СОДЕРЖАНИЕ

[Введение 5](#_Toc75259817)

[1 Разработка технического задания. 6](#_Toc75259818)

[1.1 Исходное задание на проектирование. 6](#_Toc75259819)

[1.1 Анализ организационно-технологической структуры предприятия. 6](#_Toc75259820)

[1.1.1 Обследование общих закономерностей функционирования организация. 6](#_Toc75259821)

[1.1.2 Выявление организационной структуры организации, определение функций и основных бизнес-процессов. 6](#_Toc75259822)

[1.1.3 Обследование функций структуры и деятельности базового подразделения. 15](#_Toc75259823)

[1.1.4 Анализ существующей реализации базового бизнес-процесса. 18](#_Toc75259824)

[1.2 Формулировка требований к системе. Концепция системы. 19](#_Toc75259825)

[1.2.1 Определение состава функциональных требований к системе. 19](#_Toc75259826)

[1.2.2 Определение состава сценариев, реализующих требования к системе. 19](#_Toc75259827)

[1.2.3 Разработка содержания сценариев. 20](#_Toc75259828)

[1.2.4 Определение требований к пользовательскому интерфейсу. 24](#_Toc75259829)

[1.2.5 Окончательная формулировка требований к системе. Разработка концепции системы 28](#_Toc75259830)

[2 Анализ и проектирование. 30](#_Toc75259831)

[2.1 Определение классов граничных объектов и классов сущностей. 30](#_Toc75259832)

[2.2 Определение методов объектов. 31](#_Toc75259833)

[3 Разработка системы централизованного хранения и обработки данных. 36](#_Toc75259834)

[3.1 Формирование требований к базе данных 36](#_Toc75259835)

[3.2 Формирование сущностей и их атрибутов 37](#_Toc75259836)

[3.2.2. Выявление связей множеств сущностей и их характеристик 39](#_Toc75259837)

[3.3. Даталогическое проектирование 41](#_Toc75259838)

[3.4. Разработка сценариев работы с данными 47](#_Toc75259839)

[3.5. Разработка механизмов реализации сценариев серверной компоненты 51](#_Toc75259840)

[4. Специальная часть. Разработка веб-приложения на платформе ASP.NET с использованием паттерна MVC 57](#_Toc75259841)

[4.1. Описание технологии 57](#_Toc75259842)

[4.2. Реализация 59](#_Toc75259843)

[5. Технология разработки и программная реализация. 68](#_Toc75259844)

[5.1. Выбор технологии 68](#_Toc75259845)

[5.1.1. Выбор операционной системы. 68](#_Toc75259846)

[5.1.2. Выбор взаимодействия пользователя с операционной системой. 68](#_Toc75259847)

[5.1.3. Выбор языка и среды программирования. 68](#_Toc75259848)

[5.1.4. Выбор технологии взаимодействия пользовательских компонент с данными. 68](#_Toc75259849)

[5.2. Определение физической архитектуры приложения 69](#_Toc75259850)

[6. Аппаратная и административная интеграция информационной системы. 86](#_Toc75259851)

[6.1. Разработка схемы развертывания системы. Установка БД и клиентского приложения 86](#_Toc75259852)

[6.2. Формулировка требований к физическим устройствам и сетевому оборудованию, состав рабочих мест 87](#_Toc75259853)

[6.3. Выбор состава аппаратных средств 88](#_Toc75259854)

[6.4. Расчет потребности персонала 89](#_Toc75259855)

[6.5. Разработка среды интеграции 90](#_Toc75259856)

[7. Общие вопросы администрирования. 95](#_Toc75259857)

[7.1. Определение стратегии администрирования на уровне руководства и целей предприятия. 95](#_Toc75259858)

[7.2. Определение объектов администрирования на уровне предприятия. 96](#_Toc75259859)

[7.2.1. Определение объектов администрирования на уровне разрабатываемой информационной системы 96](#_Toc75259860)

[7.3. Политика администрирования. 97](#_Toc75259861)

[7.3.1. Политика администрирования на уровне предприятия. 97](#_Toc75259862)

[7.3.2. Политика администрирования на уровне разрабатываемой ИС. 98](#_Toc75259863)

[8. Вопросы информационной безопасности. 99](#_Toc75259864)

[8.1. Вопросы информационной безопасности на уровне предприятия 99](#_Toc75259865)

[8.1.1. Анализ угроз 100](#_Toc75259866)

[8.2. Контроль доступа на объекты и в помещения. 108](#_Toc75259867)

[8.3. Обеспечение безопасности с помощью аппаратных средств. 109](#_Toc75259868)

[8.4. Обеспечение безопасности с помощью программных средств. 109](#_Toc75259869)

[8.5. Определение политики управления доступом пользователей. 110](#_Toc75259870)

[8.6. Антивирусная защита информации 110](#_Toc75259871)

[8.7. Политика внешнего доступа к информационной системе. 111](#_Toc75259872)

[8.8. Защита персональных данных 112](#_Toc75259873)

[9. Расчет экономической эффективности. 114](#_Toc75259874)

[9.1. Расчет затрат на проектирование 114](#_Toc75259875)

[9.2. Расчет эксплуатационных затрат 117](#_Toc75259876)

[9.3. Экономическая эффективность 120](#_Toc75259877)

[9.4. Срок окупаемости разработанной системы. 120](#_Toc75259878)

[9.5. Технико-экономические показатели проекта 121](#_Toc75259879)

[Заключение 123](#_Toc75259880)

[Список литературы. 124](#_Toc75259881)

[Приложение 1. Техническое задание 125](#_Toc75259882)

**Аннотация**

Автор дипломной работы: Карпова Екатерина Павловна

Тема работы: «Разработка информационной системы автоматизации процесса расчета заработных плат сотрудников предприятия ООО «Мажор»

Специальность: Информационные системы и технологии

Город: Кемерово

Год выполнения: 2021

Количество страниц: 135

Количество таблиц: 18

Количество иллюстраций: 94

Описание:

Объект исследования: ООО «Мажор»;

Практическая цель дипломного проекта: сокращение издержек при расчете заработных плат сотрудников;

Задачи дипломной работы:

Анализ организации: ООО «Мажор»;

Аналитика проблемных бизнес-процессов деятельности предприятия;

Проектирование информационного сервиса;

Разработка технического задания;

Разработка основных механизмов и алгоритмов сервиса;

Разработка пользовательского интерфейса;

Отладка и тестирование;

Перенос разработок на рабочие базы;

# Введение

В данной работе рассматривалась деятельность предприятия ООО «Мажор», которая направлена на осуществление деятельности ресторана и услуг по доставке питания, а также развлекательных услуг.

Процесс «Обеспечение деятельности предприятия» является основным для данного учреждения. Он включает в себя несколько подпроцессов, одним из которых является процесс «Обеспечение деятельности предприятия», содержащий в себе подпроцесс «Расчет заработной платы сотрудников». На данный момент этот процесс занимает много времени из-за необходимости ручного ввода данных частично в 1С: Предприятие и частично в таблицы Excel, а также из-за заполнения этих же данных в договор, которое производится отдельно в Word.

Цель выпускной работы: Разработка информационной системы автоматизации процесса расчета заработной платы сотрудников.

Для реализации выше поставленной цели необходимо решить следующие задачи:

* анализ деятельности учреждения;
* разработка технического задания;
* решение вопросов, связанных с анализом и проектированием информационной системы;
* разработка системы централизованного хранения и обработки данных;
* разработка интерфейса и программная реализация;
* решение вопросов, связанных с интеграцией в информационную систему предприятия;
* решение вопросов администрирования;
* решение вопросов информационной безопасности;
* расчет экономической эффективности;

# Разработка технического задания.

## Исходное задание на проектирование.

Необходимо разработать информационную систему для автоматизации процесса «Расчет заработной платы сотрудников» для ООО «Мажор», г. Кемерово.

Техническое задание в полном варианте представлено в приложении 1.

## Анализ организационно-технологической структуры предприятия.

### Обследование общих закономерностей функционирования организация.

Полное название: Общество с ограниченной ответственностью «Мажор»

Сокращенное название: ООО «Мажор»

Местонахождение и почтовый адрес: 650066, Российская Федерация, г. Кемерово, пр. Октябрьский 53/3.

Контактный телефон: +7(3842) 35-12-56.

Основной вид деятельности компании – деятельность ресторанов и услуг по доставке продуктов питания.

### Выявление организационной структуры организации, определение функций и основных бизнес-процессов.

Структура предприятия включает в себя 5 отделов:

* Технический отдел – регулярная проверка корректности работы информационных систем предприятия, ликвидация возможных неполадок. Проверка работы и устранения неполадок в других системах предприятия. В состав отдела входят инженер-программист и слесарь-электрик.
* Администрация – принимает решение о приеме/увольнении сотрудников, заключая договора, осуществляет контроль за функционированием предприятия. Обеспечение работы по эффективному обслуживанию посетителей заведения, созданию для них комфортных условий. Контролирует соблюдение работниками организации трудовой и производственной дисциплины, правил и норм охраны труда, техники безопасности, требований производственной санитарии и гигиены. Осуществляет контроль над исполнением работниками указаний руководства организации. В составе подразделения генеральный директор и администраторы.
* Бухгалтерия - формирование учетной политики, ведение бухгалтерского учета, своевременное представление полной и достоверной бухгалтерской отчетности. Начисление заработных плат остальным сотрудникам. Ведение первичного бухгалтерского учета. Прием, контроль документации (товарно-транспортных накладных, кассовых, кадровых документов и т.д.). Выплата заработной платы сотрудникам. Составление и сдача налоговой отчетности в фонды РФ и налоговую инспекцию. Участие в инвентаризациях. Составление отчетных калькуляций себестоимости продукции (продукты питания, напитки, услуги). В составе подразделения главный бухгалтер и бухгалтер.
* Сфера обслуживания - предоставление услуг предприятия гостям, прием и выдача заказов, бронирование столов. Обеспечение комфортного отдыха и пользования услугами предприятия. В составе подразделения официанты, гардеробщики, уборщики, маркеры.
* Кухня – приготовление еды, выдача заказов. Закуп продуктов питания. Составление и редактирование меню. Планирование и организация технологического процесса по приготовлению пищи, контроль и анализ эффективности использования технологического оборудования и рабочего времени. Контроль соблюдения технологии приготовления пищи. Составление технологических карт. Составление заявки на необходимые продовольственные товары, полуфабрикаты и сырье, обеспечение их своевременного приобретения и получения со склада. Контроль ассортимента, количества и сроков его поступления и реализации. В составе подразделения шеф-повар и повара.

Организационная структура предприятия представлена на рис. 1-2 в виде состава подразделений и состава сотрудников.

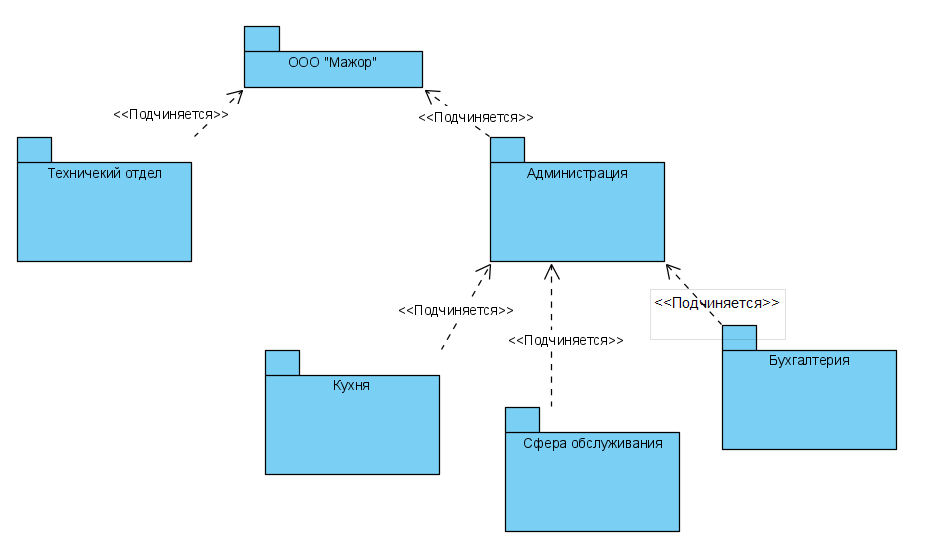


Рис. 1. Структура подразделений.

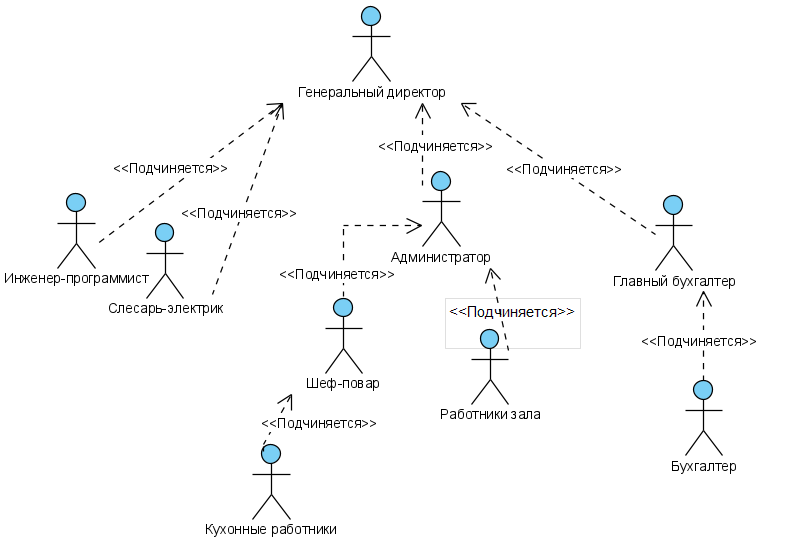


Рис. 2. Состав сотрудников.

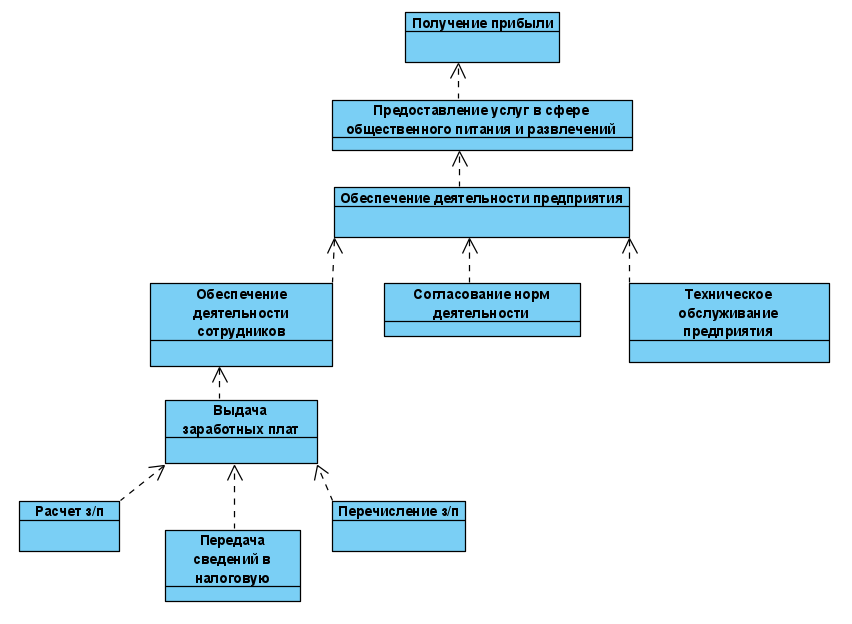
Таблица 1 описывает функции выявленных подразделений предприятия ООО «Мажор» и их основные характеристики.

*Таблица 1*

*Описание функций подразделений   
и их основных характеристик*

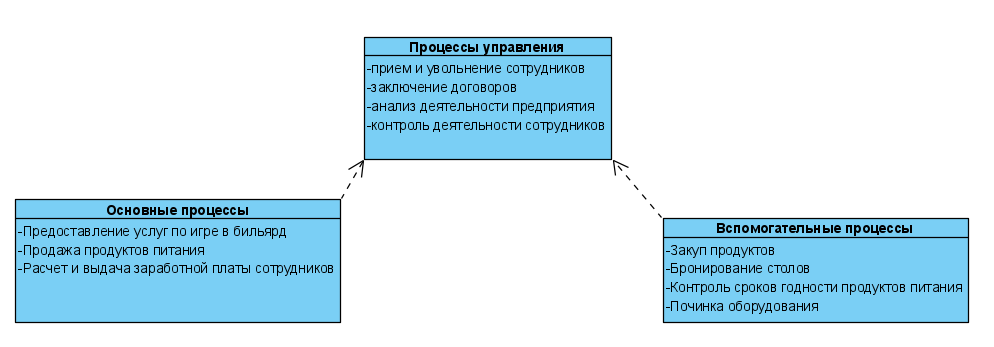
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код подразделения | Наименование подразделения | Выполняемые функции | Число сотрудников | Функциональная подчиненность | Структурная подчиненность |
| 001  (ВО) | Волна на Октябрьском | - | 22 | - | - |
| 002  (АДМ) | Администрация | Решение о приеме/увольнении сотрудников, заключение договоров, контроль за функционированием предприятия и деятельностью сотрудников | 4 | - | ВО |
| 003  (БУХ) | Бухгалтерия | Расчет заработной платы сотрудников, ее выдача, заключение договоров с поставщиками, контроль исполнения договоров и поставок, налоговый учет, оплата счетов | 2 | АДМ | ВО |
| 004  (КХ) | Кухня | Приготовление еды, закуп продуктов питания, составление и обновление меню | 3 | АДМ | ВО |
| 005  (СО) | Сфера обслуживания | Предоставление услуг предприятия гостям, прием и выдача заказов, бронирование столов | 9 | АДМ | ВО |
| 006  (ТО) | Технический отдел | Ремонт оборудования, проверка систем | 3 | - | ВО |

Основными целями деятельности предприятия являются получение прибыли, сокращение издержек, увеличение количества клиентов. Выполнение основных целей обеспечивается, в первую очередь, выполнением составляющими подцелями, представленными на рис. 3.

**

*Рис. 3.Цели бизнес-процессов*

Все бизнес-процессы предприятия можно разделить на три группы (рис. 4):

**

*Рис. 4. Состав бизнес-процессов*

Бизнес-процессы управления обеспечивают координацию основных и вспомогательных бизнес-процессов предприятия. Их выполнение осуществляет отдел администрации. Бизнес-процессы управления включают в себя:

Прием и увольнение сотрудников

Заключение договоров

Анализ деятельности предприятия

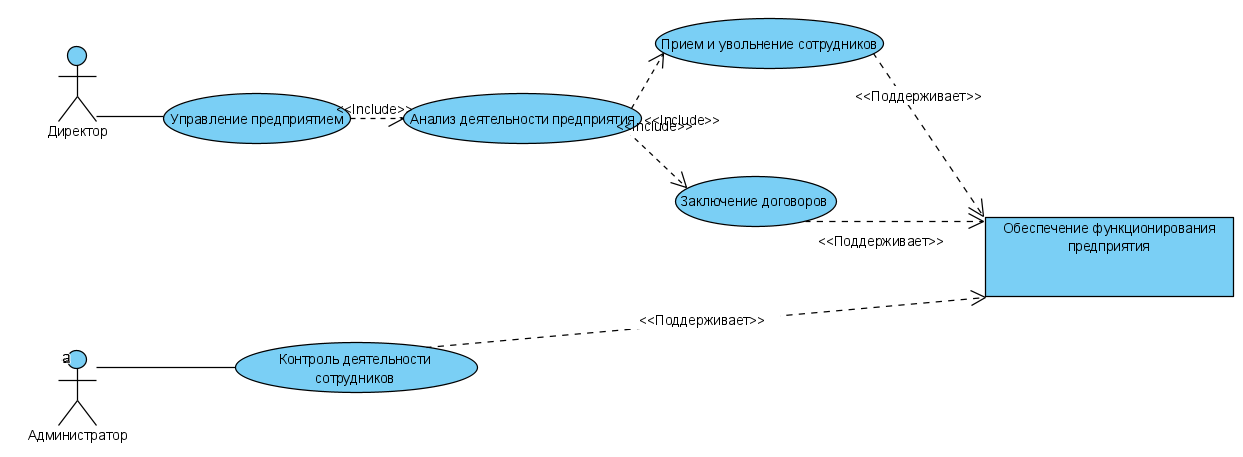
Контроль деятельности сотрудников

Бизнес-процессы управления предприятием рассмотрены в таблице 2. Диаграмма декомпозиции бизнес-процессов управления представлена на рис. 5.

*Таблица 2*

*Бизнес-процессы управления*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код БП | Наименование БП | Степень важности БП | Отдел-владелец БП | Отдел-исполнитель БП | БП-источник | БП-преемник |
| УС | Прием и увольнение сотрудников | A | ВО | АДМ | УС | УС |
| Д | Заключение договоров | A | ВО | АДМ | УС | Д |
| ДП | Анализ деятельности предприятия | А | ВО | АДМ | УБ, ППП | ДП |
| КД | Контроль деятельности сотрудников | А | ВО | АДМ | УБ, ППП | ЗП |

  
Рис. 5. Декомпозиция бизнес-процессов управления

Основные бизнес-процессы направлены непосредственно на получение прибыли от деятельности предприятия. Их выполнение осуществляют отделы бухгалтерии, работники зала, кухня. Основные бизнес-процессы включают в себя:

Предоставление услуг по игре в бильярд

Продажа продуктов питания

Расчет и выдача заработной платы сотрудников

Основные бизнес-процессы предприятия рассмотрены в таблице 3. Диаграмма декомпозиции основных бизнес-процессов представлена на рис. 6.

*Таблица 3*

*Основные бизнес-процессы*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код БП | Наименование БП | Степень важности БП | Отдел-владелец БП | Отдел-исполнитель БП | БП-источник | БП-преемник |
| ППП | Продажа продуктов питания | A | ВО | КХ, СО | ППП | ППП |
| УБ | Предоставление услуг по игре в бильярд | A | ВО | СО | УБ | УБ |
| ЗП | Расчет и выдача заработной платы сотрудникам | A | АДМ | БУХ | ЗП | ЗП |

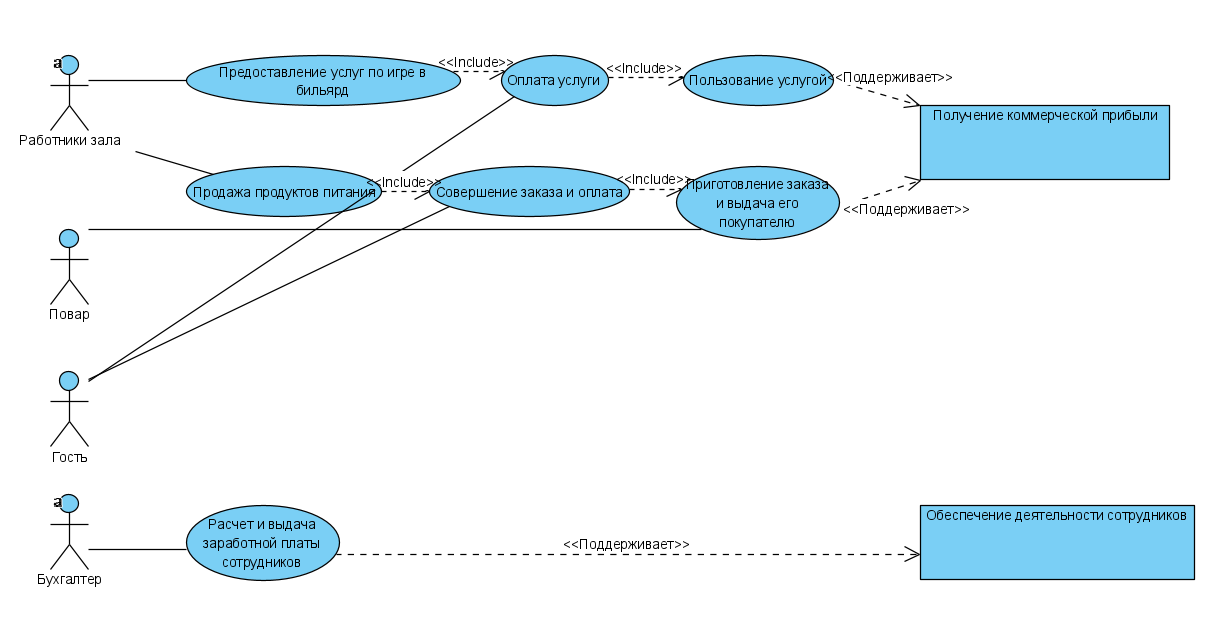


Рис. 6. Декомпозиция основных бизнес-процессов

Вспомогательные бизнес-процессы - поддерживают инфраструктуру организации. Их выполнение осуществляют системный отдел, работники зала, кухня. Вспомогательные бизнес-процессы включают в себя:

Закуп продуктов

Бронирование столов

Контроль сроков годности продуктов питания

Починка оборудования

Вспомогательные бизнес-процессы предприятия рассмотрены в таблице 4. Диаграмма декомпозиции Основные бизнес-процессы представлена на рис. 7.

*Таблица 4*

*Вспомогательные бизнес-процессы*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код БП | Наименование БП | Степень важности БП | Отдел-владелец БП | Отдел-исполнитель БП | БП-источник | БП-преемник |
| ЗПР | Закуп продуктов | A | ВО | КХ | ППП | ППП |
| БС | Бронирование стола | С | ВО | СО | ППП, УБ | ППП, УБ |
| СГ | Контроль сроков годности продуктов питания | A | АДМ | КХ | ППП | ППП |
| ПОЧ | Починка оборудования | А | АДМ | ТО | ППП, УБ | ППП, УБ |

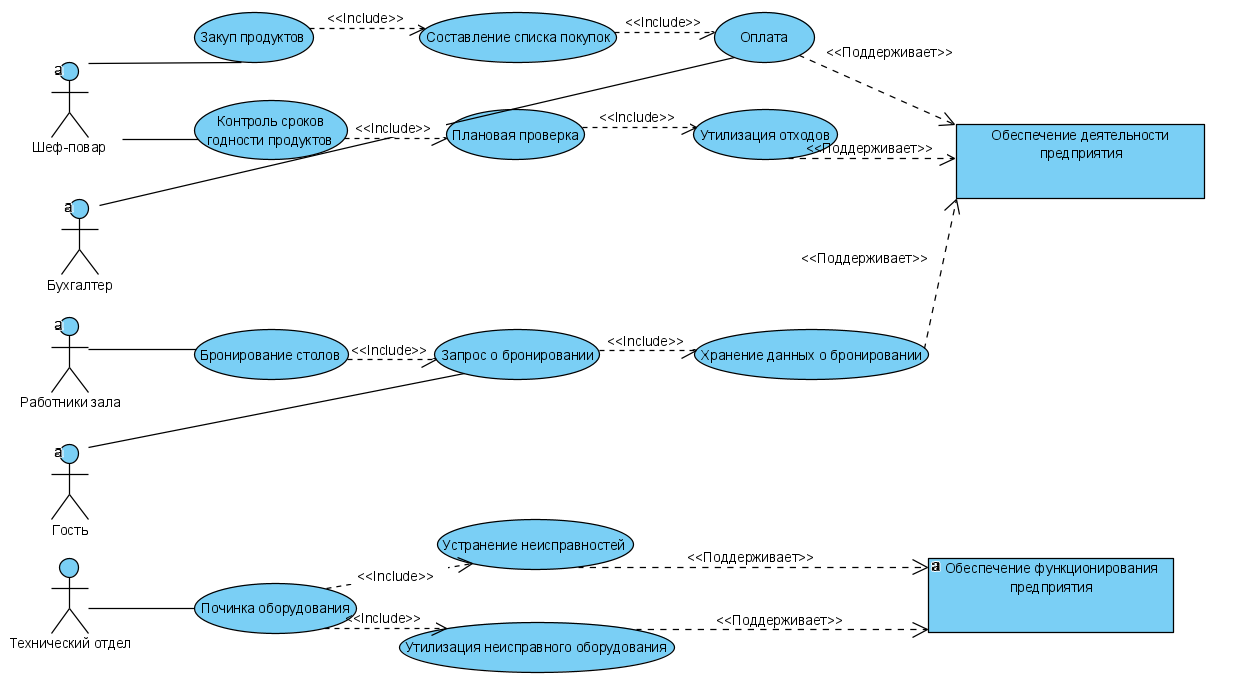


Рис. 7. Декомпозиция вспомогательных бизнес-процессов

В качестве базового бизнес-процесса для подробного рассмотрения и автоматизации принят процесс "Расчет заработной платы сотрудников". Данный бизнес-процесс участвует в бизнес-процессах: контроль деятельности сотрудников, обеспечение деятельности сотрудников. На текущий момент данный бизнес-процесс имеет большие временные издержки, что отражается на работе бухгалтерии и, следовательно, предприятия в целом. В связи с неавтоматизированной системой, данный процесс требует много времени, что приостанавливает деятельность бухгалтерии. От качества выполнения данного процесса зависит эффективность работы отдела бухгалтерия. Далее рассмотрим этот проблемный бизнес-процесс более подробно.

### Обследование функций структуры и деятельности базового подразделения.

Процесс «Расчет заработной платы сотрудников» является базовым и нуждается в автоматизации.

Процесс «Расчет з/п сотрудников» включает в себя ввод в систему ФИО сотрудника, заработную плату которого нужно рассчитать, а также ввода периода расчета. Все данные сохраняются.

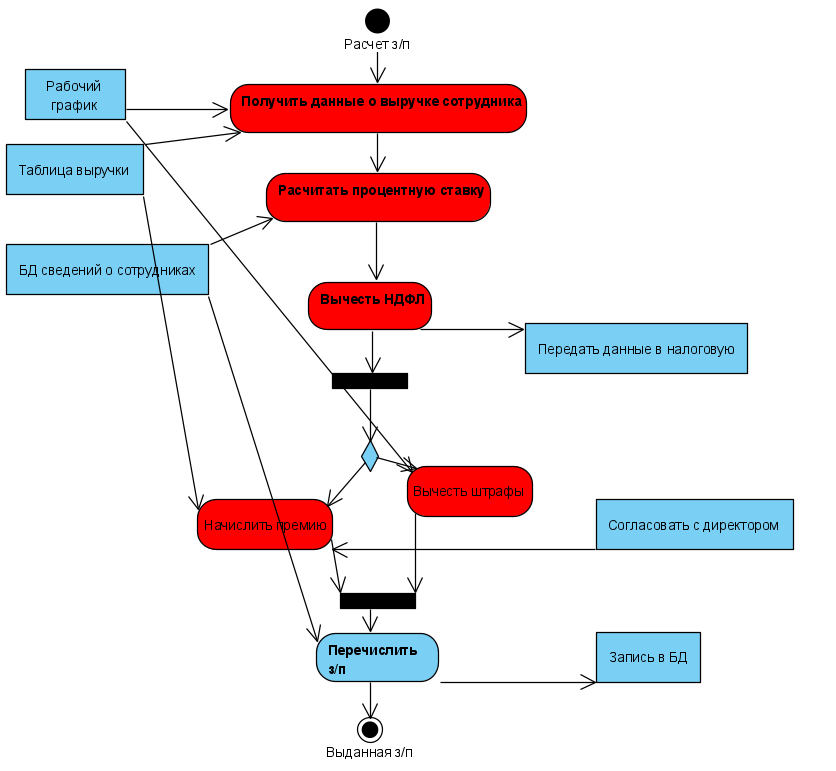
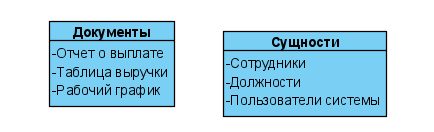


Рис. 8. Декомпозиция процесса «Расчет заработной платы сотрудников».

Были выявлены классы бизнес-объектов предметной области, которые можно разделить на две группы: документы и сущности:

  
Рис. 9. Выявленные классы бизнес-объектов

Классы бизнес-объектов «Документы» с указанием атрибутов представлены на рис. 10.

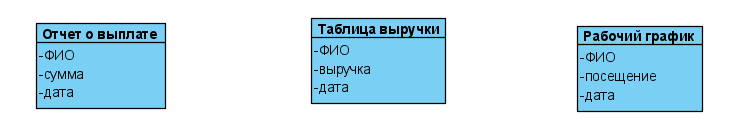


Рис. 10. Диаграмма классов бизнес-объекта «Документы»

К документам относятся:

* Документ «Отчет о выплате» - отчет о проведении выплаты сотруднику, создается автоматически по факту проведения выплаты и нажатию подтверждающей кнопки, включает в себя информацию о выплате (данные сотрудника, сумма выплаты, дата произведения выплаты).
* Таблица выручки – сведения о выручках сотрудников за отработанные смены.
* Рабочий график – документ, содержащий сведения о запланированных рабочих сменах и их посещении сотрудниками.

Класс бизнес-объектов «Сущности» с указанием атрибутов представлены на рис. 11.

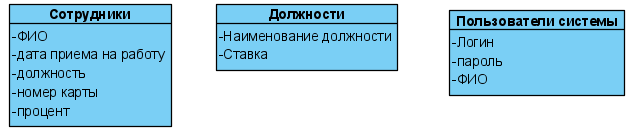


Рис. 11. Диаграмма классов бизнес-объекта «Сущности»

Сущностями являются:

* Сотрудники – сотрудники предприятия.
* Должности – должности, имеющиеся на предприятии.
* Пользователи системы – пользователи, имеющие доступ к ИС.

В результате анализа проведенной декомпозиции были выявлены следующие автоматизируемые бизнес-решения для процесса «Расчет заработной платы сотрудников»: «Получение данных о выручке сотрудника», «Расчет процентной ставки», «Вычет НДФЛ», «Вычет штрафов», «Начисление премий» (рис. 12).

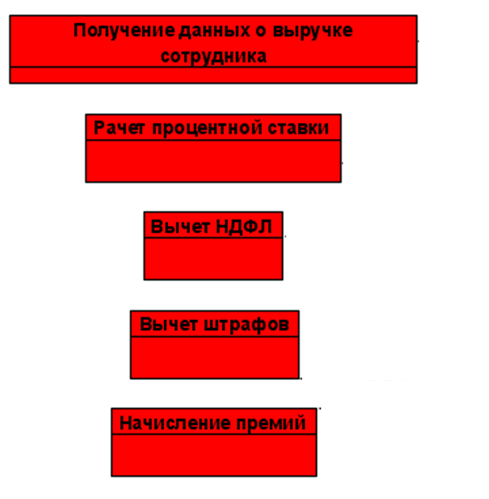


Рис. 12. Автоматизируемые бизнес-решения.

Все эти бизнес-решения можно сгруппировать следующим образом: 1) Внесение и сохранение данных; 2) Поиск и просмотр данных; 3) Формирование отчета.

### Анализ существующей реализации базового бизнес-процесса.

Бизнес-процесс «Расчет заработной платы сотрудников» выполняется вручную сотрудниками бухгалтерии (бухгалтером и главным бухгалтером). В настоящее время в этом процессе для работы с данными используют программу «1С: Предприятие» (для учета данных и дальнейшей работы с ними), Excel (для составления списков данных) и Word (для составления и печати). Сотрудникам приходится вводить одни и те же данные в разные программы и после вручную составлять отчет, что приводит к большим издержкам при выполнении бизнес-процесса. Сотрудники тратят большое количество времени на расчет.

Поэтому целесообразно автоматизировать данный бизнес-процесс для достижения целей: сокращения временных издержек, уменьшения нагрузки на сотрудников, точности выполнения бизнес-процесса.

## Формулировка требований к системе. Концепция системы.

### Определение состава функциональных требований к системе.

Каждому ранее выявленному автоматизируемому элементу поставлены в соответствие функциональные требования. Кроме того, в процессе проектирования было выявлено новое функциональное требование – «Настройка доступа к функционалу - чтение или чтение/запись».

Все функциональные требования разобьем на три функциональные подсистемы согласно диаграмме, приведенной на рис. 13.

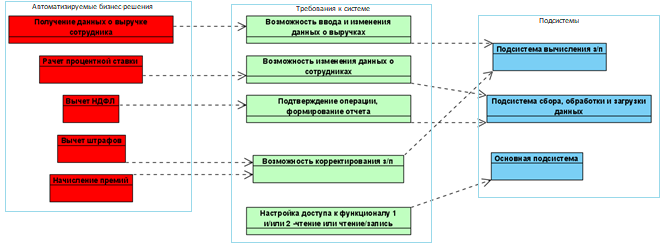


Рис. 13. Функциональные требования к системе.

### Определение состава сценариев, реализующих требования к системе.

Для определения состава сценариев, реализующих требования, была построена диаграмма вариантов использования (рис. 15). Выявлены два типа пользователей системы:

1. Главный бухгалтер – имеет полный доступ к функционалу (чтение илии чтение/запись), может вносить изменения в данные о выручках и сотрудниках, должностей, создание отчета. Его роль может играть любой другой сотрудник, которому выдан данный уровень доступа.
2. Сотрудник – имеет частичный доступ к функционалу (чтение). Ему доступны сценарии просмотра всех таблиц и создания отчета.

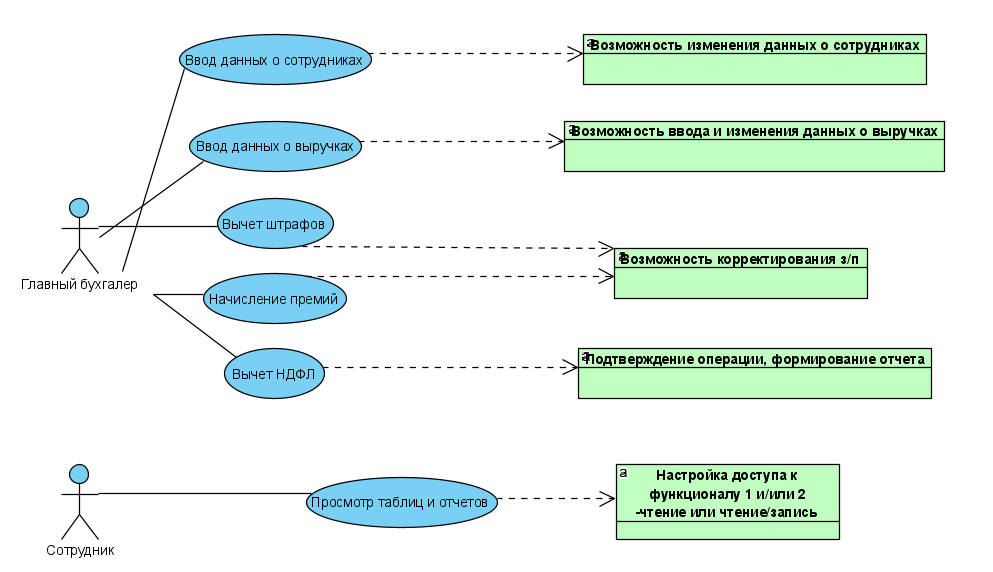
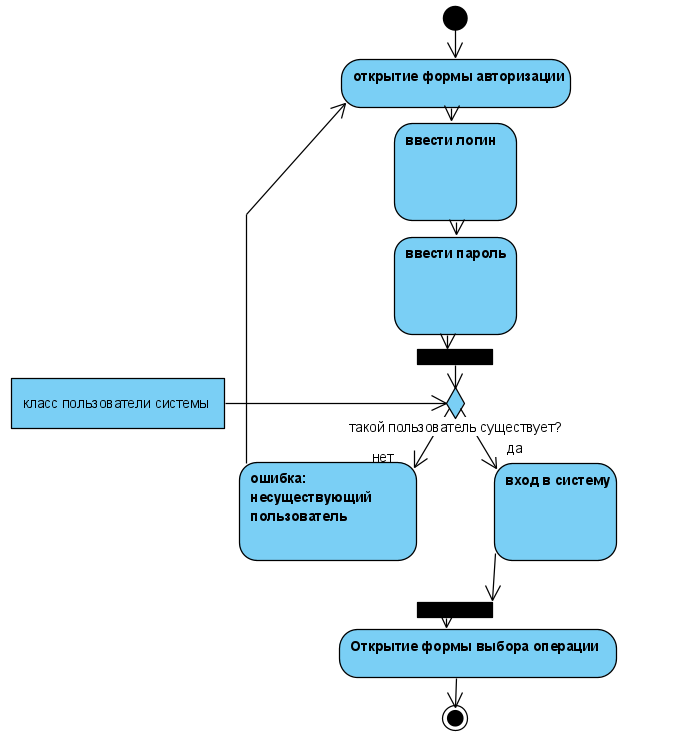


Рис. 15. Состав сценариев

### Разработка содержания сценариев.

При разработке содержания сценариев были построены диаграммы деятельности, с помощью которых был определен набор используемых объектов-сущностей и граничных объектов.

Состав объектов процесса «Авторизация» (соответствующая диаграмма деятельности представлена на рис. 16):

*  
Рис. 16. Диаграмма деятельности для процесса «Авторизация»*

Граничные объекты:

Форма «Авторизация» – меню системы, с помощью которого осуществляется вход в программу по уникальному логину и паролю пользователя системы;

Форма выбора операции – основное меню программы.

Объекты-сущности:

Класс «Пользователи системы» – класс сущностей, содержит сведения о существующих пользователях системы.

Деятельности:

Открыть форму авторизации;

Ввод логина и пароля;

Выполнить/отклонить вход в систему;

Открытие формы выбора операции.

Состав объектов процесса «Изменение данных о сотруднике» (соответствующая диаграмма последовательности приведена на рис. 17):

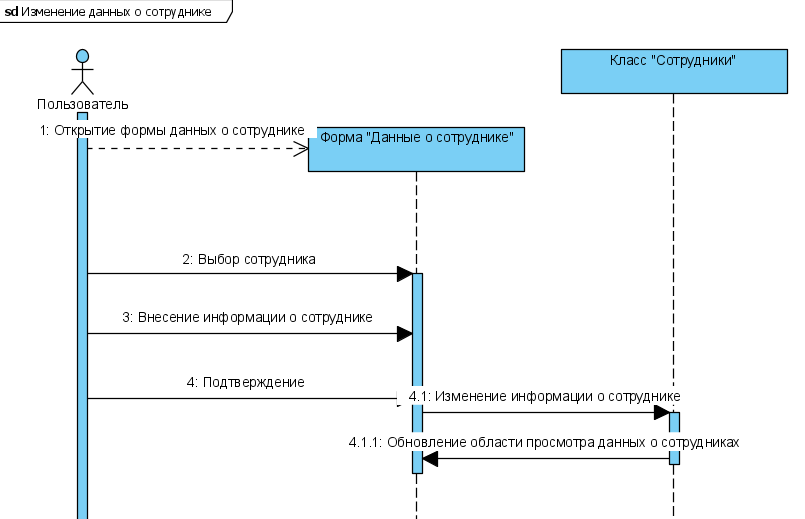


Рис. 17. Диаграмма последовательности для процесса «Изменение данных о сотруднике»

Граничные объекты:

Форма «Данные о сотруднике» - граничный объект взаимодействия с системой, с помощью которого осуществляется добавление, изменение и удаление соответствующих данных.

Объекты сущности:

Класс «Сотрудники» – класс сущностей, список всех сотрудников отдела, предназначенный для выбора сотрудника, для которого редактируются данные.

Деятельности:

Открытие формы данных о сотруднике;

Выбор сотрудника;

Внесение информации;

Подтверждение изменений.

Состав объектов процесса «Расчет заработной платы сотрудников» (соответствующая диаграмма последовательности приведена на рис. 18):

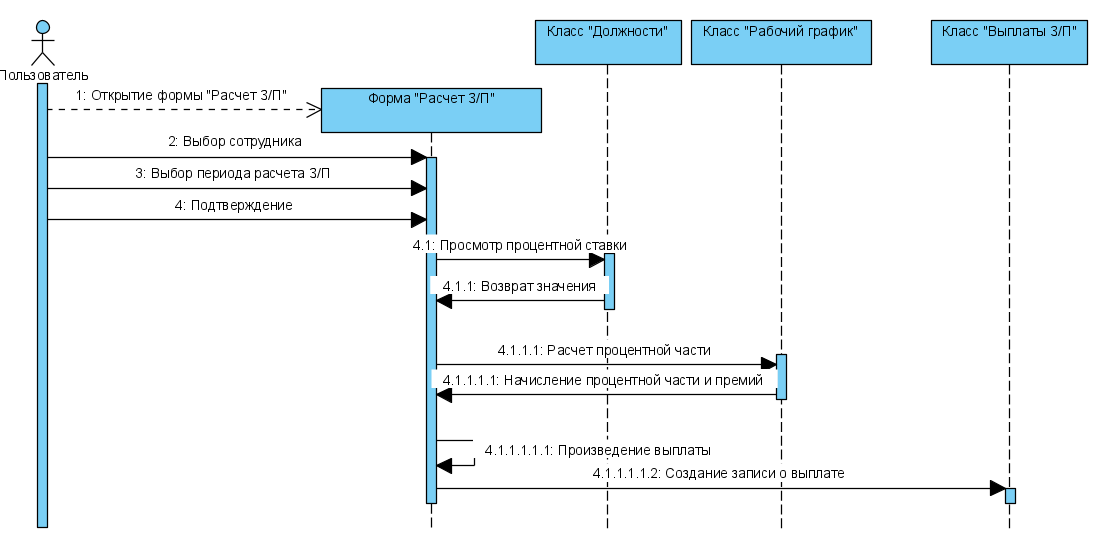


Рис. 18. Диаграмма последовательности для процесса «Расчет заработной платы сотрудников»

Граничные объекты:

Форма «Расчет з/п» – форма осуществления расчета.

Объекты сущности:

Класс «Должности» - класс сущностей, список всех должностей предприятия, предназначенный для учета процентной ставки.

Класс «Рабочий график» - класс сущностей, содержащий сведения о выходах сотрудников на работу, предназначен для учета процентной части з/п.

Класс «Выплаты» - класс сущностей, содержащий сведения о проведенных выплатах.

Деятельности:

Открыть форму «Расчет з/п» – открытие формы для расчета;

«Выбор сотрудника», «Выбор периода расчета З/П» - ввод необходимых данных для выполнения операции;

Подтверждение – старт расчета;

«Просмотр процентной ставки», «Расчет процентной части» и возврат значений – автоматизированные действия;

Произведение выплаты – требует подтверждения пользователя;

Создание записи о выплате – формирование отчета.

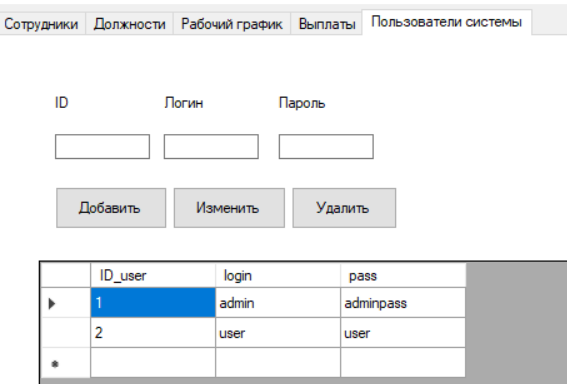
### Определение требований к пользовательскому интерфейсу.

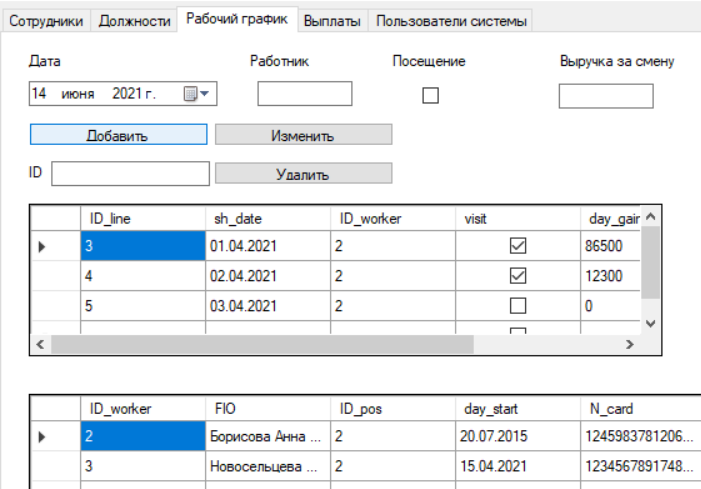
Пользовательский интерфейс должен разрабатываться с использованием стандартных элементов Windows Forms: кнопок Button, текстовых полей TextBox, выпадающих списков ComboBox, таблиц DataGridView, переключателей RadioButton и CheckBox, списков ListBox. Для каждого бизнес-требования целесообразно создать отдельную экранную форму.

Выделены следующие страницы системы:

* Авторизация – форма входа в систему;
* Сотрудники – вкладка, содержащая список сотрудников. Содержит кнопки добавления и удаления сотрудника, изменения данных.
* Должности – страница, которая содержит в себе список должностей. Присутствуют кнопки для добавления и удаления должности, изменения данных.
* Рабочий график – страница, которая содержит в себе данные о входах сотрудников на работу. Присутствуют кнопки добавления, изменения и удаления данных.
* Выплаты – страница, которая содержит в себе список произведенных выплат. Присутствуют кнопки добавления, изменения и удаления данных.
* Пользователи системы – страница, которая содержит в себе список пользователей системы. Присутствуют кнопки добавления, изменения и удаления данных.

Предварительные эскизы пользовательских интерфейсов приведены на рис. 19-25.

*  
Рис 19. Интерфейс страницы «Пользователи системы»*

*  
Рис 20. Интерфейс страницы «Рабочий график»*

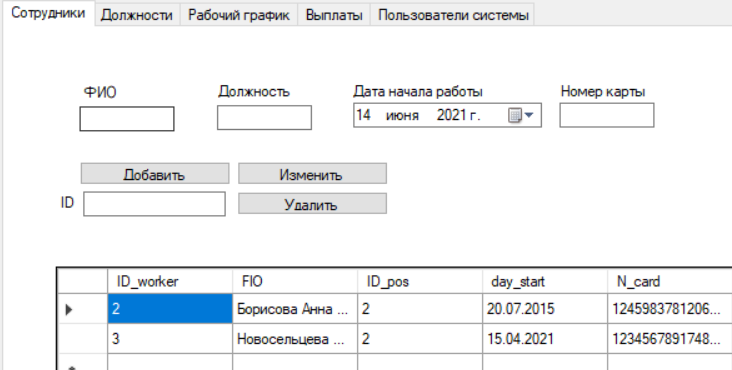


Рис. 21. Интерфейс страницы «Сотрудники»

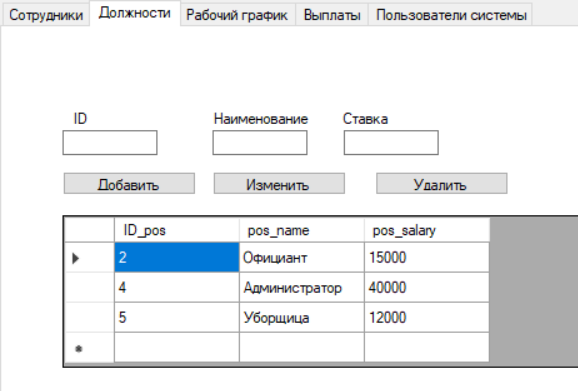
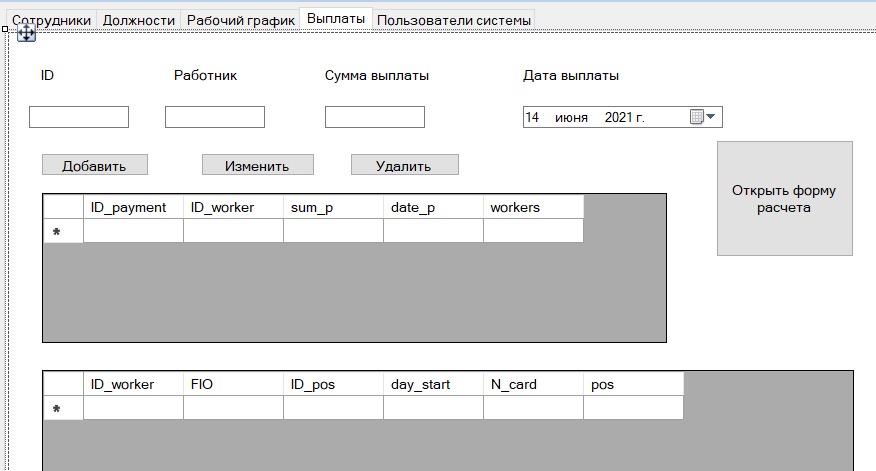
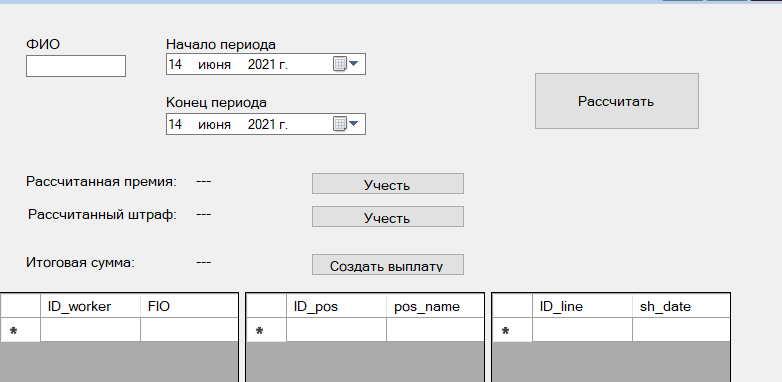
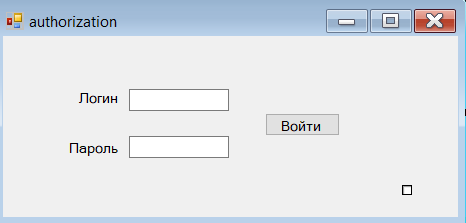


Рис. 22. Интерфейс страницы «Должности»

  
Рис. 23. Интерфейс страницы «Выплаты»

  
Рис. 24. Интерфейс формы «Форма расчета»

  
Рис. 25. Интерфейс формы «Авторизация»

### Окончательная формулировка требований к системе. Разработка концепции системы

**Функциональные требования к системе:**

1. Возможность ввода, изменения и удаления данных о сотрудниках. Модель включает в себя: ФИО сотрудника, дату приема на работу, должность, номер банковской карты, процентную ставку заработной платы.
2. Возможность ввода, изменения и удаления данных о должностях. Модель включает в себя: наименование должности, ставку по должности.
3. Возможность ввода, изменения и удаления данных о пользователях системы. Модель включает в себя: ФИО сотрудника, его логин и пароль.
4. Настройка доступа к функционалу - чтение или чтение/запись.

**Нефункциональные требования к системе:**

1. Производительность системы – обеспечить достаточно быстрый поиск и вывод нужной информации;
2. Скорость ввода или изменения уже существующих данных должна составлять не более 1 минуты;
3. Время отклика системы на запрос должно составлять не более 10 секунд;
4. Требование информационной безопасности – аутентификация по логину и паролю пользователя.

**Проектные требования:**

* Сервер: ЦП – уровня Pentium Core2Due; ОЗУ – 2 Гб;
* предполагаемый тип СУБД – MS SQL Server 2008;
* IIS, Windows Server 2008;
* персонал, работающий с информационной системой, должен обладать навыками работы с ПК.

**Требования разработчика:**

Система должна строиться на следующих технологиях:

* Microsoft Visual Studio 2017;
* Entity Framework интерфейс

# Анализ и проектирование.

## Определение классов граничных объектов и классов сущностей.

На основе объектов, выявленных при разработке сценариев, было определено 5 классов сущностей, их описание с указанием атрибутов представлено ниже:

* + - 1. «workers» – данные о сотруднике - соответствует объекту «Список сотрудников». Имеет атрибуты:
* ФИО – имя сотрудника(string);
* Должность – должность сотрудника (string);
* Дата приемана работу (date);
* Номер карты (string).
  + - 1. «payments» – данные о выплатах - определен на основе соответствующего объекта «Выплаты». Имеет атрибуты:
* ФИО – данные сотрудника, для которого произведена выплата (string);
* Сумма выплаты (float);
* Дата выплаты (date).
  + - 1. «w\_shedule» – рабочий график - определен на основе объекта «Рабочий график». Имеет атрибуты:
* Дата (date);
* ФИО – данные сотрудника (sting);
* Посещение (bool);
* Выручка за смену (float).
  + - 1. «pos» – список должностей - определен на основе объекта «Должности». Имеет атрибуты:
* Наименование должности (string);
* Ставка по должности (float).
  + - 1. «users» – список пользователей системы – определен на основе объекта «Пользователи системы». Имеет атрибуты:
* Логин пользователя (string);
* Пароль пользователя (string);

В качестве граничных были выявлены следующие классы:

1. Форма «Авторизация» - форма входа в приложение.
2. Форма «Выбор операции» - форма выбора вкладки для просмотра/изменения данных.
3. Форма «Данные о сотруднике» – форма данных о сотрудниках.
4. Форма «Расчет з/п» – форма непосредственно выполняющая расчет з/п.

## Определение методов объектов.

Для определения методов объектов были построены диаграммы последовательности для сценариев требований (рис. 26, 27, 28). Для каждой диаграммы представлена таблица с описанием методов. Каждое сообщение, направленное на объект, рассматривается как метод данного объекта и включается как операция, реализующая данный метод в соответствующий класс. Определенные на данном шаге методы представлены в таблицах 5-7.

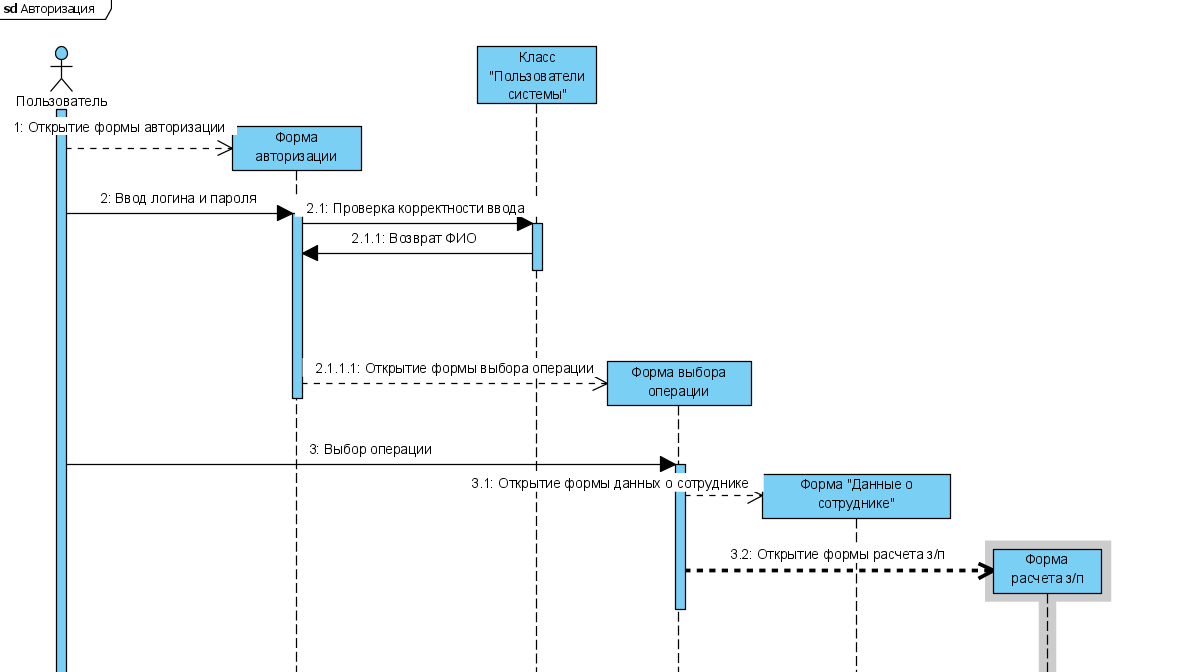


Рис. 26. Диаграмма последовательности для сценария «Составление отчета»

Таблица 5

*Определенные методы объектов*

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Класс | Входные параметры | Тип возврата |
| Проверка корректности | Форма авторизации | Логин, пароль | Форма выбора операции |

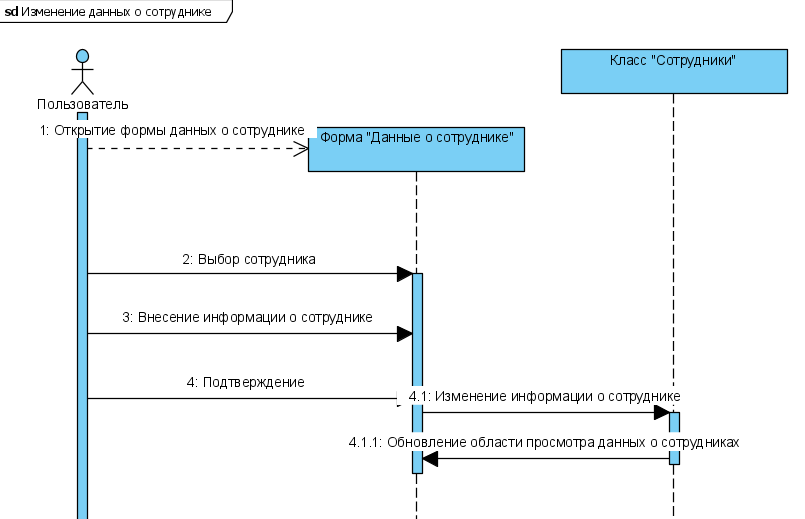


Рис. 27. Диаграмма последовательности для сценария «Изменение данных о сотруднике»

Таблица 6

*Определенные методы объектов*

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Класс | Входные параметры | Тип возврата |
| Вызов формы данных о сотруднике() | Общая форма | - | Форма данных о сотруднике () |
| Изменение информации | worker | ФИО, должность, номер карты, дата начала работы | worker |

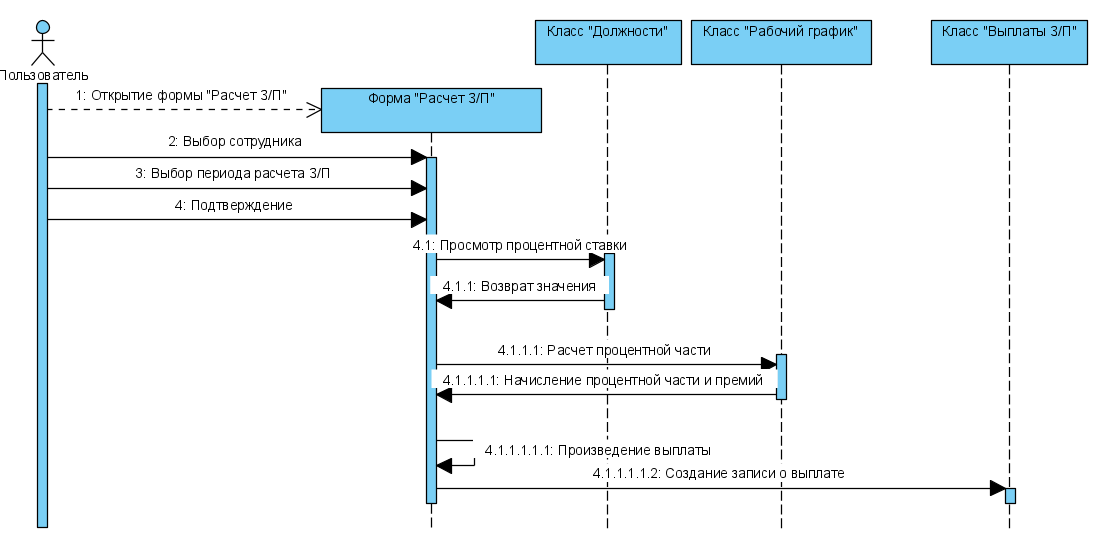


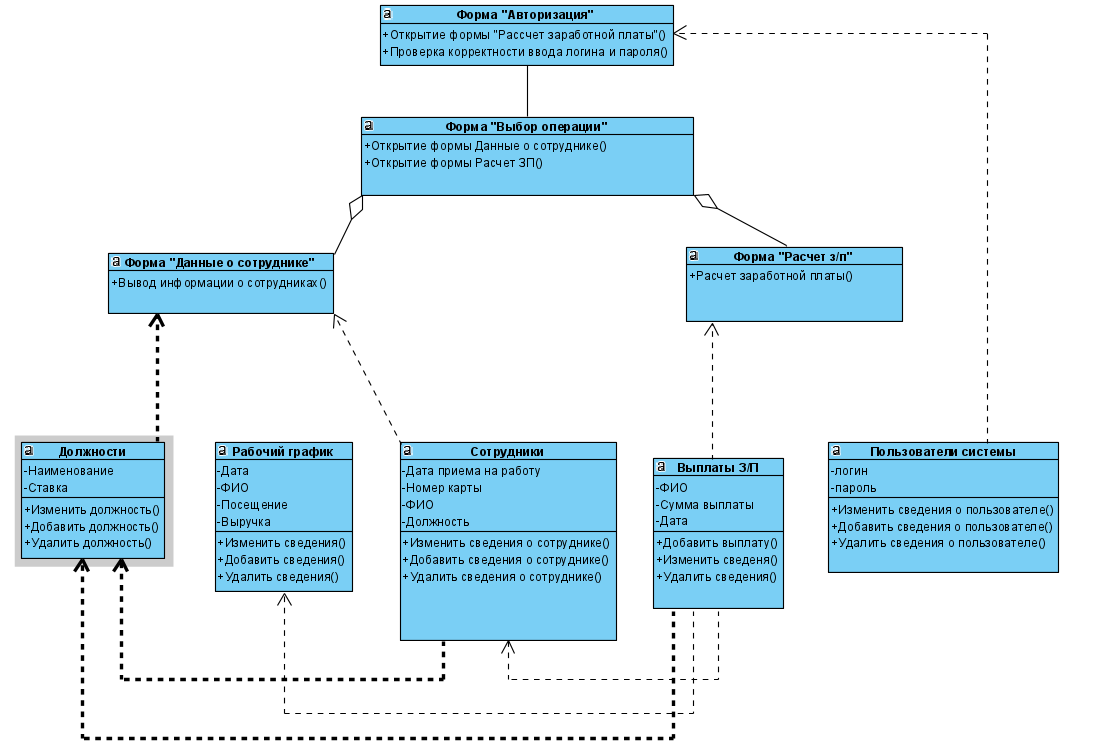
Рис. 28. Диаграмма последовательности для сценария «Расчет з/п»

Таблица 7

*Определенные методы объектов*

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Класс | Входные параметры | Тип возврата |
| Вызов формы расчета з/п() | Общая форма | - | Форма расчета з/п () |
| Выбор периода расчета | Salary\_f | Дата : DateTime | void |
| Просмотр процентной ставки | pos | pos\_salary | Salary\_sum : float |
| Расчет процентной части | w\_shedule | day\_gain | Salary\_sum : float |
| Создание записи о выплате | Salary\_f | Salary\_sum | void |

Итоговая диаграмма классов с указанием всех методов представлена на рис. 29.



*Рис. 29. Итоговая диаграмма классов с методами выявленных классов*.

# Разработка системы централизованного хранения и обработки данных.

## Формирование требований к базе данных

На основании требований заказчика была выбрана реляционная база данных под управлением Microsoft SQL Server. На предприятии имеются все ресурсы для ее поддержки.

При проектировании базы нужно было предположить возможность создания пользователей с разным уровнем доступа к информации, которая хранится в базе данных, а также функцию создания резервной копии данных. База данных должна содержать в себе достаточную информацию об объектах (сотрудниках, должностях) для их добавления, удаления и редактирования.

Еще одно требование заказчика – реализация доступа к данным с помощью технологии Entity Framework. Данная технология позволяет работать с данными независимо от типа хранилища.

База данных должна обеспечить выполнение основной функции хранения данных.

Клиентская компонента должна реализовывать следующие пункты:

* Чтение или чтение/запись в зависимости от уровня доступа пользователя.
* Ввод данных осуществляется пользователем при помощи клавиатуры.
* Подтверждение действий пользователем осуществляется при помощи экранных кнопок.
* Для упрощения ввода данных пользователем могут использоваться выпадающие списки и другие элементы форм.

Серверная компонента реализует следующие пункты:

* Выборка данных.
* Обращение к БД происходит с целью добавления, удаления или изменения данных.
* Проверка целостности введенных данных.

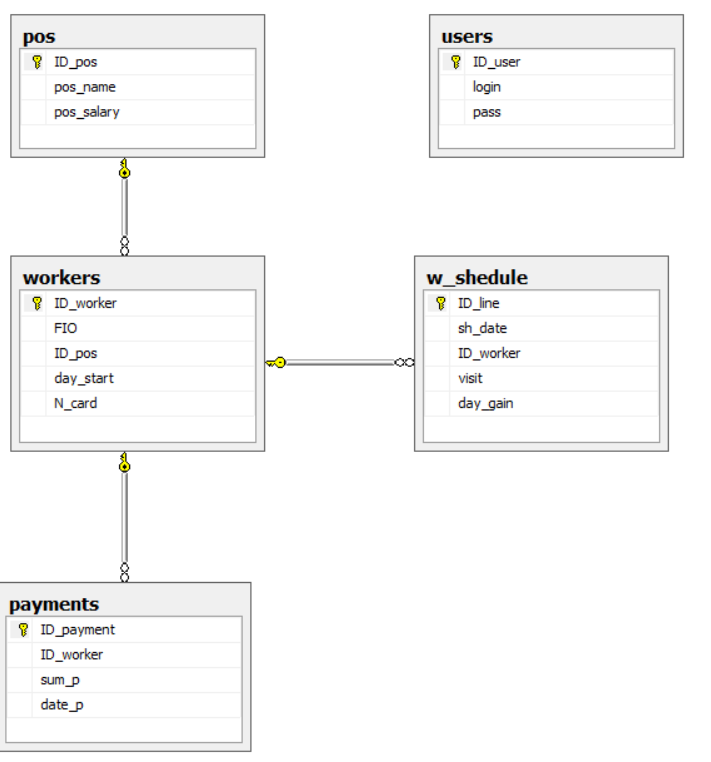
## Формирование сущностей и их атрибутов

Выявим сущности и их атрибуты:

* Сущность «workers» определена на основе класса проектирования «workers». Имеет атрибуты:
* ID\_worker – код сотрудника (int, первичный ключ);
* FIO – имя сотрудника(string, NOT NULL);
* ID\_pos – код должности сотрудника (int, NOT NULL);
* day\_start – дата приема на работу (date, NOT NULL);
* N\_card – номер карты (string, NOT NULL).
* Сущность «payments» определена на основе класса проектирования «payments». Имеет атрибуты:
* ID\_payment - код выплаты (int, первичный ключ);
* ID\_worker - код сотрудника (int, NOT NULL);
* sum\_p – сумма выплаты (float, NOT NULL);
* date\_p – дата выплаты (date, NOT NULL).
* Сущность «w\_shedule» определена на основе класса проектирования «w\_shedule». Имеет атрибуты:
* ID\_line – номер строки (int, первичный ключ);
* sh\_date - дата (date, NOT NULL);
* ID\_worker – код сотрудника (int, NOT NULL);
* visit – посещение (bool, NOT NULL);
* Сущность «pos» определена на основе класса проектирования «pos». Имеет атрибуты:
* ID\_pos – код должности (int, первичный ключ);
* pos\_name – наименование должности (string, NOT NULL);
* pos\_salary – ставка по должности (float, NOT NULL).
* Сущность «workers» определена на основе класса проектирования «workers». Имеет атрибуты:
* ID\_user – код пользователя (int, первичный ключ);
* login – логин пользователя (string, NOT NULL);
* pass – пароль пользователя (string, NOT NULL);

**3.2.1 Построение результирующей ER-диаграммы**

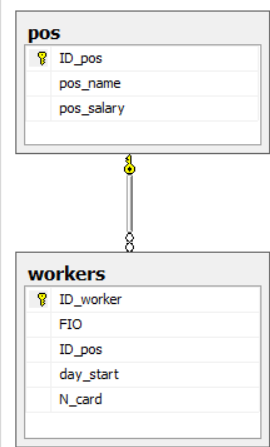
На основе анализа требований к базе данных и к разрабатываемой информационной системе, а также на основании анализа предметной области, построена ER-диаграмма, показанная на рис. 30.

  
Рис. 30. ER-диаграмма

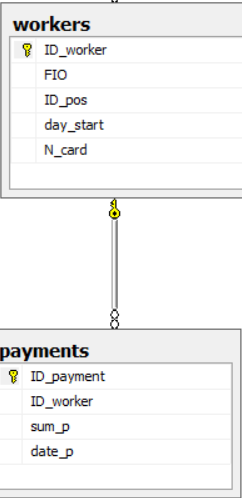
### Выявление связей множеств сущностей и их характеристик

В процессе проектирования были выявлены следующие связи отношений:

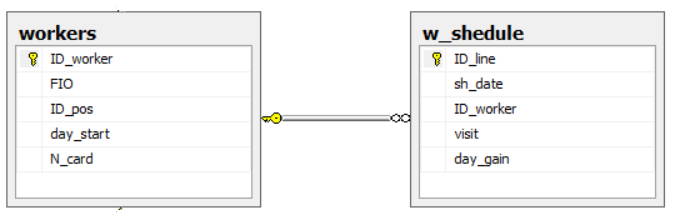
* «pos» и «workers» имеют связь один ко многим, так как одну и ту же должность могут иметь несколько сотрудников.

*  
Рис. 31. Отношение «*pos *-* workers*»*

* «workers» и «payments» имеют связь один ко многим, так как для одного сотрудника может быть произведено множество выплат.

*  
Рис. 32. Отношение «*workers *-* payments*»*

* «workers» и «w\_shedule» имеют связь один ко многим, так как у одного сотрудника может быть множество смен в рабочем графике.

*  
Рис. 33. Отношение «*workers *–* w\_shedule*»*

### Даталогическое проектирование

Для того, чтобы можно было хранить и администрировать базу данных, применяются системы управления базами данных. И именно Microsoft SQL Server является одной из наиболее популярных СУБД в мире, которая подходит для различных проектов: от небольших приложений до многоуровневых высоконагруженных проектов. MS SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

SQL Server характеризуется такими особенностями как:

* Производительность.
* Надежность и безопасность (SQL Server предоставляет шифрование данных).
* Простота в использовании.

Центральным аспектом в MS SQL Server, как и в любой СУБД, является база данных. База данных представляет собой хранилище данных, организованных определенным способом. Под ее управлением могут работать базы данных в широком диапазоне: от уровня небольшого предприятия до распределенных систем масштаба корпорации.

Итак, данный язык структурированных запросов был выбран по следующим причинам: простота словаря языка, высокая производительность обработки запросов, относительно небольшие требования к ПО, дружественность интерфейса для программиста.

Данная база данных будет реализована с помощью Microsoft SQL Server 2008 R2 Management Studio.

Для взаимодействия с данными при помощи моделей будет использоваться подход CodeFirst .

CodeFirst – подход, предполагающий минимальное участие программиста в проектировании сущностей базы данных. Требуется лишь написать программный код, а остальное сделает Entity Framework и Visual Studio. Code First позволяет с минимальными усилиями изменить проект с использованием баз данных в качестве источников данных вместо стандартных коллекций .NET.

Так как классы сущностей были спроектированы в пункте 2.1, выберем вариант создания таблиц БД на основе классов-сущностей.

Для того, чтобы использовать Entity Framework, а также подключить Microsoft SQL Server, для начала потребовалось добавить классы проектирования сущностей в проект.

На основе этапа проектирования (2.1), были реализованы следующие классы (листинги 1-5):

class workers

{

[Key]

public int ID\_worker { get; set; }//код работника

public string FIO { get; set; }//фио

[ForeignKey("pos")]

public int ID\_pos { get; set; }//должность

public DateTime day\_start { get; set; }//дата начала работы

public string N\_card { get; set; }//номер карты

public pos pos { get; set; }

public workers()

{

}

}

*Листинг 1. Модель workers, описывающая сущность Сотрудники*

*Листинг 2. Модель payments, описывающая сущность Выплаты*

class payments

{

[Key]

public int ID\_payment { get; set; }//код выплаты

[ForeignKey("workers")]

public int ID\_worker { get; set; }//фио получателя

public double sum\_p { get; set; }//сумма выплаты

public DateTime date\_p { get; set; }//дата выплаты

public workers workers { get; set; }

public payments()

{

}

}

*Листинг 3. Модель w\_shedule, описывающая сущность Рабочий график*

class w\_shedule

{

[Key]

public int ID\_line { get; set; }//код строки

public DateTime sh\_date { get; set; }//дата смены

[ForeignKey("workers")]

public int ID\_worker { get; set; }//фио сотрудника

public bool visit { get; set; }//посещение

public double day\_gain {get;set;}//выручка за смену

public workers workers { get; set; }

public w\_shedule()

{

}

}

*Листинг 4. Модель pos, описывающая сущность Должности*

class pos

{

[Key]

public int ID\_pos { get; set; }//код должности

public string pos\_name { get; set; }//название должности

public double pos\_salary { get; set; }//ставка по должности

public pos()

{

}

}

*Листинг 5. Модель users, описывающая сущность Пользователи системы*

class users

{

[Key]

public int ID\_user { get; set; }

public string login { get; set; }

public string pass { get; set; }

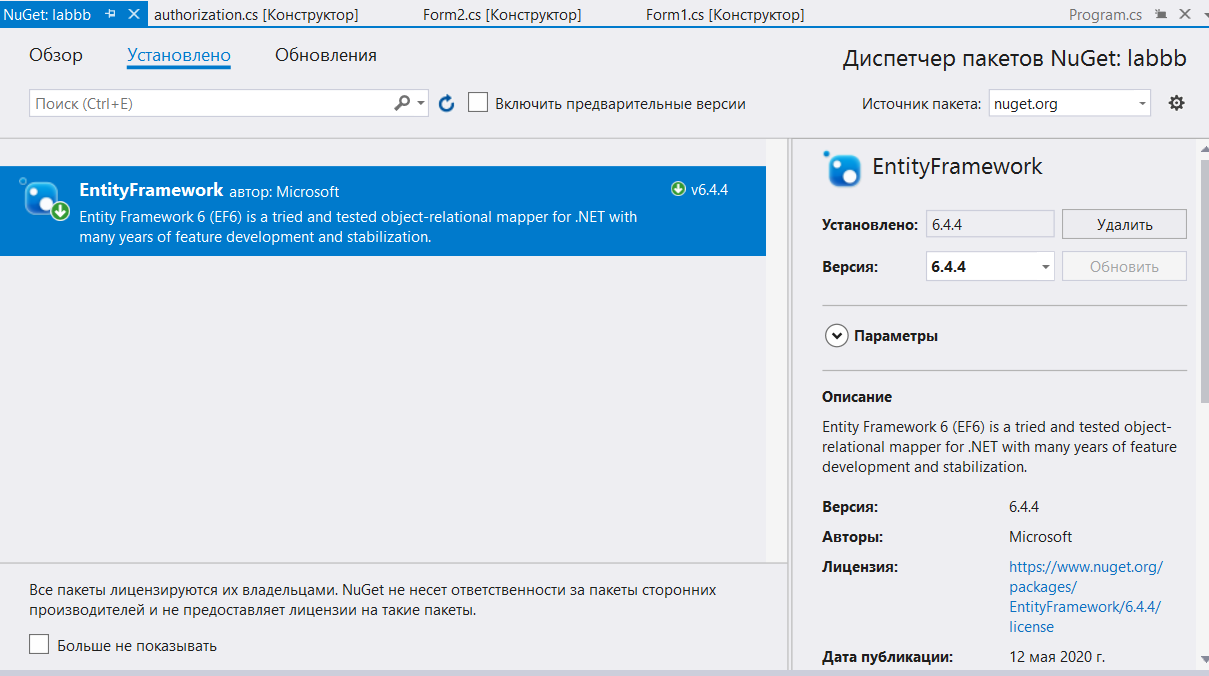
public users()

{

}

}

Далее к проекту нужно добавить пакет Entity Framework:

 *Рис. 34. Подключение пакета Entity Framework*

Далее в проект добавляется класс Context, содержащий данные о соединениях с источниками данных и наборах данных, они и будут представлять собой таблицы базы данных.

*Листинг 6. Класс Context*

class Context : DbContext

{

static Context()

{

Database.SetInitializer<Context>(new MyContextInitializer());

}

public DbSet<workers> workerss { get; set; }

public DbSet<users> userss { get; set; }

public DbSet<payments> paymentss { get; set; }

public DbSet<w\_shedule> w\_shedules { get; set; }

public DbSet<pos> poss { get; set; }

public Context()

: base("Con1")

{ }

}

Следующий шаг – настройка подключения к базе данных. В случае Entity Framework все настройки хранятся в виде строки подключения в файле App.config.

*Листинг 7. Строка подключения к базе данных в разделе App.config*

<connectionStrings>

<add name="Con1"

connectionString="Data Source=msi\sqlexpress;Initial Catalog=test1; Integrated Security=True"

providerName="System.Data.SqlClient" />

</connectionStrings>

Для заполнения базы данных тестовыми данными необходимыми при разработке используется класс MyContextInitializer.

*Листинг 8. Класс MyContextInitializer*

class MyContextInitializer : DropCreateDatabaseIfModelChanges<Context>

{

protected override void Seed(Context db)

{

pos p1 = new pos { pos\_name = "Официант", pos\_salary = 15000 };

pos p2 = new pos { pos\_name = "Администратор", pos\_salary = 40000 };

db.poss.Add(p1);

db.poss.Add(p2);

workers w1 = new workers { FIO = "Борисова Анна Николаевна", ID\_pos = 1, day\_start = new DateTime(2015, 7, 20), N\_card = "1245983781206581" };

db.workerss.Add(w1);

w\_shedule wsh1 = new w\_shedule { sh\_date = new DateTime(2021, 4, 1), ID\_worker = 1, visit = true, day\_gain = 86500 };

w\_shedule wsh2 = new w\_shedule { sh\_date = new DateTime(2021, 4, 2), ID\_worker = 1, visit = true, day\_gain = 12300 };

db.w\_shedules.Add(wsh1);

db.w\_shedules.Add(wsh2);

db.SaveChanges();

}

}

После первого запуска проекта в Microfoft SQL Server автоматически создается база данных, спроектированная по классам моделей (рис.30).Созданные на основе моделей таблицы описаны ниже.

**Таблица workers**

* ID\_worker - идентификатор сотрудника, тип int. Аннотация [Key] означает, что данный атрибут является первичным ключом.
* FIO – имя сотруднка, тип string.
* ID\_pos – код должности сотрудника, тип int. Аннотация [ForeignKey(“pos”)] означает связь с таблицей pos по данному полю.
* day\_start – дата приема сотрудника на работу, тип DateTime.
* N\_card – номер карты сотрудника для перечисления заработной платы, тип string.

**Таблица payments**

* ID\_payment - идентификатор должности, тип int. Аннотация [Key] означает, что данный атрибут является первичным ключом.
* ID\_worker - код сотрудника, тип int. Аннотация [ForeignKey(“workers”)] означает связь с таблицей workers по данному полю.
* sum\_p – сумма выплаты, тип double.
* date\_p – дата выплаты, тип DateTime.

**Таблица w\_shedule**

* ID\_line - идентификатор строки, тип int. Аннотация [Key] означает, что данный атрибут является первичным.
* sh\_date – дата смены, тип DateTime.
* ID\_worker - код сотрудника, тип int. Аннотация [ForeignKey(“workers”)] означает связь с таблицей workers по данному полю.
* visit – посещение смены сотрудником, тип bool.
* day\_gain – выручка за смену, тип double.

**Таблица pos**

1. ID\_pos - идентификатор должности, тип int. Аннотация [Key] означает, что данный атрибут является первичным.
2. pos\_name – наименование должности, тип string.
3. pos\_salary – ставка по должности, тип double.

**Таблица users**

1. ID\_user - идентификатор пользователя, тип int. Аннотация [Key] означает, что данный атрибут является первичным ключом.
2. login – логин, тип string.
3. pass – пароль, тип string.

### Разработка сценариев работы с данными

Сценарии добавления данных:

* Добавление данных сотрудника.

При добавлении нового сотрудника в таблицу вводятся следующие данные: ФИО, должность (из выпадающего списка), дата приема на работу, номер карты. Код сотрудника (ID\_worker) генерируется автоматически (к последнему идентификатору прибавляется единица). При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Добавление данных выплаты.

При добавлении новой выплаты, в таблицу вводятся следующие данные: ФИО сотрудника, которому производится выплата (из выпадающего списка), сумма выплаты, дата выплаты. Код выплаты (ID\_payment) генерируется автоматически (к последнему идентификатору прибавляется единица). При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Добавление данных в рабочий график.

При добавлении новой строки в рабочем графике, в таблице вводятся следующие данные: дата смены, ФИО сотрудника (из выпадающего списка), посещение его смены, выручка за отработанную смену. Код строки (ID\_line) генерируется автоматически (к последнему идентификатору прибавляется единица). При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Добавление данных должности.

При добавлении новой должности, в таблице вводятся следующие данные: наименование должности, ставка по должности. Код должности (ID\_pos) генерируется автоматически (к последнему идентификатору прибавляется единица). При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework

* Добавление данных пользователя системы.

При добавлении нового пользователя системы, в таблице вводятся следующие данные: логин, пароль. Код пользователя (ID\_user) генерируется автоматически (к последнему идентификатору прибавляется единица). При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework

Сценарии изменения данных:

* Изменение данных сотрудника.

При данном сценарии происходит изменение в таблице workers. На странице пользователь выбирает сотрудника из выпадающего списка и заново заполняет его данные. При подтверждении ввода данных, в таблице создается новая запись, а старая удаляется. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Изменение данных выплаты.

При данном сценарии происходит изменение в таблице payments. На странице пользователь вводит новые данные выплаты с указанием ее идентификатора. При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Изменение данных в рабочем графике.

При данном сценарии происходит изменение в таблице w\_shedule. На странице пользователь вводит новые данные стоки графика с указанием ее идентификатора. При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Изменение данных должности.

При данном сценарии происходит изменение в таблице pos. На странице пользователь выбирает должность из списка и вводит для нее новые данные. При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

* Изменение данных пользователя системы.

При данном сценарии происходит изменение в таблице users. На странице пользователь вводит новые данные с указанием идентификатора изменяемой строки. При подтверждении ввода данных, в таблице создается новая запись. Обработка реализована через метод класса C# с использованием технологии Entity Framework.

Сценарии удаления данных:

* Удаление данных сотрудника.

При удалении пациента происходит удаление записи из таблицы workers по его идентификатору (ID\_worker). Обработка реализована через метод класса C# с использованием технологии EntityFramework.

* Удаление данных выплаты.

При удалении выплаты происходит удаление записи из таблицы payments по ее идентификатору (ID\_payment). Обработка реализована через метод класса C# с использованием технологии EntityFramework.

* Удаление данных в рабочем графике.

При удалении смены в рабочем графике происходит удаление записи из таблицы w\_shedule по ее идентификатору (ID\_line). Обработка реализована через метод класса C# с использованием технологии EntityFramework.

* Удаление данных должности.

При удалении должности происходит удаление записи из таблицы pos по ее идентификатору (ID\_pos). Обработка реализована через метод класса C# с использованием технологии EntityFramework.

* Удаление данных пользователя системы.

При удалении пользователя системы происходит удаление записи из таблицы users по его идентификатору (ID\_user). Обработка реализована через метод класса C# с использованием технологии EntityFramework.

### Разработка механизмов реализации сценариев серверной компоненты

Все операции, то есть получение, создание, удаление, обновление данных в проекте реализованы с помощью Entity Framework.

*Таблица 8*

*Методы Entity Framework*

|  |  |  |
| --- | --- | --- |
| **Описание** | **Операция** | **Метод E/F** |
| Добавление  строки в таблицу | Create | Add(новыйЭкземпляр); |
| Удаление строки  из таблицы | Delete | Remove(экземпляр); |
| Редактирование  строки | Update | Entry(экземпляр).State = EntityState.Modified; Экзмепляр.Свойство = Новое значение; |
| Выборка всех  строк таблицы в список на C# | Read | var Lists =  db.НаборДанных.ToList(); |

Для того, чтобы использовать эти методы, необходим экземпляр класса «Context» для взаимодействия с хранилищем данных. После внесения изменений необходимо вызвать метод db.SaveChanges() для сохранения изменений.

Взаимодействие с данными:

* Добавление данных:

*Листинг 9. Добавление сотрудника*

private void button1\_Click(object sender, EventArgs e)//добавить сотрудника

{

workers workers = new workers()

{

FIO = textBox1.Text,

ID\_pos = Convert.ToInt32(comboBox1.SelectedValue),

day\_start = Convert.ToDateTime(dateTimePicker1.Text),

N\_card = textBox4.Text

};

db.workerss.Add(workers);

db.SaveChanges();

dataGridView1.Refresh();

}

*Листинг 10. Добавление должности*

pos pos = new pos()

{

pos\_name = textBox5.Text,

pos\_salary = Convert.ToDouble(textBox6.Text)

};

db.poss.Add(pos);

db.SaveChanges();

dataGridView2.Refresh();

*Листинг 11. Добавление строки в рабочем графике*

w\_shedule w\_shedule = new w\_shedule()

{

sh\_date = Convert.ToDateTime(dateTimePicker2.Text),

ID\_worker = Convert.ToInt32(comboBox2.SelectedValue),

visit = Convert.ToBoolean(checkBox1.Checked),

day\_gain = Convert.ToDouble(textBox10.Text),

};

db.w\_shedules.Add(w\_shedule);

db.SaveChanges();

dataGridView3.Refresh();

*Листинг 12. Добавление сведения о выплате*

payments payments = new payments()

{

ID\_worker = Convert.ToInt32(comboBox3.SelectedValue),

sum\_p = Convert.ToDouble(textBox12.Text),

date\_p = Convert.ToDateTime(dateTimePicker3.Text)

};

db.paymentss.Add(payments);

db.SaveChanges();

dataGridView4.Refresh();

*Листинг 13. Добавление пользователя системы*

users users = new users()

{

login = textBox14.Text,

pass = textBox15.Text

};

db.userss.Add(users);

db.SaveChanges();

dataGridView5.Refresh();

* Изменение данных (изменение данных в кортеже влечет изменение данных в связанных с ним кортежах):

*Листинг 14. Изменение данных о сотруднике*

private void button11\_Click(object sender, EventArgs e)//изменить сотрудника

{

workers worker1 = db.workerss.Find(Convert.ToInt32(textBox11.Text));

db.Entry(worker1).State = EntityState.Modified;

worker1.FIO = textBox1.Text;

worker1.ID\_pos = Convert.ToInt32(comboBox1.SelectedValue);

worker1.day\_start = Convert.ToDateTime(dateTimePicker1.Text);

worker1.N\_card = textBox4.Text;

db.SaveChanges();

dataGridView1.Refresh();

}

*Листинг 15. Изменение должности*

pos pos1 = db.poss.Find(Convert.ToInt32(textBox13.Text));

db.Entry(pos1).State = EntityState.Modified;

pos1.pos\_name = textBox5.Text;

pos1.pos\_salary = Convert.ToDouble(textBox6.Text);

db.SaveChanges();

dataGridView2.Refresh();

*Листинг 16. Изменение строки в рабочем графике*

w\_shedule sh1 = db.w\_shedules.Find(Convert.ToInt32(textBox9.Text));

db.Entry(sh1).State = EntityState.Modified;

sh1.sh\_date = Convert.ToDateTime(dateTimePicker2.Text);

sh1.ID\_worker = Convert.ToInt32(comboBox2.SelectedValue);

sh1.visit = Convert.ToBoolean(checkBox1.Checked);

sh1.day\_gain = Convert.ToDouble(textBox10.Text);

db.SaveChanges();

dataGridView3.Refresh();

*Листинг 17. Изменение сведения о выплате*

payments p1 = db.paymentss.Find(Convert.ToInt32(textBox16.Text));

db.Entry(p1).State = EntityState.Modified;

p1.ID\_worker = Convert.ToInt32(comboBox3.SelectedValue);

p1.sum\_p = Convert.ToDouble(textBox12.Text);

p1.date\_p = Convert.ToDateTime(dateTimePicker3.Text);

db.SaveChanges();

dataGridView4.Refresh();

*Листинг 18. Изменение пользователя системы*

users u1 = db.userss.Find(Convert.ToInt32(textBox8.Text));

db.Entry(u1).State = EntityState.Modified;

u1.login = textBox14.Text;

u1.pass = textBox15.Text;

db.SaveChanges();

dataGridView5.Refresh();

* Удаление данных (удаление кортежа данных влечет удаление связанных с ним кортежей):

*Листинг 19. Удаление данных о сотруднике*

private void button12\_Click(object sender, EventArgs e)//удалить сотр.

{

workers w1 = db.workerss.Find(Convert.ToInt32(textBox11.Text));

db.workerss.Remove(w1);

db.SaveChanges();

dataGridView1.Refresh();

}

*Листинг 20. Удаление должности*

pos p1 = db.poss.Find(Convert.ToInt32(textBox13.Text));

db.poss.Remove(p1);

db.SaveChanges();

dataGridView2.Refresh();

*Листинг 21. Удаление строки в рабочем графике*

w\_shedule sh1 = db.w\_shedules.Find(Convert.ToInt32(textBox9.Text));

db.w\_shedules.Remove(sh1);

db.SaveChanges();

dataGridView3.Refresh();

*Листинг 22. Удаление сведения о выплате*

payments pay1 = db.paymentss.Find(Convert.ToInt32(textBox16.Text));

db.paymentss.Remove(pay1);

db.SaveChanges();

dataGridView4.Refresh();

*Листинг 23. Удаление пользователя системы*

users u1 = db.userss.Find(Convert.ToInt32(textBox8.Text));

db.userss.Remove(u1);

db.SaveChanges();

dataGridView5.Refresh();

* Сценарий авторизации:

*Листинг 24. Авторизац*ия

public partial class authorization : Form

{

Context db;

public authorization()

{

InitializeComponent();

db = new Context();

db.userss.Load();

}

private void button1\_Click(object sender, EventArgs e)

{

var Lists = db.userss.ToList();

users us1 = new users { login = textBox1.Text, pass = textBox2.Text };

if (Lists.Contains(us1))

{

Form1 f1 = new Form1();

f1.Show();

}

else

{

MessageBox.Show("Пользователь не существует");

}

}

}

# 

# Специальная часть. Разработка веб-приложения на платформе ASP.NET с использованием паттерна MVC

В дальнейшем при использовании автоматизированной системы расчета заработной платы планируется перенос ее в формат веб-приложения, поэтому далее будет рассмотрена технология для реализации данного решения, встроенная в среду разработки Visual Studio.

## Описание технологии

В качестве специальной части была выбрана технология разработки веб-приложений ASP.NET, выпущенная компанией Microsoft и подходящая для множества различных языков программирования.

Данная технология входит в состав платформы [.NET Framework](https://ru.wikipedia.org/wiki/Microsoft_.NET), содержит множество готовых элементов управления, что значительно упрощает создание интерактивных web-сайтов.

На данный момент эта технология популярна в сфере разработки внутрикорпоративных программных продуктов.

В ASP.NET поддерживается несколько программных моделей для создания веб-приложений:

* [ASP.NET Web Forms](https://ru.wikipedia.org/w/index.php?title=ASP.NET_Web_Forms&action=edit&redlink=1) (самая старая модель для создания модульных веб-страниц с управляемыми событиями);
* [ASP.NET Web Pages](https://ru.wikipedia.org/w/index.php?title=ASP.NET_Web_Pages&action=edit&redlink=1) (веб-страницы, позволяющие вставлять динамическое содержимое прямо в HTML-разметку);
* [ASP.NET MVC](https://ru.wikipedia.org/wiki/ASP.NET_MVC_Framework) (фреймворк для создания веб-страниц с использованием шаблона проектирования [MVC](https://ru.wikipedia.org/wiki/Model-View-Controller)).

ASP.NET использует схему MVC (Модель-Вид-Контроллер). Каждая из этих частей отвечает за определенные действия:

* Контроллер (controller) представляет собой класс, который обеспечивает связь между пользователем и системой, представлением и хранилищем данных. Контроллер получает данные, вводимые пользователем, и

обрабатывает их, а в зависимости от результатов обработки отправляет пользователю определенный вывод.

* Представление (view) — это визуальная часть, пользовательский интерфейс приложения. Как правило, это html-страница, которую пользователь видит, зайдя на сайт.
* Модель (model) – это класс, описывающий логику используемых данных. Она является независимым компонентом, то есть какие-либо изменения контроллера или представления не затрагивают модель.

Веб-приложение на ASP.NET состоит из следующих основных файлов и каталогов:

**App\_Data** – здесь хранятся XML-файлы или базы данных;

**App\_Start** – папка, содержащая ряд основных настроек конфигурации для проекта, в том числе определение маршрутов и фильтров, а также пакетов содержимого;

**Content** – CSS-файлы и изображения;

**Controllers** – классы контроллеров;

**Models** – классы моделей;

**Scripts**–  [JavaScript](https://professorweb.ru/my/javascript/js_theory/level1/javascript_index.php)-файлы;

**Views** – в этой папке хранятся представления, обычно сгруппированные по папкам с именами контроллеров, с которыми они связаны;

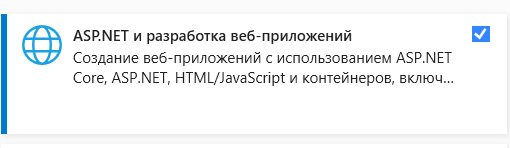
**Views/Web.config** – это конфигурационный файл. Содержит в себе конфигурационную информацию, обеспечивающую обработку представлений;

**Global.asax** – это глобальный класс приложения ASP.NET. В его файле отделенного кода (Global.asax.cs) регистрируется конфигурация маршрутов, а также предоставляется любой код, который должен выполняться при запуске или завершении приложения либо в случае возникновения необработанного исключения;

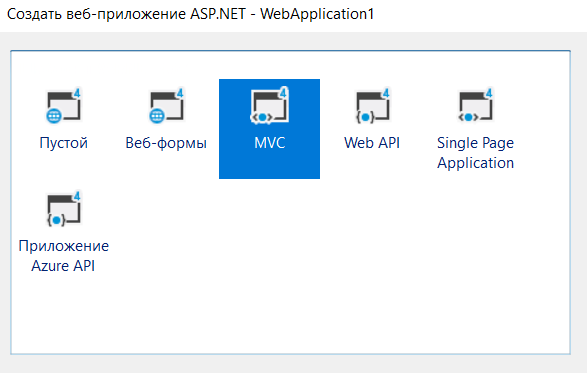
**Web.config** – конфигурационный файл для приложения.

## Реализация

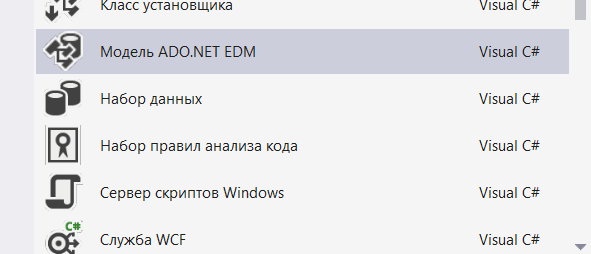
В качестве примера разрабатываемого решения будет приведена реализация выкладки приложения «Должности». Для начала требуется создать проект ASP.NET WebApp, предварительно установив расширение “ASP.NET и разработка веб-приложений” (рис. 35):

*  
Рис. 35. Установка расширения*

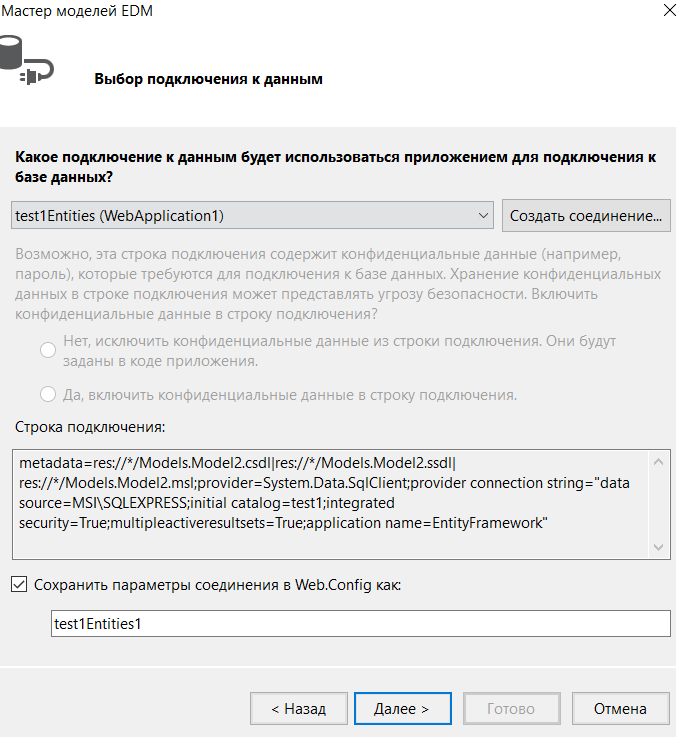
В качестве шаблона проекта выбираем MVC.

*  
Рис. 36. Шаблоны проекта*

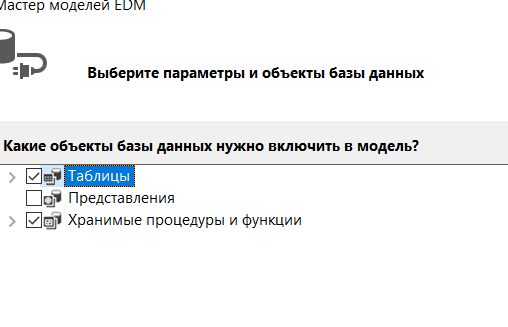
Для начала в папке Models создаем элемент “Модель ADO.NET EDM”, содержание модели – конструктор EF из базы данных.

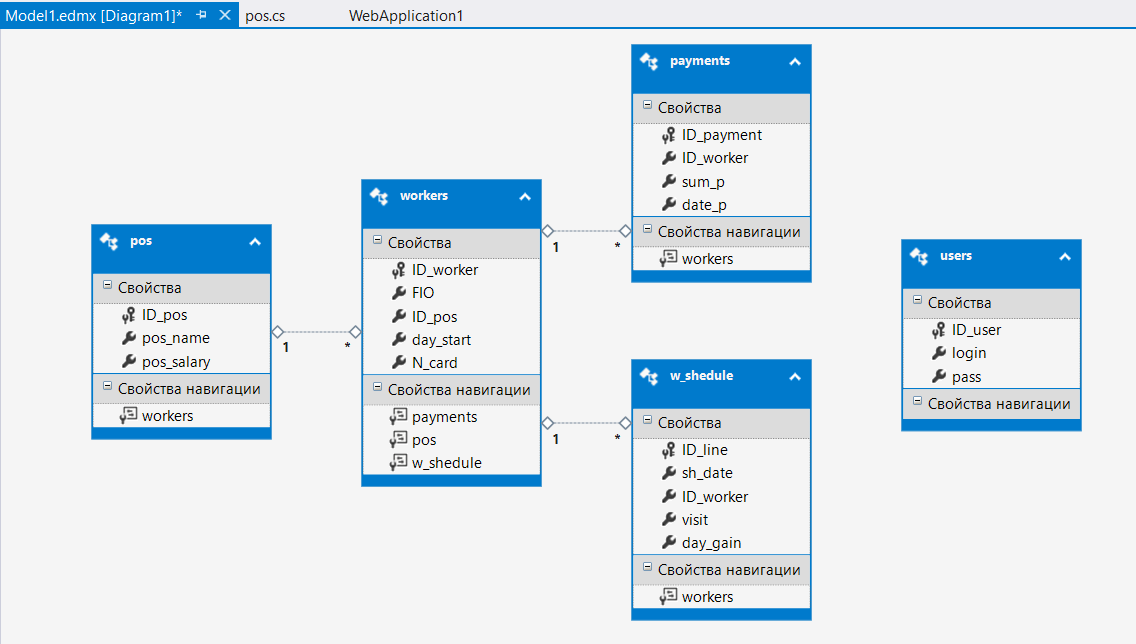
*  
Рис. 37. Создание модели ADO.NET*

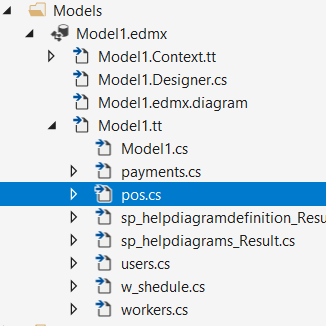
Далее выбираем подключение или выбираем существующее из списка. Выбор происходит по имени базы данных в системе MS SQL Server.

*  
Рис. 38. Создание подключения*

После чего выбираем элементы базы данных, которые необходимо добавить в модель, запускаем создание модели.

*  
Рис. 39. Выбор объектов*

*  
Рис. 40. Результат подключения к базе данных*

*  
Рис. 41. Компоненты модели*

Далее в каталоге Controllers был создан контроллер, отвечающий за реализацию выборки должностей из базы данных, а также все необходимые элементы для его работы.

class Context : DbContext

{

static Context()

{

Database.SetInitializer<Context>(new MyContextInitializer());

}

public DbSet<pos> poss { get; set; }

public Context()

: base("Con1")

{ }

}

class MyContextInitializer : DropCreateDatabaseIfModelChanges<Context>

{

protected override void Seed(Context db)

{

pos p1 = new pos { pos\_name = "Официант", pos\_salary = 15000 };

pos p2 = new pos { pos\_name = "Администратор", pos\_salary = 40000 };

db.poss.Add(p1);

db.poss.Add(p2);

db.SaveChanges();

}

}

class pos

{

[Key]

public int ID\_pos { get; set; }//код должности

public string pos\_name { get; set; }//название должности

public double pos\_salary { get; set; }//ставка по должности

public pos()

{

}

}

public class posController : Controller

{

// GET: pos

Context db = new Context();

[HttpGet]

public ActionResult GetPos()

{

var ppos = db.poss.ToList();

db.SaveChanges();

return View(ppos);

}

}

*Листинг 26. Класс posController*

*Листинг 27. ViewStart представление*

@{

Layout = "~/Views/Shared/\_Layout.cshtml";

}

*Листинг 28. Layout.cshtml.*

@model IEnumerable<WebApplication1.Models.pos>

@using WebApplication1.Models

@{

ViewBag.Title = "Должности";

}

<link href="~/Content/Site.css" rel="stylesheet" />

<div>

<h2>Заполните данные</h2>

@using (Html.BeginForm("GetPos", "ppos", FormMethod.Post))

{<div class="row mt">

<div class="col-md-3">

<p>Введите наименование: </p>

</div>

<div class="col-md-3">

<input class="inp" type="text" name="pos\_name" required />

</div>

</div>

<div class="row">

<div class="col-md-3">

<p>Введите ставку: </p>

</div>

<div class="col-md-3">

<input class="inp" type="number" name="pos\_salary" required />

</div>

</div>

<div class="row">

<div class="col-md-3">

<input class="btn btn-primary" type="submit" value="Добавить" />

</div>

</div>

}

<div>

<table>

<tr class="nametab">

<th><p>Должность</p></th>

<th><p>Ставка</p></th>

</tr>

@foreach (pos pos in Model)

{

<tr>

<td><p>@pos.pos\_name</p></td>

<td><p>@pos.pos\_salary</p></td>

</tr>

}

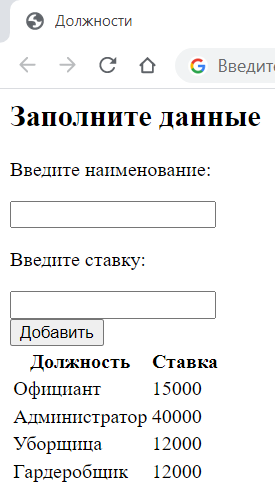
</table>

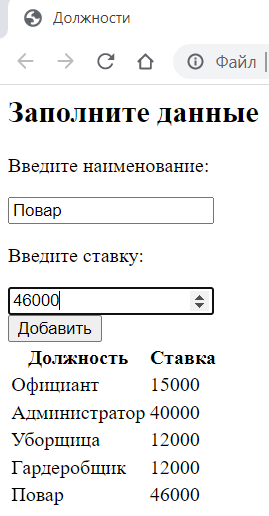
</div>

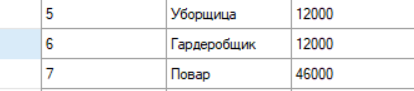
</div>

Содержание таблицы до запуска программы:

*  
Рис. 42. Содержание таблицы «Должности»*

*  
Рис. 43. Вкладка «Должности»*

*  
Рис. 44. Добавление данных*

*  
Рис. 45. Содержание таблицы после выполнения программы*.

# Технология разработки и программная реализация.

## Выбор технологии

## Выбор операционной системы.

В качестве операционной системы для развертывания ИС была выбрана операционная система Windows 10. Выбор данной операционной системы обусловлен тем, что это высокоскоростная, защищенная и надежная система, используемая в настоящее время на предприятии. Смена данной ОС будет нецелесообразной.

## Выбор взаимодействия пользователя с операционной системой.

Для взаимодействия пользователя с системой был выбран интерфейс «Приложение в стиле операционный системы». Данный интерфейс обусловлен интуитивной понятностью при использовании.

## Выбор языка и среды программирования.

В качестве среды разработки была выбрана среда разработки Visual Studio 2017 Community, в качестве основного языка программирования – C#. Данные среда и язык разработки являются наиболее подходящими для решения поставленной задачи по созданию информационной системы и для работы предприятия в целом.

## Выбор технологии взаимодействия пользовательских компонент с данными.

В качестве технологии взаимодействия пользовательских компонент с данными выбрана технология ADO.NET Entity Framework - модель доступа к данным для приложений, основанных на Microsoft .NET. Выбор технологии основан на выборе операционной системы.

## Определение физической архитектуры приложения

* + 1. **Определение состава компонент**

При формировании компонент рекомендуемо ориентироваться по трем логическим сервисным уровням:

* пользовательскому;
* прикладному;
* уровню данных.

**Пользовательский уровень** — отображает данные и позволяет пользователю редактировать их.

**Прикладной уровень** – реализация бизнес-правил и ограничений на данные. Сервисы этого уровня используются пользовательским уровнем, но при этом он не привязан к какому-либо клиенту, его сервисы доступны любому клиенту. Бизнес-правила выражаются в форме прикладных алгоритмов, корпоративных правил и т. д., обычно реализуются отдельным модулем на централизованном сервере, что дает возможность доступа к нему сразу нескольким клиентам.

**Уровень данных** — прикладной уровень не знает, как и где хранится обрабатываемая им информация. В этом вопросе он полагается на сервисы доступа к данным, выполняющие всю работу по получению и передаче данных. Сервисы доступа к данным также реализуются в виде изолированных модулей, «знающих» о месте хранения информации. Таким образом, если хранилище перемещено или изменен его формат, потребуется обновить только сервисы доступа к данным. Каждый модуль доступа к данным, как правило, отвечает и за целостность хранилища (например, реляционной базы данных). Для многоуровневых приложений в качестве хранилища информации подходят простые СУБД, необходимые для обслуживания данных в таблицах и быстрой выборки информации.

Разрабатываемая информационная система будет построена по двухуровневой технологии – пользовательский уровень и уровень данных.

* + 1. **Разработка компонент**

В качестве клиента используется «тонкий» клиент, т.е. используется сервер приложений. Вся бизнес-логика перенесена на сторону сервера.

Каждая форма представляет собой одно логически завершённое действие, т.е. одна форма предназначена для одного конкретного действия.

Для быстрого доступа ко всем функциям клиентского приложения используется основная форма.

Для повышения простоты и удобства использования, все поля, списки и кнопки подписаны без использования сокращений.

Клиентское приложение имеет следующие функции:

* заполнение данных сотрудников, должностей, рабочего графика, выплат, пользователей системы;
* изменение и удаление данных в таблицах.
* Создание выплаты.

В системе выделяются следующие функции:

* *Заполнение данных о сотрудниках, должностях, рабочем графике, выплатах, пользователях системы* (Доступно администратору).

Подразумевает: Ввод данных о сотрудниках, должностях, рабочем графике, выплатах, пользователях системы.

Номера объектам выдаются автоматически.

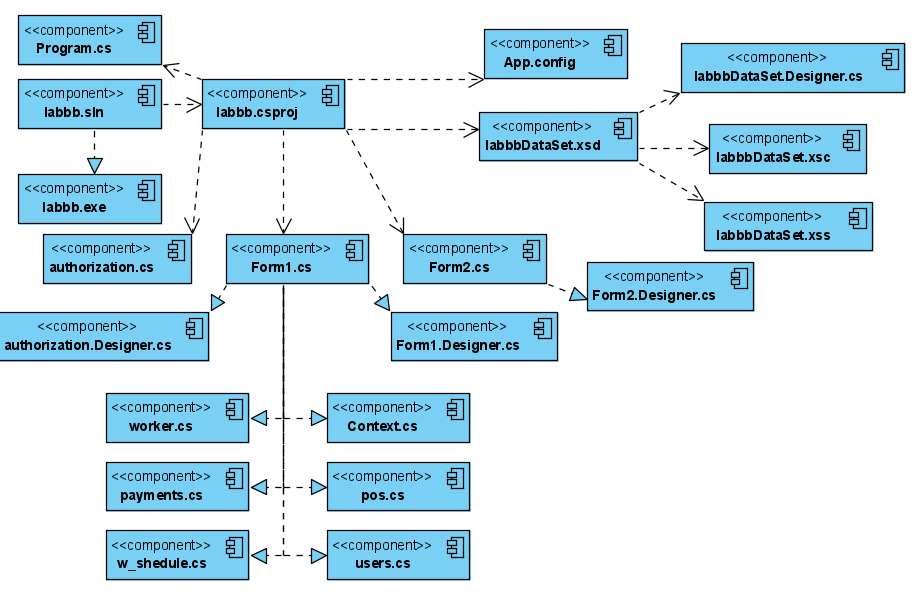
* *Редактирование и удаление данных* (Доступно администратору).

Подразумевает: Изменение и удаление данных в таблицах.

Изначальные данные о редактируемом объекте загружаются в системные поля автоматически.

* *Создание выплаты* (Доступно администартору).

Подразумевает: создание новой выплаты в таблице с использованием специальной формы.

*  
Рис. 46. Диаграмма компонентов системы*

* + 1. **Уточнение состава экранных форм. Определение конкретных типов управляющих элементов для форм**

Клиентская программа содержит следующие формы:

* Форма авторизации;
* Основная форма;
* Форма создания выплаты;

**Основная форма** – элемент Form1.cs, служит для редактирования таблиц.

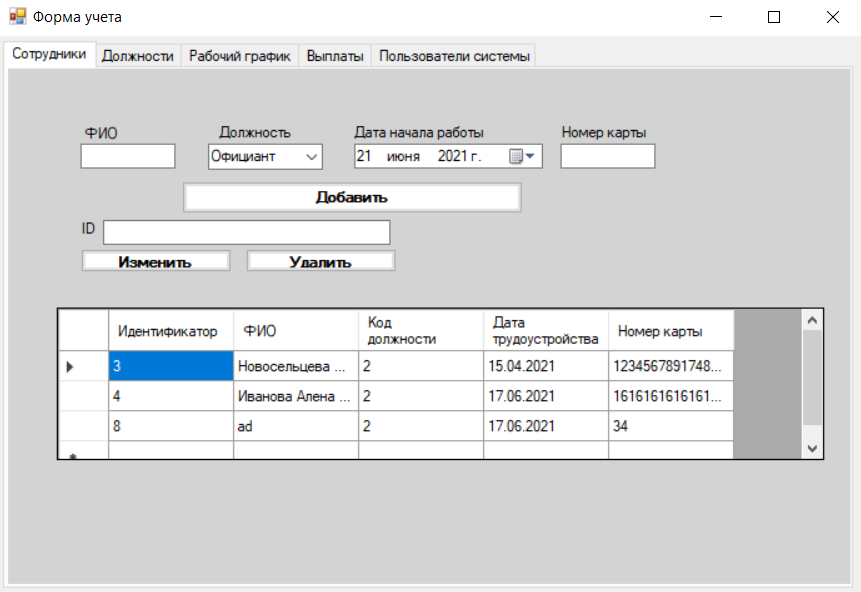
Элементы управления:

* tabCon – TabControl, созержащий в себе пять вкладок (workers\_f, pos\_f, sh\_f, pays\_f, users\_f), каждая из которых ниже будет рассмотрена более подробно.

Вкладка **workers\_f** – служит для добавления, удаления и изменения информации о сотрудниках.

Элементы управления:

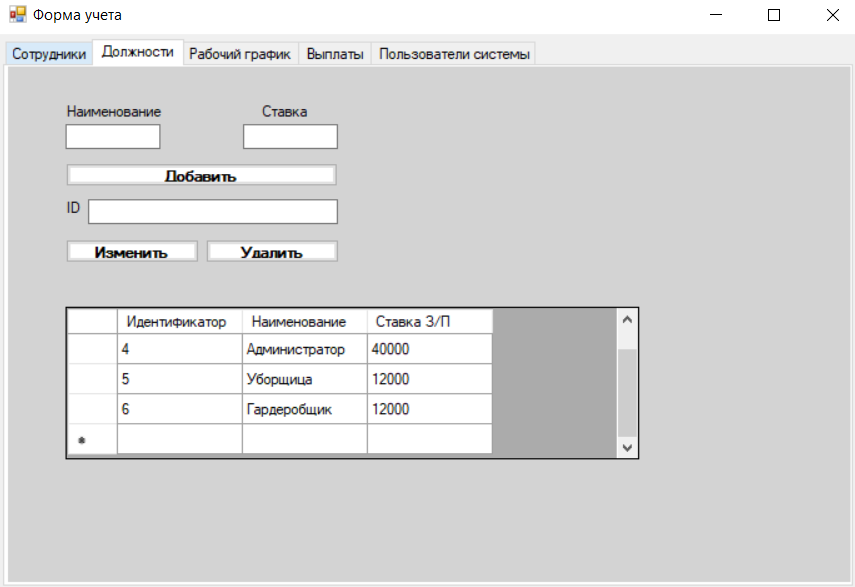
* button1, button11, button12 – кнопки, отвечающие за добавление, изменение и изменение класса «Сотрудики» соответственно.
* textbox1, textbox4, textbox11 – поля ввода ФИО, номера карты и индентификатора сотрудника, а также соответствующие им label1, label4, label 18 – подписи полей (ФИО, Номер карты, ID).
* comboBox1 – выпадающий список для выбора должности сотрудника, а таже соответствующий ему label2 – подпись (Должность).
* dateTimePicker1 – поле ввода даты, а также соответствующий ему label3 – подпись (Дата начала работы).
* dataGridView1 – таблица с данными о сотрудниках.

*  
Рис. 47. Вкладка «Сотрудники»*

Вкладка **pos\_f** – служит для добавления, удаления и изменения информации о должностях, имеющихся на предприятии.

Элементы управления:

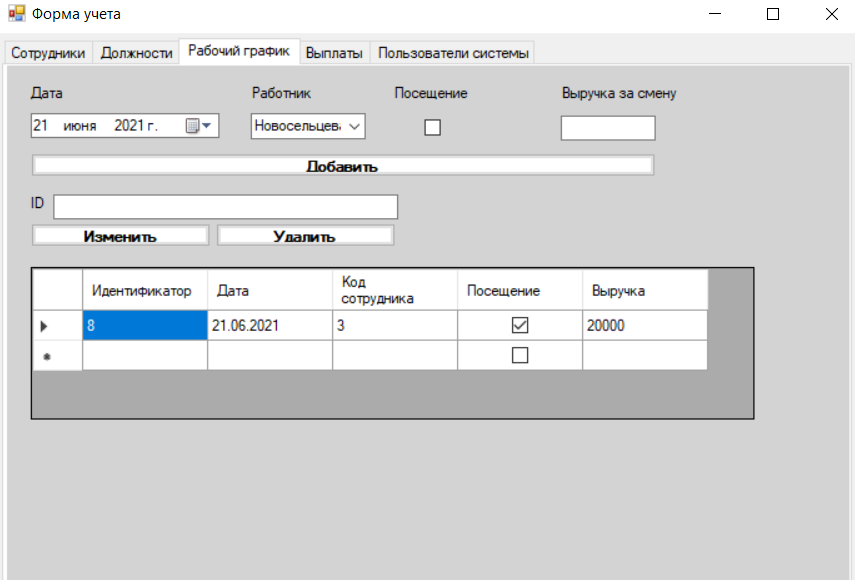
* button2, button13, button14 – кнопки, отвечающие за добавление, изменение и изменение класса «Должности» соответственно.
* textbox5, textbox6 – поля ввода наименования должности и ставки заработной платы, а также соответствующие им label5, label6 – подписи полей (Наименование, Ставка).
* dataGridView2 – таблица с данными о должностях.

  
*Рис. 48. Вкладка «Сотрудники»*

Вкладка **sh\_f** – служит для добавления, удаления и изменения информации о рабочем графике.

Элементы управления:

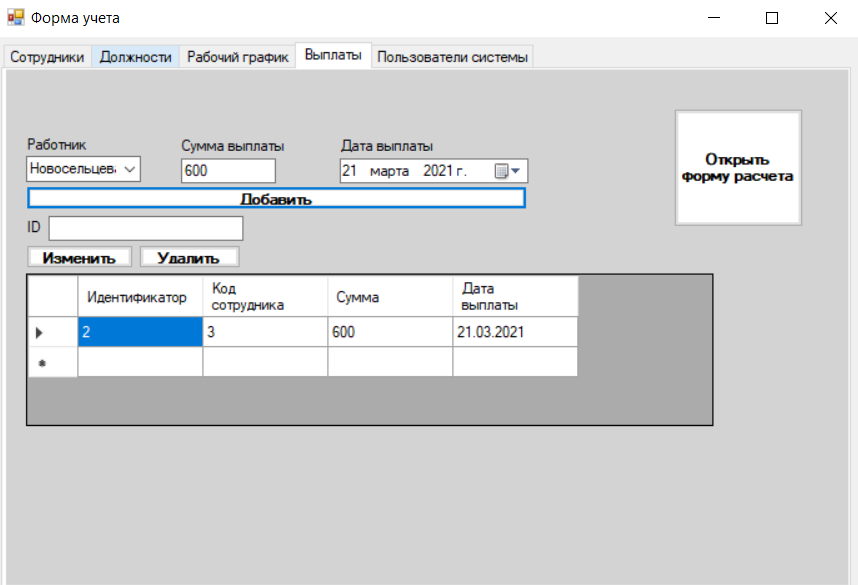
* button3, button9, button10 – кнопки, отвечающие за добавление, изменение и изменение класса «Рабочий график» соответственно.
* textbox10 – поле ввода выручки сотрудника за отработанную смену, а также соответствующий ему label10 – подпись поля (Выручка за смену).
* comboBox2 – выпадающий список для выбора сотрудника, а таже соответствующий ему label8 – подпись (Работник).
* dateTimePicker2 – поле ввода даты смены, а также соответствующий ему label7 – подпись (Дата).
* checkBox1 – флажок, а также соответствующий ему label9 – подпись (Посещение). Флажок ставится, если у сотрудника запланирована смена и он ее посетил, и не ставится, если смена запланирована, но сотрудник не пришел на работу.
* dataGridView3 – таблица с данными о рабочем графике.

*  
Рис. 49. Вкладка «Рабочий график»*

Вкладка **pays\_f** – служит для добавления, удаления и изменения информации о рабочем графике.

Элементы управления:

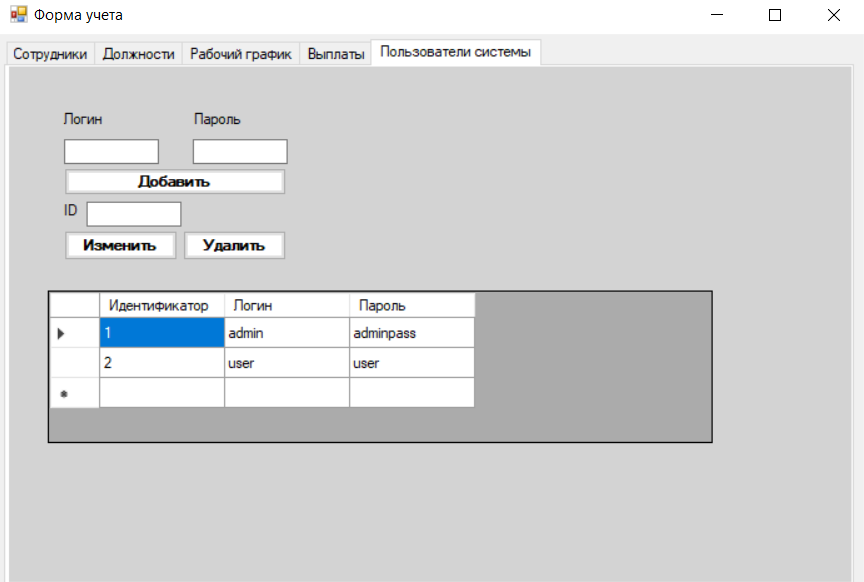
* button4, button15, button16 – кнопки, отвечающие за добавление, изменение и изменение класса «Выплаты» соответственно.
* button6 – кнопка для открытия Form2.
* textbox12 – поле ввода суммы выплаты сотруднику, а также соответствующий ему label12 – подпись поля (Сумма выплаты).
* comboBox3 – выпадающий список для выбора сотрудника, а таже соответствующий ему label11 – подпись (Работник).
* dateTimePicker3 – поле ввода даты выплаты, а также соответствующий ему label13 – подпись (Дата).
* dataGridView4 – таблица с данными о выплатах.

*  
Рис. 50. Вкладка «Выплаты»*

Вкладка **users\_f** – служит для добавления, удаления и изменения информации о рабочем графике.

Элементы управления:

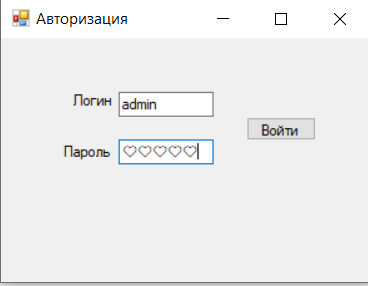
* button5, button7, button8 – кнопки, отвечающие за добавление, изменение и изменение класса «Пользователи системы» соответственно.
* textbox14, textbox15 – поля ввода логина и пароля, а также соответствующие им label14, label15 – подписи полей (Логин, пароль).
* dataGridView5 – таблица с данными о пользователях системы.

*Рис. 51. Вкладка «Пользователи системы»*

**Форма авторизации** – элемент authorization.cs, служит для входа в приложение.

Элементы управления:

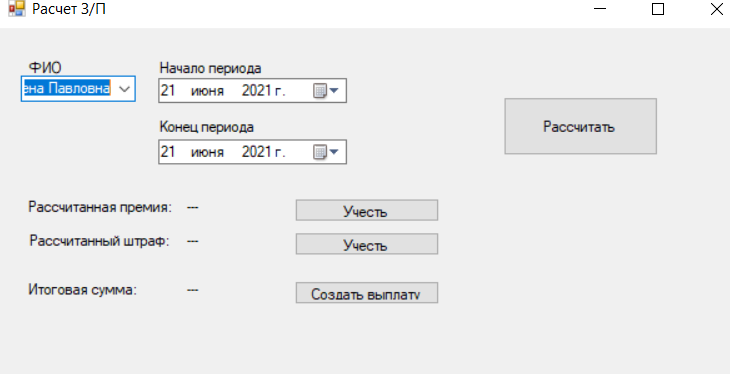
* button1 – кнопка для выполнения входа в приложение.
* textbox1, textbox2 – поля ввода логина и пароля, а также соответствующие им label1, label2 – подписи полей (Логин, Пароль).

*  
Рис. 52. Форма авторизации*

**Форма расчета заработной платы** – элемент Form2.cs, служит для расчета заработной платы и записи ее в таблицу выплат.

Элементы управления:

* button1, button2, button3, button4 – кнопки для выполнения расчета, а также учета премий, штрафов и создания выплаты.
* label4, label5, label8 – надписи «Расчитанная премия», «Расчитанный штраф», «Итоговая сумма».
* label6, label7, label9 – надписи для отображения расчитанных сумм.
* comboBox1 – выпадающий список для выбора сотрудника, чья заработная плата будет рассчитываться, а также соответствующий ему label1 – подпись поля (ФИО).
* dateTimePicker1, dateTimePicker2 – поля ввода даты начала и окончания периода расчета заработной платы, а также соответствущие им label2 и label3 – подписи полей (Начало периода, Конец периода).

*  
Рис. 53. Форма расчета заработной платы*

* + 1. **Определение технологии доступа к компоненте данных**

Основными компонентами работы с данными являются:

* Клиентская программа;
* Хранилище данных;
* Программное обеспечение Microsoft SQL Server.

****** *Рис. 50. Компоненты работы с данными*

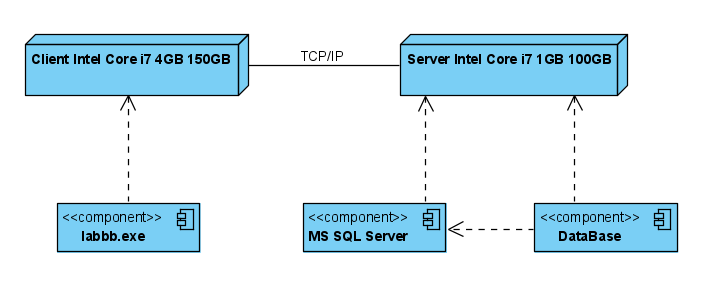


Рис. 51. Диаграмма компонент

* 1. **Тестирование**

Все требования, сформированные к разрабатываемой системе, должны быть подтверждены тестированием.

1. Обеспечение возможности добавления, удаления и изменения данных в таблицах «Сотрудники», «Должности», «Выплаты», «Рабочий график», «Пользователи системы».

Внесение данных осуществляется с помощью ввода данных в соответствующие поля приложения, за исключением поля ID (создается автоматически).

Тестирование данного сценария осуществляется путем:

* Ввода некорректных (символьных) значений в поля, которые должны быть заполнены числовыми значениями.
* Ввода нулевых значений в поля, которые должны быть заполнены.

Изменение данных осуществляется с помощью ввода данных в соответствующие поля приложения, включая поле-индентификатор (ID).

Тестирование данного сценария осуществляется путем:

* Ввода некорректных (символьных) значений в поля, которые должны быть заполнены числовыми значениями.
* Ввода нулевых значений в поля, которые должны быть заполнены.

Удаление данных осуществляется по введенному значению в поле-идентификатор (ID).

Тестирование данного сценария осуществляется путем:

* Ввода некорректных значений в поле, которое должно быть заполнено.

1. Обеспечение корректности подсчета заработной платы.

Подсчет осуществляется в базе данных, данные передаются и просчитываются, данные должны иметь верный формат для их счета.

Тестирование данного сценария осуществляется путем:

* Невозможность ввода некорректных (символьных) значений в поля, которые должны быть заполнены числовыми значениями.
* Ввода нулевых значений (отсутствие ввода в поля), которые должны быть заполнены.

Примеры тестов:

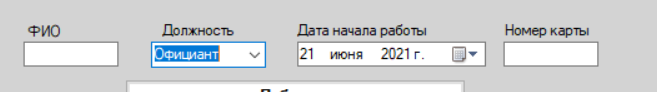
*Таблица 9*

*Тест-кейсы*

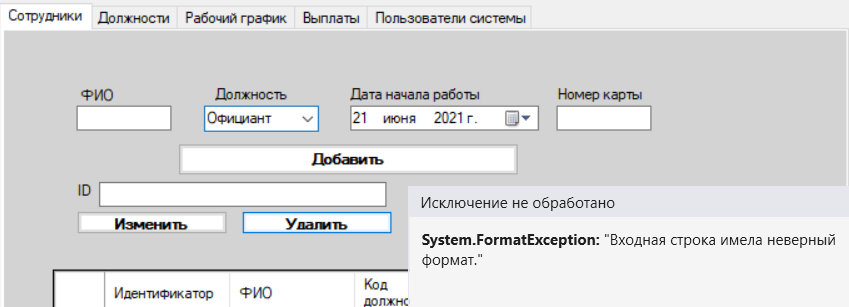
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест | Позитивный/негативный тест | Страница | Ожидаемый результат | Пройден/нет |
| 1 | Внесение корректных данных сотрудника | + | Сотрудники | Данные сотрудника внесены | + |
| 2 | Внесение некорректных данных сотрудника | - | Сотрудники | Ошибка | + |
| 3 | Изменение данных на корректные | + | Сотрудники | Данные сотрудника изменены | + |
| 4 | Изменение данных на некорректные | - | Сотрудники | Ошибка | + |
| 5 | Удаление по идентификатору | + | Сотрудники | Данные удалены | + |
| 6 | Удаление без указания идентификатора | - | Сотрудники | Ошибка | + |
| 7 | Внесение корректных данных должности | + | Должности | Данные должности внесены | + |
| 8 | Внесение некорректных данных должности (текст вместо ставки з/п) | - | Должности | Ошибка | + |
| 9 | Изменение данных на корректные | + | Должности | Данные сотрудника изменены | + |
| 10 | Изменение данных на некорректные | - | Должности | Ошибка | + |
| 11 | Удаление по идентификатору | + | Должности | Данные удалены | + |
| 12 | Удаление без указания идентификатора | - | Должности | Ошибка | + |
| 13 | Внесение корректных данных должности | + | Рабочий график | Данные должности внесены | + |
| 14 | Внесение некорректных данных должности (текст вместо выручки) | - | Рабочий график | Ошибка | + |
| 15 | Изменение данных на корректные | + | Рабочий график | Данные сотрудника изменены | + |
| 16 | Изменение данных на некорректные | - | Рабочий график | Ошибка | + |
| 17 | Удаление по идентификатору | + | Рабочий график | Данные удалены | + |
| 18 | Удаление без указания идентификатора | - | Рабочий график | Ошибка | + |
| 19 | Внесение корректных данных выплаты | + | Выплаты | Данные должности внесены | + |
| 20 | Внесение некорректных данных выплаты (текст вместо суммы выплаты) | - | Выплаты | Ошибка | + |
| 21 | Изменение данных на корректные | + | Выплаты | Данные сотрудника изменены | + |
| 22 | Изменение данных на некорректные | - | Выплаты | Ошибка | + |
| 23 | Удаление по идентификатору | + | Выплаты | Данные удалены | + |
| 24 | Удаление без указания идентификатора | - | Выплаты | Ошибка | + |
| 25 | Внесение корректных данных пользователя | + | Пользовате-ли системы | Данные должности внесены | + |
| 26 | Внесение некорректных данных пользователя | - | Пользовате-ли системы | Ошибка | + |
| 27 | Изменение данных на корректные | + | Пользовате-ли системы | Данные сотрудника изменены | + |
| 28 | Изменение данных на некорректные | - | Пользовате-ли системы | Ошибка | + |
| 29 | Удаление по идентификатору | + | Пользовате-ли системы | Данные удалены | + |
| 30 | Удаление без указания идентификатора | - | Пользовате-ли системы | Ошибка | + |
| 31 | Авторизация существующего пользователя | + | Авторизация | Открытие Form1 | + |
| 32 | Авторизация несуществующего пользователя | - | Авторизация | Ошибка | + |
| 33 | Расчет з/п с введением корректных данных | + | Форма расчета заработной платы | Выполнение расчета | + |
| 34 | Расчет з/п с введением некорректных данных | - | Форма расчета заработной платы | Ошибка | + |

Подтверждение прохождения негативных тестов представлены ниже на рисунках 52-57.

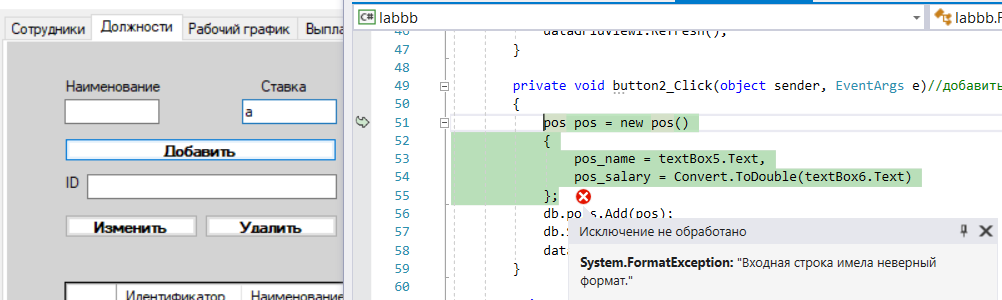
В тест-кейсах №2, 4 невозможно добавить или изменить данные на некорректные, так как обязательные поля имеют изначальное заполнение:

*  
Рис. 52. Тест-кейсы №2, 4*

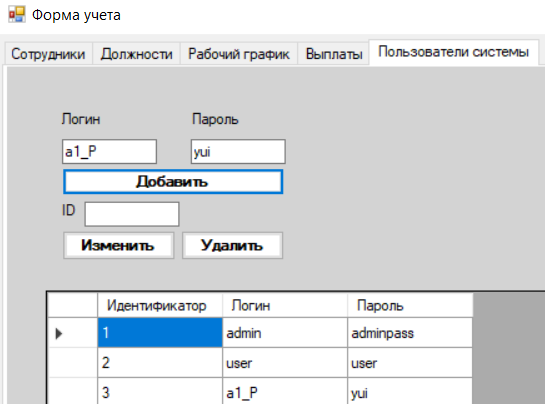
Для тест-кейсов 6, 12, 18, 24, 30 ошибки аналогичны:

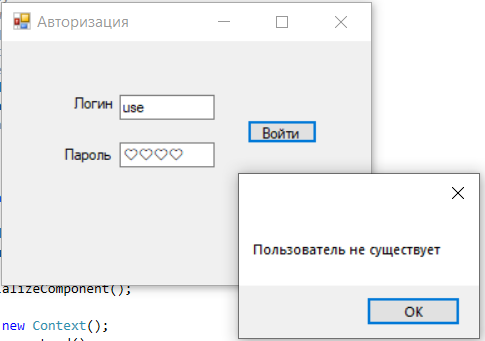
*  
Рис. 53. Тест-кейс №6, 12, 18, 24, 30.*

Также аналогичные ошибки появляются при проверке тест-кейсов №8, 10, 14, 16, 20, 22.

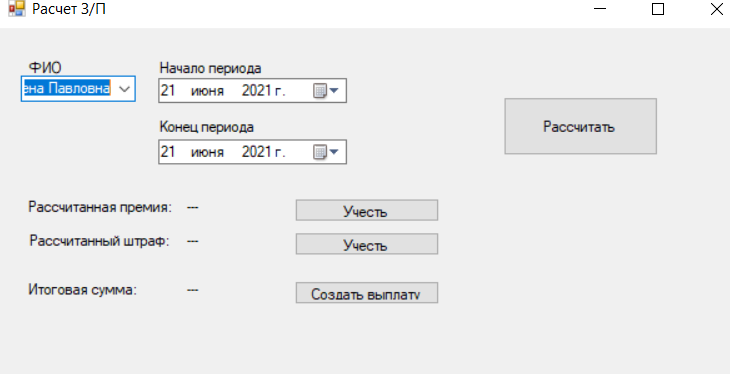
*  
Рис. 54. Результат тест-кейсов №8, 10, 14, 16, 20, 22*

В тест-кейсах 26, 28 невозможно добиться негативного результата, так как входные строки имеют символьный тип данных, то есть могут принимать любые значения:

*  
Рис. 55. Тест-кейс №26, 28*

*  
Рис. 56. Тест-кейс №32*

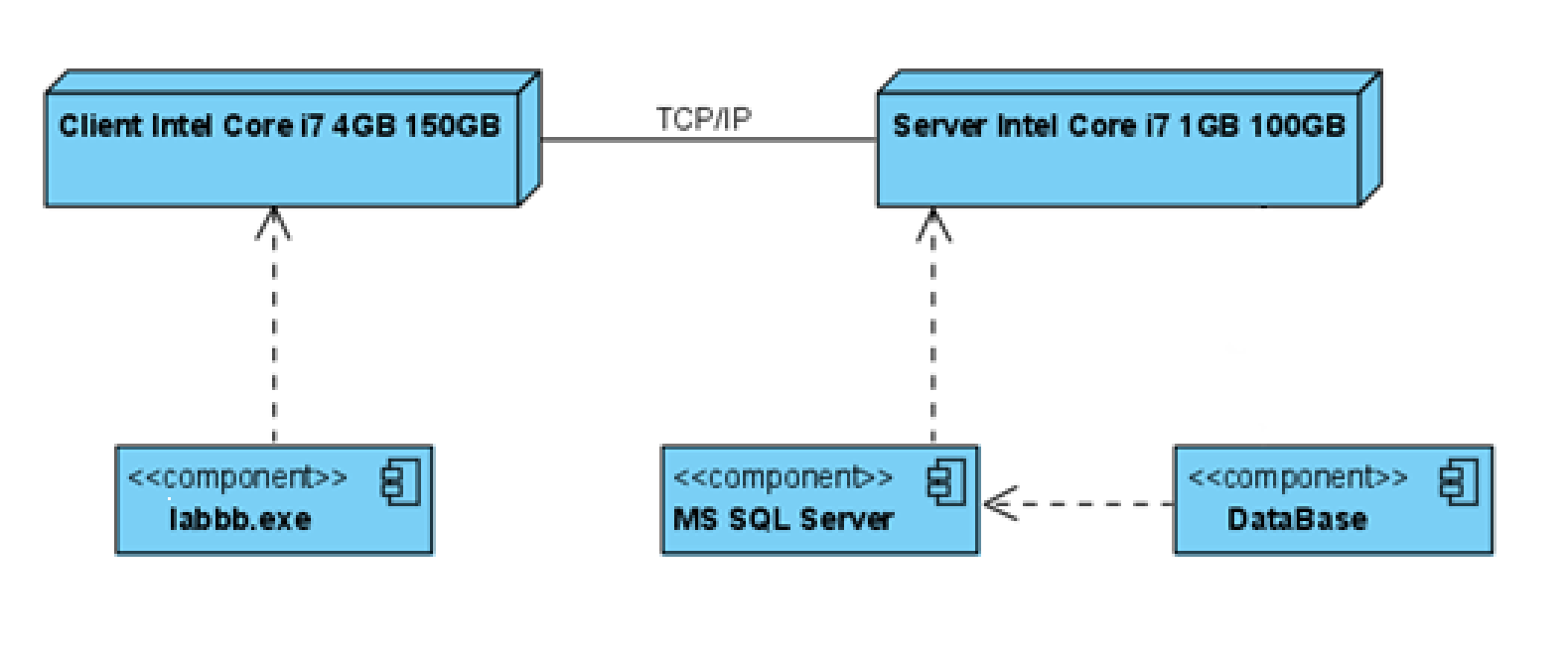
Для тест-кейса №34 невозможно получить негативный результат, так как все поля формы имеют предзаполение:

*Рис. 57. Тест-кейс №34*

# Аппаратная и административная интеграция информационной системы.

## Разработка схемы развертывания системы. Установка БД и клиентского приложения

Процесс развертывания информационной системы начинается с присоединения БД к рабочему серверу предприятия, установка нового программного обеппечения не требуется, так как на предприятии уже используется MS SQL Server 2008R2 и SQL Server Management Studio. На машину, являющуюся клиентом, устанавливается разработанное клиентское приложение. На рисунке 58 изображена схема развертывания

*  
Рис. 58. Диаграмма развертывания*

Для развертывания информационной системы необходимо выполнить следующие действия:

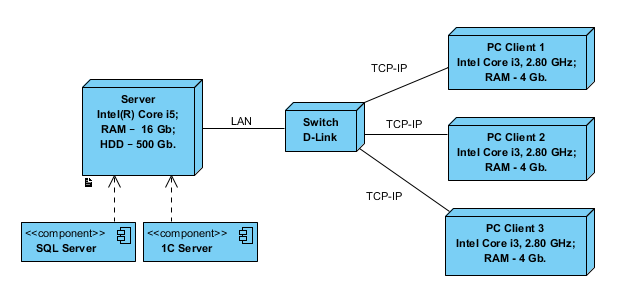
1.1. Используя файл test1BD.bak, создать копию базы данных в среде Management Studio – для версии SQL Server 2008.

1.2. Используя файл test1Scripts.sql, создать базу данных, таблицы, механизмы БД, а также выполнить заполнение созданных таблиц – для других версий SQL Server.

2. Скопировать папку labbb с клиентским приложением на локальный компьютер.

3. В файл labbb.exe.config добавить имя сервера, к которому будет осуществляться подключение. Нужное имя сервера должно быть установлено в качестве значения атрибута «Data Source».

4. Запустить приложение, используя исполняемый файл labbb.exe.



*Рис. 86. Диаграмма развертывания.*

### Формулировка требований к физическим устройствам и сетевому оборудованию, состав рабочих мест

Данная информационная система разработана для последующей установки на ПК с установленной операционной системой Microsoft Windows 10, которая уже установлена на данном предприятии. Смена данной ОС будет нецелесообразной, так как это высокоскоростная, защищенная и надежная система.

Для функционирования информационной системы необходимо следующее оборудование:

* + Сервер СУБД – будет использоваться для размещения и управления базами данных исходя из планируемого объема записей БД. На предприятии имеется сервер с настроенным MS SQL Server.
  + Рабочие станции, то есть ПК – будут использоваться как клиенты информационной системы. В отделе, куда предполагается внедрить ИС, находится две рабочих станции, которые должны быть подключены к локальной вычислительной сети.
  + Сетевой кабель для объединения рабочих станций в единую ЛВС. Все рабочие станции включены в единую ЛВС.

Состав рабочих мест состоит из рабочего места главного бухгалтера и бухгалтера, системный администратор приезжает на предприятие по факту вызова.

План-схем состава рабочих мест выявлено не было, но это не повлияло на качество выполняемой работы.

Все рабочие станции, установленные на рабочих местах, удовлетворяют рекомендуемым требованиям операционной системы.

Для передачи данных необходима сеть с пропускной способностью 100 Мбит/с. В подразделении уже настроена и проведена сеть с такой пропускной способностью.

### Выбор состава аппаратных средств

В качестве рабочих станций рекомендуется использовать компьютеры со следующими характеристиками:

* Intel® Pentium® E 2220 (2400MHz)
* Память 1GB DDRII PC-6400
* Жесткий диск SATA-II Seagate Pipeline HD 250 Гб
* Монитор Acer K192HQLb

Имеющиеся на данный момент рабочие станции полностью удовлетворяют требующимся характеристикам аппаратных средств.

Характеристики сервера:

* ОС – Windows 10;
* Тип системы – 64-разрядная операционная система;
* CPU – Intel(R) Core i5;
* RAM – 16 Gb;
* HDD – 500 Gb.

Имеющиеся на предприятии ПК полностью удовлетворяют требования для создания сети.

Соединение между ПК пользователей и сервером реализуется посредством сетевого оборудования.

### Расчет потребности персонала

Расчет потребности персонала производится с учетом требований, предъявленных к системе.

Для обеспечения безопасности и надежности информационной системы необходим следующий персонал:

**Инженер-программист** для обеспечения возможности расширения информационной системы, а также возможности ее внедрения в другие системы.

На предприятии ООО «Мажор» имеется инженер-программист, обеспечивающий работоспособность системы на предприятии.

Информационной системой будут пользоваться сотрудники бухгалтерии – бухгалтер и главный бухгалтер.

Система имеет разные уровни доступа:

* Бухгалтер - поддержание в актуальном состоянии информации, хранящейся в системе.
* Главный бухгалтер - обеспечением целостности, достоверности и сохранности циркулирующих в автоматизированной информационной системе данных;

**Вывод**: на предприятии имеются необходимые сотрудники, функциями администрирования системы управляет сотрудник из числа тех, кто будет пользоваться системой, а также имеющий разрешение на редактирование данных и знающий бизнес-процессы отдела.

### Разработка среды интеграции

Локально-вычислительная сеть — это сеть, предназначенная для обработки, хранения и передачи данных. ЛВС представляет собой логически разделенную на структурные подсистемы кабельную систему здания, которая включает в себя кабельную локальную сеть, активное сетевое оборудование, серверы и рабочие станции.

Основные требования к ЛВС:

Гибкость в подключении к глобальным сетям;

Стабильность рабочих процессов при изменениях в порядке подключения сетевых устройств, их перемещения или изъятия из структуры сети;

Высокая производительность системы за счет объединения технических ресурсов без потери в скорости обмена информацией;

Доступность и простота администрирования.

Задачи, которые позволяет выполнять ЛВС:

* Доступ к приложениям на сервере;
* Централизованный доступ к данным;
* Объединение рабочих станций в единую информационную систему.
* Совместное использование техники и данных, ускорение рабочих процессов;
* Совместная работа с документами.

Грамотно настроенная локальная сеть – это универсальный инструмент, который увеличивает коэффициент производительности нескольких устройств с одновременным упрощением задачи по их управлению.

#### 6.5.1 Выбор сетевой архитектуры и технологии

##### *6.5.1.1 Выбор архитектуры*

На предприятии ООО «Мажор» уже настроена локально-вычислительная сеть со следующими параметрами, представленными в таблице 6.1.

*Таблица 10*

*Основные характеристики сети*

|  |  |
| --- | --- |
| **Компонент/характеристика** | **Вариант сети** |
| Топология | Звезда\* |
| Линия связи | Витая пара\*\* |
| Сетевые адаптеры | Fast Ethernet |
| Ретрансляторы (повторители, концентраторы, коммутаторы, мосты, маршрутизаторы, шлюзы) | Коммутаторы (2 шт.), роутер (1 шт.). |
| Управление совместным использованием ресурсов | Клиент-серверная архитектура построения |
| Поддерживаемые приложения | Электронная почта, работа с БД. |

\*Топология ЗВЕЗДА – базовая топология компьютерной сети, которая предполагает подключение каждого компьютера отдельным кабелем к отдельному порту устройства. Данная типология наиболее удобна при поиске поврежденных сетевых элементов: кабеля, сетевых адаптеров или разъёмов.

\*\*Витая пара – кабель связи, который представляет собой провода, изготовленных из меди, свитых между собой. Такая пара достаточно помехоустойчивая.

##### *6.5.1.2 Выбор технологии и аппаратных средств. Расчет сети*

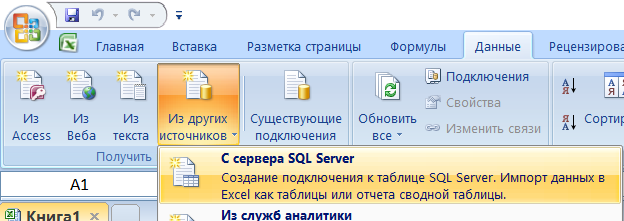
На предприятии установлена стандартная сеть на базе технологии Fast Ethernet.

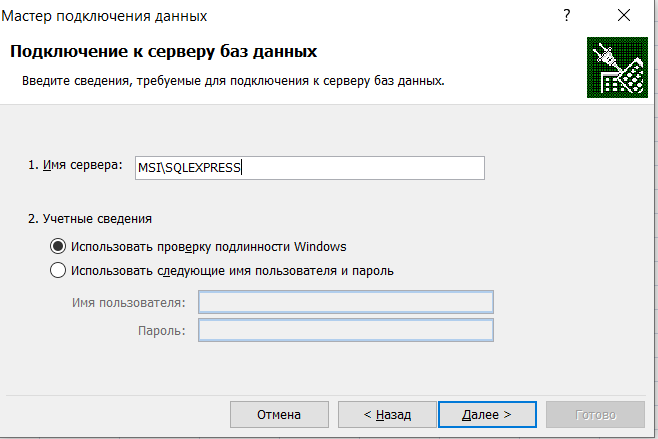
Так как сеть уже создана, а ее последующее расширение не связано напрямую с разрабатываемой системой, то стоимость сети в проект не закладывается.

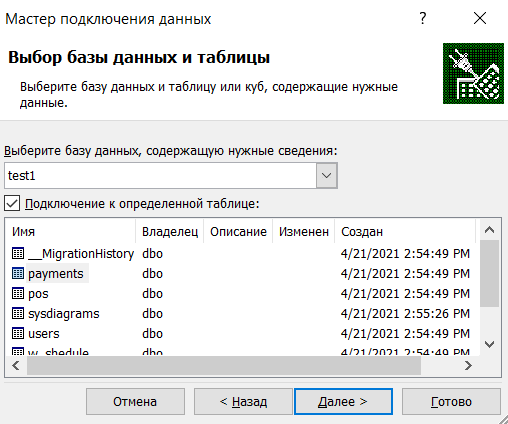
Закупка сервера и необходимого для него программного и аппаратного обеспечения в организацию не требуется, так как сервер уже установлен. Таким образом, затрат на реорганизацию существующей сети нет.

##### *6.5.1.3 Разработка среды связи с вешними объектами.*

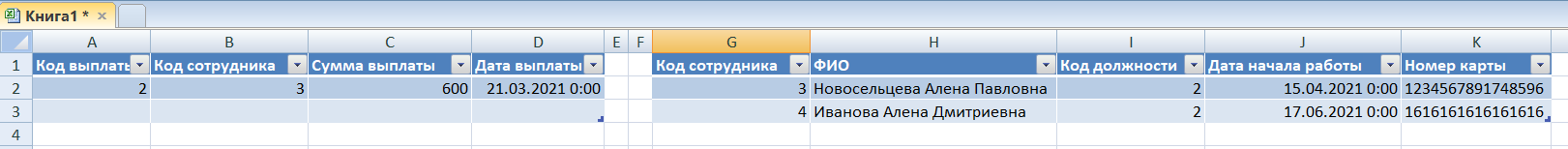
Экспорт составленных отчетов реализуется в файл формата Excel, для дальнейшего использования (печати или хранения) (рис. 87-90).

*  
Рис. 87. Подключение к БД*

*  
Рис. 88. Подключение к серверу баз данных*

*  
Рис. 89. Выбор базы данных и таблицы*

Для простоты восприятия отчета также добавлена таблица «Сотрудники», а также все столбцы русифицированы:

* *Рис. 90. Созданный документ Excel.*

# Общие вопросы администрирования.

Администрирование системы – это совокупность действий по обслуживанию ПО и контролю работы серверных станций, мониторинг работы общесистемного программного обеспечения серверов. Обеспечивается это путём управления, позволяющего свести к минимуму затраты времени и ресурсов, которые направлены на управление системой, и в тоже время максимизировать доступность, производительность и продуктивность системы.

Принципы:

1. Обеспечение устойчивой работы и непрерывного доступа.
2. Мониторинг процессов программных комплексов и серверных систем.
3. Немедленное исправление ошибок и сбоев в работе.
4. Улучшение программных комплексов, обновление систем, анализ и установка новых перспективных модулей.
5. Проведение технических экспертиз электронных курсов.

## Определение стратегии администрирования на уровне руководства и целей предприятия.

Целями администрирования для обеспечения работы отдела бухгалтерии являются:

- обеспечение целостности данных, то есть данные всегда остаются корректными и достоверными;

- обеспечение конфиденциальности данных, то есть существует аутентификация пользователей и разделение прав доступа (администратор, пользователь) к информации в базе данных;

- обеспечение доступности данных, то есть обеспечение бесперебойной работы информационной системы и возможность восстановления информации при возникновении сбоев.

## Определение объектов администрирования на уровне предприятия.

Объектами администрирования являются:

* рабочие станции под управлением системного администратора;
* база данных и её объекты – работа осуществляется с помощью Microsoft SQL Server;
* сетевые устройства для обеспечения связи между компьютерами или для выхода в интернет (маршрутизатор);
* программные компоненты (установка программных компонент, происходит по требованию или же в случае необходимости).

### Определение объектов администрирования на уровне разрабатываемой информационной системы

В процессе разработки были определены следующие объекты администрирования на уровне разрабатываемой ИС:

* разработанное приложение;
* база данных;
* рабочие станции;
* сервер;

## Политика администрирования.

### Политика администрирования на уровне предприятия.

Политика администрирования на уровне предприятия отражается в следующих пунктах:

* Рабочие станции. Необходимое оборудование на предприятии уже настроено. Настройкой рабочих станций, устранением различных проблем занимается инженер-программист.
* Сервера. Резервные копии файлов хранятся на сервере. За периодическое обновление копий конфигурационных файлов отвечает инженер-программист. При любом нарушении системы резервные копии, хранящиеся на сервере, используются для ее восстановления.
* Программное обеспечение. Набор и конфигурация программного обеспечения рабочих станций, установка программного обеспечения осуществляется инженером-программистом.
* Принтеры. Общий доступ к принтеру нескольким компьютерам реализован с помощью специального сервера печати, настроенного на сервере. Инженер-программист следит за правильной работой принтеров и их настройкой.
* Сеть. Для адресации сети используется адресное пространство разрешенного частного адресного пространства Интернет. Все сетевые компоненты идентифицированы и учтены в базе данных сетевого администрирования. Следит за актуальностью информации базы данных сетевого администрирования инженер-программист, доступ к данному ресурсу имеет он и главный бухгалтер. Также возможен удаленный доступ к информационно-техническим сервисам предприятия через канал связи удаленного доступа.

### Политика администрирования на уровне разрабатываемой ИС.

Политика администрирования подразумевает разделение прав доступа к информации, хранящейся в информационной системе, различных типов пользователей.

Объектами разработанной информационной системы являются вкладки «Сотрудники», «Должности», «Рабочий график», «Выплаты», «Пользователи системы» и форма «Расчет З/П».

Выделяется два типа пользователей:

* Администратор (главный бухгалтер) - имеет все права доступа к системе.
* Пользователь (бухгалтер) - имеет право на добавление, и просмотр данных вкладок «Сотрудники», «Выплаты», «Рабочий график» и формы «Расчет З/П», просмотр вкладки «Должности», а также их сохранение на компьютер и дальнейшее использование.

Каждый сотрудник предприятия имеет свой логин и пароль для входа в систему.

# Вопросы информационной безопасности.

## Вопросы информационной безопасности на уровне предприятия

Для обеспечения информационной безопасности необходимо учитывать следующие факторы:

* Физическая безопасность - обеспечение защиты от механических повреждений или проникновений в систему;
* Информационная безопасность – обеспечение защиты информации с помощью установки паролей доступа, применения шифрования;
* Программная безопасность – обеспечение защиты путем установки программ – блокираторов атак из сети.

Существует несколько видов угроз в отношении информационных систем:

* нарушение конфиденциальности – информация становится известна тем, кто не имеет права доступа к ней;
* нарушение целостности – любое незапланированное изменение информации, хранящейся в системе;
* отказ служб – блокировка доступа к некоторым ресурсам системы посредством преднамеренных действий пользователем системы.

Для обеспечения безопасности информации на предприятии реализованы следующие методы защиты:

* антивирусные пакеты для защиты от вирусного программного обеспечения;
* ограничение прав доступа к важной информации, использование паролей;
* инструкции по эксплуатации для защиты от ошибок при

использовании программных средств;

* датчики с системой сигнализации, реагирующие на температуру, для защиты от возникновения техногенных факторов.

На предприятии применяются следующие виды инженерно-технической защиты информации:

* физическая – круглосуточная охрана внутри здания и прилегающей территории, система видеонаблюдения, вход в отдел бухгалтерии проводится через чипы, имеющиеся только у сотрудников;
* аппаратная – использование источников бесперебойного питания, выявление и ликвидация каналов утечки информации;
* программная – использование резервного копирования информации и автоматически обновляемого антивирусного программного обеспечения.

### Анализ угроз

Защита информации включает в себя комплекс мероприятий, направленных на обеспечение информационной безопасности. Информационная безопасность - это защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, чреватых нанесению ущерба владельцам или пользователям.

Информация с точки зрения информационной безопасности обладает следующими категориями:

1. Доступность – гарантия того, что сейчас поддерживается полноценный и надежный доступ к информации для лиц, имеющих соответствующие полномочия.
2. Конфиденциальность – гарантия того, что конкретная информация доступна только тому кругу лиц, для кого она предназначена; нарушение этой категории называется хищением, либо раскрытием информации;
3. Целостность – гарантия того, что информация сейчас существует в ее исходном виде, то есть при ее хранении или передаче не было произведено несанкционированных изменений; нарушение этой категории называется фальсификацией сообщения;

В отношении информационных систем принято считать, что информационная безопасность информационных систем обеспечена в том случае, если для любых информационных ресурсов в системе поддерживается определенный уровень: конфиденциальности; целостности; доступности.

Соответственно все угрозы можно разделить на три основных вида угроз:

1. Угроза нарушения конфиденциальности – то есть, информация становится доступной и известной тому, кто не имеет разрешения для доступа к ней.
2. Угроза нарушения целостности - умышленное искажение информации, хранящейся в вычислительной системе.
3. Угроза нарушения доступности - блокировка доступа к некоторым ресурсам или снижение её работоспособности, что может привести к отказу в обслуживании.

Для обеспечения мер безопасности, проведен анализ возможных угроз (таблица 11) по действующему стандарту ГОСТ Р 51275-2006 и составлена общая оценка ущерба (таблица 12).

*Таблица 11*

*Анализ возможных угроз*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Внутренние | | | Внешние | | |
| № | Название | Вероятность | № | Название | Вероятность |
| Объективные | 1 | Передача сигналов по проводным линиям связи | 2 | 16 | Явления техногенного характера: непреднамеренные электромагнитные облучения ОИ | 4 |
| 2 | Передача сигналов по оптико-волоконным линиям связи | 3 | 17 | Явления техногенного характера: радиационные облучения ОИ; | 3 |
| 3 | Передача сигналов в диапазоне радиоволн и в оптическом диапазоне длин волн | 2 | 18 | Явления техногенного характера: сбои, отказы и аварии систем обеспечения ОИ. | 4 |
| 4 | Излучения акустических сигналов | 1 | 19 | Природные явления, стихийные бедствия: термические факторы (пожары и т.д.); | 5 |
| 5 | Электромагнитные излучения и поля | 1 | 20 | Природные явления, стихийные бедствия: климатические факторы (наводнения и т.д.); | 1 |
| 6 | Побочные электромагнитные излучения элементов (устройств) ТС ОПИ | 1 | 21 | Природные явления, стихийные бедствия: механические факторы (землетрясения и т.д.); | 2 |
| 7 | Побочные электромагнитные излучения на частотах работы высокочастотных генераторов устройств, входящих в состав ТС ОПИ | 1 | 22 | Природные явления, стихийные бедствия: электромагнитные факторы (грозовые разряды и т.д.); | 3 |
| 8 | Побочные электромагнитные излучения на частотах самовозбуждения усилителей, входящих в состав ТС ОПИ. | 1 | 23 | Природные явления, стихийные бедствия: биологические факторы (микробы, грызуны и т.д.); | 2 |
| 9 | Модуляция паразитного электромагнитного излучения информационными сигналами; | 2 | 24 | Природные явления, стихийные бедствия: химические факторы (химически агрессивные среды и т.д.). | 1 |
| 10 | Модуляция паразитного электромагнитного излучения акустическими сигналами. | 1 |  |  |  |
| 11 | Наводка в электрических цепях ТС, имеющих выход за пределы ОИ | 2 |  |  |  |
| 12 | Наводка в линиях связи | 1 |  |  |  |
| 13 | Наличие акустоэлектрических преобразователей в элементах ТС ОИ. | 1 |  |  |  |
| 14 | Дефекты, сбои и отказы, аварии ТС и систем ОИ. | 9 |  |  |  |
| 15 | Дефекты, сбои и отказы программного обеспечения ОИ. | 9 |  |  |  |
| Субъективные | 25 | Разглашение защищаемой информации лицами, не имеющими права доступа к защищаемой информации; | 3 | 45 | Доступ к защищаемой информации с применением технических средств разведки: | 4 |
| 26 | Разглашение защищаемой информации лицами, имеющими к ней право доступа, через передачу информации по открытым линиям связи; | 4 | 46 | Доступ к защищаемой информации с применением технических средств съема информации. | 4 |
| 27 | Разглашение защищаемой информации лицами, имеющими к ней право доступа, через  обработку информации на незащищенных ТС обработки информации; | 5 | 47 | Несанкционированный доступ к защищаемой информации путем подключения к техническим средствам и системам ОИ; | 4 |
| 28 | Разглашение защищаемой информации лицами, имеющими к ней право доступа, через  опубликование информации в открытой печати и других средствах массовой информации; | 6 | 48 | Несанкционированный доступ к защищаемой информации путем использования закладочных средств [устройств]; | 3 |
| 29 | Разглашение защищаемой информации лицами, имеющими к ней право доступа, через  копирование информации на незарегистрированный носитель информации; | 5 | 49 | Несанкционированный доступ к защищаемой информации путем применение вирусов или другого вредоносного программного кода (троянские программы, клавиатурные шпионы, активное содержимое документов); | 8 |
| 30 | Разглашение защищаемой информации лицами, имеющими к ней право доступа, через  передачу носителя информации лицам, не имеющим права доступа к ней; | 4 | 50 | Несанкционированный доступ к защищаемой информации путем несанкционированного физического доступа к ОИ; | 5 |
| 31 | Разглашение защищаемой информации лицами, имеющими к ней право доступа, через  утрату носителя информации. | 5 | 51 | Несанкционированный доступ к защищаемой информации путем хищения носителя информации. | 4 |
| 32 | Неправомерные действия со стороны лиц, имеющих право доступа к защищаемой информации, путем несанкционированного изменения информации; | 4 | 52 | Блокирование доступа к защищаемой информации путем перегрузки технических средств обработки информации ложными заявками на ее обработку. | 2 |
| 33 | Неправомерные действия со стороны лиц, имеющих право доступа к защищаемой информации, путем  несанкционированного копирования защищаемой информации. | 6 | 53 | Действия криминальных групп и отдельных преступных субъектов | 3 |
| 34 | Несанкционированный доступ к информации путем подключения к техническим средствам и системам ОИ; | 6 | 54 | Искажение, уничтожение или блокирование информации с применением технических средств путем преднамеренного силового электромагнитного воздействия; | 1 |
| 35 | Несанкционированный доступ к информации путем использования закладочных средств [устройств]; | 8 | 55 | Искажение, уничтожение или блокирование информации с применением технических средств путем преднамеренного силового воздействия различной физической природы; | 1 |
| 36 | Несанкционированный доступ к информации путем использования программного обеспечения технических средств ОИ | 3 | 56 | Искажение, уничтожение или блокирование информации с применением технических средств путем использования программных или программно-аппаратных средств при осуществлении | 3 |
| 37 | Несанкционированный доступ к информации путем хищения носителя защищаемой информации; | 3 | 57 | Искажение, уничтожение или блокирование информации с применением технических средств путем воздействия программными средствами в комплексе с преднамеренным силовым электромагнитным воздействием. | 1 |
| 38 | Несанкционированный доступ к информации путем нарушения функционирования ТС обработки информации. | 2 |  |  |  |
| 39 | Недостатки организационного обеспечения защиты информации при задании требований по защите информации (требования противоречивы, не обеспечивают эффективную защиту информации и т.д.); | 1 |  |  |  |
| 40 | Недостатки организационного обеспечения защиты информации при несоблюдении требований по защите информации; | 1 |  |  |  |
| 41 | Недостатки организационного обеспечения защиты информации при контроле эффективности защиты информации. | 1 |  |  |  |
| 42 | Ошибки обслуживающего персонала ОИ при эксплуатации ТС; | 2 |  |  |  |
| 43 | Ошибки обслуживающего персонала ОИ при эксплуатации программных средств; | 2 |  |  |  |
| 44 | Ошибки обслуживающего персонала ОИ при эксплуатации средств и систем защиты информации. | 2 |  |  |  |

Оценка вероятности возникновения угрозы:

1. очень вероятна – 9-10 баллов,

2. вероятна – 5-8 баллов,

3. маловероятна – 3-5 баллов.

4. практически невероятна 1-2 балла.

Оценка степени ущерба возникновения угрозы:

1. полная потеря данных – 9-10 баллов,

2. частичная потеря данных – 3-8 балла,

3. возможная потеря данных – 1-2 балла.

*Таблица 12*

*Оценка угроз*

|  |  |  |  |
| --- | --- | --- | --- |
| Угрозы | Вероятность возникновения | Ущерб | Общая оценка угрозы |
| Отказы источников питания и скачки напряжения | 8 | 4 | 32 |
| Природные явления (молния) | 4 | 8 | 32 |
| Пожары | 4 | 8 | 32 |
| Ошибки пользователей | 5 | 6 | 30 |
| Воровство/вандализм | 3 | 8 | 24 |
| Несанкционированный доступ | 5 | 9 | 45 |
| Компьютерные вирусы | 2 | 6 | 12 |
| Сбои ПО | 3 | 9 | 29 |
| Сбои аппаратного обеспечения | 4 | 6 | 24 |
| Механические повреждения кабеля | 2 | 6 | 12 |

В следующих пункт описаны меры, принятые на предприятии, для предупреждения всех вышеописанных факторов.

## Контроль доступа на объекты и в помещения.

В помещение бухгалтерии вход возможен при наличии чипа, который выдается сотрудникам. Реализована постоянная охрана здания и помещений. В помещение с серверным оборудованием вход разрешается только специалистам, которые обслуживают данное оборудование. У каждого сотрудника есть ключ от помещения, в котором они работают. Сотрудник обязан сдать ключ от помещения после окончания рабочего дня сотрудникам охраны и отметится в журнале. Ключ выдается конкретному сотруднику, имеющему доступ в помещение.

Важная документация, служебная информация хранится в сейфе, доступ к которому имеет только главный бухгалтер и генеральный директор предприятия.

## Обеспечение безопасности с помощью аппаратных средств.

На предприятии обеспечено надежное функционирование аппаратных средств и предотвращение доступа к ним посторонних лиц. Для этого применяются следующие меры:

* помещения, имеющие аппаратные средства, оборудованы огнетушителями;
* все аппаратные средства промаркированы и подлежат инвентаризационному учету;
* эксплуатация аппаратных средств ведется по назначению, по правилам, установленным производителем;
* аппаратные средства периодически диагностируются, а их конфигурация обновляется по мере необходимости;
* для работы сервера используются источники бесперебойного питания.

## Обеспечение безопасности с помощью программных средств.

Для обеспечения безопасной работы с программными средствами применяются следующие правила:

* использование автоматически обновляемого антивирусного программного обеспечения;
* резервные копии баз данных периодически обновляются на отдельных носителях информации;
* на компьютерах установлено только лицензионное программное обеспечение;
* перед установкой программное обеспечение протестировано лицом ответственным за это;
* программное обеспечение настроено таким образом, чтобы удовлетворять политику информационной безопасности;
* с сотрудниками проведен инструктаж по использованию программного обеспечения;
* доступ к программному обеспечению осуществляется после процедуры идентификации и авторизации.

## Определение политики управления доступом пользователей.

Для каждого сотрудника определены права доступа к информации. Права каждого сотрудника определяются нормативными документами и фиксируются в документах, там же указаны лица, раздающие права.

Каждый сотрудник имеет свою учетную запись пользователя, с индивидуально назначенными правами, логином и паролем. Пароль периодически меняется для всех сотрудников. При вводе пароля в ПО, пароль не отображается в явном виде. Данные обо всех логинах и паролях сотрудников также хранятся централизованно. Доступ к общим сетевым папкам, в которых хранится обрабатываемая информация, также осуществляется посредством выданных логина и пароля.

Серверы защищены всеми доступными средствами. На компьютерах, на которых ведётся обработка защищаемой информации, установлены средства защиты от несанкционированного доступа.

## Антивирусная защита информации

Для обеспечения защиты от заражения вирусами файлов и приложений на каждом ПК установлено антивирусное программное обеспечение. Используемые антивирусные средства являются лицензионными, протестированными до установки и настроеными в соответствии с применением. Пользователям запрещено отключать антивирусное ПО, а также препятствовать обновлению вирусных баз.

Пользователь, загружающий какие-либо данные или программы с внешнего носителя, должен перед загрузкой сканировать этот носитель на наличие на нем вирусов. Также пользователи должны периодически проводить полное сканирование своих ПК на наличие вредоносных программ.

Обновление базы данных антивирусных сигнатур производится автоматически.

## Политика внешнего доступа к информационной системе.

Для сотрудников, которые используют в своей работе службы электронной почты, существуют требования к политике безопасности.

* Аппаратное и программное обеспечение для системы электронной почты принадлежит организации. Все сообщения, созданные, переданные или полученные с помощью системы электронной почты, являются собственностью организации.
* Сообщения должны быть конфиденциальны по отношению к другим сотрудникам предприятия, доступ к этим сообщениям может получить только сотрудник, которому они адресованы.
* Система электронной почты обеспечивает надежную доставку сообщений как внутри предприятия, так и в сети интернет.
* Почтовый сервер проверяет каждое сообщение электронной почты, которое проходит через него, на наличие вирусов.
* Секретная информация не передается в сеть интернет.

## 8.8. Защита персональных данных

Персональные данные – это любая информация, относящаяся к определенному или определяемому на основании такой информации физическому лицу, в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация.

В разработанной ИС хранятся персональные данные сотрудников, такие как: фамилия, имя и отчество, номер банковской карты. Данная информация защищена, защита обеспечена следующим образом:

* все персональные данные хранятся на одном сервере;
* сервер ограничен от общей локальной сети и доступа в интернет межсетевым экраном;
* сервер находится в отдельном помещении, доступ к которому имеет только системный администратор;
* используется сертифицированное ПО;

Для исполнения настоящей политики предприятием утверждены следующие локальные нормативные правовые акты:

1. Приказ о назначении ответственных лиц за работу с персональными данными сотрудников, за компьютерную обработку персональных данных сотрудников, за обеспечение информационной безопасности.
2. Перечень должностей, сотрудников, допущенных к обработке персональных данных.
3. Порядок доступа сотрудников к сведениям конфиденциального характера.
4. Перечень обрабатываемых персональных данных.
5. Модель угроз безопасности персональных данных при их обработке в информационной системе персональных данных.
6. Акты классификации информационных систем персональных данных.

Таким образом, методы защиты информации не требуют каких-либо доработок.

# Расчет экономической эффективности.

## Расчет затрат на проектирование

Проектные затраты на разработку системы определяются по формуле 0:

, (0)

где – затраты на приобретение оборудования

– заработная плата исполнителям и разработчикам проекта

– отчисления во внебюджетные государственные фонды,

– затраты на электроэнергию

– накладные расходы

– затраты на амортизацию

– прочие расходы

Затраты на приобретение оборудования не требуются, т.к. на предприятии имеется всё необходимое оборудование, его достаточно для внедрения системы.

= 0.

Затраты на заработную плату исполнителям и разработчикам проекта : разрабатывается программа проведения проектных работ, определяются ориентировочные трудозатраты этапов работ по исполнителям и стоимость этих затрат (человек-часов) по формуле 1:

= , (1)

где – заработная плата j-го исполнителя;

– ориентировочные трудозатраты i-го этапа j-го исполнителя в часах;

– стоимость человека-часа j-го исполнителя.

Общие затраты на заработную плату определим по формуле 2:

= , (2)

где J=1..n – количество исполнителей

Стоимость человека-часа можно рассчитать по формуле 3:

= / (\*), (3)

где – стоимость человека-часа; – месячный оклад; – количество рабочих дней в месяце (в количестве 21 дня); – 8-ми часовой рабочий день

Средняя заработная плата программиста составляет 45000 руб. в месяц. Вычисляем стоимость человек-часа:

= 45000/(21\*8)= 267,86 руб/ч.

Для вычисления ориентировочных трудозатрат используется формула 4

= \* , (4)

где – ориентировочные трудозатраты; – дни, затраченные на выполнение i-го этапа работы; – 8-ми часовой рабочий день.

Работа руководителя проекта включает в себя следующие этапы, представленные в таблице 13.

*Таблица 13*

*Расчет ориентировочных трудозатрат*

|  |  |  |  |
| --- | --- | --- | --- |
| № | **Этап работы** | **Затраченные дни** | **Ориентировачные трудозатраты  чел-час,** |
| 1 | Постановка задачи | 1 | 8 |
| 2 | Определение хода работы | 1/2 | 4 |
| 3 | Анализ поставленной задачи | 1/2 | 4 |
| 4 | Изучение объекта автоматизации | 1 | 8 |
| 5 | Составление плана работы | 1/2 | 4 |
| 6 | Согласование и корректировка плана работы | 2 | 16 |
| 7 | Описание и характеристика исследуемого объекта автоматизации | 5 | 40 |
| 8 | Оформление проекта | 6 | 48 |
| 9 | Тестирование проекта | 1 | 8 |
| **Итого:** | | 17,5 | 140 |

Заработная плата:

= 267,86\*140 = 37500,4 руб.

Отчисления во внебюджетные государственные фонды приняты в размере 30%, и рассчитываются по формуле 5:

= \*0,3 (5)

= 37500,4 \* 0,3 = 11250,12 руб.

Затраты на электроэнергию считаем по формуле 6:

= (6)

При проектировании ИС использовался только ноутбук, следовательно, затраты на электроэнергию считаются только для ноутбука. Для расчета данной затраты были использованы следующие данные:

 - потребляемая мощность - 0,4 кВт/ч

 - фактическое время - 140 часа.

- цена на электроэнергию – 3,43 руб. за 1 кВт/час.

 - коэффициент использования - 80%.

 = 0,4 \* 140 \* 3,43 \* 0,8 = 153,66 руб.

Накладные расходы принимаются в размере 10% от заработной платы исполнителей и рассчитываются по формуле 7:

= \*0,1 (7)

= 37500,4\*0,1 = 3750,04 руб.

Расходы на амортизацию составляют 12,5% от стоимости оборудования, т.к. закупка оборудования не потребовалась, используется стоимость уже имеющегося оборудования. Приблизительная стоимость оборудования на предприятии составляет около 110 тыс. рублей. Формула 8 для расчёта:

= \*0,125 (8)

= 110000 \* 0,125 = 13750 руб.

Прочие расходы составляют 3% от суммы всех предшествующих затрат (формула 9).

= 0,03\*(+) (9)

= 0,03\*(0+37500,4+11250,12+153,66+3750,04+13750,0) = 1992,13 руб.

Смета проектных затрат приведена в таблице 14.

*Таблица 14*

*Смета затрат*

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование затрат** | **Сумма, руб** |
| 1 | Расходы на оборудование ( | 0 |
| 2 | Расходы на заработанную плату ( | 37500,4 |
| 3 | Отчисление в ВГФ ( | 11250,12 |
| 4 | Расход на электроэнергию ( | 153,66 |
| 5 | Накладные расходы () | 3750,04 |
| 6 | Расходы на амортизацию () | 13750 |
| 7 | Прочие расходы () | 1992,13 |
| **Итого: 68396,35** | | |

## Расчет эксплуатационных затрат

Затраты на внедрение системы вычисляются по формуле 10:

= +++, (10)

где – эксплуатационные затраты; – затраты на ремонт и содержание оборудования; – затраты на оборудование; – заработная плата обслуживающего персонала; – отчисления во внебюджетные государственные формы; – затраты на электроэнергию для эксплуатации оборудования; – накладные расходы; – затраты на амортизацию; – прочие затраты.

Затраты на оборудование (формула 11):

=++, (11)

где – цена оборудования; – транспортные расходы (10% от ); – расходы, связанные с монтажом оборудования (8% от )

= 0 руб.;

= 0 \* 0,1 = 0 руб.;

= 0\* 0,08 = 0 руб.

Отсюда получаем затраты на оборудования: = 0 руб.

Затраты на ремонт и содержание оборудования принимаются в размере 2,5% от стоимости основных средств (используем стоимость имеющегося оборудования, т.к. нет потребности к закупке нового оборудования). Рассчитываются по формуле 12.

= \* 0,025 (12)

= 110000\* 0,025 = 2750 руб.

Формула затрат на заработную плату обслуживающего персонала 13:

= DK, (13)

где D – годовой должностной оклад работника; К – районный коэффициент ( = 1,3).

= (20000 \* 12) \* 1,3 = 312000 руб.

Отчисления в ВГФ рассчитываются по формуле 14:

= \*0,3 (14)

= 312000 \* 0,3 = 93600 руб.

Затраты на электроэнергию рассчитывают по стоимости, потребляемой токоприемником электроэнергии, как указано в формуле 15:

= РТ, (15)

где Р – средняя потребляемая мощность, кВт/час; Т – время работы токоприемника, час; – коэффициент использования по мощности (0,92–0,95); - коэффициент использования по времени (0,92–0,95); – цена за электроэнергию за 1 кВт/час., руб.

Р = 0,7 кВт/час,

Т = 257\*8=2056 часов (количество рабочих дней – 257, рабочий день восьмичасовой), = 0,92 , =0,94, = 3,43 руб.

= 0,7\*2056\*0,92\*0,94\*3,43 = 4269,05 руб.

Накладные расходы принимаются в размере 10% от заработной платы обслуживающего персонала и рассчитываются по формуле 16:

= \*0,1, (16)

тогда = 312000\*0,1 = 31200 руб.

Расходы на амортизацию составляют 12,5% от стоимости оборудования (110 тыс. руб.) и рассчитываются по формуле 8:

= \*0,125 = 110 000 \*0,125 = 13750 руб.

Прочие расходы составляют 3% от суммы всех предшествующих затрат. Для расчета прочих расходов применяется формула 9.

= 0,03\*(+ +)

= 0,03\*(0+2750+312000+93600+4269,05+31200+13750) = 13727,07 руб.

Смета проектных затрат приведена в таблице 15.

*Таблица 15*

*Смета затрат*

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование затрат** | **Сумма, руб** |
| 1 | Расходы на оборудование ( | 0 |
| 2 | Затраты на ремонт | 2750 |
| 3 | Расходы на заработанную плату ( | 312000 |
| 4 | Отчисление на ВГФ ( | 93600 |
| 5 | Расход на электроэнергию ( | 4269,05 |
| 6 | Расходы на амортизацию () | 13750 |
| 7 | Накладные расходы () | 31200 |
| 8 | Прочие расходы () | 13727,07 |
| **Итого: 471296,07** | | |

## Экономическая эффективность

Экономическая эффективность определяется по формуле 17:

= - (17)

За аналогичную систему примем систему 1С Предприятие, её стоимость с учётом оборудования, затрат на сотрудников и т.д. Примем общую стоимость аналогичной системы равную 810000.

Тогда экономическая эффективность будет равна:

= 810000 – 471296,07 = 338703,93‬ руб.

Общие затраты: стоимость действующей ИС рассчитывается как сумма затрат на проектирование и на эксплуатацию (формула 18).

= + (18)

= 68396,35+ 471296,07 = 539692,42 руб.

## Срок окупаемости разработанной системы.

Срок окупаемости: период времени, необходимый для того, чтобы доходы покрыли затраты.

Срок окупаемости информационной системы определяется как период времени, в течении которого экономический эффект от эксплуатации системы сравнивается с капитальными затратами на данную систему.

Срок окупаемости определяется по формуле 19.

Т = / , (19)

где Т – срок окупаемости, лет; – суммарные затраты на разработку и внедрение системы, руб; – годовая экономия, руб

Т = 539692,42 / 338703,93= 1,59 года (менее 1 года 8 месяцев)

## Технико-экономические показатели проекта

Технико**-**экономические показатели проекта представлены в таблице 16.

*Таблица 16*

*Технико-экономические показатели проекта*

|  |  |  |
| --- | --- | --- |
| **Наименование показателя** | **Ед. изм.** | **Значение** |
| **Проектные затраты** | | |
| Затраты на оборудование | руб | 0 |
| Заработная плата исполнителям проекта | руб | 37500,4 |
| Отчисления ВГФ | руб | 11250,12 |
| Затраты на электроэнергию | руб | 153,66 |
| Накладные расходы | руб | 3750,04 |
| Затраты на амортизацию | руб | 13750 |
| Прочие затраты | руб | 1992,13 |
| **Итого** | **руб** | 68396,35 |
| **Эксплуатационные затраты** | | |
| Затраты на оборудование | руб | 0 |
| Затраты на ремонт и содержание оборудования | руб | 2750 |
| Заработная плату обслуживающего персонала | руб | 312000 |
| Отчисления ВГФ | руб | 93600 |
| Расходы на электроэнергию | руб | 4269,05 |
| Затраты на амортизацию | руб | 13750 |
| Накладные расходы | руб | 31200 |
| Прочие расходы | руб | 13727,07 |
| **Итого** | **руб** | **471296,07** |
| **Всего затрат** | **руб** | **539692,42** |
| **Экономическая эффективность** | **руб** | **338703,93** |
| **Срок окупаемости** | **год** | **>1г 8мес** |

# Заключение

Разработана система автоматизации процесса расчета заработных плат на предприятии ООО «Мажор», реализующая ввод, изменение, удаление и вывод данных о сотрудниках, должностях, имеющихся на предприятии, рабочем графике, выплатах заработных плат и пользователях информационной системы, а также позволяющая формировать отчет о выплатах заработной платы. Система хранения на базе SQL сервера обеспечит хранение данных в структурированном виде и упростит задачу добавления новых объектов. Внедрение этой системы позволит сократить временные затраты и улучшить работу отдела бухгалтерии в целом.

В результате выполнения ВКР поставленная цель реализована.

# Список литературы.

1. Ванеев О.Н. Методические указания по выполнению выпускной работы бакалавров направление подготовки 09.03.02 «Информационные системы и технологии» Профиль 01 «Системная интеграция и автоматизация информационных процессов»/ Ванеев О.Н. — Кемерово: КузГТУ, 2017 — 43с.
2. Ванеев О.Н. МУ по курсу "методы и средства проектирования информационных систем" по направлению 230400-062 "Информационные системы и технологии" / Кемерово: КузГТУ
3. Ванеев О.Н., МУ по выполнению курсового проекта по курсу "Проектирование информационных систем" по направлению "Информационные системы и технологии"/ Кемерово: КузГТУ
4. Ванеев О.Н., МУ по курсу «Управление данными» для студентов специальности "Информационные системы и технологии"/ Кемерово: КузГТУ
5. Ларман, Крэг. Применение UML 2.0 и шаблонов проектирования. Практическое руководство. 3-е издание. : Пер. с англ. – М. : ООО "И.Д. Вильяме", 2013. – 736 с. : ил. – Парал. тит. англ.
6. Основы ASP.NET [Электронный ресурс] URL: <https://professorweb.ru/my/ASP_NET/base/level1/base_aspnet_index.php> (дата обращения 13.02.2021).
7. Руководство по ASP.NET MVC 5 [Электронный ресурс] URL: <https://metanit.com/sharp/mvc5/> (дата обращения 12.02.2021).
8. Шилдт, Герберт. Полный справочник по C#. / Пер. с англ. – М. : Издательский дом "Вильямс", 2004. – 752 с. : ил. — Парал. тит. англ.
9. Entity Framework [Электронный ресурс]URL: <https://metanit.com/sharp/entityframework/1.1.php>(дата обращения 03.02. 2021).

# Приложение 1. Техническое задание

1. **Общие сведения**
   1. **Полное наименование системы и её условное обозначение**

«Система расчета заработной платы официанта», «ЗПО»

* 1. **Указание на заказчика и разработчика системы**

Заказчик: ООО «Мажор»

Адрес фактический: г.Кемерово. пр.Октябрьский 53/3  
Телефон/факт: +7(3842)351256

Разработчик: Карпова Е.П.

* 1. **Перечень документов, на основе которых производится разработка.**

Работа выполняется на основании договора №1000 от 01.07.2020г. между ООО «Мажор» и Карповой Е.П.

* 1. **Плановые сроки начала и окончания работы по созданию системы.**

05.02.2021г – 30.05.2021г.

* 1. **Сведения об источниках и порядке финансирования работ.**

Финансирование проекта производится финансовыми средствами предприятия.

* 1. **Порядок оформления и предъявления заказчику результатов работ по созданию системы.**

Работы по созданию ЗПО сдаются Разработчиком поэтапно по мере их готовности.

1. **Назначение и цели создания системы.**
   1. **Назначение системы**

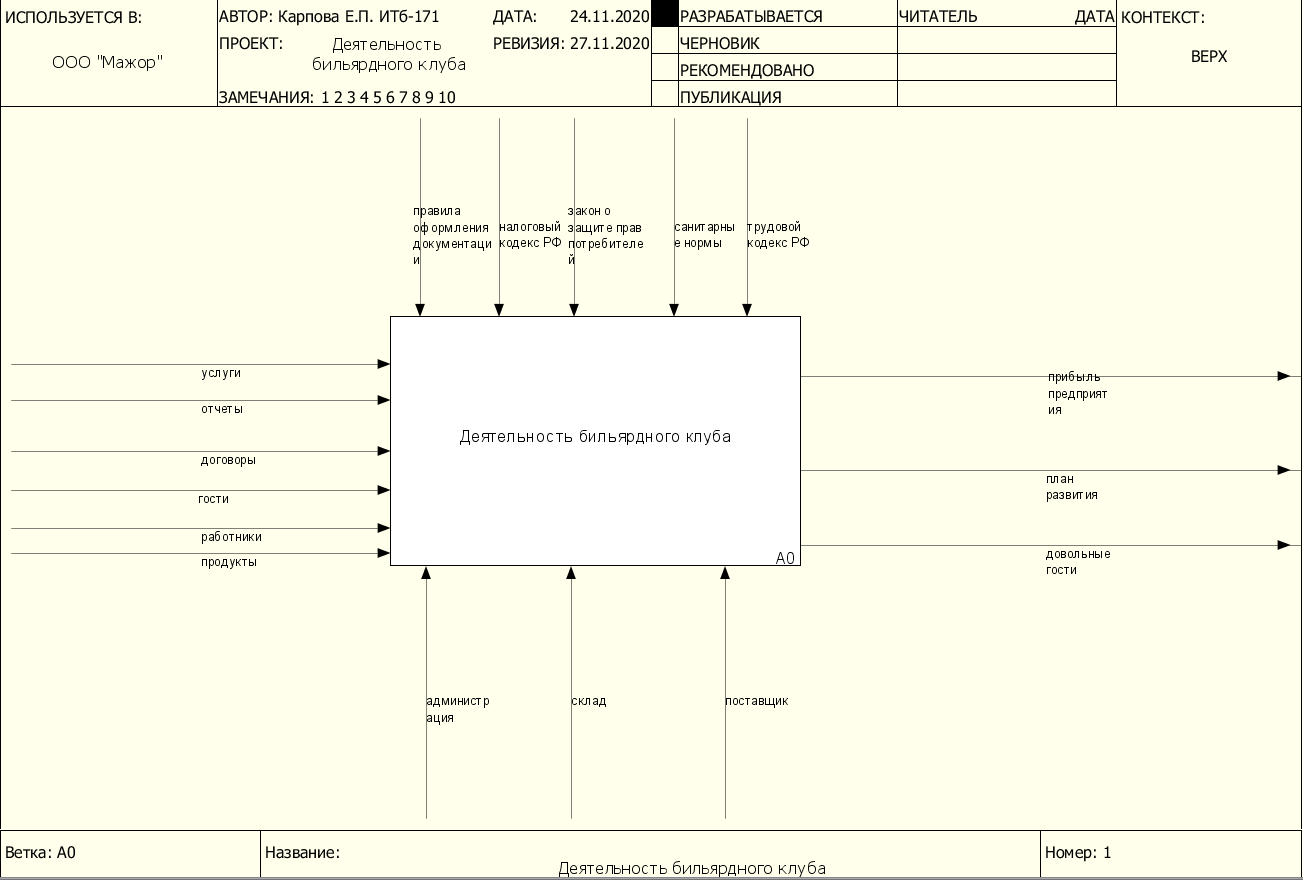
ЗПО предназначена для автоматизации и, следовательно, повышения скорости расчета заработных плат сотрудников.

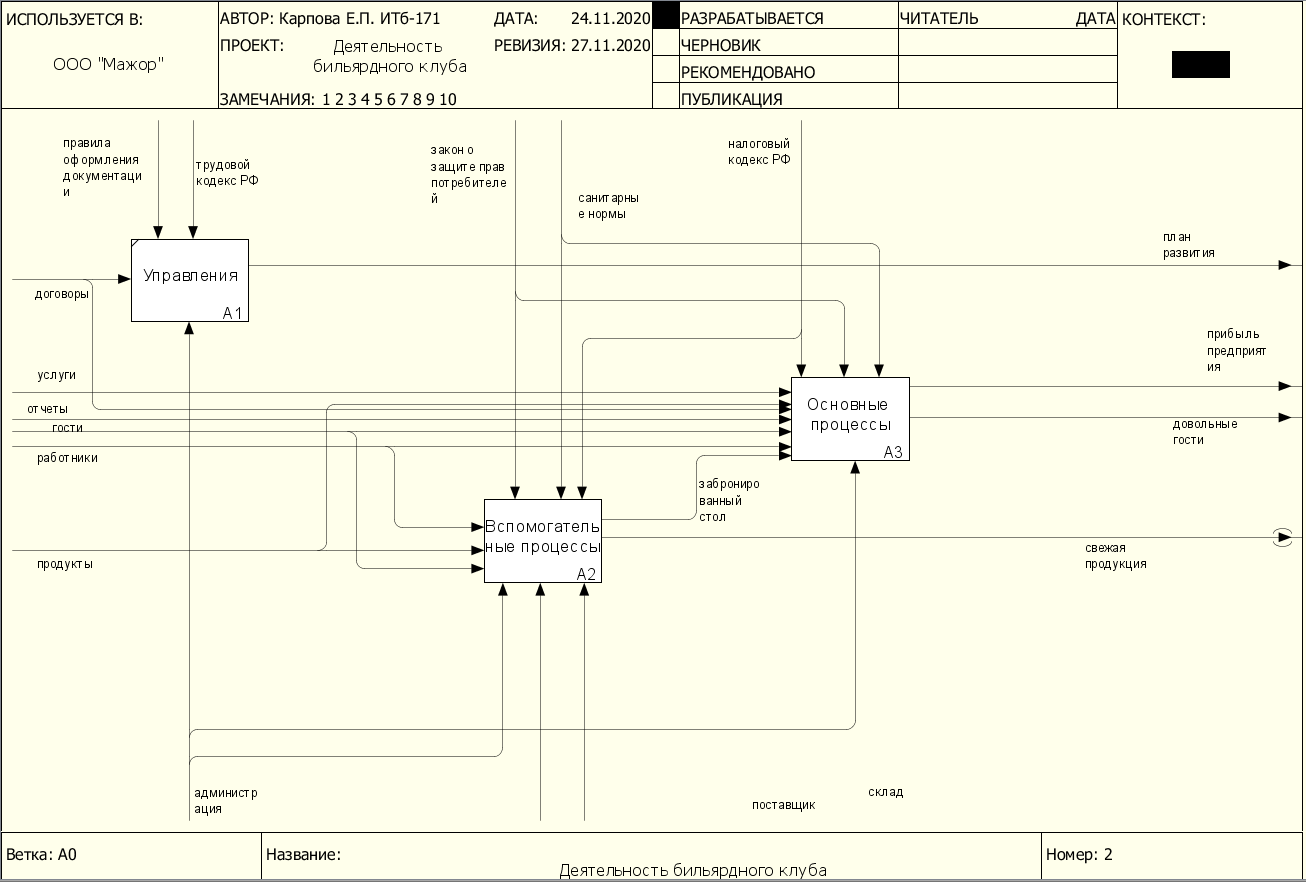
* 1. **Цели создания системы**

ЗПО создается с целью обеспечения сбора и обработки информации из системы R-keeper, необходимой для расчетов, а также проведения самих расчетов.

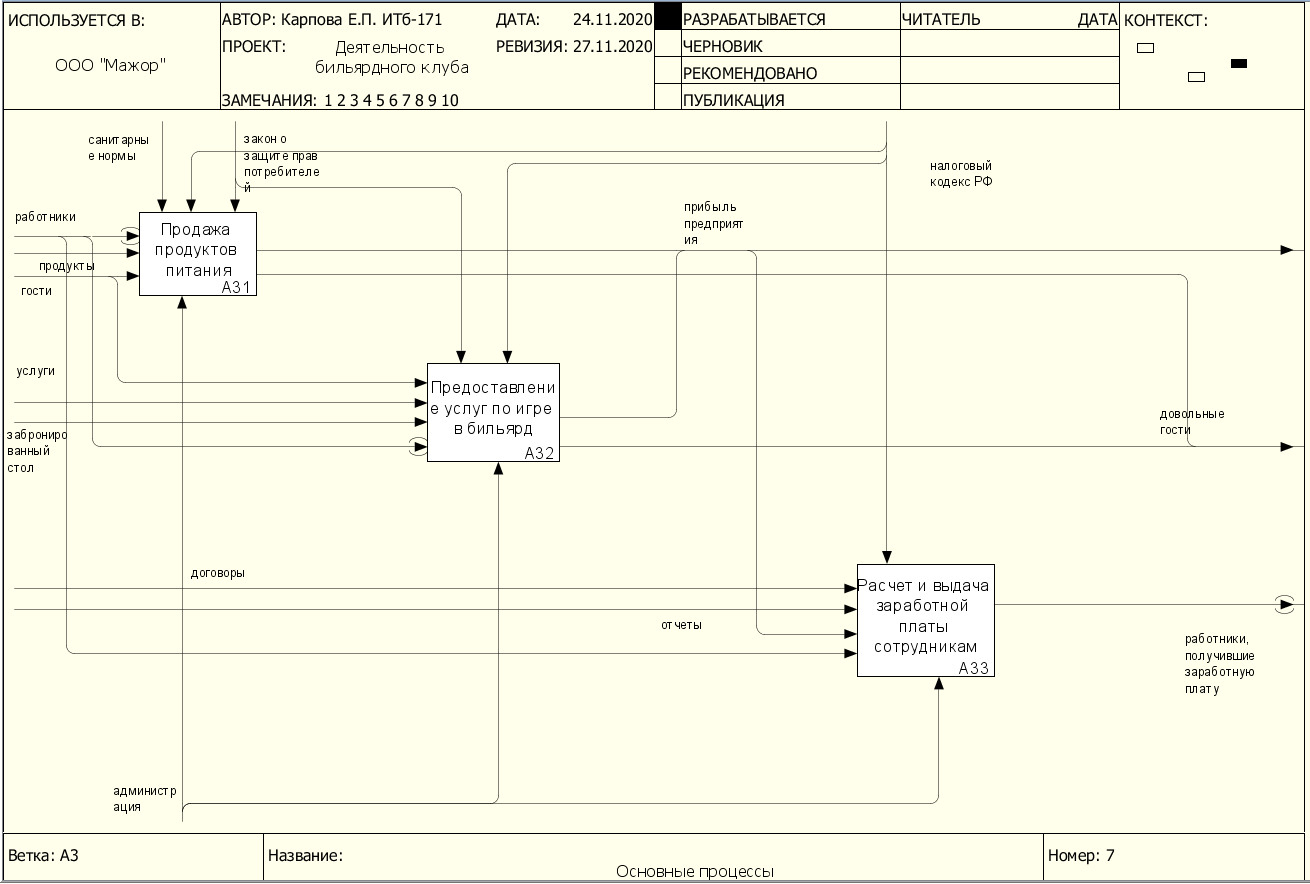
1. **Характеристика объектов автоматизации.**

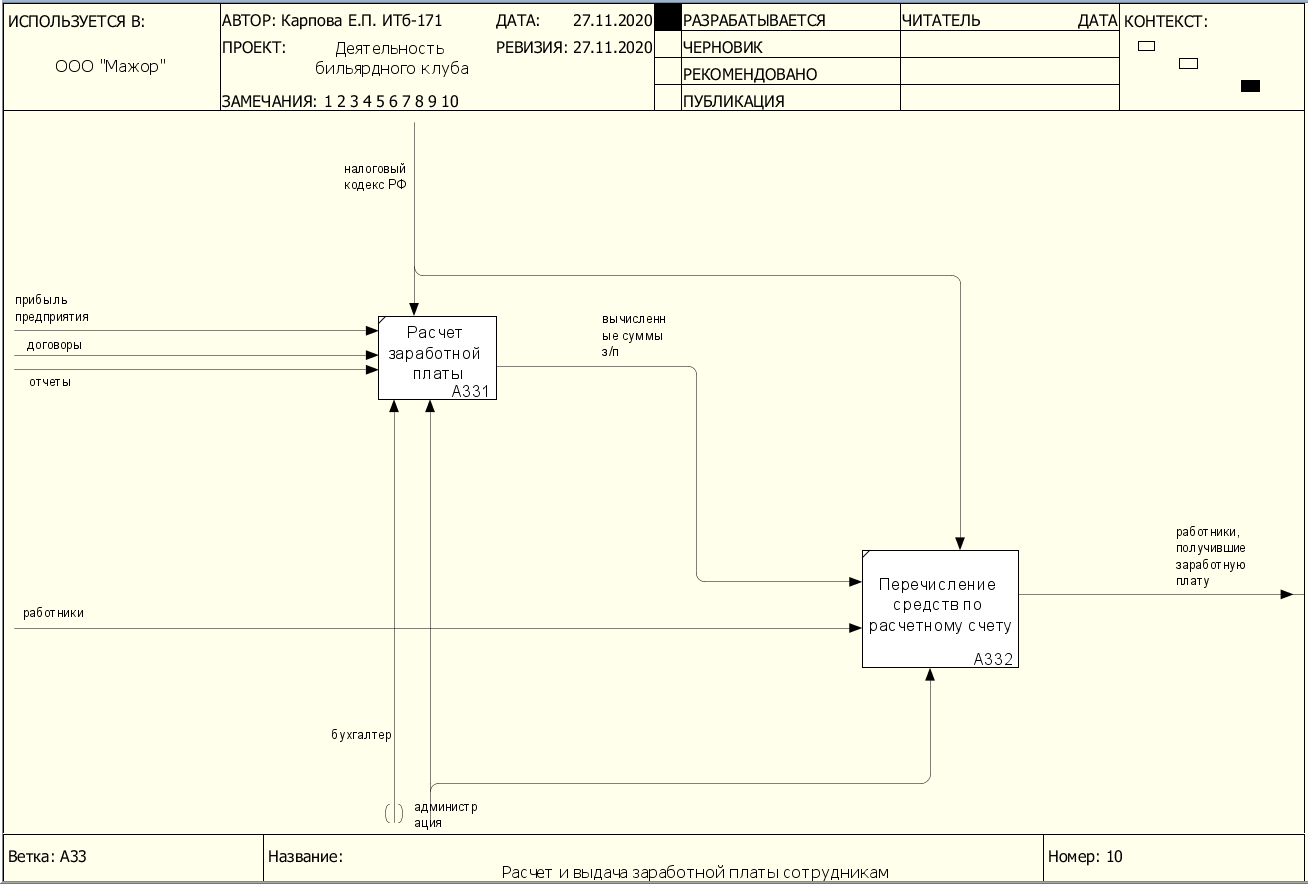
Рассмотрим деятельность предприятия и выделим бизнес-процессы:

*  
Рис. 91. Контекстная диаграмма предприятия*

**

*Рис. 92. Декомпозиционная диаграмма первого уровня*

*  
Рис. 93. Декомпозиционная диаграмма второго уровня*

**

*Рис. 94. Декомпозиционная диаграмма третьего уровня*

Описание базового бизнес-процесса «Расчет заработной платы сотрудников». Условием для начала выполнения бизнес-процесса является окончание расчетного периода (календарного месяца). В выполнении бизнес-процесса принимают участие сотрудники бухгалтерии.

Последовательность действий выполнения бизнес-процесса:

* Бухгалтер вычисляет сумму процентной ставки в зависимости от общей суммы по официанту за месяц и процента, указанного в его трудовом договоре, сумму узнает из в программе R-keeper BackOfficeReports, все дальнейшие вычисления производятся при помощи Microsoft Excel.
* Главный бухгалтер из вычисленной суммы удерживает 13% подоходного налога, оставшаяся сумма передается бухгалтеру.
* Бухгалтер начисляет премию в зависимости от переработки по выручке официанта за месяц.
* Бухгалтер удерживает из з/п штрафы, при их наличии.
* Бухгалтер информирует работников о сумме з/п.
* Бухгалтер перечисляет сотрудникам з/п с расчетного счета.

1. **Требования к системе.**
   1. **Требования к системе в целом.**
      1. **Требования к структуре и функционированию системы**

Система ЗПО должна иметь двухуровневую архитектуру, первый уровень – импорт данных из системы R-keeper, второй – отбор и операции над нужными для расчетов данными.

Система должна обеспечивать поддержку компьютеров с Windows 2000 Professional или Windows XP и выше.

* + 1. **Требования к численности и квалификации персонала системы и режиму его работы.**

В состав персонала, необходимого для обеспечения эксплуатации ЗПО, необходимо задействовать:

- Системный администратор – 1 человек, знание системы ЗПО, режим работы – гибкий график, выезд на Предприятие для устранения неполадок и дополнительной настройки системы.

- Бухгалтер – 1 человек, знание соответствующей предметной области, режим работы в соответствии с основным рабочим графиком Заказчика.

* + 1. **Требования к надежности**

*Таблица 17*

*Требования к надежности*

|  |  |
| --- | --- |
| Вероятное условие | Требование |
| Нарушения в работе системы внешнего электроснабжения продолжительностью до 15 мин. | Функционирование в полном объеме. |
| Выход из строя компьютера с хранящимися данными | Уведомление системного администратора предприятия |

Уровень надежности должен достигаться согласованным применением организационных, организационно-технических мероприятий и программно-аппаратных средств. Надежность должна обеспечиваться за счет:

- применения технических средств, системного и базового программного обеспечения, соответствующих классу решаемых задач;

- своевременного выполнения процессов администрирования Системы ЗПО;

- соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;

- предварительного обучения пользователей.

Время устранения отказа должно быть следующим:

- при перерыве и выходе за установленные пределы параметров электропитания - не более 15 минут.

- при перерыве и выходе за установленные пределы параметров программного обеспечения - не более 2 часов.

* + 1. **Требования безопасности**

Доступ к компьютерам ограничен (идентификации доступа по логину и паролю), к сети ограничен (идентификации доступа по логину и паролю), доступ к DMS ограничен (идентификации доступа по логину и паролю), доступ к оборудованию и в служебные помещения предприятия посторонним лицам запрещен (используется система доступа по чипам).

* + 1. **Требования к патентной чистоте**.

По всем техническим и программным средствам, применяемым в системе, должны соблюдаться условия лицензионных соглашений и обеспечиваться патентная чистота

**4.1.8. Требования по стандартизации и унификации**

Разработка системы должна осуществляться с использованием стандартных методологий функционального моделирования: IDEF0, DFD и информационного моделирования IE и IDEF1Х в рамках рекомендаций по стандартизации Р50.1.028-2001 «Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования».

Моделирование должно выполняться в рамках стандартов, поддерживаемых программными средствами моделирования ERWin 4.х и BPWin. 4.х.

Для работы с БД должнен использоваться язык запросов SQL в рамках стандарта ANSI SQL-92.

Для разработки пользовательских интерфейсов и средств генерации отчетов (любых твердых копий) должны использоваться встроенные возможности ПО.

В системе должны использоваться (при необходимости) общероссийские классификаторы и единые классификаторы и словари для различных видов алфавитно-цифровой и текстовой информации.

**4.1.9. Дополнительные требования**

ЗПО должно разрабатываться и эксплуатироваться на уже имеющемся у Заказчика аппаратно-техническом комплексе. Необходимо создать отдельные самостоятельные зоны разработки и тестирования системы ЗПО. Для зоны разработки и тестирования должны использоваться те же программные средства, что и для зоны эксплуатации.

* 1. **Требование к функциям (задачам).**
     1. **Подсистема сбора, обработки и загрузки данных**

Данная подсистема загружает данные из системы R-keeper и позволяет управлять, редактировать и удалять эти данные непосредственно на ПК в течение всего периода функционирования системы. Вид – таблицы и базы данных.

* + 1. **Подсистема вычисления заработной платы**

Данная подсистема автоматически высчитывает з/п сотрудника исходя из данных в соответствующих таблицах в зависимости от текущего расчетного периода. Выдает пользователю итоговое значение.

* 1. **Требования к видам обеспечения**
     1. **Требования к математическому обеспечению.**

Не предъявляются

* + 1. **Требования к информационному обеспечению.**

В системе должна быть выделена область хранения необходимых данных. Система должна поддерживать возможность экспорта данных в смежные системы. Система должна использовать классификаторы и справочники, которые ведутся в системе-источнике данных.  
Для реализации подсистемы хранения данных должна использоваться СУБД Microsoft SQL. Информация в базе данных должна сохраняться при возникновении аварийных ситуаций, связанных со сбоями электропитания. Резервное копирование данных должно осуществляться регулярно, в объемах, достаточных для восстановления информации в подсистеме хранения данных. Хранение данных должно производиться не более чем за пять предыдущих лет, все устаревшие данные могут быть перемещены в архив.

* + 1. **Требования к лингвистическому обеспечению.**

При реализации системы должен применяться язык SQL.

* + 1. **Требования к программному обеспечению.**

На ПК должен быть установлен MS SQL Server 2008R2.

* + 1. **Требования к техническому обеспечению.**

*Таблица 18*

*Требования к техническому обеспечению*

|  |  |
| --- | --- |
| Свободного дискового пространства | от 5 Гбайт |
| Объем оперативной памяти | не менее 512 Мбайт |
| Процессор | Pentium Celeron 2000 МГц и выше |
| Операционная система | Windows 2000, XP и выше |

* + 1. **Требования к организационному обеспечению.**

Основными пользователями системы ЗПО являются сотрудники бухгалтерии.

Обеспечивает эксплуатацию Системы системный администратор, сотрудник Заказчика. Состав сотрудников каждого из подразделений определяется штатным расписанием Заказчика, которое, в случае необходимости, может изменяться.

К организации функционирования Системы ЗПО и порядку взаимодействия персонала, обеспечивающего эксплуатацию, и пользователей предъявляются следующие требования:

- в случае возникновения со стороны функционального подразделения необходимости изменения функциональности системы ЗПО, пользователи должны сообщить Разработчику свои пожелания.

- системный администратор должен заранее информировать всех пользователей системы (с указанием точного времени и продолжительности) о переходе её в профилактический режим.

К защите от ошибочных действий персонала предъявляются следующие требования:

- для всех пользователей должна быть запрещена возможность удаления преднастроенных объектов и отчетности;

- для снижения ошибочных действий пользователей должно быть разработано полное и доступное руководство пользователя.

1. **Состав и содержание работ по созданию системы.**

Работы по созданию системы выполняются в три этапа:

1. Проектирование. Разработка эскизного проекта. Разработка технического проекта (продолжительность – 2 месяца)

2. Разработка рабочей документации. Адаптация программ (продолжительность – 1 месяц).

3. Ввод в действие (продолжительность – 2 недели).

6. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

Для ввода системы в действие требуется:

– осуществить развертывание клиентской части приложения на терминальном сервере;

– осуществить развертывание серверной части приложения на серверном сервере;

– обеспечить сетевое соединение пользовательских компьютеров с терминальным сервером, обеспечить соединение терминального сервера с серверным сервером;

7. Требования к документированию.

7.1. Проектирование. Разработка эскизного проекта. Разработка технического проекта.

На данных этапах выходными являются следующие документы:

– пояснительная записка к эскизному проекту;

– пояснительная записка к техническому проекту;

– схема функциональной структуры.

7.2. Разработка рабочей документации. Адаптация программ.

На данных этапах выходными являются следующие документы:

– общее описание системы;

– технологическая инструкция;

– руководство пользователя;

– схема базы данных (набора данных);

– спецификация;

– описание программы;

– текст программы.

7.3. Ввод в действие.

На данном этапе выходными являются следующие документы:

– акт приемки в опытную эксплуатацию;

– протокол испытаний;

– акт завершения работ.

8. Источники разработки.

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

– ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»;

– ГОСТ 24.701-86 «Надежность автоматизированных систем управления»;

– ГОСТ 15150-69 «Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды»

– ГОСТ 21958-76 «Система "Человек-машина". Зал и кабины операторов. Взаимное расположение рабочих мест. Общие эргономические требования»;

– ГОСТ 12.1.004-91 «ССБТ. Пожарная безопасность. Общие требования»;

– ГОСТ Р 50571.22-2000 «Электроустановки зданий».