**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ бюджетное образовательное учреждение высшего профессионального образования**

**«Санкт-Петербургский государственный электротехнический университет**

**“ЛЭТИ” им.В.И.Ульянова (Ленина)»**

**КАФЕДРА МОЭВМ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ**

**по дисциплине «Объектно – ориентированное программирование на языке Java»**

Выполнил: Шарапов И.Д.

Факультет: КТИ

Группа: №3312

Руководитель Павловский М.Г.

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2024

Оглавление

[Техническое задание 4](#_Toc185974269)

[Введение 4](#_Toc185974270)

[Основание для разработки 4](#_Toc185974271)

[Назначение разработки 4](#_Toc185974272)

[Требования к программе 4](#_Toc185974273)

[1. Функциональные требования 4](#_Toc185974274)

[2. Надёжность 5](#_Toc185974275)

[3. Условия эксплуатации 5](#_Toc185974276)

[4. Совместимость 5](#_Toc185974277)

[Программная документация 5](#_Toc185974278)

[Стадии и этапы разработки 5](#_Toc185974279)

[Контроль и приёмка 5](#_Toc185974280)

[Проектирование ПК 6](#_Toc185974281)

[Описание вариантов использования 6](#_Toc185974282)

[Создание прототипа интерфейса пользователя 6](#_Toc185974283)

[Разработка объектной модели ПК 11](#_Toc185974284)

[Построение диаграммы программных классов 14](#_Toc185974285)

[Схема хранения данных 16](#_Toc185974286)

[Описание поведения ПК 17](#_Toc185974287)

[Построение диаграммы действий 18](#_Toc185974288)

[Руководство оператора 19](#_Toc185974289)

[Назначение программы 19](#_Toc185974290)

[Условия выполнения программы 20](#_Toc185974291)

[Описание задачи 20](#_Toc185974292)

[Входные и выходные данные 21](#_Toc185974293)

[Выполнение программы 21](#_Toc185974294)

[Заключение 31](#_Toc185974295)

[Приложение 32](#_Toc185974296)

# Техническое задание

## Введение

Программный комплекс (ПК) предназначен для управления данными о водителях, транспортных средствах и зарегистрированных нарушениях правил дорожного движения. Система обеспечивает ввод, редактирование, удаление данных, а также их сохранение и загрузку из базы данных *SQLite*.

## Основание для разработки

Основанием для разработки является курсовой проект по дисциплине «Объектно-ориентированное программирование». Проект разрабатывается с использованием языка *Java* и библиотеки *Swing* для создания графического интерфейса.

## Назначение разработки

ПК предназначен для использования в отделениях ГИБДД или других организациях, связанных с учётом нарушений. Система позволяет вести учёт водителей, автомобилей и зарегистрированных нарушений.

## Требования к программе

### Функциональные требования

Программа должна поддерживать:

* Добавление, редактирование и удаление записей о водителях, автомобилях и нарушениях.
* Ведение связей между таблицами (водители → машины, машины → нарушения).
* Проверку корректности вводимых данных (даты, VIN, номера ВУ и гос. номеров).
* Сохранение и загрузку данных в *SQLite*.

### Надёжность

Программа должна работать устойчиво при вводе некорректных данных, выводя сообщения об ошибках. Должен быть предусмотрен контроль уникальности номеров ВУ и гос. номеров.

### Условия эксплуатации

Программа рассчитана на использование в операционной системе *Windows*.

### Совместимость

Программа разрабатывается на языке *Java* с использованием *JDK 17* и библиотеки *Swing*. Поддерживается база данных *SQLite*.

## Программная документация

Программная документация должна соответствовать стандартам ЕСПД и включать:

* описание процесса проектирования ПК;
* руководство оператора;
* исходные тексты программы.

## Стадии и этапы разработки

1. Разработка технического задания.
2. Создание объектной модели данных и диаграммы классов.
3. Реализация программы с графическим интерфейсом.
4. Тестирование функционала.
5. Подготовка документации.

## Контроль и приёмка

Контроль качества программы включает проверку выполнения функциональных требований и надёжности обработки данных.

# Проектирование ПК

## Описание вариантов использования

ПК предназначен для работы с тремя основными объектами: водителями, транспортными средствами и нарушениями. Пользователь может выполнять следующие операции:

* *Управление водителями:* добавление, редактирование, удаление.
* *Управление автомобилями:* добавление, редактирование, удаление.
* *Управление нарушениями:* добавление, редактирование, удаление.
* *Работа с базой данных:* создание новой базы, подключение существующей базы.
* *Поиск данных* по ключевым параметрам, таким как ФИО водителя, номер ВУ, госномер машины и дата нарушения. А также поиск даты в диапазоне.

Диаграмма прецедентов представлена на рисунке 2.1.

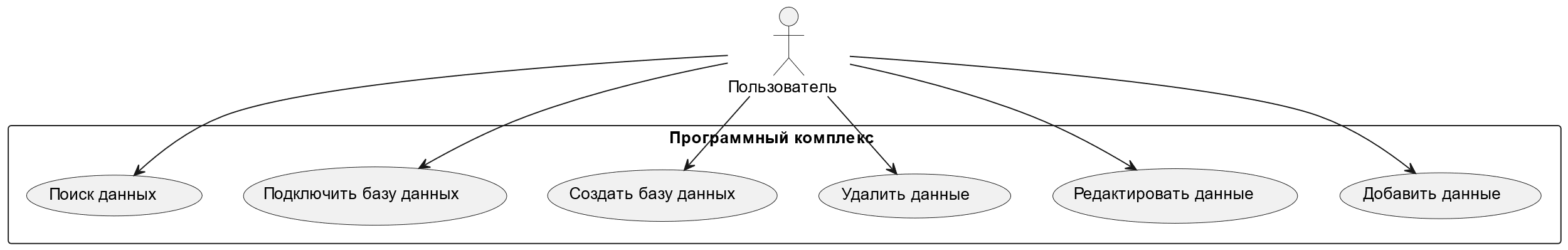


Рисунок 2.1 – Диаграмма прецендентов

## Создание прототипа интерфейса пользователя

* *Главное меню* с вкладками для управления данными (добавление, редактирование, удаление, изменение режима отображения).
* *Панель поиска* в нижней части окна, которая содержит:
* Поле ввода текста для критерия поиска.
* Выпадающий список для выбора параметра поиска.
* Кнопку «Поиск».
* *Таблицы данных* для отображения информации с возможностью прокрутки.
* *Диалоговые окна* для ввода и редактирования данных.

|  |  |  |  |
| --- | --- | --- | --- |
| **Экранная форма** | **Элементы управления** | **Действия пользователя** | **Отклик системы** |
| Добавить водителя | Поля: ФИО, пол, дата рождения, номер ВУ, срок действия ВУ. | Ввести данные, нажать «Добавить». | Добавление водителя в базу данных. |
| Редактировать водителя | Поля: ФИО, пол, дата рождения, срок действия ВУ. | Изменить данные, нажать «Сохранить». | Сохранение изменений в базе данных. |
| Удалить водителя | Таблица с записями. | Выбрать запись, нажать «Удалить». | Удаление водителя из базы данных. |
| Добавить автомобиль | Поля: номерной знак, номер ВУ владельца, VIN, модель, цвет, даты. | Ввести данные, нажать «Добавить». | Добавление автомобиля в базу данных. |
| Редактировать автомобиль | Поля: номерной знак, модель, цвет, даты. | Изменить данные, нажать «Сохранить». | Сохранение изменений в базе данных. |
| Удалить автомобиль | Таблица с записями. | Выбрать запись, нажать «Удалить». | Удаление автомобиля из базы данных. |
| Добавить нарушение | Поля: номер машины, дата, тип нарушения. | Ввести данные, нажать «Добавить». | Добавление записи о нарушении в базу данных. |
| Редактировать нарушение | Поля: номер машины, дата, тип нарушения. | Изменить данные, нажать «Сохранить». | Сохранение изменений в базе данных. |
| Удалить нарушение | Таблица с записями. | Выбрать запись, нажать «Удалить». | Удаление записи из базы данных. |
| Подключить базу данных | Диалог выбора файла. | Выбрать файл, нажать «Открыть». | Подключение к существующей базе данных. |
| Поиск данных | Поле поиска, выпадающий список, кнопка «Поиск». | Ввести критерий поиска, выбрать параметр, нажать «Поиск». | Отображение результатов поиска в таблице. |

Таблица 2.1 – Экранные формы

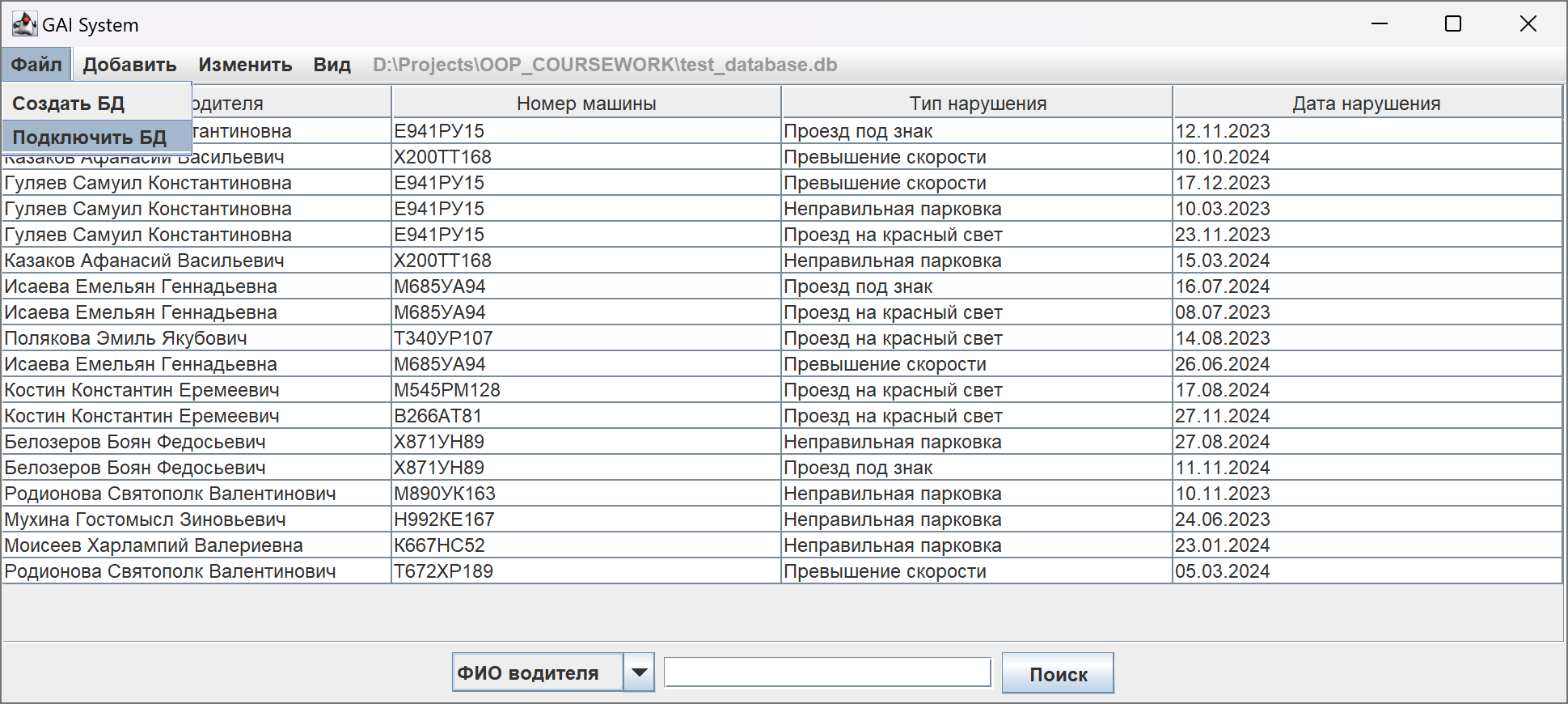


Рисунок 2.2 – Окно основной программы в режиме нарушений

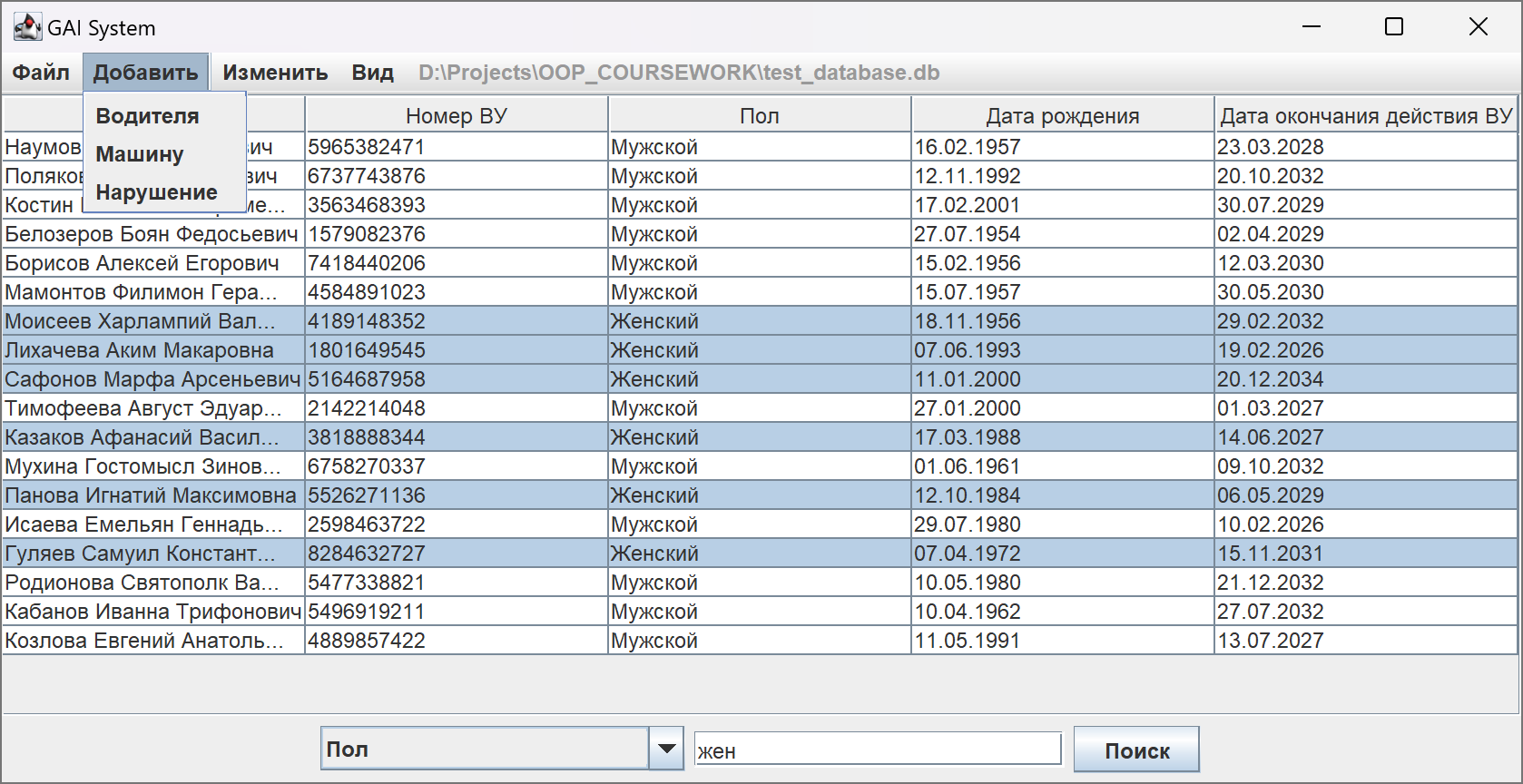


Рисунок 2.3 – Окно основной программы в режиме водителей и поиском по параметру

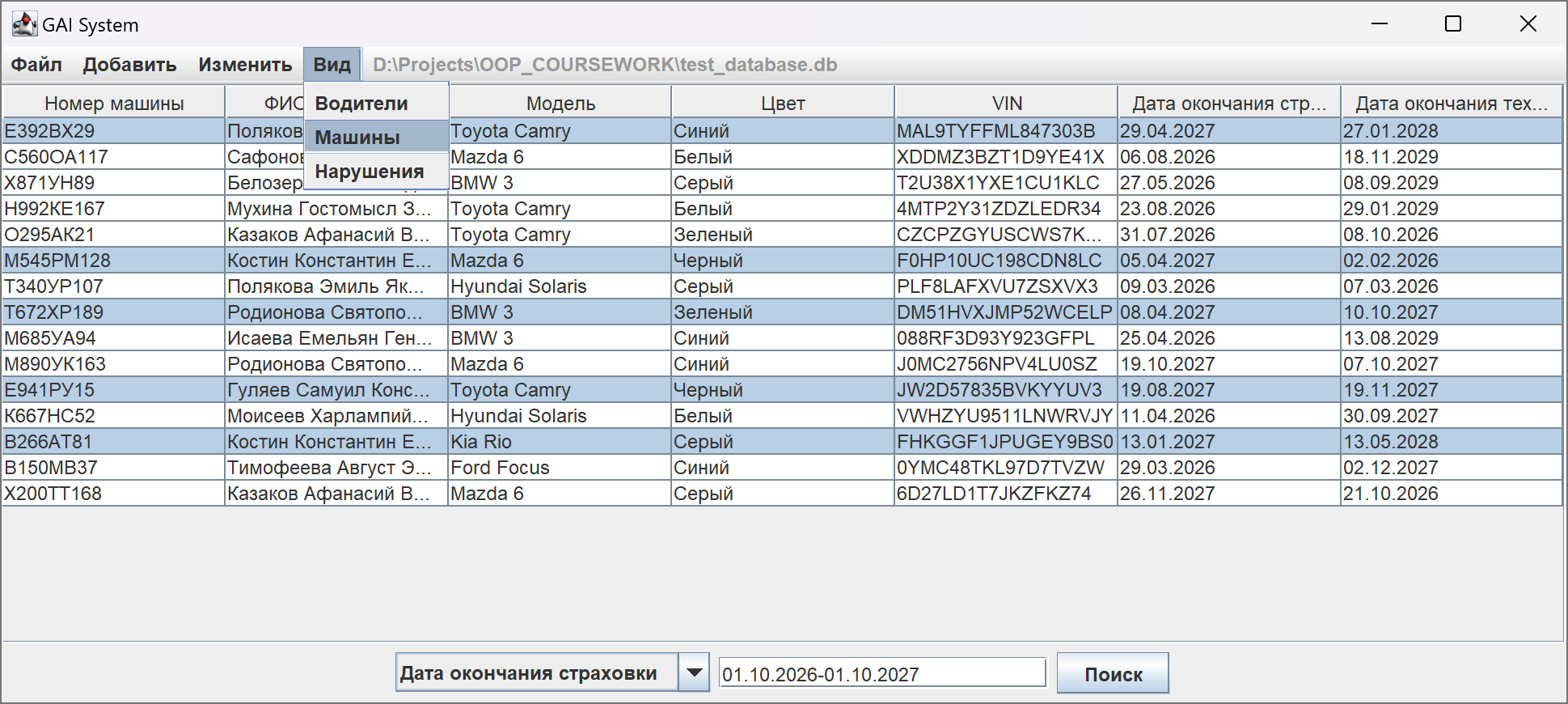


Рисунок 2.4 – Окно основной программы в режиме машин с поиском в диапазоне дат

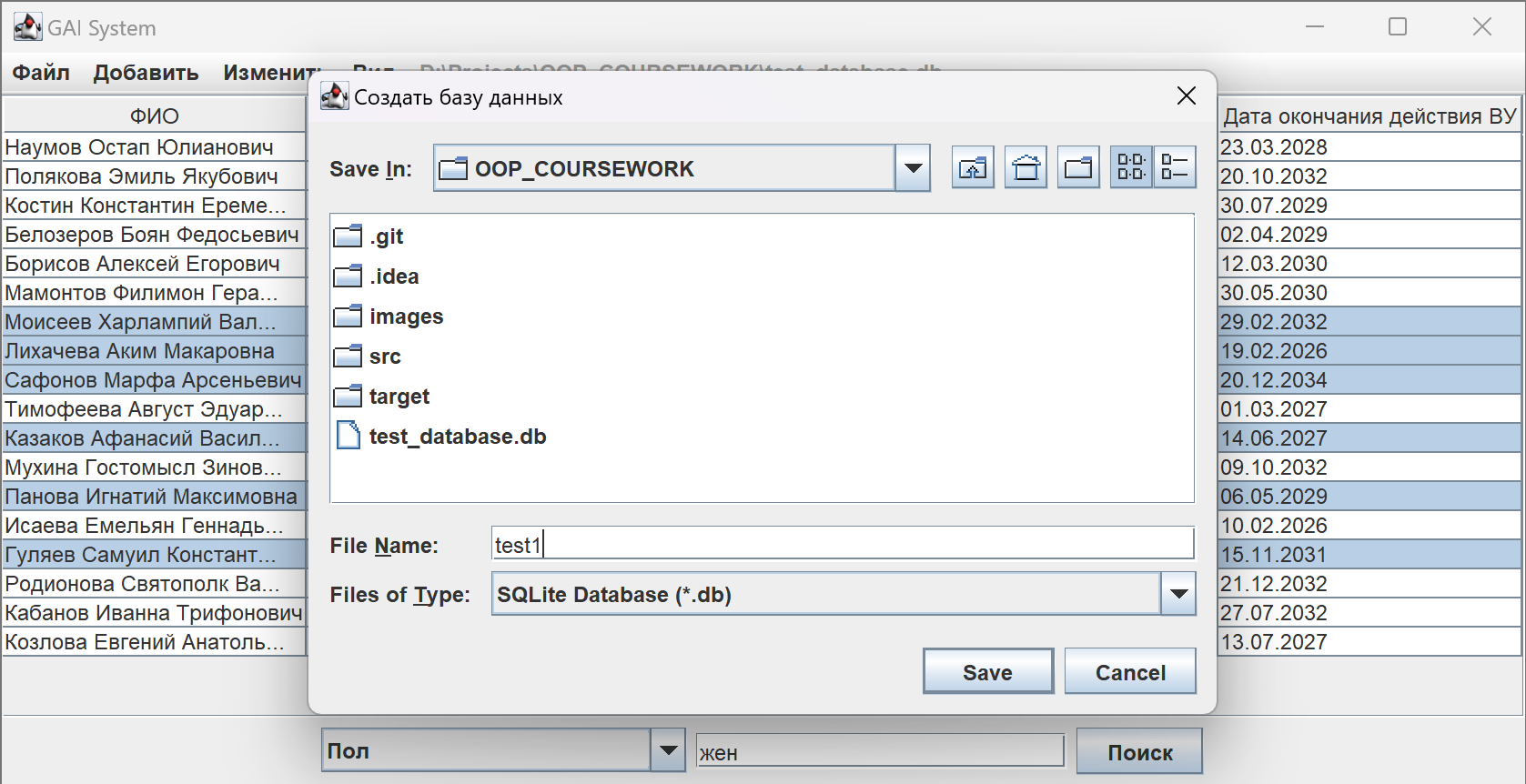


Рисунок 2.5 – Диалоговое окно создания БД

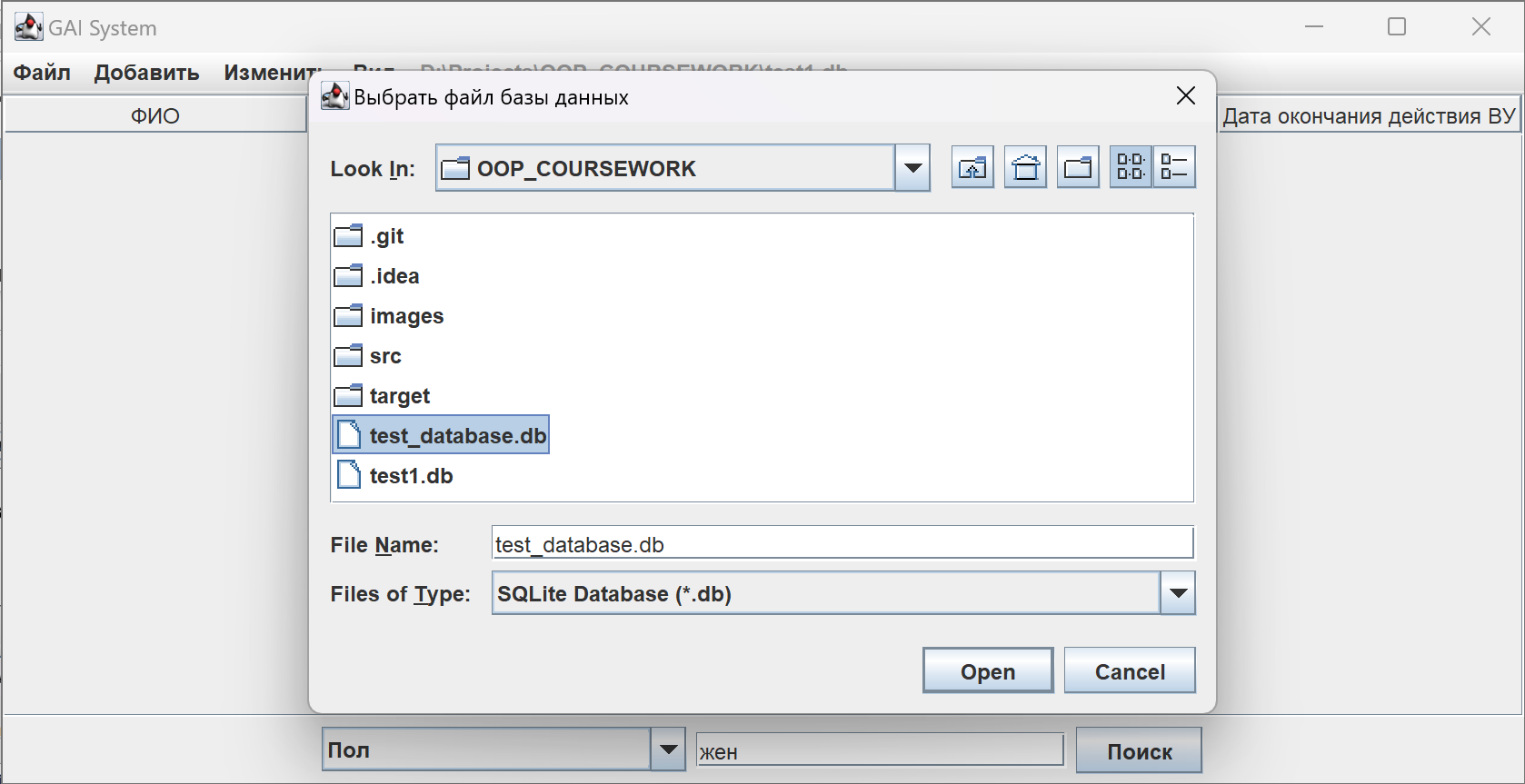


Рисунок 2.6 – Диалоговое окно подключения БД

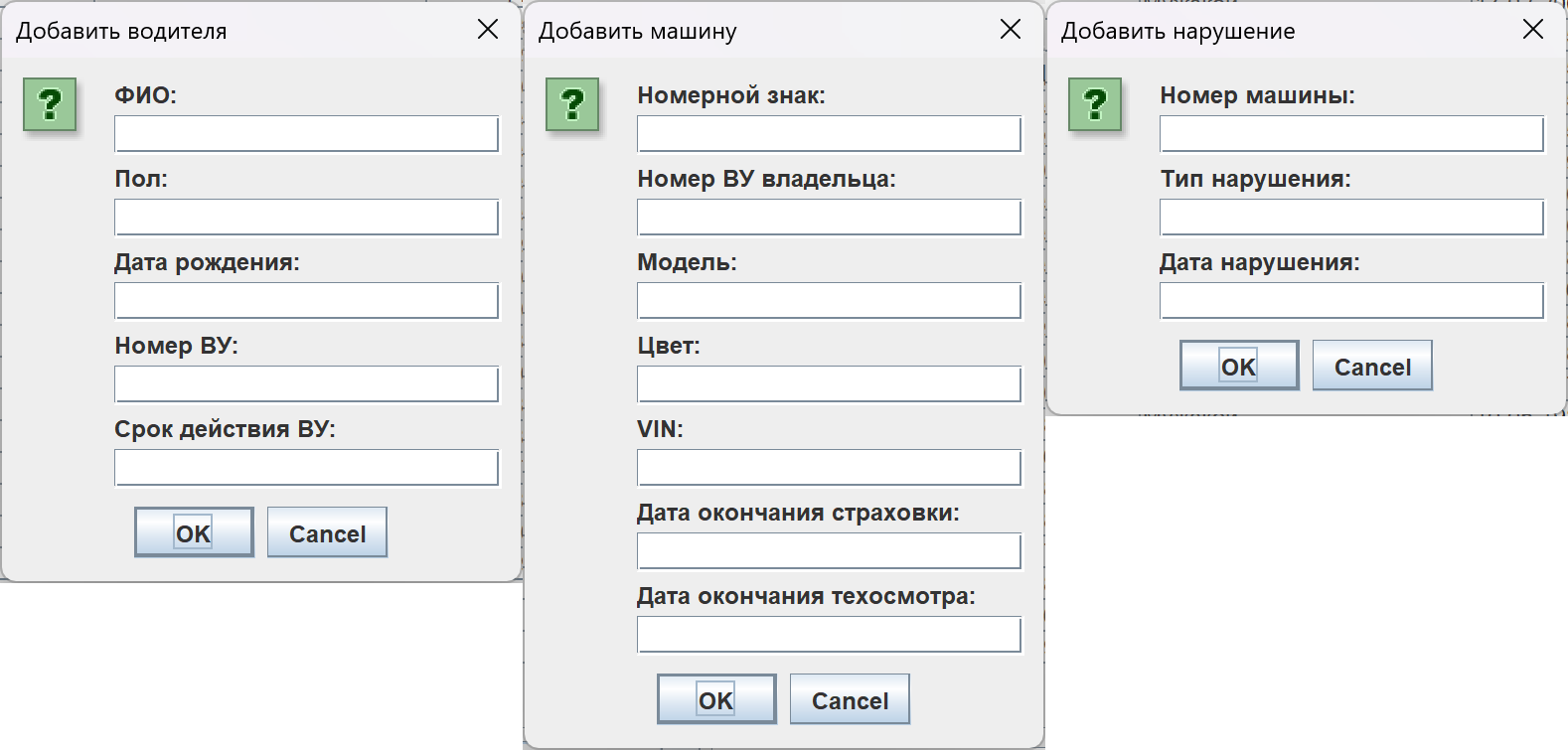


Рисунок 2.7 – Диалоговые окна добавления данных

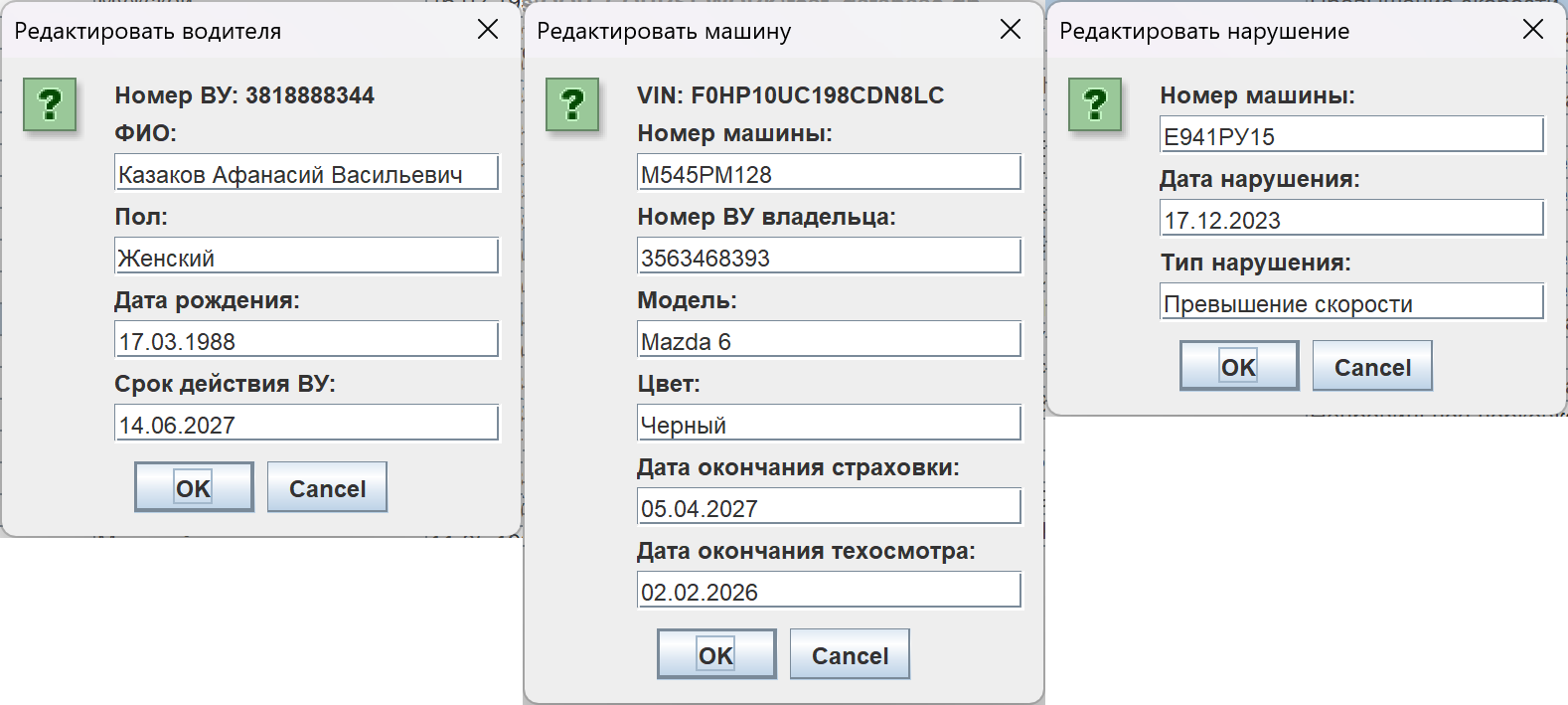


Рисунок 2.8 – Диалоговые окна редактирования данных

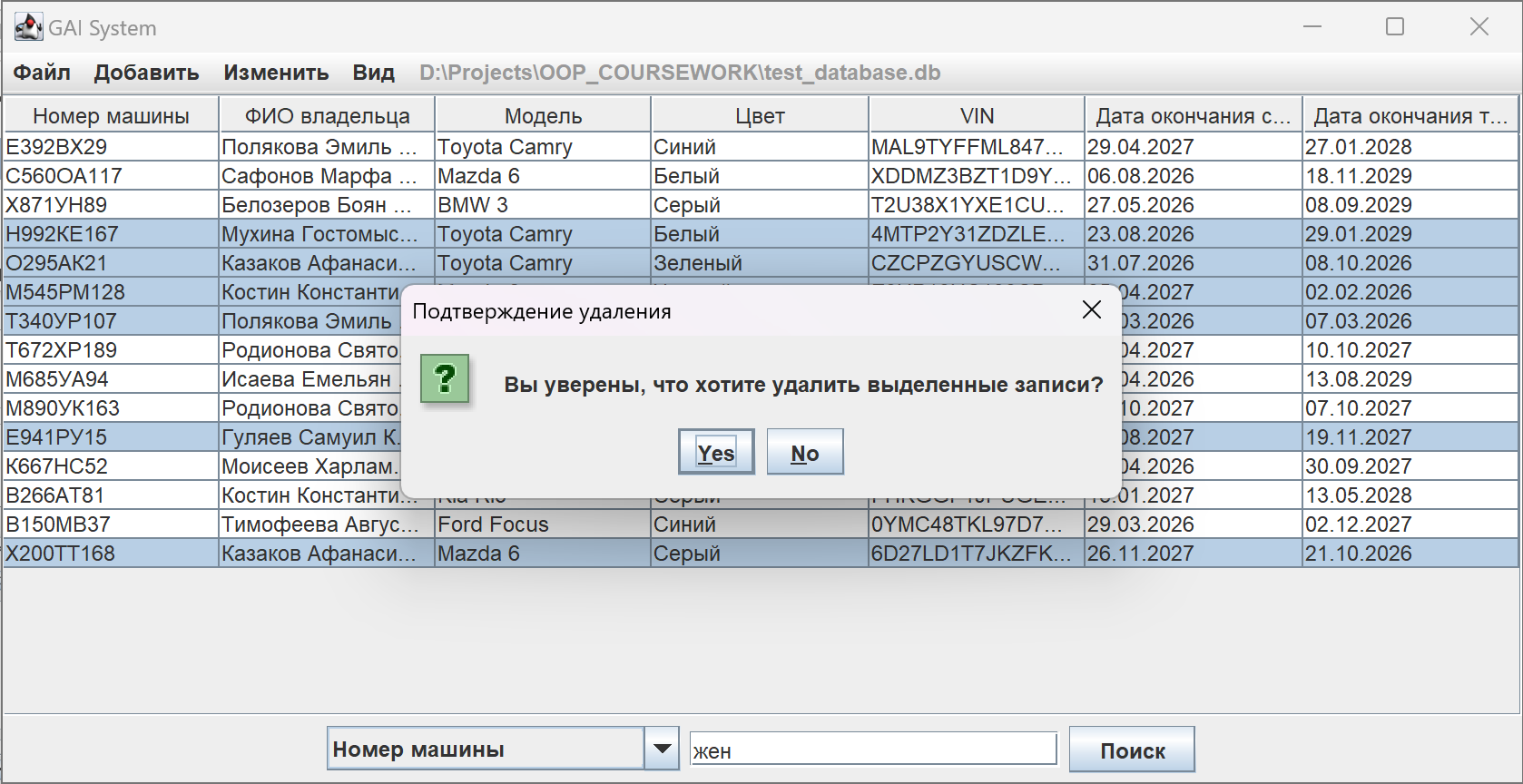


Рисунок 2.9. – Диалоговое окно подтверждения удаления

## Разработка объектной модели ПК

Объектная модель программного комплекса включает три основные сущности: Водитель, Автомобиль и Нарушение, а также пользователя системы, который управляет данными. Эти сущности связаны между собой логическими отношениями, описанными в диаграмме сущностей (Рис. 2.10).

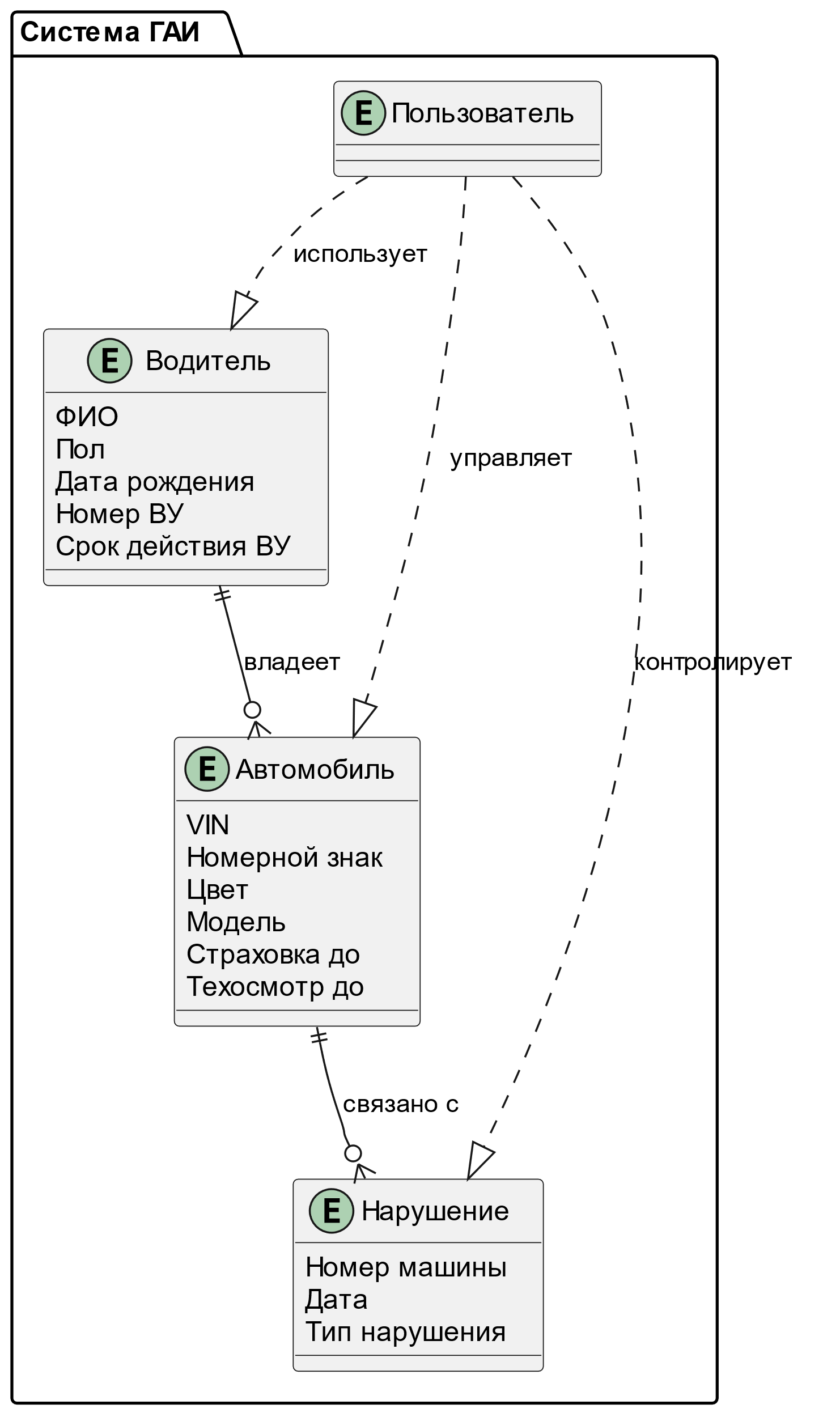


Рисунок 2.10 – Диаграмма сущностей

Детальное описание операций представлено в таблице 2.2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Сущность** | **Имя операции** | **Параметры операции** | | | **Тип возвращ. значения** | **Назначение операции** |
| **Вид** | **Название** | **Тип** |
| Водитель | Добавить | Вх. | ФИО | Строка | Логический | Добавляет нового водителя в базу данных с проверкой корректности данных. |
| Вх. | Пол | Строка |
| Вх. | Дата рождения | Дата |
| Вх. | Номер ВУ | Строка |
| Вх. | Срок ВУ | Дата |
| Редактировать | Вх. | Номер ВУ | Строка | Логический | Изменяет данные существующего водителя в базе данных. |
| Вх. | Новые данные | ФИО, пол, дата рождения, срок ВУ |
| Удалить | Вх. | Номер ВУ | Строка | Логический | Удаляет водителя и связанные с ним автомобили из базы данных. |
| Автомобиль | Добавить | Вх. | VIN | Срока | Логический | Добавляет новый автомобиль с привязкой к водителю и проверкой корректности VIN и номера. |
| Вх. | Номерной знак | Строка |
| Вх. | Цвет | Строка |
| Вх. | Модель | Строка |
| Вх. | Срок страховки | Дата |
| Вх. | Срок ТО | Дата |
| Редактировать | Вх. | Номерной знак | Строка | Логический | Изменяет данные существующего автомобиля в базе данных. |
| Вх. | Новые данные | VIN, цвет, модель, даты страховки и ТО |
| Удалить | Вх. | Номерной знак | Строка | Логический | Удаляет автомобиль и связанные с ним нарушения из базы данных. |
| Нарушение | Добавить | Вх. | Номер машины | Строк | Логический | Добавляет новое нарушение с проверкой существования машины в базе данных. |
| Вх. | Дата наруш. | Тип |
| Вх. | Тип наруш. | Строка |
| Редактировать | Вх. | Новые данные | (Номер маш. дата, тип наруш.) | Логический | Изменяет данные существующего нарушения в базе данных. |
| Удалить | Вх. | Номер машины | Строка | Логический | Удаляет запись о нарушении из базы данных. |
| Поиск | Искать по параметру | Вх. | Параметр поиска | Строка | Список записей | Выполняет поиск записей по указанному параметру. |
| Вх. | Значение поиска | Строка |
| База данных | Подключить | Вх. | Файл БД | Строка | Логический | Подключает указанную базу данных *SQLite* к программе. |

Таблица 2.2 – Описание операций

## Построение диаграммы программных классов

Диаграмма программных классов отражает структуру программы и связи между основными сущностями системы. В ней представлены классы, их атрибуты, методы, а также отношения наследования и ассоциации между классами. Диаграмма программных классов ПК представлена на рисунке 2.11.

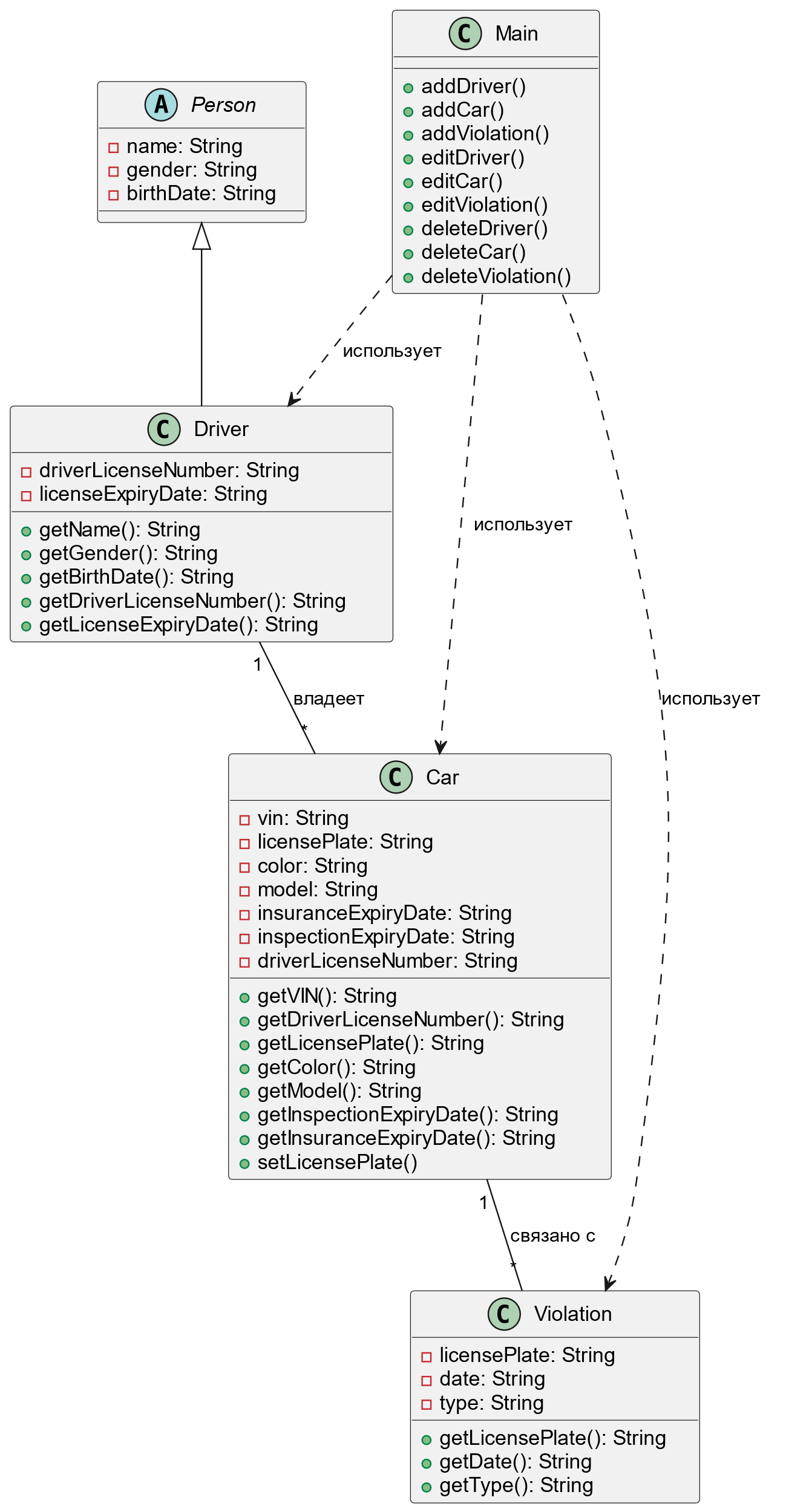


Рисунок 2.11 – Диаграмма классов

## Схема хранения данных

Данные в программном комплексе хранятся в базе данных *SQLite*. База данных включает три основные таблицы: *Drivers*, *Cars* и *Violations*. Каждая таблица содержит информацию, необходимую для управления записями о водителях, автомобилях и нарушениях.

1. Таблица *Drivers*. Содержит информацию о водителях, включая:

* Уникальный номер водительского удостоверения (PK).
* Личные данные (ФИО, пол, дата рождения).
* Срок действия водительского удостоверения.

1. Таблица *Cars*. Содержит информацию об автомобилях, включая:

* Уникальный номерной знак (PK).
* Ссылку на владельца (FK, номер ВУ).
* VIN, цвет, модель и даты действия страховки и техосмотра.

1. Таблица *Violations*. Содержит информацию о нарушениях, включая:

* Уникальный идентификатор нарушения (PK).
* Ссылку на автомобиль (FK, номерной знак).
* Дату и тип нарушения.

*Связи между таблицами:*

* Один водитель может владеть несколькими автомобилями (*Drivers* → *Cars*).
* Один автомобиль может быть связан с несколькими нарушениями (*Cars* → *Violations*).

Схема организации БД приведена на рис. 2.12.

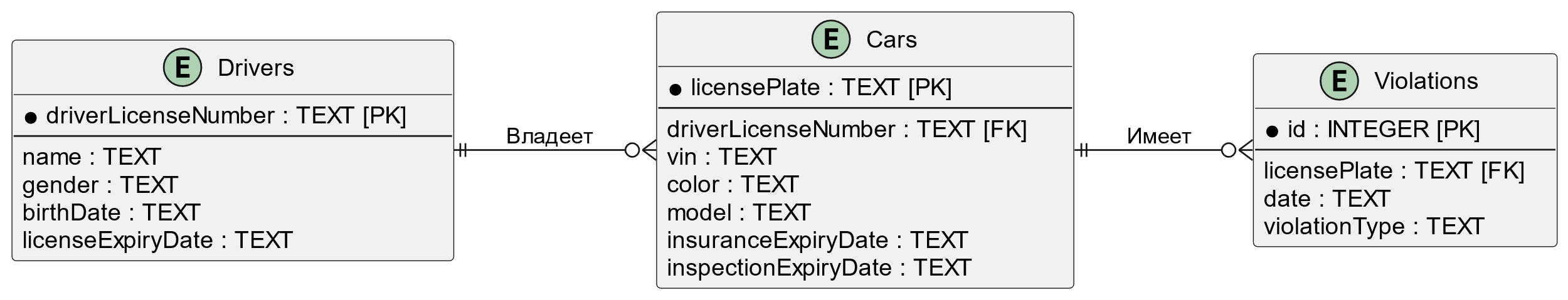


Рисунок 2.12 – Диаграмма БД

## Описание поведения ПК

Программный комплекс использует базу данных *SQLite* для хранения информации о водителях, автомобилях и нарушениях. Все изменения данных (добавление, редактирование и удаление) вносятся напрямую в подключённую базу данных.

*Основные принципы взаимодействия:*

1. Пользователь инициирует выполнение операций через графический интерфейс.
2. Программа вызывает соответствующие методы обработки данных, проверяя корректность введённых значений.
3. После успешной валидации данные передаются в базу данных для выполнения SQL-запросов.
4. База данных подтверждает выполнение операции или сообщает об ошибке.
5. Программа обновляет отображение данных и информирует пользователя о результате выполнения.

На рисунке 2.13 представлена диаграмма последовательностей, иллюстрирующие шаги выполнения операций добавления данных в базу.

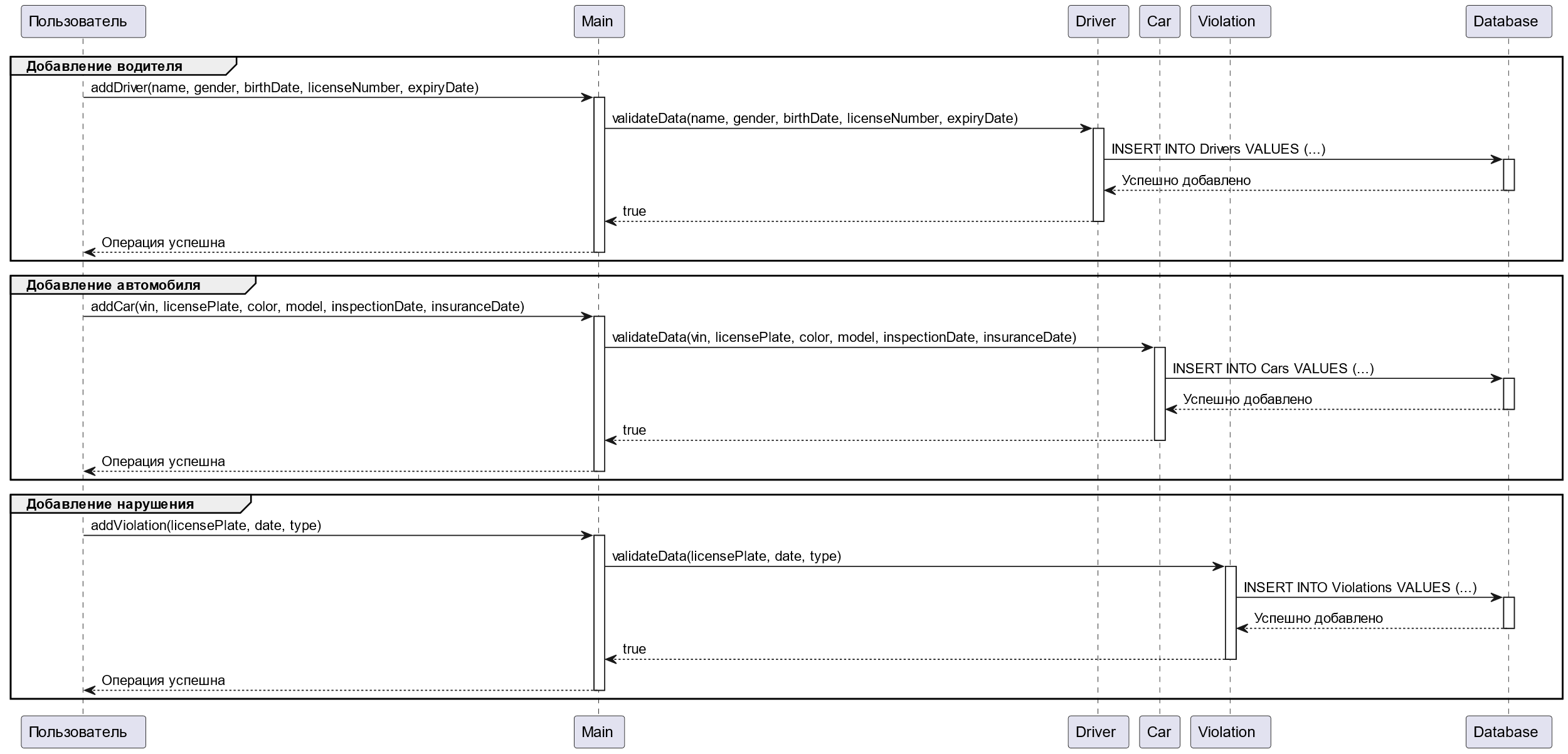


Рисунок 2.13 – Диаграмма последовательностей для операции в БД

## Построение диаграммы действий

Диаграмма действий отражает последовательность шагов, выполняемых в процессе добавления нового водителя в базу данных. Она описывает ключевые этапы выполнения операции, включая проверки корректности данных и обработку ошибок.

***Описание процесса:*** Процесс начинается с открытия окна программы и выбора вкладки «Водители». Пользователь нажимает кнопку «Добавить» и вводит данные нового водителя. Затем выполняется проверка корректности введённых данных. Если данные верны, запись сохраняется в базе данных и отображается в таблице. В случае ошибки выводится сообщение, и пользователь может исправить данные.

Диаграмма действий представлена на рисунке 2.14.

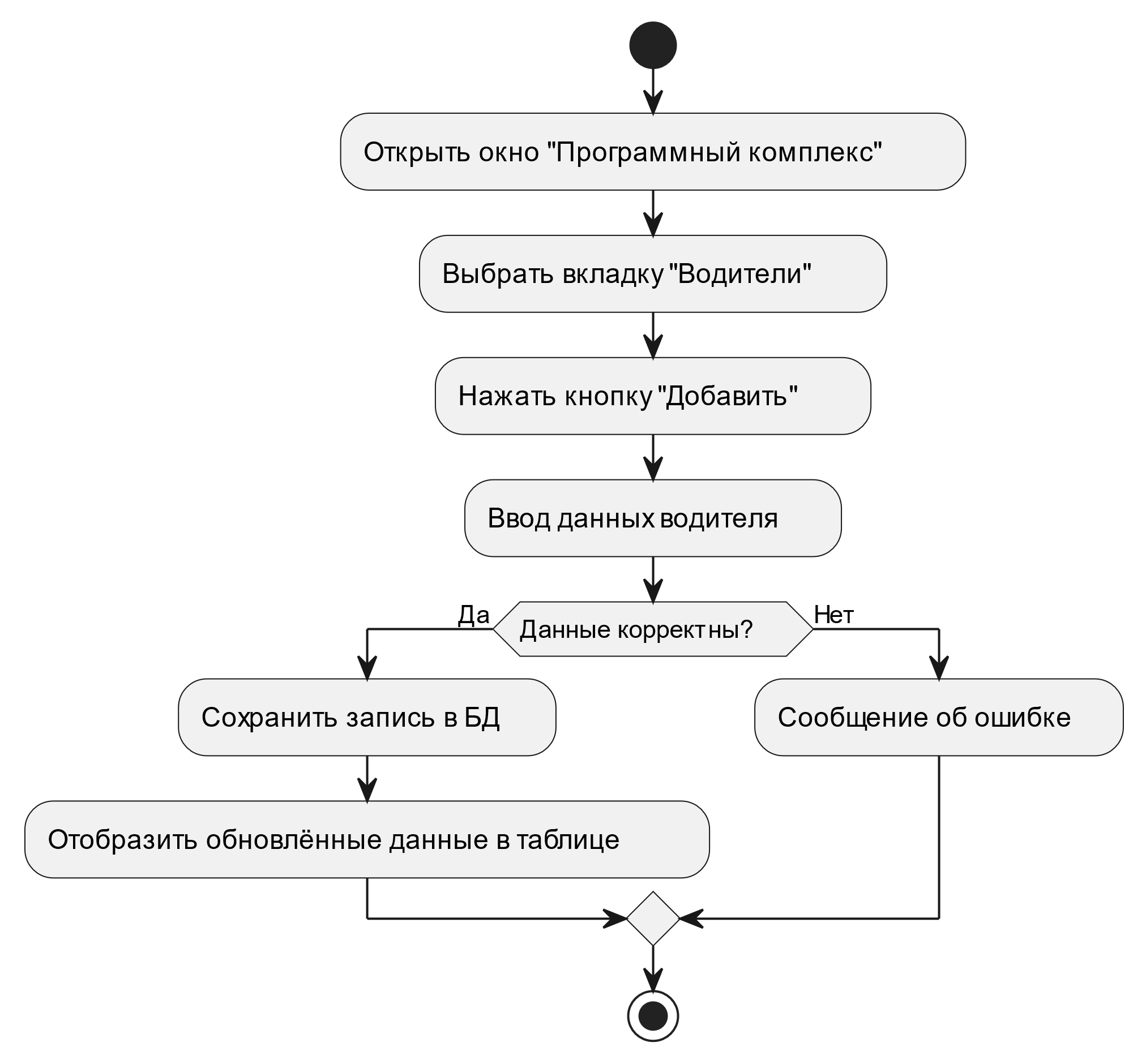


Рисунок 2.14 – Диаграмма действий

# Руководство оператора

## Назначение программы

Программный комплекс предназначен для автоматизации работы с данными о водителях, автомобилях и нарушениях правил дорожного движения. Основная цель программы — обеспечить удобный и эффективный инструмент для хранения, обработки и анализа информации, связанной с дорожным движением.

*Основные задачи программы:*

1. *Управление данными о водителях:* Добавление, редактирование и удаление информации о водителях.
2. *Управление данными об автомобилях:* Ведение учёта транспортных средств, их характеристик и статусов.
3. *Учёт нарушений:* Регистрация и контроль нарушений, связанных с транспортными средствами.
4. *Поиск и фильтрация данных:* Быстрый доступ к необходимой информации по заданным критериям.
5. *Контроль связей между сущностями:* Проверка корректности данных и связей между таблицами базы данных.

*Целевая аудитория программы:*

* Сотрудники органов дорожного контроля и администраторы баз данных.
* Специалисты, отвечающие за учёт и контроль правонарушений.
* Операторы информационных систем, работающие с базами данных транспортных средств.

*Основные преимущества программы:*

* Простой и интуитивно понятный графический интерфейс.
* Автоматическая проверка данных перед добавлением в базу.
* Поддержка работы с *SQLite* для удобного хранения информации.
* Логические связи между таблицами, обеспечивающие целостность данных.
* Возможность быстрого редактирования и поиска информации.

## Условия выполнения программы

Программа разработана для работы в операционной системе *Windows* с установленной поддержкой *Java* и некоторыми дополнительными библиотеками. Она предназначена для эффективного управления данными в базе данных *SQLite* и обеспечивает выполнение основных операций, включая добавление, редактирование и удаление записей.

Минимальные системные требования:

* *Операционная система:* Windows 7 и выше.
* *Процессор:* С тактовой частотой от 1.8 ГГц.
* *Оперативная память*: Не менее 2 Гб.
* *Жёсткий диск:* Не менее 500 Мб свободного пространства.
* *Программное обеспечение:* Java

## Описание задачи

Программа предназначена для ведения учёта данных о водителях, автомобилях и нарушениях правил дорожного движения. Основное назначение — автоматизация процессов добавления, редактирования и удаления записей, а также упрощение поиска и анализа информации в базе данных.

*Основные сущности, с которыми работает программа:*

* *Водитель:* информация о водителе (ФИО, пол, дата рождения, номер ВУ, срок действия ВУ).
* *Автомобиль:* данные об автомобиле (VIN, номерной знак, цвет, модель, страховка, техосмотр).
* *Нарушение:* записи о нарушениях (номер машины, дата, тип нарушения).

*Возможности программы:*

1. Создание записей о водителях, автомобилях и нарушениях.
2. Редактирование данных с проверкой корректности и уникальности значений.
3. Удаление записей с учётом связей между сущностями.
4. Поиск и фильтрация данных по ключевым параметрам.
5. Проверка целостности данных и обработка ошибок при работе с базой данных.

## Входные и выходные данные

*Входные данные:*

* *Водитель:* ФИО, пол, дата рождения, номер водительского удостоверения, срок действия ВУ.
* *Автомобиль:* VIN, номерной знак, цвет, модель, срок действия страховки и техосмотра.
* *Нарушение:* номер машины, дата нарушения, тип нарушения.

*Выходные данные:*

* Список водителей с указанием личных данных и сроков действия ВУ.
* Список автомобилей с характеристиками и привязкой к владельцу.
* Список нарушений с привязкой к автомобилю и указанием типа нарушения.
* Отчёты по количеству водителей, автомобилей и зарегистрированных нарушений.

Программа обрабатывает входные данные, проверяя их корректность, и предоставляет отчёты в табличном формате для анализа и контроля.

## Выполнение программы

При запуске программы на экране появится главное окно (Рис. 3.1). В правом верхнем углу можно увидеть основные кнопки программы. При нажатии на них откроется контекстное меню с выбором действий (Рис 3.2). Если не подключена база данных (сообщение о состоянии подключения отображается справа от всех кнопок меню), активными остаются только кнопки меню «Файл» и «Вид».

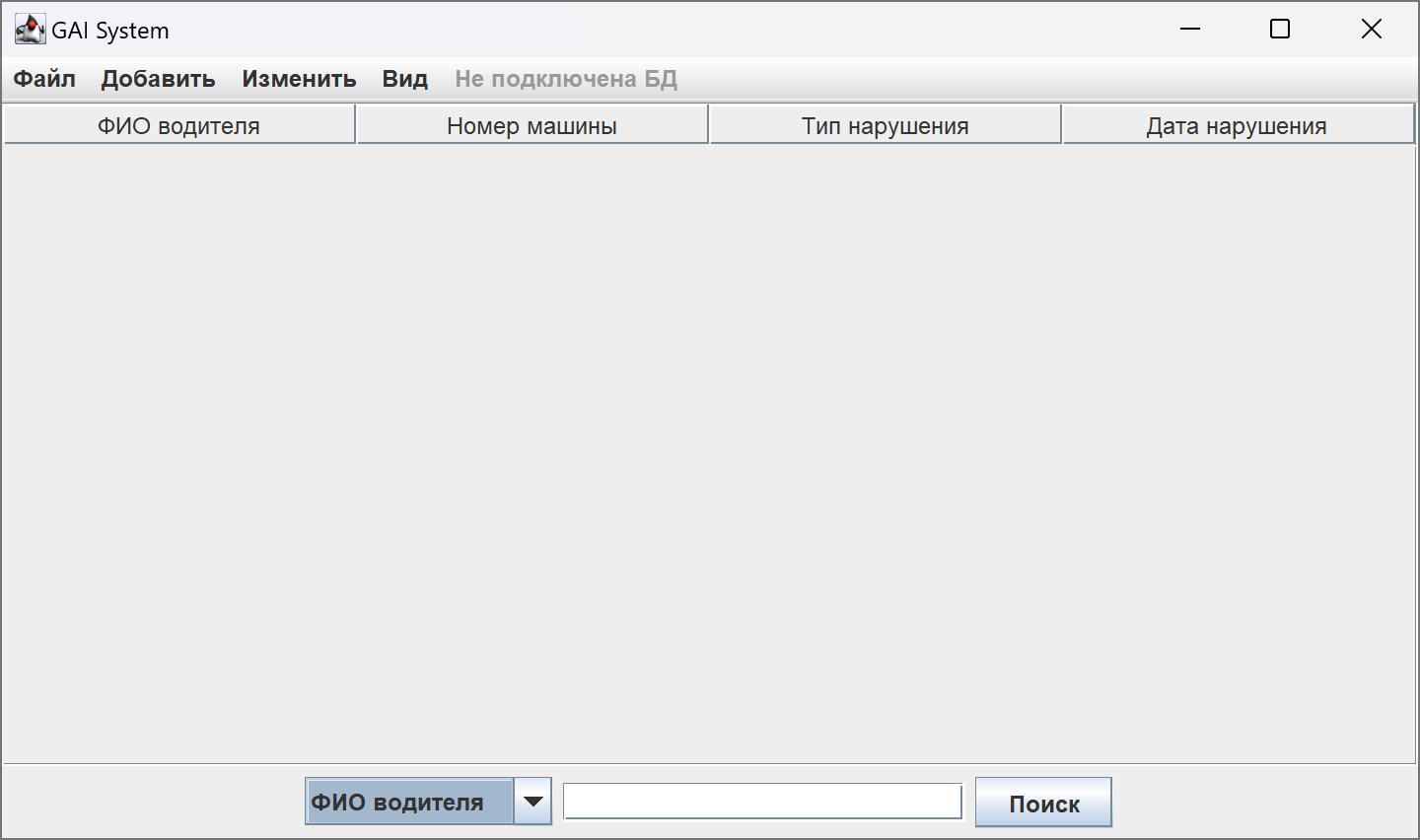


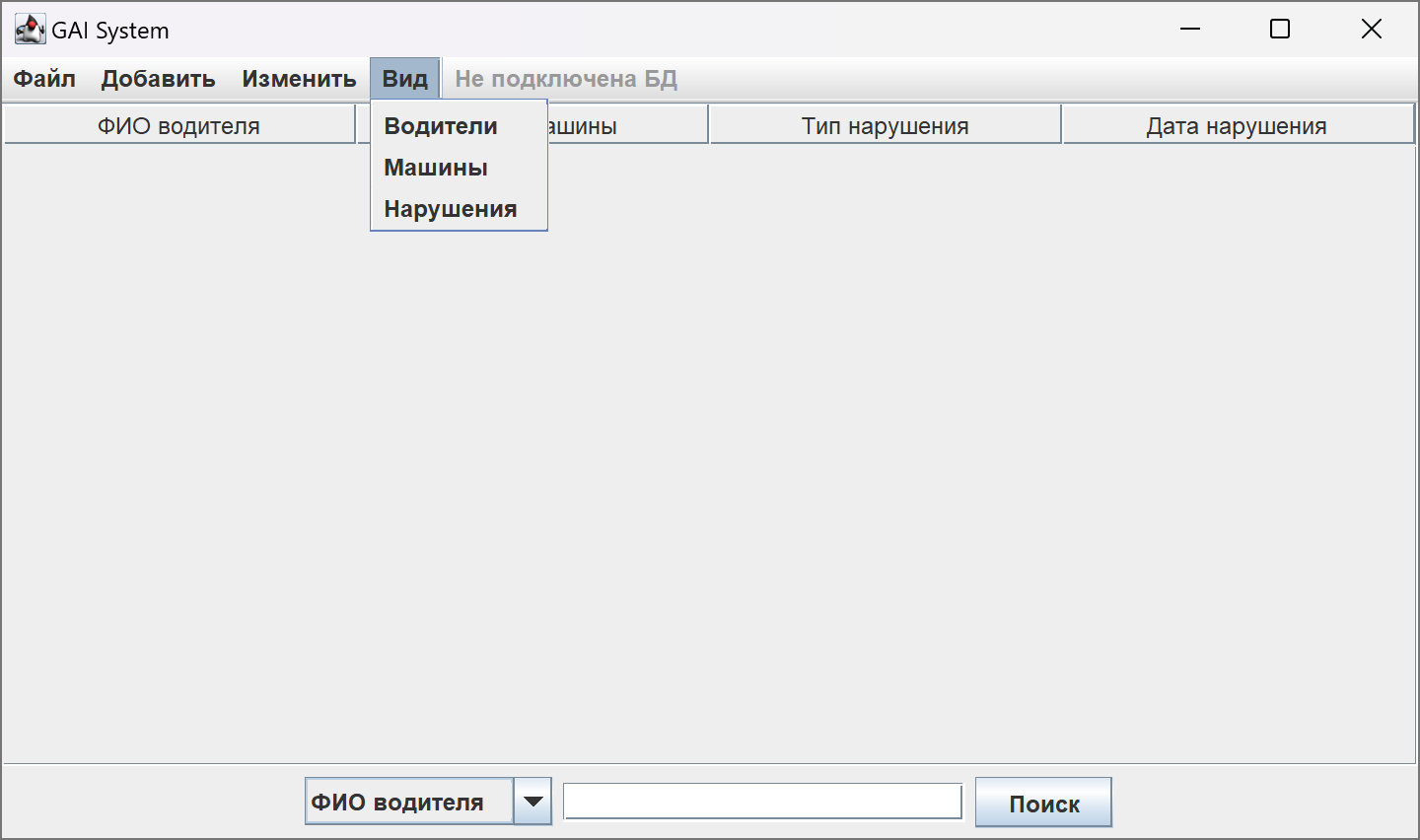
Рисунок 3.1 – Главное окно ПК  


Рисунок 3.2 – Отображение контекстного меню

Для работы с данными нужно создать новую или подключиться к существующей БД. Для этого нужно выбрать одно из действий: «Файл» → «Создать БД» или «Файл» → «Подключить БД» (Рис. 3.3). В открывшимся диалоговом окне можно выбрать расположение БД (Рис. 3.4).

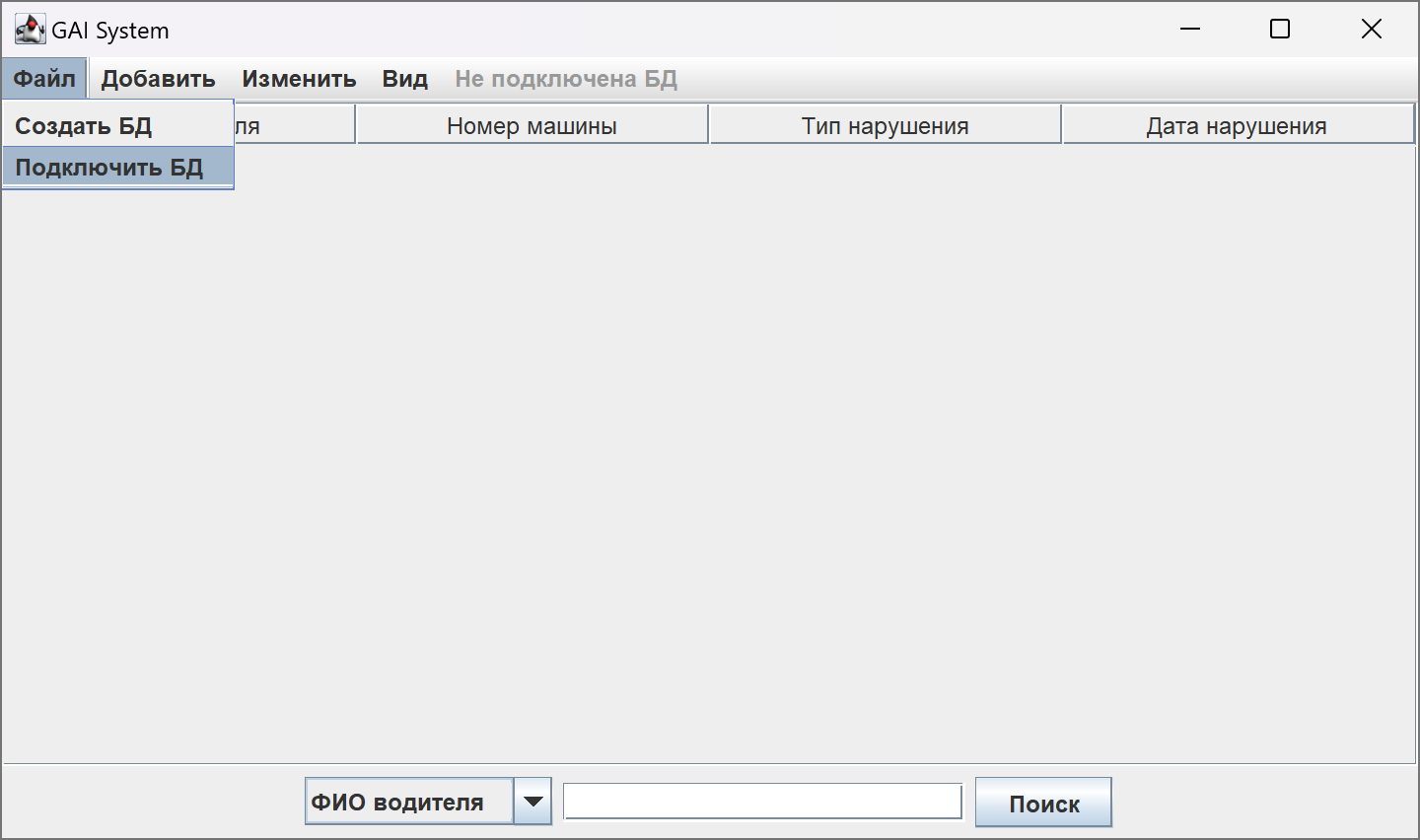


Рисунок 3.3 – Выбор режима подключения БД

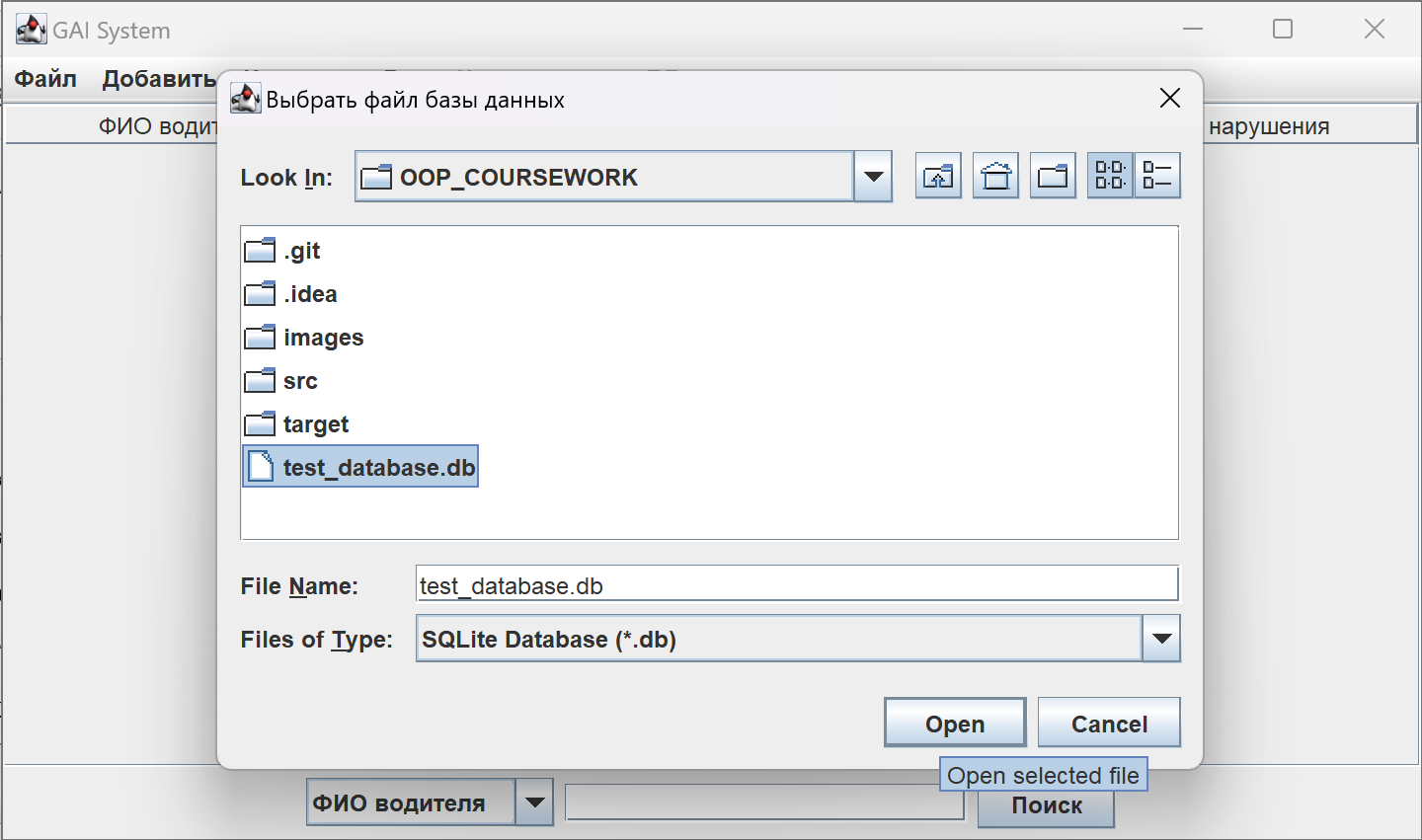


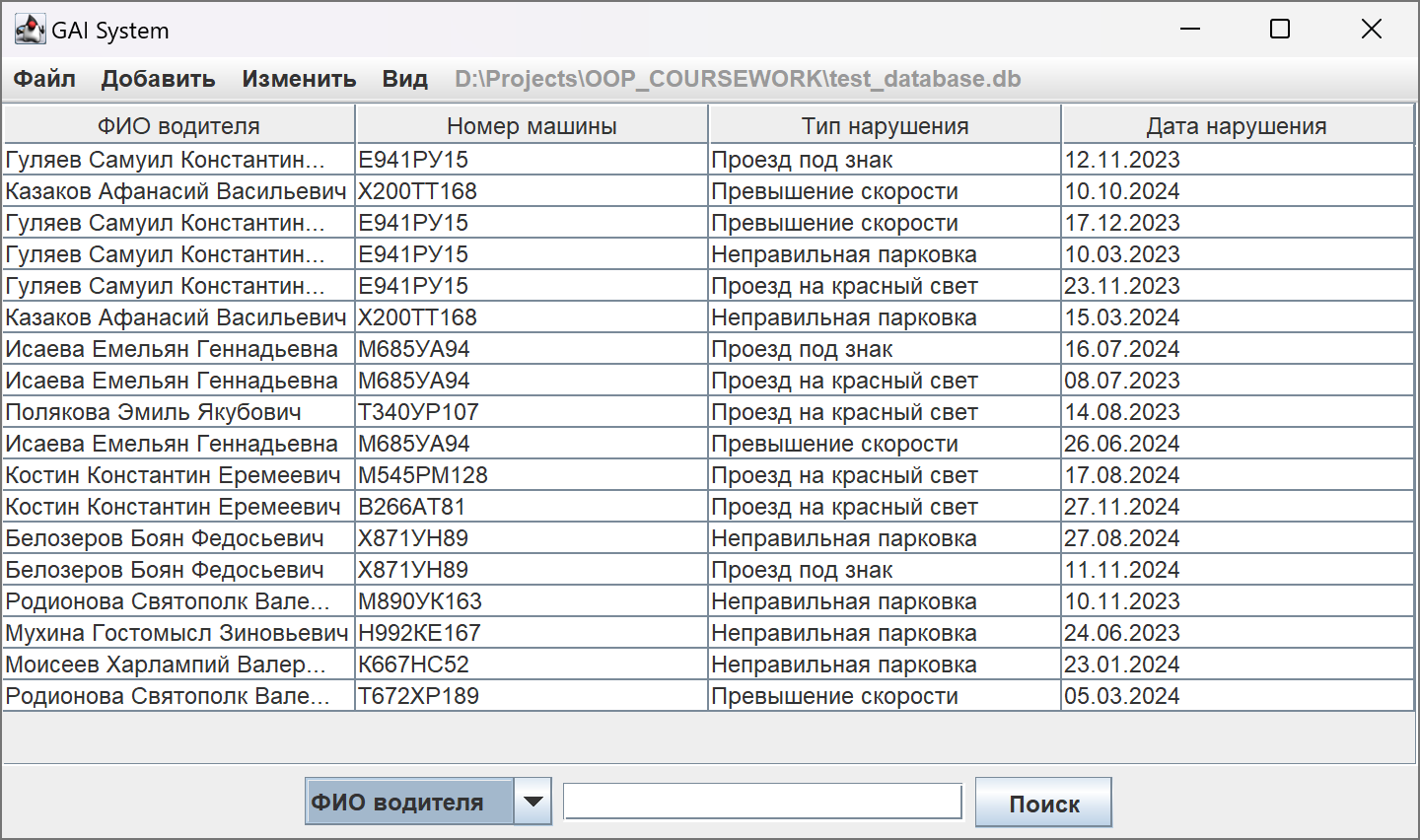
Рисунок 3.4 – Выбор файла БД

После подключения или создания БД данные будут отображаться в главном окне в виде таблицы (Рис. 3.5). В меню будет отображаться путь к файлу, который сейчас подключён. Доступно 3 режима отображения данных:

«Вид» → «Водители» – отображает данные о водителях (Рис. 3.6)

«Вид» → «Машины» – отображает данные о машинах (Рис 3.7)

«Вид» → «Нарушения» – отображает данные о нарушениях (Рис 3.5)

  
Рисунок 3.5 – Режим отображения нарушений

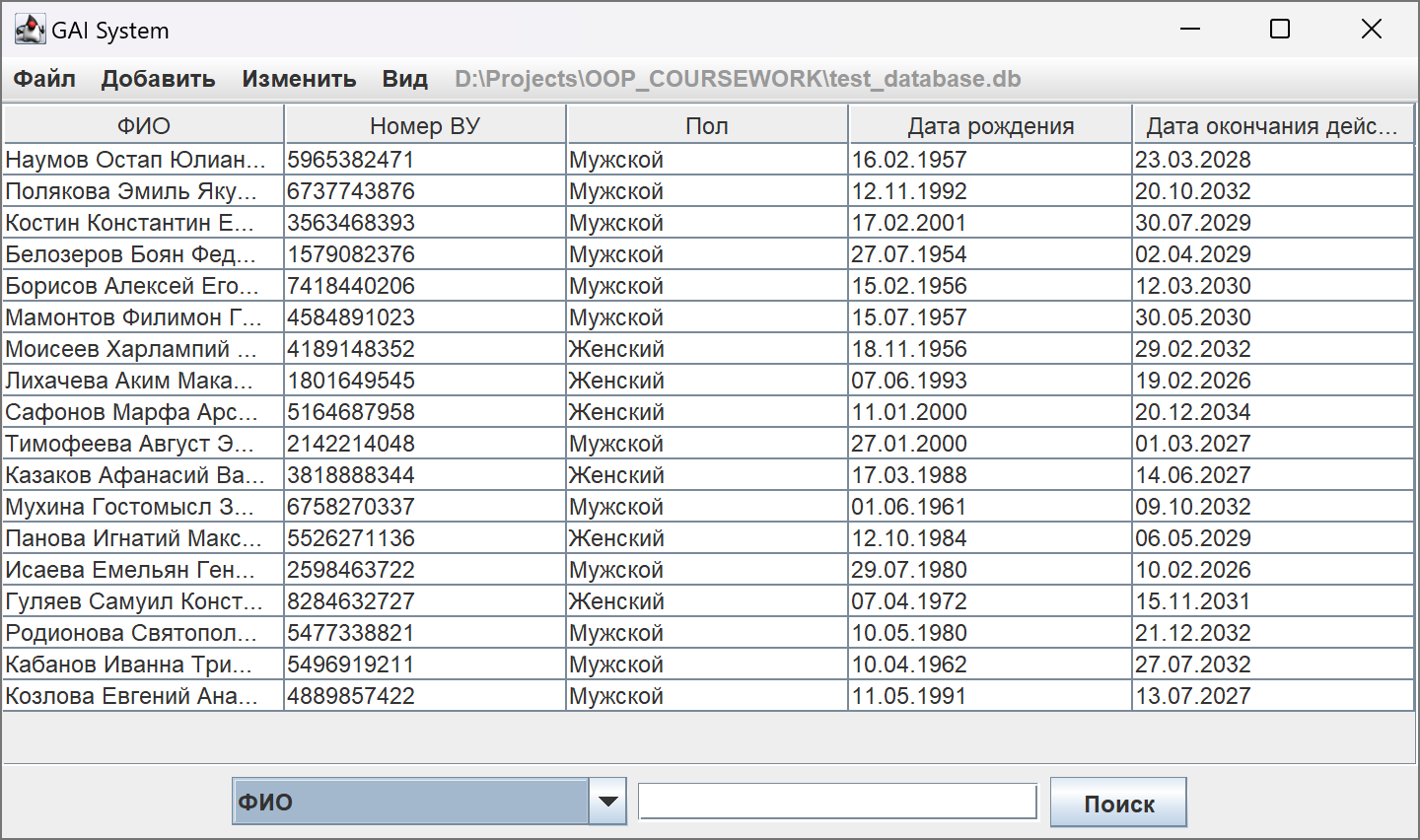


Рисунок 3.6 – Режим отображения водителей

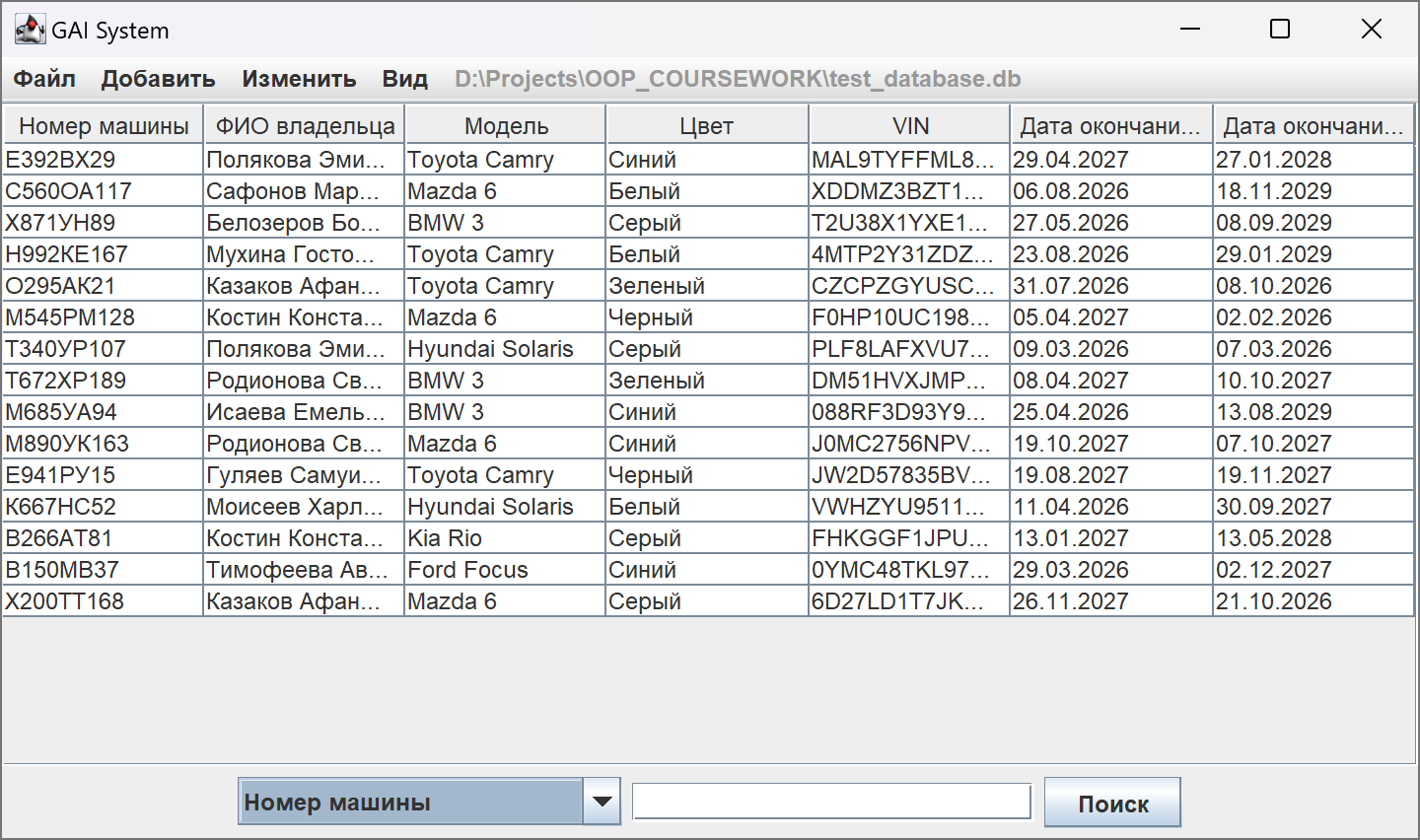


Рисунок 3.7 – Режим отображения машин

Теперь, когда БД подключена мы можем добавлять данные. Разберём на примере добавления машины («Добавить» → «Машину»). Отроется диалоговое окно (Рис. 3.8), в котором нужно ввести все данные. После ввода всех данных нужно нажать кнопку «ОК», программа проверит все данные и добавит их в БД. В случае попытки сохранения некорректных данных программа выдаст предупреждение и укажет, где вероятнее всего допущена ошибка (Рис. 3.9).

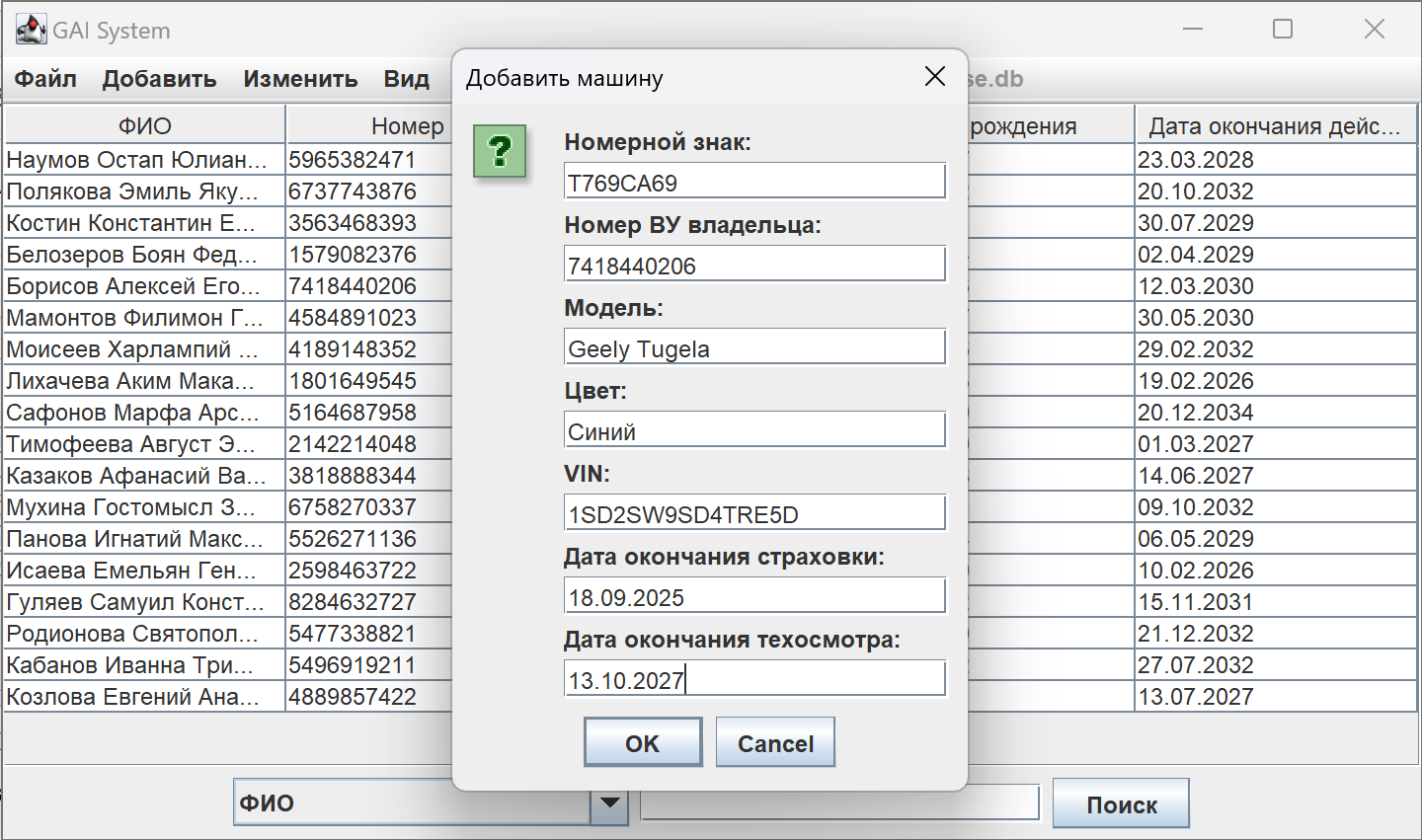


Рисунок 3.8 – Добавление машины

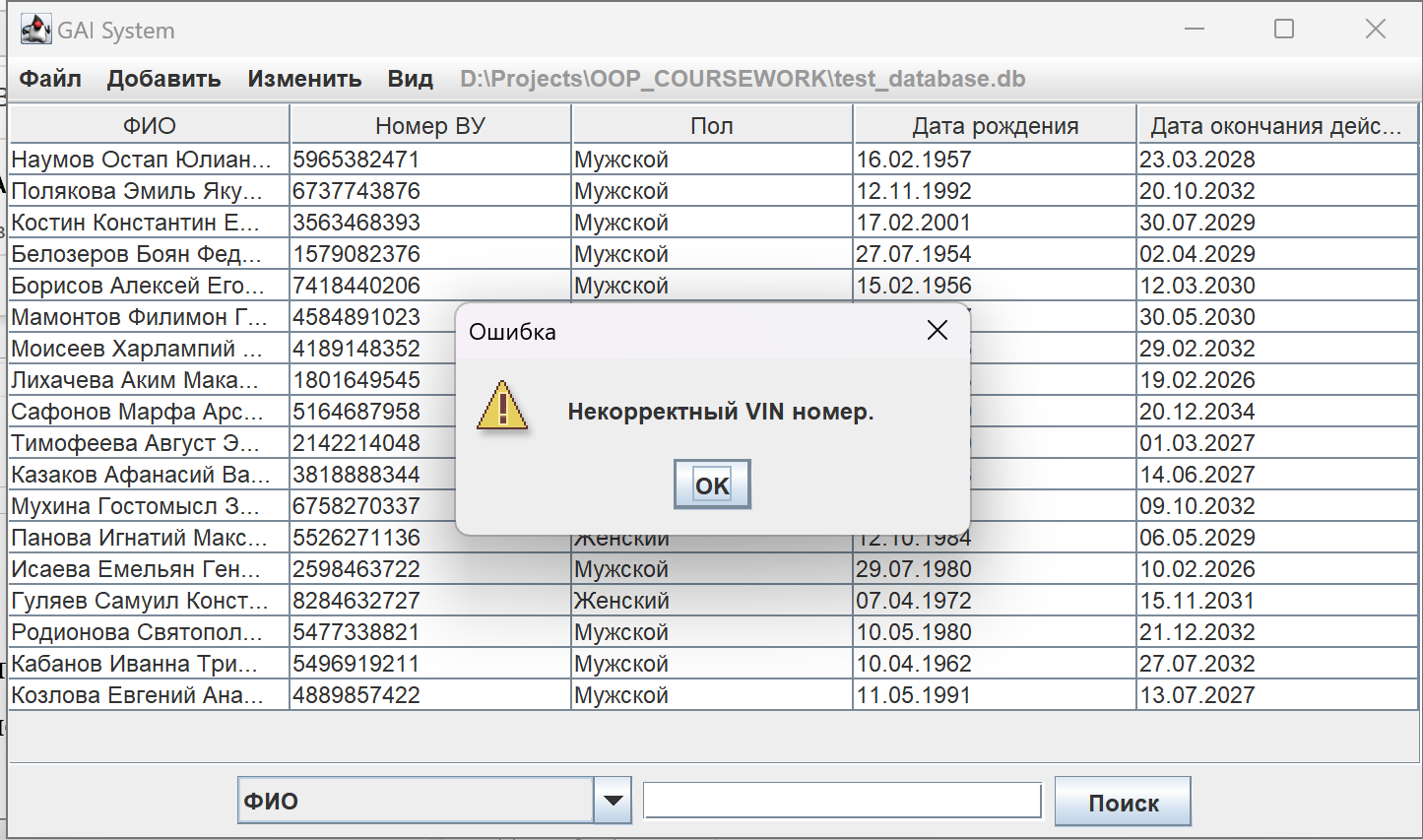


Рисунок 3.9 – Пример предупреждения

Нажимая на записи в таблице их можно выделять. Также можно выделить несколько записей, если нажать на «shift» или «ctrl» (Рис. 3.10). Выделенные записи можно удалить если нажать «Изменить» → «Удалить». Появится окно подтверждения (Рис. 3.11), и, если нажать «Yes», эти записи и все зависящие от них будут удалены (например, если удалить запись о водители будут также удалены сведенья о его машинах и нарушениях).

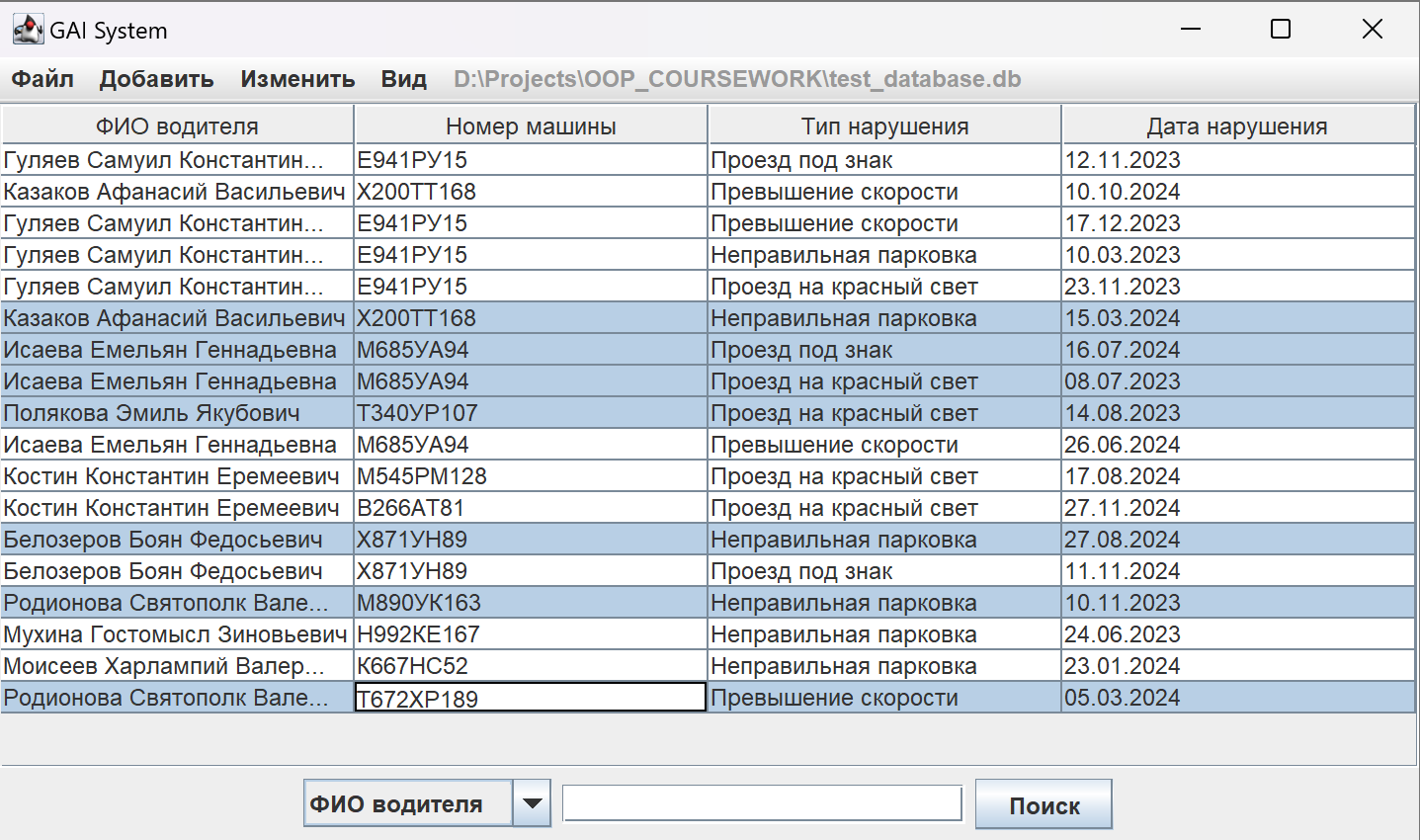
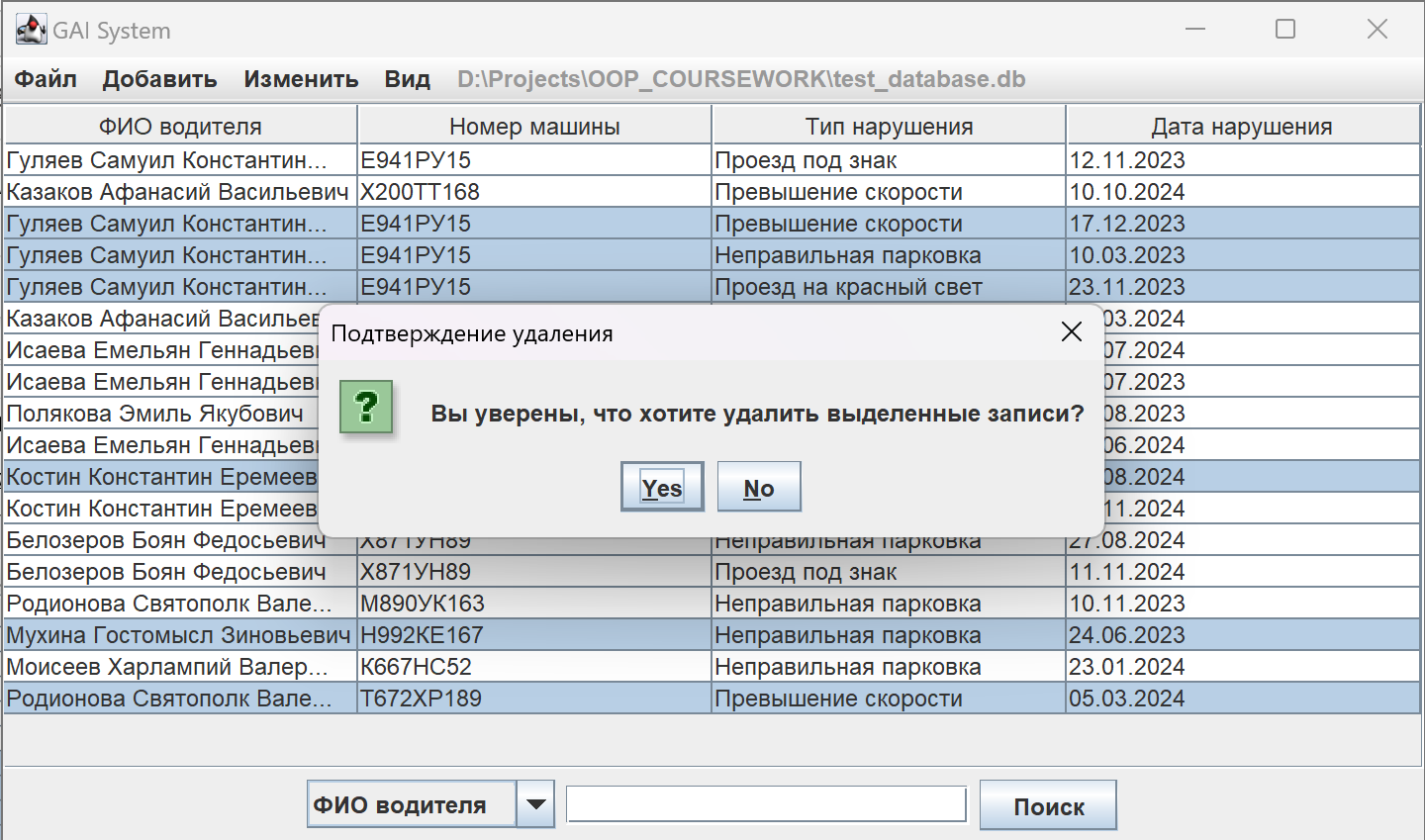


Рисунок 3.10 – Пример выделения записей

  
Рисунок 3.11 – Окно подтверждения удаления

Записи также можно выделить с помощью поиска, расположенного внизу окна программы. Если выбрать столбец, ввести строку в поле и нажать кнопку «Поиск», выделяться все записи, в которых содержится введённая подстрока (Рис 3.12).

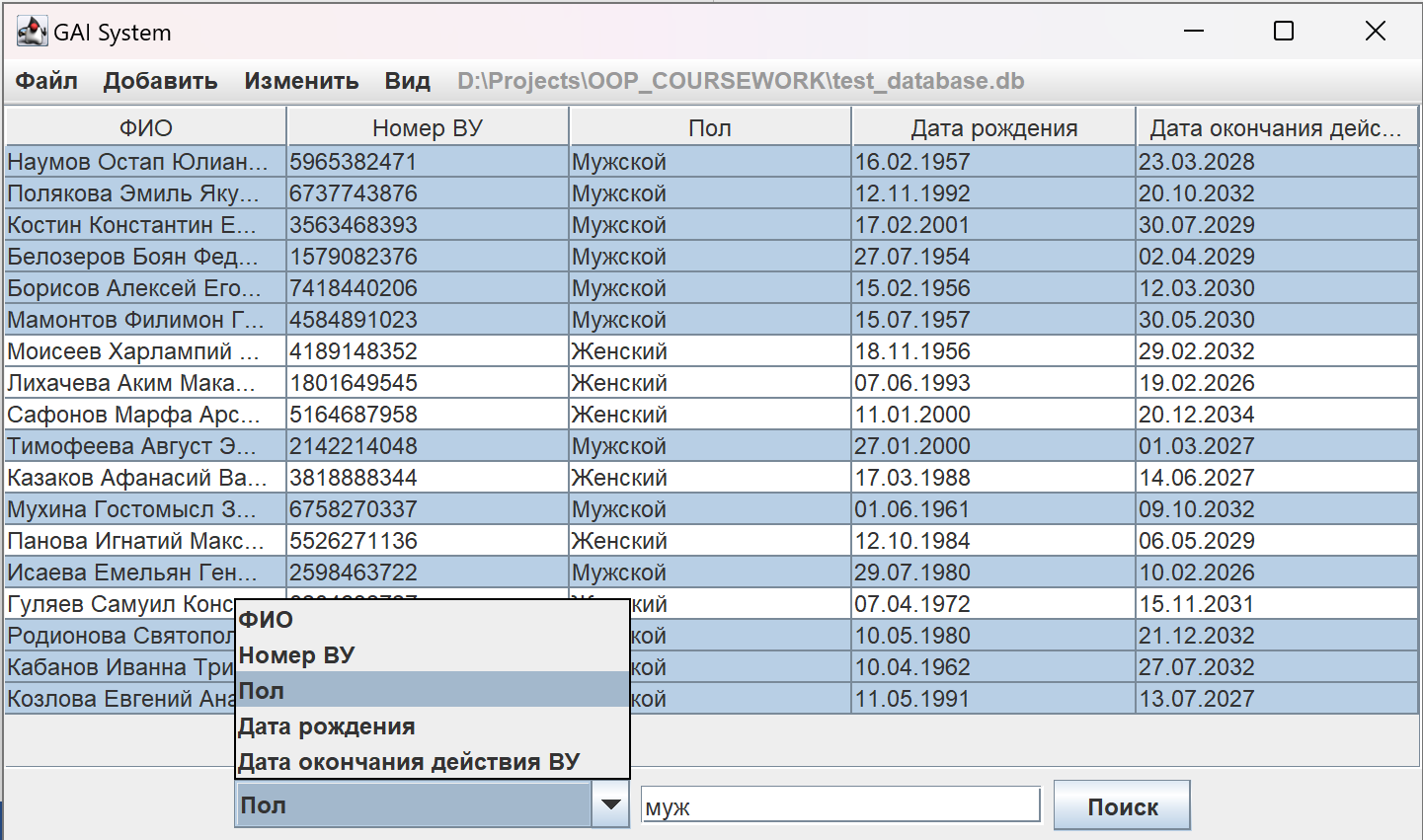


Рисунок 3.12 – Пример поиска по строке

Если требуется найти события произошедшие за определённый период, то выбрав в меню соответствующий столбец с датами в поле поиска нужно ввести строку в формате «dd.mm.yyyy-dd.mm.yyyy» и нажать «Поиск» (Рис. 3.13). Первая дата – начало периода, вторая – конец периода (концы включаются).

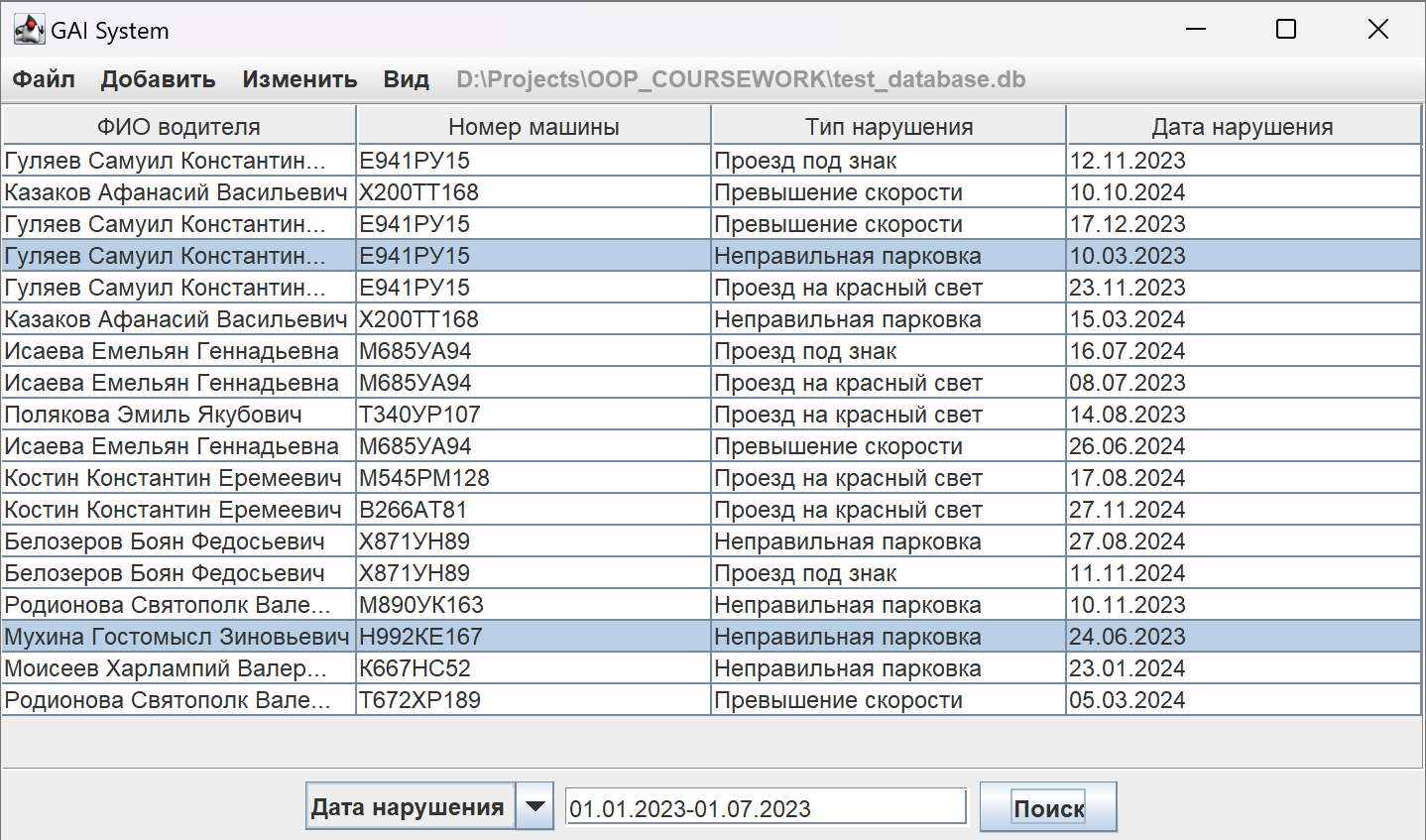


Рисунок 3.13 – Поиск за определённый период.

Если нужно отредактировать определённую запись, то нужно выделить её в таблице и нажать «Изменить» → «Редактировать». Откроется диалоговое окно, в котором можно изменить доступные поля (Рис 3.14). Недоступные для изменения поля (например VIN) – такие поля, изменение которые проводит к потере идентичности этого объекта. Также, в случае ввода неправильных данных, программа укажет на некорректные данные. Закрыть программу можно нажав на крестик в правом верхнем углу. Все данные сохраняться автоматически.

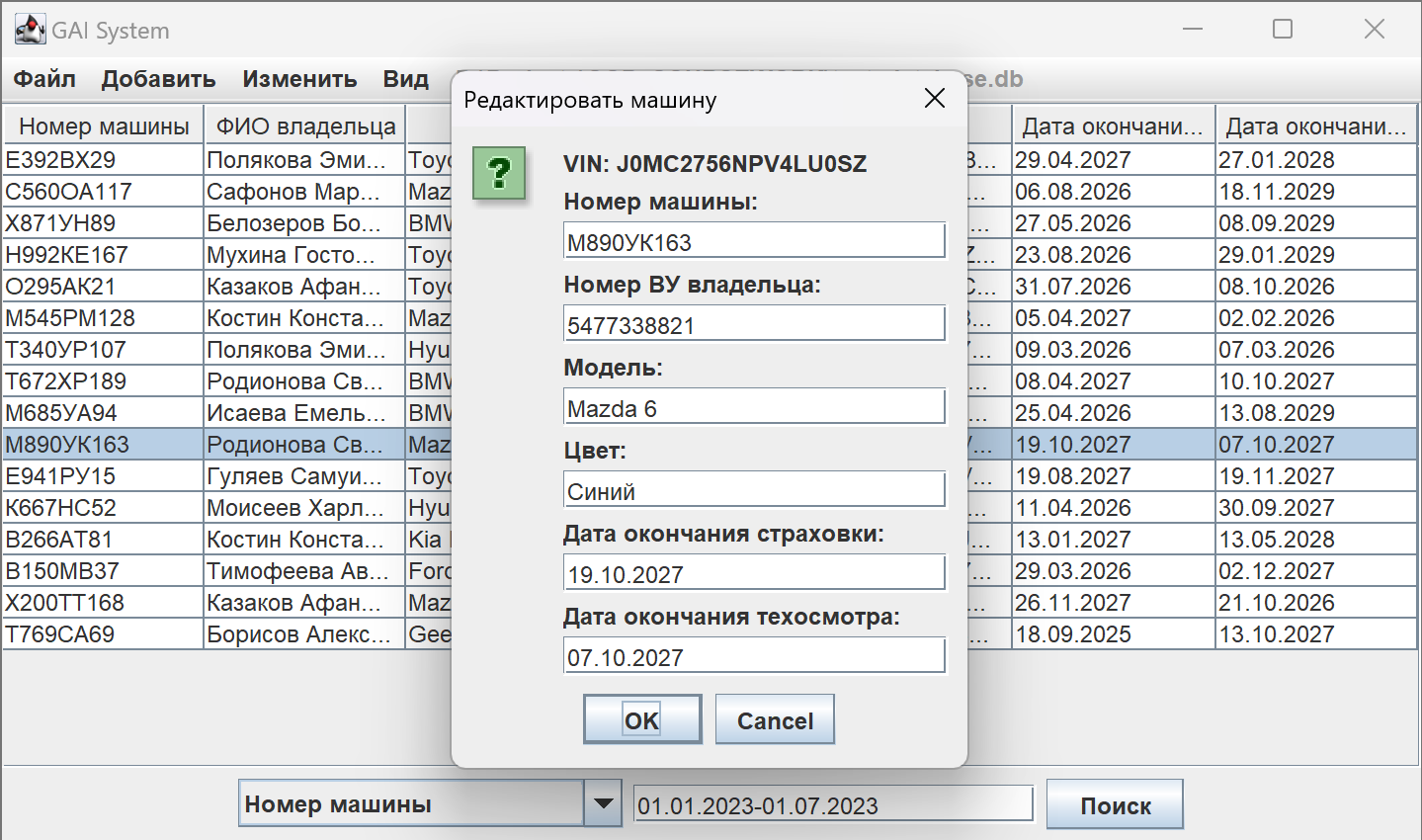


Рисунок 3.14 – Окно редактирования записи

# Заключение

В результате выполненной работы разработан программный комплекс для автоматизации учёта данных о водителях, автомобилях и нарушениях правил дорожного движения. Программа обеспечивает выполнение основных операций, включая добавление, редактирование, удаление и поиск данных, а также гарантирует контроль связей между сущностями и целостность базы данных.

*Разработанный комплекс включает:*

* Объектную модель, описывающую структуру данных и их взаимодействие.
* Диаграммы классов, последовательностей и действий, демонстрирующие логику работы системы.
* Руководство для оператора, описывающее назначение, функциональные возможности и порядок работы с программой.

Программный комплекс удовлетворяет поставленным требованиям, предоставляет удобный интерфейс и может быть использован для практического решения задач учёта и управления данными в рамках информационных систем.

# Приложение

Ссылка на репозиторий: <https://github.com/DexTver/OOP_COURSEWORK>