

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)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Автоматизированная информационная система кадрового учёта»

Студент

В.С. Кудрявцев
(И.О. Фамилия)

(личная подпись)

Руководитель

Т.Г. Любивая
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема: Автоматизированная информационная система кадрового учёта.

Ключевые слова: КАДРОВЫЙ УЧЁТ, БИЗНЕС-ПРОЦЕСС, ОТДЕЛ КАДРОВ, АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ.

Актуальность исследования определяется необходимостью внедрения автоматизированной информационной системы (АИС) кадрового учета с целью повышения производительности обработки данных, обеспечения кадрового документооборота.

Объект исследования – деятельность отдела кадров ООО «Успех».

Предмет исследования – автоматизированная информационная система кадрового учёта.

Цель выпускной квалификационной работы – разработка автоматизированной информационной системы кадрового учёта ООО «Успех».

В бакалаврской работе проведен анализ деятельности отдела кадров ООО «Успех». Разработаны функциональные модели деятельности отдела кадров «Как есть» и «Как должно быть». Обоснован выбор инструментальных средств для разработки информационной системы: СУБД MySQL, язык программирования PHP. Выполнено проектирование логической и физической моделей базы данных. Разработано программное обеспечение автоматизированной информационной системы кадрового учета. Проведено тестирование программного продукта.

Работа включает: страниц – 45, рисунков – 19, таблиц – 7, источников – 23.

Оглавление

Введение.....	4
Глава 1 Функциональное моделирование предметной области.....	5
1.1 Характеристика деятельности ООО «Успех»	5
1.2 Анализ бизнес-процессов ООО «Успех».....	6
1.3 Анализ существующих технологий автоматизации работы отдела кадров	10
1.4 Постановка задачи на разработку проекта АИС.....	14
1.5 Разработка функциональной модели бизнес-процесса «Как должно быть».....	16
Глава 2 Логическое проектирование автоматизированной информационной системы кадрового учета.....	17
2.1 Логическая модель и её описание	17
2.2 Информационное обеспечение автоматизированной информационной системы.....	19
2.3 Характеристика результатной информации.....	21
2.4 Требования к аппаратно-программному обеспечению АИС	22
Глава 3 Физическое проектирование автоматизированной информационной системы кадрового учета.....	24
3.1 Выбор архитектуры АИС	24
3.2 Выбор технологии проектирования базы данных АИС.....	25
3.3 Выбор системы управления базой данных АИС	29
3.4 Разработка физической модели данных АИС.....	32
3.5 Разработка и описание модулей АИС с примерами кода	36
3.6 Тестирование программного продукта	41
Список используемой литературы	44
Приложение А Создание базы данных для проекта в СУБД MySQL	46
Приложение Б Фрагмент отчета проверки верности	50

Введение

Актуальность исследования определяется необходимостью внедрения автоматизированной информационной системы (АИС) кадрового учета с целью повышения производительности обработки данных, обеспечения кадрового документооборота.

Объектом исследования является деятельность отдела кадров ООО «Успех».

Предметом исследования является автоматизированная информационная система кадрового учёта.

Целью выпускной квалификационной работы является разработка автоматизированной информационной системы кадрового учета ООО «Успех».

Для достижения указанной цели в работе решаются следующие задачи:

- анализ деятельности компании ООО «Успех»;
- построение концептуальной модели деятельности отдела кадров;
- проектирование логической и физической моделей базы данных;
- разработка программного обеспечения автоматизированной информационной системы;
- проведение тестирования программного продукта.

Бакалаврская работа включает: введение, три главы, заключение, список используемой литературы, приложения.

В первой главе проведено исследование деятельности компании «ООО Успех», разработаны функциональные модели деятельности отдела кадров.

Вторая глава посвящена проектированию логической модели базы данных.

В третьей главе выполнено проектирование физической модели базы данных, разработано программное обеспечение АИС и проведено его тестирование.

Заключение содержит результаты выполненной работы в соответствии с указанными выше задачами.

Глава 1 Функциональное моделирование предметной области

1.1 Характеристика деятельности ООО «Успех»

Компания ООО «Успех» осуществляет свою деятельность на рынке торгового оборудования с 2004 года.

Основным видом деятельности компании является продажа, подключение и сопровождение торгово-кассового оборудования самых известных Российских производителей. Поставка оборудования осуществляется по всем регионам Российской Федерации.

Дополнительные виды деятельности: регистрация ИП и ООО, изготовление печатей, регистрация электронно-цифровых подписей, сервисное обслуживание контрольно-кассовой техники.

Организация выполняет следующие виды услуг:

- подготовку контрольно-кассовой техники (ККТ) к эксплуатации (диагностику и тестирование контрольно-кассовых машин, составление технического заключения о состоянии ККТ, проведение консультационных работ по правилам эксплуатации ККТ);
- регистрацию ККТ;
- установку и обслуживание фискального регистратора на объекте заказчика;
- взаимодействие с органами ИФНС и ОФД (постановку на учёт, перерегистрацию, замену фискальных накопителей, восстановление утраченных документов);
- осуществление (по разрешению клиента) контроля за функционированием личного кабинета ОФД.

На рисунке 1 представлена организационная структура компании ООО «Успех».



Рисунок 1 – Схема организационной структуры ООО «Успех»

Отдел кадров ООО «Успех» выполняет следующие функции:

- учет движения кадров;
- учет персональных данных;
- учет отпусков;
- учет рабочего времени.

Функционирование успешной современной организации невозможно представить без автоматизации деятельности подразделения, занимающегося кадровым учетом и сопровождением сотрудников.

1.2 Анализ бизнес-процессов ООО «Успех»

1.2.1 Выбор технологии моделирования бизнес-процессов

В качестве технологии моделирования бизнес-процессов выбран графический стандарт IDEF0. Аббревиатура IDEF0 известна уже многим, это методология, которая позволяет стандартизировать работу с бизнес-процессами. Она была создана в 50-х гг. XX века в США, когда создавался

аэрокосмический проект. С тех пор она доказала, что является эффективной, за что её приняли в качестве федерального стандарта.

Графический стандарт IDEF0 является частью методологии SADT (Structured Analysis and Design Technique – метод структурного анализа и проектирования). В методологию SADT включено пятнадцать разных моделей, комплекс которых должен обеспечить исследование структуры, параметров и особенностей различных систем организационно-экономического и производственно-технического характера.

IDEF0 – является функциональной моделью, образующей ядро выстраивания всего остального комплекса, в ней связываются организационная структура, материальные потоки, информационные потоки, сама деятельность компании и управляющие воздействия, формируя единую конструкцию. Также можно называть нотацией графический стандарт, принятый для моделирования процессов. Таким образом, нотация является системой требований и правил, по которым должна строиться модель деятельности организации.

1.2.2 Разработка и анализ модели бизнес-процесса «Как есть»

Функциональная модель деятельности отдела кадров ООО «Успех» составлена в виде контекстной диаграммы и её декомпозиции с помощью методологии IDEF0. По результатам функционального моделирования будет разработана информационная системы отдела кадров.

На рисунке 2 представлена контекстная диаграмма «Деятельность отдела кадров ООО «Успех».

На данной диаграмме отображены основные элементы работы отдела кадров. На входе: заявление, документы. Управляют работой отдела кадров: нормативно-правовые документы, должностные инструкции. Механизмами являются: менеджер по персоналу, директор. На выходе: утвержденный приказ, трудовой договор, личная карточка, отчеты.

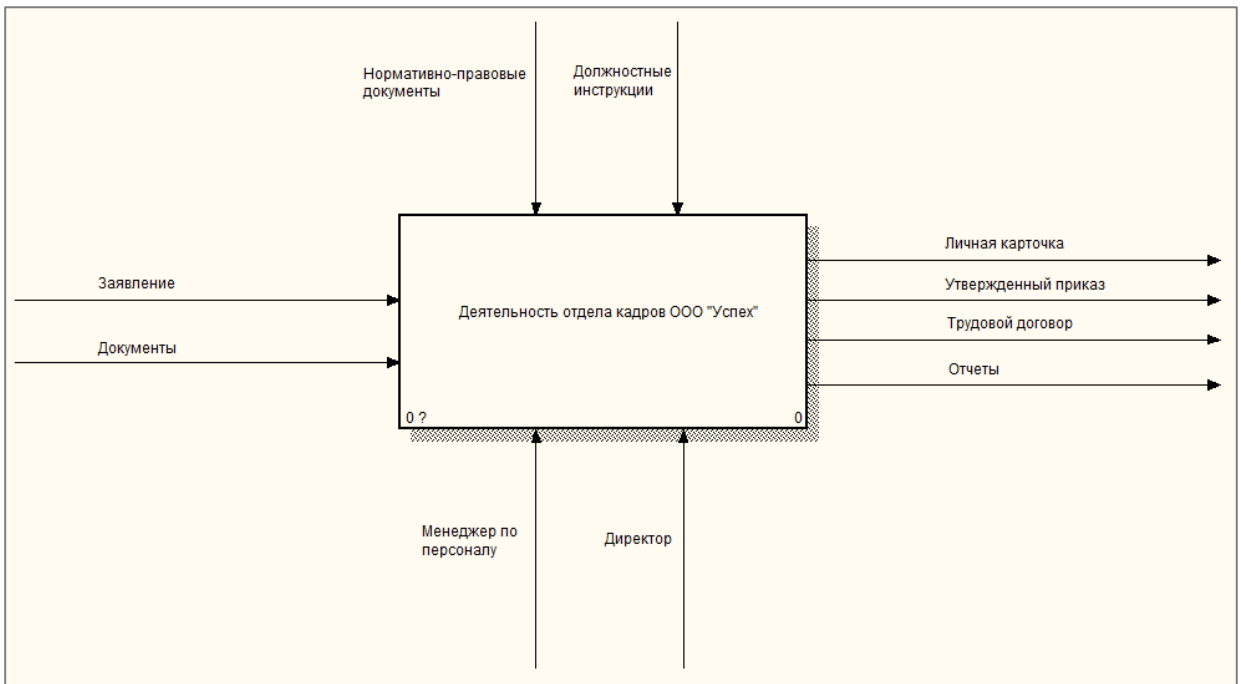


Рисунок 2 – Контекстная диаграмма «Деятельности отдела кадров ООО «Успех»»

Декомпозиция процесса «Деятельность отдела кадров ООО «Успех»» приведена на рисунке 3.

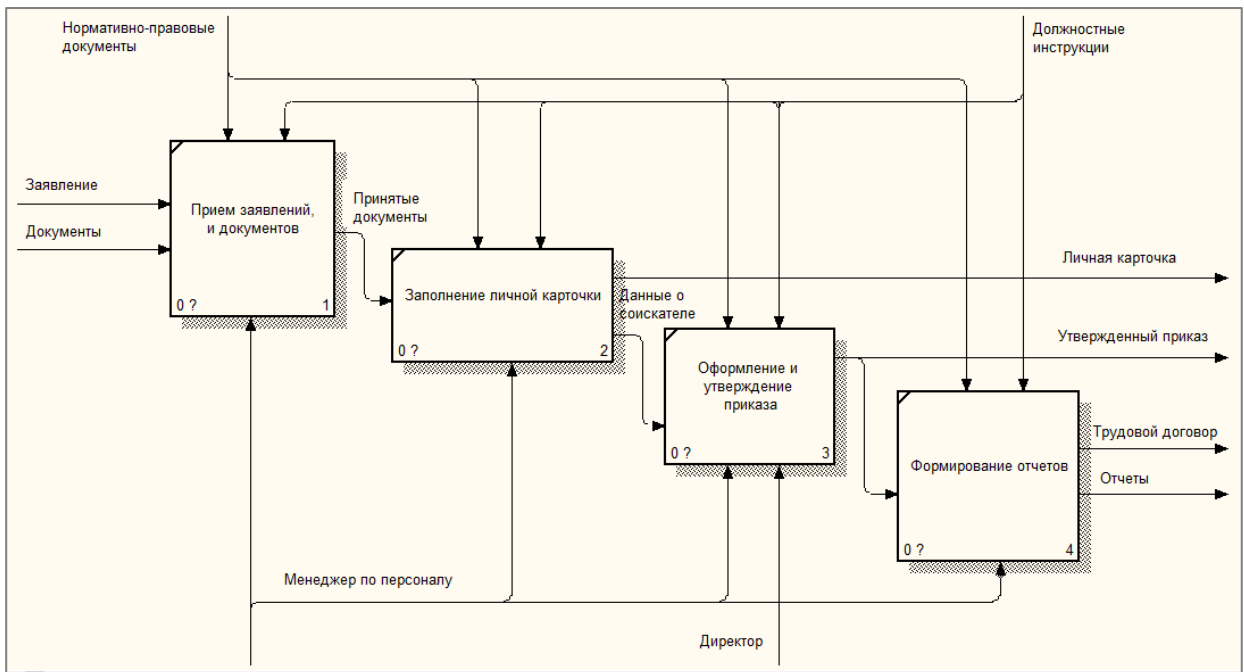


Рисунок 3 – Декомпозиция контекстной диаграммы

На данной диаграмме основной процесс разделен на четыре подпроцесса:

- прием заявлений и документов;
- заполнение личной карточки;
- оформление и утверждение приказа;
- оформление отчетов.

В каждом процессе имеются входные данные и основными являются заявления. Выходные данные представлены в виде отчетов, приказов и распоряжений.

Анализируя диаграммы IDEF0 можно сказать, что процессы отлажены и функционируют, так как требуется для стандартной работы отдела кадров компании.

1.2.3 Формирование требований, необходимых для автоматизации

При анализе диаграмм были найдены следующие недостатки существующего исполнения процесса «Деятельность отдела кадров»:

- отсутствие автоматизированной информационной системы негативно сказывается на скорости работы с информацией;
- затрата времени руководителей отделов в ходе приема на работу соискателей, что в результате приводит к нерациональному использованию ресурсов компании.

Исходя из вышеперечисленных недостатков, было принято решение о выявлении требования к проектируемой информационной системе, согласно которым она должна иметь:

- наличие отчетов;
- наличие возможности печати;
- распределение доступа;
- создание пользователей и ведение базы;
- простой интерфейс;
- возможность развернуть систему на сервере;

- отображение результатов;
- возможность использования системы на CMS-платформах;
- не дорогой по разработке или по цене;
- возможность изменения программистом на предприятии.

Требования к программному интерфейсу:

- интерфейс должен быть интуитивно понятным;
- должна быть понятная навигация на сайте.

Требования к проектируемой системе сформированы, что позволяет на их основании провести анализ рынка программного обеспечения, а также сравнительный анализ имеющихся продуктов.

1.3 Анализ существующих технологий автоматизации работы отдела кадров

1.3.1 Используемое программное обеспечение для кадрового учета

На момент проведения исследования в ООО «Успех» применялись различные прикладные программные средства для кадрового учета, которые также являются самыми распространенными программными продуктами.

В деятельности отдела кадров компании применяется следующее программное обеспечение:

- 1С:Зарплата и управление персоналом (ведение кадрового учета, формирование отчетности);
- Microsoft Office (работа с различными видами документов: тексты, базы данных, таблицы, презентации);
- ABBYY FineReader (распознавание информации в документах, предоставленных в графическом формате);
- Гарант (поиск юридической информации по кадровому делу);
- Консультант Плюс (поиск юридической информации по кадровому делу);

– Skype (осуществление коммуникаций, проведение собеседований).

Таким образом, отдел кадров компании ООО «Успех» обеспечен наличием программных продуктов для кадровой работы, но они не отвечают всем требованиям, выявленным в предыдущем подразделе. В частности, для подготовки и заполнения анкет на работу сотрудников сегодня применяется программа Word, Excel используется для обработки и представления полученных результатов.

1.3.2 Сравнительная характеристика существующего программного обеспечения для ведения кадрового учета

На рынке программного обеспечения существует множество программных разработок для ведения кадрового учета, которые включают генерацию документов, в том числе и кадровых приказов и других видов документов. Были рассмотрены наиболее популярные программные решения. Результаты сравнительного анализа представлены в таблице 1.

Таблица 1 – Сравнительная оценка аналогов программного обеспечения

Требование / Аналог	GenDOC FT v1.5	Докс.Онлайн	Aspose.Words for .NET
1	2	3	4
Наличие отчетов	+	-	-
Наличие возможности печати	+	+	+
Распределение доступа	-	-	-
Создание пользователей и ведение базы	-	-	-
Простой интерфейс с широким функционалом	+	-	-
Доступ из любого необходимого места	+	+	+
Возможность развернуть систему на сервере	+	+	+
Отображение результатов	+	+	+

Продолжение таблицы 1

1	2	3	4
Возможность использования системы на CMS-платформах	-	-	-
Не дорогой по разработке или по цене	+	+	+
Возможность изменения программистом на предприятии	+	+	+

Принцип работы генератора документов GenDOC FT основан на создании офисных документов (doc, rtf, txt, pdf, xls) по шаблону используя данные, находящиеся на странице в Excel.

Программа имеет ряд преимуществ перед другими аналогичными программами: доступны все возможности Excel; не требуется установка новых программ; не требуется знание новых программ.

Недостатки GenDOC FT: отсутствует бесплатная техническая поддержка; не современный визуальный интерфейс; отсутствие возможности использования баз данных.

Докс.Онлайн – инструмент для генерации документов по шаблону. Функции: возможность готовить персональные коммерческие предложения на основании шаблонов; неограниченное количество шаблонов, удобное создание и редактирование, ведение реестра документов; генерация любых документов, возможность интеграции с любой системой; для создания шаблона не нужны специальные знания, легко создать по примеру типовых документов и следуя инструкции; шаблоны хранятся в Google Docs, поэтому редактировать их можно совместно с коллегами.

Aspose.Words for .NET «представляет собой продвинутый API предназначенный для того, чтобы обработать документы формата Word, и который дает возможность для проведения ряда задач сразу из приложения на .NET.

Aspose.Words for .NET включает в себя определенные наборы API, которые предназначены для того, чтобы их можно было использовать с фреймворком .NET, с ядром .NET и платформой Xamarin, что позволяет создавать кроссплатформенные приложения, обладающие возможностями генерировать, преобразовывать, модифицировать, печатать документы без использования Microsoft Word.

Aspose.Words for .NET поддерживает фреймворк .NET, .NET Core & платформу Xamarin. Может быть использован с Mono & некоторыми другими платформами через COM Interop» [5].

Возможности программы: генерация писем по шаблону; генерация отчетов по шаблону; вставка форматированного текста, параграфов, изображений, таблиц и другого контента в документы Word; заполнение таблиц в документах Word с использованием выгрузки данных из базы данных; создание ярлыков для писем; создание почтовых сообщений через слияние данных из нескольких таблиц; вставка водяные знаки в документ; объединение и разделение документов; установка защиты на документ; поиск и замена текста, перенумерованный контент документа; определение цифровой подписи в документах.

Рассмотренные программные средства предназначены для автоматизации процессов работы с документами в организации, позволяют генерировать, сохранять и печатать документы, в том числе и кадровые приказы, обеспечивающие работу отдела кадров организации.

У каждой системы имеются свои преимущества и недостатки, и можно сделать вывод, что требуется разработать систему, которая будет подходить по всем требованиям автоматизации.

Для этого следует провести обоснование структуры и функциональности автоматизированной информационной системы кадрового учета, где будут описаны основные задачи для автоматизации работы отдела кадров ООО «Успех».

1.4 Постановка задачи на разработку проекта АИС

Информационное обеспечение системы управления кадрами – это, по сути, комплекс установленных решений по количеству, размещению и формам организации информации, которая постоянно присутствует в управленческой системе в процессе ее работы. Информационное обеспечение содержит актуальные данные, нормативно-справочную информацию, классификаторы технико-экономической информации и внутренние системы документации.

АИС применяет только ту информацию, которая необходима для наиболее благоприятной реализации стратегических, тактических и оперативных задач кадровой политики.

В результате, описывая особые аспекты функциональности АИС кадрового учета, можно выделить ряд конкретных групп функций:

- группа функций АИС «Учет деятельности персонала»;
- группа функций АИС «Перестановки персонала»;
- группа функций АИС «Обучение персонала»;
- группа функций АИС «Мотивация и стимулирование труда персонала»;
- группа функций АИС «Формирование кадровых документов»;
- группа функций АИС «Учет кадровых документов» отвечает за учет утвержденных (подписанных) кадровых документов.

Автоматизированная система кадрового учета должна быть направлена на оптимизацию деятельности управления кадрами, а также приводить к росту их производительности. Процессы автоматизации хранения и обработки данных сотрудников увеличивает эффективность работы отдела кадров.

Таким образом, в результате изучения и анализа предметной области были выделены задачи, которые будут автоматизированы посредством

разработки системы автоматизации кадрового учета в ООО «Успех».

Задачи автоматизации:

- учет рабочего времени;
- формирование отчетности;
- регулирование оборота документов, которые относятся к личному составу;
- учет кадровой информации;
- регулирующая деятельность по отношению к процедурам поддержания квалификации сотрудников;
- деятельность по кадровому учету и слежение за вакансиями;
- регулирующая деятельность по отношению к развитию сотрудников;
- регулирующая деятельность по отношению к режиму организации;
- формирование и печать кадровых документов.

В процессе функционирования программы, необходим ряд структурных элементов, которые содержат результирующую информацию, созданную на основании данных, полученных из базы данных и алгоритмов программного средства. Перечень страниц, которые генерирует программа: сотрудники; мотивация сотрудников; отпуск сотрудников; повышение квалификации; формирование кадровых документов; тренинги; командировки; должности; отделы; кадровые приказы.

Таким образом, в результате проектирования структуры приложения была создана структура информационной системы, которая включает ряд функций, позволяющих автоматизировать деятельность отдела кадров ООО «Успех». Структура информационной системы включает следующие элементы: сотрудники; мотивация сотрудников; отпуск сотрудников; повышение квалификации; формирование кадровых документов; тренинги; командировки; должности; отделы; кадровые приказы.

1.5 Разработка функциональной модели бизнес-процесса «Как должно быть»

На основе выявленных недостатков работы отдела кадров ООО «Успех» разрабатывается модель «Как должно быть», которая изображена на рисунке 4.

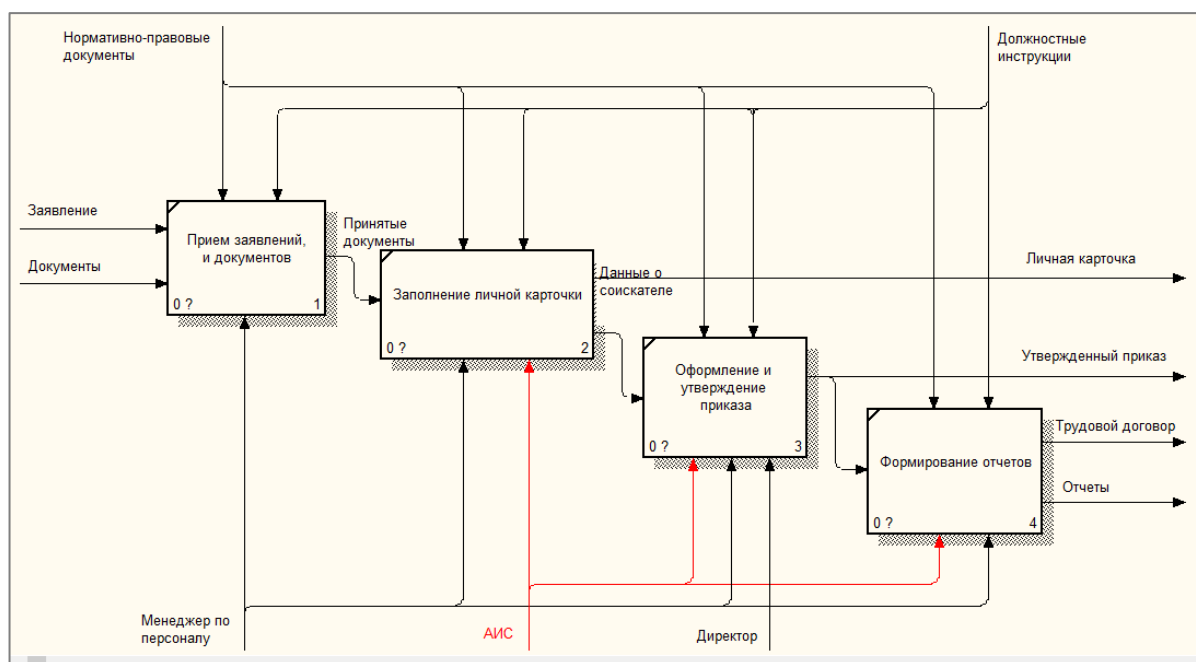


Рисунок 4 – Модель бизнес-процесса «Как должно быть»

Использование проектируемой автоматизированной информационной системы наиболее целесообразно на этапах обработки документов и формирования отчетов.

Вывод по первой главе

В первой главе была дана характеристика отдела кадров ООО «Успех». Представлены функциональные модели процесса кадрового учета «Как есть» и «Как должно быть». Проведен анализ автоматизированных информационных систем для данной предметной области. Выполнена постановка задачи на разработку проекта АИС. Определены группы функций АИС кадрового учета.

Глава 2 Логическое проектирование автоматизированной информационной системы кадрового учета

2.1 Логическая модель и её описание

Под логическим проектированием понимается процесс трансформации информационных требований, заключенных в информационной структуре. Результатом является база данных с СУБД-ориентированной структурой, а также спецификационные документы по программам прикладного назначения. На данном этапе происходит сравнительный анализ между разными моделями СУБД, которые были смоделированы предварительно.

Понятие «логический уровень» являет собой записи, элементы данных и взаимосвязи между записями.

Большая часть существующих сегодня методик для создания БД за основу берет различные ER–модели. Таким образом, моделирование предметной области опирается при своем создании применение графических диаграмм, включающих сравнительно маленькое количество элементов [2].

Основными конструктивными элементами модели «сущность-связь» (entity-relationship model, ER-model) являются сущности, их свойства (атрибуты) и связи между сущностями.

Атрибут, связь и сущность – главные термины ER–модели. У сущности имеются атрибуты – уточняющая деталь по отношению к классификации, идентификации, выражению состояния, числовой характеристики.

Связью является ассоциация, которая реализуется посредством графики. Устанавливается между сущностями.

Сущностью является объект, для которого должна быть доступна и сохранена информация. Все экземпляры сущности должны быть отличными от иных, относящихся к той же сущности.

На рисунке 5 представлена логическая модель базы данных.

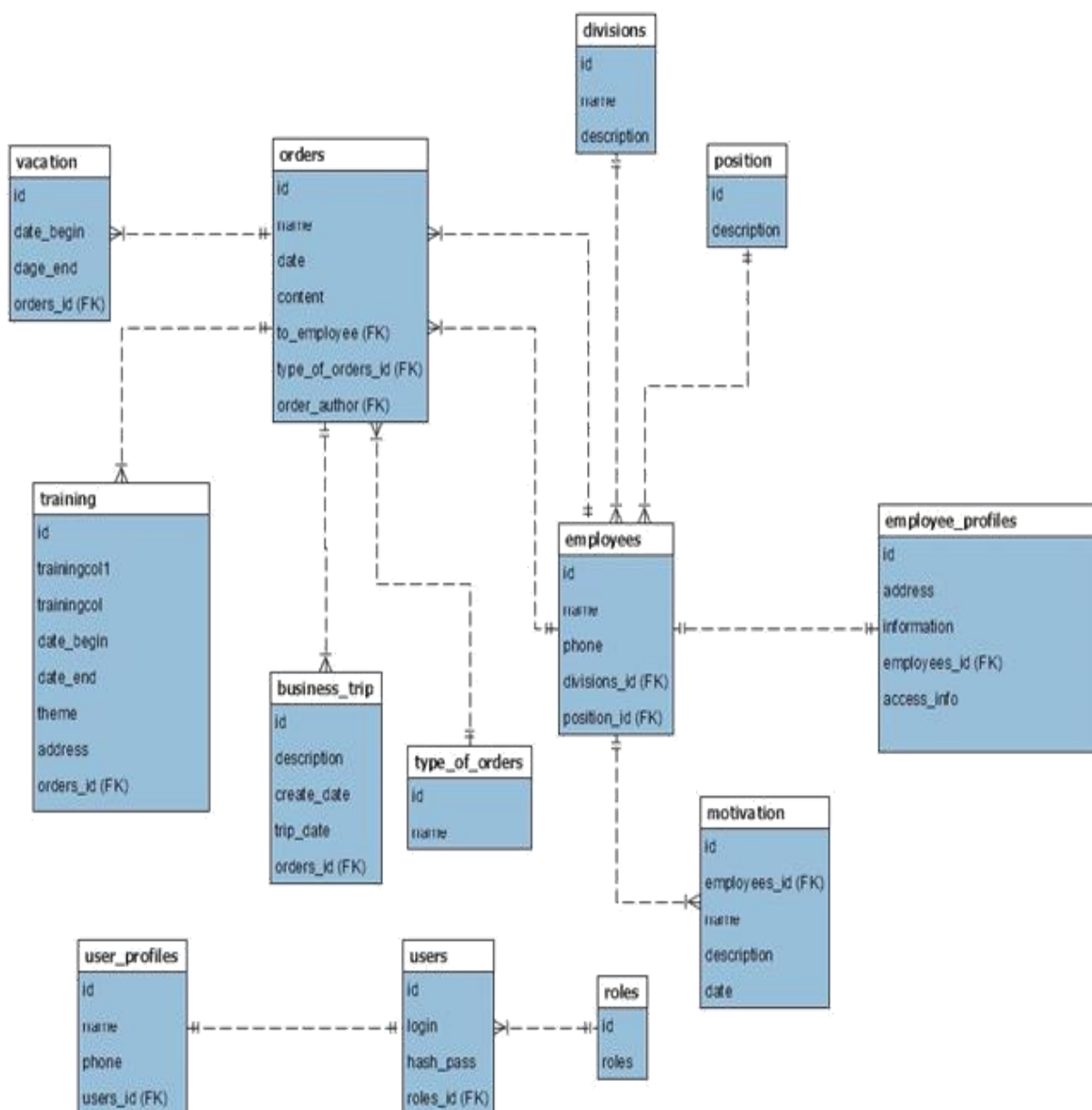


Рисунок 5 – Логическая модель базы данных

Данная модель представляет следующие сущности:

- сотрудники – employees;
- мотивация сотрудников – orders;
- отпуск сотрудников – vacation;
- повышение квалификации – motivation;
- тренинги – training;
- командировки – business_trip;
- должности – position;

- отделы – divisions;
- страницы пользователей – employee_profiles;

В результате логического моделирования предметной области была разработана логическая модель базы данных.

2.2 Информационное обеспечение автоматизированной информационной системы

2.2.1 Используемые классификаторы и системы кодирования

Кодирование – это процесс надления объектов кодовыми обозначениями.

Кодирование осуществляется с целью обозначения объектов в единообразном ключе, а также обеспечить необходимую достоверность информации, которая кодируется.

Система кодирования выбирается в зависимости от того, сколько классификационных признаков имеют объекты, и какая система классификации разработана.

Система классификации является совокупностью правил, по которым множество объектов делятся на подмножества.

Классифицирование является процессом распределения этого множества объектов на подмножества или результатом распределения объектов заданного множества по упорядоченным признакам. Система классификации может быть иерархической и многоаспектной.

Иерархическая система классификации – это когда исходное множество разбивается на подмножества, имеющие установленные отношения соподчиненности (иерархию).

Количество классификационных признаков определяет уровни классификации. Уровень классификации является совокупностью групп классификации, которые располагаются на одной ступени.

Многоаспектная система классификации предполагает использование нескольких классификационных признаков, не зависящих между собой.

Система кодирования порядкового типа – это последовательный порядок регистрации объектов. Здесь нет признаков классификации, а значит, в дальнейшем получить промежуточные итоги будет невозможно.

В таблице 2 представлены сводные характеристики классификаторов.

Таблица 2 – Сводная характеристика классификаторов

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Система классификации	Вид классификатора
Код отпуска	XXXXX	Порядковая	Иерархическая	Локальный
Код кадрового приказа	XXXXX	Порядковая	Иерархическая	Локальный
Код отдела	XXXXX	Порядковая	Иерархическая	Локальный
Код должности	XXXXX	Порядковая	Иерархическая	Локальный
Код сотрудника	XXXXX	Порядковая	Иерархическая	Локальный
Код командировки	XXXXX	Порядковая	Иерархическая	Локальный
Код типа заказа	XXXXX	Порядковая	Иерархическая	Локальный
Код профиля пользователя	XXXXX	Порядковая	Иерархическая	Локальный
Код профиля сотрудника	XXXXX	Порядковая	Иерархическая	Локальный
Код пользователя	XXXXX	Порядковая	Иерархическая	Локальный
Код роли	XXXXX	Порядковая	Иерархическая	Локальный
Код профиля пользователя	XXXXX	Порядковая	Иерархическая	Локальный

Характеристика классификаторов: использование пятизначного кода, порядковой системы кодирования, иерархической системы классификации и вид классификатора локальный.

Структурная формула классификатора: $\Phi = [XXXXX]$.

2.2.2 Характеристика нормативно-справочной и входной оперативной информации

Для решения задачи автоматизации работы отдела кадров используется оперативная информация. Описание нормативно-справочной информации представлено в таблице 3.

Таблица 3 – Перечень нормативно-справочных документов

Название документа	Содержание	Периодичность
Список сотрудников	Сотрудники	Перед внедрением
Список кадровых приказов	Кадровые приказы	Перед внедрением
Список тренингов	Перечень тренингов	Перед внедрением
Список покупателей	Покупатели	Перед внедрением
Список должностей	Должности	Перед внедрением
Список ролей	Роли	Перед внедрением
Список мотиваций	Средства мотивации сотрудников	Перед внедрением
Список отпусков	Отпуска сотрудников	Перед внедрением

Входной информацией являются данные о сотрудниках, отделах, должностях, тренингах, профилях.

2.3 Характеристика результатной информации

Вся результатная информация создается по запросу, составляется в электронном виде и при необходимости выводится на носитель.

Результатная информация включает в себя страницы приложения, которые можно представить в виде модулей системы:

- главная страница;
- регистрация;
- авторизация;
- сотрудники;
- мотивация сотрудников;
- отпуск сотрудников;
- повышение квалификации;
- формирование кадровых документов;
- тренинги;
- командировки;
- должности;
- отделы;
- кадровые приказы.

2.4 Требования к аппаратно-программному обеспечению АИС

В составе технического обеспечения информационной системы используются: сервер базы данных и приложений, персональные компьютеры пользователей.

Системные требования (64-разрядный сервер):

- процессор с архитектурой x86-64 (Intel с поддержкой EM64T, AMD с поддержкой AMD64);
- оперативная память 2048 Мб и выше;
- жесткий диск 40Гб и выше;
- SVGA-видеокарта.

Системные требования (ПК пользователя):

- процессор Intel Pentium Celeron 1800 МГц и выше;
- оперативная память 256 Мб и выше;
- жесткий диск 40Гб и выше;
- SVGA-видеокарта.

Требования к программному обеспечению сервера: на сервере для работы системы, не имеет значения, какая операционная система установлена, но требуется установленный Apache и MySQL для работы сервера.

Для работы с базой данных требуется установка любого инструмента для проектирования баз данных, например, Workbench, Navicat.

Web-приложение требует установки одного из следующих интернет-обозревателей: Mozilla FireFox, Google Chrome, Opera. Для администрирования на клиентской машине дополнительно требуется установка инструментов для проектирования баз данных для удаленного управления.

Вывод по второй главе

Во второй главе осуществлен выбор технологии логического моделирования автоматизированной информационной системы.

Рассмотрены используемые классификаторы и системы кодирования. Дана характеристика входной и выходной информации. Определены функции автоматизированной информационной системы кадрового учета ООО «Успех».

Выполнено логическое проектирование базы данных приложения, которая включает таблицы: `business_trip`, `divisions`, `employee_profiles`, `employees`, `motivation`, `orders`, `position`, `roles`, `training`, `type_of_orders`, `user_profiles`, `users`, `vacation`.

Определены требования к аппаратно-программному обеспечению автоматизированной информационной системы кадрового учета.

Глава 3 Физическое проектирование автоматизированной информационной системы кадрового учета

3.1 Выбор архитектуры АИС

Концепция клиент-сервер строится на том, что центральный сервер, кроме хранения файлов базы данных, должен также заниматься основной обработкой данных. Обращение к серверу пользователей осуществляется через специальный язык структурированных запросов (SQL), которым описан список выполняемых сервером задач. Сервер принимает запросы и запускает обработку данных. Пользователю в ответ присылается уже отработанный набор данных. С помощью технологии клиент-сервер можно избежать пересылки больших объемов информации в сети, так как вся обработка будет осуществляться центральным сервером. Также данный подход устраняет конфликты, если один и тот же файл редактируется несколькими пользователями.

Функционирование трехуровневой архитектуры («Тонкий клиент»-сервер приложений-сервер базы данных) происходит в интернете и Интернет-сетях. Та часть, которая взаимодействует с клиентом («тонкий клиент»), представлена HTML-страницей в веб-браузере или в качестве Windows-приложения, работающего с веб-сервисами. На сервер приложений вынесена вся программная логика, с его помощью формируются все запросы к базе данных, которые пересылаются серверу базы данных для выполнения. Сервер приложений может быть сформирован, как веб-сервер или как специализированная программа.

Распределенная база данных размещена на нескольких компьютерах. Возможно пересечение информации или ее дублирование. Чтобы управлять данными базы данных, используется система управления распределенными базами данных. Обращения пользователей к данным, которые находятся на других компьютерах, скрыты в системе [7].

Веб-сервер использует архитектуру клиент-сервера. Пользователь входит через свой браузер на сайт и там, в зависимости от задач пользователя, выполняются запросы, которые подразумевают собой отправку запроса на сервер. На сервере происходит обработка запроса, собираются данные, и отправляется ответ обратно пользователю.

3.2 Выбор технологии проектирования базы данных АИС

Проектируемая система предназначена для автоматизации работы отдела кадров. Система реализована в виде web-сайта, поэтому для её разработки требуются языки для web-программирования.

Имеются стандартные средства, без которых не обходится ни один сайт, такими являются: HTML, CSS, Javascript.

HTML используется, когда создаются веб-страницы. Браузер интерпретирует (обрабатывает) данный язык, отображая его, как документ, который человеку удобен для восприятия [8].

CSS представляет собой язык, позволяющий описывать внешний вид документа, при написании которого использовался язык разметки HTML. Чтобы описать структуру, пользуются языком HTML, чтобы описать её внешний вид, используются именно CSS [5].

JavaScript – является языком программирования, который позволяет придать веб-странице интерактивность в браузере, в результате чего данный язык получил широкое применение. Язык JavaScript выполняет задачу манипулирования элементами DOM-модели Веб-страницы [8].

Имея эти средства можно создать простой сайт без какой-либо логики и не особо сложных сценариев.

Для полноценной разработки web-приложения требуется выбрать язык программирования более высокого уровня. Существует множество языков программирования, предназначенных для выполнения различных задач. Каждый из них характеризуется уникальным набором операторов и особым

синтаксисом. Рассмотрены три наиболее популярных языка, применяемых в веб-разработке: PHP, Ruby и Python.

PHP – это язык сценариев общего назначения, используемый достаточно широко, имеющий открытый исходный код. PHP является языком программирования, который разрабатывался именно, чтобы создавать web-приложения (сценарии), исполнение которых происходит на web-сервере. PHP имеет такое преимущество, как способность быстро создавать динамически генерируемые веб-страницы, что очень удобно разработчикам сайтов.

Ruby является динамическим императивным объектно-ориентированным языком программирования. Его характерной чертой является динамическая типизация и автоматическое управление памятью. Использование языка Ruby целесообразно при веб-разработке, как часть открытого web-фреймворка Rails.

Для скриптов различного назначения широко используется Python в качестве интерпретируемого языка. Python, как и Ruby, преследует цель приближения синтаксиса написанной на нём реальной программы к псевдокоду, который описывает задачу, а значит, программист может уменьшить программный объем.

Сравнение указанных выше языков представлено в таблице 4.

Таблица 4 – Сравнение языков веб-программирования

Преимущества	Недостатки
PHP	
<ol style="list-style-type: none"> 1. Свободное программное обеспечение. 2. Легко осваиваемое. 3. Поддержка разработчиков. 4. Множество библиотек. 5. Возможность использования в локальной среде. 6. Возможность разворачивания практически на любом сервере. 	<ol style="list-style-type: none"> 1. Нельзя использовать в создании системных компонентов и десктопных приложений. 2. Недостаточно сильные средства для работы с исключениями 3. Сложность настройки сервера и разворачивание приложений 4. Проблемы с безопасностью

Продолжение таблицы 4

Преимущества	Недостатки
Ruby	
<ol style="list-style-type: none"> 1. Открытая разработка. 2. Совместимость с многими платформами. 3. Возможность внедрения HTML-разметки. 4. Передовые методы манипулирования текстами и строками. 5. Высокая интеграция высокопроизводительных серверов 6. Простой синтаксис 7. Простой программный интерфейс. 8. Возможность расширения языка через библиотеки. 9. Наличие встроенного отладчика. 	<ol style="list-style-type: none"> 1. Могут быть сложности при изучении языка. 2. Недостаточно информационных ресурсов об использовании. 3. Относительно других языков низкая производительность. 4. Длительная разработка.
Python	
<ol style="list-style-type: none"> 1. Открытая разработка. 2. Легок в освоении. 3. Хорошо читаемый код. 4. Быстрое прототипирование и динамическая семантика. 5. Множество полезных библиотек. 6. Унифицированный программный интерфейс и механизм импорта. 7. Хорошо разработанные механизмы модульности. 	<ol style="list-style-type: none"> 1. Плохая поддержка многопоточности. 2. Коммерческая поддержка средств разработки отсутствует. 3. Средства для работы с базами данных ограничены. 4. Низкая производительность.

В данной работе предлагается использовать язык программирования PHP. Данный язык программирования хорошо себя зарекомендовал среди web-разработчиков, имеет достаточную функциональность для разработки web-приложения.

Вся разработка происходит на платформе CMS (Content Management System – система управления содержимым), которая представляет собой инструментальную среду, позволяющую создавать и администрировать сайты. В системе управления сайтом или в CMS-движке предусмотрены готовые наборы функций для дизайна, шаблоны структуры и наполнения данными. Обслуживание ресурса с помощью CMS сайта более удобно: можно корректировать информацию на страницах, управлять доступом к контенту, совершать отправку почты [13].

На современном рынке существует множество разных CMS систем, но рассматриваться будут самые популярные и используемые системы: 1С:Битрикс, WordPress, Joomla и Drupal. На базе проведенного анализа, были сделаны сравнения и выявлены преимущества и недостатки каждой платформы (таблица 5).

Таблица 5 – Сравнение CSM платформ

Преимущества	Недостатки
1С-Битрикс	
<ol style="list-style-type: none"> 1. Множество разных модулей. 2. Административная панель интуитивно понятная и удобная. 3. Гибкая и кастомизированная система. 4. Наличие регулярных обновлений. 5. Имеется магазин готовых решений. 6. Наличие модуля поисковой оптимизации. 7. Интеграция с «1С: Предприятие». 	<ol style="list-style-type: none"> 1. Высокие требования к ресурсам. 2. Сложная архитектура и избыточность кода. 3. Привлечение профессионалов для интеграции с «1С: Предприятие». 4. Высокая стоимость разработки и поддержки. 5. Платное обновление.
Wordpress	
<ol style="list-style-type: none"> 1. «Движок» имеет свободное распространение. 2. Наличие шаблонов дизайна и плагинов. 3. Простая установка, настройка и администрирование. 4. Широкий функционал. 5. Доступность. 6. Интеграция с внешними сервисами: Instagram, Livejournal, Atom, онлайн-чатами, с ringback, trackback, RSS и пр. 	<ol style="list-style-type: none"> 1. Сбои при автоматическом обновлении. 2. Небольшой функционал базовой версии. 3. Вероятность конфликта плагинов. 4. Производство плагинов осуществляется другими компаниями. 5. Высокая загруженность сервера. 6. Низкий уровень безопасности.
Joomla	
<ol style="list-style-type: none"> 1. Свободно распространяемое обеспечение. 2. Наличие открытого исходного кода. 3. Высокая популярность. 4. Доступность, простая установка. 5. Наличие обучающих материалов. 6. Широкий функционал базовой версии. 7. Новостная рассылка, осуществляемая через встроенный менеджер. 8. Количество создаваемых страниц не ограничено. 	<ol style="list-style-type: none"> 1. Наличие сложной панели управления. 2. Сложность модернизации системы. 3. Отсутствие технической поддержки модулей. 4. Низкий уровень безопасности. 5. Трудности в оптимизации при работе с поисковыми системами.
Drupal	
<ol style="list-style-type: none"> 1. Наличие открытого исходного кода. 2. Гибкие конфигурации. 3. Множество различных модулей. 4. Минимальное время генерации страниц. 5. Отсутствие сложных запросов к БД. 	<ol style="list-style-type: none"> 1. Сложность в освоении. 2. Наличие сложного интерфейса. 3. Встроенные SEO-инструменты. 4. Сложности в установке. 5. Отсутствие технической поддержки.

На основании проведенного анализа можно сделать вывод. «1С-Битрикс» предназначена для работы больших, сложных и высоконагруженных проектов. Если сайт небольшой, то целесообразнее использовать более простую CMS, что будет способствовать экономии средств для того, чтобы развивать продукты и услуги.

WordPress также хорошо использовать, чтобы вести блоги или создавать небольшие коммерческие проекты. Однако, необходимо не забывать про риски. Может произойти одномоментное прекращение работы плагинов, которые применяются для функционала сайта. Соответственно, нецелесообразно использовать «движок», если разрабатываются интернет-магазин или большие и сложные сайты.

Для небольших проектов хорошо подходит Joomla. Данный движок подходит для создания информационных сайтов, интернет-магазинов несложной структуры, персональных страниц брендов. Аналогично другим таким же решениям, достоинства могут стать источником проблем – часто отсутствует техподдержка для подключаемых модулей, нет должной защиты и соответствующей документации.

Drupal является хорошим решением для тех, у кого есть навыки программирования. Кроме того, нужен будет мощный хостинг, чтобы обеспечивал высокую производительность и скорость работы сайта.

Для задачи данной работы подходит множество платформ, но выбор отдается WordPress, так как имеются навыки в работе с данной платформой, имеется понятный и настраиваемый интерфейс, имеется понятная и подробная документация.

3.3 Выбор системы управления базой данных АИС

Система управления базой данных, представляет собой комплекс программно-языковых средств для создания баз данных и управления данными. То есть, система управления базой данных является совокупностью

программ, которые позволяют разрабатывать, осуществлять контроль и администрирование баз данных. Функционирование многих сайтов невозможно при отсутствии баз данных, что обуславливает повсеместное использование системы управления базой данных.

В системе будет использоваться реляционная база данных, соответственно будут рассмотрены реляционные системы управления базой данных. Наиболее популярные реляционные системы управления базой данных: SQLite: очень мощная встраиваемая реляционная система управления базой данных; MySQL: самая популярная и часто используемая реляционная система управления базой данных; PostgreSQL: самая продвинутая и гибкая реляционная система управления базой данных.

Преимуществом SQLite является то, что это файловая база данных, то есть, все данные хранятся в одном файле, обеспечивая легкое перемещение. Она является стандартизированной, то есть, в SQLite используется SQL. Данную систему хорошо применять, когда база данных разрабатывается и тестируется: многие пользователи требуют на этапе разработки сделать масштабируемое решение. Учитывая, что в SQLite достаточно большой функционал, а работа в ней проста и удобна, она связана библиотекой, использовать её для разработок целесообразно. Недостатками SQLite являются то, что пользовательское управление отсутствует: пользователи в продвинутых базах данных могут управлять связями в таблицах на основе привилегий, в SQLite такой возможности нет. В данной системе нет возможности повысить производительность за счет изменения настроек.

MySQL является самой популярной среди всех крупных серверных баз данных. Несмотря на то, что MySQL не может обеспечить полную реализацию SQL-стандартов, она имеет достаточно большой функционал. Преимуществами MySQL являются то, что она относительно простая: установить MySQL достаточно легко. Имеются сторонние инструменты, в том числе, визуальные, которые упрощают первое ознакомление и работу с базой данных. Также в данной системе предусмотрен широкий функционал:

большая часть функций SQL поддерживаются в MySQL. Можно отметить высокую безопасность: MySQL имеет много встроенных функций, обеспечивающих безопасность. Данная система также является мощной и масштабируемой в MySQL можно обрабатывать очень большие объемы данных, что позволяет её использовать в масштабируемых приложениях. Кроме того, выделим высокую скорость: MySQL работает производительно за счет несоблюдения некоторых стандартов. Недостатками MySQL являются очевидные ограничения: MySQL по определению не может сделать все, она имеет ограниченный функционал.

PostgreSQL представляет собой наиболее продвинутую реляционную систему управления базы данных, которая в своей работе, прежде всего, ориентируется на стандарты, чтобы им соответствовать, и наличие возможности расширения. В PostgreSQL, или Postgres, обеспечено полное соответствие SQL-стандартам ANSI/ISO.

Отличием от других реляционных систем управления базами данных является то, что она имеет объектно-ориентированный функционал, а также полностью поддерживает концепт ACID (Atomicity, Consistency, Isolation, Durability). Преимуществами PostgreSQL являются абсолютная SQL-совместимость.

Это объектно-ориентированная система управления базой данных, то есть, она является не только реляционной. Недостатками PostgreSQL являются недостаточная производительность может хуже работать в простых операциях чтения, чем её конкуренты; она недостаточно популярна в силу своей сложности.

Учитывая указанные недостатки, поиск подходящего решения проблематичен. В данной работе будет использоваться MySQL, так как данная система проста в обращении, имеет достаточный функционал, не требовательна и соответственно не нагружает сервер, а также платформа системы управления содержимым работает на MySQL.

Роль системы управления базой данных будет выполнять MariaDB.

MariaDB является ветвью реляционной системой управления базой данных MySQL. Она имеет хорошую совместимость с приложениями, которые используют MySQL. Выбор в пользу MariaDB обусловлен тем, что развитие MySQL уже несколько заторможено.

MariaDB имеет улучшенный встроенный оптимизатор запросов, а также быструю и безопасную репликацию. В механизме хранения данных используются более быстрые индексы. Производительность перекодировки символов в MariaDB достаточно высокая, используется пул потоков. Кроме того, есть и другие улучшения, повышающие эффективность работы данной системы управления базой данных.

Установка системы управления базой данных не может происходить без веб-сервера, поэтому требуется перед установкой системы управления базой данных установить сервер, которым является программное обеспечение Apache.

Apache HTTP-сервер – это так называемый свободный веб-сервер, представляющий собой кроссплатформенное программное обеспечение. Apache имеет встроенный механизм виртуальных хостов. Это обеспечивает возможность обслуживания на одном IP-адресе множества веб-проектов (доменных имен), при этом каждое из них получает собственное отображение содержимого. Каждому виртуальному хосту можно отдельно настроить модуль и ядро, а, кроме того, возможно установление ограничений на доступ к отдельным файлам или ко всему сайту.

3.4 Разработка физической модели данных АИС

После выбора системы управления и логического моделирования была смоделирована БД на физическом уровне, то есть были сгруппированы методы доступа, данные и индексы. Представление модели можно увидеть на рисунке 6.

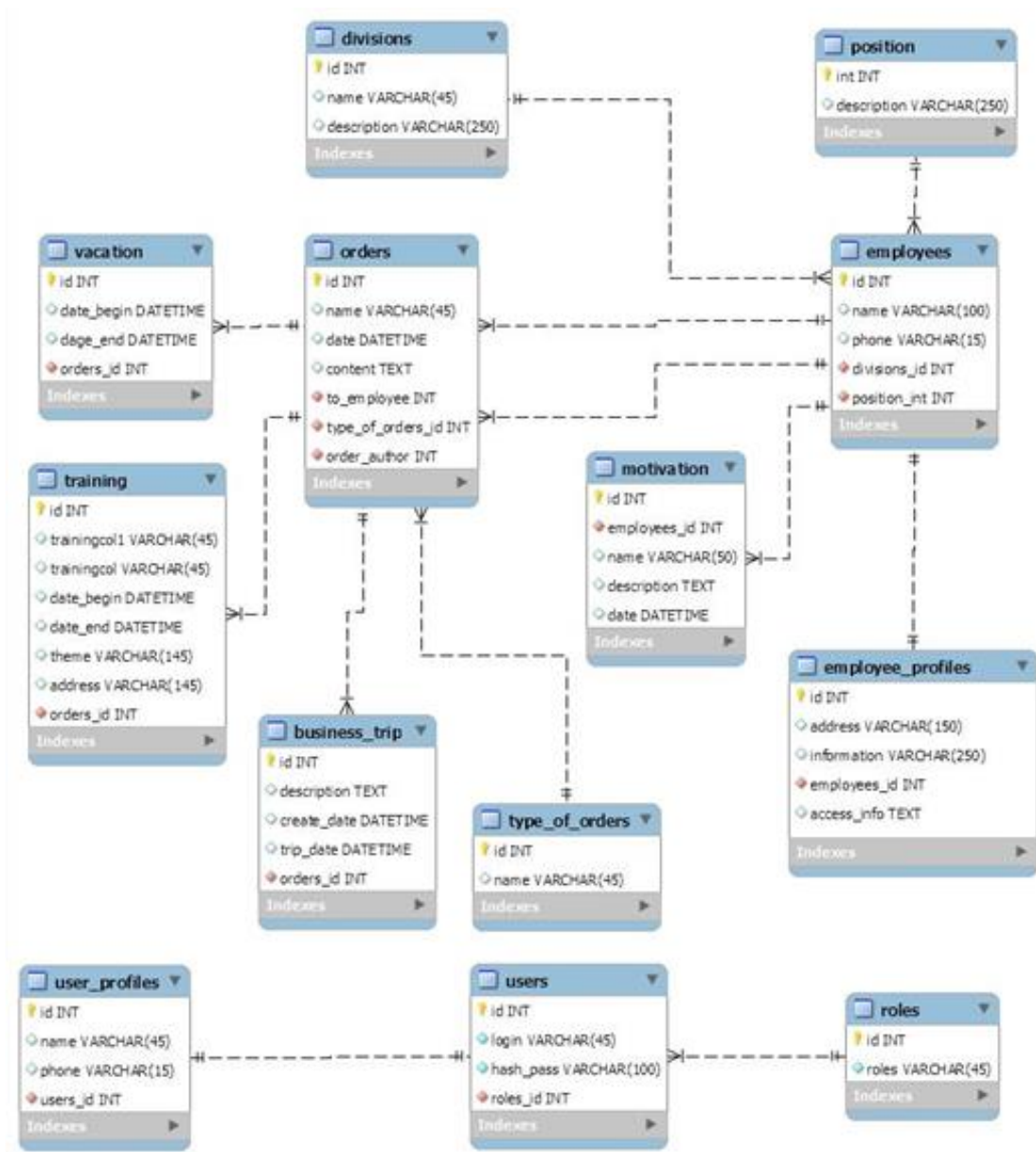


Рисунок 6 – Физическая модель данных

Любая физическая модель данных обязана отвечать характеристикам конкретной системы управления базами данных. Особенность конкретной системы управления БД отличается рядом ограничений на именование объектов базы данных, ограничений на те типы данных, которые поддерживает. Кроме того, отличительная черта конкретной системы управления БД во время физического моделирования содержит выбор решений, взаимосвязанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и

устройствам, методов доступа к данным), создание индексов [2].

SQL-скрипт создания базы данных для проекта в СУБД MySQL представлен в Приложении А. После выполнения запроса его необходимо выполнить на сервере MySQL.

В результате выполнения представленного SQL-скрипта на сервере MySQL, в базе данных приложения будут созданы таблицы, предназначением которых является хранение данных, которые обеспечивают работу проектируемого приложения.

Для создания базы данных необходимо скрипт выполнить на сервере MySQL. Перед этим с помощью phpMyAdmin необходимо создать базу данных, что показано на рисунке 7.



Рисунок 7 – Создание базы данных

Окно ввода запроса к базе данных показано на рисунке 8

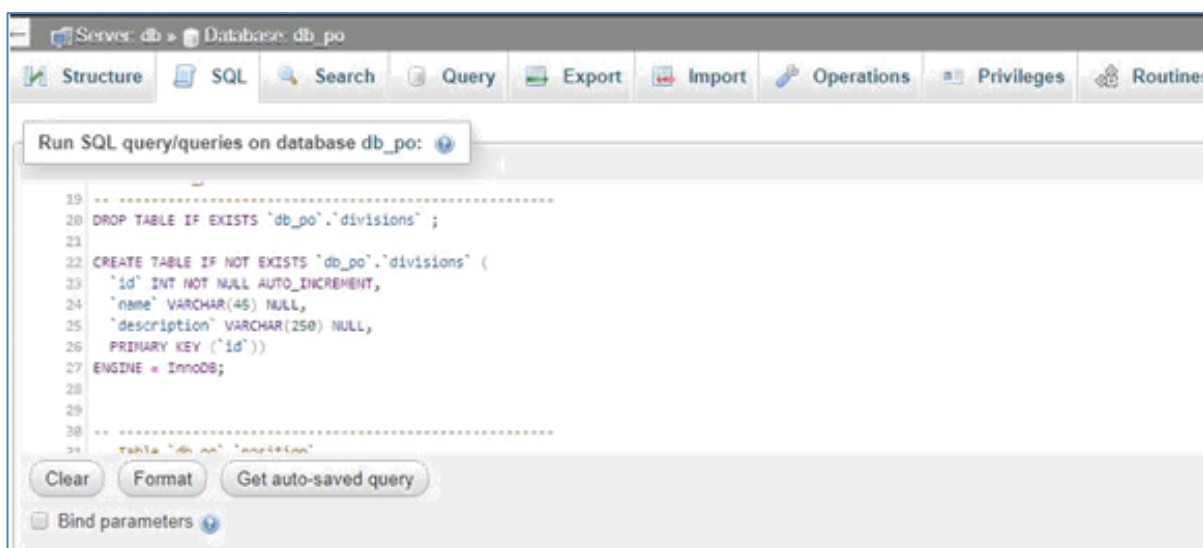


Рисунок 8 – Окно ввода SQL-запроса

После выполнения запроса на сервере MySQL были созданы таблицы для проекта. Структура созданной базы данных показана на рисунке 9.

Table	Action	Rows	Type	Collation	Size	Overhead
business_trip	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
divisions	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 K18	-
employees	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 K18	-
employee_profiles	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
motivation	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
orders	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	64.0 K18	-
position	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 K18	-
roles	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 K18	-
training	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
type_of_orders	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 K18	-
users	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
user_profiles	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
vacation	☆ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 K18	-
13 tables	Sum		InnoDB	utf8mb4_unicode_ci	400.0 K18	0 0

Рисунок 9 – Структура базы данных на сервере MySQL

В таблице «orders», которая показана на рисунке 10, сохраняются данные о кадровых приказах, которые включают название приказа, дату его создания, содержание кадрового приказа, сотрудник для которого предназначен приказ, вид кадрового приказа, сотрудник, который оформил (подписал) кадровый приказ.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(45)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
3	date	datetime			Yes	NULL			Change Drop More
4	content	text	utf8mb4_unicode_ci		Yes				Change Drop More
5	to_employee	int(11)			No	None			Change Drop More
6	type_of_orders_id	int(11)			No	None			Change Drop More
7	order_author	int(11)			No	None			Change Drop More

Рисунок 10 – Таблица «orders»

В результате проведенных действий была создана база данных на сервере MySQL, которая обеспечит хранение и обработку данных, которые требуется сохранять в приложении.

3.5 Разработка и описание модулей АИС с примерами кода

При входе в приложение, не авторизованному пользователю не предоставляется никаких возможностей для работы.

Для получения доступа к системе пользователю необходимо пройти регистрацию и дождаться активации его учетной записи.

За функционал регистрации отвечает модуль регистрации, который включает файлы register.php (рисунок 11) и RegistrationForm.php (рисунок 12).

```
<div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;">
<form method="post">
  <div class="form-group">
    <label for="inputEmail">Адрес электронной почты</label>
    <input type="email" class="form-control" id="inputEmail" placeholder="Ваш email"
      name="email"
      value="<?> $form->getEmail(); ?>">
  </div>
  <div class="form-group">
    <label for="InputUsername">Имя пользователя</label>
    <input type="text" class="form-control" id="InputUsername"
      placeholder="Ваше имя" name="login" value="<?> $form->getLogin(); ?>">
  </div>
  <div class="form-group">
    <label for="InputPassword">Пароль</label>
    <input type="password" class="form-control" id="InputPassword" placeholder="Пароль"
      name="password">
  </div>
  <div class="form-group">
    <label for="InputPasswordConfirm">Проверка пароля</label>
    <input type="password" class="form-control" id="InputPasswordConfirm"
      placeholder="Проверка пароля"
      name="passwordConfirm">
  </div>
  <button type="submit" class="btn btn-primary">Отправить</button>
  <a href="index.php" class="btn btn-primary">Отмена</a>
</div>
```

Рисунок 11 – Работа с файлом register.php

```

4)
function __construct( Array $data ) {
    $this->email      = isset( $data['email'] ) ? $data['email'] : null;
    $this->login      = isset( $data['login'] ) ? $data['login'] : null;
    $this->password   = isset( $data['password'] ) ? $data['password'] : null;
    $this->passwordConfirm = isset( $data['passwordConfirm'] ) ? $data['passwordConfirm'] : null;
}

public function validate() {
    return ! empty( $this->email ) && ! empty( $this->login ) && ! empty( $this->password ) && ! empty( $this->passwordConfirm ) && $this->passwordsMatch();
}

public function passwordsMatch() {
    return $this->password == $this->passwordConfirm;
}
}

```

Рисунок 12 – Работа с файлом RegistrationForm.php

Модуль авторизации представлен файлами LoginForm.php (рисунок 13), login.php (рисунок 14), logout.php (рисунок 15).

```

3
class LoginForm {
4     private $login;
5     private $password;
6
7     /**
8      * @param array $data
9      */
10    public function __construct( Array $data ) {
11        $this->login = isset( $data['login'] ) ? $data['login'] : null;
12        $this->password = isset( $data['password'] ) ? $data['password'] : null;
13    }
14
15    /**
16     * @return bool
17     */
18
19
20
}

```

Рисунок 13 – Работа с файлом LoginForm.php

```

71
72 <form method="post">
73     <div class="form-group">
74         <label for="inputEmail">Имя</label>
75         <input type="text" class="form-control" id="inputEmail" placeholder="Ваш номер"
76             name="login"
77             value="<? $form->getLogin(); ?>"
78     </div>
79     <div class="form-group">
80         <label for="InputPassword">Пароль</label>
81         <input type="password" class="form-control" id="InputPassword" placeholder="Пароль"
82             name="password">
83     </div>
84     <button type="submit" class="btn btn-primary">Войти</button>
85     <a href="index.php" class="btn btn-primary">Отмена</a>
86 </form>

```

Рисунок 14 – Работа с файлом login.php

```
header.php x register.php x RegistrationForm.php x LoginForm.php x footer.php x
1 <?php
2 session_start();
3 require_once('includes/Session.php');
4
5 Session::destroy();
6
7 header( string: 'Location: index.php?msg=Вы вышли!');
```

Рисунок 15 – Работа с файлом logout.php

После авторизации пользователя ему предоставляется возможность формировать и печатать кадровые приказы. За функцию выбора вида документа отвечает файл add_personnel_order.php (рисунок 16), который формирует форму ввода данных кадрового приказа и предоставляет возможность его распечатать.

Созданные файлы приложения обеспечивают требуемый функционал для приложения, позволяют защитить приложение от несанкционированного доступа путем реализации системы регистрации и авторизации, позволяют создавать, просматривать и печатать кадровые приказы.

Внешний вид созданных страниц показан на рисунках 17-19.

```
<form method="post">
  <div class="form-group">
    <label for="name">Название</label>
    <input type="text" class="form-control" id="name" placeholder="Название (номер)"
      name="name"
      value=""
    >
  </div>

  <div class="form-group">
    <label for="date">Дата</label>
    <input type="datetime-local" class="form-control" id="date" placeholder="Дата"
      name="date"
    >
  </div>

  <div class="form-group">
    <label for="content">Содержание приказа</label>
    <input type="text" class="form-control" id="content" placeholder="Содержание приказа"
      name="content"
    >
  </div>

  <div class="form-group">
    <label for="to_employee">Для работника</label>
    <input type="text" class="form-control" id="to_employee" placeholder="№0 сотрудника"
      name="to_employee"
    >
  </div>

  <div class="form-group">
    <label for="type_of_orders_id">Тип приказа</label>
    <input type="text" class="form-control" id="type_of_orders_id" placeholder="Тип приказа"
      name="type_of_orders_id"
    >
  </div>
```

Рисунок 16 – Работа с формой создания кадрового приказа

Представленная на рисунке 17 форма предназначена для регистрации пользователя и включает поля «Адрес электронной почты», «Имя пользователя», «Пароль», «Подтверждение пароля». Внизу формы находятся кнопки «Отправить» и «Отмена».

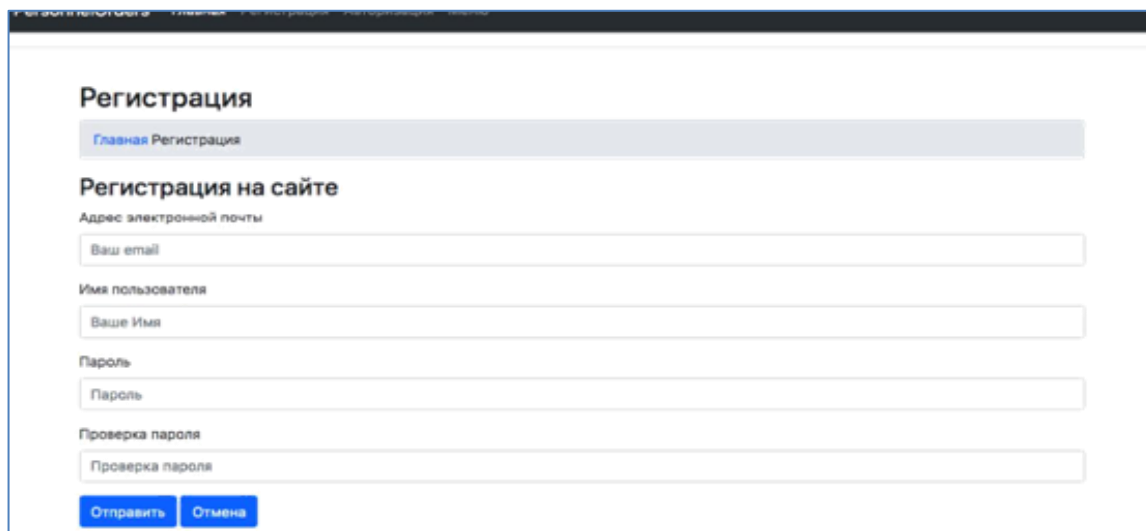
The image shows a web browser window with a registration form. The browser's address bar shows 'personeledgers'. The page title is 'Регистрация'. Below the title is a navigation bar with 'Главная' and 'Регистрация' (highlighted). The main heading is 'Регистрация на сайте'. The form includes four input fields: 'Адрес электронной почты' (with placeholder 'Ваш email'), 'Имя пользователя' (with placeholder 'Ваше имя'), 'Пароль' (with placeholder 'Пароль'), and 'Проверка пароля' (with placeholder 'Проверка пароля'). At the bottom of the form are two buttons: 'Отправить' and 'Отмена'.

Рисунок 17 – Внешний вид формы регистрации на сайте

На рисунке 18 показан внешний вид формы авторизации в системе. Форма авторизации включает поля ввода логина и пароля, а также кнопки «Войти» и «Отмена».

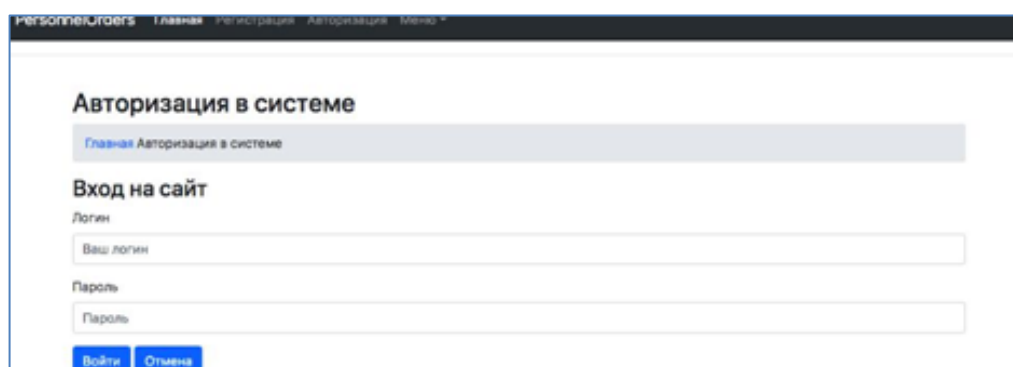
The image shows a web browser window with an authorization form. The browser's address bar shows 'personeledgers'. The page title is 'Авторизация в системе'. Below the title is a navigation bar with 'Главная', 'Регистрация', and 'Авторизация' (highlighted). The main heading is 'Авторизация в системе'. Below the heading is a navigation bar with 'Главная' and 'Авторизация в системе' (highlighted). The main heading is 'Вход на сайт'. The form includes two input fields: 'Логин' (with placeholder 'Ваш логин') and 'Пароль' (with placeholder 'Пароль'). At the bottom of the form are two buttons: 'Войти' and 'Отмена'.

Рисунок 18 – Внешний вид формы авторизации

Страница добавления кадрового приказа показана на рисунке 19. Форма добавления кадрового приказа включает поля для добавления

названия приказа, даты добавления, содержания приказа, ФИО работника, тип приказа, а также ФИО составителя приказа. Внизу формы находятся кнопки «Сохранить» и «Отмена».

The screenshot shows a web interface for adding a personnel order. The title is 'Добавить кадровый приказ'. The form contains the following fields and elements:

- Название:** Поощрение Иванова А. А.
- Дата:** 16/04/2020, 09:10
- Содержание приказа:** Includes a 'Файл' button and a 'Вид' dropdown menu.
- Preview:** A preview of the generated order document. It includes:
 - Header: 'информация организации' and 'до ОДЗД'
 - Table:

ПРИКАЗ	Помощь	Дата составления
	45 бр	16/04/2020
 - Text: '(распоряжение) о поощрении работника'
 - Employee info: 'Иванов А. А.', 'Табельный номер: 879076', 'Финанс. код: 000000'
 - Position: 'Бухгалтер', 'структурное подразделение: бухгалтер'
- Для работника:** Иванов А. А.
- Тип приказа:** (empty)
- Мотивация:** (empty)

Рисунок 19 – Формирование кадрового приказа

Форма добавления кадрового приказа включает поля для добавления названия приказа, даты добавления, содержания приказа, ФИО работника, тип приказа, а также ФИО составителя приказа. Внизу формы находятся кнопки «Сохранить» и «Отмена».

После формирования кадрового приказа его можно распечатать. Представленная возможность печати кадрового приказа позволяет как распечатать приказ на принтере, так и сохранить на компьютере в формате .pdf для дальнейшего использования. Таким образом, реализован удобный интерфейс взаимодействия пользователя с системой, обеспечена реализация описанного функционала за счет создания необходимых блоков программного кода.

3.6 Тестирование программного продукта

Тестирование предполагает проверку функционирования административной части сайта и внешнего интерфейса. Тестирование административной части: регистрация администратора; вход в административную часть сайта. Результаты тестирования административной части показали положительный результат.

Далее рассмотрим результаты тестирования внешнего интерфейса, которые представлены в таблице 6.

Таблица 6 – Результаты тестирования внешнего интерфейса

Наименование показателя	Chrome	FireFox	Microsoft Edge	Safari
Десктоп				
Корректность отображения	пройден	пройден	пройден	пройден
Ошибки JavaScript	отсутствуют	отсутствуют	отсутствуют	отсутствуют
Юзабилити	пройден	пройден	пройден	пройден
Мобильные устройства				
Корректность отображения	пройден	пройден	пройден	пройден
Ошибки JavaScript	отсутствуют	отсутствуют	отсутствуют	отсутствуют
Юзабилити	пройден	пройден	пройден	пройден

В таблице указано, что ошибки в конфигурации системы отсутствуют.

Проверка верности HTML-кода: для данного вида тестирования имеется большое количество утилит – от простых скриптов на perl до мощных валидаторов, проверяющих весь сайт на соответствие стандартам.

Для тестирования в данном исследовании был использован валидатор от World Wide Web Consortium, W3C по адресу <https://validator.w3.org/>

W3C – это организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины.

Результаты данного тестирования представлены в таблице 7. Фрагмент отчета по проверке с сайта W3C приведен в Приложении Б.

Таблица 7 – Результаты тестирования верности HTML-кода страниц сайта

Тип ошибки	Количество
Предупреждение	14
Ошибка	1

Все предупреждения относятся к базовым файлам WordPress, которые не относятся (не модифицируются) к проекту.

Ошибка является дублированным идентификатором элементов, которые создаются текущей Темой (Theme), то есть к созданному приложению также не имеющая отношения.

Следовательно, можно говорить об успешном прохождении тестирования разработанного программного продукта.

Выводы по третьей главе

С целью автоматизации кадрового учета, была разработана информационная система компании ООО «Успех». Разработка выполнена с помощью языков HTML, CSS, Javascript и PHP.

Использовался веб-сервер Apache с архитектурой клиент-сервер с использованием СУБД MySQL MariaDB, который обрабатывал SQL запросы для работы с базой данных.

Тестирование показало отсутствие существенных ошибок в кодах разработанного программного продукта.

Заключение

С целью автоматизации кадрового учета была разработана информационная система ООО «Успех».

В бакалаврской работе решены следующие задачи:

- проанализирована деятельность ООО «Успех»;
- изучены бизнес-процессы в организации;
- определены функции и структура информационной системы;
- спроектирована и создана информационная система кадрового учета на предприятии.

При выполнении работы изучена деятельность и дана характеристика компании ООО «Успех». При рассмотрении задач управления кадрами в организации проанализированы основные задачи и функции отдела кадров, а также бизнес-процессы в ООО «Успех». Представлена функциональная модель бизнес-процессов деятельности отдела кадров ООО «Успех».

Проведен сравнительный анализ программного обеспечения для ведения кадрового учета. Создана структура и определены функции информационной системы для отдела кадров.

Разработана логическая модель базы данных информационной системы. Спроектированные таблицы в базе данных обеспечивают реализацию функциональности информационной системы.

Выбрана СУБД для реализации проекта, определена среда проектирования базы данных, а также выбраны язык программирования для реализации проекта. В качестве СУБД выбрана MySQL, как язык программирования выбран PHP.

Заключительным этапом работы над проектом была организация экранных форм приложения и его тестирование.

В процессе работы получены практические навыки работы проектирования базы данных в СУБД MySQL и проектирования приложения на платформе WordPress.

Список используемой литературы

1. Блинов А. О. Реинжиниринг бизнес-процессов: учеб. пособие / А.О. Блинов. – М.: ЮНИТИ-ДАНА, 2015. – 343 с.
2. Варзунов А. В. Анализ и управление бизнес процессами: учеб. пособие / А. В. Варзунов, Е. К. Торосян, Л. П. Сажнева. – СПб: Университет ИТМО, 2016. –112 с.
3. Варламова Т.П. Финансово-экономический анализ: учеб. Пособие Т. П. Варламова, М. А. Варламова. – Саратов, 2016. –132 с.
4. Гоше, Х. HTML5. Для профессионалов / Х. Гоше. – СПб.: Питер, 2015. – 560 с.
5. Дакет Дж. HTML и CSS. Разработка и дизайн веб-сайтов / Дж. Дакет. – М.: Эксмо, 2020. – 480 с.
6. Долганова О. И. Моделирование бизнес-процессов: учебник и практикум для академического бакалавриата / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова; под ред. О. И. Долгановой. – М.: Издательство Юрайт, 2016. – 289 с.
7. Дронов В. А. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов: учеб. пособие / В. А. Дронов – СПб.: БХВ-Петербург, 2016. – 480 с.
8. Дунаев В. В. HTML, скрипты и стили: учеб. пособие / В. В. Дунаев. – СПб.: БХВ-Петербург, 2017. – 527 с.
9. Иванищева А. А. Анализ бесплатных популярных CMS платформ по созданию web-сайта / А. А. Иванищева, Х. И. Комилов, М. Д. Гехаев // Инновационная наука. – 2019. –№3. – С. 37-39
10. Колисниченко Д. Н. PHP и MySQL. Разработка Web-приложений: учеб. пособие / Д. Н. Колисниченко. – СПб.: БХВ-Петербург, 2017. – 560 с.

11. Костерин В. В. Разработка сайтов и web-приложений: учеб. пособие / В. В. Костерин, Е. В. Бунова, С. А. Богатенков. – Челябинск: Издательский центр ЮУрГУ, 2016. – 110 с.
12. Котляров В. П. Основы тестирования программного обеспечения: учеб. пособие / В. П. Котляров. – М.: ИНТУИТ, 2016. - 335 с.
13. Любимов Е. В. Управление бизнес-процессами: моделирование: учебное пособие / Е. В. Любимов. – Владивосток: Изд-во ВГУЭС, 2018. – 68 с.
14. Макфарланд Д. Новая большая книга CSS / Д. Макфарланд. – Спб.: Питер, 2019. – 70 с.
15. Мальцев И. П. Проектирование сайтов / И. П. Мальцев. – М.: SelfPub, 2018. – 118 с.
16. Михеев А. Г. Процессное управление на свободном программном обеспечении: учеб. пособие / А. Г. Михеев. – М.: ИНТУИТ, 2016. – 230 с.
17. Тельно Ю. Ф. Инжиниринг предприятия и управление бизнес-процессами: методология и технология : учеб. пособие / Ю. Ф. Тельнов, И. Г. Фёдоров. – М.: ЮНИТИ-ДАНА, 2015. – 207 с.
18. Харрис Э. PHP/MySQL для начинающих : учеб. пособие / Э. Харрис. – М.: КУДИЦ-Образ, 2016. – 384 с.
19. Beyer B., Lewandowski P., Oprea A., Blankinship P., Adkins H., Stubblefield A.. Building Secure and Reliable Systems : O'Reilly Media, 2020.
20. Duckett J., PHP & MySQL: Server-side Web Development. – Wiley, 2020.
21. Nixon R., Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Learning PHP, MYSQL, Javascript, CSS & HTML5). – O'Reilly Media, 2018.
22. Williams A. WordPress for Beginners 2019: A Visual Step-by-Step Guide to Mastering WordPress. – Independently published, 2018.
23. Krogh J. MySQL Concurrency : Apress, 2021.

Приложение А

Создание базы данных для проекта в СУБД MySQL

```
CREATE TABLE IF NOT EXISTS
`db_po`.`divisions` ( `id` INT NOT NULL
AUTO_INCREMENT, `name` VARCHAR(45)
NULL,
`description` VARCHAR(250) NULL,
PRIMARY KEY (`id`))
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`position` (
`int` INT NOT NULL AUTO_INCREMENT,
`description` VARCHAR(250) NULL,
PRIMARY KEY (`int`))
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS
`db_po`.`employees` ( `id` INT NOT NULL
AUTO_INCREMENT, `name` VARCHAR(100)
NULL,
`phone` VARCHAR(15) NULL,
`divisions_id` INT NOT NULL,
`position_int` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `fk_employees_divisions1_idx` (`divisions_id` ASC),
INDEX `fk_employees_position1_idx` (`position_int` ASC),
CONSTRAINT `fk_employees_divisions1`
FOREIGN KEY (`divisions_id`)
REFERENCES `db_po`.`divisions` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_employees_position1`
FOREIGN KEY (`position_int`)
REFERENCES `db_po`.`position` (`int`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS
`db_po`.`type_of_orders` ( `id` INT NOT NULL
AUTO_INCREMENT, `name` VARCHAR(45)
NULL,
PRIMARY KEY (`id`))
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`orders` (
`id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(45) NULL,
`date` DATETIME NULL,
`content` TEXT NULL,
`to_employee` INT NOT NULL,
`type_of_orders_id` INT NOT NULL,
```

Продолжение Приложения А

```
order_author` INT NOT NULL,  
PRIMARY KEY (`id`),  
INDEX `fk_orders_employees1_idx` (`to_employee` ASC) ,  
INDEX `fk_orders_type_of_orders1_idx` (`type_of_orders_id` ASC) ,  
INDEX `fk_orders_employees2_idx` (`order_author` ASC) ,  
CONSTRAINT `fk_orders_employees1`  
FOREIGN KEY (`to_employee`)  
REFERENCES `db_po`.`employees` (`id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT `fk_orders_type_of_orders1`  
FOREIGN KEY (`type_of_orders_id`)  
REFERENCES `db_po`.`type_of_orders` (`id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT `fk_orders_employees2`  
FOREIGN KEY (`order_author`)  
REFERENCES `db_po`.`employees` (`id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
CREATE TABLE IF NOT EXISTS `db_po`.`roles` (  
`id` INT NOT NULL AUTO_INCREMENT,  
`roles` VARCHAR(45) NOT NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB;  
CREATE TABLE IF NOT EXISTS `db_po`.`users` (  
`id` INT NOT NULL AUTO_INCREMENT,  
`login` VARCHAR(45) NOT NULL,  
`hash_pass` VARCHAR(100) NOT NULL,  
`roles_id` INT NOT NULL,  
PRIMARY KEY (`id`),  
INDEX `fk_users_roles_idx` (`roles_id` ASC) ,  
CONSTRAINT `fk_users_roles`  
FOREIGN KEY (`roles_id`)  
REFERENCES `db_po`.`roles` (`id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
CREATE TABLE IF NOT EXISTS `db_po`.`motivation` (  
`id` INT NOT NULL  
AUTO_INCREMENT, `employees_id` INT NOT NULL, `name` VARCHAR(50) NULL,  
`description` TEXT NULL,  
`date` DATETIME NULL,  
INDEX `fk_forms_employees1_idx` (`employees_id` ASC) ,  
PRIMARY KEY (`id`),  
CONSTRAINT `fk_forms_employees1`  
FOREIGN KEY (`employees_id`)  
REFERENCES `db_po`.`employees` (`id`)  
ON DELETE NO ACTION
```

Продолжение Приложения А

```
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`user_profiles` ( `id` INT NOT NULL
AUTO_INCREMENT, `name` VARCHAR(45) NULL,
`phone` VARCHAR(15) NULL,
`users_id` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `fk_user_profiles_users1_idx` (`users_id` ASC) , CONSTRAINT
`fk_user_profiles_users1` FOREIGN KEY (`users_id`)
REFERENCES `db_po`.`users` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`business_trip` (
`id` INT NOT NULL AUTO_INCREMENT,
`description` TEXT NULL,
`create_date` DATETIME NULL,
`trip_date` DATETIME NULL,
`orders_id` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `fk_business_trip_orders1_idx` (`orders_id` ASC) ,
CONSTRAINT `fk_business_trip_orders1`
FOREIGN KEY (`orders_id`)
REFERENCES `db_po`.`orders` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`employee_profiles` ( `id` INT NOT NULL
AUTO_INCREMENT, `address` VARCHAR(150) NULL,
`information` VARCHAR(250) NULL,
`employees_id` INT NOT NULL,
`access_info` TEXT NULL,
PRIMARY KEY (`id`),
INDEX `fk_employee_profiles_employees1_idx` (`employees_id` ASC) , CONSTRAINT
`fk_employee_profiles_employees1` FOREIGN KEY (`employees_id`)
REFERENCES `db_po`.`employees` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`vacation` ( `id` INT NOT NULL
AUTO_INCREMENT, `date_begin` DATETIME NULL, `dage_end` DATETIME NULL,
`orders_id` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `fk_vacation_orders1_idx` (`orders_id` ASC) ,
CONSTRAINT `fk_vacation_orders1`
FOREIGN KEY (`orders_id`)
REFERENCES `db_po`.`orders` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```


Продолжение Приложения А

```
ENGINE = InnoDB;
CREATE TABLE IF NOT EXISTS `db_po`.`training` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `trainingcol1` VARCHAR(45) NULL,
  `trainingcol` VARCHAR(45) NULL,
  `date_begin` DATETIME NULL,
  `date_end` DATETIME NULL,
  `theme` VARCHAR(145) NULL,
  `address` VARCHAR(145) NULL,
  `orders_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_training_orders1_idx` (`orders_id` ASC),
  CONSTRAINT `fk_training_orders1`
  FOREIGN KEY (`orders_id`)
  REFERENCES `db_po`.`orders` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB.
```

Приложение Б

Фрагмент отчета проверки верности HTML-кода

1. **Warning** : The `type` attribute is unnecessary for JavaScript resources.
From line 12, column 3; to line 12, column 33, ed/ `<script type="text/javascript">wi`
2. **Warning** : The `type` attribute for the `style` element is not needed and should be omitted.
From line 17, column 3; to line 17, column 25, script> `<style type="text/css">img.w`
3. **Warning** : The `type` attribute is unnecessary for JavaScript resources.
From line 35, column 1; to line 35, column 103
`=all' /><script type='text/javascript' src='http://max.ciman.localhost/wp-includes/js/jquery/jquery.js?ver=1.12.4-wp'></scri`
4. **Warning** : The `type` attribute is unnecessary for JavaScript resources.
From line 36, column 1; to line 36, column 111
`</script><<script type='text/javascript' src='http:// max.ciman.localhost /wp-includes/js/jquery/jquery-migrate.min.js?ver=1.4.1'></scri`
5. **Warning** : The `type` attribute for the `style` element is not needed and should be omitted.
From line 45, column 1; to line 45, column 23, t=xml" /> `<style type="text/css">.recen`