

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

02.03.03 Математическое обеспечение и администрирование
информационных систем

(код и наименование направления подготовки, специальности)

Технология программирования

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему «Разработка системы онлайн-тренинга сотрудников колл-центра
сервисной компании»

Студент

Е.С. Карпов

(И.О. Фамилия)

(личная подпись)

Руководитель

С.В. Мкртычев

(И.О. Фамилия)

(личная подпись)

Консультанты

К.А. Селиверстова

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доценткафедры ПМИ, А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 20__ г.

Тольятти 2019

АННОТАЦИЯ

Тема: Разработка системы онлайн-тренинга сотрудников колл-центра сервисной компании.

Ключевые слова: СИСТЕМА, ОНЛАЙН-ТРЕНИНГ, КОЛЛ-ЦЕНТР, СЕРВИСНАЯ КОМПАНИЯ, ВЕБ-ПРОГРАММИРОВАНИЕ.

Целью бакалаврской работы является разработка системы онлайн-тренинга, обеспечивающая высокую эффективность подготовки операторов колл-центра сервисной компании.

Объектом исследования является тренинг сотрудников колл-центра сервисной компании.

Предмет исследования – система онлайн-тренинга сотрудников колл-центра сервисной компании.

Методы исследования: объектно-ориентированное программирование, технология клиент-сервер.

В первой главе рассматриваются требования к системе онлайн-тренинга колл-центра сервисной компании.

Вторая глава посвящена моделированию системы онлайн-тренинга колл-центра сервисной компании.

В третьей главе описан процесс реализации системы онлайн-тренинга колл-центра сервисной компании.

В заключении подводятся итоги исследования, формируются окончательные выводы по изучаемой тематике.

Бакалаврская работа состоит из 50 страниц и включает рисунков 23, таблиц 2, источников 21.

ABSTRACT

The title of the graduation work is Development system of online training employees call-center service company.

The aim of the work is development system of online training, that increases the level of training of call center operators of a service company.

The object of the graduation work is training of call center employees of the service company.

The subject of the graduation work is system online training employees call center service company.

The graduation work describes in details work system training employees call center service company.

Call center service company must round the clock to execute customer offers via telephone communication channels, which makes it possible to effectively implement reference and information tasks, provide accurate information about the customer base, and establish trusting relationships with each customer. Without high-quality work of the call center, it is impossible to implement a customer-oriented strategy, which is the main driving factor in maintaining the competitiveness of a modern service company.

It can be concluded that to acquire qualities and skills of working with clients, it is necessary to organize the process of training call center operators with extensive use of online training systems.

The graduation work consists of an explanatory note on 50 pages, introduction on 2 pages, including 23 figures, 2 tables, the list of 21 references including 5 foreign sources.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
Глава 1 ТРЕБОВАНИЯ К СИСТЕМЕ ОНЛАЙН-ТРЕНИНГА СЕРВИСНОЙ КОМПАНИИ	7
1.1 Функциональные особенности систем онлайн-тренинга операторов колл-центра сервисной компании.....	7
1.2 Анализ существующих систем онлайн-тренинга	8
1.2.1 Система онлайн-обучения «НеоДиалог».....	8
1.2.2 Автоматизированная обучающая система «Действия оперативно-диспетчерского персонала при нештатных и аварийных ситуациях»	9
1.2.3 Система обучения и оценки персонала на платформе eLearning 3000	11
Глава 2 МОДЕЛИРОВАНИЕ СИСТЕМЫ ОНЛАЙН-ТРЕНИНГА КОЛЛ-ЦЕНТРА СЕРВИСНОЙ КОМПАНИИ	14
2.1 Инфо-логическое моделирование СОТ	14
2.2 Логическая модель данных СОТ	16
2.2 Архитектура СОТ.....	18
2.3 Программная архитектура приложения СОТ	19
2.4 Функциональная структура СОТ.....	20
Глава 3 РЕАЛИЗАЦИЯ СИСТЕМЫ ОНЛАЙН-ТРЕНИНГА ОПЕРАТОРОВ КОЛЛ-ЦЕНТРА СЕРВИСНОЙ КОМПАНИИ.....	23
3.1 Выбор средств реализации СОТ.....	23
3.2 Разработка физической модели данных СОТ	23
3.2 Разработка диаграммы компонентов СОТ	25
3.3 Разработка программного приложения СОТ	26
3.4 Тестирование СОТ	44
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	48

ВВЕДЕНИЕ

Одной из ключевых служб современной компании, работающей в сфере оказания сервисных услуг, является колл-центр.

Колл-центр сервисной компании должен круглосуточно исполнять запросы клиентов по телефонным каналам связи, максимально эффективно реализовывать справочно-информационные задачи, наполнять достоверной информацией клиентскую базу, устанавливать доверительные отношения с каждым клиентом[8].

Другими словами, без качественной работы колл-центра невозможна реализация клиенто-ориентированной стратегии, которая является основным движущим фактором поддержания конкурентоспособности современной сервисной компании.

Основу персонала колл-центра составляют операторы, к которым предъявляются следующие требования:

- грамотная и четкая речь, хорошая дикция;
- отличное знание персонального компьютера;
- коммуникабельность;
- эмоциональная устойчивость.

Для приобретения указанных качеств и навыков работы с клиентами необходима организация процесса обучения операторов колл-центра с широким использованием тренинговых онлайн-систем.

Таким образом, **актуальность** темы бакалаврской работы обусловлена необходимостью разработки системы онлайн-тренинга сотрудников колл-центра сервисной компании.

Объект исследования бакалаврской работы – тренинг сотрудников колл-центра сервисной компании.

Предмет исследования бакалаврской работы – система онлайн-тренинга сотрудников колл-центра сервисной компании.

Цель исследования – разработка системы онлайн-тренинга, обеспечивающая высокую эффективность подготовки операторов колл-центра сервисной компании.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать модель системы онлайн-тренинга сотрудников колл-центра сервисной компании;
- построить логическую и физическую модели данных системы онлайн-тренинга;
- выбрать программные средства реализации системы онлайн-тренинга;
- реализовать и протестировать систему онлайн-тренинга.

Методы исследования: объектно-ориентированное программирование, технология клиент-сервер.

Практическая значимость бакалаврской работы заключается в разработке системы онлайн-тренинга, обеспечивающая высокую эффективность подготовки операторов колл-центра сервисной компании.

Данная бакалаврская работа состоит из введения, трех глав, заключения и списка литературы.

В первой главе рассматриваются требования к системе онлайн-тренинга колл-центра сервисной компании.

Вторая глава посвящена моделированию системы онлайн-тренинга колл-центра сервисной компании.

В третьей главе описан процесс реализации системы онлайн-тренинга колл-центра сервисной компании.

В заключении подводятся итоги исследования, формируются окончательные выводы по изучаемой тематике.

Бакалаврская работа состоит из 50 страниц и включает рисунков 23, таблиц 2, источников 21.

Глава 1 ТРЕБОВАНИЯ К СИСТЕМЕ ОНЛАЙН-ТРЕНИНГА СЕРВИСНОЙ КОМПАНИИ

1.1 Функциональные особенности систем онлайн-тренинга операторов колл-центра сервисной компании

Система онлайн-тренинга(COT) - это средство обучения профессиональным навыкам с использованием таких платформ, как персональные компьютеры, ноутбуки и мобильные устройства. Она позволяет проходить онлайн-обучение в любое время и в любом месте и снижает зависимость от традиционных методов аудиторного обучения с инструктором [14].

Основные функциональные и архитектурные особенности систем онлайн-тренинга [20]:

- мониторинг профессионального роста обучаемого и формирование пользовательских отчетов;
- простота настройки и масштабируемость;
- поддержка интеграция с облачными вычислениями;
- работа на нескольких устройствах;
- онлайн-контроль и тестирование;
- поддержка авторских методик;
- гибкость конфигурирования.

Кроме того, COT сотрудников колл-центра сервисной компании должна соответствовать следующим требованиям:

- поддержка процесса обучения операторов колл-центра;
- использование современных веб-технологий;
- отсутствие функциональной избыточности;
- адаптация к специфике деятельности компании;
- низкая стоимость владения.

Рассмотрим известные системы онлайн-тренинга на предмет соответствия вышеперечисленным требованиям.

1.2 Анализ существующих систем онлайн-тренинга

1.2.1 Система онлайн-обучения «НеоДиалог»

СОТ «НеоДиалог» - это система дистанционного обучения, которая разработана по типу бизнес-симулятора [10].

Система моделирует ситуации общения, связанные с различными областями коммуникации - деловые переговоры, беседа с клиентом, продажи.

Онлайн тренажёр «НеоДиалог», разрабатывается с 2001 года авторским коллективом НИЦ «ЯМИР», имеется свидетельство государственной регистрации, патент.

Система обеспечивает решение следующих задач:

- увеличение количества продаж и числа заключенных контрактов в среднем на 10-15% (данные официальных отчётов);
- индивидуальный подход в обучении, систематичность тренинга, объективная обратная связь, удобная система мониторинга и контроля качества занятий;
- обучение большого числа сотрудников, включая новичков, содержанию предлагаемой услуги (продукта), корпоративным стандартам общения с клиентом, а также навыкам невербальной коммуникации;
- значительная экономия средств при обучении персонала;
- выработка стрессоустойчивости у сотрудников компании, мотивация личностного развития;
- объективная экспресс-оценка навыков и способностей соискателей на вакансии компании;
- установка и адаптация тренажёра производится под конкретные запросы компании Заказчика или контакт - центра.

Онлайн-тренинг представляет собой последовательность подготовленных заданий, каждое из которых отображается в окне браузера пользователя (рисунок 1.1).

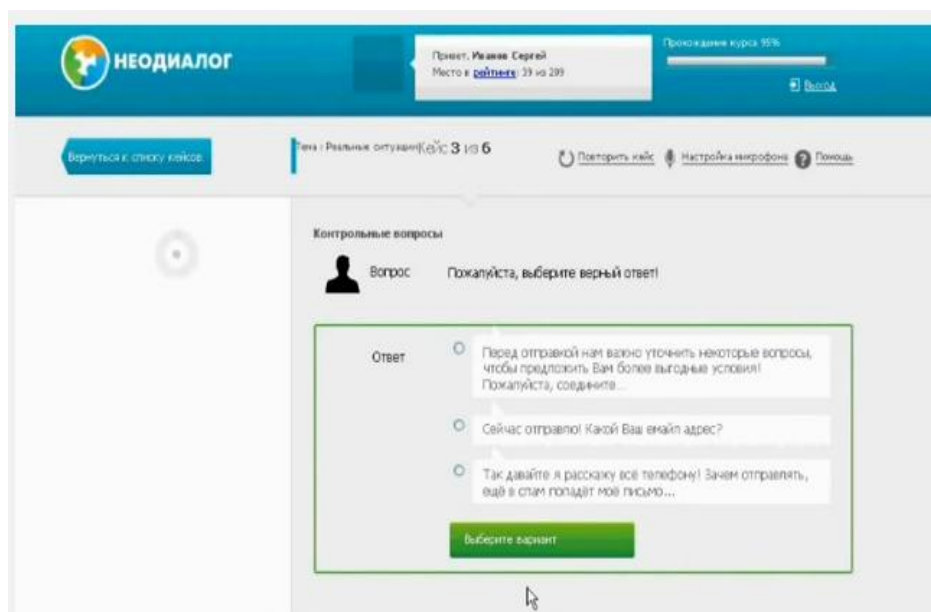


Рисунок 1.1 – Главное окно системы «Неодиалог»

Задания содержат аудио и текстовые материалы, и основываются на тесном интерактивном взаимодействии с проходящим обучение пользователем.

После прослушивания фразы «виртуального собеседника» (ситуационный кейс) пользователь должен произнести в микрофон свой ответ (или прочитать рекомендованную в данной ситуации ответную фразу). В интерфейсе тренажёра есть возможность прослушать пример речевого ответа на прослушанный ситуационный кейс.

Ответ передается на Web-сервер для проведения анализа и формирования итоговых показателей. Таким образом, сотрудник улучшает свой ответ до оптимальных показателей, после чего переходит к следующему заданию, ситуационному кейсу или экзамену, что и обеспечивает тренинг.

Пользователь после каждого задания имеет возможность прослушать свой ответ, пример идеального ответа, с помощью рекомендаций обратить внимание на свои ошибки и научиться их исправлять.

1.2.2 Автоматизированная обучающая система «Действия оперативно-диспетчерского персонала при нештатных и аварийных ситуациях»

Автоматизированная обучающая система «Действия оперативно-диспетчерского персонала при нештатных и аварийных ситуациях» (далее –

АОС) предназначена для изучения теоретических вопросов и проверки знаний оперативно-диспетчерского и эксплуатационного персонала при нештатных и аварийных ситуациях. АОС содержит электронный учебник и программный модуль контроля [6]. В электронном учебнике в текстовой и визуальной форме представлен учебный материал.

Функция электронного учебника – представление изучаемого материала с максимальным облегчением понимания и освоения. При этом используются иные, по сравнению с обычным учебником возможности человеческого мозга, в частности зрительная и эмоциональная память (анимационные ролики).

Программный модуль контроля обучения предназначен для контроля знаний и умения действовать при аварийных ситуациях (рисунок 1.2).

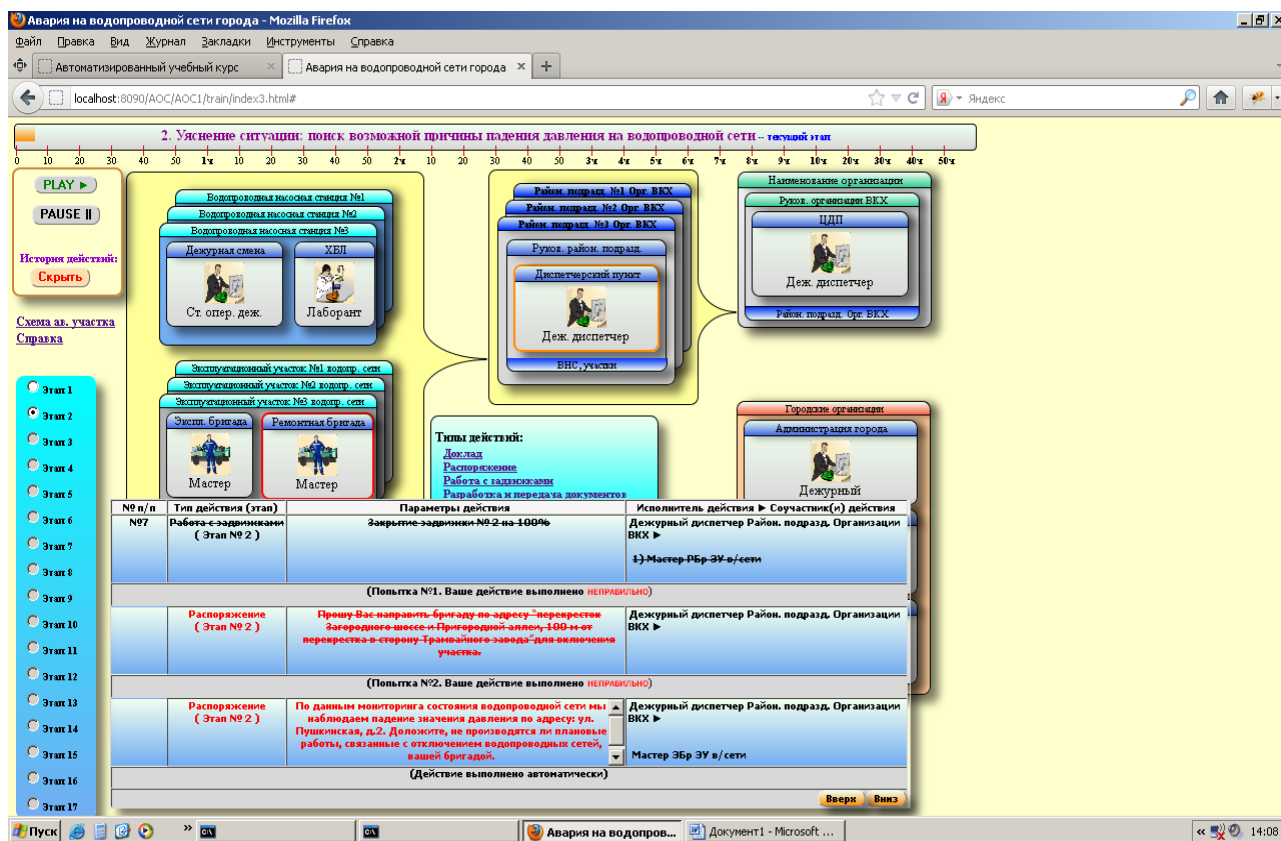


Рисунок 1.2 - Модуль контроля АОС

Использование АОС позволит повысить качество подготовки специалистов, сократить сроки подготовки персонала и эффективного контроля уровня квалификации персонала.

АОС построена на основе современных информационных технологий.

АОС может использоваться как в локальном исполнении (с установкой на произвольное число локальных ПК), так и в сетевом исполнении в составе системы дистанционного обучения предприятия.

1.2.3 Система обучения и оценки персонала на платформе eLearning 3000

Основное направление назначения системы – поддержка электронного и дистанционного обучения и оценки персонала [9].

На рисунке 1.3 представлена функциональная схема системы.



Рисунок 1.3 - Функциональная схема онлайн-системы подготовки персонала на платформе eLearning 3000

В систему входят следующие модули:

- учебные курсы;
- тренинги;
- тренажеры (работа с программным обеспечением, работа с оборудованием, ситуационные задачи).

eLearningServer 3000 - это современная система управления процессом обучения (LearningManagementSystem), позволяющая создавать многофункциональные центры дистанционного образования как в рамках локальной сети (интранет), так и в глобальной сети Интернет.

Система eLearning Server 3000 v2.0 представляет собой платформенно-независимый продукт, созданный на базе технологии PHP и использующий базу данных MySQL, роль Web-сервера может играть Apache или MicrosoftIIS.

Для проведения сравнительного анализа исследуемых аналогов их характеристики сведены в таблице 1.1.

Таблица 1.1 - Сравнительный анализ систем онлайн-тренинга

Характеристика (макс. балл) / COT	COT «Неодиалог»	АОС оперативно-диспетчерского персонала	COT на платформе eLearning 3000
поддержка процесса обучения операторов колл-центра (10)	10	8	7
использование современных веб-технологий (10)	10	10	10
отсутствие функциональной избыточности (10)	5	7	5
адаптация к специфике деятельности компании (10)	7	2	5
низкая стоимость владения (10)	5	5	5
Сумма баллов	37	32	32

Анализ известных систем онлайн-тренинга показал, что ни одна из них не удовлетворяет в полной мере вышеперечисленным требованиям, что обусловило

целесообразность разработки новой СОТ операторов колл-центра сервисной компании.

Выводы к первой главе

1. СОТ сотрудников колл-центра сервисной компании должна соответствовать следующим требованиям: поддержка процесса обучения операторов колл-центра, использование современных веб-технологий, отсутствие функциональной избыточности, адаптация к специфике деятельности компании, низкая стоимость владения.

2. Анализ известных систем онлайн-тренинга показал, что ни одна из них не удовлетворяет в полной мере вышеперечисленным требованиям, что обусловило целесообразность разработки новой СОТ операторов колл-центра сервисной компании.

Глава 2 МОДЕЛИРОВАНИЕ СИСТЕМЫ ОНЛАЙН-ТРЕНИНГА КОЛЛ-ЦЕНТРА СЕРВИСНОЙ КОМПАНИИ

2.1 Инфологическое моделирование СОР

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных нужно строить по аналогии с естественным языком[5].

Основными конструктивными элементами инфологической модели являются сущности, связи между ними и их свойства - атрибуты.

Сущность – любой различимый объект, информацию о котором необходимо хранить в базе данных. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе.

Атрибут – поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей. Атрибуты используются для определения того, какая информация должна быть собрана о сущности.

Связь – ассоциирование двух или более сущностей.

Основная задача СОР заключается в осуществлении контроля квалификации сотрудников по результатам тестирования.

Исходя из анализа предметной области деятельности колл-центра, были определены основные сущности предметной области:

- сотрудник/кандидат;
- тренер;
- учебный материал;
- тест;
- результат.

Тренер дает рекомендации по подбору материалов для изучения, а также составляет и размещает учебные материалы на сайте в соответствующем разделе. Тренер составляет и размещает тесты.

Сотрудник компании или кандидат на получение работы в колл-центре, изучает теоретический материал по определенному направлению. Действующий сотрудник проходит тестирование для подтверждения профессиональной квалификации, а кандидат как подготовительный этап к собеседованию.

По результатам тестирования определяется результат.

В зависимости от вида тестирования результат или не сохраняется (тренинговое тестирование), или сохраняется в базе данных СОР (контрольное и зачетное тестирование).

С учетом вышеизложенного построена инфологическая модель СОР (рисунке 2.1).

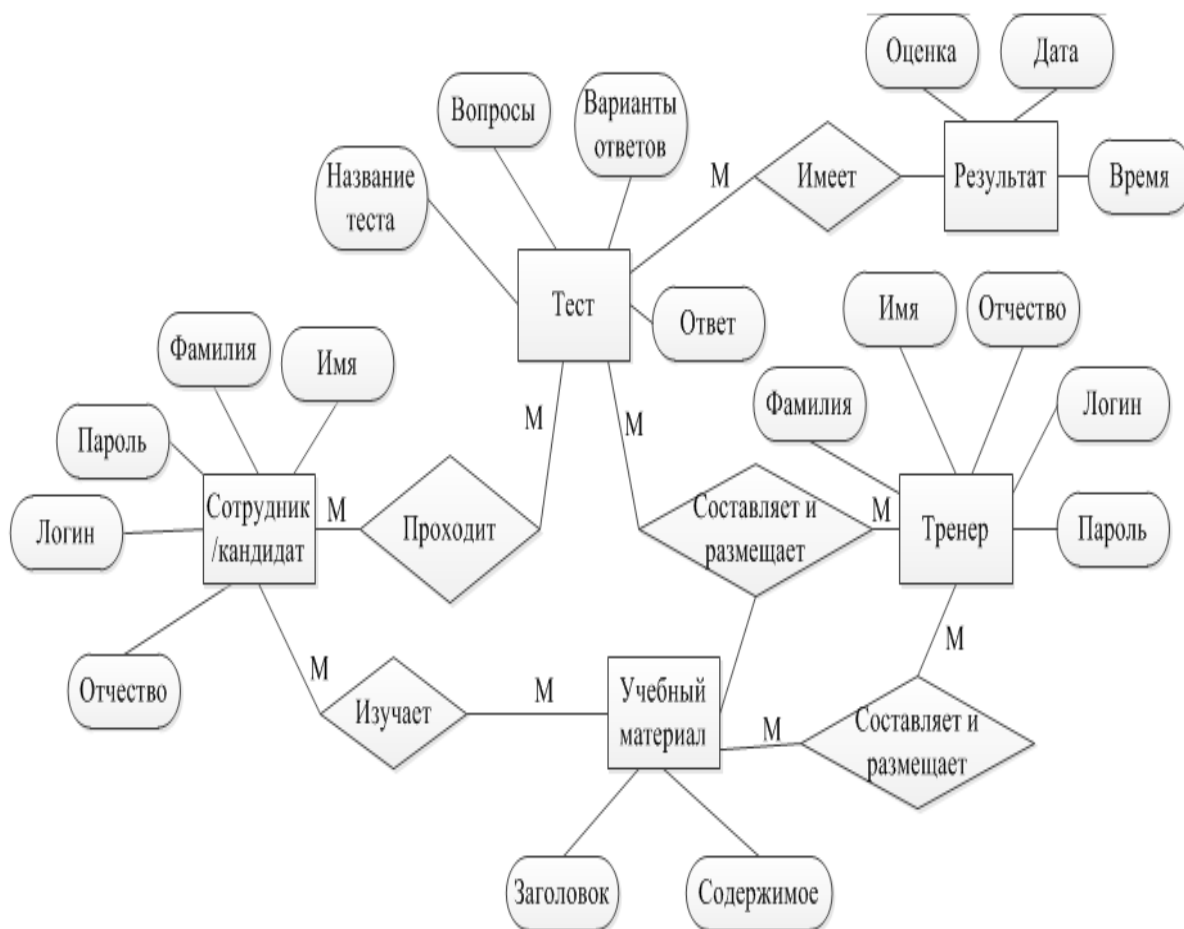


Рисунок 2.1—Инфологическая модель СОР

Инфологическая модель является основой для разработки логической модели данных СОР.

2.2 Логическая модель данных СОР

Логическая модель данных является начальным прототипом будущей базы данных.

Логическая модель данных описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью.

Логическая модель данных строится в терминах информационных единиц, но без привязки к конкретной СУБД. Типы логических моделей это есть не что иное, как модели представления данных.

В современных информационных системах используется реляционная модель базы данных, обеспечивающая высокую производительность обработки и целостность данных [3].

Для быстрого поиска в реляционных таблицах создаются индексы по одному или нескольким полям таблицы. Значения индексов хранятся в упорядоченном виде и содержат ссылки на записи таблицы.

Для автоматической поддержки целостности связанных данных, находящихся в разных таблицах, используются первичные и внешние ключи.

Для выборки данных из нескольких связанных таблиц используются значения одного или нескольких совпадающих полей.

Основным средством разработки логической модели данных в настоящий момент являются различные варианты ER-диаграмм (Entity-Relationship, диаграммы сущность-связь).

ER-диаграмма – это графическое представление взаимосвязей сущностей.

Для построения ER-диаграммы использован облачный сервис DBDesigner [13].

На рис. 2.2 представлена логическая модель данных СОР, построенная на основе ее инфологической модели.

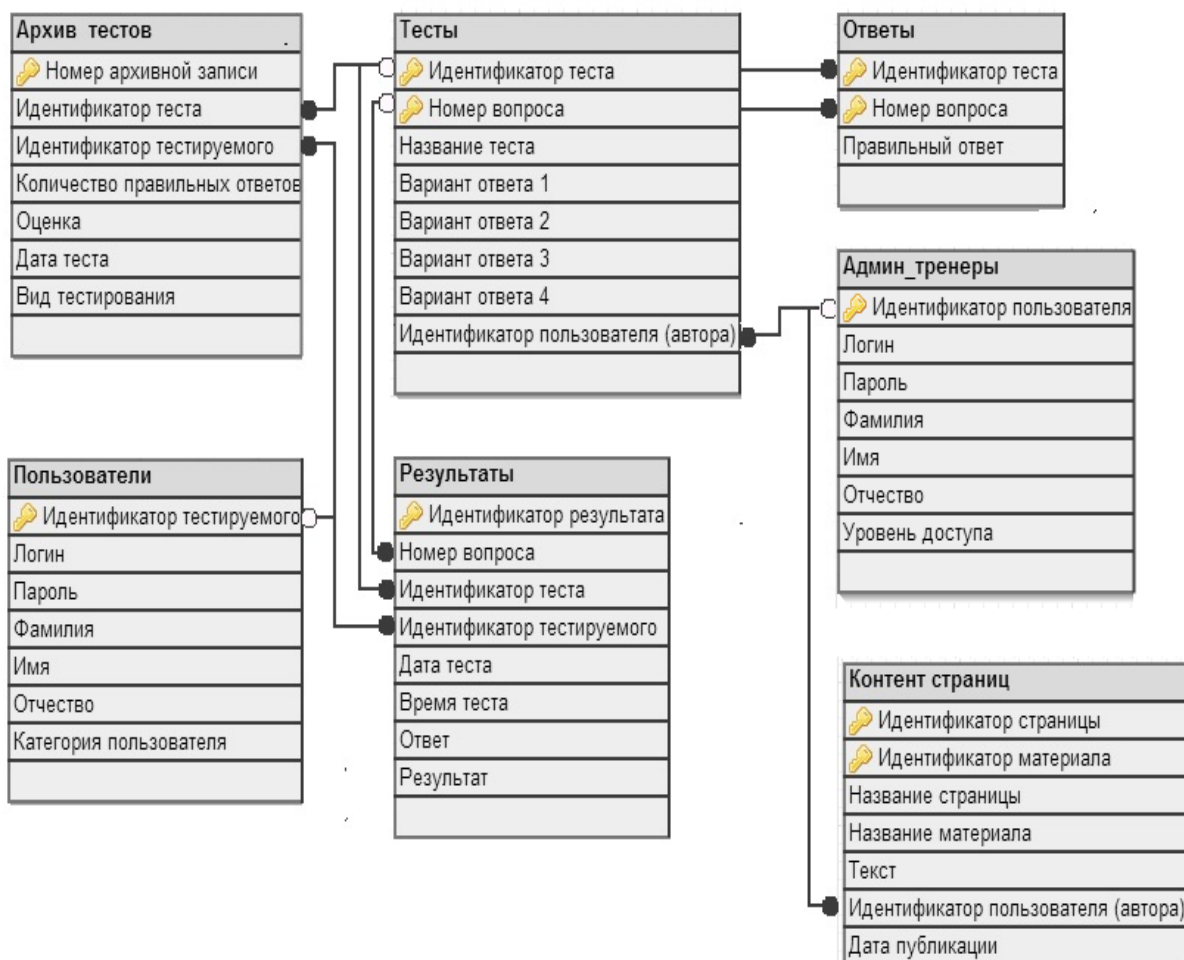


Рисунок 2.2 - Логическая модель данных СОТ

В таблице 2.1 представлено описание типов связей между сущностями СОТ.

Таблица 2.1 - Описание связей между сущностями СОТ

Главная таблица	Подчиненная таблица	Тип связи
Тест	Ответы	1:1
Тест	Результаты	1:M
Тест	Архив тестов	1:M
Пользователи	Результаты	1:M
Админ_тренеры	Тесты	1:M
Админ_тренеры	Контент страниц	1:M

Представленная логическая модель является основой для разработки физической модели СОР.

2.2 Архитектура СОР

Для реализации СОР используется трехуровневая модель архитектуры клиент-сервер (рисунок 2.3) [12].

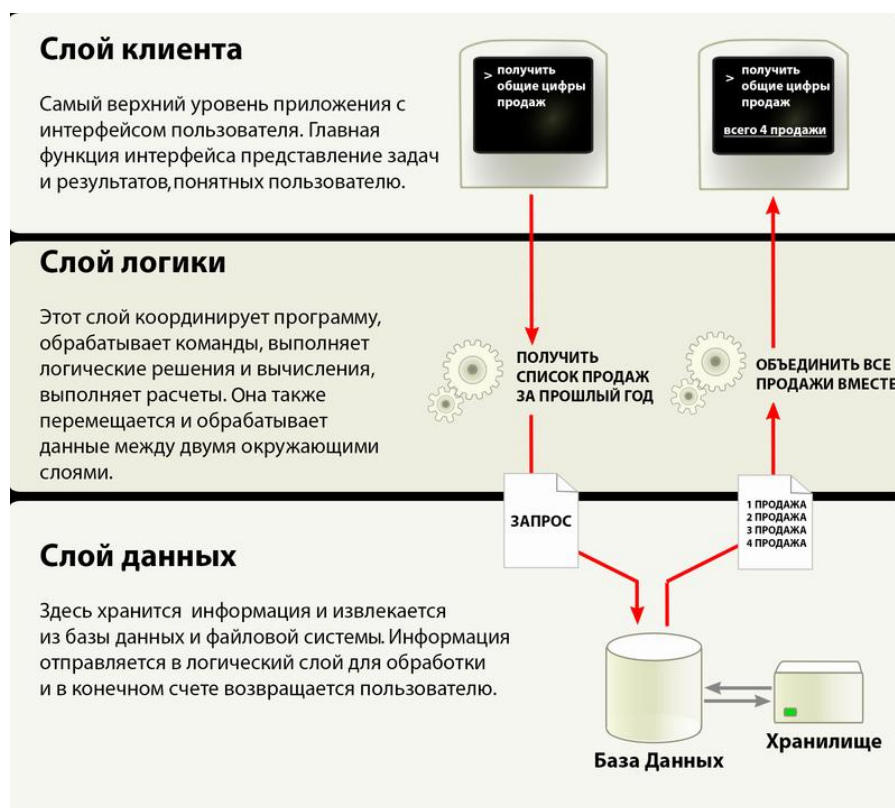


Рисунок 2.3 - Трехуровневая модель архитектуры клиент-сервер

Клиент-серверная архитектура - это вычислительная модель, в которой сервер размещает, доставляет и управляет большинством ресурсов и услуг, которые потребляет клиент.

Архитектура этого типа имеет один или несколько клиентских компьютеров, подключенных к центральному серверу через сеть или подключение к Интернету.

Для реализации слоя клиента (1-й уровень) используется веб-браузер.

Для реализации слоя логики (2-й уровень) используется веб-сервер.

Для реализации слоя данных (3-й уровень) используется сервер баз данных (СБД).

2.3 Программная архитектура приложения СОР

Программная архитектура относится к фундаментальным структурам программной системы и дисциплине создания таких структур и систем.

Она представляет собой абстракцию, отображающую программные элементы системы и связи между ними.

Иными словами, программная архитектура - архитектура взаимодействия различных классов в рамках программного приложения [1].

Для разработки программной архитектуры СОР использует диаграмму классов языка UML[4].

Диаграммы классов являются одним из наиболее востребованных типов UML-диаграмм, т.к. они отображают структуру конкретной системы путем моделирования ее классов, атрибутов, операций и отношений между объектами [15].

На рисунке 2.4 представлена диаграмма классов СОР.

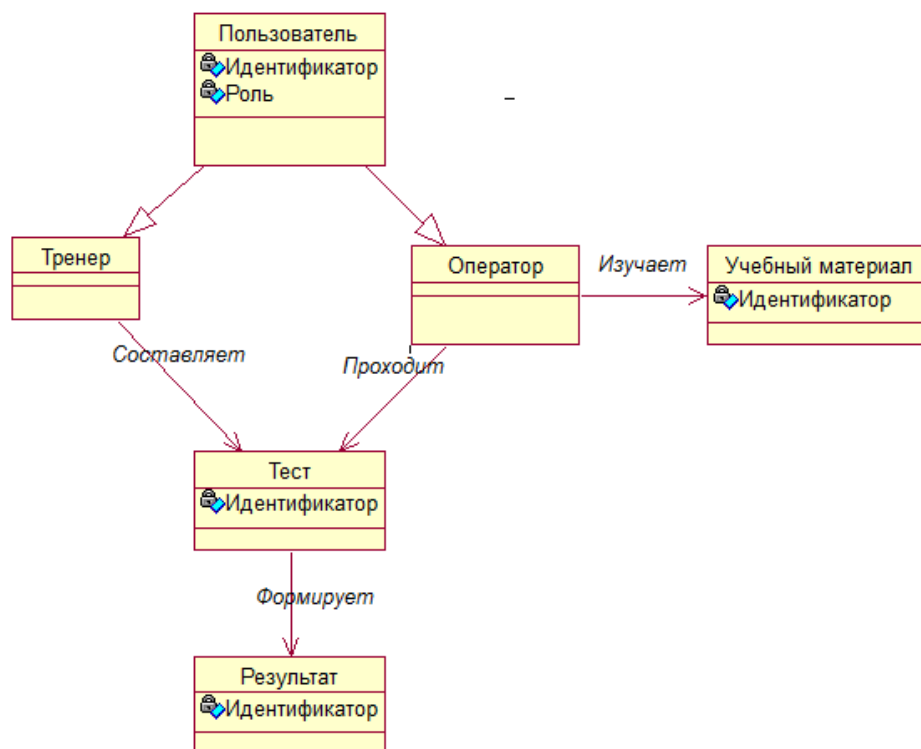


Рисунок 2.4 – Диаграмма классов СОР операторов колл-центра сервисной компании

Все представленные на диаграмме классы отражают вышеописанные сущности СОР.

Как следует из диаграммы, классы Тренер и Оператор являются наследниками класса Пользователей и отличаются друг от друга ролями в рамках СОР.

2.4 Функциональная структура СОР

Функциональная структура СОР в виде структурной схемы меню ее программного приложения представлена на рисунке 2.5.

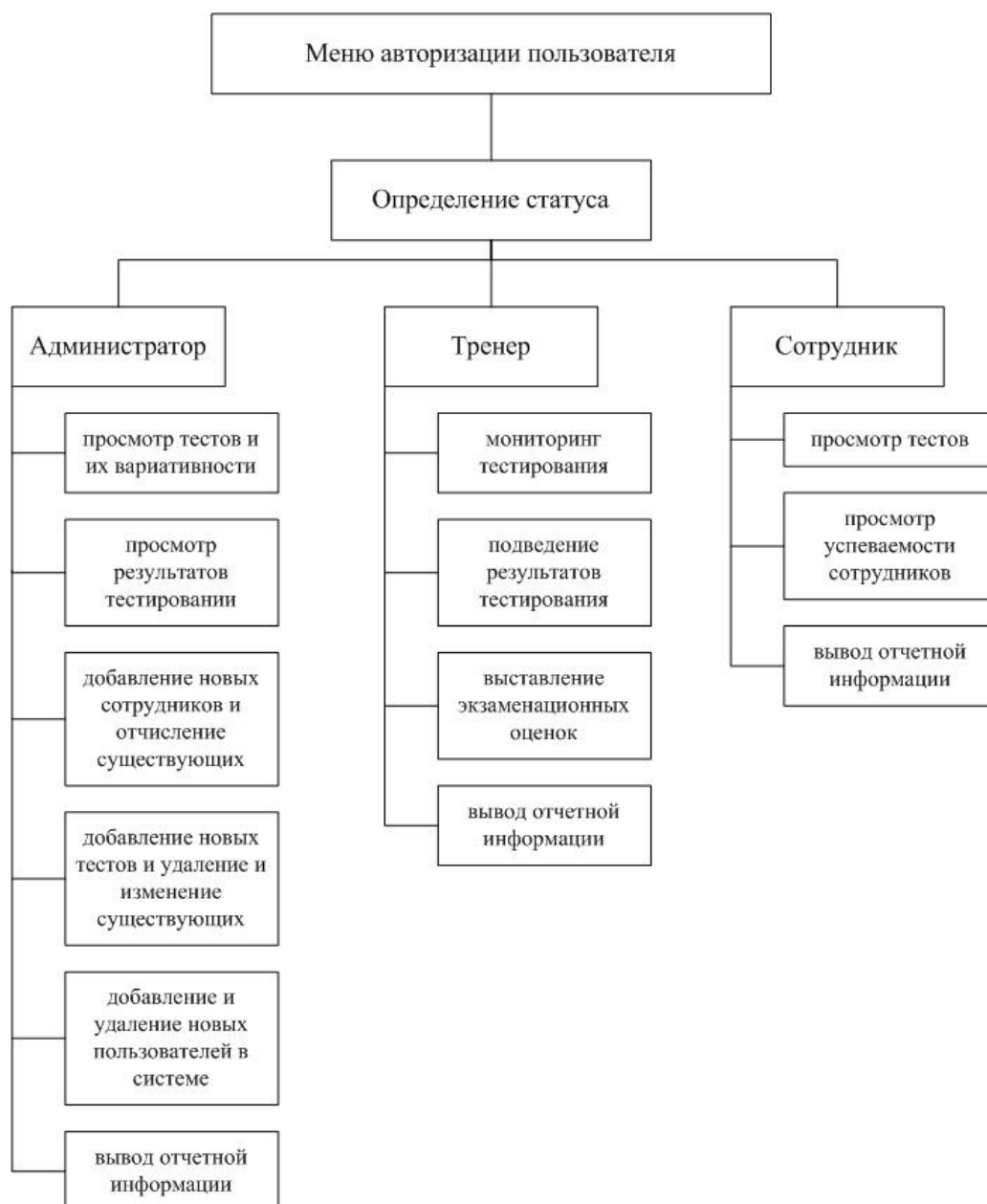


Рисунок 2.5 -Структурная схема меню программного приложения СОР

Перед началом работы с СОР пользователь должен пройти процедуру авторизации, путем ввода регистрационных данных в форму, представленную системой.

После ввода соответствующей информации, система определит статус учётной записи:

- администратор;
- тренер;
- тестируемый (сотрудник/кандидат).

Если статус определён как «Администратор», то для данного типа учётной записи предусмотрены следующие функции:

- просмотр тестов и их вариативности;
- просмотр результатов тестирования;
- добавление новых сотрудников и отчисление существующих;
- добавление новых тестов и удаление и изменение существующих;
- добавление и удаление новых пользователей в системе;
- вывод отчетной информации.

Если статус вошедшего пользователя определён как «тренер», то доступными являются следующие функции:

- мониторинг тестирования;
- подведение результатов тестирования;
- выставление экзаменационных оценок;
- вывод отчетной информации.

Если пользователь авторизовался со статусом «сотрудник», тогда доступными являются следующие функции:

- просмотр тестов;
- просмотр успеваемости сотрудников;
- вывод отчетной информации.

Реализация описанной функциональности СОР представлена в следующей главе.

Выводы ко второй главе

1. Логическая модель данных строится в терминах информационных единиц, но без привязки к конкретной СУБД. Типы логических моделей это есть не что иное, как модели представления данных.

2. Как веб-приложениеСОТ реализуется в трехуровневой архитектуре «клиент-сервер».

3. Диаграммы классов являются одним из наиболее востребованных типов UML-диаграмм, т.к. они отображают структуру конкретной системы путем моделирования ее классов, атрибутов, операций и отношений между объектами.

Глава 3 РЕАЛИЗАЦИЯ СИСТЕМЫ ОНЛАЙН-ТРЕНИНГА ОПЕРАТОРОВ КОЛЛ-ЦЕНТРА СЕРВИСНОЙ КОМПАНИИ

3.1 Выбор средств реализации СОР

В качестве средства реализации СОР выбрана кросс-платформенная среда разработки веб-приложений ХАМРР релиз 5.6 для Windows [11].

ХАМРР – это полностью бесплатный и простой в установке дистрибутив Apache, содержащий систему управления базами данных (СУБД) MySQL, языки программирования PHP и Perl.

В качестве веб-сервера используется программа Apache.

Среда ХАМРР создана с открытым исходным кодом, чтобы обеспечивает простоту ее установки и использования.

При разработке программного кода приложения использован язык объектно-ориентированный язык PHP [17-19].

Выбор средства реализации обусловлен популярностью продукта ХАМРР, а также профессиональными пристрастиями разработчика СОР.

3.2 Разработка физической модели данных СОР

Физическая модель данных системы сопоставляется с логической моделью, но на нее оказывает влияние физическая организация данных конкретной системы управления базой данных. Отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для ключевых атрибутов создаются уникальные индексы, домены преобразуются в типы данных, принятые в конкретной СУБД.

Для создания таблиц базы данных в СУБД MySQL используется графическая оболочка phpMyAdmin [7].

Таблицы базы данных (БД) MySQL формируются с помощью трех файлов: *.frm - формат таблицы; *.myd – данные; *.myi – индексы.

Перечень таблиц базы данных callcenter представлен на рисунке 3.1, физическое размещение каталога с таблицами – на рисунке 3.2.

Table	Action	Records ¹	Type	Collation	Size	Overhead
CCAnswers		6	MylSAM	utf8_bin	1.2 KiB	80 B
CCArchivTest		65	MylSAM	cp1251_general_ci	6.6 KiB	-
CCResults		10	MylSAM	cp1251_general_ci	2.6 KiB	184 B
CCTesttrain		1	MylSAM	cp1251_general_ci	4.9 KiB	-
CCUsers		1	MylSAM	utf8_bin	2.0 KiB	-
CCUserstest		3	MylSAM	utf8_bin	2.1 KiB	-
CC_Test1		3	MylSAM	utf8_bin	2.1 KiB	-
CC_Test2		2	MylSAM	utf8_bin	2.0 KiB	-
CC_Test3		0	MylSAM	utf8_bin	1.0 KiB	-
CC_Test4		1	MylSAM	utf8_bin	2.0 KiB	-
10 table(s)	Sum	92	MylSAM	cp1251_general_ci	26.6 KiB	264 B

Рисунок 3.1 -Переченьтаблиц БДcallcenter на локальном сервере

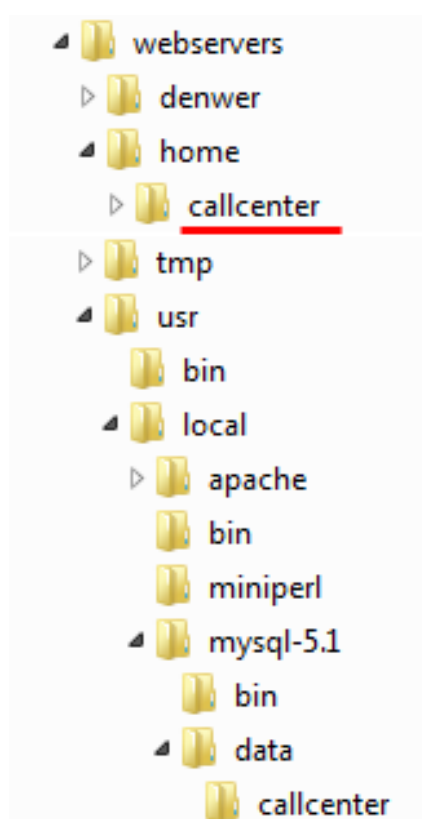


Рисунок 3.2 - Размещение базы данных callcenter и каталога с файлами БД callcenter на сервере

Нарисунке 3.3 представлена физическая модель данных СОР, разработанная с учетом особенностей СУБД MySQL.

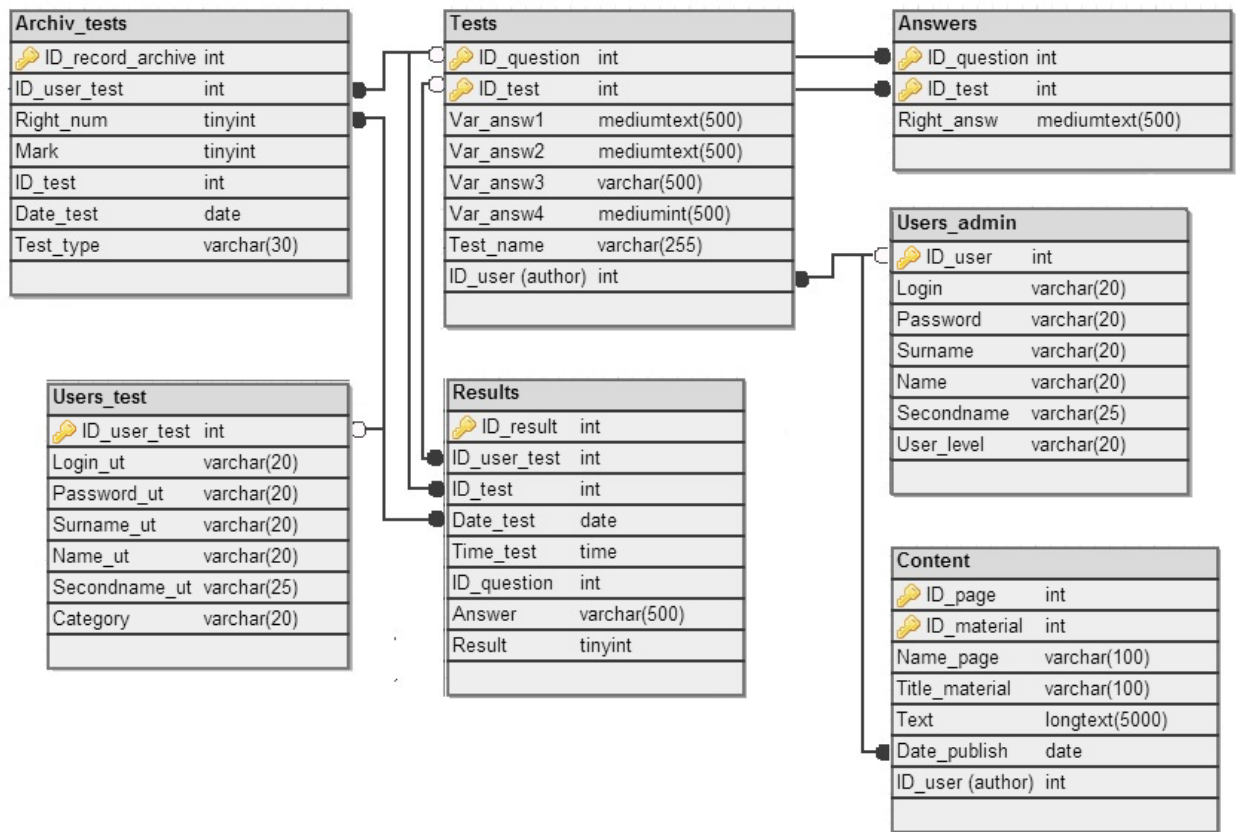


Рисунок 3.3 - Физическая модель данныхСОТ

Генерация таблиц осуществлена с помощью SQL- кода, полученного на основе логической модели данных СОТ.

3.2 Разработка диаграммы компонентовСОТ

Диаграмма компонентов СОТпредставлена на рисунке3.4.

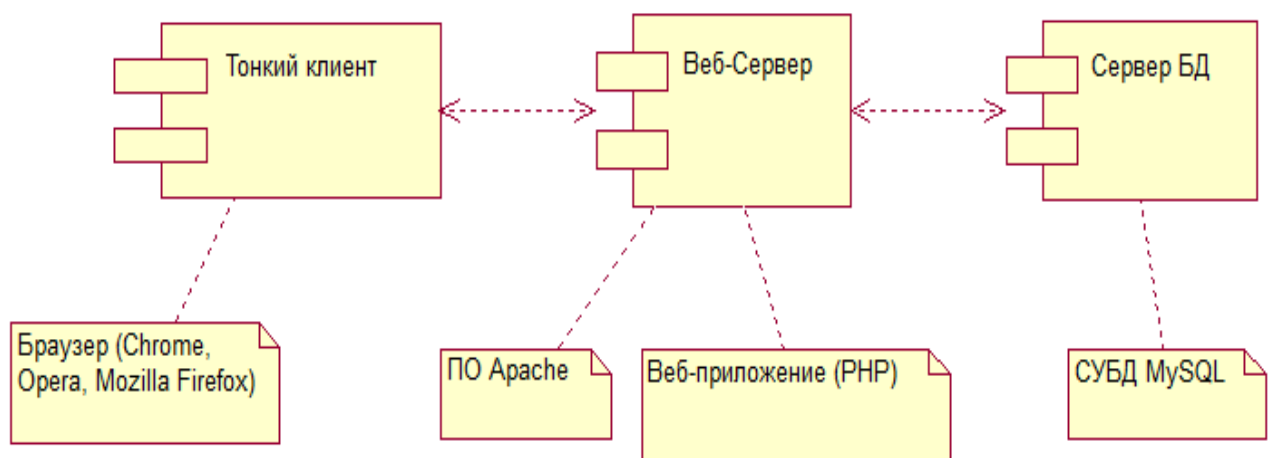


Рисунок 3.4 - Диаграмма компонентов СОТ

На диаграмме представлены следующие компоненты:

- Тонкий клиент - персональный компьютер, предоставляющий доступ к программному клиенту COT– браузеру (Chrome, Opera, MozillaFirefoxи др.). Доступ к клиенту осуществляется набором соответствующей URI адреса в строке браузера;
- веб -сервер (Apache). В качестве контейнера для содержимого сайтов чаще всего используются виртуальные хосты с тематическими названиями, содержащиеся внутри каталога«WWW»;
- сервер базы данных на основе СУБД MySQL;
- локальная(Инtranет) и глобальная компьютерная сети (Интернет)-сети обмена данными, как в рамках компании, так и для внешних пользователей.

Клиентская машина генерирует запросы по работе с данными, которые отправляются на Web-сервер через сеть.

Сервер обрабатывает запрос, формирует обращение к БД, которая отправляет ответное сообщение с запрошенными данными.

Веб-сервер преобразует данные в веб-страницу, которая просматривается в браузере на компьютере клиента.

3.3Разработка программного приложения COT

Приложение COT представляет собой динамический веб-сайт.

Для правильного отображения кириллических шрифтов в различных браузерах и при использовании разных операционных систем (Windows, Linux) в текстовом редакторе Notepad++, который использован для написания программного кода, выбрана кодировка UTF-8, а на каждой странице сайта прописана строка: <metacharset="utf-8">.

Страницы сайта имеют постоянную часть («шапку», «подвал», меню) и динамическое содержимое, которое меняется в зависимости от выбранных команд меню. Такая функциональность реализована с помощью операторов includeи switch и путем обработки переменных, передаваемых по ссылкам.

Для хранения информации, которая динамически подгружается на страницы сайта, в директории engine создан файл functions.php.

Например, на главной странице при выборе различных вакансий(передача переменных по ссылке, например, Оператор call-центра) информация в правой части страницы изменяется соответствующим образом посредством следующего программного кода:

```
$m=$_GET['m'];          /* использование суперглобальной переменной */
include "engine/functions.php"; /* файл с динамически подгружаемой
информацией*/
switch ($m){
case 1: M1(); break;    /* информация об операторе call-центра*/
case 2: M2(); break;    /* информация об операторе
тех.поддержки*/
case 3: M3(); break;
case 4: M4(); break;
case 5: M5(); break;
default:M1();break;
}
```

Использование в PHP глобальных переменных типа \$_GET['переменная']; позволяет поставить первый эшелон защиты для данных, получаемых из текстовых полей или из адресной строки браузера. Такие переменные проходят проверку на сервере и сохраняются в глобальных массивах \$_POST[] или \$_GET[] в зависимости от используемого метода передачи данных. Затем они извлекаются и подвергаются обработке в PHP- коде [6].

На страницах, где требуется передача конфиденциальных данных (логины, пароли), используется метод POST. В отличие от метода GET, где данные сразу передаются на сервер в открытой адресной строке в формате «?идентификатор=значение &идентификатор=значениеи т.д.», при использовании метода POST на сервер передается одна пара «идентификатор=значение» за один раз, и данные в адресной строке не отображаются.

Интерфейс ввода данных отделен от файлов-обработчиков, взаимодействующих с базой данных. Данные передаются как переменные по ссылкам или в сессиях.

Например, обработка данных регистрации, производится в файле enter/registration.php:

```

echo "<html lang='ru'><meta charset='utf-8'/>";
require_once("../bd.php"); //файл с параметрами соединения с сервером и базой
данных
$log_glob=$_POST['email'];// пользуемся
$psw_glob=$_POST['pass'];//переменными
$firstname_glob=$_POST['firstname']; //изглобального массива$_POST
$secondname_glob=$_POST['secondname'];
$lastname_glob=$_POST['lastname'];
if (($log_glob == "") || ($psw_glob == "") || ($firstname_glob == "") ||
($secondname_glob == "") || ($lastname_glob == "")) // проверка,
нетлипустыхзначений
{
echo "<script>alert('Вы заполнили не все поля!')</script>";
echo "<META HTTP-EQUIV='Refresh' CONTENT='0;
URL=../?st=2#join_form'>";}
else
{
$f="Select Login from CCUserstest where Login='$log_glob'";
//выбираемзаписьдляпроверкисовпаденияименипользователя
    $p=mysql_query($f);
    $r=mysql_fetch_row($p);
    if($r[0]==$log_glob) //если запись в таблице уже существует, выводим
сообщение
    {
        echo "<script>alert('Имя $r[0] уже существует!')</script>";
        echo "<META HTTP-EQUIV='Refresh' CONTENT='0;
URL=../?st=2#join_form'>";    }
    else
        {$i="Insert into CCUserstest values
(',$log_glob','$psw_glob','$firstname_glob','$secondname_glob','$lastname_glob','$n
ewName)";
        mysql_query($i);
        .....
        .....
echo "<p align='center'><a href =
'../?menu=2#login_form'>Авторизироваться</a></p>";
}
}

```

Для устранения проблемы «белых» страниц при выполнении запросов к базе данных используется асинхронная обработка данных по технологии Ajax.

Обработка запросов идет в асинхронном режиме, и пользователь может продолжать работу с приложением.

Фрагмент кода query_bd.js:

```
varxmlHttpRequestAddNote = createXmlHttpRequestObject();
function createXmlHttpRequestObject ()
{// для хранения ссылки на объект XMLHttpRequest
varxmlHttpRequest;
// этот участок кода работает во всех браузерах, за исключением ie6 и более
старых версий
try
{// попытаться создать объект XMLHttpRequest
xmlHttpRequest = newXMLHttpRequest();}
catch (e){ //создаем объект запроса для InternetExplorer
try{
xmlHttpRequest = new ActiveXObject("Msxml2.XMLHTTP");
} catch(e){
try{
xmlHttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
} catch(e){
xmlHttpRequest = null;
}}}}
if(xmlHttpRequest == null)alert("БраузернеподдерживаетAJAX!");
}
functionaddNote()
{if(xmlHttpRequestAddNote)
    {try
        {if (xmlHttpRequestAddNote.readyState == 4 || xmlHttpRequestAddNote.readyState ==
0)
            {xmlHttpRequestAddNote.open("POST", "function.php", true);
xmlHttpRequestAddNote.setRequestHeader("Content-Type", "application/x-
www-form-urlencoded");
xmlHttpRequestAddNote.onreadystatechange = handleAddNote;
xmlHttpRequestAddNote.send(params);
            }
            else
                setTimeout('addNote()', 1000);
            }
            catch(e)
                {alert("Ошибка: " + e.toString());        }}}}
```

При тестировании пользователь проходит процесс авторизации один раз при входе в режим тестирования, и может проходить контрольное тестирование

несколько раз, не вводя логин и пароль в течение сеанса работы с браузером до его закрытия. Для этого использован механизм сессий. Сессия – это набор глобальных переменных, которые регистрируются и могут быть переданы со страницы на страницу:

```
session_start();
/*Регистрируем сессионные переменные*/
$_SESSION['d']=$userZ['ID_user']; //идентификатор пользователя
$_SESSION['log']=$userZ['Login']; //логин
$_SESSION['f']=$userZ['Surname']; //фамилия
$_SESSION['i']=$userZ['Name_user'];//имя
$_SESSION['o']=$userZ['Second_name']; //отчество
```

Сессии и cookies предназначены для хранения сведений о пользователях при переходах между несколькими страницами. При использовании сессий данные сохраняются во временных файлах на сервере. Файлы с cookies хранятся на компьютере пользователя, и по запросу отсылаются браузером серверу.

Использование сессий и cookies очень удобно и оправдано в таких приложениях как интернет-магазины, форумы, доски объявлений, когда, во-первых, необходимо сохранять информацию о пользователях на протяжении нескольких страниц, а, во-вторых, своевременно предоставлять пользователю новую информацию. Протокол HTTP является протоколом «без сохранения состояния», то есть он не имеет встроенного способа сохранения состояния между двумя транзакциями. Когда пользователь открывает сначала одну страницу сайта, а затем переходит на другую страницу этого же сайта, то невозможно установить, что оба запроса относятся к одному пользователю. Сеанс, по сути, представляет собой группу переменных, которые, в отличие от обычных переменных, сохраняются и после завершения выполнения PHP-сценария.

При работе с сессиями различают следующие этапы:

- открытие сессии;
- регистрация переменных сессии и их использование;
- закрытие сессии.

Функция получает `session_id()` и/или устанавливает `id` текущей сессии. Это уникальный 32-значный номер, например, 9320f030439e38e16b058e2115617f78.

Чтобы пользователь не вводил каждый раз пароль и логин при входе на сайт, регистрируется переменная для подсчета количества входов каждого пользователя:

```
if (!isset($_SESSION['counter'])) {$_SESSION['counter']=0;}  
else {$_SESSION['counter']++;}
```

А затем она анализируется:

```
if ((($_SESSION['counter']==0) || ($_SESSION['counter']==1) )&&  
(empty($_SESSION['f']))) )  
{include "engine/enter/index.php";}  
else  
{include "test.html";} /*-- Контрольное тестирование --*/
```

При закрытии окна браузера сессия автоматически прекращается и повторный вход возможен только через процедуру авторизации. Сессию можно завершить принудительно с помощью функции `session_destroy()`.

При обработке паролей пользователей системы используется функция SHA-1 (от англ. SecureHashAlgorithmVersion1 — безопасный алгоритм хеширования, версия 1) - семейство криптографических алгоритмов — однонаправленных хеш-функций, включающее в себя алгоритмы SHA-224, SHA-256, SHA-384 и SHA-512. Хеш-функции предназначены для создания «отпечатков» или «дайджестов» сообщений произвольной битовой длины. Применяются в различных приложениях или компонентах, связанных с защитой информации.

Например, набор символов «123» с помощью функции SHA-1 превращается в следующую последовательность:

123 40bd001563085fc35165329ea1ff5c5ecbdbbeef.

При разработке приложений с web-интерфейсом широко используются функции для работы с регулярными выражениями. Например, для проверки правильности e-mail адреса при отправке сообщения администрации компании применена функция `preg_match()`:

```
function isEmail($email)
{
    return(preg_match("/^[_\.[:alnum:]]+@((([:alnum:]]|[:alnum:]-
_\.[:alnum:]])*[:alnum:]]\.)+(ad|ae|aero|af|ag|ai|al|am|an|ao|aq|ar|arpa|as|at|au|aw|az|ba|bb|bd|be|bf
|bg|bh|bi|biz|bj|bm|bn|bo|br|bs|bt|bv|bw|by|bz|ca|cc|cd|cf|cg|ch|ci|ck|cl|cm|cn|co|com|co
op|cr|cs|cu|cv|cx|cy|cz|de|dj|dk|dm|do|dz|ec|edu|ee|eg|eh|er|es|et|eu|fi|fj|fk|fm|fo|fr|ga|gb|
gd|ge|gf|gh|gi|gl|gm|gn|gov|gp|gq|gr|gs|gt|gu|gw|gy|hk|hm|hn|hr|ht|hu|id|ie|il|in|info|int|i
o|iqr|ir|is|it|jm|jo|jp|ke|kg|kh|ki|km|kn|kp|kr|kw|ky|kz|la|lb|lc|li|lk|lr|ls|lt|lu|lv|ly|ma|mc|m
d|mg|mh|mil|mk|ml|mm|mn|mo|mp|mq|mr|ms|mt|mu|museum|mv|mw|mx|my|mz|na|na
me|nc|ne|net|nf|ng|ni|nl|no|np|nr|nt|nu|nz|om|org|pa|pe|pf|pg|ph|pk|pl|pm|pn|pr|pro|ps|pt|
pw|py|qa|re|ro|ru|rw|sa|sb|sc|sd|se|sg|sh|si|sj|sk|sl|sm|sn|so|sr|st|su|sv|sy|sz|tc|td|tf|tg|th|tj
tk|tm|tn|to|tp|tr|tt|tv|tw|tz|ua|ug|uk|um|us|uy|uz|va|vc|ve|vg|vi|vn|vu|wf|ws|ye|yt|yu|za|z
m|zw)$((([0-9][0-9]?[0-1][0-9][0-9]|[2][0-4][0-9]|[2][5][0-5]))\.)\{3\}([0-9][0-9]?[0-
1][0-9][0-9]|[2][0-4][0-9]|[2][5][0-5]))$/i" , $email));
}
```

В корневой директории сайта расположен файл .htaccess (от.англ.hypertextaccess) — файл дополнительной конфигурации веб-сервера Apache, который позволяет задавать дополнительные параметры и разрешения для работы веб-сервера в отдельных каталогах таких как управляемый доступ к каталогам, переназначение типов файлов и т.д., без изменения главного конфигурационного файла.

В данном случае в нем прописаны указания о применении кодировки UTF-8 и о возможности интерпретации php-скриптов внутри файлов с расширением *.html:

```
AddType "text/html; charset=utf-8" .html .htm .shtml
AddType application/x-httpd-php .html
AddDefaultCharsetOff
```

Таким образом, функциональность сайта реализована с использованием различных приемов, позволяющих оптимизировать загрузку и обработку данных.

В настоящее время большинство браузеров поддерживают версию языка описания документа CSS3, которая сильно расширена по сравнению с предыдущими версиями.

Предусматриваются особенности отображения в разных браузерах. Например, на нескольких страницах сайта реализованы «всплывающие окна», плавно появляющиеся при нажатии на соответствующие

элементы управления.

Плавность достигается за счет использования CSS свойства transition:

```
-webkit-transition: opacity .5s, top .5s;  
-moz-transition: opacity .5s, top .5s;  
-ms-transition: opacity .5s, top .5s;  
-o-transition: opacity .5s, top .5s;  
transition: opacity .5s, top .5s
```

В данном примере свойства opacity (прозрачность) и top (вертикальное положение) меняются в течение 0,5 с. Подобные эффекты украшают дизайн сайта.

Особую динамичность создают так называемые псевдоклассы (link, visited, active, hover), которые позволяют учитывать различные условия или события при определении свойств HTML-тега. Например, изменение цвета шрифта ссылки и перемещение её вверх всех слоев при наведении мыши:

```
#menu li:hover > a {  
    color: #fafafa;  
    z-index: 1000  
}
```

Разметка страниц выполнена с использованием элементов <div> и <table>. Блоки <div> могут наслаиваться с перекрытием один на другой, динамически менять размеры, ширину и цвет границ, шрифтов, позиционирование и т.д. Элемент <table> дает более жесткое позиционирование, но в то же время может обеспечить отображение страницы на весь экран независимо от его ширины, если весь контент страницы размещается между тегами <tablewidth=100%>.

Особую значимость приобрели в последнее время атрибуты class и id элементов в связи с использованием новой технологии jQuery. jQuery - это библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу на странице с помощью идентификаторов, обращаться к атрибутам и содержимому элементов, манипулировать ими [21].

Коллажи и рисунки для оформления страниц сайта созданы с использованием программ Paint.NET 3.5.11 и портативной версии Adobe Photoshop CS4 Portable. Для написания программного кода использован

Notepad++

текстовый редактор с открытым исходным кодом с подсветкой синтаксиса большого количества языков программирования и разметки.

Таким образом, при реализации дизайна программного модуля применены современные приемы и технологии.

Для организации печати использован специальный скрипт print.js.

На рисунке 3.5 представлен фрагмент страницы «Тестирование» СОР.

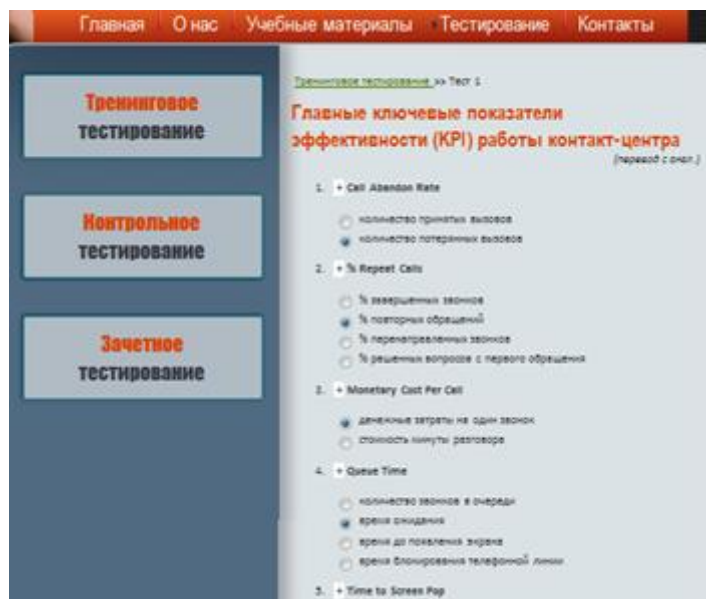


Рисунок 3.5 -Экран страницы тренировочного тестирования

Тренировочное тестирование предназначено для предварительной подготовки к контрольному и зачетному тестированию и может проводиться любым пользователем неограниченное число раз.

Регистрация не требуется. Правильные ответы помечаются знаком "+", неправильные - знаком "-". Результаты тестов не сохраняются. Для реализации данной страницы использован язык JavaScript.

Контрольное тестирование доступно только для зарегистрированных пользователей предусматривает сохранение результатов в базе данных, чтобы пользователь мог проследить динамику результатов тестов.

Пользователь проходит процесс регистрации и авторизации и может тестироваться неограниченное число раз.

Данные обрабатываются в php скриптах и сохраняются в базе данных. Пользователь получает сообщение об успешной регистрации

Если при вводе данных поля остались пустыми и/или пользователь пытается зарегистрировать уже существующий логин, то программа формирует соответствующие сообщения.

Для хранения информации о пользователях в базе данных создана таблица CCUserstest с набором необходимых полей. Обработка данных производится в файле enter/registration.php

После авторизации пользователь может проходить контрольное тестирование несколько раз, не вводя логин и пароль в течение сеанса работы с браузером до его закрытия. Для этого использован механизм сессий.

После успешной авторизации пользователь может выбрать тест (рисунок 3.6) и начать процесс тестирования.

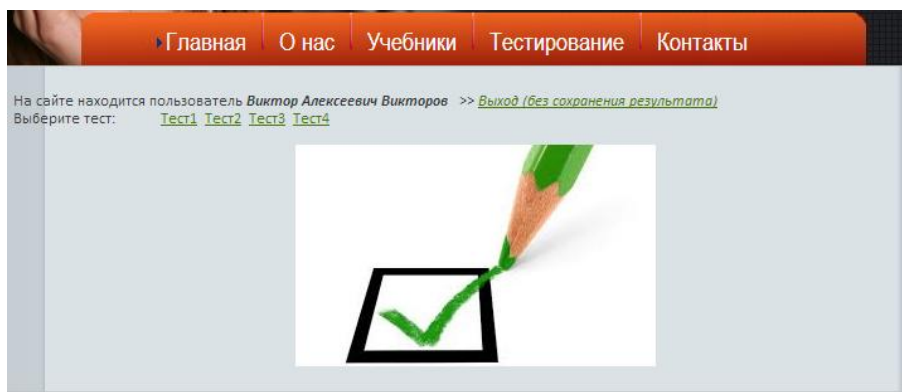


Рисунок 3.6 - Экран режима контрольного тестирования

Список тестов формируется с помощью следующей процедуры:

```
include "../bd.php";
$result = mysql_list_tables('callcenter'); /*-- списоктаблицбазыданных --*/
if(!$result) {
    print "Не удастся получить список таблиц.\n";
    print 'Ошибка работы с базой данных: ' .mysql_error();
    exit;
}
$rgLetters=array('a'=>'a','b'=>'б','c'=>'c','d'=>'д','e'=>'e','f'=>'ф','g'=>'г','h'=>'х','i'=>'и',
'j'=>'ж','k'=>'к','l'=>'л','m'=>'м','n'=>'н','o'=>'о','p'=>'п','q'=>'к','r'=>'р','s'=>'с','t'=>'т','u'
'=>'ю','v'=>'в','w'=>'в','x'=>'кс','z'=>'з','y'=>'я','A'=>'А','B'=>'Б','C'=>'С','D'=>'Д','E'=>'Е',
'F'=>'Ф','G'=>'Г','H'=>'Х','I'=>'И','J'=>'Ж','K'=>'К','L'=>'Л','M'=>'М','N'=>'Н','O'
```

```

=>'O','P'=>'П','Q'=>'К','R'=>'Р','S'=>'С','T'=>'Т','U'=>'Ю','V'=>'В','W'=>'В','X'=>'КС'
,'Z'=>'З','Y'=>'Я');
echo " <ul>Выберите тест:";
while ($row = mysql_fetch_row($result)) { /*-- массив всех таблиц базы данных --
*/
    $pieces = explode("_", $row[0]); /*-- выбираем только тесты --*/
    $pieces[1] = str_replace(array_keys($rgLetters),array_values($rgLetters),
$pieces[1]);
    echo " <li ><a
href=?tbl=".$row[0]."&nam=".$row[0].">".$pieces[1]."</a>&nbsp;&nbsp;&nbsp;</li>";
}
echo "</ul>";

```

При создании тестов в названии используется префикс `CC_` (от названия `callcenter`). Функция `explode("_", $row[0])` разбивает название теста на две части, используя символ `"_"`, и в дальнейшем названия тестов используются без префикса.

В СУБД MySQL в названиях таблиц и полей используются только латинские символы. Для транслитерации используется массив `$rgLetters` и функция `str_replace()`, и названия тестов на странице для удобства пользователей выводятся кириллицей.

После выбора теста на страницу загружаются данные из соответствующей таблицы БД (рисунок 3.7).

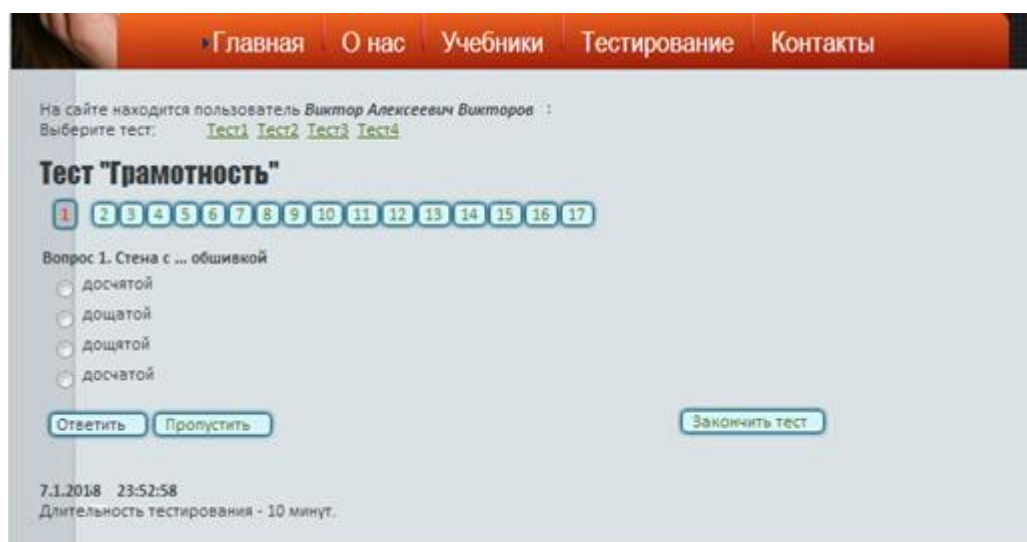


Рисунок 3.7 - Экранвыбора теста

В режиме контрольного тестирования задаются:

- время начала тестирования;

– длительность тестирования, которая зависит от количества вопросов в тесте.

```

if (!isset($_SESSION['timeStart'])) {$_SESSION['timeStart']=time();} /*
временачалатестирования */
$timeStart=$_SESSION['timeStart'];
if (!isset($_SESSION['duration']))
{ /* длительностьтестирования */
    if ($total<=10) {$_SESSION['duration']=600;}
    if (($total>10) &&($total<=15) {$_SESSION['duration']=900;}
    if (($total>15) &&($total<=20) {$_SESSION['duration']=1200;}
}
$duration=$_SESSION['duration'];
if ((time()-$timeStart)>=$duration) /*если длительность тестирования больше
заданного предела, то сессия разрушается*/
{session_destroy();
    echo " <META HTTP-EQUIV='Refresh' CONTENT='0';
URL=../../test.html?test=2'>";
}

```

На рисунке 3.8представлены этапы тестирования. Кнопка активного вопроса выделяется с помощью следующей процедуры:

```

for ($page=1; $page<=$total_pages; $page++) /* навигаторвопросов*/
    { if ($page==($pn+1))/*если страница текущая, то она не является
ссылкой*/
        {echo "<span class='btnL' title='Текущийвопрос'>".$page."</span>";}
    else
        {echo "<a class='btnC'
href='index.php?tbl=".$tbl."&nam=".$nam."&pn=".$page-1.">".$page."</a>";}
    }

```

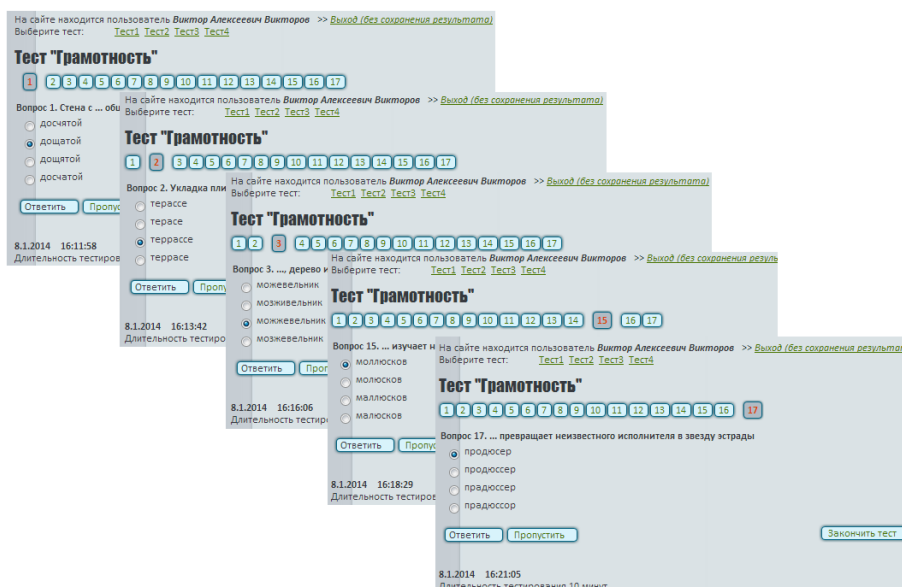


Рисунок 3.8 -Экраныэтапов прохождения теста

При нажатии на кнопку «Ответить» происходит процесс сравнения выбранного варианта ответа с правильным вариантом, который хранится в таблице CCAnswers, и запись результата в таблицу CCRResults.

```
$sl="Select * from CCAnswers where ID_question=$pg and
Test_name='$nam'";
$ransw = mysql_fetch_array(mysql_query($sl));
if ($r==$ransw['Right_answer'])
{ $a=1; /* если ответ правильный, то результат=1*/ }
else
{ $a=0; /* если ответ неправильный, то результат=0*/ }
$rzt="Select ID_question from CCRResults where ID_question=$pg and login='$login'
and Test_name='$nam' and date_test='$dt' and time_test='$timeStart'";
$tr= mysql_fetch_array(mysql_query($rzt));
if(empty($tr[0])) /*если в данном тестировании ответ на вопрос еще не записан,
то вставить запись*/
{ $ins="INSERT INTO `callcenter`.`CCRResults` ( `ID_test`, `login`, `Test_name`,
`date_test`, `time_test`, `ID_question`, `Answer`, `result`) VALUES (NULL, '$login',
'$nam', '$dt', '$timeStart', '$pg', '$r', '$a)"; }
else /*если в данном тестировании ответ на вопрос записан, то обновить
запись*/
{ $ins="UPDATE `callcenter`.`CCRResults` SET `Answer`= '$r', `result`=''$a' where
ID_question=$pg and login='$login' and date_test='$dt' and Test_name='$nam' and
time_test='$timeStart'"; }
mysql_query($ins);}
```

После ответа на последний вопрос пользователь может закончить тест и посмотреть результат или продолжить тестирование (рисунок 3.9).



Рисунок 3.9 - Экран результата тестирования

Подсчет результата текущего тестирования осуществляется с помощью SQLзапроса:

```
$z="SELECTLogin, Test_name, date_test, time_test, result, SUM( result )
FROM CCRResults GROUP BY Login, Test_name, date_test, time_test, result
HAVING Login = '$login' and result=1";
```

Результат в процентах вычисляется следующим образом:

```
$ransw=round(($tz[5]*100/$cntQ),1);
$swans=round((( $cntQ-$tz[5])*100/$cntQ),1);
```

Для сохранения статистики создана таблица CCArchivTest, куда сохраняются результаты всех тестирований с помощью SQL запроса:

```
$sarch="INSERT INTO `callcenter`.`CCArchivTest` (`ID_testT`, `LoginT`, `DateT`,
`TimeT`, `RightN`, `WrongT`, `RightProc`, `WrongProc`, `Mark`, `Test_name`)
VALUES (NULL, '$login', '$dt', '$tt', '$tz[5]', '$wr', '$ransw', '$swans',
'$mark', '$nam)";
```

При этом в таблице CCRResults удаляются все строки, относящиеся к текущему тестированию:

```
$dr="Delete FROM CCRResults WHERE Login='$login' and date_test='$dt' and
time_test='$timeStart' and Test_name='$nam";
```

Если время тестирования превысило заданный предел, то сессия завершается, пользователь получает соответствующее сообщение и перенаправляется на страницу входа в контрольное тестирование:

```
if ((time()-$timeStart)>=$duration)
{session_destroy();
echo "<h3 >Время тестирования закончилось.</h3>";
echo "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href= '../..'/test.html'
class='btnC'>Выход&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</a>";
}
```

Пользователь может посмотреть статистику тестирования, нажав на ссылку «Показать статистику пользователя» (рисунок 3.10).

Реализовано с помощью SQL запроса:

```
$stat="Select * fromCCArchivTestWhereLoginT='$login";
```

Статистика тестирования						
Пользователь Виктор Алексеевич Викторов						
Дата	Время	Количество правильных ответов	Количество неправильных ответов	Процент правильных ответов	Процент неправильных ответов	Оценка
2014-01-08	10:08:08 10	7	41	59	41	УДОВЛЕТВОРИТЕЛЬНО
2014-01-08	10:54:03 16	1	6	94	6	ОТЛИЧНО

Рисунок 3.10 - Отчет результата тестирования

Зачетное тестирование для сотрудников сервисной организации проводится для подтверждения квалификации.

Пользователь получает учетную запись администратора, проходит процесс авторизации и имеет ограниченное число попыток. Пароли хранятся в таблице SS_Users. Тест считается пройденным, если результатом являются оценки ХОРОШ и ОТЛИЧНО.

Зачетное тестирование обрабатывается аналогично контрольному тестированию, но добавляется переменная для определения количества попыток `$_SESSION['countTest']`.

Если количество попыток >3 , то сессия закрывается, и пользователь получает подтверждение о прохождении тестирования с лучшим из трех результатов.

Кандидаты на работу в сервисную компанию могут познакомиться с контактной информацией на странице «Контакты» СОР и отправить сообщение администрации компании с интересующими вопросами.

Перед отправкой сообщения данные проходят проверку на правильность ввода.

Таким образом, в результате проделанной работы реализован веб-интерфейс пользователя СОР.

Для входа в панель управления предусмотрена ссылка на каждой странице, выполненная в виде картинка с контактными телефонами сайта. После нажатия появляется модальное окно для авторизации.

Если пользователь попытается попасть на панель управления из адресной строки, система выведет соответствующее сообщение:

```
if (empty($_SESSION['login']))
{echo "Доступ запрещен!";
echo "<META HTTP-EQUIV='Refresh' CONTENT='2'; URL=./index.html>";}
```

После успешной авторизации Тренер попадает на Панель управления, где он может добавлять, редактировать, удалять тесты и учебные материалы (рис. 3.11).

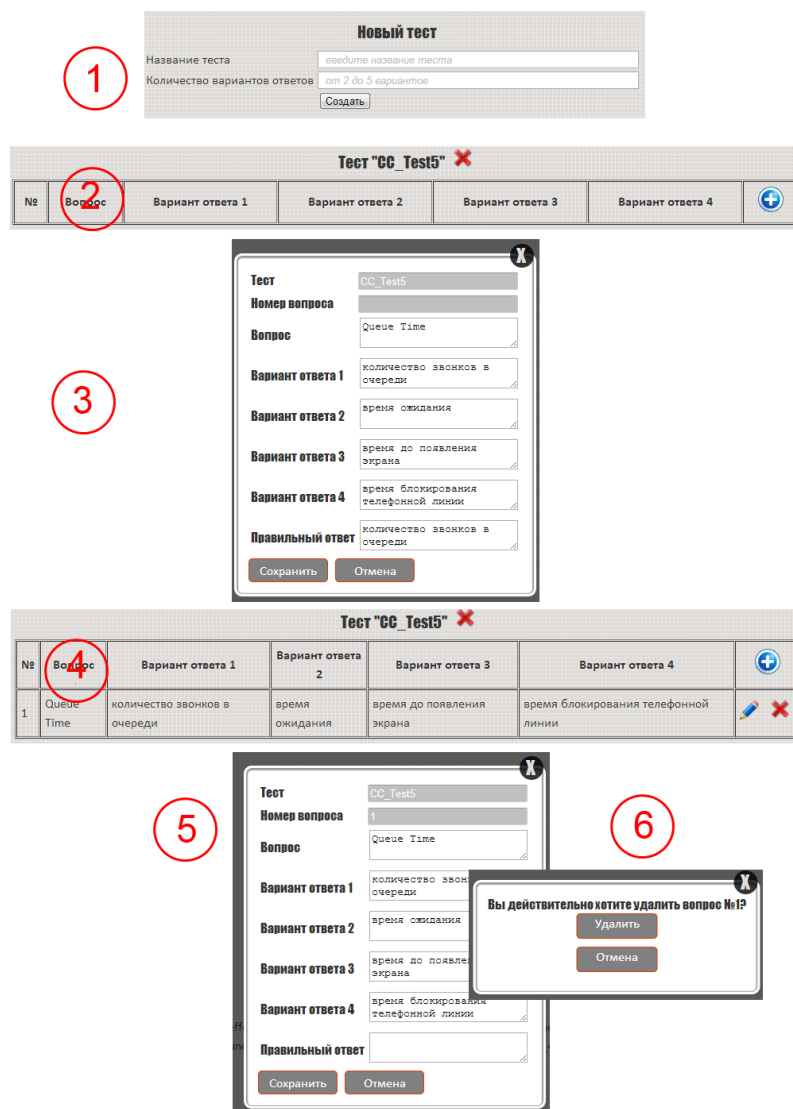


Рисунок 3.11 - Этапы управления учебными и контрольными материалами

Функции:

- формирование «шапки» таблицы;
- создание нового теста;
- добавление вопроса;
- сохранение вопроса;
- редактирование вопроса;
- удаление вопроса.

Все этапы работы с данными реализованы в файле
 Z:\home\callcenter\www\engine\createtest.php.

Например, удаление таблицы:

```
if ($wd==1)
{$queryDel="Drop table $testname"; mysql_query($queryDel);
```

```
echo "Таблица удалена!"; echo "<br><a href='tables.php'>Вернуться</a>";
exit; }
```

Все пиктограммы сопровождаются соответствующими всплывающими подсказками.

Редактирование учебных материалов производится в специальном редакторе, где возможно форматирование текста, вставка изображений и т.д. (рисунок 3.12).

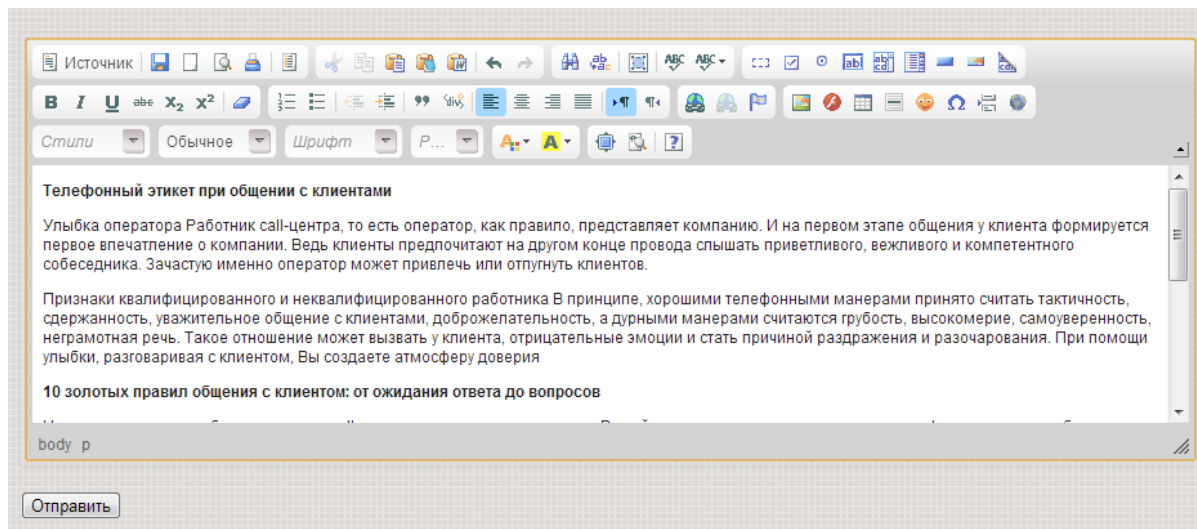


Рисунок 3.12 - Редактирование учебных материалов

После успешной авторизации в модальном всплывающем окне администратор попадает на панель управления (рисунок 3.13), которая разделяется на две области: администрирования таблиц БД и редактирования контента страниц сайта.

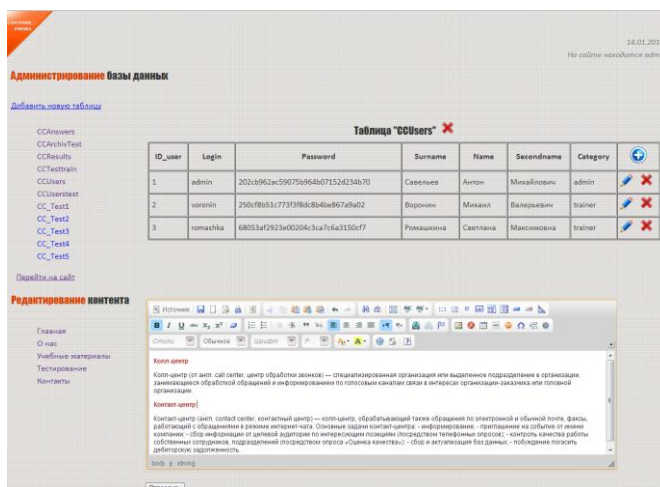


Рисунок 3.13 - Панель управления для администратора

Обработка действий администратора (добавление, удаление,

редактирование данных) проводится аналогично обработке действий тренера.

Но в отличие от тренера администратор имеет полный доступ ко всем таблицам базы данных и ко всему контенту.

На рисунок 3.14 представлен процесс добавления новой записи в таблицу CCUsers в модальном окне, содержимое которого формируется следующим образом:

```
$qr="SELECT * FROM $table";
echo $qr;
$rs=mysql_query($qr);
echo "<table>
while ($field=mysql_fetch_field($rs))
{echo "<tr><td>". $field->name."</td><td><textarea cols='22'
name='answ".$i."></textarea></td></tr>"; } echo "</table>
```

Рисунок 3.14 - Добавление нового пользователя

На рисунке 3.15 представлен результат добавления нового пользователя.

Таблица "CCUsers" ✖

ID_user	Login	Password	Surname	Name	Secondname	Category	
1	admin	202cb962ac59075b964b07152d234b70	Савельев	Антон	Михайлович	admin	
2	voronin	250cf8b51c773f3f8dc8b4be867a9a02	Воронин	Михаил	Валерьевич	trainer	
3	romashka	68053af2923e00204c3ca7c6a3150cf7	Ромашкина	Светлана	Максимовна	trainer	
4	dreamer	6f2268bd1d3d3ebaabb04d6b5d099425	Дремова	Виолетта	Павловна	trainer	

Рисунок 3.15 - Результат добавления нового пользователя

Помимо панели управления администратор имеет полный доступ непосредственно к файлам и может редактировать код напрямую, используя знания веб-программирования и веб-дизайна.

Для редактирования таблиц БД администратор может воспользоваться утилитой `phpmyadmin`.

3.4 Тестирование СОР

Существует три метода тестирования: метод «белого ящика», метод «черного ящика», метод «серого ящика» [16].

Тестирование методом «белого ящика» (whitebox) предполагает обработку системы как «прозрачного объекта» и позволяет заглянуть внутрь, фокусируя внимание на использовании знаний о конкретном программном обеспечении для правильного подбора тестовых данных. Синонимами понятия метода «белого ящика» являются: структурное тестирование, метод прозрачного ящика, метод стеклянного ящика.

В отличие от метода «черного ящика» данный метод основан на использовании определенных знаний программного кода, необходимых для контроля корректности данных на выходе. Тест является правильным только в том случае, когда тестировщик знает, что конкретно должна делать программа. Таким образом тестировщик может контролировать ожидаемый результат. Тестирование методом «Белого ящика» не обрабатывает случайные ошибки, но наряду с этим весь видимый код должен быть удобочитаемым.

Тестирование методом «черного ящика» (blackbox) предполагает обработку системы как «непрозрачного объекта», таким образом, знание внутренней структуры в явном виде не используется. Тестирование этим методом обычно подразумевает проверку функциональных возможностей. Синонимами понятия метода «черного ящика» являются: поведенческое тестирование, функциональное тестирование, метод непрозрачного ящика, метод закрытого ящика. При тестировании программного обеспечения методом «Черного ящика» тестировщик знает только набор вводимых параметров и

ожидаемые на выходе результаты, каким образом программа достигает этих результатов, ему не известно. Тестировщик никогда не проверяет программный код и не нуждается в дополнительном знании программы, кроме как ее технического описания.

Тестирование с использованием метода «серого ящика» (graybox) предполагает рассмотрение в качестве объектов тестирования не все приложение в целом, а отдельные его интерфейсы, как пользовательские, так и прикладные (бизнес-объекты, СУБД, модули системы). В сравнении с методикой «черного ящика» затраты на разработку и запуск тестов значительно выше, поэтому ее применение при тестировании web-приложений обычно ограничивается на уровне прикладных интерфейсов системы, что связано с частыми вариациями интерфейса пользователя.

Тестирование методом «черного» ящика тренером колл-центра выявило проблемы при отображении кириллических символов из таблиц СУБД MySQL.

Использование вместо типа VARCHAR() типа данных VARBINARY(количество символов) - двоичные данные переменной длины, в кодировке UTF-8 позволило устранить эту проблему.

Тестирование методом «белого» ящика сотрудниками колл-центра компании помогло выявить проблему в логической последовательности действий при авторизации пользователя. Проблема устранена путем введения оператора switch вместо if... else.

COT имеет модульную конструкцию, и её функционал может быть расширен в соответствии с кругом задач, возникающих при эксплуатации.

Выводы к третьей главе

1. В качестве средства реализации COT выбрана кросс-платформенная среда разработки веб-приложений XAMPP релиз 5.6 для Windows. Выбор средства реализации обусловлен популярностью продукта XAMPP, а также профессиональными пристрастиями разработчика COT.

2. Физическая модель данных СОР построена на основе ее логической модели данных с учетом функциональных особенностей СУБД MySQL.
3. Тестирование СОР сотрудниками колл-центра компании позволило выявить и устранить ошибки в веб-приложении системы.

ЗАКЛЮЧЕНИЕ

Представленная бакалаврская работа посвящена актуальной проблеме разработки системы онлайн-тренига колл-центра сервисной компании.

В ходе выполнения бакалаврской работы достигнуты следующие результаты:

- разработана модель программной архитектуры системы онлайн-тренинга сотрудников колл-центра сервисной компании;
- построены логическая и физическая модель данных системы онлайн-тренинга для СУБД MySQL;
- выбраны программные средства реализации системы онлайн-тренинга PHP+MySQL;
- реализована и протестировано программное обеспечение системы онлайн-тренинга.

Результаты бакалаврской работы могут быть рекомендованы для решения задач разработки программного обеспечения систем онлайн-тренига колл-центра сервисной компании.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ 34.320-96. Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.

Научная и методическая литература

2. Трутнев Д. Р. Архитектуры информационных систем. Основы проектирования: учеб. Пособие / Д.Р. Трутнев. – СПб.: НИУ ИТМО, 2012. – 66 с.

Электронные ресурсы

3. Емельянова Т. В. Моделирование баз данных [Электронный ресурс] : учебное пособие / Т. В. Емельянова, А. М. Кольчатова, Н. Ю. Зюзина. —Саратов : Ай Пи Эр Медиа, 2018. — 62 с.— Режим доступа: <http://www.iprbookshop.ru/74560.html>(дата обращения 10.06.2019).

4. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. Курс лекций [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / А. В. Леоненков. — Москва, Саратов : ИНТУИТ, Вузовское образование, 2017. — 318 с. - Режим доступа: <http://www.iprbookshop.ru/67388.html>(дата обращения 10.06.2019).

5. Лукьянов Г. В. Информационная модель в проектировании информационных систем [Электронный ресурс] : учебное пособие / Г. В. Лукьянов. — М. : Московский гуманитарный университет, 2016. — 29 с. — Режим доступа: <http://www.iprbookshop.ru/74699.html>(дата обращения 10.06.2019).

6. Сайт Санкт-петербургской академии безопасности [Электронный ресурс].- Режим доступа:<http://www.nsab.ru> (дата обращения 10.06.2019).

7. Сайт phpMyAdmin[Электронный ресурс].<https://www.phpmyadmin.net/>(дата обращения 10.06.2019).

8. Самолюбова А. Б. CallCenter на 100%: Практическое руководство по организации Центра обслуживания вызовов [Электронный ресурс] / А. Б. Самолюбова. — М. : Альпина Паблишер, 2019. — 350 с. — Режим доступа: <http://www.iprbookshop.ru/82972.html>(дата обращения 10.06.2019).

9. Система подготовки и оценки персонала компании на платформе eLearning 3000 [Электронный ресурс]. — Режим доступа: <http://www.hit-service.ru/learnware.html> (дата обращения 10.06.2019).

10. Система онлайн-обучения «НеоДиалог» [Электронный ресурс]. — Режим доступа: <http://www.neodialog.ru/call.htm> (дата обращения 10.06.2019).

11. Среда разработки XAMPP [Электронный ресурс]. — Режим доступа : <https://www.apachefriends.org/ru/index.html> (дата обращения 10.06.2019)

12. Трёхуровневая архитектура «клиент-сервер» [Электронный ресурс]. — Режим доступа:https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0(дата обращения 10.06.2019).

13. DBDesigner[Электронный ресурс]. — Режим доступа:<https://www.dbdesigner.net/>(дата обращения 10.06.2019).

14. WhatisanOnlineTrainingSystem[Электронный ресурс]. — Режим доступа: <https://www.proprofs.com/c/lms/what-is-online-training-system/>(дата обращения 10.06.2019).

Литература на иностранном языке

15. Alan Dennis, Barbara Haley Wixom, David Tegarden: Systems Analysis and Design with UML - 4th Edition, Wiley, 2012.

16. Monsma J.R. Model-based testing of Web applications, Radboud University, 2015.

17. Nixon R. Learning PHP, MySQL, JavaScript, CSS & HTML5 - 3rd Edition, 2014.

18. Schwartz B. and others. High Performance MySQL. — 3rd Edition, 2012.

19. Valade J., Suehring S. PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies, 2013, -720p.
20. Wilson J.P. The Call Centre Training Handbook: A Complete Guide to Learning and Development in Contact Centres, Kogan Page, 2009, - 338 p.
21. Zakas N.C. The Principles of Object-Oriented JavaScript, 2014, - 120 p.