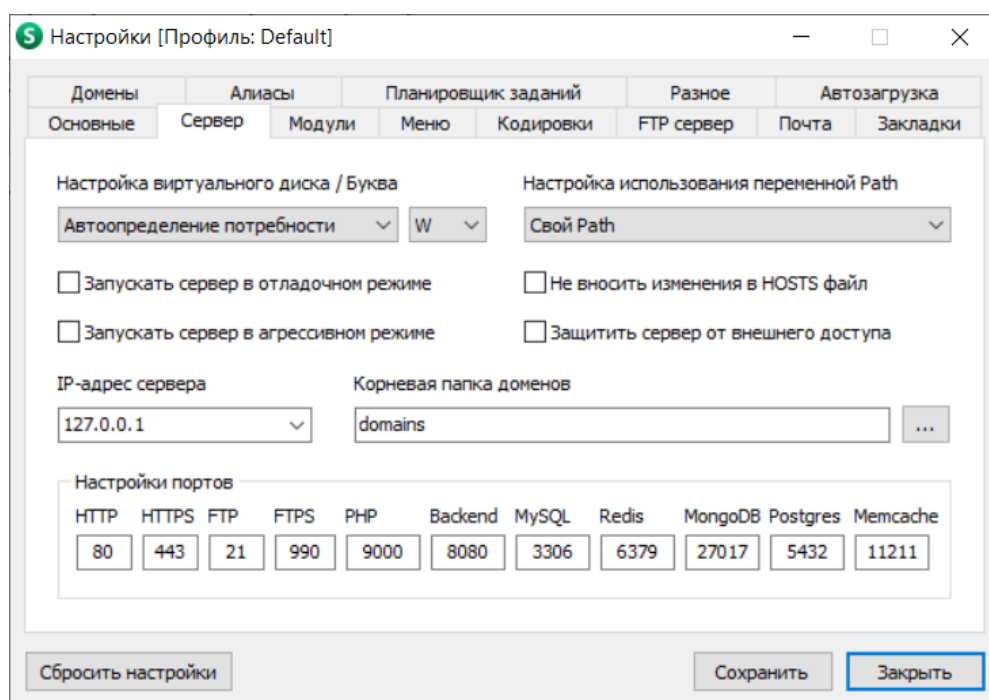


Рисунок 23 - Параметры вкладки «Сервер»

- создание базы данных в PhpMyAdmin;
- импорт файла barbershop.sql;
- создание папки «barbershop» в каталоге «domain»;
- открыть ссылку <http://barbershop/> в браузере;

Альтернативный вариант развёртывания системы это – веб-хостинг с БД. Веб-хостинг — это способ размещения сайта в сети интернет. Как только вы разместили свой сайт на сервере — кто угодно может получить доступ к нему, набрав доменное имя в строке браузера. Доступ к сайту возможен 24 часа в сутки, 7 дней в неделю, 365 дней в год.[8]

База данных может находиться как на серверах компании у которой покупался хостинг, так и на стороннем решении (например Heroku).

Сравнение перечисленных вариантов относительно локального сервера:

- надёжность и стабильность машины зависит от провайдера хостинга либо от предприятия в случае с локальным сервером;
- локальный сервер на базе программного продукта OpenServer 0руб\месяц. Минимальная стоимость хостинга 100-150 руб\месяц;
- в случае с локальным сервером компьютер необходимо держать включенным 24\7. В варианте с хостингом или выделенным сервером это необязательно.

В классическом представлении система должна находиться на каком-либо физическом носителе. Это может быть диск, флешка или винчестер. Технологии не стоят на месте и есть облачные решения, а именно GitHub. GitHub - это социальная сеть для разработчиков, и по совместительству самая большая площадка для OpenSource проектов. Позволяет хранить, делиться и разрабатывать проекты многим несвязанным между собой программистам.

Информационная система «барбершоп» опубликована на данном портале (<https://github.com/AntonRakov/vkr>). Преимущество заключается в том, что систему можно развернуть без физического носителя информации с помощью GIT и нескольких команд. Нужен только интернет.

4. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

Для расчета экономического эффекта использования информационной системы барбершопа выбрана методика функционально-стоимостного анализа.

Для проведения функционально-стоимостного анализа необходимо:

- построить функциональные модели необходимых процессов;
- подсчитать количество необходимых действий за определённый период времени;
- рассчитать базовую величину стоимости каждого процесса;
- разложить второстепенные затраты согласно базовой величине стоимости на процессы.[9]

4.1 Расчет стоимости разработки системы

Стоимость автоматизации информационной системы «Барбершоп» состоит из следующих составляющих:

- затраты на заработную плату участвующим в процессе разработки ИС;
- затраты на расходные материалы;
- затраты на амортизацию оборудования и нематериальных компонентов.

Стоимость разработки ИС автоматизации считается по следующей формуле:

$$C_{ис} = З + М + А, \quad (1)$$

где $C_{ис}$ – стоимость разработки ИС «Барбершоп», рубли;

З – затраты по заработной плате группе специалистов, задействованной в разработке системы, рубли;

М – затраты на расходные материалы, необходимые в процессе разработке системы, рубли;

А – амортизация оборудования и нематериальных активов, которые используются в процессе разработки информационной системы, рубли.

Для расчета затрат на выплату заработной платы группе специалистов, задействованным в разработке системы, целесообразно составить квалификационный план проекта разработки системы.[10]

В таблице 1 отображен квалификационный план проекта разработки системы

Таблица 1 – Квалификационный план проекта разработки системы

Наименование специалиста	Оклад, Р/мес.	З/п, Р/день (21 раб. день)	Количество человек	Выполняемые задачи и длительность выполнения (рабочие дни)
Руководитель проекта	75600	3600,00	1	Разработка требования к системе, план проекта, (18 дней)
Разработчик (программист)	70100	3338,09	1	Программирование компонентов системы (21 день)
Тестировщик	65000	3095,23	1	Тест системы (18 дней)
Аналитик	40000	1904,76	1	Проектирование диаграмм на этапах разработки технического и рабочего проектов. (18 дней)

Общая длительность выполнения проектных работ составляет:

$T_{ис} = 75$ рабочих дней.

Затраты по заработной плате рассчитываются следующим образом:

$$З = З_{зп} + СВ, \quad (2)$$

где $З_{зп}$ – заработная плата задействованных специалистов, рубли;

$СВ$ – страховые взносы в государственный фонд, рубли.

$$З_{зп} = \sum (O_i / Д * t_i), \quad i=1..n, \quad (3)$$

где n – количество задействованных специалистов, человек;

O_i – оклад i -го специалиста, рубли;

$Д$ – количество рабочих дней в месяце, дни;

t_i – время участия специалиста в проекте (количество дней), рассчитывается на основании плана проектных работ, дни.

Отчисления в Фонд оплаты труда составляют 30%:

$$СВ = З_{зп} * 0,3 \quad (4)$$

На основании сроков разработки системы и квалификационного плана выполнения проектных работ, затраты на заработную плату участвующих в разработке специалистов составят

$$З_{зп} = 3600,00 * 18 + 3338,09 * 21 + 3095,25 * 18 + 1904,76 * 18 = 224900,07\text{Р}$$

С данной суммы в фонд оплаты труда необходимо произвести отчисления в размере: $СВ = 224900,07 * 0,3 = 67470,02\text{Р}$

В итоге затраты по заработной плате составят:

$$З = 224900,07 + 67470,02 = 292370,09\text{Р}$$

Основными расходными материалами, использованными при разработке информационной системы, являются электричество, необходимое для работы ПК, и бумага. В процессе разработки ИС необходимо задействовать две единицы компьютеров. В сухом остатке компьютеры будут задействованы в проекте 552 часа. Номинальная мощность ПК составляет 250 Вт/ч.

Расчеты затрат на расходные материалы необходимо рассчитать по следующим формулам:

$$М = Б + Э, \quad (5)$$

где $М$ – стоимость использованных расходных материалов, рубли;

$Б$ – стоимость бумаги, рубли;

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

Э – стоимость электричества, рубли;

Стоимость бумаги и электричества необходимо рассчитать по нижеописанным формулам:

$$Б = К * Ц, \quad (6)$$

$$Э = Р * Ц * Т, \quad (7)$$

где К – количество упаковок бумаги, количество;

Р – мощность ПК;

Ц – цена используемого ресурса, рубли;

Результаты подсчёта затрат на расходные материалы отображены в таблице 2

Таблица 2 – Затраты на расходные материалы

Наименование	Цена, Р	Количество, ед.	Стоимость, Р (с учетом НДС = 20%)
			за Т _{ис}
Электроэнергия	4,48	0,25 кВт/ч * 552ч	741,888
Бумага	240	1 упаковка	240,00
Всего			981,88

Амортизация, входящая в формулу стоимости информационной системы - амортизация оборудования, используемая для разработки ИС.

$$A = A_1, \quad (8)$$

где А – общая амортизация;

А₁ – амортизация оборудования.

В таблице 3 показаны расчеты нормы амортизации оборудования, а на таблице 7 отображены затраты на амортизацию оборудования и нематериальных активов, которые используются в процессе разработки ИС.

Таблица 3 – Расчеты нормы амортизации оборудования

Наименование	Стоимость, Р	Срок эксплуатации, лет	Норма амортизации, Р/мес.	Норма амортизации, Р/день
Компьютер	35000	5	$35000/5/12 = 583,33$	$583,33/21/8 = 3,47$
Всего			583,33	3,47

Таблица 4 – Подсчет амортизации оборудования и нематериальных активов

Амортизационные активы	Норма амортизации, Р/день	Длительность, дней	Стоимость, Р
Компьютер 1	40,00	38	1 520,00
Компьютер 2	40,00	37	1480,00
Всего			3000,00

Исходя из рассчитанных данных, стоимость разработки ИС автоматизации «Барбершопа» составляет:

$$C_{ис} = 292370,09 + 981,88 + 3000,00 = 204\,902,8 = 296351,97\text{Р}$$

4.2 Расчет стоимости выполнения процесса до автоматизации

Использование метода ФСА в целях оценки экономической эффективности следует начать с построения функциональной модели процесса, для которого необходимо провести расчёты.

До момента автоматизации предприятия процесс осуществления деятельности барбершопа длится из расчёта 31 рабочего дня и состоит из следующих действий:

- создание расписания с учётом рабочего графика;
- проверка наличие свободного интервала времени на оказание услуги по запросу клиента;
- создание записи в журнале расписания;

- контроль корректности расписания;

В данном случае осуществление деятельности барбершопа выполняется одним менеджером.

Подсчитаем стоимость процесса ведения деятельности барбершопа до автоматизации. В целях определения стоимости всего процесса, нужно подсчитать стоимость каждой единицы операции процесса, которая в любом случае складывается из затрат на расходники, амортизацию оборудования и нематериальных активов, и конечно из расходов на заработную плату команде специалистов, выполняющих операции процесса.

Из этого следует, затраты на проведение процесса до автоматизации подсчитываются по данной формуле:

$$C_{до} = \sum Z_{Oi} + \sum M_{Oi} + \sum A_{Oi}, \quad i=1..n, \quad (9)$$

где n – количество операций в процессе;

Z_{Oi} – заработная плата работника при выполнении i -ой действия;

M_{Oi} – затраты на расходники, нужные для выполнения i -ой операции;

A_{Oi} – амортизация оборудования и нематериальных активов для i -ой операции.

Время выполнения данных действий рассчитывалось из учета следующих данных:

- среднее количество записей, принимаемое менеджером на рабочем месте за 31 день равна 1600;
- длительность приема одной записи менеджером равна 0,05 ч;
- среднее число запросов клиентов к менеджеру за 31 день равна 700;
- среднее количество записей в запросе клиента равна 1;
- длительность проверки наличия одного свободного временного периода равна 0,05 ч;
- длительность обработки документов о количестве времени для оказания услуг равна 0,04 ч;
- длительность заполнения менеджером бумажного журнала по оказанию услуг для одного запроса клиента равна 0,004 ч.

В таблице 5 представлен расчет затрат на заработную плату сотрудникам, выполняющим процесс до автоматизации

Таблица 5 – Расчет затрат на заработную плату сотрудникам, выполняющим процесс до автоматизации

Операция	Время выполнения операции, час	з\п, Р/час	Количество сотрудников, чел	Затраты с учетом СВ
О ₁ -Принять запрос на запись	$1600 * 0,05 = 80$	$35000/31/8 = 141,12$	1	$16666,40 + 4999.92 = 21666,32$
О ₂ -Проверить наличие свободного времени	$700 * 1 * 0,05 = 35$	$35000/31/8 = 141,12$	1	$7291,55 + 2187.47 = 9479,02$
О ₃ -Обработать документы о количестве свободного времени	$700 * 1 * 0,04 = 14$	$35000/31/8 = 141,12$	1	$2916,62 + 874.99 = 3791,61$
О ₄ -Заполнить бумажный бланк по оказанию услуг	$700 * 0,004 = 1,4$	$35000/31/8 = 141,12$	1	$291,66 + 87.5 = 379,16$
Сумма				46690,93

При подсчетах затрат на расходные материалы нужно обратить внимание на следующие данные:

- стоимость 1 упаковки бумаги объемом 500 единиц (листов) составляет 240Р;
- потребляемая мощность компьютера $P=0,25$ кВт/ч; при выполнении операции О₁, О₂, О₃ задействован 1 компьютер.

В таблице 6 предоставлен подсчет затрат на расходные материалы до момента автоматизации

Таблица 6 – Расчет затрат на расходные материалы до автоматизации

Наименование операции	Затрачиваемые ресурсы	Количество, ед.	Время выполнения операции, час	Цена, Р.	Затраты, Р
О ₁ – Принять запрос на запись	электроэнергия	0,25кВ/ч	58,8	4,48	$4,48 * 14,7 = 65,85$

Наименование операции	Затрачиваемые ресурсы	Количество, ед.	Время выполнения операции, час	Цена, Р.	Затраты, Р
О ₂ – Проверить наличие свободного времени	электроэнергия	0,25кВ/ч	41,58	4,48	4,48 * 10,39 =4 6,54
О ₃ –Обработать документы о количестве свободного времени	электроэнергия	0,25кВ/ч	21,42	4,48	4,48 * 5,35 = 23,96
О ₄ – Заполнить бумажный бланк по оказанию услуг	бумага	700 листов	–	0,466	0,466 * 700 = 325,20
Всего					771,55
НДС (20%)					154,31

В таблицах 7, 8 приведены подсчеты нормы амортизации и затрат на амортизацию оборудования и нематериальных активов, которые используются в процессе осуществления работы барбершопа до момента автоматизации.

Таблица 7 – Расчеты норм амортизации оборудования

Наименование	Стоимость, Р.	Срок эксплуатации, лет	Норма амортизации, Р/мес.	Норма амортизации, Р/час
Компьютер	35000	5	35000/5/12 = 583,33	583,33/21/8 = 3,47
Всего			583,33	3,47

Таблица 8 – Расчет амортизации оборудования и нематериальных активов

Наименование действия	Амортизационные компоненты	Кол-во, шт.	Норма амортизации, Р/час	Длительность, час	Стоимость, Р
О ₁ – Принять запрос на запись	Компьютер	1	5	58,8	294,00
О ₂ – Проверить наличие времени	Компьютер	1	5	41,58	207,90
О ₃ – Обработать запись на получение услуги	Компьютер	1	5	21,42	107,10
Сумма					609,00

Опираясь на данные, приведенные в таблицах 5-8, стоимость процесса осуществления деятельности барбершопа до автоматизации составит:

$$C_{\text{до}} = 46690,93 + 925,86 + 609,00 = 48225,79 \text{ (руб./месяц)}.$$

4.3 Расчет стоимости выполнения системы после автоматизации

Сейчас рассмотрим идентичный процесс после внедрения ИС автоматизации деятельности барбершопа.

После внедрения ИС автоматизации длительность процесса осуществления деятельности барбершопа равняется 31 дню. Разрабатываемая система даст возможность менеджеру сократить рассматриваемый процесс всего до двух операций:

- генерация таблицы расписание;
- создание записи.

Это стало возможно с помощью того, что система самостоятельно ведет подсчет количества свободного времени после записи других клиентов, так же благодаря концентрации данных и информации в едином месте, менеджеру легче понимать общую картину движений записей, так же у менеджера есть возможность быстро получить и извлекать необходимые данные о клиенте для успешной работы с ним.

Так как данная система имеет круглосуточный режим функционирования то для определения стоимости всего процесса, необходимо рассчитать стоимость каждой операции процесса, которая в любом варианте складывается из затрат на расходные материалы и компоненты, а также амортизацию оборудования и нематериальных активов, затрат на заработную плату специалистов, выполняющих операции процесса за 31 день. [11]

Опираясь на сложившийся результат, затраты на выполнение процесса после автоматизации ИС рассчитываются по данной формуле:

$$C_{\text{после}} = \sum Z_{O_i} + \sum M_{O_i} + \sum A_{O_i}, \quad i=1..n, \quad (10)$$

где n – количество операций в процессе, количество;

Z_{oi} – заработная плата сотрудника при выполнении i -ой операции, рубли;

M_{oi} – затраты на расходные материалы, необходимые при выполнении i -ой операции, рубли;

A_{oi} – амортизация оборудования и нематериальных активов для i -ой операции, рубли;

Время выполнения выделенных операций рассчитывалось с учетом из следующих данных:

- среднее количество записей, принимаемое менеджером на рабочем месте за 31 день равно 1600;
- длительность приема одной записи менеджером равна 0,05 ч;
- среднее число запросов клиентов к менеджеру за 31 день равно 700;
- среднее количество записей в запросе клиента равно 1;
- длительность проверки наличия одного свободного временного периода равна 0,05 ч;
- длительность обработки документов о количестве времени для оказания услуг равна 0,04 ч;
- длительность заполнения менеджером бумажного журнала по оказанию услуг для одного запроса клиента равна 0,004 ч.

В таблице 9 представлен расчет затрат на заработную плату сотрудникам, выполняющим процесс после автоматизации

Таблица 9 – Расчет затрат на заработную плату сотрудникам, выполняющим процесс после автоматизации

Операция	Время выполнения операции, час	з\п, Р/час	Количество сотрудников	Затраты, Р с учетом СВ
О ₁ -Принять запрос	$1600 * 0,02 = 32$	$35000/21/8 = 208,33$	1	$6666,56 + 1999.97 = 8666,53$

Операция	Время выполнения операции, час	з\п, Р/час	Количество сотрудников	Затраты, Р с учетом СВ
О ₂ -Проверить наличие свободного времени	$700 * 1 * 0,02 = 14$	$35000/21/8 = 208,33$	1	$2916,62+874.99=3791,61$
Всего				19041,35

При расчетах затрат на расходные материалы необходимо учесть следующие данные:

- потребляемая мощность компьютера $P=0,25$ кВт/ч;
- при выполнении операции О₁, О₂, задействован 1 компьютер.

В таблице 10 представлены затраты на расходные материалы после автоматизации

Таблица 10 – Расчет затрат на расходные материалы после автоматизации

Наименование операции	Затрачиваемые ресурсы	Количество, ед.	Время выполнения операции, час	Цена, Р	Затраты, Р
О ₁ – Принять кофемашин	электроэнергия	0,25кВ/ч	29,4	4,48	$4,48 * 7,35 = 32,92$
О ₂ – Проверить наличие комплектующих ПК	электроэнергия	0,25кВ/ч	18,9	4,48	$4,48 * 4,725 = 21,16$
О ₃ – Совершить Оказание услуг	электроэнергия	0,25кВ/ч	0,63	4,48	$4,48 * 0,15 = 0,672$
О ₄ – Выдать кофемашин	упаковочный пакет	700	—	0,5	$700*0,5 = 350,00$
Всего					404,752
НДС (20%)					80.95

В таблицах 11, 12 приведены расчеты норм амортизации и затрат на амортизацию оборудования и нематериальных активов, используемых в процессе осуществления деятельности Магазин кофемашин после автоматизации.

Таблица 11 – Расчеты норм амортизации оборудования и программного обеспечения

Наименование	Стоимость, Р	Срок эксплуатации, лет	Норма амортизации, Р/мес.	Норма амортизации, Р/час
Компьютер	35000	5	$35000/5/12 = 583,33$	$583,33/21/8 = 3,47$
Всего			583,33	3,47

Таблица 12 – Расчеты амортизации оборудования и нематериальных активов

Наименование операции	Амортизационные активы	Кол-во, шт.	Норма амортизации, Р/час	Длительность, час	Стоимость, Р
О ₁ – Принять запрос	Компьютер	1	5	29,4	147,00
Наименование операции	Амортизационные активы	Кол-во, шт.	Норма амортизации, Р/час	Длительность, час	Стоимость, Р
О ₂ – Проверить наличие свободного времени	Компьютер	1	5	18,9	94,5
Всего					273,00

$$A_{O_i} = 270,76 + 950 = 1223,00$$

Исходя из данных, приведенных в таблицах 10 – 12, стоимость процесса осуществления деятельности барбершопа после автоматизации составляет:

$$C_{\text{после}} = 19041,35 + 485,702 + 1223,00 = 20750,052 \text{ (Р/месяц)}.$$

4.4 Расчет показателей эффективности системы

На основании вышеизложенных расчетов, определим экономический эффект от автоматизации процесса осуществления деятельности барбершопа.

Экономический эффект рассчитывается по формуле

$$\Xi = (C_{\text{до}} - C_{\text{после}}) * Ч, \quad (11)$$

где Ч – число месяцев работы;

В году 12 месяцев Ч = 12.

$$\Xi = (48225,79 - 20750,052) * 12 = 329708,86\text{Р}.$$

В результате получаем, что экономический эффект от автоматизации процесса осуществления деятельности Магазин кофемашин не менее 329708,86Р в год.

Исходя из полученных результатов расчета годового экономического эффекта, можно рассчитать коэффициент экономической эффективности:

$$E = \Xi / C_{\text{ис}} \quad (12)$$

где Ξ – годовой экономический эффект;

$C_{\text{ис}}$ – стоимость разработки системы.

$$C_{\text{ис}} = 329708,86 / 296351,97 \approx 1,11$$

Срок окупаемости данного проекта можно рассчитать по формуле:

$$T = 1 / C_{\text{ис}}, \quad (13)$$

Таким образом, срок окупаемости данного проекта составляет

$$T = 1 / 1,11 = 0,9 \text{ года}.$$

Рассчитанные показатели свидетельствуют об экономической эффективности проектируемой системы автоматизации системы «Барбершоп»

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы разработана информационная система предприятия «Барбершоп», которое специализируется на предоставлении парикмахерских услуг населению. Информационная система предоставляет клиенту информацию о предприятии, а также предоставляет возможность записаться на услугу в режиме онлайн.

Клиент имеет возможность выбрать необходимую услугу, мастера, день и время, а также просматривать информацию об имеющихся записях в его личном кабинете, регистрация в котором происходит в несколько кликов. Нужно ввести только номер телефона, имя и пароль.

Если у клиента возникают вопросы по работе предприятия или ему нужна помощь в работе с системой, то на этот случай есть форма обратного звонка на главной странице. Клиент оставляет контактную информацию и администратор предприятия связывается с ним по телефону и осуществляет консультацию по интересующим вопросам.

В данной информационной системе реализован функционал для администратора предприятия, который затрагивает все необходимые случаи использования системы. Администратор имеет возможность работы с заявками на обратный звонок, мастерами, расписанием, клиентами и администраторами.

В ходе работы проанализированы бизнес-процессы и функционал, необходимый пользователям ИС для реализации данного проекта. ИС разработана с учетом масштабирования и возможностью добавления нового функционала.

В процессе реализации рассмотрены различные технические решения поставленной задачи. Произведен выбор платформ и архитектуры для реализации всех компонентов.

					<i>ВлГУ.09.02.04.16.3.00 ПЗ</i>	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

В результате выполнения данного проекта были решены задачи, поставленные в начале работы. Информационная система не является совершенной на данный момент. Дальнейший план развития предполагает создание мобильного приложения для ОС Android и IOS, а также рефакторинг ИС для соответствия SPA-архитектуре.

					<i>ВлГУ.09.02.04.16.3.00 ПЗ</i>	<i>Лист</i>
						53
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Разработка системы управления электронным документооборотом на примере ООО "Курортное". [Электронный ресурс]. - Режим доступа: (<http://bibliofond.ru>)

2) Проектирование и разработка WEB приложения. [Электронный ресурс]. - Режим доступа: (<http://bibliofond.ru>)

3) Система управления базами данных MySQL [Электронный ресурс]. - Режим доступа: (<https://depix.ru>)

4) Прикладные информационные системы. [Электронный ресурс]. - Режим доступа: (<http://elibrary.ru>)

5) SQL Injection от А до Я | Codeby.net - Форум программистов. [Электронный ресурс]. - Режим доступа: (<https://codeby.net>)

6) Chapter 2 Cybersecurity Risk Management, Risk Assessment and Asset Evaluation | Protection of Transportation Infrastructure from Cyber Attacks: A Primer | The National Academies Press. [Электронный ресурс]. - Режим доступа: (<https://nap.edu>)

7) Проектирование информационной системы учета научных публикаций в среде Adobe Dreamweaver. [Электронный ресурс]. - Режим доступа: (<http://refleader.ru>)

8) Возможности виртуального выделенного сервера в поддержке сайта компании. [Электронный ресурс]. - Режим доступа: (<http://elibrary.ru>)

9) Разработка автоматизированной информационной системы обработки заявок [Электронный ресурс]. - Режим доступа: (<http://revolution.allbest.ru>)

10) Проектирование ИС. Экономическая эффективность. [Электронный ресурс]. - Режим доступа: (<https://otherreferats.allbest.ru>)

11) Online портал в сфере услуг — план разработки. [Электронный ресурс]. - Режим доступа: (<http://turboreferat.ru>)

12) PHP: Hypertext Preprocessor [Электронный ресурс]. - Режим доступа: (<https://www.php.net/>)

13) RedBeanPHP :: Welcome [Электронный ресурс]. - Режим доступа: (<https://redbeanphp.com/index.php>)

14) MySQL [Электронный ресурс]. - Режим доступа: (<https://www.mysql.com/>)

15) Open Server Panel - Локальный веб-сервер для Windows [Электронный ресурс] - . Режим доступа: (<https://ospanel.io/>)

ПРИЛОЖЕНИЕ А.

Код реализации информационной системы.

А.1 Модуль работы с базой данных

Подключение базы данных

```
<?php
require 'lib/rb.php';
R::setup( 'mysql:host=localhost;dbname=barbershop','root',
'' );
if ( !R::testconnection() )
{
    exit ('Нет соединения с базой данных');
}
session_start();
```

А.2 Код реализация главной страницы

```
<?php
require 'db.php';
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="css/styles.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.m
in.js"></script>
    <script src="js/modal.js"></script>
    <title>Главная</title>
</head>
<body>
    <div class="page-wrapper">
        <header>
            <div class="menu">
                <div class="menu_left">
                    <ul>
                        <li><a
href="tel:+79028842458">+79028842458</a></li>
                        <li><a href="#">Главная</a></li>
```

```

        </ul>
    </div>
    <div class="menu_center">
        
    </div>
    <div class="menu_right">
        <ul>
            <li><a href="lk.php">Личный
кабинет</a></li>
            <li>Мира 2В</li>
        </ul>
    </div>
</div>
<div class="header_bottom">
    <h1>Лучший барбершоп в городе</h1>
    <p>Создайте свой новый удивительный внешний вид с
помощью услуг парикмахерской! От стрижки до горячего бритья, вы
можете испытать лучший уровень парикмахерского искусства в этом
районе.</p>
    <a class="btn trigger" href="#">Записаться
сейчас</a>
</div>
</header>
<main>
    <div class="block1">
        <div class="left_block">
            
            
            
        </div>
        <div class="right_block">
            <div class="text_block">
                <h2>Кто мы такие</h2>
                <p>Парикмахерская-это команда
высококвалифицированных парикмахеров, специализирующихся на
предоставлении наилучшей стоимости. Мы делаем это, предоставляя
высококачественные салонные услуги.</p>
                <a href="about.html" class="btn
trigger">Узнать больше</a>
            </div>
        </div>
    </div>
    <div class="block2">
        <div class="left_block">
            <h2>Наш <br/> сервис</h2>
            <p>Парикмахерская предлагает мужские стрижки
мирового класса, стрижку бороды и бритье горячей бритвой. Вот лишь
некоторые из услуг, которыми мы славимся.</p>
            <a href="about.html" class="btn
trigger">Узнать больше</a>
        </div>
        <div class="right_block">
            <div id="beard_trim">

```



```

70x62.png" alt="">
        Стрижка бороды</a>
        <p>Хорошо подстриженная борода-
обязательный элемент любого мужского образа</p>
    </div>
    <div id="mustache_trim">
        
        <a href="#">Стрижка усов</a>
        <p>Усы также нужно регулярно подстригать-
обязательный элемент любого мужского образа</p>
    </div>
    <div id="traditional_haircuts">
        
        <a href="#">Традиционные стрижки</a>
        <p>Одна из самых популярных услуг наших
парикмахеров</p>
    </div>
    <div id="shaves">
        
        <a href="#">Бритьё</a>
        <p>Наши услуги по бритью сделают вас
действительно красивым</p>
    </div>
</div>
<div class="block3">

</div>
<div class="block4">

</div>
</main>
<footer>
    <div class="pages">

    </div>
    <div class="description">

    </div>
    <div class="follow_us">

    </div>
</footer>
<!-- Модальное окно -->
<div class="modal-wrapper">
    <div class="modal">
        <div class="head">
            <p>Форма обратного звонка</p>
            <a class="btn-close trigger" href="#">
                <i class="fa fa-times" aria-hidden="true">x</i>

```

```

        </a>
    </div>
    <div class="content">
        <div class="good-job">
            <i class="fa fa-thumbs-o-up" aria-
hidden="true"></i>
            <form action="" method="POST">
                <p>Как к Вам обращаться</p>
                <input type="text" name="name"
class="input_text">
                <p>Номер телефона</p>
                <input type="tel" name="phone_number"
placeholder="10 цифр, без +7" pattern="\s?[\(\){0,1}9[0-
9]{2}[\)]{0,1}\s?\d{3}[-]{0,1}\d{2}[-]{0,1}\d{2}"
class="input_text">
                <input type="submit" id="submit_form">
            </form>
            <?php
                $name = $_POST['name'];
                $phone_number = $_POST['phone_number'];

                if (!empty($name) &&
!empty($phone_number)) {
                    $requests = R::dispense('requests');
                    $requests->fio = $name;
                    $requests->phone_number =
$phone_number;
                    R::store($requests);
                }

            ?>
        </div>
    </div>
</div>
</div>
</div>
<iframe src="https://yandex.ru/map-
widget/v1/?um=constructor%3Aa5a5a41b4c5755a46c19cd550aca0567d335
f7185e8896558c909f2e7617676e&source=constructor" width="100%"
height="400" frameborder="0"></iframe>
</div>
</body>
</html>

```

А.3 Код реализация страницы записи

```

<?php
    require 'db.php';

    ?>

    <?php if (((!empty($_SESSION['logged_user'])) AND
($_SESSION['logged_user']['access_level'] == '1') )OR
(!empty($_SESSION['logged_user'])) AND
($_SESSION['logged_user']['access_level'] == '2')) : ?>

```

					ВЛГУ.09.02.04.16.3.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		59

```

        <meta http-equiv="refresh" content="0;URL=admin.php" />

<?php elseif ($_SESSION['logged_user']['access_level'] == '0') :
?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Личный кабинет</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <main class="auth_page">
        <div class="second-menu">
            <ul>
                <li><a
href="tel:+79028842458">+79028842458</a></li>
                <li><a href="index.php">Главная</a></li>
                <li><a href="lk.php">Личный кабинет</a></li>
                <li>Мира 2В</li>
            </ul>
        </div>
        <div class='spoiler-wrap enable'>
            <div class='spoiler-head'>Текущие записи</div>
            <div class='spoiler-body'>
                <?php
                    $id_client = $_SESSION['logged_user']['id'];
                    if(R::count('records', "client = ?",
array($id_client)) < 1){
                        echo("Записи на текущий момент
отсутствуют");
                    }
                    else{
                        $active_records = R::findAll('records',
"client = ?", array($id_client));
                        foreach($active_records as $item){
                            $date_record = $item['date_record'];
                            $date_record = R::findOne('days',
"id = ?", array($date_record));
                            $date_record = $date_record['day'];
                            $time_record = $item['time_record'];
                            $time_record = R::findOne('times',
"id = ?", array($time_record));
                            $time_record =
$time_record['timing'];
                            $master_record = $item['master'];
                            $master_record =
R::findOne('masters', "id = ?", array($master_record));
                            $master_record =
$master_record['fio'];

```

```

                                echo("Вы записаны: <br>" .
$date_record . " " . $time_record . " " . "к барберу " .
$master_record . "<br>");
                                }
                                }
                                ?>
                                </div>
                                </div>
                                <?php
                                $masters = R::find('masters');
                                foreach ($masters as $item){
                                    $id_master = $item['id'];
                                    echo"<div class='spoiler-wrap disabled'>";
                                    echo"<div class='spoiler-head'>" .
$item['fio'] . " - " . $item['biography'] . "</div>";
                                    echo"<div class='spoiler-body'>";
                                    $days = R::find('days');
                                    foreach ($days as $item){
                                        $day = $item['day'];
                                        $id_day = $item['id'];
                                        echo"<table>";
                                        echo"<tr>";
                                        if ($day >= date("Y-m-d")){
                                            if($day < date("Y-m-d",
strtotime("+7 days"))){
                                                echo"<td>" . $day . "</td>";
                                            }
                                            $times = R::find('times');
                                            foreach ($times as $item){
                                                $id_time = $item['id'];
                                                $record =
R::findOne('records','date_record=? AND time_record=? AND
master=?',array($id_day,$id_time,$id_master));
                                                $record = $record['id'];
                                                $time = $item['timing'];
                                                if ($record != ''){
                                                    if($day < date("Y-m-d",
strtotime("+7 days"))){
                                                        echo"<td
style='text-decoration: line-through'>" . mb_substr($time, 0, 5)
. "</td>";
                                                    }
                                                    }
                                                    else{
                                                        if($day < date("Y-m-d",
strtotime("+7 days"))){
                                                            echo"<td><a
href='php/create_record.php?time=" . $id_time . "&day=" .
$id_day . "&master=" . $id_master . "'>" . mb_substr($time, 0,
5) . "</a></td>";
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                        echo"</tr>";

```

```

        echo"</table>";
    }
    }
    echo"</div>";
    echo"</div>";
}
?>

</main>
<script src="http://code.jquery.com/jquery-latest.min.js"
type="text/javascript"></script>
<script src="js/spoiler.js" type="text/javascript"></script>
</body>
</html>

<?php else : ?>
    <meta http-equiv="refresh"
content="0;URL=authorization.php"/>
<?php endif; ?>

```

А.4 Код реализация авторизации пользователя

```

<?php
    require 'db.php';
?>

<?php if (((!empty($_SESSION['logged_user'])) AND
($_SESSION['logged_user']['access_level'] == '1') )OR
(!empty($_SESSION['logged_user'])) AND
($_SESSION['logged_user']['access_level'] == '2')) : ?>
    <meta http-equiv="refresh" content="0;URL=admin.php" />

<?php elseif (!empty($_SESSION['logged_user']) AND
($_SESSION['logged_user']['access_level'] == '0')) : ?>
    <meta http-equiv="refresh" content="0;URL=lk.php" />

<?php else : ?>
    <?php

        $data = $_POST;
        if ( isset($data['do_login']) )
        {
            $user = R::findOne('users', 'phone_number = ?',
array($data['phone_number']));

            if ( $user )
            {
                //логин существует
                if ( password_verify($data['password'],
$user->password) )
                {
                    //если пароль совпадает, то нужно
авторизовать пользователя

                    $_SESSION['logged_user'] = $user;

```

```

                                echo '<meta http-equiv="refresh"
content="0;URL=admin.php" />';
                                // header(
"refresh:2;url=recording.php" );
                                }else
                                {
                                    $errors[] = 'Неверно введен пароль
и/или логин';
                                }

                                }else
                                {
                                    $errors[] = 'Неверно введен логин и/или
пароль';
                                }

                                }

                                ?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Авторизация</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <main class="auth_page">
        <div class="second-menu">
            <ul>
                <li><a
href="tel:+79028842458">+79028842458</a></li>
                <li><a href="index.php">Главная</a></li>
                <li><a href="lk.php">Личный кабинет</a></li>
                <li>Мира 2В</li>
            </ul>
        </div>
        <div class="auth_form">
            <div>
                <h2>Необходимо авторизоваться</h2>
            </div>
            <div>
                <form action="authorization.php" method="POST">
                    <table>
                        <tr>
                            <td><strong>Номер
телефона</strong></td>
                            <td><input type="text"
name="phone_number" value="<?php echo @$data['phone_number']; ?>"
placeholder="10 цифр, без +7" pattern="\s?[\\]{0,1}9[0-
9]{2}[\\]{0,1}\\s?\\d{3}[-]{0,1}\\d{2}[-]{0,1}\\d{2}"></td>

```

```

        </tr>
        <tr>
            <td><strong>Пароль</strong></td>
            <td><input type="password"
name="password" value="<?php echo @$data['password']; ?>"></td>
        </tr>
        <tr>
            <td colspan="2"><button type="submit"
name="do_login">Войти</button></td>
        </tr>
        <tr>
            <td colspan="2" class="singup_link"><a
href="singup.php">Создать аккаунт</a></td>
        </tr>
        <?php
        if ( ! empty($errors) )
        {
            //выводим ошибки авторизации
            echo "<tr><td colspan='2'" .
array_shift($errors) . "</td></tr>";
        }
        ?>
    </table>
</form>
</div>
</div>
</main>
</body>
</html>
<?php endif; ?>

```

A.5 Код реализации панели администратора

```

<?php
    require 'db.php';
?>

<?php if (((!empty($_SESSION['logged_user'])) AND
($_SESSION['logged_user']['access_level'] == '1') )OR
(!empty($_SESSION['logged_user'])) AND
($_SESSION['logged_user']['access_level'] == '2'))): ?>

    <!-- Основная страница -->
    <!DOCTYPE html>
    <html lang="ru">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <link rel="stylesheet" href="css/styles.css">
        <title>Панель управления</title>
    </head>
    <body>
        <div class="admin_panel">

```

```

<h1>Панель управления</h1>
<div id="logout">
    <a href="logout.php">Выйти</a>
    <a href="index.php">Главная</a>
</div>
<div class="spoiler-wrap disabled">
    <div class="spoiler-head">Обратный звонок</div>
    <div class="spoiler-body">
        <table>
            <tr>
                <td>#</td>
                <td>ФИО</td>
                <td>Номер телефона</td>
                <td>Действия</td>
            </tr>
            <?php
                $users = R::findAll('requests');
                foreach ($users as $item){
                    echo"<tr>";
                    echo"<td>" . $item['id'] .
"</td>";
                    echo"<td>" . $item['fio'] .
"</td>";
                    echo"<td>" . $item['phone_number']
. "</td>";
                    echo"<td><a
href='php/delete.php?id=". $item['id']
."&table=requests">Удалить</a></td>";
                    echo"</tr>";
                }
            ?>
        </table>
    </div>
</div>
<div class="spoiler-wrap disabled">
    <div class="spoiler-head">Записи</div>
    <div class="spoiler-body">
        <table>
            <tr>
                <td>#</td>
                <td>Клиент</td>
                <td>Дата</td>
                <td>Время</td>
                <td>Мастер</td>
                <td>Услуга</td>
                <td>Действия</td>
            </tr>
            <?php
                $records = R::findAll('records',' ORDER BY
date_record, time_record' );
                foreach ($records as $item){
                    $client = R::findOne('users', "id =
?", array($item['client']));
                    $client = $client['name'];

```



```

        $date_record = R::findOne('days', "id
= ?", array($item['date_record']));
        $date_record = $date_record['day'];
        $time_record = R::findOne('times', "id
= ?", array($item['time_record']));
        $time_record = $time_record['timing'];
        $master = R::findOne('masters', "id =
?", array($item['master']));
        $master = $master['fio'];
        echo"<tr>";
        echo"<td>" . $item['id'] .
"</td>";

        echo"<td>" . $client . "</td>";
        echo"<td>" . $date_record .
"</td>";

        echo"<td>" . $time_record .
"</td>";

        echo"<td>" . $master . "</td>";
        echo"<td>" . $item['service'] .
"</td>";

        echo"<td><a
href='php/delete.php?id=". $item['id']
."&table=records">Удалить</a></td>";
        echo"</tr>";
    }
    ?>
</table>
</div>
</div>
<div class="spoiler-wrap disabled">
    <div class="spoiler-head">Клиенты</div>
    <div class="spoiler-body">
        <table>
            <tr>
                <td>#</td>
                <td>Имя</td>
                <td>Номер телефона</td>
            </tr>
            <?php
                $users = R::find('users', ' access_level <
1 ');

                foreach ($users as $item){
                    echo"<tr>";
                    echo"<td>" . $item['id'] .
"</td>";

                    echo"<td>" . $item['name'] .
"</td>";

                    echo"<td>" . $item['phone_number']
. "</td>";

                    echo"</tr>";
                }
            ?>
        </table>
    </div>

```

```

</div>
<div class="spoiler-wrap disabled">
  <div class="spoiler-head">Мастера</div>
  <div class="spoiler-body">
    <table>
      <tr>
        <td>#</td>
        <td>ФИО</td>
        <td>Номер телефона</td>
        <td>Биография</td>
        <td>Действия</td>
      </tr>
    <?php
      $users = R::findAll('masters');
      foreach ($users as $item){
        echo"<tr>";
        echo"<td>" . $item['id'] .
"</td>";
        echo"<td>" . $item['fio'] .
"</td>";
        echo"<td>" . $item['phone_number']
. "</td>";
        echo"<td>" . $item['biography'] .
"</td>";
        echo"<td><a
href='php/delete.php?id=". $item['id']
."&table=masters">Удалить</a></td>";
        echo"</tr>";
      }
    ?>
  </table>
  <p>Добавление мастера</p>
  <?php
    $data = $_POST;

    // проверка формы на пустоту полей
    $errors = array();
    if ( trim($data['fio']) == '' )
    {
      $errors_masters[] = 'Введите ФИО';
    }

    if ( trim($data['phone_number']) == '' )
    {
      $errors_masters[] = 'Введите номер
телефона';
    }

    if ( $data['biography'] == '' )
    {
      $errors_masters[] = 'Введите биографию';
    }

    if ( empty($errors_masters) )

```

```

        {
            //ошибок нет, теперь регистрируем
            $user = R::dispense('masters');
            $user->fio = $data['fio'];
            $user->phone_number =
$data['phone_number'];
            $user->biography = $data['biography'];
            R::store($user);
            // header( "refresh:0;url=admin.php" );
        }
    ?>

<div>
    <form action="admin.php" method="POST">
        <table>
            <tr>
                <td><strong>ФИО</strong></td>
                <td><input type="text" name="fio"
value="<?php echo @$data['fio']; ?>"></td>
            </tr>
            <tr>
                <td><strong>Номер
телефона</strong></td>
                <td><input type="tel"
name="phone_number" placeholder="10 цифр, без +7"
pattern="\s?[\(\){0,1}9[0-9]{2}[\)]{0,1}\s?\d{3}[-]{0,1}\d{2}[-
]{0,1}\d{2}"></td>
            </tr>
            <tr>
                <td><strong>Биография</strong></td>
                <td><input type="text"
name="biography" value="<?php echo @$data['biography']; ?>"></td>
            </tr>
            <tr>
                <td colspan="2"><button
type="submit" name="do_signup">Продолжить
регистрацию</button></td>
            </tr>
            <tr>
                <td colspan="2"><?php
if ( ! empty($errors_masters) )
{
    //выводим ошибки авторизации
    echo "<tr><td colspan='2'>" .
array_shift($errors) . "</td></tr>";
}
        ?>
            </table>
        </form>
    </div>
</div>
<?php if($_SESSION['logged_user']['access_level'] == '2'){
?>

```

```

        <div class="spoiler-wrap disabled">
            <div class="spoiler-head">Управляющий персонал
(admin)</div>
            <div class="spoiler-body">
                <table>
                    <tr>
                        <td>#</td>
                        <td>Имя</td>
                        <td>Номер телефона</td>
                        <td>Уровень доступа</td>
                        <td>Действия</td>
                    </tr>
                </table>
                <?php
                    $users = R::find('users', ' access_level >
0 ');

                    foreach ($users as $item){
                        echo"<tr>";
                        echo"<td>" . $item['id'] .
"</td>";
                        echo"<td>" . $item['name'] .
"</td>";
                        echo"<td>" . $item['phone_number']
. "</td>";

                        if($item['access_level'] == 1){
                            echo"<td>Менеджер</td>";
                        }
                        if($item['access_level'] == 2){
                            echo"<td>Администратор</td>";
                        }
                        echo"<td><a
href='php/delete.php?id=". $item['id']
."&table=users'>Удалить</a></td>";
                        echo"</tr>";
                    }
                ?>
            </table>
            <p>Добавление персонала</p>
            <?php
                $data = $_POST;

                // проверка формы на пустоту полей
                $errors_users = array();
                if ( trim($data['name']) == '' )
                {
                    $errors_users[] = 'Введите имя';
                }

                if ( trim($data['phone_number']) == '' )
                {
                    $errors_users[] = 'Введите номер
телефона';
                }

                if ( $data['password'] == '' )

```

```

        {
            $errors_users[] = 'Введите пароль';
        }

        if ( $data['password_2'] != $data['password']
)
        {
            $errors_users[] = 'Повторный пароль введен
не верно!';
        }

        //проверка на существование одинакового номера
телефона
        if ( R::count('users', "phone_number = ?",
array($data['phone_number'])) > 0)
        {
            $errors_users[] = 'Пользователь с таким
номером уже существует!';
        }

        if ( empty($errors_users) )
        {
            //ошибок нет, теперь регистрируем
            $user = R::dispense('users');
            $user->name = $data['name'];
            $user->phone_number =
$data['phone_number'];
            $user->password =
password_hash($data['password'], PASSWORD_DEFAULT); //пароль
нельзя хранить в открытом виде, мы его шифруем при помощи функции
password_hash для php > 5.6
            $user->access_level =
$data['access_level'];
            R::store($user);
            // header( "refresh:0;url=admin.php" );
        }
    ?>

<div>
    <form action="admin.php" method="POST">
        <table>
            <tr>
                <td><strong>Имя</strong></td>
                <td><input type="text" name="name"
value="<?php echo @$data['name']; ?>"></td>
            </tr>
            <tr>
                <td><strong>Номер
телефона</strong></td>
                <td><input type="tel"
name="phone_number" placeholder="10 цифр, без +7"
pattern="\s?[\(\){0,1}9[0-9]{2}[\)]{0,1}\s?\d{3}[-]{0,1}\d{2}[-
]{0,1}\d{2}"></td>
            </tr>

```

```

        <tr>
            <td><strong>Пароль</strong></td>
            <td><input type="password"
name="password" value="<?php echo @$data['password']; ?>"></td>
        </tr>
        <tr>
            <td><strong>Повторите
пароль</strong></td>
            <td><input type="password"
name="password_2" value="<?php echo @$data['password_2'];
?>"></td>
        </tr>
        <tr>
            <td><strong>Уровень
доступа</strong></td>
            <td><select name="access_level">
                <option
value="0">Пользователь</option>
                <option
value="1">Менеджер</option>
                <option
value="2">Администратор</option>
            </select>
        </td>
        </tr>
        <tr>
            <td colspan="2"><button
type="submit" name="do_signup">Продолжить
регистрацию</button></td>
        </tr>
        <tr>
            <td colspan="2"><?php
            if ( ! empty($errors) )
            {
                //выводим ошибки авторизации
                echo "<tr><td colspan='2'" .
array_shift($errors) . "</td></tr>";
            }
            ?>
        </td>
        </tr>
    </table>
</form>
</div>
</div>
<?php } ?>
</div>
<script src="http://code.jquery.com/jquery-latest.min.js"
type="text/javascript"></script>
<script src="js/spoiler.js" type="text/javascript"></script>
</body>
</html>

<?php else: ?>
    <meta http-equiv="refresh" content="0;URL=authorization.php"
/>

```

```
<?php endif; ?>
```

А.6 Код реализации создания записи

```
<?php
require '../db.php';

$id_time = $_GET['time'];
$id_day = $_GET['day'];
$id_master = $_GET['master'];
$id_client = $_SESSION['logged_user']['id'];

if(R::count('records', "client = ?", array($id_client)) < 2){
    header('Refresh: 0; url=../lk.php');
    $record = R::dispense('records');
    $record->client = $id_client;
    $record->date_record = $id_day;
    $record->time_record = $id_time;
    $record->master = $id_master;
    R::store($record);
}
else{
    header('Refresh: 5; url=../lk.php');
    echo "Превышено число записей";
}

?>
```

					ВлГУ.09.02.04.16.3.00 ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		