|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**ТЕХНИЧЕСКИЙ ПРОЕКТ**

на разработку

информационной системы по учёту автотранспортных

происшествий

2024

**Содержание**

[1. Требования к системе 3](#_Toc189938754)

[1.1. Функциональные требования 6](#_Toc189938755)

[1.2. Нефункциональными требованиями к ИС являются: 7](#_Toc189938756)

[2. Уточнение структуры данных 13](#_Toc189938757)

[2.1. Выделение сущностей 13](#_Toc189938758)

[2.2. Построение логической ER-диаграммы 13](#_Toc189938759)

[2.3. Построение диаграммы потоков данных 14](#_Toc189938760)

[3. Укрупненный сценарий использования 16](#_Toc189938761)

[4. Разработка алгоритма решения задачи 17](#_Toc189938762)

[5. Уточнение структуры системы 22](#_Toc189938763)

[6. Определение языка разработки 23](#_Toc189938764)

[7. Определение системы управления базами данных 24](#_Toc189938765)

[8. Определение библиотеки для создания пользовательского интерфейса 25](#_Toc189938766)

# Требования к системе

Ниже представлены диаграммы IDEF0 детализирующие автоматизированные процессы информационной системы.

На рисунке 1 отображена работа ИС по учёту автотранспортных происшествий. Инспектор начинает деятельность с запросов на регистрацию автотранспортных происшествий от участников. После поступления запросов, инспектор приступает к обработке и внесению информации о происшествиях в систему. После этого начинается деятельность сотрудников ГИБДД, она регулируется с помощью ГОСТов: ГОСТ Р 41.77-2015, ГОСТ Р 41.97-2014 и ГОСТ Р ИСО/МЭК 27002-2019.

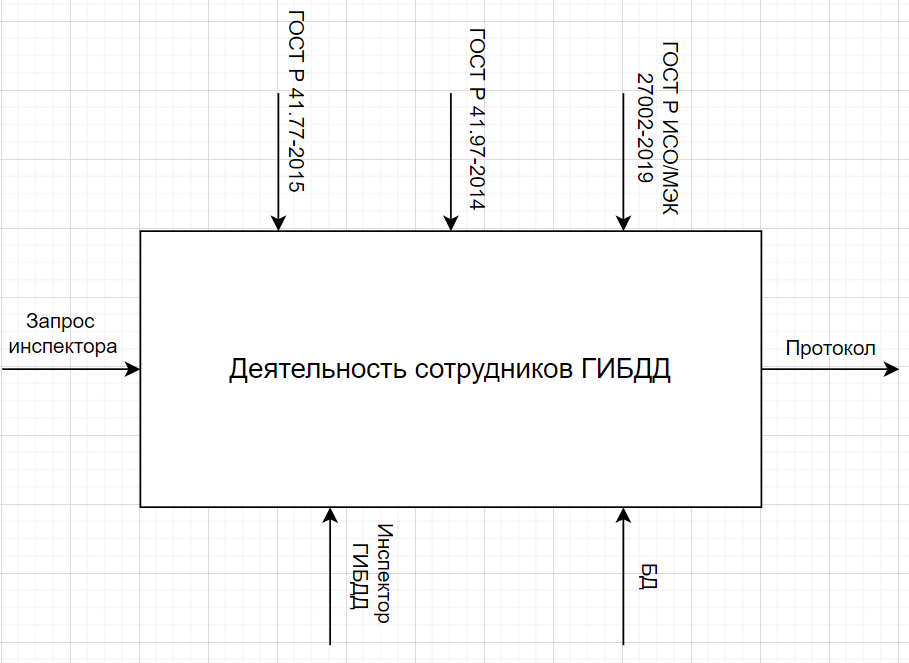


Рисунок 1 – Диаграмма IDEF0

На рисунке 2 представлена декомпозиция процесса «Поиска происшествия» для пользователя. На данной диаграмме более детально показана работа ИС. После запроса сотрудника ГИБДД, инспектор производит поиск происшествия в системе. После нахождения происшествий, происходит их выбор для дальнейшего оформления протокола. Список с информацией всех происшествиях заносятся в БД, от которых формируются отчеты.

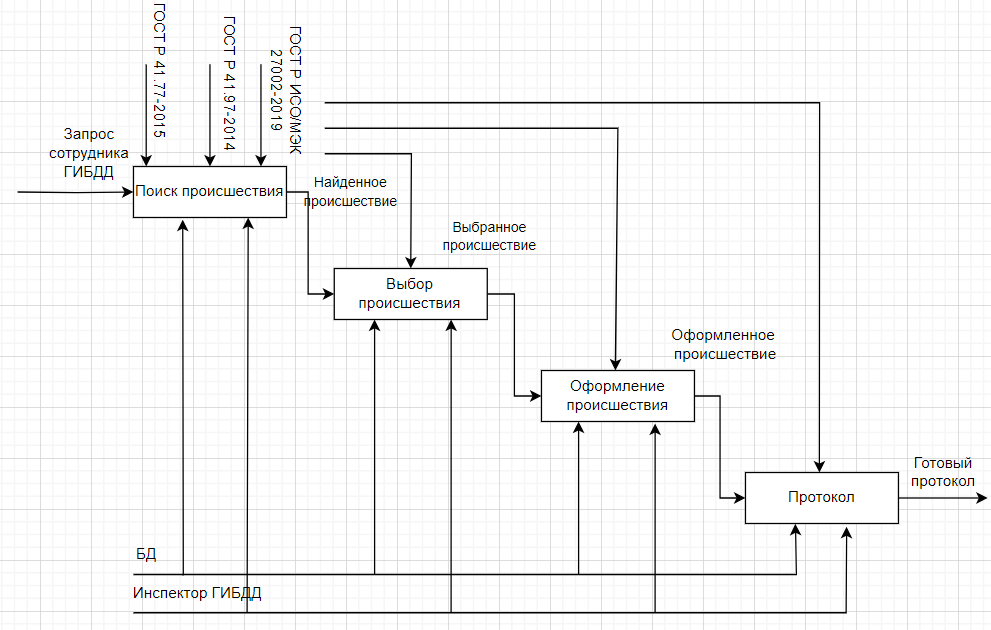


Рисунок 2 – Декомпозиция процесса «Поиск происшествия»

На рисунке 3 представлена декомпозиция процесса нажатия кнопки «Редактировать происшествие» для пользователя. На данной диаграмме показан процесс изменения списка «Происшествия». При нажатии кнопки «Редактировать происшествие», открывается редактор происшествий, после этого инспектор может внести необходимые изменения. После внесения изменений, инспектор сохраняет и просматривает изменения, чтобы всё успешно сохранилось.

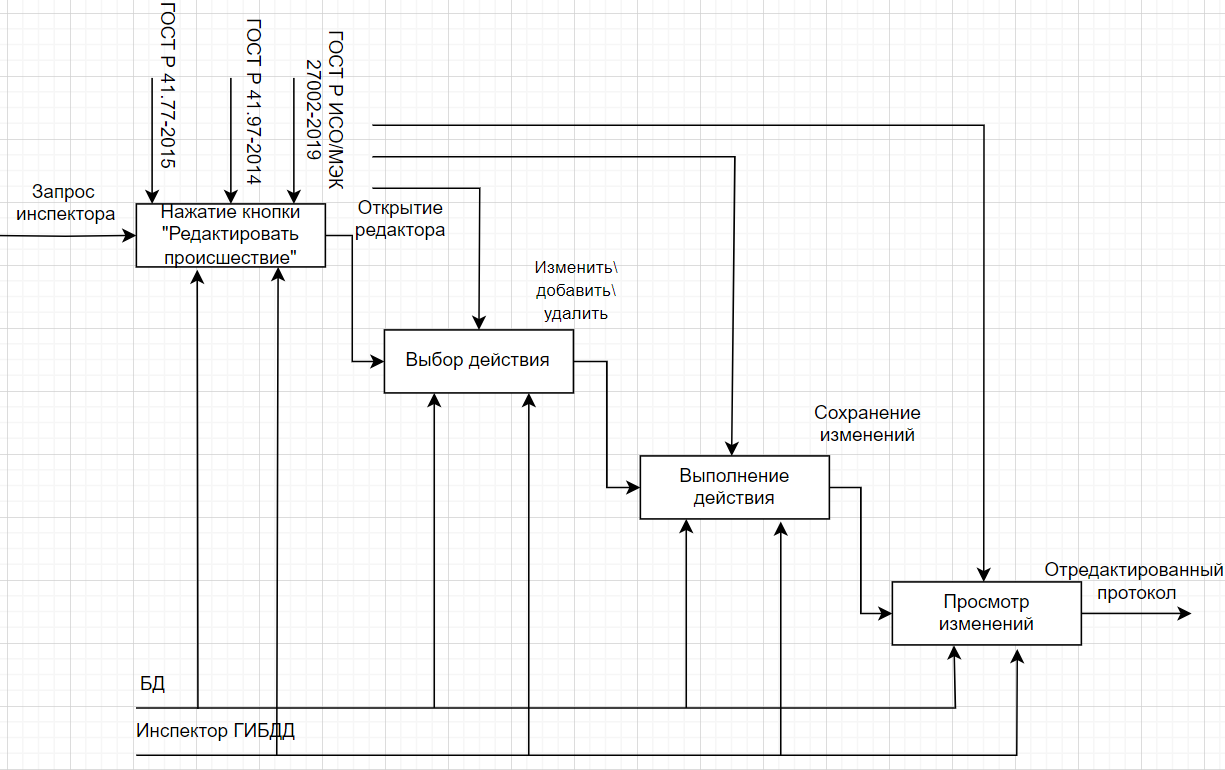


Рисунок 3 – Декомпозиция процесса нажатия кнопки «Редактировать происшествие»

На рисунке 4 представлена декомпозиция процесса нажатия кнопки «Отчеты» для пользователя. При нажатии кнопки «Отчеты», открывается окно с последним сформированным отчетом. В этом окне с помощью кнопки «Сформировать отчет», при нажатии на кнопку открывается подокно, где можно выбрать период. После внесенных критерий и сохранения их, формируется отчет. Инспектор просматривает сформированный отчет чтобы убедиться, что все успешно сохранилось.

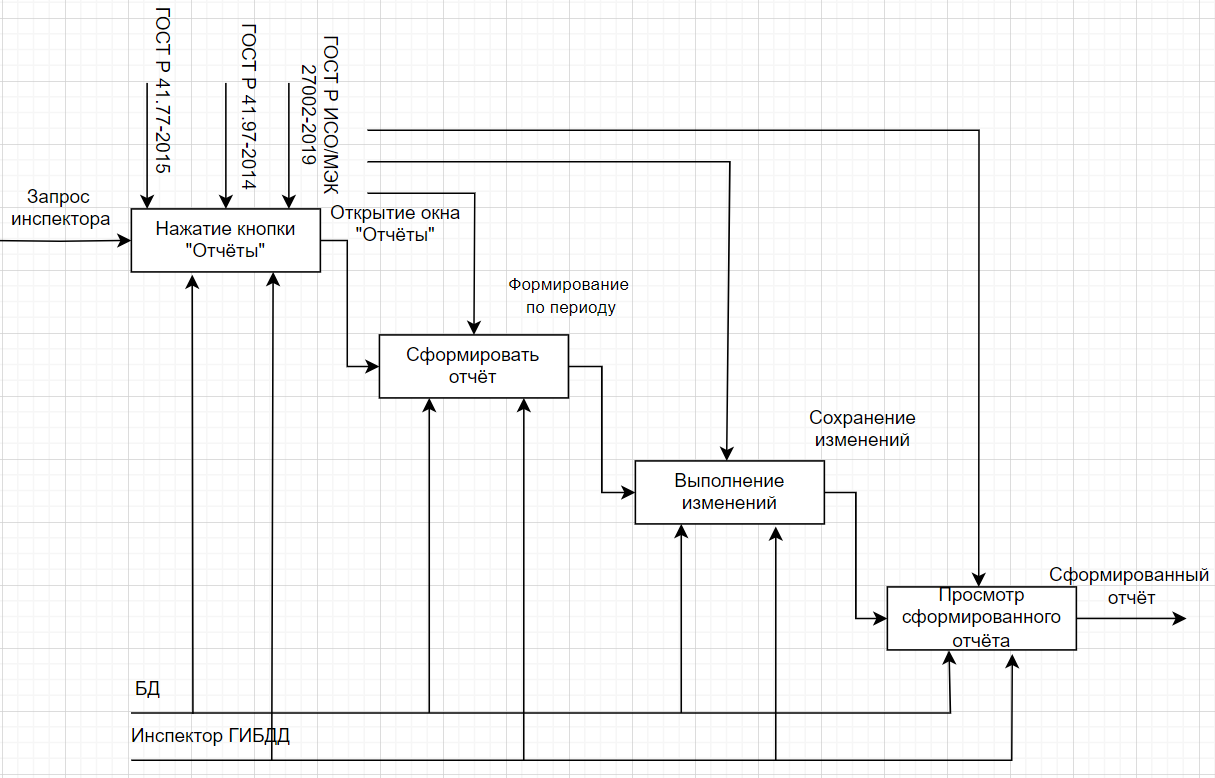


Рисунок 4 – Декомпозиция процесса нажатия кнопки «Отчеты»

Функциональные требования

Система реализации учёта автотранспортных происшествий должна предоставлять возможность:

* Инспектору иметь доступ к полному и актуальному списку завершённых происшествий, зарегистрированных в системе, с соответствующей информацией о каждом завершённом происшествии.
* Инспектору предоставляется доступ к полному и актуальному списку происшествий, находящихся в расследовании, зарегистрированных в системе, с соответствующей информацией о каждом происшествии в статусе «В расследовании».
* Инспектору возможность формирования отчета по происшествиям.
* Инспектору управлять списком происшествий, имеющим статус «Завершённые», иметь функции изменения, добавления и удаления происшествий из списка.
* Инспектору управлять списком происшествий, имеющим статус «В расследовании», иметь функции изменения, добавления и удаления происшествий из списка.

Создаваемая система будет работать с БД, которая содержит всю информацию о происшествиях, участниках и статусах происшествий.

Нефункциональными требованиями к ИС являются:

* требования к производительности:

Система должна обеспечивать быстрый доступ к спискам происшествий, а также быструю обработку запросов на создание происшествий и отчётов.

* требования к надежности:

1. ИС должна быть стабильной и надежной, чтобы избежать потери данных или сбоев в работе.;
2. Каждый элемент интерфейса должен быть рабочим, срабатывать при нажатии, что бы не возникало проблем с использованием;
3. интерфейс должен четко отображать все компоненты приложения на экране.

**1.3. Интерфейсные требования**

**1.3.1. Пользовательский интерфейс и взаимодействие**

Интерфейс должен быть простым с понятными элементами управления и навигацией. Все данные должны быть представлены в удобной для восприятия форме (графики, таблицы, списки).

Далее представлены макеты экранных форм, на которых представлен примерный интерфейс информационной системы.

Ниже представлен прототип начального экрана информационной системы.



Рисунок 5 – Прототип экранной формы для инспектора

На ней присутствуют следующие элементы:

* в левой верхней части экрана располагается надпись «Список происшествий»;
* ниже располагается поле для списка происшествий, которые являются завершёнными;
* ещё ниже находится поле для списка происшествий, которые находятся в расследовании;
* под списком происшествий находится кнопка «Отчеты»
* в правой верхней части экрана находится строка «Поиск происшествий (введите id происшествия)»;
* ниже находится поле для найденных происшествий;
* в левом нижнем углу расположена кнопка «Редактировать происшествие»;
* в правом нижнем углу расположена кнопка «Добавить происшествие».

На рисунке 6 представлена экранная форма, которая следует после нажатия кнопки «Добавить происшествие» инспектором.

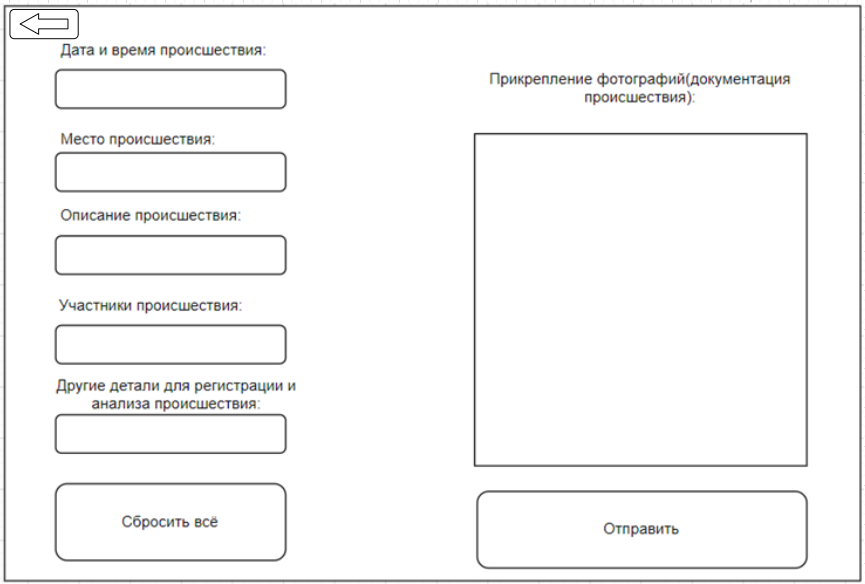


Рисунок 6 – Прототип экранной формы для регистрации нового автотранспортного происшествия

На ней присутствуют следующие компоненты:

* в левом верхнем углу кнопка, для возвращения на предыдущую форму;
* в верхней части в левом углу надпись «Дата и время происшествия»;
* ниже расположено поле для ввода даты и времени происшествия;
* ниже находится надпись «Место происшествия»;
* ниже расположено поле для ввода адреса места происшествия;
* ниже находится надпись «Описание происшествия»;
* ниже расположено поле для ввода информации о происшествии;
* ниже находится надпись «Участники происшествия»;
* ниже расположено поле для ввода информации о участниках происшествия;
* ниже находится надпись «Другие детали для регистрации и анализа происшествия»;
* ниже расположено поле для ввода дополнительной информации;
* ниже находится кнопка «Сбросить всё»;
* в верхнем правом углу надпись «Прикрепление фотографий (документация происшествия)»;
* ниже расположено поле для загрузки фотографий происшествия;
* в правом нижнем углу находится кнопка «Отправить».

На рисунке 7 представлена экранная форма, которая следует после нажатия кнопки «Редактировать происшествие» инспектором.

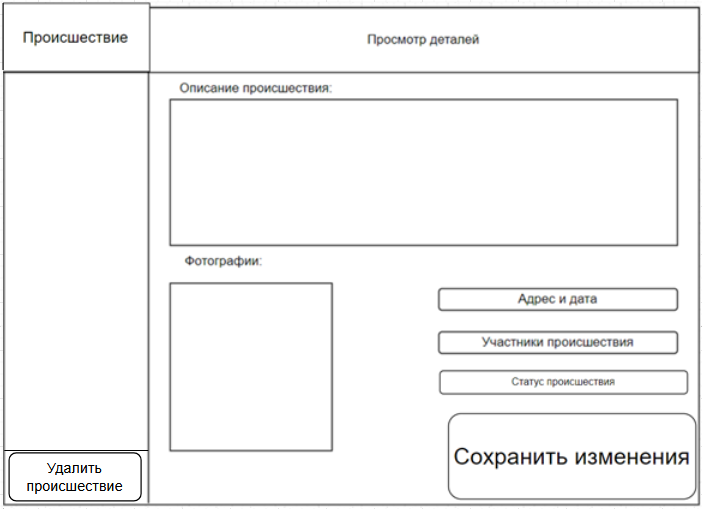


Рисунок 7 – Прототип экранной формы редактирования происшествия инспектором

На экране присутствуют следующие элементы:

* в левом верхнем углу расположена надпись «Происшествие»;
* в левом нижнем углу находится кнопка «Удалить происшествие»;
* в правом верхнем углу находится надпись «Просмотр деталей»;
* ниже находится надпись «Описание происшествия»;
* ниже поле просмотра описания происшествия;
* ниже расположена надпись «Фотографии»;
* в левом нижнем углу поле для просмотра фотографий;
* в правом углу находится поле «Адрес и дата»;
* ниже расположено поле «Участники происшествия»;
* ниже находится поле «Статус происшествия»;
* в правом нижнем углу находится кнопка «Сохранить изменения».

На рисунке 8 представлена экранная форма, которая следует после нажатия кнопки «Отчёты» инспектором.

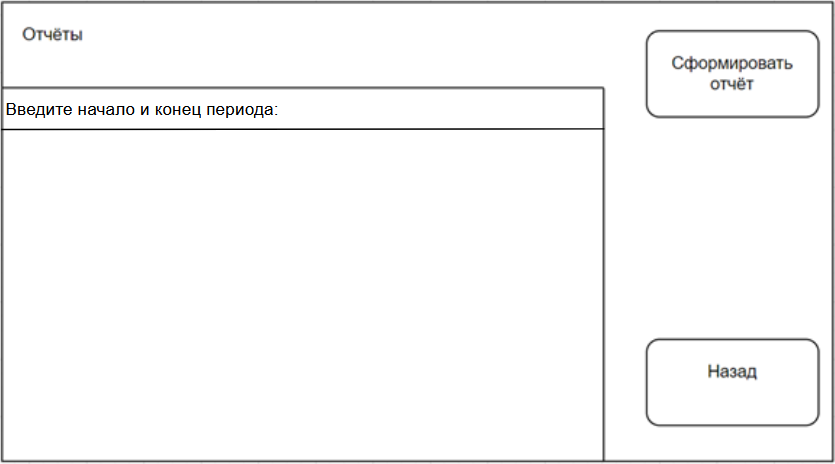


Рисунок 8 – Прототип экранной формы создания отчётов

На экране присутствуют следующие элементы:

* в левом верхнем углу находится надпись «Отчёты»;
* под надписью находится поле для ввода начала и конца отчетного периода;
* ниже находится поле для вывода сформированных отчётов;
* в правом верхнем углу находится кнопка «Сформировать отчёт»;
* в правом нижнем углу находится кнопка «Назад».

**1.3.2. Поддержка мультиязычности**

Интерфейс должен поддерживать только русский язык, мультиязычность не предусмотрена.

Уточнение структуры данных

В ходе уточнения данных были выполнены следующие задачи, показанные в Таблице 2.

Таблица 2 – Задачи для уточнения структуры данных.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ задачи** | **Название** | **Содержание** | **Результат** |
| 1 | Выделение сущностей | Выделение основных сущностей программы, а также их описание | Готовая таблица с сущностями и их описанием |
| 2 | Построение логической ER-диаграммы | Схематическое представление модели базы данных в виде логической ER-диаграммы | Готовая логическая ER-диаграмма |
| 3 | Построение диаграммы потоков данных | Определить, как будут передвигаться данные в программе; построить DFD диаграмму | Готовая диаграмма потоков данных (DFD) |

Выделение сущностей

Краткое описание основных сущностей показано в Таблице 3.

Таблица 3 – Сущности и их описание

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Название сущности** | **Описание сущности** |
| 1 | Происшествия | Информация о зарегистрированных автотранспортных происшествиях, включая дату, время, место происшествия, описание и статус. |
| 2 | Участники | Данные о лицах, участвовавших в происшествиях, включая их персональную информацию, транспортные средства и контактные данные. |
| 3 | Отчёты | Сводные данные о происшествиях, сгруппированные по периоду. |
| 4 | Статус | Текущий этап обработки происшествия, включая два варианта: «В расследовании» и «Завершённые». |

Построение логической ER-диаграммы

Логическая ER-диаграмма показана на рисунке 9.

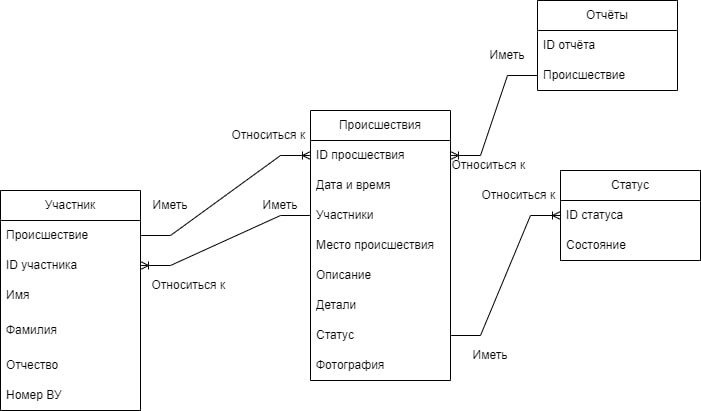


Рисунок 9 – Логическая модель базы данных

Построение диаграммы потоков данных

DFD (Data Flow Diagram) – это графическое представление потоков данных в системе, показывающее, как информация перемещается через различные процессы. DFD используется для моделирования и анализа информационных систем, позволяя представить их структуру, функции и потоки данных между различными компонентами.

На рисунке 10 представлена DFD-диаграмма для учёта автотранспортных происшествий.

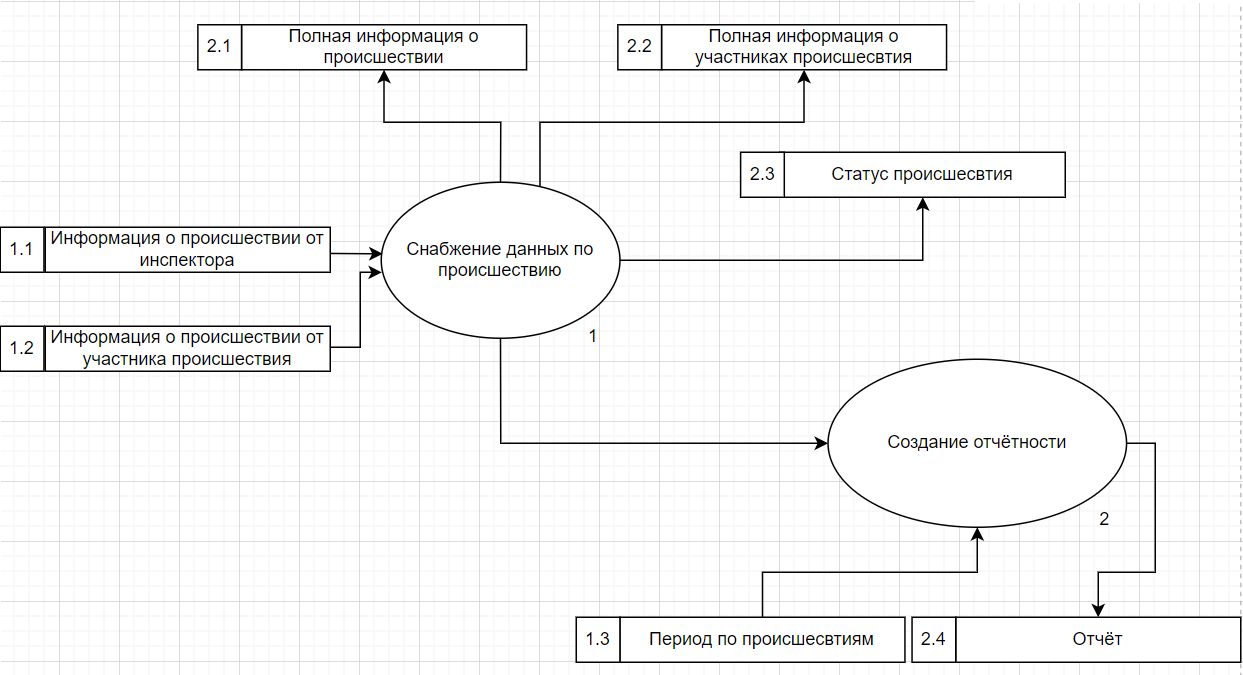


Рисунок 10 – DFD диаграмма

На данной диаграмме присутствуют 2 блока работы: «Снабжение данных по происшествию», «Создание отчетности»

«Снабжение данных по происшествию» содержит внешние данные: «Информация о происшествии от инспектора» (информация, которую заносит в систему инспектор), «Информация о происшествии от участников происшествия» (информация, которую сообщают участники происшествия инспектору).

«Создание отчетности» содержит внешние данные «Период по происшествиям».

Блок «Снабжение данных по происшествию» содержит хранилище данных:

* «Полная информация о происшествии», где находится такая информация о происшествии как, дата и время, место происшествия, описание, статус, фотографии и другие детали происшествия.
* «Полная информация о участниках происшествия», где находится такая информация о участниках происшествия как, имя, фамилия, отчество и номер водительского удостоверения.
* «Статус происшествия», где находится такая информация о статусах происшествия как, состояние статуса происшествия.

Блок «Создание отчетности» содержит хранилище данных: «Отчет», где можно создать отчет по происшествиям, там находится информация о дате и времени происшествий за определенный период времени.

Укрупненный сценарий использования

Инспектор входит в информационную систему, где ему отображается список зарегистрированных автотранспортных происшествий. Система предоставляет возможности для добавления новых происшествий, редактирования существующих данных, обновления статуса и формирования отчетов. С помощью гибкого интерфейса инспектор выбирает необходимое действие.

Для добавления или редактирования происшествий в правой части экрана доступна кнопка «Добавить»/«Изменить». При нажатии на нее открывается форма, где можно заполнить или обновить данные о происшествии, включая дату и время, место происшествия, описание и статус, такой как «В расследовании» или «Завершённые». После внесения изменений обновлённая информация сразу отображается в общем списке.

Для управления участниками происшествий инспектор может выбрать соответствующую кнопку в левой части экрана. Открывается окно со списком участников, где можно добавлять новых, изменять данные существующих или удалять их из списка. При добавлении нового участника вводятся такие данные, как ФИО, информация о транспортном средстве и контактные данные. Все изменения автоматически сохраняются и обновляются в системе.

Система позволяет инспектору использовать фильтрацию и поиск для удобного отображения данных. Например, можно отфильтровать список происшествий по их статусу — «В расследовании» или «Завершённые». Для сброса фильтрации достаточно нажать кнопку «Сбросить фильтрацию».

В разделе отчетов инспектор может сгенерировать сводные данные. Нажатие на кнопку «Отчеты» открывает окно с возможностью выбора параметров формирования: период (например, месяц или три месяца), категория происшествий и тип диаграммы. После выбора параметров нажимается кнопка «Сформировать отчет», и система создает отчет с отображением данных в удобной визуальной форме.

Инспектор также может обновлять статус происшествий. Выбрав конкретное происшествие и нажав кнопку «Изменить», он может изменить статус на «Завершённые», если расследование завершено. Обновленный статус сразу отображается в общем списке происшествий.

Удаление данных из системы также доступно через соответствующую кнопку. Инспектор может удалить происшествие, участника или запись отчета, после чего система обновляет данные, отображая актуальную информацию.

Разработка алгоритма решения задачи

Схема алгоритма «Добавление происшествия» представлена на рисунке 11.

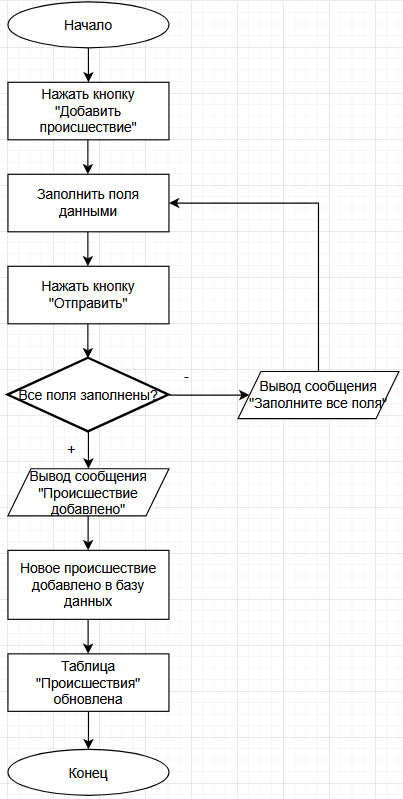


Рисунок 11 – схема алгоритма «Добавление происшествия»

Аналогично данному алгоритму работают все процессы добавления данных в БД.

Схема алгоритма «Изменение происшествия» представлена на рисунке 12.

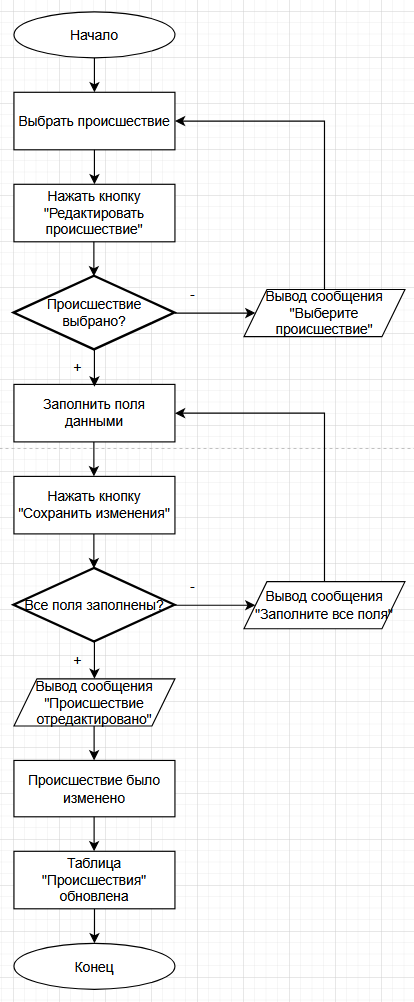


Рисунок 12 – схема алгоритма «Изменение происшествия»

Аналогично данному алгоритму работают все процессы изменения данных в БД.

Схема алгоритма «Удаление происшествия» представлена на рисунке 13.

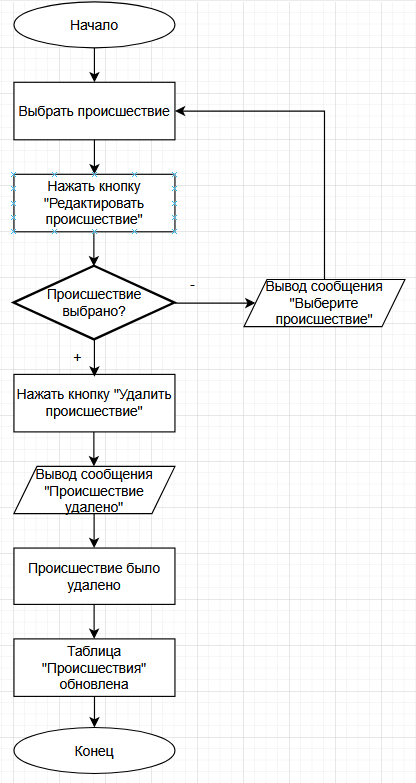


Рисунок 13 – схема алгоритма «Удаление происшествия»

Аналогично данному алгоритму работают все процессы удаления данных в БД.

Схема алгоритма «Формирование отчёта» представлена на рисунке 14

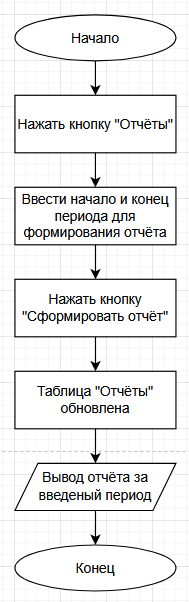
.

Рисунок 13 – схема алгоритма «Формирование отчёта»

Схема алгоритма «Поиск происшествия по id» представлена на рисунке 14.

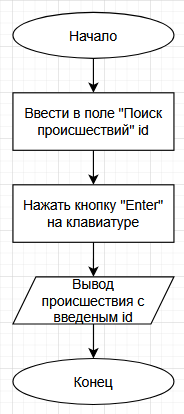


Рисунок 14 – схема алгоритма «Поиск происшествия по id»

Уточнение структуры системы

При уточнении структуры системы была описана подробная информация об основных объектах показана в Таблице 4.

Таблица 4 – Информация об объектах

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Название сущности | Название атрибута | Тип |
| 1 | Список происшествий | Дата и время | Дата |
| Участники | Целое число |
| Место происшествия | Текст |
| Описание | Текст |
| Детали | Текст |
| Статус | Целое число |
| Фотография | Фотография |
| 2 | Список отчётов | Происшествие | Текст |
| 3 | Список участников | Происшествие | Список чисел |
| Имя | Текст |
| Фамилия | Текст |
| Отчество | Текст |
| Номер ВУ | Целое число |
| 4 | Список статусов | Состояние | Текст |

Определение языка разработки

Для разработки функционала системы будет использован язык программирования Python. Этот язык является высокоуровневым, интерпретируемым, поддерживает объектно-ориентированный подход и имеет широкий набор библиотек, что делает его оптимальным для разработки многофункциональных приложений.

Основные особенности Python:

* **Универсальность и гибкость:** Python подходит для разработки различных приложений, от веб-сервисов до аналитических систем и автоматизации задач.
* **Простота синтаксиса:** синтаксис Python позволяет быстрее разрабатывать и поддерживать код, снижая вероятность ошибок и упрощая восприятие для новых разработчиков, участвующих в развитии системы.
* **Поддержка библиотек и фреймворков:** Python предоставляет большое количество библиотек для решения разнообразных задач, включая обработку данных, веб-разработку и интеграцию с внешними сервисами.
* **Встроенные возможности работы с данными:** Python содержит мощные инструменты для обработки данных, анализа и визуализации, что значительно упрощает ведение учета и отчетности, включая анализ причин аварийности и формирование статистических отчетов.

Преимущества разработки на Python:

* **Широкая поддержка сообщества:** Наличие активного сообщества разработчиков и обширной документации значительно облегчает процесс разработки, отладки и поддержки системы.
* **Множество библиотек для анализа данных:** Такие библиотеки, как pandas, NumPy и Matplotlib, ускоряют процесс обработки и анализа данных.
* **Гибкость интеграции:** Python легко интегрируется с другими языками и системами, что позволяет подключить систему к дополнительным модулям и сервисам.

Использование Python обеспечит высокую скорость разработки, удобство поддержки и гибкость в расширении функционала системы.

Определение системы управления базами данных

Для хранения данных будет использоваться система управления базами данных SQLite, управляемая через SQLiteStudio. SQLite – это легковесная встраиваемая СУБД, не требующая серверной инфраструктуры и идеально подходящая для приложений с небольшим и средним объемом данных.

Основные особенности SQLiteStudio:

* **Встраиваемая база данных:** SQLite не требует серверной настройки, а все данные хранятся в одном файле, что значительно упрощает развертывание и администрирование системы.
* **Поддержка стандартов SQL:** Система поддерживает стандартный SQL-синтаксис, что облегчает выполнение сложных запросов, извлечение необходимых данных и управление структурой базы данных.
* **Удобный графический интерфейс:** SQLiteStudio предоставляет интуитивно понятный графический интерфейс для работы с базой данных, управления таблицами, создания запросов и визуального анализа данных, что упрощает работу с информацией о ДТП для различных пользователей.
* **Легковесность и быстродействие: SQLite** требует минимум ресурсов и обеспечивает высокую производительность при работе с данными о ДТП, обеспечивая быстрый доступ к информации и эффективную обработку запросов.

Преимущества использования SQLiteStudio:

* **Простота установки и использования:** Отсутствие серверной части делает SQLite удобной для быстрого разворачивания и использования в различных условиях.
* **Легкость резервного копирования и восстановления данных:** Данные хранятся в одном файле, что значительно упрощает создание резервных копий и восстановление данных в случае сбоев.
* **Интеграция с языками разработки:** SQLite интегрируется с Python, что позволяет расширять функционал системы значительных затрат и с использованием широких возможностей языка Python.

Определение библиотеки для создания пользовательского интерфейса

Для разработки графического интерфейса системы учета ДТП будет использоваться библиотека PyQt5. PyQt5 предоставляет набор инструментов для создания кросс-платформенных интерфейсов на базе фреймворка Qt, что позволит создать современный, интуитивно понятный и удобный в использовании дизайн.

Основные особенности PyQt5:

* **Обширный набор виджетов:** Библиотека поддерживает множество виджетов, таких как кнопки, текстовые поля, выпадающие списки, таблицы и графики, что упрощает создание удобного и функционального интерфейса для ввода, просмотра и анализа данных.
* **Гибкость и настраиваемость:** PyQt5 позволяет детально настраивать элементы интерфейса, что важно для создания привлекательного и информативного дизайна, облегчающего восприятие данных.
* **Кросс-платформенность:** Приложения на PyQt5 работают на Windows, macOS и Linux, что обеспечивает гибкость в развертывании системы на различных устройствах.
* **Интеграция с Python:** PyQt5 легко взаимодействует с Python-кодом, что обеспечивает удобную связку между интерфейсом и логикой приложения.

Преимущества использования PyQt5:

* **Визуальное проектирование интерфейсов:** Поддержка Qt Designer позволяет создавать интерфейс в визуальном редакторе, ускоряя процесс разработки и прототипирования, а также упрощая внесение изменений в дизайн.
* **Широкая поддержка сообщества:** PyQt5 имеет большую базу ресурсов и документации, а также активное сообщество разработчиков, что делает его использование удобным и понятным, а также обеспечивает быстрый доступ к решениям возникающих проблем.
* **Поддержка сложных элементов интерфейса:** Библиотека позволяет реализовать продвинутые интерфейсы с вкладками, таблицами, картами (с использованием сторонних библиотек) и мультимедийными элементами (например, для просмотра фотографий).

Использование PyQt5 позволит создать привлекательный и функциональный интерфейс для системы учета ДТП, который будет удобен в использовании, поддержке и позволит эффективно работать с данными о дорожно-транспортных происшествиях.