

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 Автоматизация формирования кадровых приказов на предприятии  
(на примере ООО «Марс»)

Студент

В. О. Соколова

(И.О. Фамилия)

(личная подпись)

Руководитель

Н. Н. Казаченок

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

Тольятти 2020



**Росдистант**  
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

## Аннотация

Бакалаврская работа посвящена актуальной тематике автоматизации формирования кадровых приказов на предприятии.

В процессе работы выявлены недостатки в деятельности кадровой службы предприятия в направлении работы с кадровыми приказами, их учетом и формированием; проанализированы бизнес-процессы кадровой службы, представлена функциональная модель бизнес-процессов функционирования отдела кадров ООО «Марс»; рассмотрены программы-аналоги, которые автоматизируют процессы функционирования кадровой службы и работу с документацией; обоснован выбор инструментов для создания системы - выбрана СУБД для реализации проекта, определена среда проектирования базы данных, а также выбраны язык программирования для реализации проекта. Для моделирования базы данных использованы возможности MySQL Workbench, как язык программирования выбран PHP; определена архитектура приложения; разработана модель данных для хранения информации кадровой службы, алгоритмы и структура приложения; проверена работоспособность созданной системы; проведен экономический расчет.

## Оглавление

Введение.....	4
Глава 1 Анализ предметной области, изучение бизнес-процессов и проектирование структуры информационной системы.....	6
1.1 Анализ деятельности предприятия ООО «Марс» .....	6
1.1.1 Понятие и виды кадровых приказов. Организационная структура компании ООО «Марс» .....	6
1.1.2 Управление кадрами в организации .....	8
1.2 Исследование бизнес-процессов предприятия .....	12
1.2.1 Выбор методологии исследования бизнес-процессов .....	12
1.2.2 Анализ бизнес-процессов предприятия.....	13
1.3 Сравнительный анализ систем для генерации форм документов....	15
1.4 Обоснование структуры и функционала информационной системы для отдела кадров.....	20
Глава 2. Логическое моделирование предметной области .....	25
2.1. Логическая модель и ее описание .....	25
2.2. Используемые классификаторы и системы кодирования.....	29
2.3. Характеристика нормативно-справочной и входной оперативной информации .....	30
2.4. Характеристика базы данных .....	30
2.5. Характеристика результатной информации.....	34
Глава 3. Физическое проектирование информационной системы.....	36
3.1. Выбор системы управления базами данных информационной системы .....	36
3.2. Разработка физической модели данных информационной системы .....	40
3.3 Разработка программного обеспечения.....	43
3.4. Оценка и обоснование экономической эффективности разработки информационной системы .....	50

3.4.1 Выбор методики расчёта экономической эффективности .....	50
3.4.2 Расчёт показателей экономической эффективности проекта .	51
Заключение .....	58
Список используемой литературы и список используемых источников.....	60
Приложение А Фрагмент кода авторизации в системе.....	66
Приложение Б Фрагмент кода регистрации в системе .....	75
Приложение В Фрагмент кода выхода с системы .....	78
Приложение Г Фрагмент кода добавления приказа .....	79
Приложение Д Создание базы данных для проекта в СУБД MySQL .....	80

## Введение

Современные информационные системы способствуют разработке решений и организации координационных мероприятий по выполнению задач. Информационные системы способствуют упрощению в процедуре анализа деятельности предприятия, в том числе оказывают помощь при планировании бизнес-процессов. Перед работодателем во время ведения хозяйственной деятельности часто могут возникать вопросы по управлению трудовыми взаимоотношениями с сотрудниками, организации труда и урегулированию данных процессов, которое осуществляется за счет учета законодательных нормативов и выполнения кадровых приказов руководства. В целом на сегодняшний день наблюдается отсутствие упорядоченной и структурированной системы, которая занимается вопросами кадрового делопроизводства. Таким образом, кадровое делопроизводство представляется как система, деятельность которой заключается в решении задач по управлению кадрами, а также в организации стабильной работы с документами по личному составу предприятия, который непосредственно имеет отношение к его сотрудникам, их перемещения внутри общества и учета рабочего времени.

Совершенствование системы управления персоналом, поиск новых решений, изучение современного опыта позволит повысить экономический эффект деятельности предприятия. Информатизация кадровой службы заключается в том, чтобы оптимально использовать существующие сегодня возможности информационных технологий в вопросах кадровой политики, а также в модернизации содержания, методик и организационных форм кадрового учета, формирования кадровых приказов на предприятии. Главным образом совершенствование осуществляется за счет автоматизации ряда бизнес-процессов, которые осуществляются в организации.

Объект исследования – бизнес-процессы деятельности организации по формированию кадровых приказов.

Предмет исследования – автоматизация формирования кадровых приказов на предприятии (на примере ООО «Марс»).

Цель работы – разработать веб-ориентированную информационную систему формирования кадровых приказов на предприятии (на примере ООО «Марс»).

Задачи работы:

- проанализировать деятельность ООО «Марс»;
- изучить бизнес-процессы в организации;
- разработать функционал и структуру информационной системы;
- спроектировать и создать информационную систему.

Методологической основой исследования является диалектический метод как общий подход к научному познанию предмета исследования. В процессе исследования применялся научный аппарат информатики, методы разработки программного обеспечения и моделирования бизнес-процессов.

Эмпирическую основу исследования составили материалы о деятельности ООО «Марс», нормативные документы, официальные документы разработчиков программного обеспечения, которое использовалось для создания информационной системы.

Теоретическая и практическая значимость исследования заключается в том, что разработанная в работе информационная система для организации может быть использована в практической деятельности предприятий. Теоретическая основа работы носит методический характер для создания подобных систем.

Структура работы: работа состоит из введения, трех глав, заключения, списка использованной литературы и приложений.

## **Глава 1 Анализ предметной области, изучение бизнес-процессов и проектирование структуры информационной системы**

### **1.1 Анализ деятельности предприятия ООО «Марс»**

#### **1.1.1 Понятие и виды кадровых приказов. Организационная структура компании ООО «Марс»**

Приказ представляет собой правовой документ. Ответственность за издание приказа возлагается непосредственно на сотрудника, который является руководителем органа управления (предприятия и т.п.). Основываясь на свойстве единства это необходимо, чтобы орган управления смог выполнить поставленные перед ним задачи оперативного характера и серьезной важности, приказ приводится в исполнение [19].

Требование вести учет документов в журнале определено определяется следующими нормативно-правовыми документами: Временные нормативы по проведению комплектационных и учетных работ по отношению к кадровому составу укрупненного межотраслевого характера; Списки документов, которые возникают в процессе управленческой деятельности и являются типовыми, с обязательной пометкой о сроках хранения документов.

Далее рассмотрим организационную структуру организации. Так, организационно-правовой формой анализируемого предприятия выступает общество с ограниченной ответственностью. Таким образом, строго отвечает по обязательствам в области имеющего у него имущества. Данное ООО выступает коммерческим предприятием.

Организационная структура рассматриваемой фирмы относится к линейно-функциональному типу. Это обуславливается тем, что компания придерживается принцип единоначалия в управлении, структурные отделы сформированы линейным методом, а функциональные задачи управления поделены между ними [21].

Организационная структура управления ООО «Марс» представлена на рисунке 1.

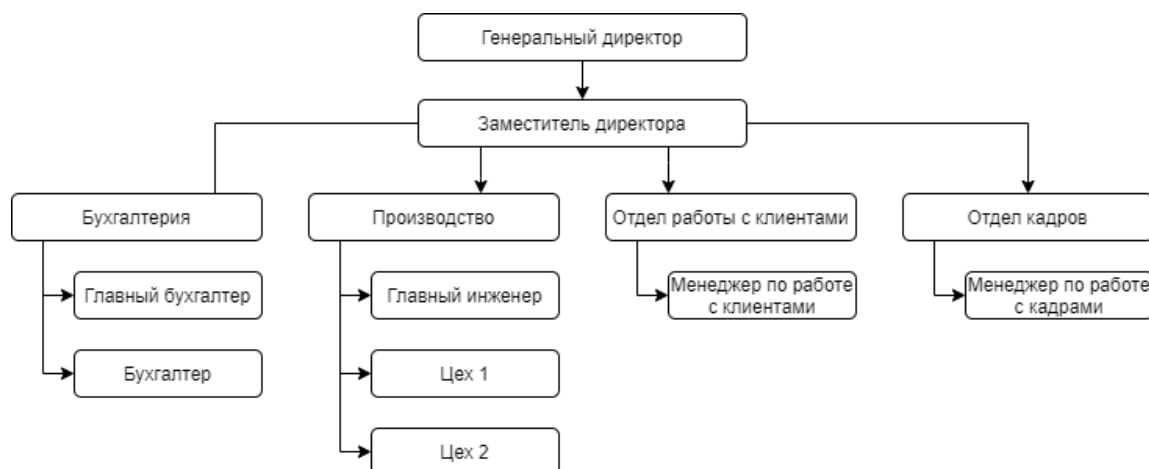


Рисунок 1 – Организационная структура управления ООО «Марс»

Как правило, увидеть официально установленную организационную структуру можно в штатном расписании. Она обеспечивает согласованность отдельных направлений деятельности компании и усилий подразделений по реализации и разрешению главных целей и задач.

ООО «Марс» – это компания, являющаяся юридическим лицом и функционирующая в соответствии с действующим Уставом и действующей законодательной системой РФ, а именно: Конституции Российской Федерации, Гражданского Кодекса, Трудового Кодекса и Налогового Кодекса.

ООО «Марс» представляет собой юридическое лицо, которое осуществляет свою деятельность согласно действующему Уставу и законодательной системе РФ, в частности: Конституции РФ, ГК РФ, ТК и НК РФ.

Предприятие имеет право в соответствии с утвержденным порядком осуществлять открытие банковских счетов в России и за ее границами.



### 1.1.2 Управление кадрами в организации

Общее число сотрудников компании ООО «Марс» на 31 декабря 2019 года равно 59 человек, в их числе 7 представителей руководства компании, 8 сотрудников являются административным персоналом, 17 менеджеров, 27 работников – это производственный и вспомогательный кадры.

С целью наиболее объективного анализа состава кадров анализируемого предприятия правильно обратить внимание на трудовые возможности сотрудников, осуществляющих свою трудовую деятельность в данной компании. Отметим, что наиболее значимо, оценивать персонал компании по следующим параметрам: возрастной состав и общий стаж трудовой деятельности в рассматриваемой компании, что показано в таблице 1.

Таблица 1 – Распределение персонала по возрасту в 2019 году, чел.

Всего работников	Число полных лет					
	до 25	25- 35	36- 45	46-55	56-59	60 и более
59	4	11	20	14	8	2

В процентном соотношении данные относительно распределения персонала по возрасту представлены на рисунке 2.

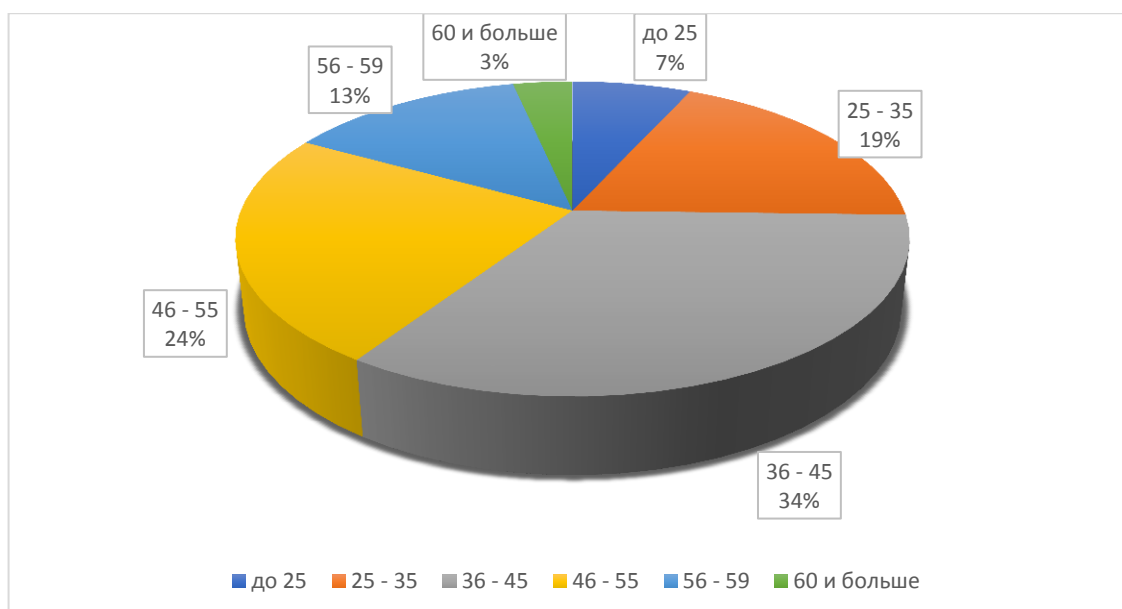


Рисунок 2 – Распределение кадров ООО «Марс» по возрасту в 2019 году, %

Исходя из сведений, приведенных в таблице 1 и на рисунке 1, возрастная категория работников преобладает в возрасте 36-45 лет (34,0%) и 46-55 лет (24%).

Кадровая служба ООО «Марс» несет ответственность за решение следующих задач:

- создание и воплощение в действительность стратегических решений по отношению к управлению кадрами;
- создание структурной системы, состоящей из необходимого количества руководящих должностей в компании, сотрудников требующихся профессий, рабочих специалистов, квалификации и специальностей;
- выполнение необходимых операций в области работы с трудовыми кадрами;
- создание трудового коллектива, который находится в стабильном состоянии и при этом развивается;
- регулярный процесс развития в области управленческой работы с кадровым составом [24];

- подготовка проектов кадровых приказов.

Действия по реализации управленческой деятельности кадрами показаны на рисунке 3.

Основная задача работ по привлечению и приему новых кадров в ООО «Марс» предполагает оперативное обеспечение общества, затрачивая на это минимум средств, соответствующим числом сотрудников, которые будут отвечать на требования, предоставленных на момент вакансий.

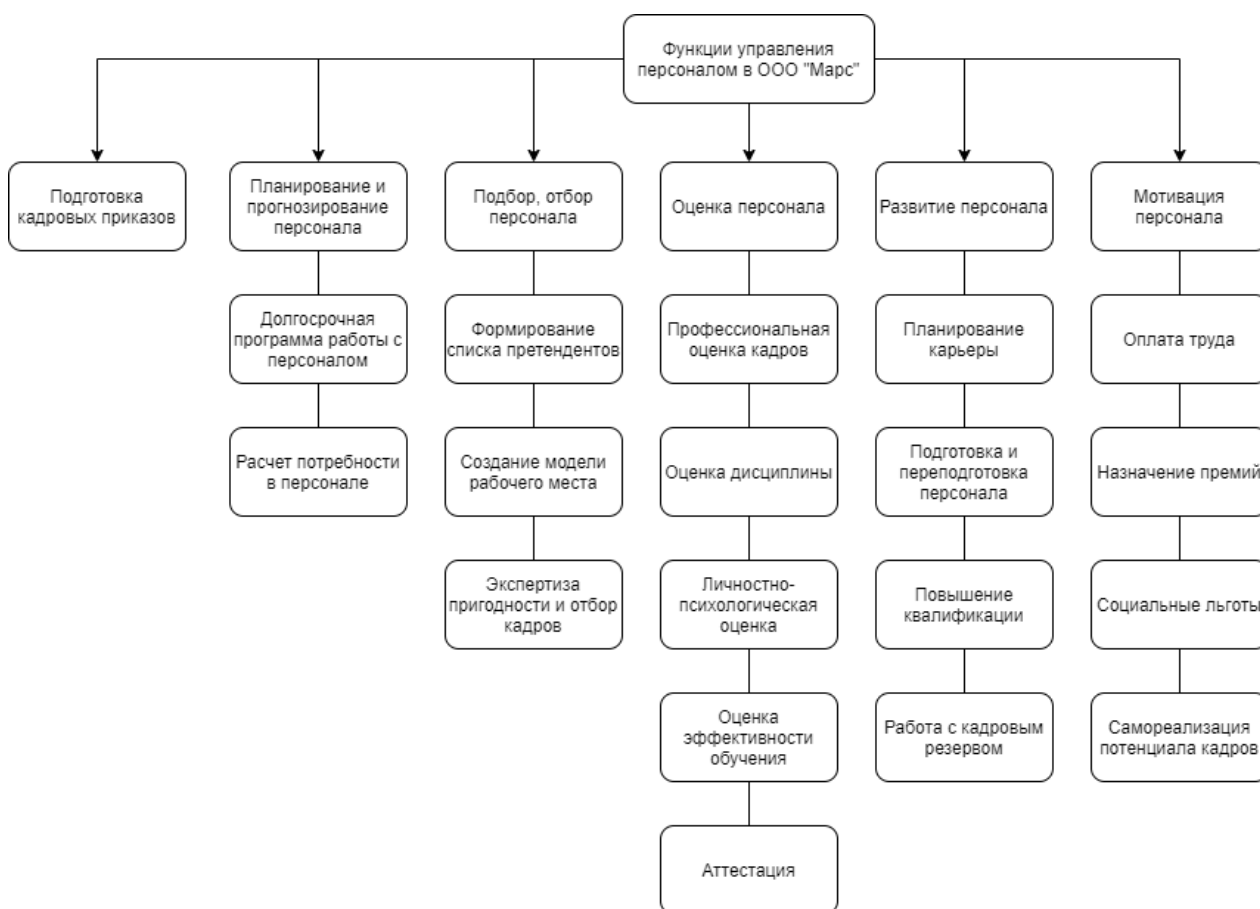


Рисунок 3 – Функции управления персоналом в ООО «Марс»

Далее обратимся к рассмотрению технологии адаптации вновь принятых работников рассматриваемой организации. В ООО «Марс» могут быть реализованы 3 сценария, которые позволят включить в работу только прибывшего работника:

- работник уже имел подобный опыт при работе в иной компании;

- сотрудник был переведен с иной должности, но внутри этой же компании, либо перевод выполняется в иное подразделение в структуре компании.
- на должность был принят специалист, не имеющий необходимого опыта, процесс адаптации может достигать полугода и проходит параллельно со сроком стажировки;

Еще одной технологией управления персоналом, используемой в ООО «Марс», выступает развитие персонала. Важно отметить, что основой развития персонала является организация процесса обучения, повышения квалификации, а также управление карьерой и профессиональным ростом.

Так, в связи с непрерывным развитием науки и техники, даже самым квалифицированным сотрудникам необходимо время от времени повышать уровень своего профессионализма, в связи с чем обязанности данного направления выполняет кадровая служба ООО «Марс».

В ООО «Марс» действуют следующие стимуляционные способы рабочего процесса трудового коллектива.

Для косвенного стимулирования в компании разработан системный подход.

Нематериальное стимулирование работников компании ООО «Марс» достаточно узко используется.

## **1.2 Исследование бизнес-процессов предприятия**

### **1.2.1 Выбор методологии исследования бизнес-процессов**

Под бизнес-процессом принято понимать логичный, последовательный, а также связанный друг с другом комплекс мероприятий, которые используют ресурсы, формируют ценность и выдают результаты. Согласно международному стандарту ISO 9000:2000 было принято такое понятие как «процесс», однако сегодня подобные понятия, можно сказать, обозначаются в качестве синонимов. Построение бизнес-процессов служит в качестве действенного и наиболее продуктивного средства для определения способов в направлении оптимизации деятельности компании, которое позволяет установить, каким образом предприятие работает и каким образом осуществляется деятельность сотрудников на своих рабочих местах [4].

Перед тем, как выполнить автоматизацию установленной предметной области требуется решить, какое направление и разновидность построения бизнес-процессов можно использовать. В практической деятельности формирование бизнес-процессов предприятия совершается поэтапно:

- этап структурного моделирования;
- этап детального моделирования.

Структуризация бизнес-процессов компании, как правило, производится посредством нотации IDEF0 либо на языке UML.

После того, как проанализирована предметная область, была разработана многофункциональная модель основных бизнес-процессов. Процессы моделирования были исполнены на основе нотации IDEF0.

Нотация IDEF0 представляет собой метод моделирования, который позволяет разработать многофункциональную модель, показывающую в полной мере структуру и функциональные возможности системы, в том числе информационные потоки и потоки материальных объектов, что объединяет эти функции. Бизнес-процессы в нотации IDEF0 выглядят как

прямоугольники, а стрелки демонстрируют взаимосвязь с прочими процессами и окружающими факторами.

IDEF0 дает возможность выстраивать многофункциональные модели, описывающие бизнес-процессы в форме иерархической системы взаимодействующих функциональных возможностей. Любая IDEF0-диаграмма содержит блоки и дуги. Блоки изображают функции выстраиваемой ИС. Дуги объединяют блоки и показывают, каким образом происходит взаимодействующие и взаимосвязывающие процессы между ними.

Нотация IDEF0 применяется для того, чтобы построить верхний уровень модели бизнес-процессов. Формирование IDEF0-диаграммы верхнего уровня отвечает за обеспечение более целостного или абстрактное описание объекта моделирования. Перед тем как утвердить построенную модель обязательно нужно провести согласование с архитекторами и IT-разработчиками для того, чтобы подтвердить, что структура бизнес-процессов наглядна, достаточно интуитивно-понятная и доступная. Модели в обязательном порядке необходимо согласовывать с ответственными специалистами организации, обладающими полной информацией по конкретному бизнес-процессу.

### **1.2.2 Анализ бизнес-процессов предприятия**

Для оценки бизнес-процессов компании использовались возможности программы AllFusion Process Modeler (BPwin), которая позволяет создавать функциональные модели в методологии IDEF0 и предназначен для функционального моделирования и анализа деятельности предприятия. Модель в AllFusion Process Modeler представляется как объединение нескольких диаграмм, каждая из которых характеризует отдельный процесс в виде разграничения его на шаги и подпроцессы. Через соединяющиеся дуги происходит описание объектов, данных и ресурсов, требующихся для того, чтобы исполнить соответствующие функции.

Функциональная модель бизнес-процессов функционирования отдела кадров ООО «Марс» представлена и составлена в виде контекстной диаграммы и ее декомпозиции с помощью методологии IDEF0, приведены на рисунках 4 – 5. По результатам функционального моделирования будет разработан функционал информационной системы отдела кадров ООО «Марс».



Рисунок 4 – Контекстная диаграмма функционирования отдела кадров

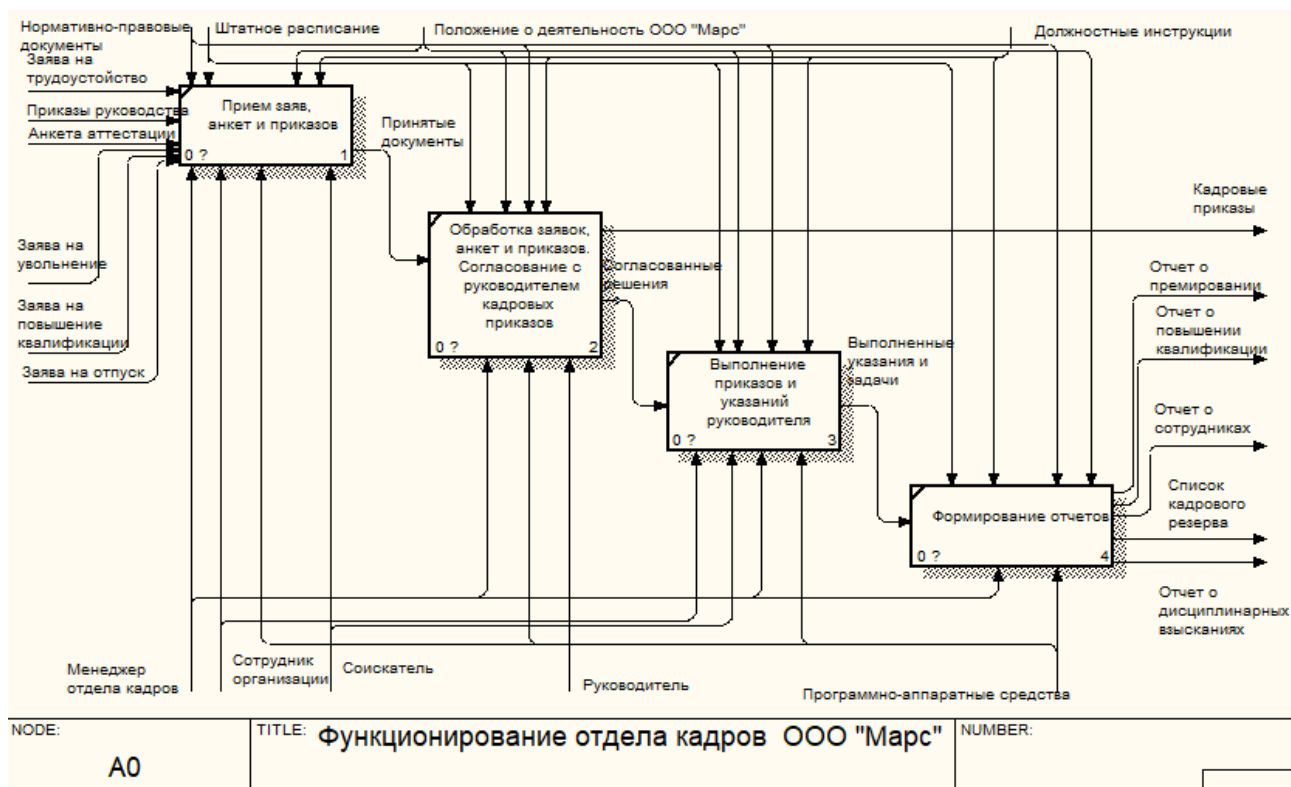


Рисунок 5 – Декомпозиция контекстной диаграммы

При декомпозиции контекстной диаграммы основной процесс разделен на 4 подпроцесса, которые включают: прием заявок, анкет и приказов; обработка заявок, анкет и приказов. Согласование с руководством кадровых приказов; выполнение приказов и указаний руководства; формирование отчетов.

### 1.3 Сравнительный анализ систем для генерации форм документов

На рынке программного обеспечения существует множество программных разработок, которые включают функционал генерации документов, в том числе и кадровых приказов и других видов документов. Рассмотрим наиболее популярных программные решения.

Генератор документов GenDOC FT v1.5. 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. Пользователь только вводит данные, нажимает кнопку и получает набор документов, сохраненных в нужном месте. Окно приложения GenDOC FT v1.5 показано на рисунке 6.

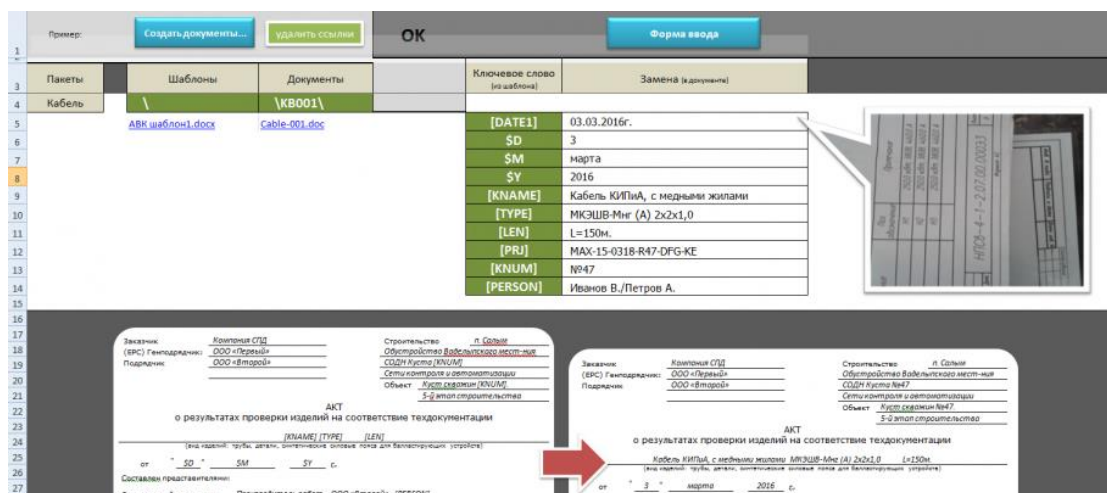


Рисунок 6 – Рабочее окно программы GenDOC FT

Программа имеет ряд преимуществ перед другими аналогичными программами:

- легко настраиваемый интерфейс;
- доступны все возможности Excel;
- не требуется установка новых программ;
- не требуется знание новых программ;

Сайт rbr.ru. Документы и бланки, представленные на сайте:

- кадровые документы;
- документы для лиц, не являющихся гражданами РФ;
- документы по охране труда (приказы, положения, протоколы);
- документы по пожарной безопасности;
- бухгалтерские документы.

Некоторые вышеописанные документы могут быть заполнены в онлайн-режиме, сохранены и распечатаны, остальные документы могут быть

рассмотрены и скачаны их примеры заполнения, скачаны бланки унифицированных и неунифицированных документов в любом удобном формате.

Для каждого документа характерно наличие справки, содержащей в себе сведения о том, как его правильно применять.

Пример генератора из сайта rbr.ru можно посмотреть на рисунке 7.

**Заполнить форму:**

Настоящее уведомление предоставляется в:

(1) Работодатель

Выберите из выпадающего списка статус работодателя или заказчика работ (услуг):

юридическое лицо

Основной вид экономической деятельности в соответствии с Общероссийским классификатором видов экономической деятельности (ОКВЭД)

Сведения о работодателе или заказчике работ (услуг):

(полное наименование юридического лица/филиала иностранного юридического лица/представительства иностранного юридического лица, фамилия, имя, отчество(при наличии) индивидуального предпринимателя/адвоката, учредившего адвокатский кабинет/(частного нотариуса/физического лица - гражданина Российской Федерации)

Номер свидетельства о государственной регистрации юридического лица.

Номер свидетельства о постановке на учет в налоговом органе (при наличии), ИНН (при наличии), КПП (при наличии)

Место нахождения (для физ. лица – адрес фактического места жительства) работодателя или заказчика работ (услуг)

(почтовый индекс, субъект Российской Федерации, район, город, населенный пункт, улица, дом, квартира/офис)

Контактный телефон:

(2) Сведения об иностранном гражданине:

Фамилия:

Имя:

Отчество (при наличии):

Гражданство:

Рисунок 7 – Форма ввода данных для генерации договора

Докс.Онлайн – инструмент для генерации документов по шаблону, окно программы показано на рисунке 8.

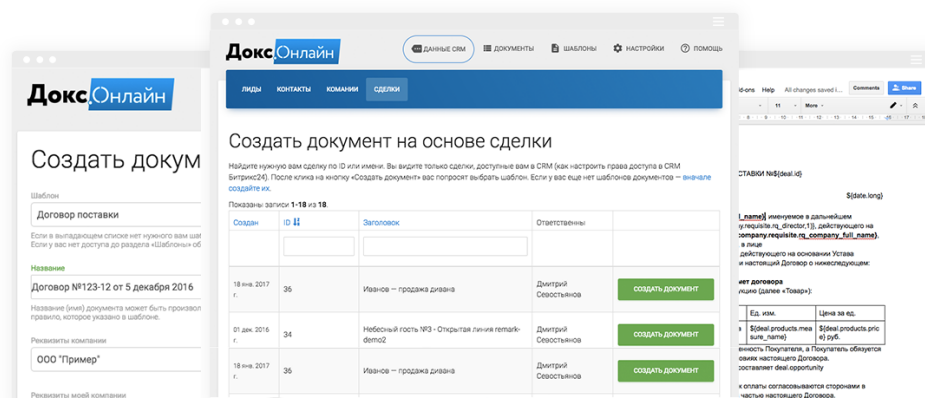


Рисунок 8 – Генератор документов Докс.Онлайн

### Функционал:

- возможность готовить персональные коммерческие предложения на основании шаблонов;
- неограниченной 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, отрисовывать и 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. Страница загрузки решения показана на рисунке 9.

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, форматируйте данные и перенумерованные поля.

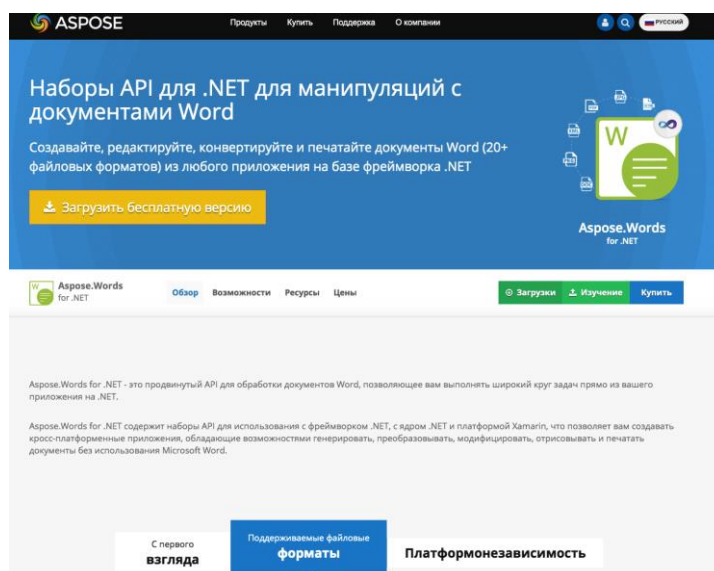


Рисунок 9 – Страница Aspose.Words for .NET

- 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 кадровой службы организации.

#### **1.4 Обоснование структуры и функционала информационной системы для отдела кадров**

Под информационной системой управления рассматривается комплекс математических и экономических методик, информации, программно-технических и технологических средств, также специалистов соответствующей квалификации, где главной целью выступает своевременная обработка информационных данных и принятие решений по вопросам управления.

Основная задача информационной системы заключается в определении причинно-следственных взаимосвязей процессами взаимодействия с результатами. 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.

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

В результате, описывая особые аспекты функционала ИС для управления персоналом, можно выделить ряд конкретных групп функций информационных систем, которые показаны на рисунке 10:

- группа функций ИС «Учет сведений о персонале»;



Рисунок 10 – Основные группы функций информационной системы кадровой службы

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

Автоматизированная система управления кадрами должна быть направлена в сторону оптимизирования руководящей работы и деятельности управления кадров, а также приводить к росту их производительности. Процессы автоматизации хранения и обработки данных сотрудников увеличивает эффективность перевода и подбора персонала.

Таким образом, в результате изучения и анализа предметной области были выделены задачи, которые будут автоматизированы посредством разработки системы автоматизации управления персоналом в ООО «Марс».

Задачи автоматизации:

- учет рабочего времени,

- формирование отчетности;
- учет кадровой информации;
- регулирование оборота документов, которые относятся к личному составу;
- регулирующая деятельность по отношению к процедурам поддержания квалификации сотрудников;
- деятельность по управлению кадровым подбором 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, представлены ниже:

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

На рисунке 11 представлена структура информационной системы.



Рисунок 11 – Структура информационной системы

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

Вывод по первой главе. Рассмотрено понятие и виды кадровых приказов. Организационная структура компании ООО «Марс». ООО «Марс» является юридическим лицом и строит свою деятельность на основании действующего Устава и действующего законодательства Российской Федерации.

Далее рассмотрено управление кадрами в организации, проанализированы основные задачи и функции отдела кадров, а также



бизнес-процессы в ООО «Марс». Представлена функциональная модель бизнес-процессов функционирования отдела кадров ООО «Марс». Проведен сравнительный анализ систем для генерации форм документов. Создана структура и функционал информационной системы для отдела кадров.

## Глава 2. Логическое моделирование предметной области

### 2.1. Логическая модель и ее описание

Под логическим проектированием рассматривается деятельность по трансформированию требований к информации в структурном информационном потоке. Результат – база данных с СУБД–ориентированной структурой, а также спецификационные документы по программам прикладного назначения.

Уровнем логики (программное представление) являются элементы данных, записи, а также реализуемые связи между записями.

Основой многих подходов нашего времени для проектной работе с БД является ER–модели. Область предметного характера моделируется с помощью графических диаграмм. Там содержится некоторое количество разнородных компонентов.

Атрибут, связь и сущность – главные термины ER–модели. У сущности имеются атрибуты – уточняющая деталь по отношению к классификации, идентификации, выражению состояния, числовой характеристики. Записаны атрибуты маленькими буквами в прямоугольниках сущность, под ее именем. Связью является ассоциация, которая реализуется посредством графики. Устанавливается между сущностями. Для каждой из них характерно наличие двух концов, причем оба должны содержать имя, степень конца связи, а также ее обязательность. Наконец, сущностью является объект (представляемый, либо реальный), для которого должна быть доступна и сохранена информация. Для ER–модели ее представляют прямоугольником. Все экземпляры сущности должны быть отличными от иных, относящихся к той же сущности.

Для подтверждения индивидуальность сущности могут быть использованы: атрибуты, их комбинации, комбинации атрибутов и связей,

отдельно связей. Главный параметр — наличие уникальности по отношению к любому экземпляру сущности.

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

Каждая сущность в модели изображается в виде прямоугольника с наименованием (по общепринятому соглашению об именовании сущностей имя сущности должно быть в единственном числе). Атрибуты на диаграммах изображаются в виде овалов, соединенных линиями со своими сущностями. Ключевые атрибуты выделяются на диаграмме подчеркиванием. Связи между сущностями обозначают линиями, которые соединяют прямоугольники соответствующих сущностей. Для сущности, находящейся со стороны «многие», линия связи может заканчиваться значком из трех расходящихся линий.

Под логическим проектированием понимается процесс трансформации информационных требований, заключенных в информационной структуре. Итог – 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» (оно же характеризуется как представление программиста) 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 [5].

Атрибут, связь и сущность – главные термины ER–модели. Сущностью является объект (представляемый, либо реальный), для которого должна

быть доступна и сохранена информация. Для ER–модели ее представляют прямоугольником.

Наиважнейшим показателем сущности выступает также атрибут, объединение атрибутов или связей, либо сочетание связей и атрибутов, что в результате могут отличать любой экземпляр сущности от иных экземпляров аналогичной сущности. На рисунке 12 представлена 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;

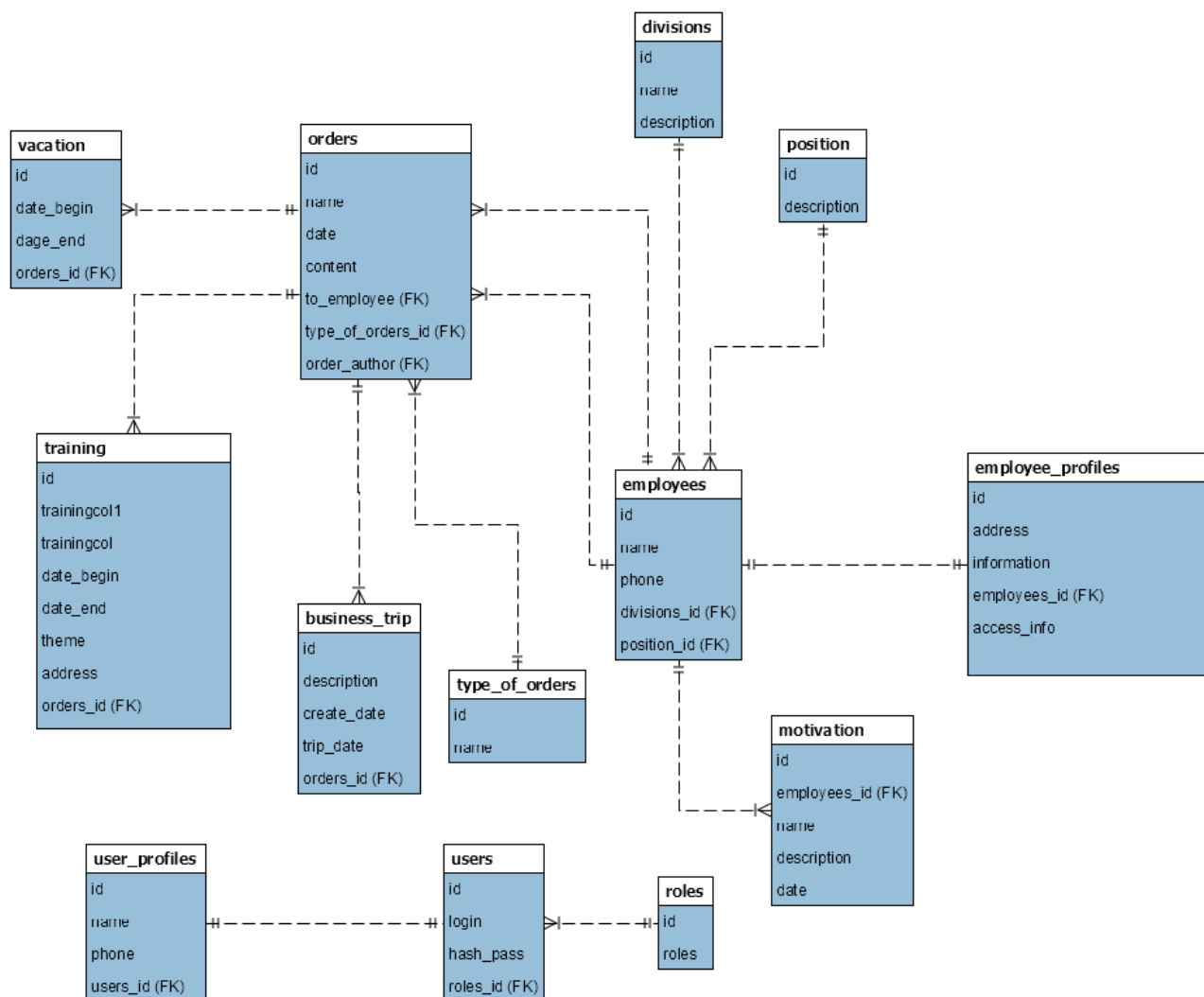


Рисунок 12 – Логическая модель данных

В результате логического моделирования предметной области была разработана логическая модель базы данных, которая трансформирует требования к информации, которая существенна в контексте предметной области и отображает объекты предметной области в виде сущностей логической модели.

## 2.2. Используемые классификаторы и системы кодирования

Эффективному поиску, обработке и передаче информации экономического характера предшествует процесс ее адаптирования. Необходимо выразить ее в цифровом виде, предварительно классифицировав и закодировав. Чаще всего данным процессам подвергаются справочные реквизиты, относящиеся к признакам. Они производят по отношению к отраженным в показателе процессам, месту, объектам и субъектам деятельности, временным затратам на процесс описательные действия. Кодирование относится к признакам, по которым в дальнейшем будет реализована группировка данных. Для рассматриваемой системы предварительно был произведен классификатор локального характера (классификация иерархическим методом).

Далее приведена структура, которая используется для создания БД, таблица 2.

Таблица 2 – Структура нормативно-справочной информации

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Вид классификатора
Код отпуска	XXXXXX	порядковая	локальный
Код кадрового приказа	XXXXXX	порядковая	локальный
Код отдела	XXXXXX	порядковая	локальный
Код должности	XXXXXX	порядковая	локальный
Код сотрудника	XXXXXX	порядковая	локальный
Код командировки	XXXXXX	порядковая	локальный
Код типа заказа	XXXXXX	порядковая	локальный
Код профиля пользователя	XXXXXX	порядковая	локальный
Код профиля сотрудника	XXXXXX	порядковая	локальный
Код пользователя	XXXXXX	порядковая	локальный
Код роли	XXXXXX	порядковая	локальный
Код профиля пользователя	XXXXXX	порядковая	локальный

Системные кодирования и классификации, и их описание. Применение одинакового кода относится ко всем кодам. Их длина формируется исходя из

максимально возможного количества объектов, которые относятся к одному типу.

### **2.3. Характеристика нормативно-справочной и входной оперативной информации**

Описание нормативно-справочной информации представлено в таблице 3.

Таблица 3 – Перечень нормативно-справочных документов

<b>Название документа</b>	<b>Содержание</b>	<b>Периодичность</b>
Список сотрудники	Сотрудники	Перед внедрением
Список кадровых приказов	Кадровые приказы	Перед внедрением
Список тренингов	Перечень тренгов	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 отпуском	Отпуска сотрудников	Перед внедрением

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

### **2.4. Характеристика базы данных**

Для реализации проекта выбрана реляционная модель базы данных. Реляционная БД опирается на реляционную модель. Само происхождение термина «реляционный» возникло от слова «relation» (с англ. «отношение»). Для того, чтобы работать с реляционными базами данных, как правило, применяют реляционные системы управления БД. Первым, кто стал инициатором применения таких БД еще в 1979 г., стал профессор Кодд, который работал в корпорации IBM [5].

Реляционная модель организует информацию в представлении в виде таблиц двумерного характера, которые в свою очередь, ни что иное как массив двумерного характера, должна иметь следующие свойства [4]:

- каждый табличный элемент служит исключительно одним компонентом данных;
- каждая ячейка в столбце таблицы однородная – каждый компонент столбца обязан быть одинакового типа (INT, FLOAT, VARCHAR и т. д.);
- каждый столбец обладает уникальным именем, идентифицирующее данный столбец;
- в таблице не может быть одинаковых строк;
- порядок следования строк и столбцов произволен.

Реляционные системы управления базами данных бывают системами, имеющих открытый код (OpenSource) и имеющих закрытый. В соответствии с исследованиями фирмы IDC, предприятия из Европы чаще предпочитают открытое программное обеспечение. Идут на это они ввиду увеличения качества и гибкости этого программного обеспечения [6].

Система управления базами данных является программным обеспечением, при помощи которого пользователи имеют возможность создания, определения и поддержания базы данных. Специфика СУБД с открытым кодом состоит в том, что они обладают при себе исходным кодом, который может быть изменен, оптимизирован, ускорен за счет того или иного алгоритма.

Выбор СУБД с целью выполнения определенной задачи является достаточно непростым мероприятием, требующим определенных навыков, знаний и внимания профессионала в этой области. Неправильный выбор СУБД может заблокировать процесс реализации данной технологии, а замена СУБД может стоить слишком дорого.

СУБД относится к очень сложной части программного обеспечения, которую трудно понять во всей полноте и является жизненно важной для



определения критериев, которые могут быть использованы администратором базы данных в процессе их отбора. Многие практики–разработчики баз данных в своих работах отмечают, что выбор критериев использования СУБД заключается в первичном анализе организационных потребностей и необходимых функций СУБД. Практики–разработчики выделяют ряд факторов выбора СУБД для реализации задачи, в частности [15]:

- требования к применению (это ограничение, которые вводятся в базу данных в приложении);
- функции и инструменты СУБД (встроенный набор инструментов, который помогает облегчить задачу разработки приложений);
- модели СУБД (иерархические, сетевые, реляционные, объектно–ориентированные);
- переносимость СУБД (системы и языки программирования, платформы)
- требования СУБД к оборудованию (минимальная скорость процессора, объем оперативной памяти, дискового пространства и т. д.);
- стоимость СУБД (затраты на приобретение и обслуживание, эксплуатационные расходы, стоимость лицензий, затраты на установку, стоимость обучения и преобразования расходов);
- сохранение целостности данных (защита данных в базе данных от несанкционированного доступа, изменения или уничтожения данных).

Практически все системы управления реляционными базами данных поддерживают функционирование с языком SQL, посредством которого можно определить и модифицировать структуру сведений, добавление, редактирование и удаление данных, в том числе совершать различные выборки информации.

Больше всего распространились серверы реляционных БД, которые базируются на клиент-серверной архитектуре. Данными серверами

обеспечивается устойчивая работа с базами данных сразу множества клиентов (ими могут являться десятки, сотни, а также тысячи и млн. клиентов – все зависимо от применяемого оборудования и ПО). Помимо того, реляционная модель данных реализуется так называемыми настольными базами данных, к примеру, dBASE, FoxPro, а также Clarion и Paradox, Access. Почти каждая ведущая настольная БД на данный момент поддерживает возможность работать как клиенты серверов БД с помощью технологий ODBC, а также BDE, ADO и других.

Реляционные БД основаны на строгой теории реляционных БД, основывающейся также на теории множеств, а также теории отношений. Реляционная БД является набором таблиц, между которыми имеются заданные связи. Строки таблиц носят название записей, а элементы, которые включены в запись — поля. В теории реляционных БД таблицы носят название отношений, записи — кортежей, а поля — атрибутов.

В таблице реляционной БД не могут содержаться повторяющиеся записи (строки). Данное требование следует из теории множеств. Наименьший набор полей, дающий возможность отличия записи от любой иной записи — ключ. Все значения ключа в рамках таблицы должны являться уникальными. Каждая таблица должна обладать хотя бы одним ключом, что прямо вытекает из того, что в таблице не могут содержаться повторяющиеся записи. Ключи таблицы могут включать одно поле – эти ключи носят название атомарных или простых ключей. Ключи могут включать несколько полей, то есть составные ключи. Таблица БД может обладать одним ключом или несколькими ключами. Один из ключей назначают как первичный ключ, а остальные — потенциальные (в теории реляционных БД) либо альтернативные (в определенных реализациях отдельных БД) [5].

Нередко применяют суррогатный первичный ключ – ключ, включающий поле (или поля), не несущие сведения из предметной сферы, а выступают как замена (суррогат) для естественных (натуральных) первичных

ключей. Как суррогатные ключи зачастую применяются счетчики (генераторы, последовательности) `autoincrement` либо глобально–уникальные идентификаторы (GUID). В правильности созданная структура БД должна отвечать специальным правилам, основанным на теории отношений и называемым как нормальные формы [4].

## **2.5. 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, которые показаны на рисунке 18:

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

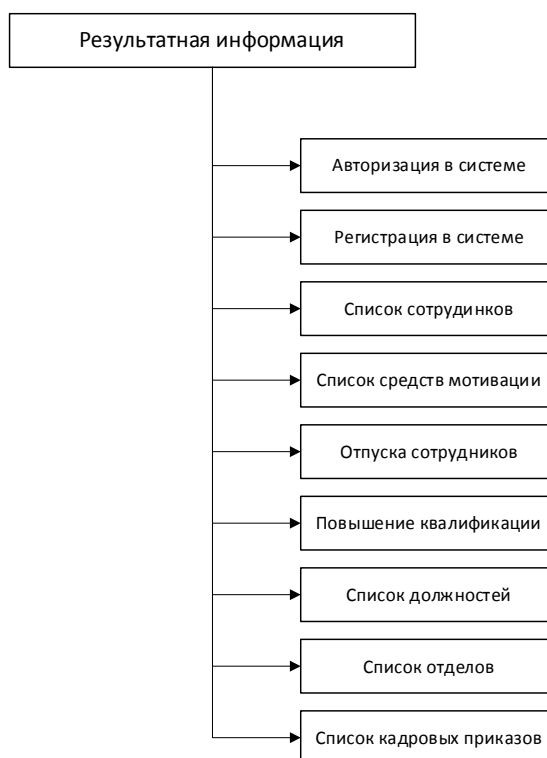


Рисунок 18 – Результатная информация

**Вывод по второй главе:** в результате выполнения второй главы разработаны алгоритмы и функции информационной системы формирования кадровых приказов для ООО «Марс». После этого была создана база данных приложения, которая включает таблицы: `business_trip`, `divisions`, `employee_profiles`, `employees`, `motivation`, `orders`, `position`, `roles`, `training`, `type_of_orders`, `user_profiles`, `users`, `vacation`. Спроектированные таблицы в базе данных обеспечивают реализацию функционала информационной системы формирования кадровых приказов для ООО «Марс».

## **Глава 3. Физическое проектирование информационной системы**

### **3.1. Выбор системы управления базами данных информационной системы**

Сегодня СУБД MySQL (Structured Query Language) можно назвать, наиболее известной и чаще всего используемой СУБД. Благодаря своему функционалу СУБД MySQL может решать множество сложных задач. Кроме того, СУБД MySQL наделена возможностью, которая способствует реализации высоко-функциональных систем.

К преимуществам СУБД MySQL относятся:

- упрощение ввода записей;
- упрощение поиска записей;
- гибкость поиска записей;
- гибкость вывода записей;
- доступ к записям многопользовательский;
- возможность передачи записей в электронном виде.

MySQL является высокопроизводительной многопоточной и многопользовательской системой управления реляционными БД, которая основана на клиент-серверной архитектуре. За последнее время данная мощная, надежная, а также дружелюбно настроенная к пользователю система управления БД выступила как стандарт для применения и в деловой, и в личной области по большому счету из-за ее продвинутого набора инструментов для того, чтобы управлять данными, нестрогой политике лицензирования, а также общемировой поддержке от сообщества, к которому относятся и простые пользователи, и серьезные создатели.

Популяризация MySQL в первую очередь связана с разнообразием особенных возможностей, таких как: скорость, обширность распространения, надежность, качество и доступность исходного кода [15].

Такое данное как время характерно влияет на процесс управления базами данных. Также к нему относится и скорость. Взаимодополнение этих двух составляющих позволяют оценить потраченное время на обработку конкретного запроса и получение данных по нему отправителю. По своей природе система MySQL –производительная система. Ее главным преимуществом является то, что в отличии от своих конкурентов она намного быстрее. Принимая во внимание проведенные тесты на производительность, которые есть в открытом доступе на сайте MySQL, можно сделать вывод о том, что рассматриваемая нами система на шаг впереди всех других конкурентных систем по управлению базой данных, даже включая аналоги, по типу Microsoft SQL Server 2000 и IBM DB2.

Зачастую, чтобы достичь высокой производительности, пренебрегают надежностью системы. Система MySQL характеризуется высокой надежностью и доступным временем работы, все данные прошли тестирование и сертификацию на использования особенно важных, высокоуровневых приложениях. Транзакции выступают гарантом в утвержденной согласованности данных и минимальном риске потери информации, с осуществлением методик, призванных минимизировать время задержки в случае сбоя сервера. К таким методам относятся репликация и кластеризация. Большую роль в обнаружении и устранении ошибок оказывают пользователи MySQL. Благодаря всему вышеперечисленному можно уверенно сказать, что в MySQL фактические ошибки отсутствуют [5].

Важной особенностью MySQL является то, что нормальный уровень производительности поддерживается даже при работе с объемными и сложными базами данных. В качестве базы данных эта система использует таблицы, в которых содержатся сотни тысяч записей. Объем таких таблиц измеряется в нескольких гигабайтах и это является вполне привычным явлением в информационных системах.

Чтобы пользоваться MySQL не требуется определенный уровень подготовки. Достаточно знать основы, на изучение которых потребуется пару

часов, чтобы новичок стал пользователем данной программы. Все необходимые материалы по изучению имеются в открытом доступе и абсолютно бесплатны. Основная работа проходит через интерфейс командной строки, но стоит отметить, что есть ряд графических инструментов, которые облегчат пользование и работу с базами данных MySQL.

Так же преимуществом среди других конкурентов является тот факт, что система MySQL довольно проста в настройке и оптимизации даже при работе с довольно требовательными приложениями. Это является отличительной чертой системы от ее платных аналогов с настройкой сотни различных параметров. Так же есть возможность обучения на данной системе, консультирование и помощь в техническом плане.

Особо важные требования стандарта ANSI SQL (ANSI — American National Standards Institute (Американский национальный институт стандартов)) поддерживаются данной системой, в свою очередь это содействует в использовании пользовательскими расширениями и типами данных, для улучшения работы и повышения функциональности и защиты потребителей.

Система MySQL поддерживается различными операционными системами, к ним относятся: UNIX, Linux, Solaris, FreeBSD, OS/2, MacOS, Windows 95, 98, Me, 2000, XP, NT и Vista, а также Intel x86, Alpha, SPARC, PowerPC и IA64.

Так же стоит отметить, что система MySQL - многопользовательская система. Это определяется тем, что сразу несколько клиентов имеют доступ на использование одной и более баз данных MySQL. Этот пункт является категорично важным при разработке веб-приложений, ведь в этом заключается возможность одновременного подключения удаленных пользователей. Так же система обеспечивают свою надежность, позволяя администраторам через систему привилегий защищать критические данные.

Для таких целей используются определенные комбинации схем проверок достоверности клиентов и хостов [4].

Система является многомиллионным программным решением, именно поэтому используется кодировка Unicode и определенные наборы латинских и китайский символов, необходимых для упорядочивания и сохранения данных.

Система MySQL включает интерфейсы программирования на других языках приложений API (Application Programming Interfaces). Это позволяет использовать и управлять базами данных на множестве языков программирования. В настоящий момент система имеет возможность работать со следующими языками: C, C++, Eiffel, Java, Perl, PHP, Python, Ruby и Tcl.

Еще одной важной особенностью данной системы является то, что ее исходный код находится в открытом доступе, что в свою очередь позволяет пользователям использовать и изменять приложение в соответствии со своими требованиями. Такое преимущество послужило популяризацией MySQL и создания активного сообщества разработчиков и пользователей данной системы.

Все вышеперечисленные преимущества сыграли роль в лидерстве системы на фоне других конкурирующих систем. Все пользователи имеют возможность протестировать систему на надежность. Также стоит отметить, что движок MySQL постоянно модернизируется, чтобы соответствовать последним технологическим требованиям.

После изучения особенностей СУБД MySQL, которая выбрана для создания информационной системы, необходимо спроектировать базу данных. Процесс проектирования базы данных состоит из трех этапов: концептуальное проектирование; логическое проектирование; физическое проектирование.



### 3.2. Разработка физической модели данных информационной системы

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 13.

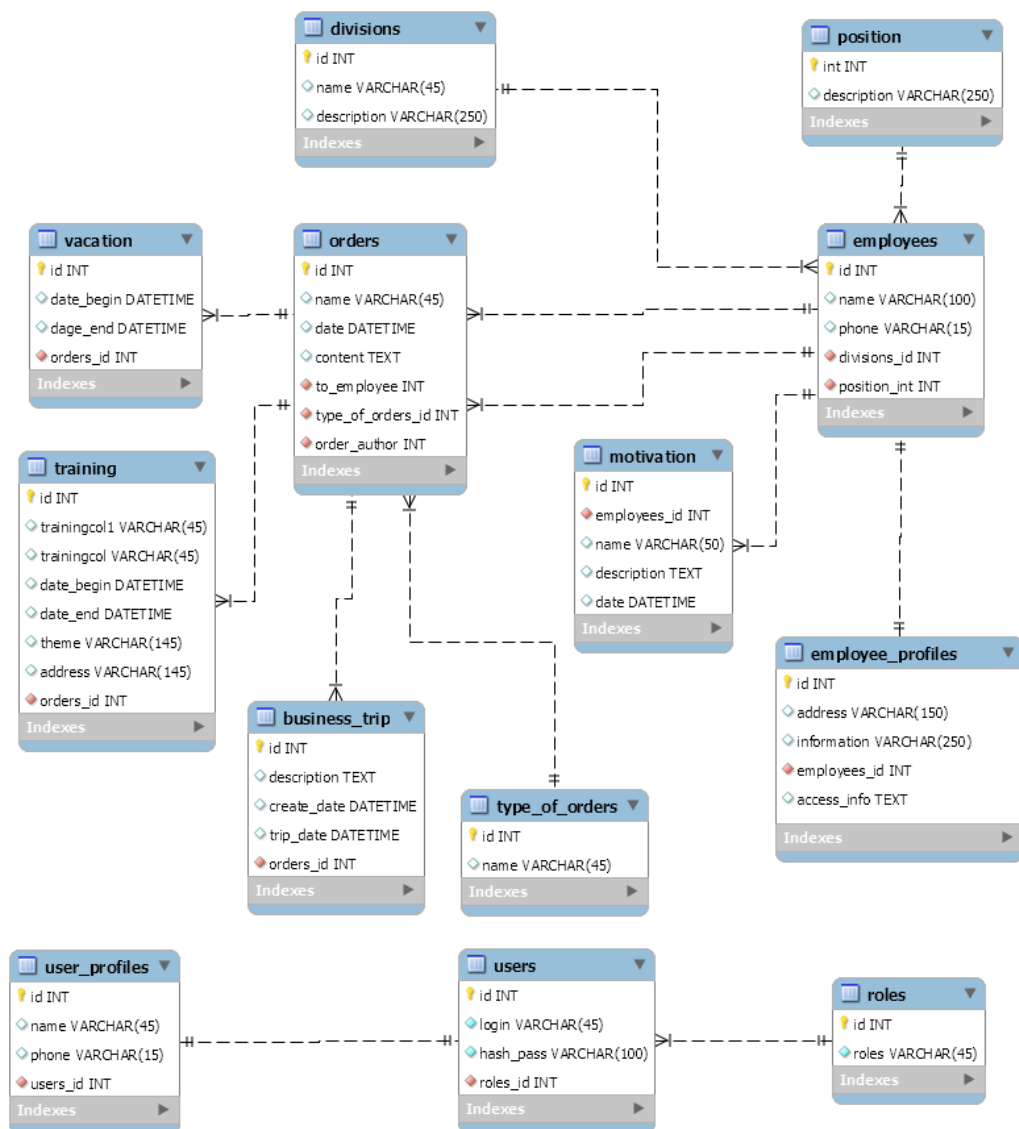


Рисунок 13 – Физическая модель данных

Также отметим, что любая физическая модель данных обязана отвечать характеристикам данных в БД конкретной системы управления базами

данных, другими словами, соответствовать непосредственно схеме 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 [5].

SQL-скрипт создания базы данных для проекта в СУБД MySQL представлен в приложении 3.

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 14.

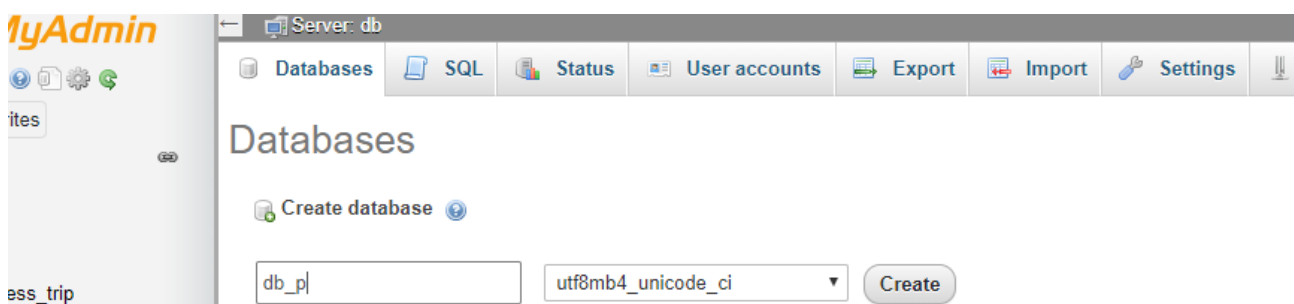


Рисунок 14 – Создание базы данных

Окно ввода запроса к базе данных показано на рисунке 15

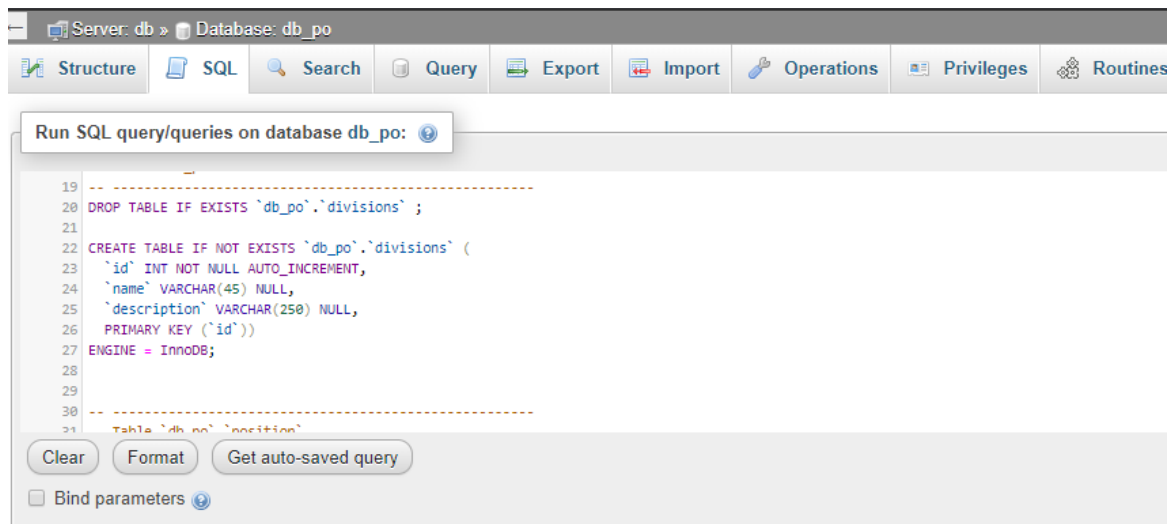


Рисунок 15 – Окно ввода SQL-запроса

После выполнения запроса на сервере MySQL были созданы таблицы для проекта. Структура созданной базы данных показана на рисунке 16.

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

Рисунок 16 – Структура базы данных на сервере MySQL

В таблице «orders», которая показана на рисунке 17, 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 17 – Таблица «orders»

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

### 3.3 Разработка программного обеспечения

При входе в приложение, не авторизованному пользователю не предоставляется никаких возможностей для работы с генератором документов. 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 18) и RegistrationForm.php (рисунок 19).

```

67 <b style="..."><? $msg; ?</b>
68 <form method="post">
69   <div class="form-group">
70     <label for="inputEmail">Адрес электронной почты</label>
71     <input type="email" class="form-control" id="inputEmail" placeholder="Ваш email"
72       name="email"
73       value="<? $form->getEmail(); ?>"
74   </div>
75   <div class="form-group">
76     <label for="InputUsername">Имя пользователя</label>
77     <input type="text" class="form-control" id="InputUsername"
78       placeholder="Ваше Имя" name="login" value="<? $form->getLogin() ?>"
79   </div>
80   <div class="form-group">
81     <label for="InputPassword">Пароль</label>
82     <input type="password" class="form-control" id="InputPassword" placeholder="Пароль"
83       name="password">
84   </div>
85   <div class="form-group">
86     <label for="InputPasswordConfirm">Проверка пароля</label>
87     <input type="password" class="form-control" id="InputPasswordConfirm"
88       placeholder="Проверка пароля"
89       name="passwordConfirm">
90   </div>
91   <button type="submit" class="btn btn-primary">Отправить</button>
92   <a href="index.php" class="btn btn-primary">Отмена</a>

```

Рисунок 18 – Работа с файлом register.php

```

*/
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 19 – Работа с файлом RegistrationForm.php

Модуль авторизации представлен файлами LoginForm.php (рисунок 20), login.php (рисунок 21), logout.php (рисунок 22).

```
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     */
```

Рисунок 20 – Работа с файлом LoginForm.php

```
71 <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">
72     <form method="post">
73         <div class="form-group">
74             <label for="inputEmail">Логин</label>
75             <input type="text" class="form-control" id="inputEmail" placeholder="Ваш логин"
76                 name="login"
77                 value="<?=$form->getLogin(); ?>"
78             />
79         </div>
80         <div class="form-group">
81             <label for="inputPassword">Пароль</label>
82             <input type="password" class="form-control" id="inputPassword" placeholder="Пароль"
83                 name="password">
84         </div>
85         <button type="submit" class="btn btn-primary">Войти</button>
86         <a href="index.php" class="btn btn-primary">Отмена</a>
87     </form>
88 </div>
```

Рисунок 21 – 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 22 – 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 23), который формирует форму ввода данных кадрового приказа и предоставляет возможность его распечатать.

```
23 <form method="post">
24   <div class="form-group">
25     <label for="name">Название</label>
26     <input type="text" class="form-control" id="name" placeholder="Название (номер)"
27       name="name"
28       value="">
29   </div>
30
31   <div class="form-group">
32     <label for="date">Дата</label>
33     <input type="datetime-local" class="form-control" id="date" placeholder="Дата"
34       name="date">
35   </div>
36
37   <div class="form-group">
38     <label for="content">Содержание приказа</label>
39     <input type="text" class="form-control" id="content" placeholder="Содержание приказа"
40       name="content">
41   </div>
42
43   <div class="form-group">
44     <label for="to_employee">Для работника</label>
45     <input type="text" class="form-control" id="to_employee" placeholder="ФИО сотрудника"
46       name="to_employee">
47   </div>
48
49   <div class="form-group">
50     <label for="type_of_orders_id">Тип приказа</label>
51     <input type="text" class="form-control" id="type_of_orders_id" placeholder="Тип приказа"
52       name="type_of_orders_id">
53   </div>
54
55   <div class="form-group">
56     <label for="order_author">Приказ составил</label>
57     <input type="text" class="form-control" id="order_author" placeholder="ФИО сотрудника"
58       name="order_author">
59 </div>
```

Рисунок 23 – Работа с формой создания кадрового приказа

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 24 – 27.

The screenshot shows a web page with a dark header containing the text "PersonnelOrders Главная Регистрация Авторизация Меню". Below the header, the page title is "Регистрация". A navigation bar contains a link "Главная" and the text "Регистрация". The main heading is "Регистрация на сайте". Below this, there are four input fields: "Адрес электронной почты" with the placeholder "Ваш email", "Имя пользователя" with the placeholder "Ваше Имя", "Пароль" with the placeholder "Пароль", and "Проверка пароля" with the placeholder "Проверка пароля". At the bottom of the form are two buttons: "Отправить" and "Отмена".

Рисунок 24 – Внешний вид формы регистрации на сайте

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

На рисунке 25 показан внешний вид формы авторизации в системе.

The screenshot shows a web page with a dark header containing the text "PersonnelOrders Главная Регистрация Авторизация Меню". Below the header, the page title is "Авторизация в системе". A navigation bar contains a link "Главная" and the text "Авторизация в системе". The main heading is "Вход на сайт". Below this, there are two input fields: "Логин" with the placeholder "Ваш логин" and "Пароль" with the placeholder "Пароль". At the bottom of the form are two buttons: "Войти" and "Отмена".

Рисунок 25 – Внешний вид формы авторизации

Форма авторизации включает поля ввода логина и пароля, а также кнопки «Войти» и «Отмена».

Страница добавления кадрового приказа показана на рисунке 26.




## Добавить кадровый приказ

[Главная](#) Добавить кадровый приказ

Название  
Поощрение Иванова А. А.

Дата  
16/04/2020, 09:10

Содержание приказа  
Файл Изменить Вид Формат



Имя организации: \_\_\_\_\_

Номер документа: 45 60      Дата составления: 16/04/2020

**ПРИКАЗ**  
(распоряжение)  
о поощрении работника

Иванов А. А. фамилия, имя, отчество	Табельный номер 879976
Бухгалтерия структурное подразделение	
бухгалтер	

TABLE » TBODY » TR » TD » P » SPAN      ПРИ ПОДДЕРЖКЕ TINY

Для работника  
Иванов А. А.

Тип приказа  
Мотивация

Приказ составил  
Макаров О. О.

**Сохранить** **Отмена**

Рисунок 26 – 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, что показано на рисунке 27.

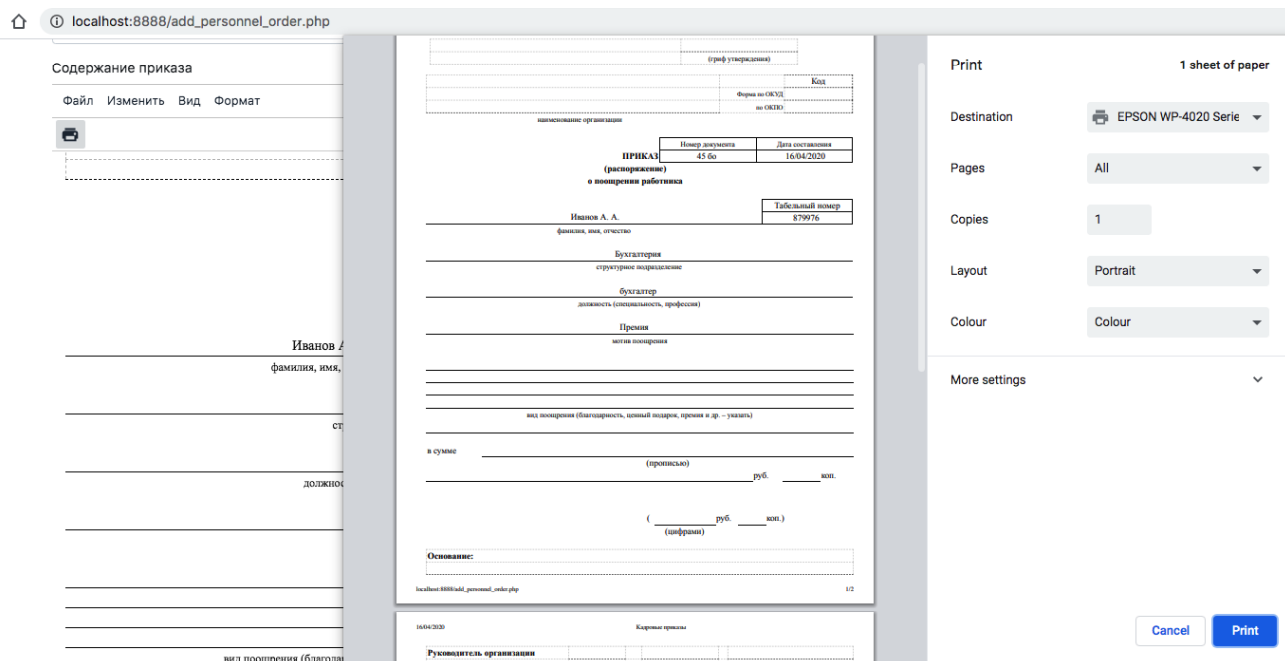


Рисунок 27 – Печать кадрового приказа

Представленная возможность печати кадрового приказа позволяет, как распечатать приказ на принтере, так и сохранить на компьютер в формате .pdf для дальнейшего использования.

Таким образом, за счет реализации внешнего вида был реализован удобный интерфейс взаимодействия пользователя с системой, обеспечена реализация описанного функционала за счет создания необходимых блоков программного кода.

### **3.4. Оценка и обоснование экономической эффективности разработки информационной системы**

#### **3.4.1 Выбор методики расчёта экономической эффективности**

Для оценки эффективности проекта приняты методики оценивания трудовых, временных и финансовых ресурсов, использованных для создания программного средства.

За основу определения требуемого объема ресурсов нужно положить:

- совокупность бизнес-процессов, которые реализуются в будущей программной системе, а также их относительную важность (приоритеты) для заказчика;
- требования к функциональной полноте, а также качеству осуществления каждого бизнес-процесса.

Как ключевые показатели оценки стоимости программной системы применяются показатели:

- сложности (размеров) программной системы;
- трудозатрат на разработку;
- длительности создания программной системы в целом, а также отдельных ее этапов;
- численности и квалификации специалистов, которых привлекают к созданию программной системы;
- фонда оплаты труда специалистов на формирование в целом программной системы, а также по определенному этапу жизненного цикла;
- прочих прямых затрат и накладных расходов, связанных с созданием программной системы.

В данном разделе рассчитывается технико-экономическое обоснование разработки информационной системы автоматизации обработки заказов в компании «Марс». Расходы, связанные с разработкой и внедрением системы

формирования кадровых приказов, а также с эксплуатацией ЭВМ, состоят из следующих затрат:

- Затраты на анализ и внедрение методов автоматизации процессов кадровой службы, в том числе: затраты на программное обеспечение, на эксплуатацию оборудования, затраты на заработную плату во время выполнения работы, затраты на заработную плату руководителю проекта и консультантам, а также затраты на единый социальный налог и накладные расходы.
- Затраты на эксплуатацию информационной системы, в том числе: затраты на обучение (ознакомление) пользователя с данной программой, затраты на эксплуатацию оборудования, затраты на единый социальный налог, затраты на накладные расходы, затраты на годовые амортизационные отчисления

### **3.4.2 Расчёт показателей экономической эффективности проекта**

Задачи руководителя проекта:

- согласовывать и утверждать все стадии проекта;
- передавать информацию и программу для введения в организацию.

Задачи экономиста:

- анализировать предметную область;
- моделировать бизнес-процессы;
- разрабатывать требования к будущей информационной системе;
- формулировать задачу программисту;
- анализировать результаты функционирования ИС.

Задачи программиста:

- разрабатывать алгоритм разрешения задач;
- выбирать среду программирования;
- разрабатывать модули программы;
- программировать объекты;

- осуществлять отладку;
- тестировать;
- устанавливать программу, готовить программную документацию.

Расходы на разработку ИС определяют по нижеследующим основным статьям:

- плата заработная;
- отчисления социальные;
- стоимость машинного времени;
- амортизация;
- расходы накладные;
- расходы на расходный материал.

Совокупные расходы на основную заработную плату составили 227760,00 руб. (на 3 человек).

Показатель общей страховой нагрузки в 2018 г. равен 30% от зарплаты работника, из них:

- 22% подлежат уплате в ПФР;
- 2,9 % в ФСС РФ;
- 5,1% в ФФОМС.

Отчисления во внебюджетные и страховые фонды рассчитывают при помощи формулы 1:

$$S_{\text{отл}} = Z_{\text{осн}} \times S_{\text{тсоц}}, \quad (1)$$

где  $Z_{\text{осн}}$  — общий ФОТ разработчиков;

$S_{\text{тсоц}}$  — общая ставка налоговых отчислений.

$$S_{\text{отл}} = 116000 \times 0,30 = 68328 \text{ руб.}$$

Машинное время равняется 50% от трудоемкости, вместе с тем трудоемкость разработки равна 312 человеко-часам при продолжительности

разработки 39 рабочих дня и 8-часовом рабочем. Показатель средней стоимости 1 машино-часа - 6 рублей.

Определение стоимости машинного времени  $S_{МАШ}$  осуществляется при помощи формулы 2:

$$S_{МАШ} = T_0 \times D_{МАШ} \times C_{МАШ}, \quad (2)$$

где  $D_{МАШ}$  — доля времени на разработку, которая приходится на применение машинного времени;

$C_{МАШ}$  — стоимость 1 часа работы ПЭВМ.

$$S_{МАШ} = 312 \times 0,5 \times 6 = 936 \text{ руб.}$$

Амортизацию рассчитывают при помощи формулы 3:

$$A_{ГОД} = S : T, \quad (3)$$

где  $A_{ГОД}$  — амортизация основных фондов за год;

$S$  — стоимость покупки ПЭВМ наряду с оргтехникой;

$T$  — период эксплуатации.

Стоимость ПК наряду с оргтехникой равна 600000 руб. Сроки полезного использования установлены в пяти лет, в таком случае:

$$A_{ГОД} = 60\ 000 : 5 = 12\ 000 \text{ руб./год.}$$

Ежемесячную сумму амортизации  $A_{МЕС}$  рассчитывают при помощи формулы 4:

$$A_{МЕС} = A_{ГОД} : 12, \quad (4)$$

$$A_{\text{МЕС}} = 12\,000 : 12 = 1000 \text{ руб./мес.}$$

Расчет амортизации в связи с разработкой и введением ИС проводится при помощи формулы 5:

$$A_{\text{РАЗМ}} = A_{\text{МЕС}} \times T_{\text{МЕС}}, \quad (5)$$

где  $T_{\text{МЕС}}$  – кол-во месяцев, требуемое для разработки и введение ИС.

$$A_{\text{РАЗМ}} = 1000 \times 2 = 2000 \text{ руб./мес.}$$

Накладные расходы равны примерно 5% от затрат на основную з/п и расчет их осуществляется при помощи формулы 6:

$$R_n = O_s \times k_{nr} = 227760 \times 0,05 = 11388 \text{ руб.}, \quad (6)$$

где  $O_s$ — зарплата суммарная;

$k_{nr}$ — коэффициент накладных затрат.

В таблице 4 указан список необходимого материала для работы, их объем и цена, а также общие расходы на покупку расходного материала.

Таблица 4 – Расходы на расходный материал при разработке ИС

Название	Количество, штук	Цена, рублей	Стоимость, рублей
Флеш-карты	1	1800	1800
Офисная бумага	1	800	800
Картриджи для принтера	1	1100	1100
Карандаши	3	20	60
Ручки	3	25	75
Стерки	1	10	10
Линейки	1	15	15
Итого:			3860

Общая стоимость создания и введения ИС будет равна 127720 руб.

Для введения разработанной ИС нужно установить ПЭВМ с соответствующим ПО. В организации в данном отделе уже установили требуемое оборудование, поэтому расходы, связанные с его покупкой, в данном случае отсутствуют.

Структура затрат на создание информационной системы для ООО «Марс» показана на рисунке 28.

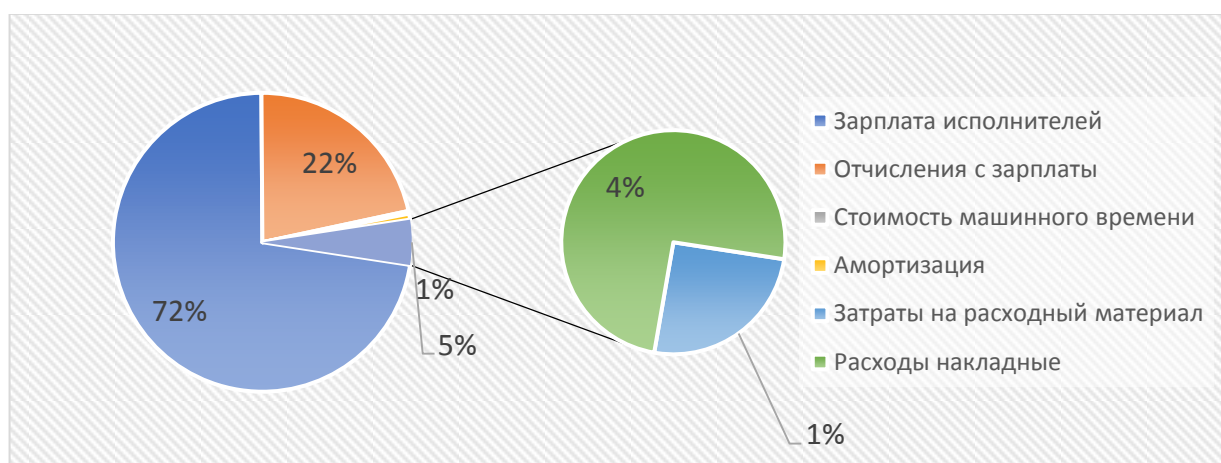


Рисунок 28 – Структура затрат на создание ИС для ООО «Марс»

Сметная калькуляция расходов на разработку и введение ИС указана в таблице 5.

Таблица 5 – Сметная калькуляция расходов на разработку и введение ИС

Показатель	Сумма (руб.)
Зарплата исполнителей	227760
Отчисления с зарплаты	68328
Стоимость машинного времени	936
Амортизация	2000
Затраты на расходный материал	3860
Расходы накладные	11388
Итого	314 272



Целесообразность создания и введения ИС может быть оценена на основании расчетов экономии, которую получили в результате создания и введения ИС силами предприятия по сравнению с альтернативными расходами на приобретение либо заказ системы у сторонних предприятий. Средняя рыночная стоимость подходящих для компании аналогов равна 400 000 рублей.

Сравнение расходов на разработку ИС собственными силами с расходами на покупку уже готового программного продукта проводится при помощи формулы 7 произведен расчет.

$$K_э = Z_{\text{ПР}} \times K_{\text{РМ}} : Z_{\text{РАЗ}}, \quad (7)$$

где  $K_э$  – коэффициент экономии;

$Z_{\text{ПР}}$  – расходы на покупку готового программного продукта;

$K_{\text{РМ}}$  – кол-во рабочих мест;

$Z_{\text{РАЗ}}$  – расходы на создание информационной системы собственными силами.

$$K_э = 400000 \times 1 : 314272 = 1,27$$

Помимо этого, стороннему программному комплексу необходимы дополнительные расходы на введение и сопровождение. К технической поддержке относится обновление ключей, версий программы, консультация по настройке и применению программы.

Расходы на сопровождение  $Z_{\text{СОПР}}$  вычисляют при помощи формулы 8 произведен расчет.

$$Z_{\text{СОПР}} = Z_{\text{СМ}} \times T_{\text{МЕС}}, \quad (8)$$

где  $Z_{\text{СМ}}$  – расходы на сопровождение за 1 мес.;

$T_{\text{МЕС}}$  – кол-во месяцев.

Следовательно,

$$Z_{\text{СОПР}} = 38000 \times 3 = 114\,000 \text{ руб.}$$

Итак, совокупные расходы на создание и сопровождение покупаемого готового продукта  $Z_{\text{СОВ}}$  равны (формула 9):

$$Z_{\text{СОВ}} = (Z_{\text{ПР}} \times K_{\text{РМ}}) + Z_{\text{СОПР}}, \quad (9)$$

$$Z_{\text{СОВ}} = 400\,000 + 114\,000 = 514\,000 \text{ руб.}$$

Соответственно, экономия от создания ИС собственными силами равна (формула 10):

$$\mathcal{E} = Z_{\text{СОВ}} - Z_{\text{РАЗ}}, \quad (10)$$

$$\mathcal{E} = 514\,000 - 314\,272 = 199\,728 \text{ рублей/год.}$$

Вышеуказанный анализ выявил, что целесообразным будет создание информационной системы обработки заявок для ООО «Марс» собственными силами.

## Заключение

В результате выполнения работы была достигнута поставленная цель – разработана веб-ориентированная информационная система формирования кадровых приказов на предприятии (на примере ООО «Марс»).

При достижении цели решены следующие задачи:

- проанализирована деятельность ООО «Марс»;
- изучены бизнес-процессы в организации;
- разработан функционал и структура информационной системы;
- спроектирована и создана информационная система формирования кадровых приказов на предприятии.

При выполнении работы рассмотрено понятие и виды кадровых приказов, изучена организационная структура компании ООО «Марс», которое является юридическим лицом и строит свою деятельность на основании действующего Устава и действующего законодательства Российской Федерации. При рассмотрении задач управления кадрами в организации, проанализированы основные задачи и функции отдела кадров, а также бизнес-процессы в ООО «Марс». Представлена функциональная модель бизнес-процессов функционирования отдела кадров ООО «Марс». Проведен сравнительный анализ систем для генерации форм документов. Создана структура и функционал информационной системы для отдела кадров. Разработаны алгоритмы и функции информационной системы формирования кадровых приказов для ООО «Марс». После этого была создана база данных приложения, которая включает таблицы: `business_trip`, `divisions`, `employee_profiles`, `employees`, `motivation`, `orders`, `position`, `roles`, `training`, `type_of_orders`, `user_profiles`, `users`, `vacation`. Спроектированные таблицы в базе данных обеспечивают реализацию функционала информационной системы формирования кадровых приказов для ООО «Марс». Выбрана СУБД для реализации проекта, определена среда проектирования базы данных, а также выбраны язык программирования для

реализации проекта. В качестве СУБД выбрана MySQL, для моделирования базы данных использованы возможности MySQL Workbench, как язык программирования выбран PHP.

В качестве языка программирования для реализации функционала системы выбран PHP. Заключительным этапом работы над проектом была организация внешнего вида страниц приложения, который реализован с применением CSS-фреймворка Bootstrap 4. Проведен экономический расчет.

## Список используемой литературы и список используемых источников

1. Арсеньев Ю. Н. Управление персоналом. Технологии: учебник / Ю. Н. Арсеньев, С. И. Шелобаев, Т. Ю. Давыдова. - М.: ЮНИТИ-ДАНА, 2012. - 193 с.
2. Бажан И. И. Управление кадровым потенциалом современной организации / Бажан И. И. // Проблемы повышения эффективности инфраструктуры. 2012. - № 33. - С. 52-59.
3. Базаров Т. Ю. Управление персоналом: учебник для вузов / Базаров Т. Ю. - М.: Наука, 2013. - 423 с.
4. Бандурин А. В. Стратегический менеджмент и кадровый потенциал организации: монография / Бандурин А.В. - СПб.: Питер, 2011. – 144 с.
5. Библиотека EasyDox [Электронный ресурс] / Режим доступа: <http://morpher.ru/EasyDox/> (дата обращения 04.02.2020).
6. Вересников Ю. К. О надежности систем обработки информации / Ю. К. Вересников // Актуальные проблемы современной науки. - 2011. - № 2 (58). - С. 193-195.
7. Веснин В. Р. Управление персоналом. Теория и практика: учебник / Веснин В.Р. - М.: Проспект, 2011. - 688 с.
8. Герасимов Б. Н. Менеджмент персонала: учебное пособие / Б. Н. Герасимов. - Ростов-на-Дону: Феникс, 2013. - 488 с.
9. Головкин А. Т. Система менеджмента персонала: учебное пособие / А. Т. Головкин - М.: Инкос, 2014. - 480 с.
10. Давыденко И. Г. Экономический анализ финансово-хозяйственной деятельности предприятия: учебное пособие / И. Г. Давыденко. - М.: КноРус, 2016. – 376 с.
11. Довгань Л. Е. Эффективность управления персоналом предприятий: диагностика и механизм обеспечения: монография / Л. Е. Довгань, Г.А. Мохонько. - Киев: НТУУ «КПИ», 2015. - 231 с.

12. Иваницкая И. И. Управление человеческими ресурсами: учебник / И. И. Иваницкая. - Сыктывкар: СЛИ, 2014. – 578 с.
13. Ильвовский Д. Системы автоматической обработки текстов / Д. Ильвовский, Е. Черняк // Открытые системы. 2014. № 1. - С. 51 – 53.
14. Каменская М. В. Проблемы трудовой мотивации сотрудников / М. В. Каменская // Вестник социально-экономических исследований. - 2012. - №2(45). - С. 286 - 289.
15. Кибанов А. Я. Управление персоналом организации: учебник / А. Я. Кибанов. - М.: ИНФРА-М, 2012. - 512 с.
16. Кожанова Е. Н. Составление актов в общем и кадровом делопроизводстве / Е.Н. Кожанова // Секретарь-референт — 2013. — № 11. — С. 45— 49.
17. Коробкина, М.А. Кадровые технологии в практике управления персоналом организаций / М.А. Коробкина // Вестник Восточно-Сибирской Открытой Академии. — 2014. — № 12 (12). — С. 9.
18. Коцюба И. Ю. Основы проектирования информационных систем. Учебное пособие / И. Ю. Коцюба, А. В. Чунаев, А. Н. Шиков. – СПб: Университет ИТМО, 2015. – 206 с.
19. Кудинов Ю. И. Практикум по основам современной информатики: учебное пособие / Ю. И. Кудинов, Ф. Ф. Пащенко, А. Ю. Келина. - СПб. [и др.]: Лань, 2011. - 350 с.
20. Лесничая И. Г. Информатика и информационные технологии: учебное пособие / И.Г. Лесничая [и др.]. - М.: Эксмо, 2008. -590 с.
21. Литвинюк А. А. Управление персоналом: учебник и практикум / А. А. Литвинюк - М.: Юрайт. - 2016. – 498 с.
22. Методы работы с документами MICROSOFT WORD при разработке веб-сервисов / И. А. Гурин, Н. А. Спирин, В. В. Лавров, Н. И. Декун // Теплотехника и информатика в образовании, науке и производстве : сборник докладов VI Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных (ТИМ'2017) с международным

участием (Екатеринбург, 11–12 мая 2017 г.). — Екатеринбург: УрФУ, 2017. — С. 199-203.

23. Мизинцева М. Ф. Оценка персонала: учебное пособие / М. Ф. Мизинцева. - М.: Юрайт, 2015. – 384 с.

24. Миронов В. В. Информационная технология персонализации электронных документов Microsoft Office в WEB-среде на основе XML / В. В. Миронов, Г.Р. Шакирова, В.Э. Яфаев // Вестник Уфимского государственного авиационного технического университета. 2008. №2. С. 112-122.

25. Михайлова Л. И. Управление персоналом: учеб. пособие / Л. И. Михайлова. - М.: Центр учебной литературы, 2011. - 248 с.

26. Михайлова, А. В. Анализ кадрового и документационного обеспечения в системе управления персоналом организации / А.В. Михайлова, А.А. Казанова // Управление персоналом и интеллектуальными ресурсами в России. — 2015. — № 2. — С. 36-42.

27. Новикова Т. Б. Описание управления бизнес-процессами предприятия на основе методологии ndef0: трудности разработки, рекомендации по совершенствованию построения диаграмм / Т. Б. Новикова, Л. В. Курзаева, В. Е. Петеляк, О. Е. Масленникова, И. Д. Белоусова // Фундаментальные исследования. – 2015. – № 8-2. – С. 318-322

28. Одинокина С. В., Основы технологий XML - СПб: НИУ ИТМО, 2013. – 56 с.

29. Официальный сайт журнала «Кадры предприятия» [Электронный ресурс]. – Режим доступа: <http://karp.ru> (дата обращения 12.02.2020).

30. Официальный сайт журнала «Работа с персоналом» [Электронный ресурс]. – Режим доступа: <http://www.hr-journal.ru/> (дата обращения 20.02.2020).

31. Официальный сайт практического журнала по кадровой работе «Кадровое дело» [Электронный ресурс]. – Режим доступа: <http://www.kdelo.ru/> (дата обращения 12.02.2020).

32. Пустозерова, В.М. Все приказы по кадрам и сопровождающие документы. — М., 2007. — 144 с.
33. Реймаров Г. А. Комплексная оценка персонала. Инженерный подход к управлению качеством труда: учебное пособие / Г. А. Реймаров - М.: ЛКИ, 2015. – 422 с.
34. Рогожин, М. Ю. О профессиональных стандартах кадровиков. — 2016. — № 4. — С. 13—16.
35. Росс В. С. Создание сайтов: HTML, CSS, PHP, MySQL [Текст]. Учебное пособие, ч. 2 / В. С. Росс — МГДД(Ю)Т, М. - 2011 - 68 с
36. Рощупкина Е. В. Кадровое дело. 100 рабочих ситуаций. – М., 2012 – 400 с.
37. Рудаков А. В. Технология разработки программных продуктов: учеб. пособие для студ. / А. В. Рудаков. - 5-е изд., стер. - М.: Издательский центр «Академия», 2010.-208с.
38. Санкина, Л.В. Регистрация кадровых документов // Кадровая служба и управление персоналом предприятия. — 2010. — № 5. — 25 с.
39. Семенихин В. В. Кадровое делопроизводство // «ГроссМедиа», «РОСБУХ», 2014. — 328 с.
40. Скавитин А. А. Управление поощрением персонала: опыт Великобритании / А. А. Скавитин // Менеджмент в России и за рубежом. - 2012. - №2. - С. 65-69.
41. Скудитис М.В. Кадровое дело: от приема до увольнения. — Спб., 2011. — 239 с.
42. Станиславчик, Е.Н. «Информационные технологии в школьном образовании»/ Е.Н. Станиславчик. - М.: Ось-89, 2017. - 128 с.
43. Тихомиров М. Ю., Тихомирова Л. В. Кадровое делопроизводство. Об-разцы правовых документов. — М., 2014. — 192 с.
44. Толымбекова Г. С. Автоматизация системы управления кадрами предприятия / Г. С. Толымбекова, Б. К. Султанова, Б. Е. Тайлак //



Международный научный журнал «Инновационная наука». – 2016. - №4. – С. 178 – 180.

45. Труханович Л.В., Щур Д.Л. Справочник по кадровому делопроизвод-ству/ Л.В. Труханович, — 6-е изд., — М.: «Дело и Сервис», 2008. — 608 с.

46. Федеральный закон «Об обществах с ограниченной ответственностью» от 8 февраля 1998 г. № 14-ФЗ (в редакции от 3 июля 2016 г. № 360-ФЗ) // Российская газета. - 2016. - №151.

47. Филоненко Н. В. Анализ управления персоналом предприятия электронного бизнеса [Электронный ресурс] / Н. В. Филоненко // Режим доступа: <https://www.scienceforum.ru/2017/pdf/40540.pdf> (Дата обращения: 13.02.2020).

48. Часов В. В. Информационное обеспечение управленческой деятельности: учебное пособие / В. В. Часов. - М.: Высшая школа, 2013. – 240 с.

49. Шаульская Л. В. Стратегия развития трудового потенциала: монография / Л. В. Шаульская. - СПб.: Питер, 2011. - 502 с.

50. Шейко Т. С. Автоматизированное рабочее место специалиста / Т. С. Шейко // Информационные системы и технологии: 49-я научная конференция аспирантов, магистрантов и студентов (Минск, 4 мая 2013 г.). – Минск: БГУИР, 2013. – С. 54 – 55.

51. Этрилл П. Финансовый менеджмент и управленческий учет для руководителей и бизнесменов: учебное пособие / П. Этрилл. - М.: Альпина Паблишер, 2015. – 656 с.

52. ECMA. ECMA-376: Office Open XML File Formats. ECMA (European Association for Standardizing Information and Communication Systems), Geneva, Switzerland (2006). URL: <http://www.ecma-international.org/publications/standards/Ecma-376.htm> (дата обращения 16.02.2020).



## Приложение А

### Фрагмент кода авторизации в системе

```
<?php
class LoginForm {
    private $login;
    private $password;

    /**
     * @param array $data
     */
    public function __construct( Array $data ) {
        $this->login = isset( $data['login'] ) ? $data['login'] : null;
        $this->password = isset( $data['password'] ) ? $data['password'] : null;
    }

    /**
     * @return bool
     */
    public function validate() {
        return ! empty( $this->login ) && ! empty( $this->password );
    }

    /**
     * @return mixed
     */
    public function getEmail() {
        return $this->login;
    }

    /**
     * @param mixed $login
     */
    public function setEmail( $login ) {
        $this->login = $login;
    }

    /**
     * @return mixed
     */
    public function getPassword() {
        return $this->password;
    }

    /**
     * @param mixed $password
     */
    public function setPassword( $password ) {
        $this->password = $password;
    }
}

<?php
class RegistrationForm {
    private $email;
```

```

private $login;
private $password;
private $passwordConfirm;
/**
 * @param array $data
 */
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;
}

/**
 * @return mixed
 */
public function getEmail() {
    return $this->email;
}

/**
 * @param mixed $email
 */
public function setEmail( $email ) {
    $this->email = $email;
}

/**
 * @return mixed
 */
public function getPasswordConfirm() {
}
}

```

```

<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Кадровые приказы</title>
<!-- Bootstrap -->
<link href="../../css/bootstrap-4.4.1.css" rel="stylesheet">
<script src="../../libraries/tinymce/js/tinymce/tinymce.js" referrerpolicy="origin"></script>

<script>
tinymce.init({
selector: '#content',
plugins: "print",
toolbar: "print",
language: 'ru',
height : "480"
});
</script>

```

```

</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <a class="navbar-brand" href="#">PersonnelOrders</a>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="/">Главная <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="register.php">Регистрация</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="login.php">Авторизация</a>
      </li>
    </div>
  </div>
</footer>
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="../js/jquery-3.4.1.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="../js/popper.min.js"></script>
<script src="../js/bootstrap-4.4.1.js"></script>
</body>
</html>

```

```

<?php
class Password
{
  const SALT_TEXT = 'This is salt text to security this website';

  private $password;
  private $hashedPassword;
  private $salt;

  function __construct($password, $saltText = null)
  {
    $this->password = $password;
    $this->salt = md5(is_null($saltText) ? self::SALT_TEXT : $saltText);
    $this->hashedPassword = md5($this->salt . $password);
  }

  public function __toString()
  {
    return $this->hashedPassword;
  }
}

```

```

<?php
*/
public static function delete($key)
{
  unset($_SESSION[$key]);
}

/**
 * destroy
 */
public static function destroy()
{

```

```

    session_destroy();
}
}

<?php
session_start();
$title = "Добавить кадровый приказ";
$msg = "";
include_once('includes/header.php');
?>
<div class="banner padd">
<div class="container">

<h2 class="white"><?= $title ?></h2>
<ol class="breadcrumb">
<li class="mr-1"><a href="index.php">Главная</a></li>
<li class="active"><?= $title ?></li>
</ol>
<div class="clearfix"></div>
</div>
</div>
<div class="inner-page padd">
<div class="container">
<div class="row">
<div class="col-md-12">

<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">

</div><!-- / Inner Page Content End -->
<?php include_once('includes/footer.php'); ?>

<?php
class DB
{
protected $connetion;

public function __construct($host, $user, $password, $db_name, $port)
{
// connecton
$this->connetion = new mysqli($host, $user, $password, $db_name, $port);

$this->query("SET NAMES UTF8");

if (!$this->connetion) {
throw new Exception('Не могу соединиться с базой данных');
}
}
}

```

```

public function query($sql)
{
if (!$this->connetion) {
return false;
}

$result = $this->connetion->query($sql);

if (mysqli_error($this->connetion)) {
throw new Exception(mysqli_error($this->connetion));
}

if (is_bool($result)) {
return $result;
}

$data = array();
while ($row = mysqli_fetch_assoc($result)) {
$data[] = $row;
}

mysqli_free_result($result);

return $data;
}

public function escape($str)
{
return mysqli_escape_string($this->connetion, $str);
}
}

<?php
session_start();
$title = "Авторизация в системе";

require_once("forms/LoginForm.php");
require_once('DB.php');
require_once('includes/Password.php');
require_once('includes/Session.php');
require_once('Dbsettings.php');

$msg = "";

try {
$db = new DB($host, $user, $password, $db_name, $port);
} catch (Exception $e) {
}

$form = new LoginForm($_POST);

if ($_POST) {
if ($form->validate()) {
/login = $db->escape($form->getLogin());
$password = new Password($db->escape($form->getPassword()));

try {
$res = $db->query("SELECT * FROM users WHERE login = '{$login}' AND hash_pass = '{$password}' LIMIT
1");
} catch (Exception $e) {
}
if (!$res) {
$msg = 'Пользователя с таким учетными данными нет';
}
}
}

```

```

    } else {
    $login = $res[0]['login'];
    Session::set('login', $login);
    header('location: index.php?msg=Вы авторизированы на сайте');
    }
    } else {
    $msg = 'Пожалуйста, заполните все поля!';
    }
}

include_once('includes/header.php');
?>
<!-- Banner Start -->
</div>
</div>
<!-- Heading -->
<h3>Вход на сайт</h3>
<!-- Form -->
<button type="submit" class="btn btn-primary">Войти</button>
<a href="index.php" class="btn btn-primary">Отмена</a>
</form>
</div><!-- / Contact form end -->
</div>
</div>
</div>
</div>
<!-- Contact Us End -->
</div><!-- / Inner Page Content End -->
<?php include_once('includes/footer.php'); ?>

<?php
session_start();
$title = "Регистрация в системе";

require_once("forms/RegistrationForm.php");
require_once('DB.php');
require_once('includes/Password.php');
require_once('includes/Session.php');
require_once('Dbsettings.php');

$msg = "";
try {
    $db = new DB($host, $user, $password, $db_name, $port);
} catch (Exception $e) {
}
$form = new RegistrationForm($_POST);

if ($_POST) {
    if ($form->validate()) {
        $email = $db->escape($form->getEmail());
        $login = $db->escape($form->getLogin());
        $password = new Password($db->escape($form->getPassword()));

        $res = $db->query("SELECT * FROM `users` WHERE email = '{$email}'");
        if ($res) {
            $msg = 'Пользователь с таким эл. адресом уже существует!';
        } else {
            $db->query("INSERT INTO `users` (email, login, password) VALUES ('{$email}','{$login}', '{$password}')");
            $msg = "Регистрация прошла успешно, можете войти на сайт";
        }
    } else {
        $msg = $form->passwordsMatch() ? 'Пожалуйста, заполните все поля' : 'Пароли не совпадают!';
    }
}

```



```

}

include_once('includes/header.php');
?>
<!-- Banner Start -->

<div class="banner padd">
<div class="container">
<!-- Image -->

<!-- Heading -->
<h2 class="white">Регистрация</h2>
<ol class="breadcrumb">
<li class="mr-1"><a href="index.php">Главная</a></li>
<li class="active">Регистрация</li>
</ol>
<div class="clearfix"></div>
</div>
</div>

<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>
</div><!-- / Inner Page Content End -->
<?php include_once('includes/footer.php'); ?>

<?php
class LoginForm {
    private $login;
    private $password;
    /**
     * @param array $data
     */
    public function __construct( Array $data ) {
        $this->login = isset( $data['login'] ) ? $data['login'] : null;
        $this->password = isset( $data['password'] ) ? $data['password'] : null;
    }
    /**
     * @return bool
     */
    public function validate() {
        return ! empty( $this->login ) && ! empty( $this->password );
    }
    /**

```

```

        * @return mixed
        */
        public function getEmail() {
            return $this->login;
        }
/**
        * @param mixed $login
        */
        public function setEmail( $login ) {
            $this->login = $login;
        }
/**
        * @return mixed
        */
        public function getLogin() {
            return $this->login;
        }
/**
        * @param mixed $login
        */
        public function setLogin( $login ) {
            $this->login = $login;
        }
/**
        * @return mixed
        */
        public function getPassword() {
            return $this->password;
        }
/**
        * @param mixed $password
        */
        public function setPassword( $password ) {
            $this->password = $password;
        }
    }

<?php
session_start();
$title = "Авторизация в системе";
require_once("forms/LoginForm.php");
$db = new DB($host, $user, $password, $db_name, $port);
} catch (Exception $e) {
}
$form = new LoginForm($_POST);

if ($_POST) {
    if ($form->validate()) {
        $login = $db->escape($form->getLogin());
        $password = new Password($db->escape($form->getPassword()));
        try {
            $res = $db->query("SELECT * FROM users WHERE login = '{$login}' AND hash_pass =
                '{$password}' LIMIT 1");
        } catch (Exception $e) {
        }
        if (!$res) {
            $msg = 'Пользователя с таким учетными данными нет';
        } else {
            $login = $res[0]['login'];
            Session::set('login', $login);
            header('location: index.php?msg=Вы авторизованы на сайте');
        }
    }
}

```

```

} else {
$msg = 'Пожалуйста, заполните все поля';
}
}
include_once('includes/header.php');
?>
<!-- Banner Start -->
<div class="banner padd">
<div class="container">
<!-- Image -->

<!-- Heading -->
<h2 class="white"><?= $title ?></h2>
<ol class="breadcrumb">
<!-- Banner End -->
<!-- Inner Content -->
<div class="inner-page padd">

<!-- Heading -->
<h3>Вход на сайт</h3>
<!-- Form -->
<b style="color: red;"><?= $msg; ?></b>
<form method="post">
<div class="form-group">
<label for="inputEmail">Логин</label>

```

## Приложение Б

### Фрагмент кода регистрации в системе

```
<?php
session_start();
$title = "Регистрация в системе";

require_once("forms/RegistrationForm.php");
require_once('DB.php');
require_once('includes/Password.php');
require_once('includes/Session.php');
require_once('Dbsettings.php');

$msg = '';
try {
    $db = new DB($host, $user, $password, $db_name, $port);
} catch (Exception $e) {
}
$form = new RegistrationForm($_POST);

if ($_POST) {
    if ($form->validate()) {
        $email = $db->escape($form->getEmail());
        $login = $db->escape($form->getLogin());
        $password = new Password($db->escape($form->getPassword()));

        $res = $db->query("SELECT * FROM `users` WHERE email = '{$email}'");
        if ($res) {
            $msg = 'Пользователь с таким эл. адресом уже существует!';
        } else {
            $db->query("INSERT INTO `users` (email, login, password) VALUES
                ('{$email}', '{$login}', '{$password}')");
            //header('location: public/register.php?msg=Регистрация прошла успешно!');
            $msg = "Регистрация прошла успешно, можете войти на сайт";
        }
    } else {
        $msg = $form->passwordsMatch() ? 'Пожалуйста, заполните все поля' : 'Пароли не
            совпадают';
    }
}

include_once('includes/header.php');
?>
<!-- Banner Start -->

<div class="banner padd">
<div class="container">
<!-- Image -->

<!-- Heading -->
<h2 class="white">Регистрация</h2>
<ol class="breadcrumb">
<li class="mr-1"><a href="index.php">Главная</a></li>
<li class="active">Регистрация</li>
</ol>
<h3>Регистрация на сайте</h3>
```

```

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>

<!-- End -->
</div><!-- / Inner Page Content End -->
<?php include_once('includes/footer.php'); ?>

<?php
class RegistrationForm {
    private $email;
    private $login;
    private $password;
    private $passwordConfirm;
    /**
     * @param array $data
     */
    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;
    }
    /**
     * @return mixed
     */
    public function getEmail() {
        return $this->email;
    }
    /**
     * @param mixed $email
     */
    public function setEmail( $email ) {
        $this->email = $email;
    }
}

```

```

    /**
     * @return mixed
     */
    public function getPasswordConfirm() {
        return $this->passwordConfirm;
    }
/**
     * @param mixed $passwordConfirm
     */
    public function setPasswordConfirm( $passwordConfirm ) {
        $this->passwordConfirm = $passwordConfirm;
    }
/**
     * @return mixed
     */
    public function getPassword() {
        return $this->password;
    }
/**
     * @param mixed $password
     */
    public function setPassword( $password ) {
        $this->password = $password;
    }
/**
     * @return mixed
     */
    public function getLogin() {
        return $this->login;
    }
/**
     * @param mixed $login
     */
    public function setLogin( $login ) {
        $this->login = $login;
    }
}

```

## Приложение В

### Фрагмент кода выхода с системы

```
<?php
session_start();
require_once('../Session.php');

Session::destroy();

header('Location: index.php?msg=Вы вышли!');
```

## Приложение Г

### Фрагмент кода добавления приказа

```
<?php
session_start();
$title = "Добавить кадровый приказ";
$msg = '';
include_once('includes/header.php');
?>

<h2 class="white"><?= $title ?></h2>
<ol class="breadcrumb">
<li class="mr-1"><a href="index.php">Главная</a></li>
<li class="active"><?= $title ?></li>
</ol>
<div class="clearfix"></div>
name="date">
</div>
<div class="form-group">
<label for="content">Содержание приказа</label>
<input type="text" class="form-control" id="order_author" placeholder="ФИО
сотрудника"
name="order_author">
</div>
<button type="submit" class="btn btn-primary">Войти</button>
<a href="index.php" class="btn btn-primary">Отмена</a>
</form>
</div>
</div>
</div>
</div><!-- / Inner Page Content End -->
<?php include_once('includes/footer.php'); ?>
```



## Приложение Д

### 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,  
  `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
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,
`date_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)
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.

```