Пермский филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет социально-экономических и компьютерных наук

Образовательная программа бакалавриата «Бизнес-информатика»

**О Т Ч Е Т**

**по проектной практике**

Прикладной проект 2

*Разработка информационной системы для учета пожаров*

Выполнил студент БИ-21-1

Капралова Елизавета Александровна

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*(подпись)*

**Проверил:**

*Сухов Александр Олегович*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(подпись)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

Пермь, 2023

# Оглавление

[Введение 5](#_Toc136561221)

[1. Анализ, выявление и проектирование требований 8](#_Toc136561222)

[1.1 Обзор аналогов в выбранной предметной области 8](#_Toc136561223)

[1.2 Классы пользователей системы и других заинтересованных лиц 8](#_Toc136561224)

[1.3 Процесс выявления требований к разрабатываемой системе 9](#_Toc136561225)

[Бизнес - требования 10](#_Toc136561226)

[Пользовательские требования 11](#_Toc136561227)

[Функциональные требования 11](#_Toc136561228)

[Атрибуты качества 12](#_Toc136561229)

[Ограничения 12](#_Toc136561230)

[1.4 Диаграмма вариантов использования 13](#_Toc136561231)

[Спецификация прецедентов 15](#_Toc136561232)

[1.5 Диаграмма активности 23](#_Toc136561233)

[1.5.1 Процесс построения диаграмм активности 23](#_Toc136561234)

[1.6 Описание документов и справочников предметной области, используемых при разработке системы 31](#_Toc136561235)

[2. Cоздание модели данных 35](#_Toc136561236)

[2.1 Проектирование базы данных 48](#_Toc136561237)

[2.1.1 Приведение к 1 НФ 48](#_Toc136561238)

[2.1.2 Приведение к 2 НФ 51](#_Toc136561239)

[2.1.3 Приведение к 3 НФ 57](#_Toc136561240)

[3. Проектирование приложения 65](#_Toc136561241)

[3.1 Диаграммы последовательностей 65](#_Toc136561242)

[3.2 Проектирование пользовательского интерфейса 73](#_Toc136561243)

[4. Разработка запросов к данным 75](#_Toc136561244)

[5. Проектирование отчетов 77](#_Toc136561245)

[6. Тестирование приложения 80](#_Toc136561246)

[7. Заключение 85](#_Toc136561247)

[Библиографический список 86](#_Toc136561248)

[Приложение А 87](#_Toc136561249)

[Приложение В 92](#_Toc136561292)

# Введение

Одной из важнейших задач, осуществляемых государством, является задача обеспечения безопасности личности и общества в целом. Немаловажный аспект подобной деятельности – защита граждан от пожаров. Пожарная безопасность в Российской Федерации осуществляется Государственной противопожарной службой МЧС России, в состав которой входит федеральная противопожарная служба и противопожарная служба субъектов Российской Федерации.

Согласно нормативно-правовой базе, регулирующей область пожарной охраны, любое возникновение пожара со всеми его последствиями должно быть строго зафиксировано. Это позволяет вести статистику по количеству пожаров, выявлять основные причины их возникновения и в конце концов корректировать государственную политику для достижения большей эффективности в области пожарной безопасности.

Фиксация всей необходимой информации о пожаре происходит путём составления донесения в строго установленной форме. Однако первая проблема заключается в том, что такие донесения составляются только лишь вручную, что не исключает появление возможных ошибок при заполнении и увеличивает время внесения всех нужных данных сотрудником МЧС. Вторая проблема состоит в том, что такие донесения сложно обрабатывать вручную: достаточно проблематично собирать по ним статистику и искать какую-либо информацию.

Чтобы облегчить процесс составления донесений о пожарах, нужно создать приложение для генерации донесений на основе сведений о произошедшем пожаре. Таким образом, **актуальность** данной темы обусловлена отсутствием в существующем программном решении генерации донесений о пожарах, которая позволит ускорить процесс формирования этих донесений, что в итоге сделает деятельность по ведению документации более эффективной.

**Целью проекта является** создание приложения, которое позволит генерировать донесения о пожарах в форме, установленной нормативно-правовой базой предметной области, а также хранить и искать необходимые данные о пожарах.

Для достижения поставленной цели были выявлены следующие **задачи**:

1. Рассмотрение существующих программных решений, помогающих формировать донесения о пожарах, выявление достоинств и недостатков рассматриваемых решений.
2. Выявление и формализация требований к системе, автоматизирующей процесс составления донесений о пожарах.
3. Описание бизнес-процессов, которые будет автоматизировать приложение.
4. Проектирование приложения, выбор технологий для его реализации.
5. Реализация приложения.
6. Тестирование и оценка качества разработанного приложения.
7. Написание документации для разработанного приложения.

**Объектом исследования** является процесс формирования донесения о пожаре в форме, установленной нормативно-правовой базой предметной области.

**Предмет исследования** – средства автоматизации данного процесса.

**Методы исследования:**

* метод глубинного интервью;
* методы сбора данных в сети Internet;
* анализ нормативно-правовых актов и иных собранных материалов по теме исследования;
* методы объектно-ориентированного проектирования и программирования.

**Характеристика источников.** Для изучения данной темы были использованы такие источники как: федеральные законы Российской Федерации [1], указы Президента Российской Федерации [2] и приказы Министерства Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС России) [3, 4, 5]. По окончании изучения материала, было замечено, что все представленные выше источники содержали в себе достаточно обширную и достоверную информацию по выбранной теме.

**Степень разработанности темы.** В процессе выполнения проекта был проведен анализ систем, позволяющих автоматизировать процесс учета пожаров и их последствий. Существует "Автоматизированная аналитическая система поддержки и управления контрольно-надзорными органами МЧС России", разработанная департаментом надзорной деятельности и профилактической работы МЧС России [6]. Она предоставляет сотрудникам МЧС доступ к широкому функционалу, однако в то же время она не предусматривает автоматизацию процессов составления донесений о пожарах. Следовательно, необходимость создания качественного программного продукта существует до сих пор.

**Практическая значимость** **курсовой работы** заключается в том, что реализуемое приложение может значительно упростить процесс формирования донесений о пожарах и сократить время на его выполнение. Программное решение может быть использовано как база для дальнейшего развития существующей системы, эксплуатируемой пожарными частями Российской Федерации.

**Структура работы**: данная работа состоит из введения, 6 глав, заключения, библиографического списка и 2 приложений.

**В первой главе** описывается процесс анализа предметной области (пожарная безопасность, работа пожарных частей) и рассматриваются существующие аналоги, а также выявляются основные требования к разрабатываемой системе и описываются процесс формализации данных требований и автоматизируемые бизнес-процессы.

**Во второй главе** выполняется описание процесса проектирования: определение поведения системы, создание модели данных.

**Третья глава** содержит информацию о проектировании приложения, создания пользовательского интерфейса, реализации операций ввода/редактирования/удаления данных, импорта данных из файлов.

**Четвертая глава** содержит информацию о разработке запросов к данным. Реализуются параметризованные запросы, запросы на удаление.

**Пятая глава** содержит информацию о процессе формировании донесения о пожаре (используемый шаблон, итоговый вариант донесения)

**Шестая глава** содержит информацию о тестировании приложения.

В **заключении** формулируются достигнутые результаты исследования.

# 1. Анализ, выявление и проектирование требований

## 1.1 Обзор аналогов в выбранной предметной области

Существует "Автоматизированная аналитическая система поддержки и управления контрольно-надзорными органами МЧС России", разработанная департаментом надзорной деятельности и профилактической работы МЧС России [6].

Она позволяет сократить время проведения контрольных мероприятий, повысив их результативность и прозрачность.

Программа обеспечивает взаимодействие в электронном виде с единым порталом государственных услуг, государственной информационной системой о государственных и муниципальных платежах, единым реестром проверок, системой досудебного обжалования, государственной автоматизированной системой «Управление» и заинтересованными федеральными органами исполнительной власти через систему межведомственного информационного взаимодействия.

Таким образом, данная система предназначена для автоматизации только контрольно-надзорных мероприятий. Она не позволяет сделать процесс работы пожарных частей, а именно процесс формирования донесений о пожарах более быстрым и с минимальным количеством ошибок.

## 1.2 Классы пользователей системы и других заинтересованных лиц

Информационная система предоставляет возможность работы нескольким категориям пользователей:

1. Диспетчеры пожарных частей.
2. Начальники караула пожарных частей.
3. Заместители начальников караула пожарных частей.
4. Начальник пожарной части.
5. Заместитель начальника пожарной части.
6. Сотрудник отдела кадров.

В самой информационной системе для учета пожаров данные категории пользователей относятся к одной категории - «сотрудник МЧС».

Помимо выделенных выше категорий пользователей, можно также обозначить другие заинтересованные в проекте лица (стейкхолдеры). Заинтересованные лица могут быть внутренними или внешними по отношению к команде проекта или разрабатывающей организации:

1. Внутренние:

* владелец продукта;
* разработчик;
* аналитик процессов;
* тестировщик;
* администратор баз данных;
* аналитик инфраструктуры
* ИТ-архитектор.

Следует отметить, что в рамках выполняемого курсового проекта все внутренние стейкхолдеры будут представлены в лице одного человека - автора курсовой работы, поскольку проект выполняется самостоятельно.

1. Внешние:

* прямые пользователи;
* государственные учреждения (пожарные части Российской Федерации);
* федеральные органы государственной власти в области пожарной безопасности.

## 1.3 Процесс выявления требований к разрабатываемой системе

Требование – условие или возможность, которой должна удовлетворять система.[[1]](#footnote-1)

Имеется много методов и технологий выявления требований. Они включают:

* интервьюирование пользователей и экспертов в предметной области;
* анкетные опросы пользователей;
* наблюдение, как пользователи выполняют свои задачи;
* изучение существующих документов системы;
* изучение подобных систем ПО, чтобы выяснить состояние в предметной области;
* изучение опытных образцов рабочих моделей для определения и подтверждения требований;
* объединенные совещания разработчиков и клиентов по разработке приложения.

Для выявления требований к разрабатываемой информационной системе, автором был выбран метод интервьюирования эксперта в области пожарной безопасности (сотрудника действующей пожарной части города Перми, занимающего должность начальника караула). В ходе интервью было выяснено, что вся отчетность (в том числе и донесения о пожарах) составляются вручную, затем печатаются, относятся на подпись к начальству, а затем повторно заносятся в базу данных. Таким образом, тратится лишнее время на составление отчетов и донесений, что делает процесс ведения документации неэффективным.

По результатам ответов респондента М. были составлены требования к разрабатываемой информационной системе, которые впоследствии подверглись классификации по Вигерсу.

Требования к ПО состоят из трех уровней — бизнес-требования, пользовательские и функциональные требования. Бизнес-требования (business requirements) описывают, почему организации нужна такая система, то есть цели, которые организация намерена достичь с ее помощью. Пользовательские требования (user requirements) описывают цели или задачи, которые пользователи должны иметь возможность выполнять с помощью продукта, который в свою очередь должен приносить пользу кому-то. Функциональные требования (functional requirements) определяют, каким должно быть поведение продукта в тех или иных условиях[[2]](#footnote-2).

### Бизнес - требования

Бизнес - требования выстраиваются на основе бизнес-правил (политика, предписание, стандарт или правило, определяющее или ограничивающее некоторые стороны бизнес-процессов).

Было выделено следующие бизнес - требование: уменьшить время и количество ошибок при формировании донесений о пожаре сотрудниками пожарных частей.

Данное бизнес - требование подчиняется бизнес-правилу: донесения о пожаре выгружаются в определенном формате (docx).

### Пользовательские требования

Были сформулированы следующие пользовательские требования:

1. Сотрудник МЧС должен иметь возможность заходить в систему под именем пользователя и паролем.
2. Начальник караула пожарной части должен иметь возможность заполнять данные о произошедшем пожаре (донесение).
3. Заместитель начальника караула пожарной части в случае отсутствия начальника караула должен иметь возможность заполнять данные о произошедшем пожаре (донесение).
4. Сотрудник отдела кадров пожарной части должен иметь возможность заполнять данные о новых сотрудниках (как персональные данные, так и профессиональные, например, номер караула, за которым сотрудник закреплен).
5. Сотрудник отдела кадров пожарной части должен иметь возможность заполнять данные о допусках водителей к пожарной машине.
6. Начальник пожарной части должен иметь возможность заполнять данные о пожарной части.
7. Заместитель начальника пожарной части в случае отсутствия начальника части должен иметь возможность внести информацию о пожарной части.
8. Диспетчер пожарной части должен иметь возможность вносить в базу данных сведения о возникновении пожара, выездах на пожары и дежурных сменах.

### Функциональные требования

Были сформулированы следующие функциональные требования:

1. У пользователя должна быть возможность войти в систему, используя свой логин и пароль.
2. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о новом сотруднике в систему.
3. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о вызовах на пожар.
4. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о выездах на пожар.
5. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о дежурных сменах.
6. У пользователя должна быть возможность формировать донесения о пожарах с последующим выводом этого донесения в Word.
7. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о пожарных частях и пожарных машинах.

### Атрибуты качества

Были сформулированы следующие атрибуты качества:

1. Система должна обслуживать одновременно не менее 500 пользователей (производительность).
2. Система должна демонстрировать уровень доступности с коэффициентом доступности не хуже, чем 93% в час (доступность).
3. Система должна демонстрировать уровень надёжности, при котором вероятность сбоя при обращении к её функциям не превышает 5% (надежность).

### Ограничения

Были сформулированы следующие ограничения:

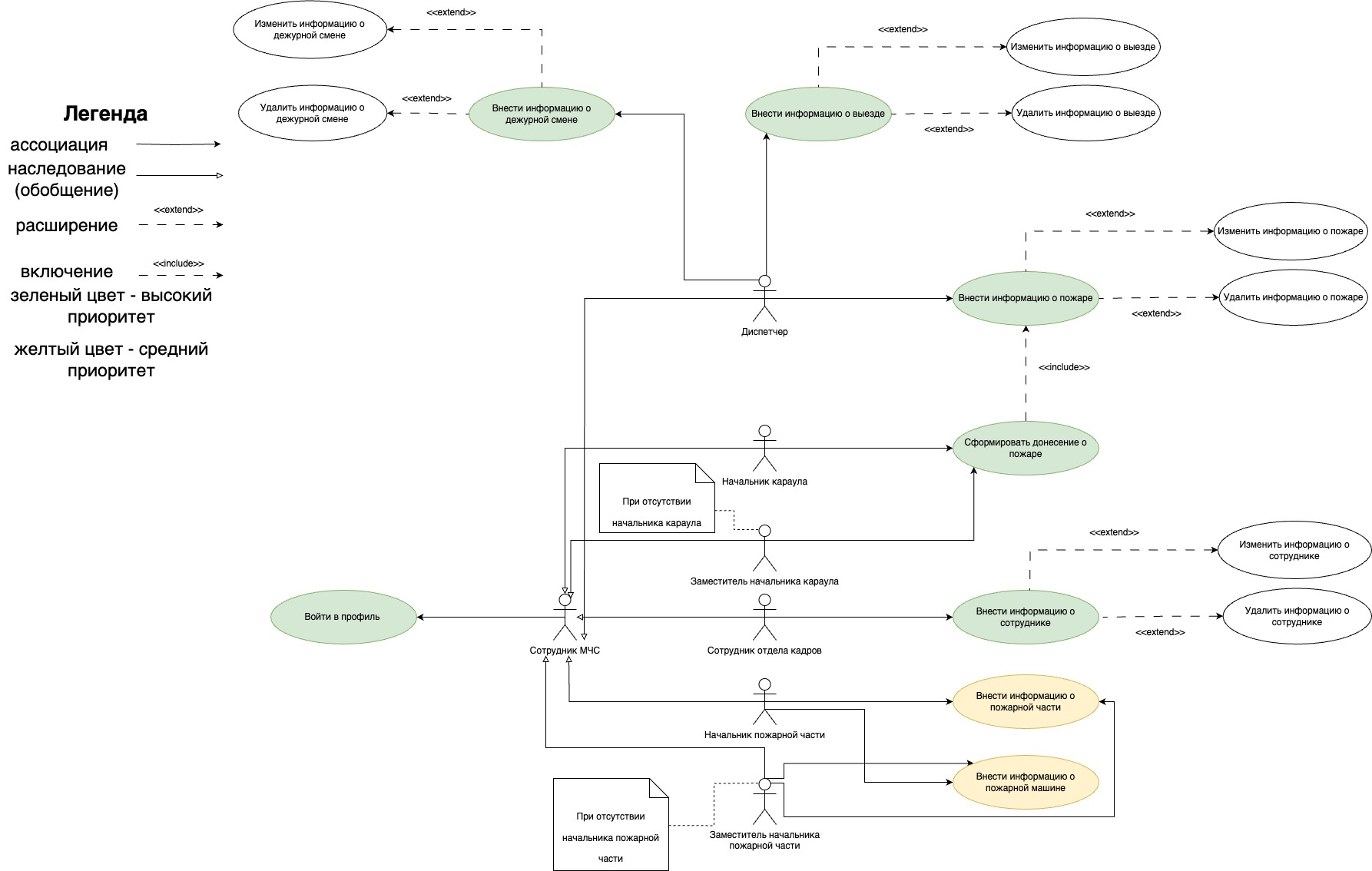
1. Доступ в систему возможен только по логину и паролю.
2. Система не поддерживает доступ с мобильных устройств.

После выявления требований к системе, они были расположены по степени важности:

1. У пользователя должна быть возможность войти в систему, используя свой логин и пароль.
2. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о вызовах на пожар.
3. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о выездах на пожар.
4. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о дежурных сменах.
5. У пользователя должна быть возможность формировать донесения о пожарах с последующим выводом этого донесения в Word.
6. У пользователя должна быть возможность добавлять, редактировать, просматривать, удалять данные о новом сотруднике в систему.
7. У пользователя должна быть возможность добавлять данные о пожарных частях и пожарных машинах.

## 1.4 Диаграмма вариантов использования

Для моделирования системы с точки зрения внешнего наблюдателя была построена диаграмма прецедентов с расширенным описанием всех прецедентов (см.рис.1).



***Рисунок 1 - Диаграмма прецедентов***

### Спецификация прецедентов

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-1. Войти в профиль | | |
| **Автор** | Капралова Е.А. | **Дата** | 14.03.2023 |
| **Актеры** | Основной: Сотрудник МЧС | | |
| **Описание** | Прецедент дает возможность Сотруднику МЧС войти в профиль в приложении. Вход в профиль подразумевает ввод логина и пароля. После ввода этих данных система выполняет вход в профиль Сотрудника МЧС. | | |
| **Триггер** | Клиент запускает приложение | | |
| **Предварительные условия** | PRE-1. Сотрудник МЧС получил свой логин и пароль | | |
| **Выходные условия** | POST-1. Вход в профиль совершен | | |
| **Основной поток** | 1.0 Войти в профиль  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1). | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Высокий | | |
| **Частота использования** | примерно 3 раза в день | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-2. Внести информацию о пожаре | | |
| **Автор** | Капралова Е.А. | **Дата** | 14.03.2023 |
| **Актеры** | Основной: Диспетчер | | |
| **Описание** | Прецедент дает возможность Диспетчеру заполнить всю необходимую информацию о пожаре. После ввода этих данных система сохраняет их в БД. | | |
| **Триггер** | Диспетчер заступает на смену, в течение которой поступают вызовы на пожар | | |
| **Предварительные условия** | PRE-1. Диспетчер зашел в свой профиль  PRE-2. База данных доступна | | |
| **Выходные условия** | POST-1. Информация о пожаре сохранена в БД | | |
| **Основной поток** | 1.0 Внести информацию о пожаре  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию "Вызов на пожар" и заполняет всю информацию (см. 1.0.Е2).  5. При поступлении вызова на пожар, пользователь заполняет необходимую информацию и выбирает функцию "Сохранить" для сохранения введенных данных (см. 1.0.Е3).  6. Система запоминает информацию в БД. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е2. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е3. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Высокий | | |
| **Частота использования** | более 5 раз в день | | |
| **Точки расширения** | UC-3. Изменить информацию о пожаре  UC-4. Удалить информацию о пожаре | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-5. Сформировать донесение о пожаре | | |
| **Автор** | Капралова Е.А. | **Дата** | 14.03.2023 |
| **Актеры** | Основной: Сотрудник МЧС | | |
| **Описание** | Прецедент дает возможность Сотруднику МЧС формировать донесения о пожарах. Составление донесения включает информацию о пожаре, введенную ранее диспетчером. После система сохраняет донесение в БД. | | |
| **Триггер** | Пожар завершен | | |
| **Предварительные условия** | PRE-1. Сотрудник МЧС зашел в свой профиль  PRE-2. База данных доступна  PRE-3. Диспетчер внес все необходимые данные о пожаре и дежурной смене | | |
| **Выходные условия** | POST-1. Донесение о пожаре сформировано | | |
| **Основной поток** | 1.0 Заполнить донесение о пожаре  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию "Отчет"  5. Система выгружает необходимые данные в Word. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Высокий | | |
| **Частота использования** | более 5 раз в день | | |
| **Точка включения** | UC-2. Внести информацию о пожаре | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-6. Внести информацию о сотруднике | | |
| **Автор** | Капралова Е.А. | **Дата** | 14.03.2023 |
| **Актеры** | Основной: Сотрудник отдела кадров | | |
| **Описание** | Прецедент дает возможность Сотруднику отдела кадров заполнить всю необходимую информацию о сотруднике МЧС. После ввода этих данных система сохраняет их в БД. | | |
| **Триггер** | 1. Новый сотрудник принят на работу  2. Потребность обновить данные у старых сотрудников | | |
| **Предварительные условия** | PRE-1. Сотрудник отдела кадров зашел в свой профиль  PRE-2. База данных доступна | | |
| **Выходные условия** | POST-1. Информация о сотруднике сохранена в БД | | |
| **Основной поток** | 1.0 Внести информацию о сотруднике  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию "Сотрудник", "Контракт сотрудника", "Допуск к пожарной машине" и заполняет всю информацию, если это необходимо.  5. Пользователь выбирает функцию "Сохранить" в каждой таблице для сохранения введенных данных (см. 1.0.Е2).  6. Система запоминает информацию в БД. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е2. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Высокий | | |
| **Частота использования** | Примерно 5 раз в месяц | | |
| **Точки расширения** | UC-7. Изменить информацию о сотруднике  UC-8. Удалить информацию о сотруднике | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-9. Внести информацию о выезде | | |
| **Автор** | Капралова Е.А. | **Автор** | Капралова Е.А. |
| **Актеры** | Основной: Диспетчер | | |
| **Описание** | Прецедент дает возможность Диспетчеру заполнить всю необходимую информацию о выезде. После ввода этих данных система сохраняет их в БД. | | |
| **Триггер** | Диспетчер заступает на смену, в течение которой поступают вызовы на пожар, у которых нужно заполнить информацию о выездах | | |
| **Предварительные условия** | PRE-1. Диспетчер зашел в свой профиль  PRE-2. База данных доступна | | |
| **Выходные условия** | POST-1. Информация о выезде сохранена в БД | | |
| **Основной поток** | 1.0 Внести информацию о выезде  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию "Выезд на пожар" и заполняет всю информацию (см. 1.0.Е2).  5. После поступления вызова на пожар, пользователь заполняет необходимую информацию о выезде и выбирает функцию "Сохранить" для сохранения введенных данных (см. 1.0.Е3).  6. Система запоминает информацию в БД. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е2. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е3. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Высокий | | |
| **Частота использования** | более 5 раз в день | | |
| **Точки расширения** | UC-10. Изменить информацию о выезде  UC-11. Удалить информацию о выезде | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-12. Внести информацию о дежурной смене | | |
| **Автор** | Капралова Е.А. | **Автор** | Капралова Е.А. |
| **Актеры** | Основной: Диспетчер | | |
| **Описание** | Прецедент дает возможность Диспетчеру заполнить всю необходимую информацию о дежурной смене. После ввода этих данных система сохраняет их в БД. | | |
| **Триггер** | Диспетчер заступает на смену и вводит все необходимые данные о ней | | |
| **Предварительные условия** | PRE-1. Диспетчер зашел в свой профиль  PRE-2. База данных доступна | | |
| **Выходные условия** | POST-1. Информация о дежурной смене сохранена в БД | | |
| **Основной поток** | 1.0 Внести информацию о дежурной смене  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию "Дежурная смена" и заполняет всю информацию (см. 1.0.Е2).  5. После поступления вызова на пожар, пользователь заполняет необходимую информацию о выезде и выбирает функцию "Сохранить" для сохранения введенных данных (см. 1.0.Е3).  6. Система запоминает информацию в БД. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е2. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля.  1.0.Е3. Пользователь не ввел всю обязательную информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Высокий | | |
| **Частота использования** | более 5 раз в день | | |
| **Точки расширения** | UC-13. Изменить информацию о дежурной смене  UC-14. Удалить информацию о дежурной смене | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-15. Внести информацию о пожарной части | | |
| **Автор** | Капралова Е.А. | **Дата** | 14.03.2023 |
| **Актеры** | Основной: Начальник пожарной части | | |
| **Описание** | Прецедент дает возможность Начальнику пожарной части (а в его отсутствие заместителю начальника пожарной части) вносить информацию о пожарной части. После ввода этих данных система сохраняет их в БД. | | |
| **Триггер** | Необходимость заполнить справочную информацию о данной пожарной части или обновить уже имеющуюся | | |
| **Предварительные условия** | PRE-1. Начальник пожарной части зашел в свой профиль  PRE-2. База данных доступна | | |
| **Выходные условия** | POST-1. Информация о пожарной части заполнена | | |
| **Основной поток** | 1.0 Внести информацию о пожарной части  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию заполнения информации о пожарной части.  5. Пользователь вводит или изменяет какую-либо информацию, затем выбирает функцию "Сохранить".  6. Система запоминает информацию в БД. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Средний | | |
| **Частота использования** | Примерно 2 раза в полгода | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор и название** | UC-16. Внести информацию о пожарной машине | | |
| **Автор** | Капралова Е.А. | **Дата** | 14.03.2023 |
| **Актеры** | Основной: Начальник пожарной части | | |
| **Описание** | Прецедент дает возможность Начальнику пожарной части (а в его отсутствие заместителю начальника пожарной части) вносить информацию о пожарной части. После ввода этих данных система сохраняет их в БД. | | |
| **Триггер** | Необходимость заполнить информацию о пожарной машине или обновить уже имеющуюся | | |
| **Предварительные условия** | PRE-1. Начальник пожарной части зашел в свой профиль  PRE-2. База данных доступна | | |
| **Выходные условия** | POST-1. Информация о пожарной машине заполнена | | |
| **Основной поток** | 1.0 Внести информацию о пожарной машине  1. Пользователь открывает приложение.  2. Пользователь вводит свой логин и пароль и нажимает на кнопку "войти" (см. 1.0.Е1).  3. Если введенный логин и пароль верны, то выполняется вход в профиль сотрудника (см. 1.1).  4. Пользователь выбирает функцию заполнения информации о пожарной машине.  5. Пользователь вводит или изменяет какую-либо информацию, затем выбирает функцию "Сохранить".  6. Система запоминает информацию в БД. | | |
| **Альтернативные потоки** | 1.1. Введенный логин и пароль не верны  1. Система выводит сообщение о том, что введенный логин или пароль не верны, необходимо повторно ввести данные. | | |
| **Исключения** | 1.0.Е1. Пользователь не ввел всю обязательную для входа информацию  1. Система выводит сообщение о том, что надо заполнить все обязательные поля. | | |
| **Приоритет** | Средний | | |
| **Частота использования** | Примерно 2 раза в полгода | | |

## 1.5 Диаграмма активности

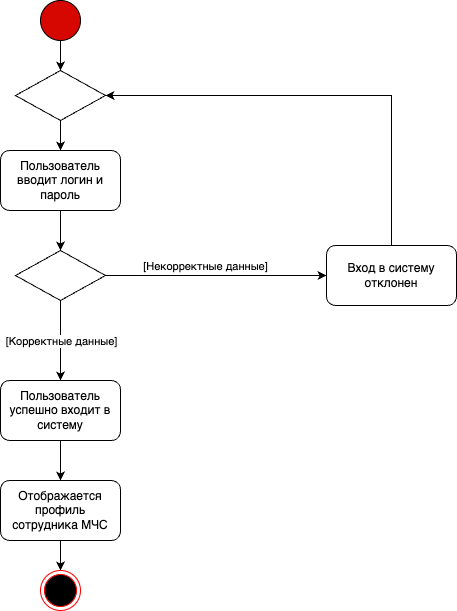
### 1.5.1 Процесс построения диаграмм активности

Для построения диаграмм активности были выбраны прецеденты, отображенные на диаграмме взаимодействия, а именно:

1. Вход в профиль.
2. Внести информации о пожаре.
3. Внести информацию о выезде.
4. Внести информацию о дежурной смене.
5. Внести информацию о сотруднике.
6. Сформировать донесение о пожаре.
7. Внести информацию о пожарной части.
8. Внести информацию о пожарной машине.

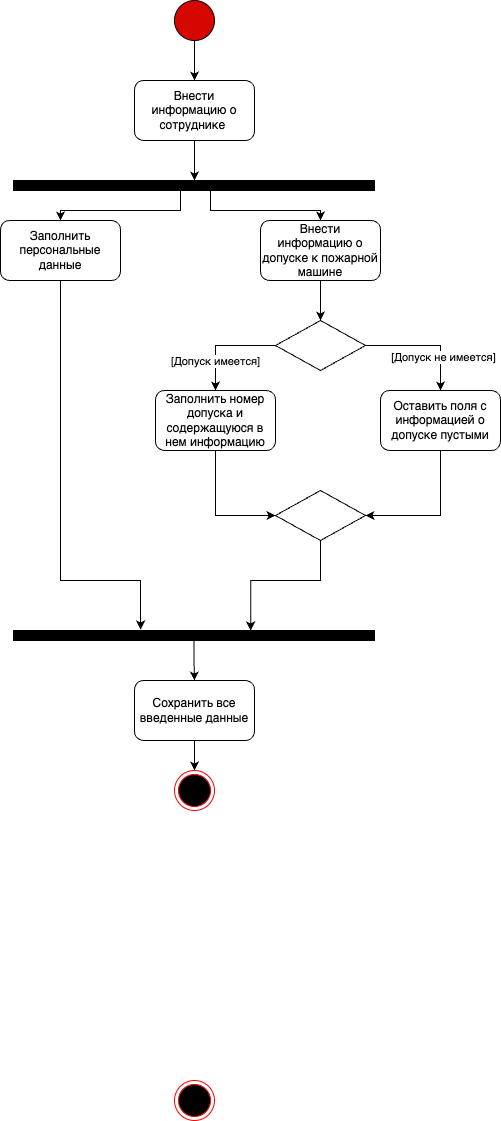
Для каждого из представленных выше прецедентов была разработана своя собственная диаграмма активности, позволяющая описать алгоритм действий (логику поведения) системы.

Рассмотрим диаграмму, построенную для прецедента "Вход в профиль" (см.рис.2). Начинаясь, поток переходит в узел слияния, который имеет два входа (первый - начало потока, второй - после отклонения входа в систему) и один выход, после которого происходит операция "Пользователь вводит логин и пароль". После выполнения этой операции используется узел принятия решений с одним входом и двумя выходами. Первый выход происходит, если данные введены корректно, что приводит к операции "Пользователь успешно вошел в систему". Второй же выход случается, когда данные некорректны, что приводит к отклонению входа в систему. После успешного входа в систему, происходит операция отображения профиля сотрудника МЧС. После этого поток завершается.



***Рисунок 2 - Диаграмма активности для прецедента "Вход в профиль"***

Рассмотрим диаграмму активности для прецедента "Внести информацию о сотруднике" (см.рис.3). Начинается поток с операции внесения информации о сотруднике. Затем открывается поток параллельных действий. Первое из параллельных действий - это заполнение персональных данных, а второе - внесение информации о допуске к пожарной машине. Второе параллельное действие переходит в узел принятия решений с одним входом и двумя выходами. Первый выход происходит, если у сотрудника имеется допуск к пожарной машине, что приводит к операции "Заполнить номер допуска и содержащуюся в нем информацию". Второй же выход случается, когда у сотрудника не имеется допуск, что приводит к операции "Оставить поля с информацией о допуске пустыми". По окончании потока параллельных действий выполняется операция сохранения введенных данных и весь поток завершается.



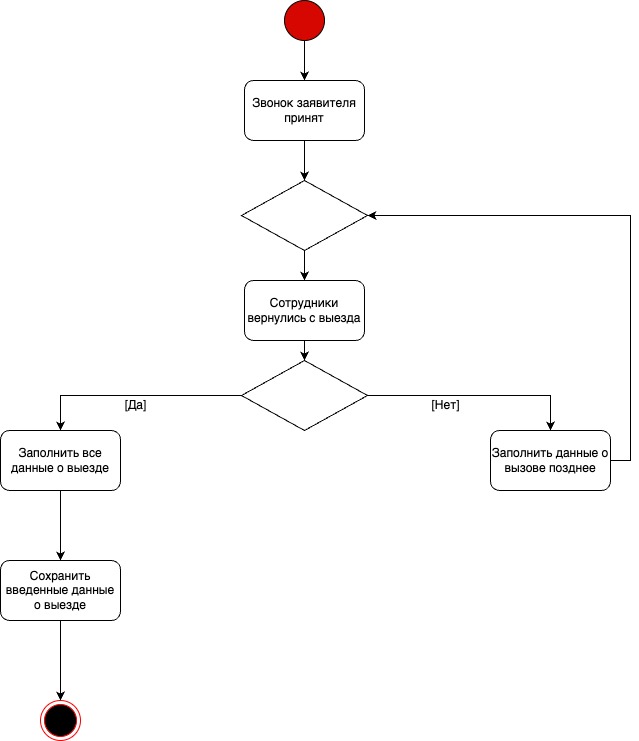
***Рисунок 3 - Диаграмма активности для прецедента "Внести информации о сотруднике"***

Перейдем к рассмотрению диаграммы активности для прецедента "Внести информацию о пожаре" (см.рис.4). Поток начинается с операции "Принять звонок заявителя", а затем происходит заполнение информации о пожаре со слов заявителя. Далее происходит передача полученной информации о пожаре тому наряду, который находится на смене. По окончании потока параллельных действий выполняется операция сохранения введенных данных и весь поток завершается.



***Рисунок 4 - Диаграмма активности для прецедента "Внести информации о пожаре"***

Рассмотрим диаграмму, построенную для прецедента "Внести информацию о выезде" (см.рис.5). Поток начинается с того, что звонок заявителя принят. Затем поток переходит в узел слияния, который имеет два входа (первый - поток после принятия звонка заявителя, второй - после принятия решения заполнить данные позднее) и один выход, после которого происходит действие "Сотрудники вернулись с выезда". После выполнения этого действия используется узел принятия решений с одним входом и двумя выходами. Первый выход происходит, если сотрудники вернулись, что приводит к операции "Заполнить все данные о выезде". Второй же выход случается, когда сотрудники пока не вернулись, что приводит к заполнению данных позже. После успешного внесения данных в систему происходит операция сохранения введенных данных о выезде. После этого поток завершается.



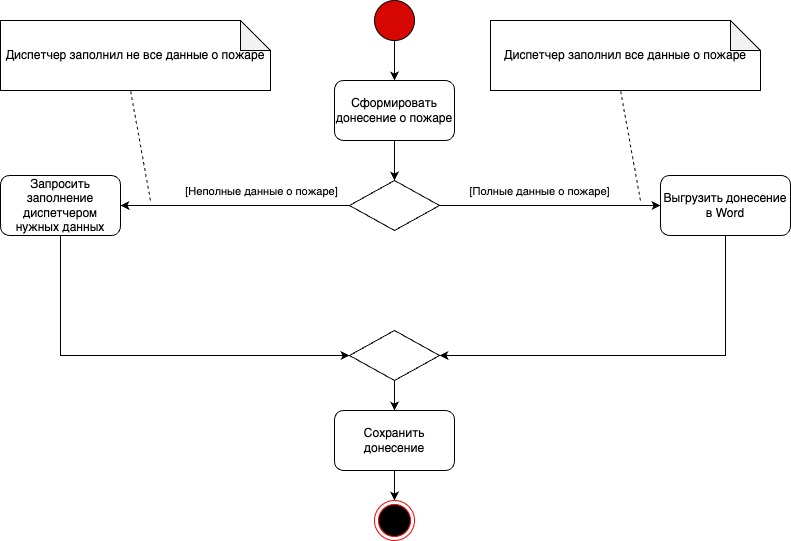
***Рисунок 5 - Диаграмма активности для прецедента "Внести информации о выезде"***

Перейдем к рассмотрению диаграммы активности для прецедента "Внести информацию о дежурной смене" (см.рис.6). Поток начинается с операции "Войти в систему", а затем происходит заполнение информации о дежурной смене. Далее происходит сохранение введенных данных и весь поток завершается.



***Рисунок 6 - Диаграмма активности для прецедента "Внести информации о дежурной смене"***

Перейдем к рассмотрению диаграммы активности для прецедента "Заполнить донесение о пожаре" (см.рис.7). Поток начинается с операции "Сформировать донесение о пожаре", а затем переходит в узел принятия решений с одним входом и двумя выходами. Первый выход происходит, если у сотрудника МЧС, заполняющего это донесение, есть не все данные о пожаре (то есть диспетчер заполнил не всю информацию при поступлении звонка), что приводит к операции "Запросить заполнение диспетчером нужных данных". Второй же выход случается, когда у сотрудника МЧС есть вся необходимая информация для формирования донесения, что приводит к операции "Выгрузить донесение в Word. После выхода из узла принятия решений выполняется операция сохранения донесения, а потом весь поток завершается.



***Рисунок 7 - Диаграмма активности для прецедента "Сформировать донесение о пожаре"***

Перейдем к рассмотрению диаграммы активности для прецедента "Внести информацию о дежурной смене" (см.рис.8). Поток начинается с операции "Войти в систему", а затем происходит заполнение информации о пожарной части. Далее происходит сохранение введенных данных и весь поток завершается.



***Рисунок 8 - Диаграмма активности для прецедента "Внести информации о пожарной части"***

Перейдем к рассмотрению диаграммы активности для прецедента "Внести информацию о дежурной смене" (см.рис.9). Поток начинается с операции "Войти в систему", а затем происходит заполнение информации о пожарной машине. Далее происходит сохранение введенных данных и весь поток завершается.



***Рисунок 9 - Диаграмма активности для прецедента "Внести информации о пожарной машине"***

## 1.6 Описание документов и справочников предметной области, используемых при разработке системы

При разработке системы были использованы документы сферы обеспечения пожарной безопасности, а также общероссийский классификатор (ОКИН).

1. **Общероссийский классификатор информации о населении (ОКИН)**

ОКИН предназначен для использования при сборе, обработке и анализе демографической, социальной и экономической информации о населении, решения задач учета, анализа и подготовки кадров предприятиями, учреждениями и организациями всех форм собственности.

ОКИН состоит из фасетов, которые можно использовать независимо друг от друга при решении различных задач.

Каждый фасет структурно состоит из двух блоков: идентификации и наименования.

В курсовой работе было использовано два фасета данного классификатора: фасет 01 "Пол" и фасет 30 "Образование".

*Классификация фасета 01 «Пол»:*

* 01 – Мужской;
* 02 – Женский.

*Классификация фасета 30 «Образование»:*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 01 - Дошкольное образование   |  | | --- | | 02 - Начальное общее образование | | 03 - Основное общее образование | | 04 - Среднее общее образование | | 05 - Среднее профессиональное образование | | 06 - Высшее образование - бакалавриат | | 07 - Высшее образование - специалитет, магистратура | | 08 - Высшее образование - подготовка кадров высшей квалификации | | 10 - Профессиональное обучение | | 11 - Дополнительное образование детей и взрослых  12 - Дополнительное профессиональное образование | |

1. **Федеральный закон "О службе в федеральной противопожарной службе Государственной противопожарной службы и внесении изменений в отдельные законодательные акты Российской Федерации" от 23.05.2016 N 141-ФЗ**

Для курсовой работы из данного документа был взят перечень званий сотрудников противопожарной службы, а именно:

* Рядовой;
* Младший сержант;
* Сержант;
* Старший сержант;
* Старшина;
* Прапорщик;
* Старший прапорщик;
* Младший лейтенант;
* Лейтенант;
* Старший лейтенант;
* Капитан;
* Майор;
* Подполковник;
* Полковник.

1. **Указ Президента РФ от 26 июня 2020 г. № 424 "Об утверждении перечня типовых должностей в федеральной противопожарной службе Государственной противопожарной службы и соответствующих этим должностям специальных званий”**

Для курсовой работы из данного документа был взят перечень должностей сотрудников противопожарной службы, а именно:

* Начальника караула;
* Помощник начальника караула;
* Командир отделения;
* Старший пожарный;
* Пожарный;
* Водитель;
* Диспетчер.

1. **Приказ №625 от 24.12.2018 Министерства Российской федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС России)**

Для курсовой работы из данного документа были взяты таблицы "Вид объекта пожара", "Причина пожара", "Виновное лицо в возникновении пожара", "Вид техники". Вся информация, используемая при разработке информационной системы, была загружена в базу данных из вышеперечисленных таблиц.

Таким образом, по окончании анализа предметной области, задач и разработки требований к системе, было составлено техническое задание на разработку информационной системы, оформленное в соответствии с ГОСТ 19.201-78 (ПРИЛОЖЕНИЕ А).

# 2. Cоздание модели данных

Для выполнения описанных выше операций по автоматизации учёта вызовов на пожары необходимо, чтобы в БД хранилась информация, описанная в таблице 2.

Таблица 2. Таблица данных

| **Имя атрибута** | **Тип данных** | **Значение по умолчанию** | **Формат ввода (маска)** | **Ограничение на значения атрибута** |
| --- | --- | --- | --- | --- |
| Табельный номер | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Фамилия сотрудника | Строка | нет | нет | Обязательное поле  Длина поля не более 50 символов |
| Имя сотрудника | Строка | нет | нет | Обязательное поле  Длина поля не более 50 символов |
| Отчество сотрудника | Строка | нет | нет | Необязательное поле  Длина поля не более 50 символов |
| Код пола | Строка | нет | Код ОКИН  00 | Обязательное поле |
| Наименование пола | Строка | нет | нет | Обязательное поле, длина поля не более 7 символов |
| Дата рождения | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Контактный номер телефона | Строка | нет | +7 (000) 000-00-00 | Обязательное поле  Длина поля не более 11 символов |
| Адрес регистрации | Строка | нет | нет | Обязательное поле  Длина поля не более 255 символов |
| Адрес фактического проживания | Строка | нет | нет | Обязательное поле  Длина поля не более 255 символов |
| Серия паспорта | Строка | нет | нет | Обязательное поле  Длина поля не более 4 символов |
| Номер паспорта | Строка | нет | нет | Обязательное поле  Длина поля не более 6 символов |
| Код\_уровня\_образования | Целое число | нет | Код ОКИН | Обязательное поле |
| Наименование уровня образования | Строка | нет | нет | Длина поля не более 50 символов  Обязательное поле |
| Серия документа | Строка | нет | нет | Длина поля не более 50 символов  Обязательное поле |
| Номер документа | Строка | нет | нет | Длина поля не более 50 символов  Обязательное поле |
| Наименование учебного заведения | Строка | нет | нет | Длина поля не более 255 символов  Обязательное поле |
| Код звания | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование звания | Строка | нет | нет   |  | | --- | |  | | Длина поля не более 50 символов  Обязательное поле |
| Дата получения звания | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Дата окончания звания | Дата | нет | Краткий формат даты  00.00.0000 | Необязательное поле |
| Код должности | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование должности | Строка | нет | нет | Длина поля не более 50 символов  Обязательное поле |
| Дата назначения на должность | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Дата увольнения | Дата | нет | Краткий формат даты  00.00.0000 | Необязательное поле |
| Номер караула | Целое число | нет | нет | Обязательное поле  Уникальное поле |
| Количество человек в карауле | Целое число | нет | нет | Обязательное поле |
| Код смены | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Дата и время начала смены | Строка | нет | 00.00.0000\ 99:00:00 | Обязательное поле |
| Дата и время конца смены | Строка | нет | 00.00.0000\ 99:00:00 | Обязательное поле |
| Номер караула на смене | Целое число | нет | нет | Обязательное поле |
| Количество человек на смене | Целое число | нет | нет | Обязательное поле |
| Код вызова | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Дата обращения заявителя | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Время обращения заявителя | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Адрес пожара | Строка | нет | нет | Обязательное поле  Длина поля не более 255 символов |
| Фамилия заявителя | Строка | нет | нет | Обязательное поле  Длина поля не более 50 символов |
| Имя заявителя | Строка | нет | нет | Обязательное поле  Длина поля не более 50 символов |
| Отчество заявителя | Строка | нет | нет | Необязательное поле  Длина поля не более 50 символов |
| Номер телефона заявителя | Строка | нет | +7 (000) 000-00-00 | Обязательное поле  Длина поля не более 11 символов |
| Код выезда | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Дата выезда | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Код вида объекта | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование объекта | Строка | нет | нет | Обязательное поле  Длина поля не более 255 символов |
| Время поступления сообщения | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Время прибытия на место пожара | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Время подачи первого ствола | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Время локализации | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Время ликвидации | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Время прибытия обратно в часть | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Время постановки в боевой расчет | Время | нет | Длинный формат времени  99:00:00;0;\_ | Обязательное поле |
| Дата окончания выезда | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Код причины | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование причины | Строка | нет | нет | Обязательное поле  Длина поля не более 255 символов |
| Код категории виновного лица | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование категории | Строка | нет | нет | Обязательное поле  Длина поля не более 255 символов |
| Количество погибших людей | Числовой | нет | нет | Обязательное поле |
| Количество погибших детей | Числовой | нет | нет | Обязательное поле |
| Количество погибших работников пожарной охраны | Числовой | нет | нет | Обязательное поле |
| Количество уничтоженных/поврежденных зданий/сооружений | Числовой | нет | нет | Обязательное поле |
| Количество уничтоженных жилых квартир | Числовой | нет | нет | Обязательное поле |
| Количество уничтоженной автотракторной и другой техники | Числовой | нет | нет | Обязательное поле |
| Спасено людей на пожаре | Числовой | нет | нет | Обязательное поле |
| Спасено материальных ценностей | Денежный | нет | нет | Обязательное поле  Значение в российских рублях |
| Номер контракта | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Дата заключения контракта | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Дата окончания контракта | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Номер допуска к пожарной машине | Строка | нет | нет | Длина поля не более 50 символов  Обязательное поле |
| Дата выдачи | Дата | нет | Краткий формат даты  00.00.0000 | Обязательное поле |
| Код пожарной части | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование пожарной части | Строка | нет | нет | Длина поля не более 255 символов  Обязательное поле |
| Адрес пожарной части | Строка | нет | нет | Длина поля не более 255 символов  Обязательное поле |
| Штатная численность | Целое число | нет | нет | Обязательное поле |
| Номер пожарной машины | Строка | нет | >L" "000" "LL | Длина поля не более 6 символов  Обязательное поле |
| Код вида пожарной машины | Целое число | нет | нет | Обязательное поле  Уникальное поле  Формируется автоматически (порядковый номер) |
| Наименование вида пожарной машины | Строка | нет | нет | Длина поля не более 255 символов  Обязательное поле |

При проектировании БД следует учесть следующие функциональные зависимости между атрибутами сущностей:

1. Код пола однозначно определяет наименование пола;
2. Код звания однозначно определяет наименование звания;
3. Код должности однозначно определяет наименование должности;
4. Код уровня образования однозначно определяет наименование уровня образования;
5. Код пожарной части однозначно определяет наименование пожарной части (в курсовой работе данный атрибут рассматривается как единое целое), начальника пожарной части, заместителя начальника пожарной части, адрес пожарной части (в курсовой работе данный атрибут рассматривается как единое целое), штатную численность, номер пожарной машины.
6. Номер пожарной машины однозначно определяет код вида пожарной машины
7. Номер контракта сотрудника однозначно определяет дату заключения, дату окончания;
8. Номер допуска к пожарной машине однозначно определяет сотрудника, дату выдачи;
9. Код вида объекта пожара однозначно определяет наименование вида объекта пожара;
10. Код причины пожара однозначно определяет наименование причины пожара;
11. Код категории виновного лица однозначно определяет наименование категории виновного лица;
12. Код вида пожарной машины однозначно определяет наименование вида пожарной машины;
13. Табельный номер сотрудника однозначно определяет фамилию сотрудника, имя сотрудника, отчество сотрудника, дату рождения, код пола сотрудника, контактный номер телефона сотрудника, адрес регистрации сотрудника, адрес фактического проживания сотрудника, номер караула, код пожарной части, серию паспорта, номер паспорта, номер контракта сотрудника;
14. Код дежурной смены однозначно определяет дату и время начала смены, дату и время окончания смены, номер караула на смене, количество человек на смене, код вызова;
15. Номер караула однозначно определяет количество человек в карауле;
16. Код вызова однозначно определяет адрес пожара, дату обращения заявителя, код вида объекта пожара, фамилию заявителя, имя заявителя, отчество заявителя, номер телефона заявителя, код выезда;
17. Код выезда однозначно определяет дату выезда, время поступления сообщения, время прибытия на место пожара, время подачи первого ствола, время локализации, время ликвидации, время прибытия обратно в часть, время постановки в боевой расчет, дату окончания выезда, количество погибших людей, количество погибших детей, количество погибших работников пожарной охраны, количество уничтоженных/поврежденных зданий сооружений, количество уничтоженных жилых квартир, количество уничтоженной автотракторной и другой техники, спасено людей на пожаре, спасено материальных ценностей;
18. Комбинация атрибутов Табельный номер сотрудника и Код должности однозначно определяет дату назначения на должность, дату увольнения с должности;
19. Комбинация атрибутов Табельный номер сотрудника и Код звания однозначно определяет дату получения звания и дату окончания звания;
20. Комбинация атрибутов Табельный номер сотрудника и Код уровня образования однозначно определяет серию документа, номер документа, наименование учебного заведения
21. Комбинация атрибутов Номер допуска к пожарной машине и Код вида пожарной машины однозначно определяет наименование видов пожарной машины, к которым допущен водитель

При установлении функциональных зависимостей и проектировании БД необходимо учесть следующее:

1. Караул связан с Сотрудником связью один-ко-многим (за одним караулом закреплено множество сотрудников, но один сотрудник закреплен только за одним караулом);
2. Сотрудник связан с Контрактом сотрудника связью один-ко-многим (у одного сотрудника в течение его службы может быть несколько контрактов);
3. Караул связан с Дежурной сменой связью один-ко-многим (один караул в течение месяца выходит на множество смен, но на одной смене только один конкретный караул);
4. Дежурная смена связана с Вызовом на пожар связью один-ко-многим (на одной дежурной смене происходит множество вызовов на пожар);
5. Вызов на пожар связан с Выездом на пожар связью один-ко-многим (на одном вызове на пожар может быть несколько выездов);
6. Вид объекта пожара связан с Вызовом на пожар связью один-ко-многим (один вид объекта может быть указан в разных заявках, поступивших в пожарную часть, но в одной заявке о вызове на пожар указывается один вид объекта пожара);
7. Причина пожара связана с Выездом на пожар связью один-ко-многим (один вид причины пожара может быть зафиксирована на разных выездах);
8. Виновное лицо связано с Выездом на пожар связью один-ко-многим (одна категория виновного лица может быть зафиксирована на разных выездах);
9. Сотрудник связан с Званием сотрудника связью многие-ко-многим (у одного сотрудника может быть несколько званий в процессе его службы, одно звание может быть у нескольких сотрудников);
10. Сотрудник связан с Должностью сотрудника связью многие-ко-многим (один сотрудник может занимать несколько должностей в процессе службы, одну должность может занимать несколько сотрудников);
11. Пол сотрудника связан с Сотрудником связью один-ко-многим (один пол может принадлежать разным сотрудникам, но у одного сотрудника только один пол);
12. Сотрудник связан с Образованием связью многие-ко-многим (у одного сотрудника может быть несколько образований, и одно образование может быть у нескольких сотрудников);
13. Сотрудник связан с Допуском к пожарной машине связью один-ко-многим (у одного сотрудника может быть несколько допусков к пожарной машине);
14. Пожарная часть связана с Сотрудником связью один-ко-многим (в одной пожарной части работает множество сотрудников);
15. Пожарная часть связана с Пожарной машиной связью один-ко-многим (в одной пожарной части находится множество машин, но машина закреплена только за одной пожарной частью);
16. Вид пожарной машины связан с Пожарной машиной связью один-ко-многим (один вид пожарной машины может относиться к нескольким машинам, но машине принадлежит только один вид);
17. Пожарная машина связана с Допуском к пожарной машине связью многие-ко-многим (на одну пожарную машину может быть выдано несколько допуском, и один допуск может быть выдан на несколько пожарных машин).

## 2.1 Проектирование базы данных

### 2.1.1 Приведение к 1 НФ

Универсальное отношение, включающее атрибуты:

* **код пожарной части;**
* наименование пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* адрес пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* штатная численность;
* номер пожарной машины;
* номер допуска к пожарной машине;
* дата выдачи допуска;
* код вида пожарной машины;
* наименование вида пожарной машины.
* **табельный номер;**
* фамилия сотрудника;
* имя сотрудника;
* отчество сотрудника;
* дата рождения;
* код пола сотрудника;
* наименование пола сотрудника;
* контактный номер телефона сотрудника;
* адрес регистрации сотрудника (в курсовой работе данный атрибут рассматривается как единое целое);
* адрес фактического проживания сотрудника (в курсовой работе данный атрибут рассматривается как единое целое);
* серия паспорта сотрудника;
* номер паспорта сотрудника;
* код уровня образования;
* наименование уровня образования сотрудника;
* код звания;
* наименование звания сотрудника;
* код должности;
* наименование должности сотрудника;
* дата назначения на должность;
* дата увольнения с должности;
* номер контракта сотрудника;
* дата заключения контракта;
* дата окончания контракта;
* **код дежурной смены;**
* дата и время начала смены;
* дата и время конца смены;
* номер караула на смене;
* количество человек на смене;
* номер караула;
* количество человек в карауле;
* **код вызова;**
* адрес пожара (в курсовой работе данный атрибут рассматривается как единое целое);
* дата обращения заявителя;
* время обращения заявителя;
* код вида объекта пожара;
* наименование объекта пожара;
* фамилия заявителя;
* имя заявителя;
* отчество заявителя;
* номер телефона заявителя;
* **код выезда;**
* дата выезда;
* время поступления сообщения;
* время прибытия на место пожара;
* время подачи первого ствола;
* время локализации;
* время ликвидации;
* время прибытия обратно в часть;
* время постановки в боевой расчет;
* дата окончания выезда;
* код причины;
* наименование причины;
* код категории виновного лица;
* наименование категории виновного лица;
* количество погибших людей;
* количество погибших детей;
* количество погибших работников пожарной охраны;
* количество уничтоженных/поврежденных зданий сооружений;
* количество уничтоженных жилых квартир;
* количество уничтоженной автотракторной и другой техники;
* спасено людей на пожаре;
* спасено материальных ценностей;

находится в первой нормальной форме (1НФ), и в соответствии с описанными выше

функциональными зависимостями первичный ключ состоит из атрибутов:

* Код пожарной части.
* Табельный номер сотрудника.
* Код дежурной смены.
* Код вызова
* Код выезда

Эта комбинация однозначно определяет значения всех остальных атрибутов и

является неизбыточной.

### 2.1.2 Приведение к 2 НФ

Атрибуты, перечисленные ниже, зависят только от Кода пожарной части:

* Наименование пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* Адрес пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* Штатная численность;
* Номер пожарной машины;
* Код вида пожарной машины;
* Наименование вида пожарной машины.

Получается, что зависимость от первичного ключа неполная, необходимо выполнить декомпозицию. После декомпозиции получаем таблицу:

**Пожарная часть:**

* **Код пожарной части**;
* Наименование пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* Адрес пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* Штатная численность;
* Номер пожарной машины;
* Код вида пожарной машины;
* Наименование вида пожарной машины.

Атрибуты, перечисленные ниже, зависят только от Табельного номера сотрудника:

* Фамилия сотрудника;
* Имя сотрудника;
* Отчество сотрудника;
* Дата рождения;
* Код пола сотрудника;
* Наименование пола сотрудника;
* Контактный номер телефона сотрудника;
* Адрес регистрации сотрудника (в курсовой работе данный атрибут рассматривается как единое целое);
* Адрес фактического проживания сотрудника (в курсовой работе данный атрибут рассматривается как единое целое);
* Серия паспорта сотрудника;
* Номер паспорта сотрудника;
* Код уровня образования;
* Наименование уровня образования сотрудника;
* Код звания;
* Наименование звания сотрудника;
* Код должности;
* Наименование должности сотрудника;
* Дата назначения на должность;
* Дата увольнения с должности;
* Номер контракта сотрудника;
* Дата заключения контракта;
* Дата окончания контракта;
* номер караула;
* количество человек в карауле;
* Номер допуска к пожарной машине;
* Дата выдачи допуска.

Получается, что зависимость от первичного ключа неполная, необходимо выполнить декомпозицию. После декомпозиции получаем таблицу:

**Сотрудник:**

* **Табельный номер;**
* Фамилия сотрудника;
* Имя сотрудника;
* Отчество сотрудника;
* Дата рождения;
* Код пола сотрудника;
* Наименование пола сотрудника;
* Контактный номер телефона сотрудника;
* Адрес регистрации сотрудника (в курсовой работе данный атрибут рассматривается как единое целое);
* Адрес фактического проживания сотрудника (в курсовой работе данный атрибут рассматривается как единое целое);
* Код пожарной части (внешний ключ, обеспечивающий связь отношений с Пожарной частью);
* Серия паспорта сотрудника;
* Номер паспорта сотрудника;
* Код уровня образования;
* Наименование уровня образования сотрудника;
* Код звания;
* Наименование звания сотрудника;
* Код должности;
* Наименование должности сотрудника;
* Дата назначения на должность;
* Дата увольнения с должности;
* Номер контракта сотрудника;
* Дата заключения контракта;
* Дата окончания контракта;
* Номер караула;
* Количество человек в карауле;
* Номер допуска к пожарной машине;
* Дата выдачи допуска.

Атрибуты, перечисленные ниже, зависят только от Кода дежурной смены:

* Дата и время начала смены;
* Дата и время конца смены;
* Номер караула на смене;
* Количество человек на смене;

Получается, что зависимость от первичного ключа неполная, необходимо выполнить декомпозицию. После декомпозиции получаем таблицу:

**Дежурная смена:**

* **Код дежурной смены**
* Дата и время начала смены;
* Дата и время конца смены;
* Номер караула на смене;
* Количество человек на смене;

Атрибуты, перечисленные ниже, зависят только от Кода вызова:

* Адрес пожара (в курсовой работе данный атрибут рассматривается как единое целое);
* Дата обращения заявителя;
* Время обращения заявителя;
* Код вида объекта пожара;
* Наименование объекта пожара;
* Фамилия заявителя;
* Имя заявителя;
* Отчество заявителя;
* Номер телефона заявителя;

Получается, что зависимость от первичного ключа неполная, необходимо выполнить декомпозицию. После декомпозиции получаем таблицу:

**Вызов на пожар:**

* **Код вызова**
* Адрес пожара
* Дата обращения заявителя
* Время обращения заявителя
* Код вида объекта пожара
* Наименование объекта пожара
* Фамилия заявителя;
* Имя заявителя;
* Отчество заявителя;
* Номер телефона заявителя;
* Код смены; (внешний ключ, обеспечивающий связь отношений с Дежурная смена).

Атрибуты, перечисленные ниже, зависят только от Кода выезда:

* Дата выезда;
* Время поступления сообщения;
* Время прибытия на место пожара;
* Время подачи первого ствола;
* Время локализации;
* Время ликвидации;
* Время прибытия обратно в часть;
* Время постановки в боевой расчет;
* Дата окончания выезда;
* Код причины;
* Наименование причины;
* Код категории виновного лица;
* Наименование категории виновного лица;
* Количество погибших людей;
* Количество погибших детей;
* Количество погибших работников пожарной охраны;
* Количество уничтоженных/поврежденных зданий сооружений;
* Количество уничтоженных жилых квартир;
* Количество уничтоженной автотракторной и другой техники;
* Спасено людей на пожаре;
* Спасено материальных ценностей.

Получается, что зависимость от первичного ключа неполная, необходимо выполнить декомпозицию. После декомпозиции получаем таблицу:

**Выезд на пожар:**

* **Код выезда;**
* Код вызова (внешний ключ, обеспечивающий связь отношений с Вызов на пожар)
* Дата выезда;
* Время поступления сообщения;
* Время прибытия на место пожара;
* Время подачи первого ствола;
* Время локализации;
* Время ликвидации;
* Время прибытия обратно в часть;
* Время постановки в боевой расчет;
* Дата окончания выезда;
* Код причины;
* Код категории виновного лица;
* Количество погибших людей;
* Количество погибших детей;
* Количество погибших работников пожарной охраны;
* Количество уничтоженных/поврежденных зданий сооружений;
* Количество уничтоженных жилых квартир;
* Количество уничтоженной автотракторной и другой техники;
* Спасено людей на пожаре;
* Спасено материальных ценностей.

### 2.1.3 Приведение к 3 НФ

В таблице "Сотрудник" имелись зависимости между неключевыми атрибутами. Их следствие – транзитивные зависимости атрибутов (Наименование пола, Наименование уровня образования, Наименование звания, Наименование должности, Дата заключения контракта, Дата окончания контракта, Количество человек в карауле, Дата выдачи допуска) от первичного ключа. Для устранения – декомпозиция. Получаем следующие таблицы:

**Пол сотрудника**

* **Код пола**
* Наименование пола

**Образование:**

* **Код\_уровня\_образования**
* Наименование уровня образования

**Образование сотрудника** *(Данная таблица фиксирует связь «М:М» между сущностями «Образование» и «Сотрудник»)****:***

* **Табельный номер**
* **Код\_уровня\_образования**
* Серия документа
* Номер документа
* Наименование учебного заведения

**Должность:**

* **Код должности**
* Наименование должности

**Должность сотрудника** *(Данная таблица фиксирует связь «М:М» между сущностями «Должность» и «Сотрудник»)****:***

* **Табельный номер**
* **Код должности**
* Дата назначения на должность
* Дата увольнения

**Звание:**

* **Код звания**
* Наименование звания

**Звание сотрудника** *(Данная таблица фиксирует связь «М:М» между сущностями «Звание» и «Сотрудник»)****:***

* **Табельный номер**
* **Код звания**
* Дата получения звания
* Дата окончания звания

**Контракт сотрудника:**

* **Номер контракта**
* Дата заключения контракта
* Дата окончания контракта
* Табельный номер (внешний ключ, обеспечивающий связь отношений с Сотрудник)

**Караул:**

* **Номер караула**
* Количество человек в карауле

**Допуск к пожарной машине:**

* **Номер допуска к пожарной машине;**
* Табельный номер (внешний ключ, обеспечивающий связь отношений с Сотрудник)
* Дата выдачи.

***Итоговый вид таблицы "Сотрудник":***

* **Табельный номер**
* Фамилия
* Имя
* Отчество
* Дата рождения
* Код пола (внешний ключ, обеспечивающий связь отношений с Пол сотрудника)
* Контактный номер телефона
* Адрес регистрации
* Адрес фактического проживания
* Караул (внешний ключ, обеспечивающий связь отношений с Караулом)
* Код пожарной части (внешний ключ, обеспечивающий связь отношений с Пожарной частью)
* Серия паспорта
* Номер паспорта

В таблице "Вызов на пожар" имелись зависимости между не ключевыми атрибутами. Их следствие – транзитивные зависимости атрибутов (Наименование объекта) от первичного ключа. Для устранения – декомпозиция:

**Вид объекта пожара:**

* **Код вида объекта**
* Наименование объекта

***Итоговый вид таблицы "Вызов на пожар":***

* **Код вызова**
* Адрес пожара
* Дата обращения заявителя;
* Время обращения заявителя;
* Код вида объекта; (внешний ключ, обеспечивающий связь отношений с Видом объекта пожара)
* Фамилия заявителя;
* Имя заявителя;
* Отчество заявителя;
* Номер телефона заявителя.
* Код смены (внешний ключ, обеспечивающий связь отношений с Дежурная смена)

В таблице "Выезд на пожар" имелись зависимости между не ключевыми атрибутами. Их следствие – транзитивные зависимости атрибутов (Наименование причины, Наименование категории) от первичного ключа. Для устранения – декомпозиция:

**Причина пожара:**

* **Код причины**
* Наименование причины

**Виновное лицо:**

* **Код категории виновного лица**
* Наименование категории

***Итоговый вид таблицы "Выезд на пожар":***

* **Код выезда;**
* Код вызова (внешний ключ, обеспечивающий связь отношений с Вызов на пожар)
* Дата выезда;
* Время поступления сообщения;
* Время прибытия на место пожара;
* Время подачи первого ствола;
* Время локализации;
* Время ликвидации;
* Время прибытия обратно в часть;
* Время постановки в боевой расчет;
* Дата окончания выезда;
* Код причины; (внешний ключ, обеспечивающий связь отношений с Причина пожара)
* Код категории виновного лица; (внешний ключ, обеспечивающий связь отношений с Виновное лицо)
* Количество погибших людей;
* Количество погибших детей;
* Количество погибших работников пожарной охраны;
* Количество уничтоженных/поврежденных зданий/сооружений;
* Количество уничтоженных жилых квартир;
* Количество уничтоженной автотракторной и другой техники;
* Спасено людей на пожаре;
* Спасено материальных ценностей.

В таблице "Пожарная часть" имелись зависимости между не ключевыми атрибутами. Их следствие – транзитивные зависимости атрибутов (Код вида пожарной машины) от первичного ключа. Для устранения – декомпозиция:

**Пожарная машина:**

* **Номер пожарной машины;**
* Код вида пожарной машины;
* Наименование вида пожарной машины;
* Код пожарной части (внешний ключ, обеспечивающий связь отношений с Пожарная часть).

***Итоговый вид таблицы "Пожарная часть":***

* **Код пожарной части**;
* Наименование пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* Адрес пожарной части (в курсовой работе данный атрибут рассматривается как единое целое);
* Штатная численность.

В таблице "Пожарная машина" имелись зависимости между не ключевыми атрибутами. Их следствие – транзитивные зависимости атрибутов (Наименование вида пожарной машины) от первичного ключа. Для устранения – декомпозиция:

**Вид пожарной машины:**

* **Код вида пожарной машины**
* Наименование вида пожарной машины

***Итоговый вид таблицы "Пожарная машина":***

* **Номер пожарной машины;**
* Код пожарной части (внешний ключ, обеспечивающий связь отношений с Пожарная часть)
* Код вида пожарной машины (внешний ключ, обеспечивающий связь отношений с Вид пожарной машины)

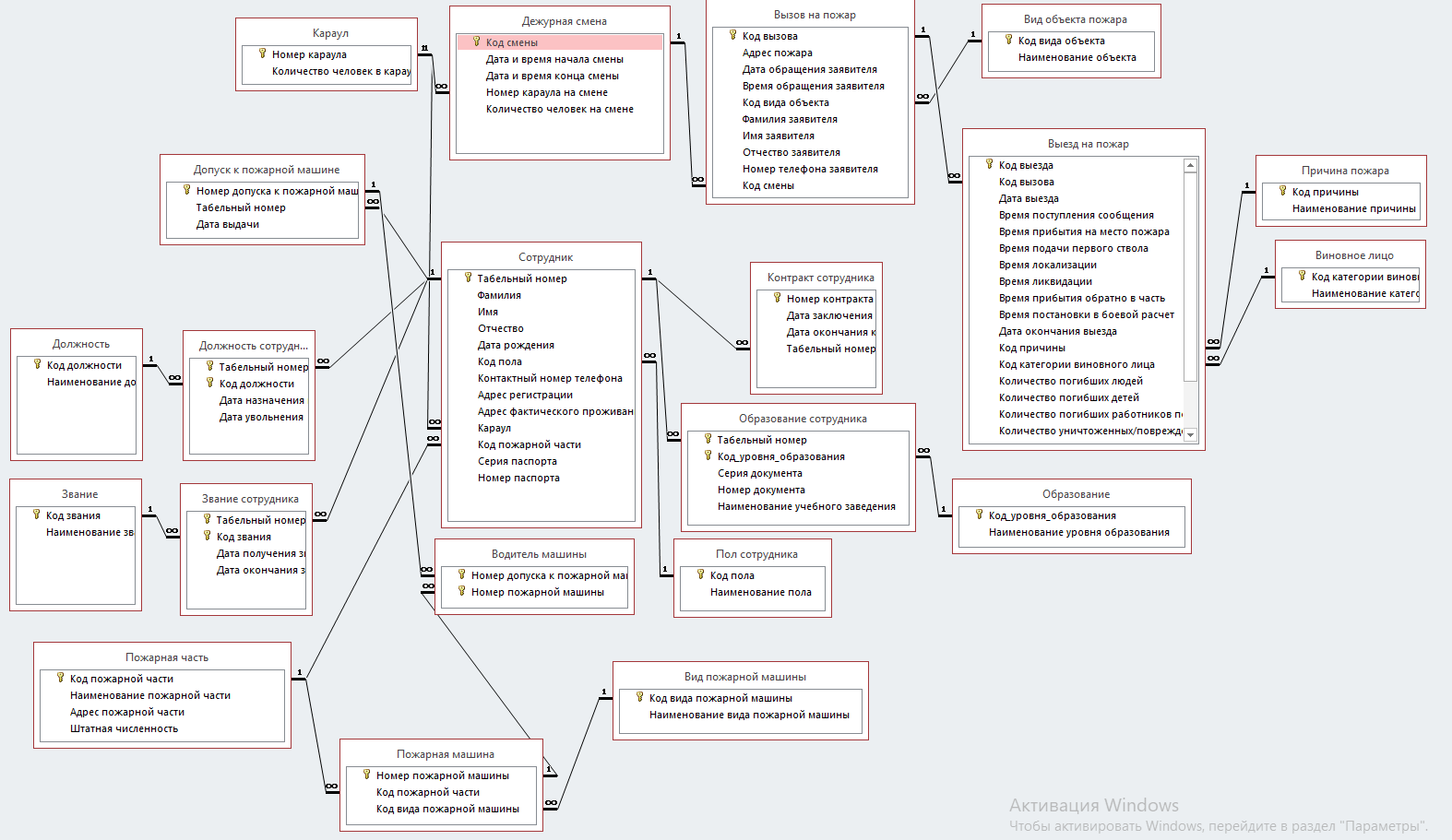
**Водитель машины** *(Таблица Водитель машины фиксирует связь «М:М» между сущностями «Пожарная машина» и «Допуск к пожарной машине»)****:***

* **Номер допуска к пожарной машине;**
* **Номер пожарной машины.**

***Итоговый вид таблицы "Дежурная смена":***

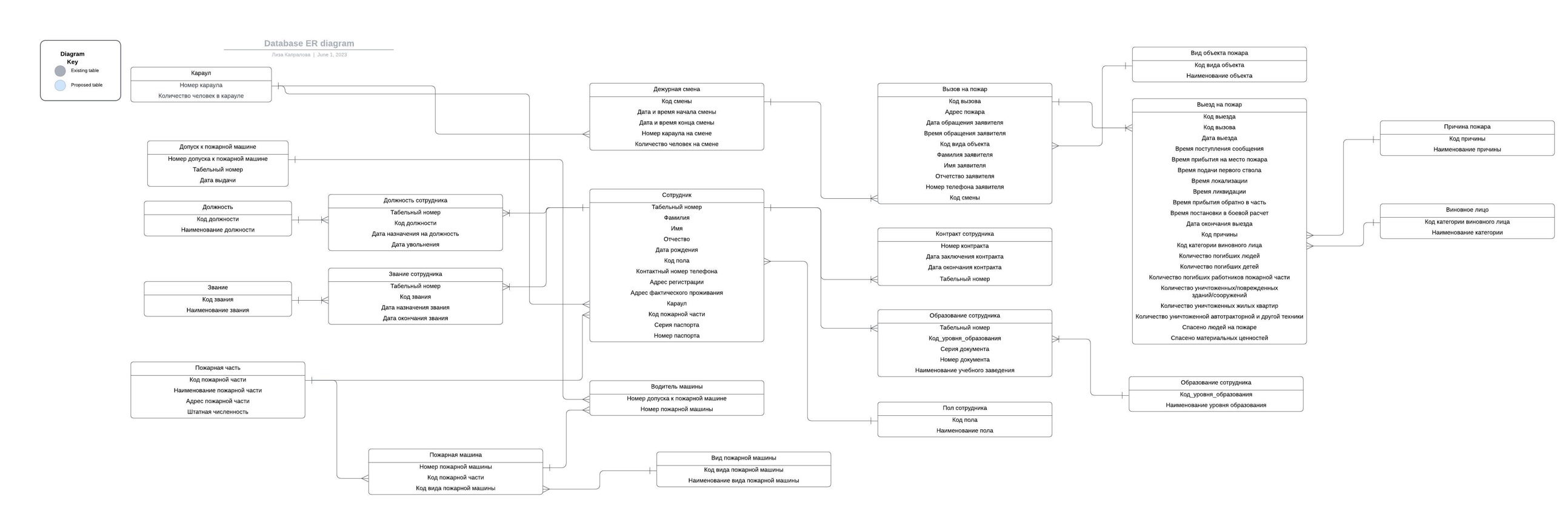
* **Код дежурной смены**
* Дата и время начала смены;
* Дата и время конца смены;
* Номер караула на смене;
* Количество человек на смене.

После завершения выполнения нормализации получаем логическую модель базы данных (см.рис.10).



***Рисунок 10 - Схема БД***

Кроме того, была построена концептуальная модель базы данных, а именно ER - диаграмма в нотации Мартина (см.рис.11).



***Рисунок 11 - Концептуальная модель БД***

# 3. Проектирование приложения

## 3.1 Диаграммы последовательностей

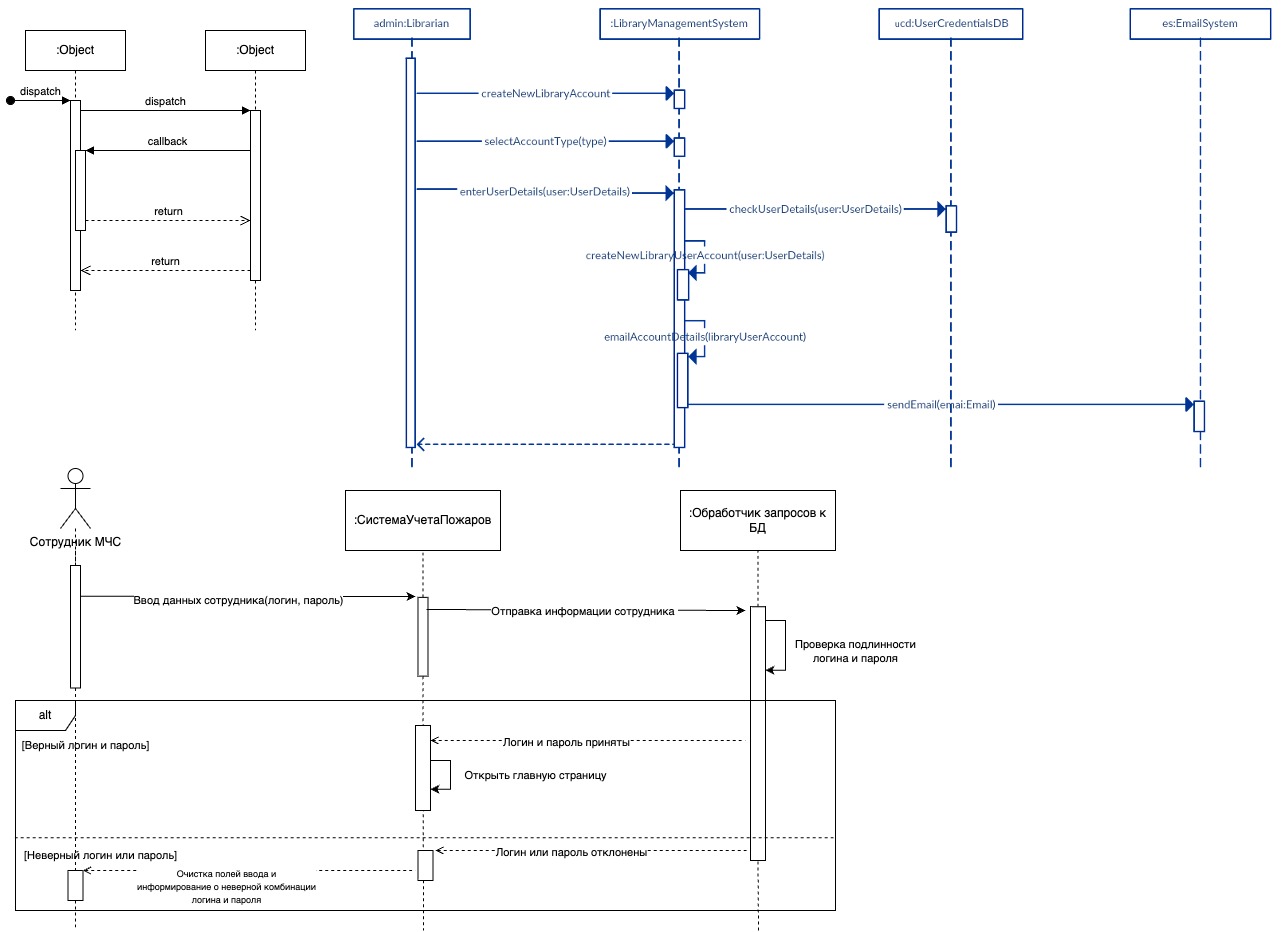
На этапе проектирования приложения была построена модель поведения системы, выполненная в нотации диаграмм последовательностей.

Для построения диаграмм последовательности были следующие прецеденты:

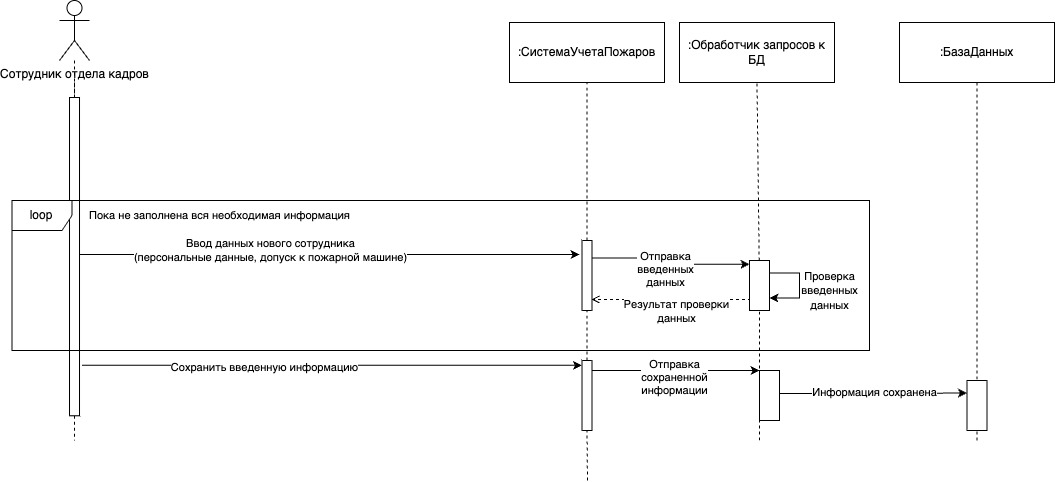
1. Вход в профиль.
2. Внести информации о пожаре.
3. Внести информацию о выезде.
4. Внести информацию о дежурной смене.
5. Внести информацию о сотруднике.
6. Сформировать донесение о пожаре.
7. Внести информацию о пожарной части.
8. Внести информацию о пожарной машине.

Для каждого из представленных выше прецедентов была разработана своя собственная диаграмма последовательности, позволяющая описать как различные части системы взаимодействуют друг с другом для выполнения функции, а также порядок, в котором происходит взаимодействие при выполнении конкретного случая использования.

Рассмотрим диаграмму, построенную для прецедента "Вход в профиль" (см.рис.12). Первым действием является то, что сотрудник МЧС вводит свои данные (логин и пароль) в системе учета пожаров. Затем эта информация обрабатывается обработчиком запросов к БД, происходит проверка подлинности введенных логина и пароля. Далее происходят альтернативные действия: первое из них - когда логин и пароль приняты, а второе - когда логин и пароль не приняты. В первом случае открывается главная страница приложения, а во втором случае система информирует сотрудника МЧС о неверной комбинации логина и пароля и производит очистку полей ввода. Сотрудник МЧС в таком случае должен ввести логин и пароль по - новой.

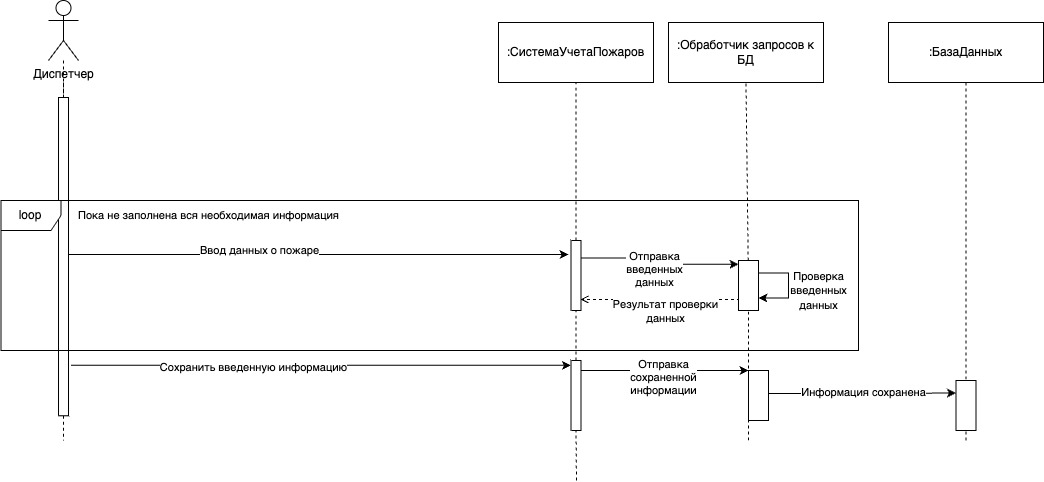
***Рисунок 12 - Диаграмма последовательности для прецедента "Вход в профиль"***

Перейдем к рассмотрению диаграммы последовательности для прецедента "Внести информацию о пожаре" (см.рис.13). Начинается все с цикла, который не прекратится до тех пор, пока не будет заполнена вся информация о сотруднике. В этом цикле происходит ввод данных о сотруднике в системе учета пожаров, отправка данных, их обработка. По завершении цикла диспетчер сохраняет введенную информацию в системе. Затем эта введенная информация отправляется в базу данных и хранится в ней.



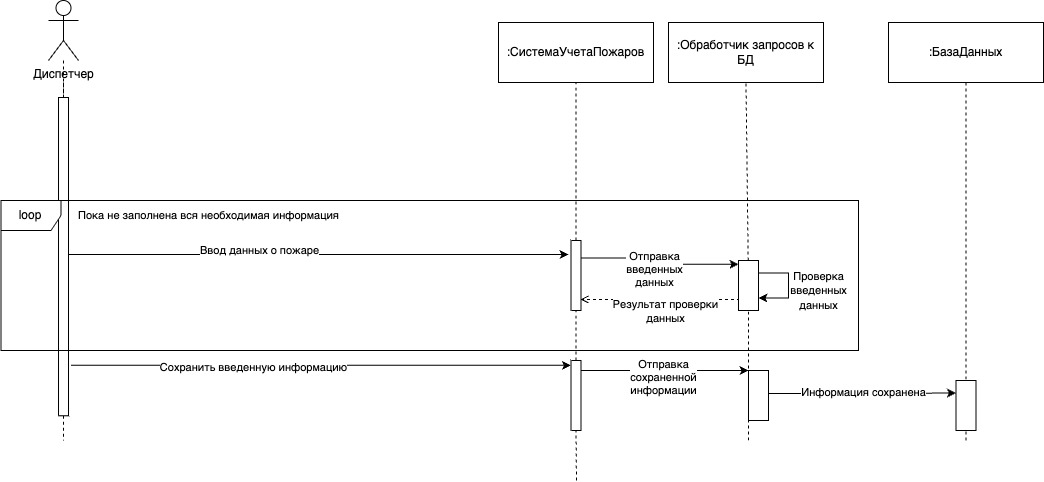
***Рисунок 13 - Диаграмма последовательности для прецедента "Внести информацию о сотруднике"***

Перейдем к рассмотрению диаграммы последовательности для прецедента "Внести информацию о пожаре" (см.рис.14). Начинается все с цикла, который не прекратится до тех пор, пока не будет заполнена вся информация о пожаре. В этом цикле происходит ввод данных о пожаре в системе учета пожаров, отправка данных, их обработка. По завершении цикла диспетчер сохраняет введенную информацию в системе. Затем эта введенная информация отправляется в базу данных и хранится в ней.



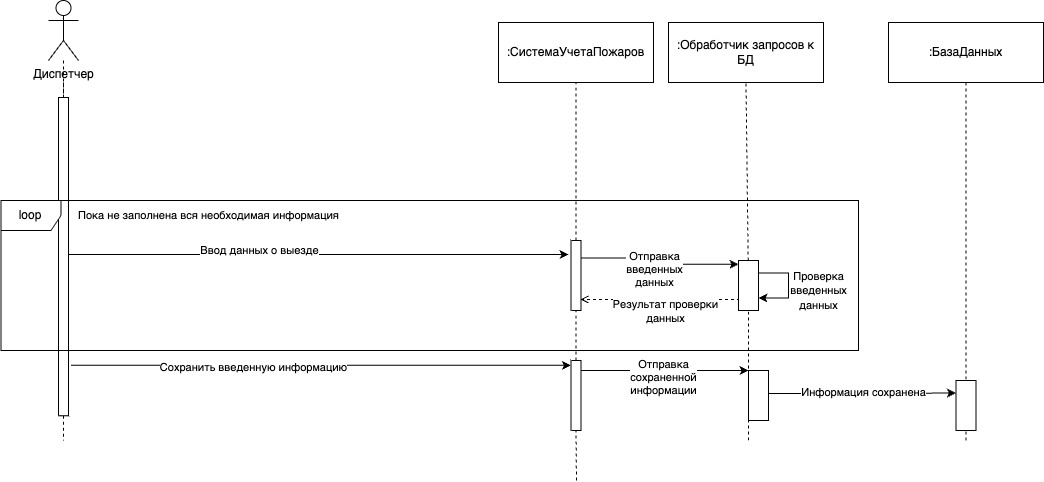
***Рисунок 14 - Диаграмма последовательности для прецедента "Внести информацию о пожаре"***

Перейдем к рассмотрению диаграммы последовательности для прецедента "Внести информацию о выезде" (см.рис.15). Начинается все с цикла, который не прекратится до тех пор, пока не будет заполнена вся информация о выезде. В этом цикле происходит ввод данных о выезде в системе учета пожаров, отправка данных, их обработка. По завершении цикла диспетчер сохраняет введенную информацию в системе. Затем эта введенная информация отправляется в базу данных и хранится в ней.

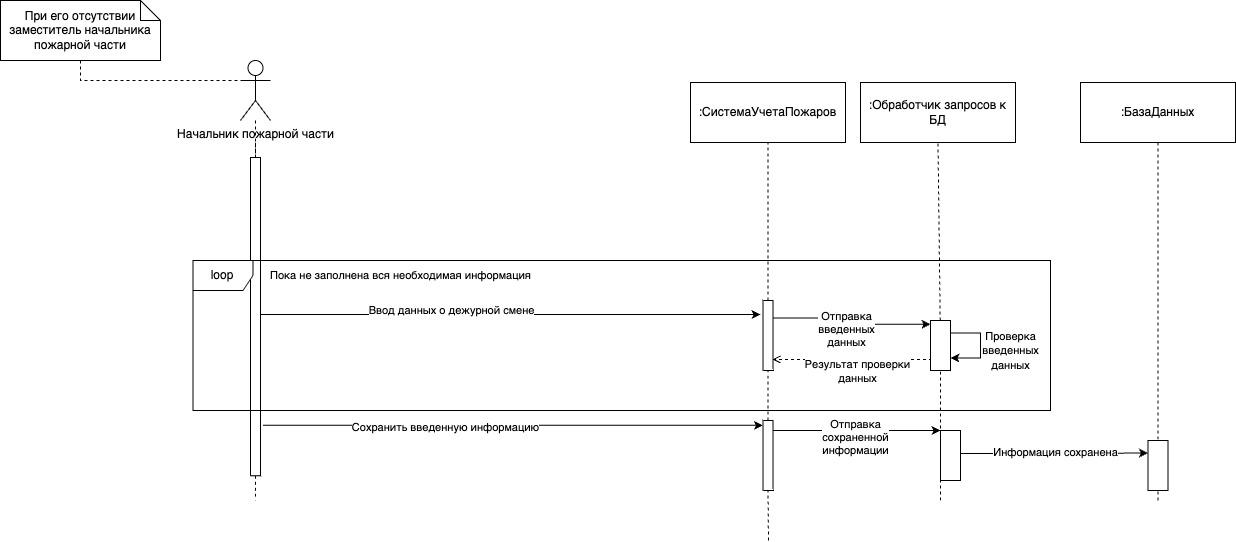


***Рисунок 15 - Диаграмма последовательности для прецедента "Внести информацию о выезде"***

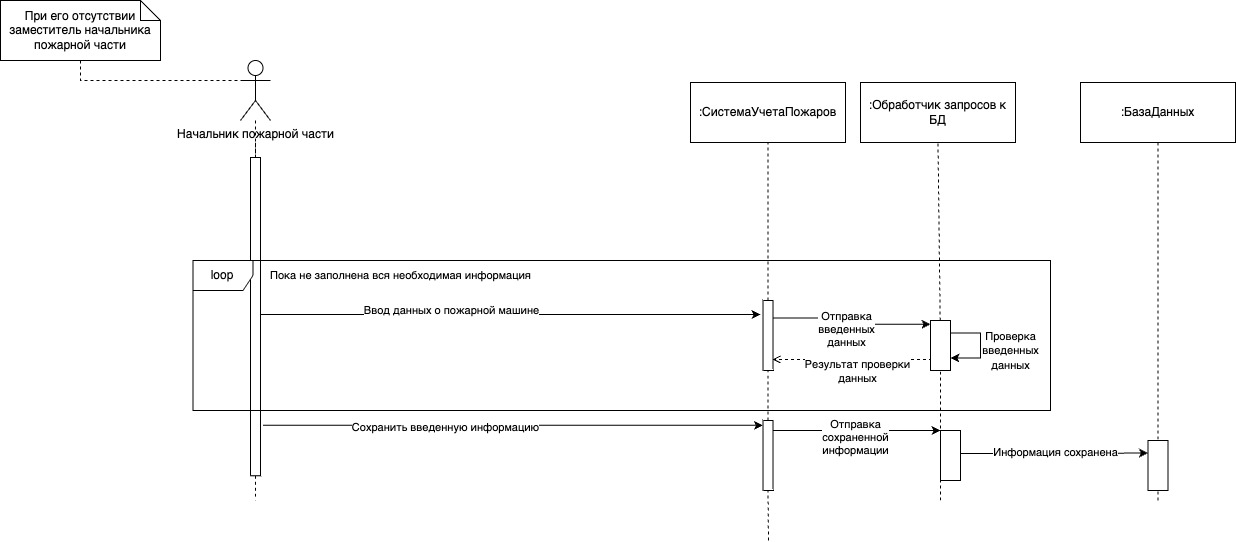
Перейдем к рассмотрению диаграммы последовательности для прецедента "Внести информацию о дежурной смене" (см.рис.16). Начинается все с цикла, который не прекратится до тех пор, пока не будет заполнена вся информация о дежурной смене. В этом цикле происходит ввод данных о смене в системе учета пожаров, отправка данных, их обработка. По завершении цикла диспетчер сохраняет введенную информацию в системе. Затем эта введенная информация отправляется в базу данных и хранится в ней.



***Рисунок 16 - Диаграмма последовательности для прецедента "Внести информацию о дежурной смене"***

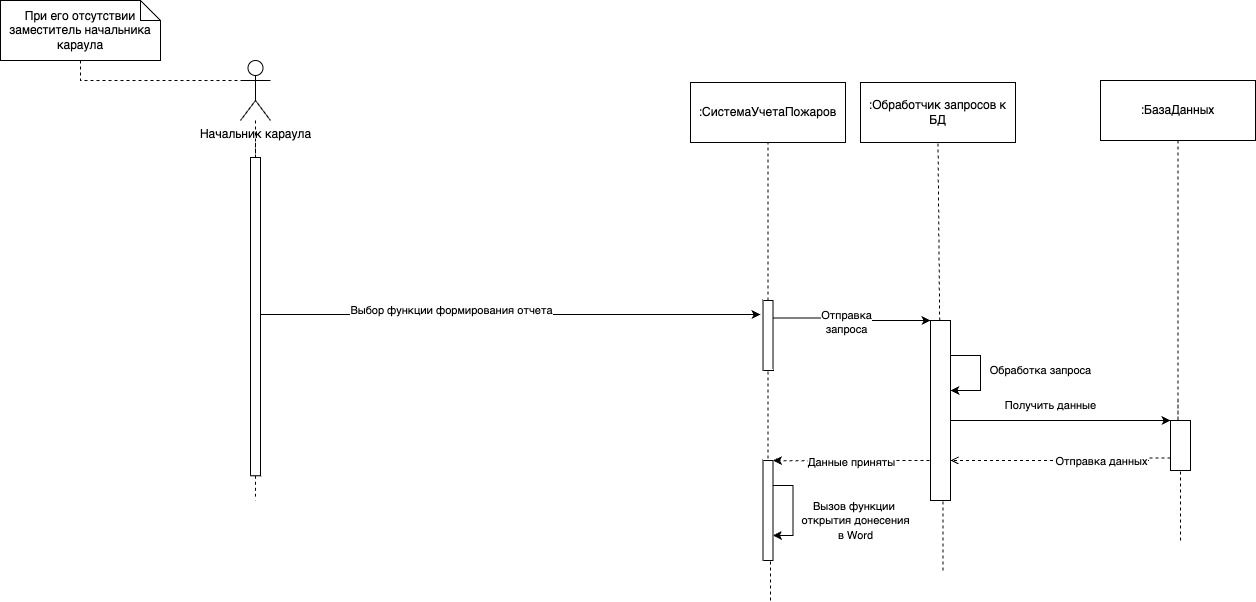
Перейдем к рассмотрению диаграммы последовательности для прецедента "Внести информацию о пожарной части" (см.рис.17). Начинается все с цикла, который не прекратится до тех пор, пока не будет заполнена вся информация о пожарной части. В этом цикле происходит ввод данных о пожарной части в системе учета пожаров, отправка данных, их обработка. По завершении цикла начальник пожарной части (при его отсутствии заместитель начальника пожарной части) сохраняет введенную информацию в системе. Затем эта введенная информация отправляется в базу данных и хранится в ней.

***Рисунок 17 - Диаграмма последовательности для прецедента "Внести информацию о пожарной части"***

Перейдем к рассмотрению диаграммы последовательности для прецедента "Внести информацию о пожарной машине" (см.рис.18). Начинается все с цикла, который не прекратится до тех пор, пока не будет заполнена вся информация о пожарной машине. В этом цикле происходит ввод данных о пожарной машине в системе учета пожаров, отправка данных, их обработка. По завершении цикла начальник пожарной части (при его отсутствии заместитель начальника пожарной части) сохраняет введенную информацию в системе. Затем эта введенная информация отправляется в базу данных и хранится в ней.

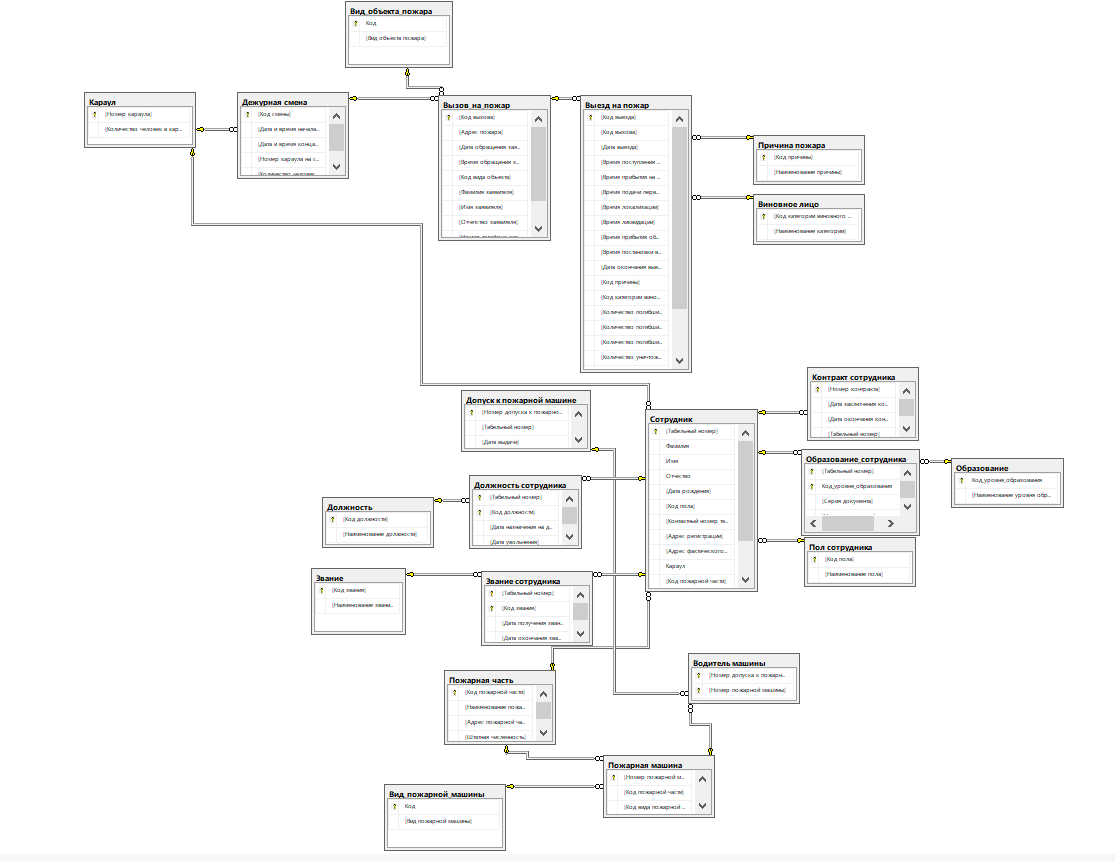
***Рисунок 18 - Диаграмма последовательности для прецедента "Внести информацию о пожарной машине"***

Перейдем к рассмотрению диаграммы последовательности для прецедента "Сформировать донесение о пожаре" (см.рис.19). Начинается все с выбора начальником караула (в его отсутствие заместителем начальника караула) функции формирования отчета. Затем происходит отправка запроса и его обработка. После этого из базы данных происходит получение нужных данных и в системе вызывается функция открытия донесения в Word.



***Рисунок 19 - Диаграмма последовательности для прецедента "Сформировать донесение о пожаре"***

На этапе проектирования приложения была построена физическая модель базы данных с учетом использования СУБД Microsoft SQL Server (см.рис.20).



***Рисунок 20 - Физическая модель БД***

## 3.2 Проектирование пользовательского интерфейса

При проектировании интерфейса приложения были предусмотрены средства ввода/редактирования/удаления данных, обеспечивающие минимизацию трудоемкости выполнения операций.

Для работы с данными были использованы стандартные элементы Windows Forms, например Button (кнопки добавления/удаления/редактирования/сохранения информации), DataGridView (отображение данных, полученных из БД, в виде таблицы), различные TextBox (текстовые поля) и Label (отображение текста, доступного только для чтения).

В некоторых таблицах с данными ввод данных в поле был организован при помощи выпадающего списка. Данные в одни выпадающие списки были загружены в базу данных из файла типа ".xlsx", а в другие выпадающие списки данные были взяты из связанных таблиц БД.

Для контроля действий пользователя при вводе информации, на некоторые поля была установлена маска ввода (маска ввода для даты рождения, номера телефона), была использована конструкция "try...catch".

Для полей типа TextBox была установлена валидация данных при помощи события Validating (если значение TextBox пустое, то появляется окно с предупреждение о том, что строка пустая).

Для столбцов DataGridView, содержащих информацию о дате или времени, был установлен соответствующий формат и нулевое (дефолтное) значение столбца (при помощи свойства DefaultCellStyle).

Для авторизации пользователя в системе была создана форма, содержащая поля для логина и пароля с проверкой правильности их ввода. Если данные введены неверно, то на экране появляется сообщение о некорректном вводе, и поля автоматически очищаются. Если данные были введены верно, то на экране появляется сообщение об успешной авторизации и после чего выполняется вход в систему. Дополнительно для пароля была реализована возможность скрыть его (на форму был добавлен элемент CheckBox с названием "Показать": при установке на него галочки пароль не скрывается; при отсутствии галочки пароль скрывается символом " • ").

Кроме того, в системе была создана дополнительная таблица исключительно для просмотра файлов типа ".xlsx". В этой форме пользователь может выбрать любой файл со своего компьютера типа ".xlsx" и просмотреть содержимое любого его листа. Для реализации такой возможности в системе, в проекте были установлены два пакета NuGet: первый - ExcelDataReader, второй - ExcelDataReader.DataSet.

Для добавления записей в некоторые таблицы был использован метод Fill. Для обновления записей в некоторых таблицах был использован метод Update. Для удаления записей в некоторых таблицах был использован метод RemoveCurrent.

# 4. Разработка запросов к данным

При разработке приложения были составлены следующие запросы.

* Запрос на удаление данных:

int kod = Convert.ToInt32(textBox1.Text);

SqlConnection connection = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=MchsProekt;Integrated Security=True"); connection.Open();

SqlCommand cmd = new SqlCommand(@"DELETE FROM [Пожарная часть] WHERE [Код пожарной части] = '" + kod + "'", connection);

cmd.ExecuteNonQuery();

connection.Close();

MessageBox.Show("Удаление выполнено успешно"); this.пожарная\_частьTableAdapter.Fill(this.mchsProektDataSet.Пожарная\_часть);

* Запросы с параметром (данные выводятся в отдельную таблицу):

1) int kod2 = Convert.ToInt32(textBox2.Text);

SqlConnection connection = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=MchsProekt;Integrated Security=True"); connection.Open();

SqlCommand cmd = new SqlCommand(@"SELECT \* FROM [Пожарная часть] WHERE [Код пожарной части] ='" + kod2 + "'", connection);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView2.DataSource = dt;

2) int kod2 = Convert.ToInt32(textBox2.Text);

SqlConnection connection = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=MchsProekt;Integrated Security=True");

connection.Open();

SqlCommand cmd = new SqlCommand(@"SELECT \* FROM Сотрудник WHERE [Табельный номер] ='" + kod2 + "'", connection);

SqlDataAdapter da = new SqlDataAdapter(cmd);

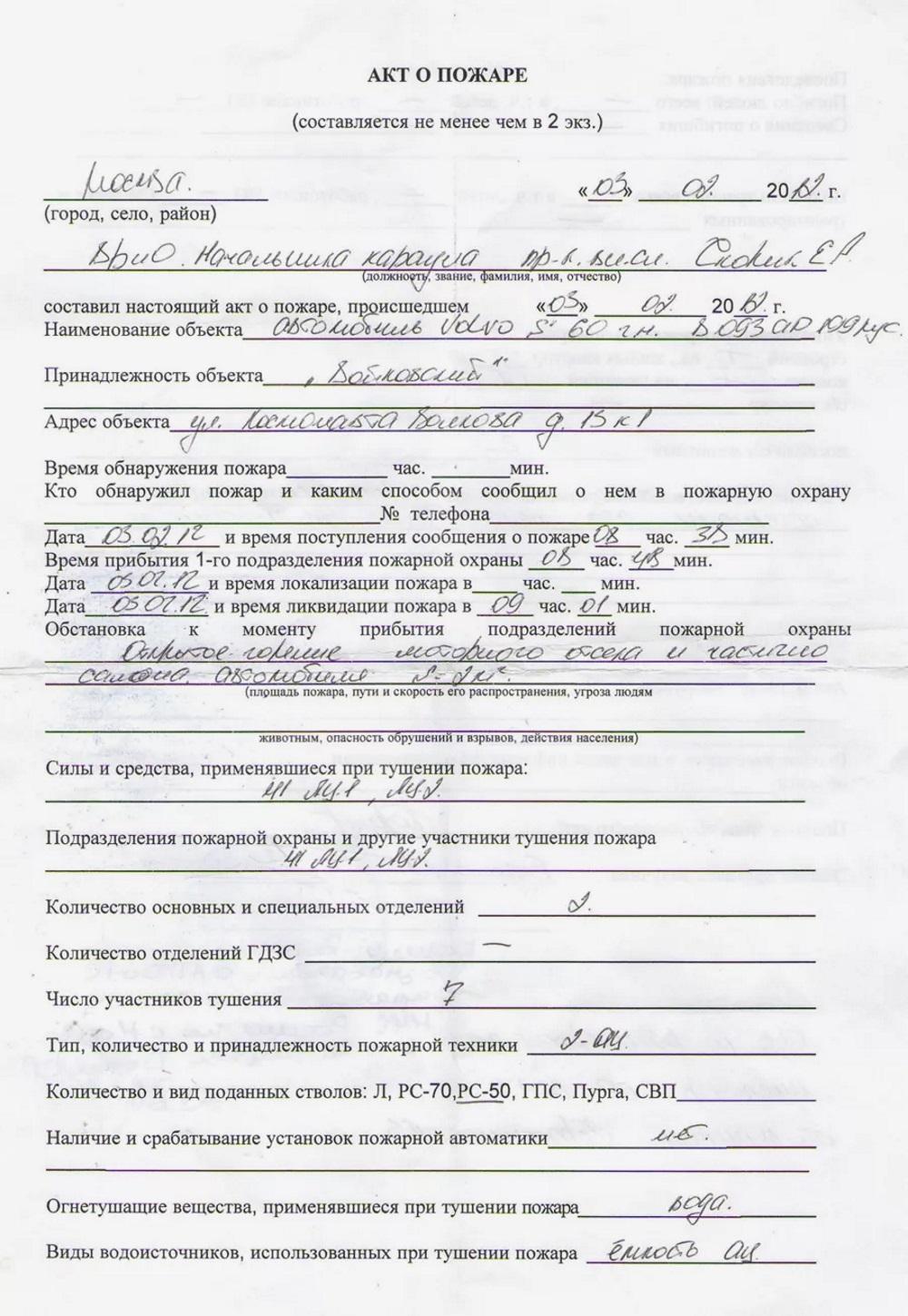
DataTable dt = new DataTable();

da.Fill(dt);

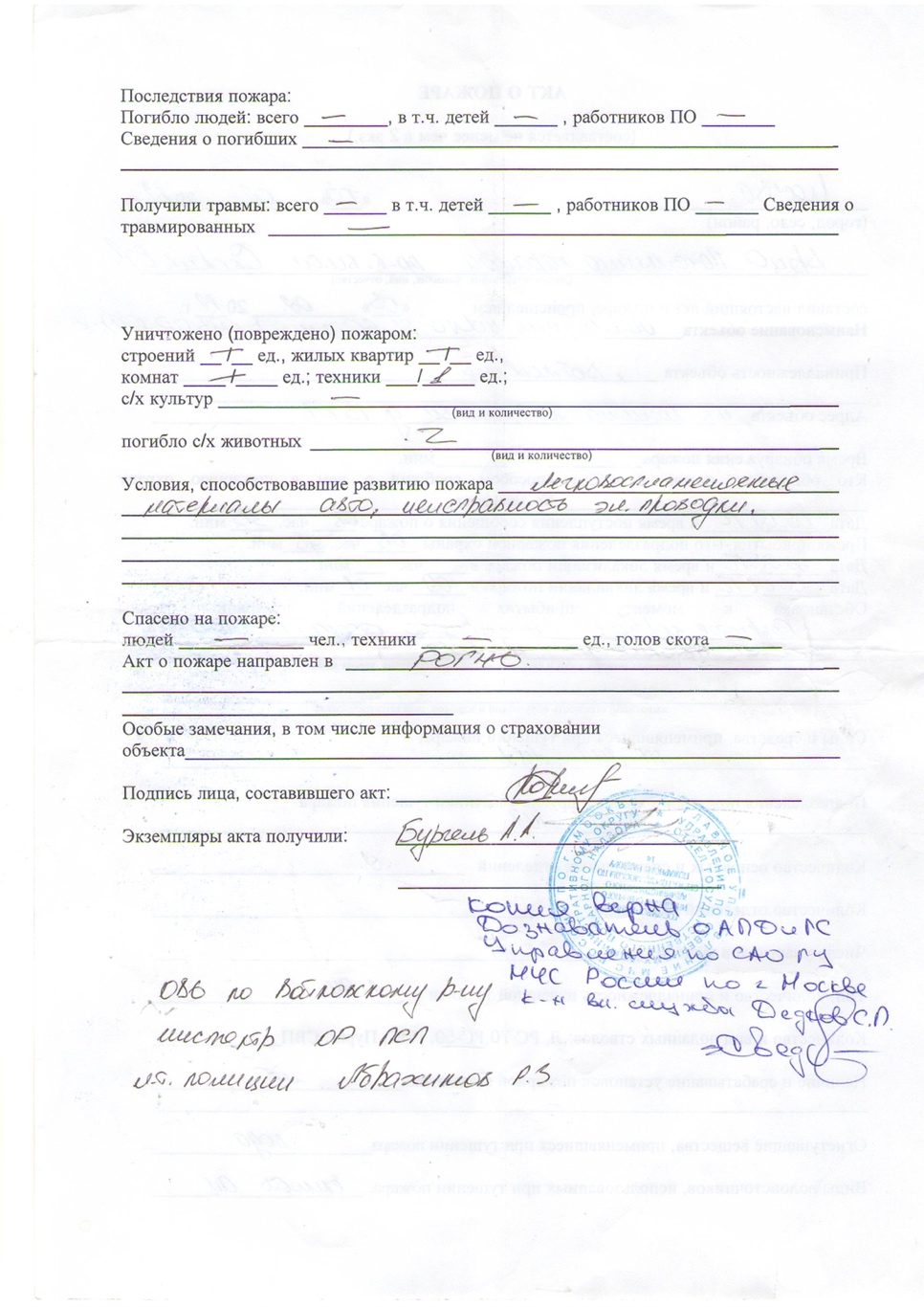
dataGridView2.DataSource = dt;

# 5. Проектирование отчетов

Основой для генерации донесения о пожаре послужил образец донесения о пожаре, установленный МЧС России (см.рис.21 и рис.22). Образец был упрощен и для формирования донесения в разрабатываемой системе были выбраны самые основные и важные поля.



***Рисунок 21 - Образец донесения***



***Рисунок 22 - Образец донесения***

Для выгрузки донесения в Word была подключения библиотека Microsoft Word Object Library. Для вывода и стилизации строк с информацией в донесении были использованы методы класса Selection (TypeText, Font, TypeParagraph и т.д.).

Сам запрос для выгрузки донесения представлял собой запрос на выборку данных. Код, обеспечивающий выгрузку данных в формате донесения в Word, продемонстрирован в приложениях (ПРИЛОЖЕНИЕ В).

Донесение, полученное из разработанного приложения, продемонстрировано ниже (см.рис.23).



***Рисунок 23 - Донесение, полученное из приложения***

# 6. Тестирование приложения

После разработки приложения было проведено тестирование по черному ящику.

В Таблице 5.1 «Результаты тестирования приложения по черному ящику» представлены результаты тестирования для всех основных функций приложения.

***Таблица 5.1 – Результаты тестирования приложения по черному ящику***

| **Функция** | **Входные данные** | **Ожидаемый результат** | **Полученный результат** |
| --- | --- | --- | --- |
| **Данные из таблиц** | | | |
| Добавление | Табельный номер – 2, Фамилия - Петров,  Имя - Петр,  Отчество - Петрович,  Дата рождения - 12.12.1992  Код пола - 01 (Выбирается из списка)  Контактный номер телефона - +7(231)232-13-12  Адрес регистрации - г.Москва, ул.Ямская, д.2, кв.56  Адрес фактического проживания - г.Москва, ул.Ямская, д.2, кв.56  Караул - 2  Код пожарной части - 4  Серия паспорта - 1234  Номер паспорта - 123456 | В таблице отображается добавленный сотрудник | В таблице отображается добавленный сотрудник |
| Табельный номер – "abc",  Фамилия - Петров,  Имя - Петр,  Отчество - Петрович,  Дата рождения - 12.12.1992  Код пола - 01 (Выбирается из списка)  Контактный номер телефона - +7(231)232-13-12  Адрес регистрации - г.Москва, ул.Ямская, д.2, кв.56  Адрес фактического проживания - г.Москва, ул.Ямская, д.2, кв.56  Караул - 2  Код пожарной части - 4  Серия паспорта - 1234  Номер паспорта - 123456 | В таблице не произойдет сохранения записи | В таблице не произойдет сохранения записи |
| Табельный номер – 2, Фамилия - "",  Имя - "",  Отчество - "",  Дата рождения - 12.12.1992  Код пола - 01 (Выбирается из списка)  Контактный номер телефона - +7(231)232-13-12  Адрес регистрации - г.Москва, ул.Ямская, д.2, кв.56  Адрес фактического проживания - г.Москва, ул.Ямская, д.2, кв.56  Караул - 2  Код пожарной части - 4  Серия паспорта - 1234  Номер паспорта - 123456 | "Пустая строка. Повторите ввод" | "Пустая строка. Повторите ввод" |
| Редактирование | Табельный номер – 2, Фамилия - Петров,  Имя - Петр,  Отчество - Петрович,  Дата рождения - 12.12.1992  Код пола - 01 (Выбирается из списка)  Контактный номер телефона - +7(231)232-13-12  Адрес регистрации - г.Москва, ул.Ямская, д.2, кв.56  Адрес фактического проживания - г.Москва, ул.Ямская, д.6, кв.56  Караул - 2  Код пожарной части - 4  Серия паспорта - 1234  Номер паспорта - 123456 | Отображение измененных данных об адресе фактического проживания в списке | Отображение измененных данных об адресе фактического проживания в списке |
| Табельный номер – "abc",  Фамилия - Петров,  Имя - Петр,  Отчество - Петрович,  Дата рождения - 12.12.1992  Код пола - 01 (Выбирается из списка)  Контактный номер телефона - +7(231)232-13-12  Адрес регистрации - г.Москва, ул.Ямская, д.2, кв.56  Адрес фактического проживания - г.Москва, ул.Ямская, д.2, кв.56  Караул - 2  Код пожарной части - 4  Серия паспорта - 1234  Номер паспорта - 123456 | В таблице не произойдет изменения записи | В таблице не произойдет изменения записи |
| Удаление | Табельный номер – 3, Фамилия - Иванов,  Имя - Василий,  Отчество - Петрович,  Дата рождения - 12.12.1992  Код пола - 01 (Выбирается из списка)  Контактный номер телефона - +7(231)232-13-12  Адрес регистрации - г.Москва, ул.Ямская, д.2, кв.56  Адрес фактического проживания - г.Москва, ул.Ямская, д.2, кв.56  Караул - 2  Код пожарной части - 4  Серия паспорта - 4321  Номер паспорта - 654321 | Удаление сотрудника из таблицы | Удаление сотрудника из таблицы |
| Код пожарной части – 8 | "Удаление не выполнено. Не существует" | "Удаление не выполнено. Не существует" |
| Поиск | Код пожарной части – 4 | Отображение информации о пожарной части | Отображение информации о пожарной части |
| Код пожарной части – 8 | "Не найдено" | "Не найдено" |
| **Вход в систему** | | | |
| Вход в систему | Логин – user1,  Пароль - user123 | "Авторизация выполнено успешно" | "Авторизация выполнено успешно" |
| Логин – uhdfb,  Пароль - djncjks | "Логин или пароль неверные" | "Логин или пароль неверные" |

# 7. Заключение

По окончании создания информационной системы для учета пожаров, были выполнены следующие задачи:

* выполнен обзор аналога разрабатываемой информационно системы;
* выявлены функциональные и нефункциональные требования;
* построены следующие диаграммы:

1. диаграмма прецедентов;
2. диаграммы активностей;
3. диаграммы последовательностей.

* спроектирована и создана база данных;
* описан интерфейс Приложения в Windows Forms;
* написано Приложение, использованы запросы, выгрузка донесения в Word и т.п.;
* выполнено тестирование Приложения,
* написана документация к разработанному Приложению.

Для того, чтобы пользователь понимал, как работать с Приложением было написано «Руководство пользователя» (приложено отдельным файлом к отчету).

# Библиографический список

1. Федеральный закон "О службе в федеральной противопожарной службе Государственной противопожарной службы и внесении изменений в отдельные законодательные акты Российской Федерации" от 23.05.2016 N 141-ФЗ (Дата обращения: 4.10.2022).
2. Указ Президента РФ от 26 июня 2020 г. № 424 "Об утверждении перечня типовых должностей в федеральной противопожарной службе Государственной противопожарной службы и соответствующих этим должностям специальных званий” (Дата обращения: 4.10.2022).
3. Приказ МЧС России от 20 октября 2017 г. № 452 "Об утверждении Устава подразделений пожарной охраны" (Дата обращения: 4.10.2022).
4. Приказ МЧС России от 16 октября 2017 г. № 444 "Об утверждении Боевого устава подразделений пожарной охраны, определяющего порядок организации тушения пожаров и проведения аварийно-спасательных работ" (Дата обращения: 4.10.2022).
5. Приказ №625 от 24.12.2018 Министерства Российской федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС России).
6. Свид. 2018617462 Российская Федерация. Свидетельство об официальной регистрации программы для ЭВМ. Автоматизированная аналитическая система поддержки и управления контрольно-надзорными органами МЧС России (ААС КНД) / Порсов Д. М., Дунаев Д. К., Харитонов А. Б., Козлов А. А., Еникеев Р. Ш., Полехин П.В.; правообладатель Российская Федерация, от имени которой выступает Министерство Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий (МЧС России) (RU). – №2018612050; заявл. 26.02.18; опубл. 25.06.18, Реестр программ для ЭВМ. – 1 с.

# Приложение А

**Техническое задание на разработку системы**

Разработка информационной системы для учета пожаров

Техническое задание

Листов 5

1. **Введение**
   1. **Наименования программы**

Наименование программы: “Информационная система для учета пожаров”. Далее по тексту – Система.

* 1. **Краткая характеристика области применения**

Система предназначена для применения в органах, обеспечивающих пожарную безопасность (пожарные части).

1. **Основания для разработки**
   1. **Наименование и (или) условное обозначение темы разработки**

Наименование темы разработки: "Разработка информационной системы для учета пожаров".

1. **Назначение разработки**
   1. **Функциональное назначение**

Функциональным назначением системы является упрощение процесса формирования донесений о пожарах.

* 1. **Эксплуатационное назначение**

Разрабатываемая программа позволяет добавлять, редактировать, удалять данные о пожарной части, пожарной технике, сотрудниках, вызовах и выездах на пожар, а также позволяет сгенерировать донесение о произошедшем пожаре.

1. **Требования к программе**
   1. **Требования к функциональным характеристикам**

Программа должна обеспечить возможность выполнения перечисленных ниже функций.

Возможность войти в систему, используя свой логин и пароль.

Возможность добавлять, редактировать, просматривать, удалять данные о новом сотруднике в систему.

Возможность добавлять, редактировать, просматривать, удалять данные о вызовах на пожар.

Возможность добавлять, редактировать, просматривать, удалять данные о выездах на пожар.

Возможность добавлять, редактировать, просматривать, удалять данные о дежурных сменах.

Возможность формировать донесения о пожарах с последующим выводом этого донесения в Word.

Возможность добавлять, редактировать, просматривать, удалять данные о пожарных частях и пожарных машинах.

* 1. **Требования к надежности**

Система должна запрашивать пароль при входе в качестве сотрудника МЧС.

Организация органами, обеспечивающими пожарную безопасность, бесперебойного питания технических средств в пожарных частях.

* 1. **Условия эксплуатации**

**Климатические условия эксплуатации**

Требований к климатическим условиям эксплуатации не предъявляется.

**Требования к видам обслуживания**

Обслуживание не требуется.

**Требования к численности и квалификации персонала**

Для управления системой достаточно одного человека, способного запустить на сервере систему управления базами. Требуемая квалификация пользователя - оператор ЭВМ.

1. **Требования к программной документации**

**Техническое задание (ГОСТ 19.201-78)**

**Руководство пользователя**

**Руководство программиста**

**Отчет**

1. **Технические показатели**

Использование разрабатываемой системы сократит время, затрачиваемое на процесс формирования донесений о пожаре.

1. **Стадии и этапы разработки**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Содержание работ** | **Порядок приемки и документы** | **Сроки** | **Ответственный** |
| 1. Анализ задачи и разработка требований к системе | Описание автоматизируемых бизнес-процессов (диаграмма активностей), обзор аналогов / алгоритмов, описание документов и справочников предметной области, разработка требований к системе, составление диаграммы прецедентов | Утверждение анализа предметной области | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |
| 2. Создание модели данных | Построение концептуальной и логической моделей базы данных | Утверждение моделей данных | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |
| 3. Проектирование приложения | Построение модели поведения системы (диаграммы последовательностей), проектирование пользовательского интерфейса и системы в целом | Утверждение приложения | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |
| 4. Разработка запросов к данным | Разработать запросы определенных типов | Утверждение запросов | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |
| 5. Проектирование отчетов | Генерация приложением нужного отчета | Утверждение генерации отчета | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |
| 6. Тестирование и отладка приложения | Выполняется тестирование приложения, результаты документируются | Утверждения результатов тестирования | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |
| 7. Подготовка документации | Составление руководства пользователя и программиста | Утверждение документации | 1.01.2023 - 1.06.2023 | Исполнитель, заказчик |

1. **Порядок контроля и приемки**

Прием приложения происходит в защитные даты с 5.06.2023 по 7.06.2023.

# Приложение В

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

{

SqlConnection connection = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=MchsProekt;Integrated Security=True");

connection.Open();

SqlCommand cmd = new SqlCommand(@"select V.[Адрес пожара]

, V.[Время обращения заявителя]

, VOP.[Вид объекта пожара]

, V.[Фамилия заявителя]

, V.[Имя заявителя]

, V.[Отчество заявителя]

, V.[Номер телефона заявителя]

, VP.[Дата выезда]

, VP.[Время поступления сообщения]

, VP.[Время прибытия на место пожара]

, VP.[Время подачи первого ствола]

, VP.[Время локализации]

, VP.[Время ликвидации]

, VP.[Время прибытия обратно в часть]

, VP.[Время постановки в боевой расчет]

, VP.[Дата окончания выезда]

, PP.[Наименование причины]

, VL.[Наименование категории]

, VP.[Количество погибших людей]

, VP.[Количество погибших детей]

, VP.[Количество погибших работников пожарной охраны]

, VP.[Количество уничтоженных/поврежденных сооружений]

, VP.[Количество уничтоженных жилых квартир]

, VP.[Количество уничтоженной автотракторной и другой техники]

, VP.[Спасено людей на пожаре]

, VP.[Спасено материальных ценностей]

from [MchsProekt].[dbo].[Вызов\_на\_пожар] as V

left join [MchsProekt].[dbo].[Выезд на пожар] as VP

on V.[Код вызова] = VP.[Код вызова]

left join [MchsProekt].[dbo].[Вид\_объекта\_пожара] as VOP

on V.[Код вида объекта] = VOP.[Код]

left join [MchsProekt].[dbo].[Причина пожара] as PP

on VP.[Код причины] = PP.[Код причины]

left join [MchsProekt].[dbo].[Виновное лицо] as VL

on VP.[Код категории виновного лица] = VL.[Код категории виновного лица]", connection);

SqlDataReader reader = cmd.ExecuteReader();

var word = new Word.Application();

word.Visible = true;

Document doc = word.Documents.Add(System.Reflection.Missing.Value);

const int FIELDS\_IN\_ROW\_COUNT = 26;

var rowsCount = reader.FieldCount / FIELDS\_IN\_ROW\_COUNT;

var reportsCount = 0;

while (reader.Read())

{

reportsCount++;

TypeHeader(word);

SetTextStyle(word);

var address = reader[0];

word.Selection.TypeText("Адрес пожара: " + address.ToString());

word.Selection.TypeParagraph();

var time = reader[1];

DateTime time8 = (DateTime)time;

time = time8.ToLongTimeString();

word.Selection.TypeText("Время обнаружения пожара: " + time.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var mesto = reader[2];

word.Selection.TypeText("Место возникновения пожара: " + mesto.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var person = reader[3];

word.Selection.TypeText("Фамилия заявителя: " + person.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var person1 = reader[4];

word.Selection.TypeText("Имя заявителя: " + person1.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var person2 = reader[5];

word.Selection.TypeText("Отчество заявителя: " + person2.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var phone = reader[6];

word.Selection.TypeText("Номер телефона: " + phone.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var data = reader[7];

word.Selection.TypeText("Дата выезда: " + data.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time1 = reader[8];

DateTime time9 = (DateTime)time1;

time1 = time9.ToLongTimeString();

word.Selection.TypeText("Время поступления сообщения: " + time1.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time2 = reader[9];

DateTime time10 = (DateTime)time2;

time2 = time10.ToLongTimeString();

word.Selection.TypeText("Время прибытия на место пожара: " + time2.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time3 = reader[10];

DateTime time11 = (DateTime)time3;

time3 = time11.ToLongTimeString();

word.Selection.TypeText("Время подачи первого ствола: " + time3.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time4 = reader[11];

DateTime time12 = (DateTime)time4;

time4 = time12.ToLongTimeString();

word.Selection.TypeText("Время локализации: " + time4.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time5 = reader[12];

DateTime time13 = (DateTime)time5;

time5 = time13.ToLongTimeString();

word.Selection.TypeText("Время ликвидации: " + time5.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time6 = reader[13];

DateTime time14 = (DateTime)time6;

time6 = time14.ToLongTimeString();

word.Selection.TypeText("Время прибытия обратно в часть: " + time6.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var time7 = reader[14];

DateTime time15 = (DateTime)time7;

time7 = time15.ToLongTimeString();

word.Selection.TypeText("Время постановки в боевой расчет: " + time7.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var data2 = reader[15];

word.Selection.TypeText("Дата окончания выезда: " + data2.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var prichina = reader[16];

word.Selection.TypeText("Причина: " + prichina.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var vinovny = reader[17];

word.Selection.TypeText("Виновное лицо: " + vinovny.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan1 = reader[18];

word.Selection.TypeText("Погибло людей: всего " + quan1.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan2 = reader[19];

word.Selection.TypeText("Погибло детей: " + quan2.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan3 = reader[20];

word.Selection.TypeText("Погибло работников пожарной охраны: " + quan3.ToString());

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan4 = reader[21];

word.Selection.TypeText("Количество уничтоженных/поврежденных сооружений: " + quan4.ToString() + " ед.");

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan5 = reader[22];

word.Selection.TypeText("Количество уничтоженных жилых квартир: " + quan5.ToString() + " ед.");

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan6 = reader[23];

word.Selection.TypeText("Количество уничтоженной автотракторной и другой техники: " + quan6.ToString() + " ед.");

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan7 = reader[24];

word.Selection.TypeText("Спасено людей на пожаре: " + quan7.ToString() + " чел.");

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

var quan8 = reader[25];

word.Selection.TypeText("Спасено материальных ценностей: " + quan8.ToString() + " руб.");

word.Selection.Font.Bold = 0;

word.Selection.TypeParagraph();

if (reportsCount < rowsCount)

{

word.Selection.InsertBreak(WdBreakType.wdPageBreak);

}

}

}

private void TypeHeader(Word.Application word)

{

SetHeaderStyle(word);

word.Selection.TypeText("Донесение о пожаре №\_\_\_\_\_\_\_\_\_\_\_\_\_");

word.Selection.TypeParagraph();

word.Selection.TypeText("Составил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

word.Selection.TypeParagraph();

}

private void SetHeaderStyle(Word.Application word)

{

word.Selection.Font.Size = 16;

word.Selection.Font.Bold = 3;

word.Selection.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphCenter;

}

private void SetTextStyle(Word.Application word)

{

word.Selection.Font.Size = 13;

word.Selection.Font.Bold = 0;

word.Selection.ParagraphFormat.Alignment = WdParagraphAlignment.wdAlignParagraphLeft;

}

1. Согласно Rational Unified Proccess, Version 2003.06.13 - методология разработки программного обеспечения, созданная компанией Rational Software [↑](#footnote-ref-1)
2. Вигерс К., Битти Дж. Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М. : Издательство «Русская редакция» ; СПб. : БХВ-Петербург, 2014 — 736 стр. : ил. [↑](#footnote-ref-2)