

Введение	5
1 Анализ технического задания	6
1.1 Описание предметной области	10
1.2 Поиск и подбор аналогов	11
2. Разработка моделей данных	144
2.1 Концептуальная модель данных	154
2.2 Логическая модель данных	166
2.3 Физическая модель данных	199
3. Разработка и реализация АИС	244
3.1 SQL запросы	244
3.2 Руководство пользователя	26
3.3 Руководство программиста	32
4. Тестирование АИС	39
Заключение	47
Список литературы	48
Приложение 1. Модели данных	49
Приложение 2. Текст программы	52

					МИВУ 10.03.01-05ПЗ		
Изм.	Лист	№ докум	Подпись	Дата	АИС ГИБДД		
Разраб		Воробьев Н.Р.					
Пров		Колпаков					
Н. Контр.							
Утв		Орлов А.А.					
					Литера	Лист	Листов
					у	4	66
					МИВЛГУ ИБ-122		

Введение

Информация находится в основе любой деятельности, а в нашем современном мире информационные технологии играют ключевую роль в организации и управлении различными процессами. Государственные учреждения не остаются в стороне и активно внедряют информационные системы для повышения эффективности своей работы. Одной из таких систем является Автоматизированная информационная система ГИБДД, предназначенная для автоматизации процессов регистрации автомобилей, нарушений ПДД, контроля транспортных средств и водителей.

Актуальность темы обусловлена необходимостью улучшения качества предоставляемых населению услуг и сокращения временных затрат на выполнение административных процедур. АИС ГИБДД позволяет значительно упростить и ускорить работу инспекторов, повысить прозрачность и точность учета данных за счет автоматизации большинства процессов.

Целью данной курсовой работы является исследование функционирования АИС ГИБДД, анализ ее компонентов, принципов работы.

Результаты исследования позволят сделать выводы о значении АИС ГИБДД для повышения эффективности работы ГИБДД, выявить возможные направления дальнейшего развития и усовершенствования системы.

1. Анализ технического задания

Современное развитие информационных систем требует выбора инструментов, которые наилучшим образом отвечают требованиям проекта. В контексте создания системы управления базами данных для ГИБДД(ГАИ) мы рассмотрим и сравним несколько технологий, чтобы понять, какие из них оптимально использовать.

Этот анализ охватывает языки программирования C# и C++, среды разработки Visual Studio и Eclipse, а также системы управления базами данных SQLite и MySQL. Приведенные сравнения помогут нам определить наиболее подходящие инструменты, учитывая их производительность, простоту использования, гибкость, безопасность и интеграцию с другими системами. Это обеспечит надежность и эффективность системы управления базами данных.

1. Характеристики C++:

Производительность: очень высокая, поскольку это язык низкого уровня, который позволяет манипулировать памятью и системными ресурсами.

Гибкость: обеспечивает детальный контроль над аппаратным обеспечением.

Использование: широко используется для разработки системного программного обеспечения, драйверов и высокопроизводительных приложений.

2. Характеристики C#:

Простота: Уровень выше, чем у C++, его легче изучать и использовать.

Платформа: тесно интегрирована с .NET, что облегчает разработку веб-приложений.

Безопасность: обеспечивает автоматическое управление памятью, что снижает вероятность возникновения ошибок, связанных с памятью.

3. Характеристики Visual Studio:

Функциональность: Обширный набор инструментов для отладки, тестирования и разработки.

Интеграция: отлично интегрируется с экосистемой Microsoft и поддерживает широкий спектр языков программирования.

Использование: очень удобная и понятная среда.

4. Характеристики Eclipse:

Открытый исходный код: бесплатный и доступный, поддерживается сообществом.

Кроссплатформенность: работает на различных операционных системах.

Плагины: существует огромное количество доступных плагинов для расширения функциональности.

5. Характеристики SQLite:

Легкий вес: не требует установки и настройки сервера.

Простота: SQLite прост в использовании и имеет компактную архитектуру.

Встроенная СУБД: отлично подходит для приложений, где требуется локальное хранение данных.

6. Характеристики MySQL:

Производительность: высокая производительность и масштабируемость для крупных проектов.

Функциональность: поддерживает сложные запросы, транзакции и интеграцию с другими СУБД.

Популярность: широко используется в веб-разработке и поддерживается большим сообществом.

Результат характеристик может быть представлен в виде таблицы 1:

Таблица 1 – Таблица характеристик

Характеристика	Производительность	гибкость	простота	интеграция	безопасность	использование
----------------	--------------------	----------	----------	------------	--------------	---------------

Продолжение таблицы 1

C++	Очень высокая, позволяет манипулировать памятью и системными ресурсами	Детальный контроль над аппаратным обеспечением	Низкий уровень, сложен для изучения	-	-	Системное ПО, драйверы, высокопроизводительные приложения
C#	Высокая, управляется средой .NET	Тесно интегрирован с .NET	Высокого уровня, легок в использовании	Тесно интегрирован с экосистемой Microsoft	Автоматическое управление памятью	Разработка веб-приложений и бизнес-логики
Visual Studio	-	-	Удобная и интуитивная среда	Отличная интеграция с Microsoft	-	Инструменты для отладки, тестирования и разработки
Eclipse	-	-	Требуется настройка и	Требуется установка плагинов	-	Бесплатная, поддерживаемая сообществом
SQLite	Легковесная и быстрая	Локальное хранилище данных	Легко использовать	Встроенная СУБД	Высокая за счет легковесности	Приложения с локальным хранилищем данных

Продолжение таблицы 1:

MySQL	Высокая производительность и масштабируемость для крупных проектов	Сложные запросы и транзакции	Требует настроек, хорошо документирован и поддерживается сообществом	Хорошая интеграция с другими СУБД и платформами	Высокая, поддерживает сложные транзакции и интеграции	Веб-разработка, поддерживается сообществом
-------	--	------------------------------	--	---	---	--

Для реализации СУБД ГИБДД(ГАИ) лучше всего подойдет язык программирования C# со средой разработки Visual Studio и базой данных SQLite, основанной на языке SQL, выбор этих инструментов разработки являются оптимальным выбором.

Они предлагают следующие преимущества:

- C#: высокая производительность и управляемость, легок в изучении и использовании и автоматическое управление памятью, которое снижает вероятность ошибок.
- Visual Studio: Мощная и интуитивно понятная, и удобная интегрированная среда разработки, поддержка множества языков и богатый набор инструментов.
- SQLite: Легковесная и простая в использовании СУБД, не требует установки и настройки сервера, прост в использовании и имеет компактную архитектуру, идеально подходит для приложений с локальным хранилищем данных.

Эти инструменты обеспечат стабильную, безопасную и эффективную работу системы управления базой данных для гостиницы.

1.1 Описание предметной области

В данной курсовой работе рассматривается предметная область ГИБДД(ГАИ).

Государственная автомобильная инспекция (ГАИ) и Государственная инспекция безопасности дорожного движения (ГИБДД) - это органы, отвечающие за обеспечение безопасности на дорогах и контроль за соблюдением правил дорожного движения.

Основные задачи:

1. Обеспечение безопасности дорожного движения:
 - Разработка и внедрение мер по предупреждению дорожно-транспортных происшествий.
2. Контроль и надзор:
 - Проведение проверок соблюдения правил дорожного движения водителями.
 - Осуществление контроля за техническим состоянием транспортных средств.
 - Выявление и привлечение к ответственности нарушителей ПДД.
3. Регистрация транспортных средств и выдача водительских удостоверений:
 - Регистрация автомобилей и других транспортных средств.
 - Выдача, замена и учет водительских удостоверений.
 - Проведение экзаменов на получение водительских прав.
4. Расследование дорожно-транспортных происшествий:
 - Оформление и расследование ДТП, установление причин и условий их возникновения.
 - Составление протоколов и направлений на административное расследование.

Основная задача курсовой работы - разработка приложения для автоматизации работы ГИБДД. Это позволит упростить ведение учета и облегчить работу персонала.

В ГАИ можно выделить следующие сущности:

Авто – транспортные средства, зарегистрированные в ГИБДД:

- Марка, модель, год, государственный регистрационный номер, страховое свидетельство, свидетельство о регистрации транспортного средства.

Водители, зарегистрированные в ГИБДД:

- ФИО, дата рождения, серия и номер паспорта, телефон, адрес, водительское удостоверение, свидетельство о регистрации транспортного средства.

Нарушения ПДД:

- Номер протокола, дата нарушения, тип нарушения, номерной знак тс, размер штрафа(руб.), краткое описание, статус штрафа и водитель.

Регистрация тс:

- Владелец, государственный номерной знак, свидетельство о регистрации тс, страховое свидетельство, дата регистрации.

Сотрудник ГИБДД(ГАИ):

- ФИО, должность, звание, серия и номер паспорта, телефон, адрес, дата рождения, выписанный протокол.

База данных будет содержать информацию о владельцах, транспортных средств, сотрудниках ГИБДД(ГАИ), нарушениях и регистрации. Это позволит автоматизировать все основные процессы работы ГИБДД(ГАИ).

1.2 Поиск и подбор аналогов

При поиске и подборе аналогов баз данных для ГИБДД(ГАИ), может быть полезно обратить внимание на решения, которые предлагают схожий функционал.

Ниже представлены популярные системы управления базами данных для управления операциями, необходимыми для ГАИ.

1.2.1 ГАИ, транспорт:

Ссылка: <https://e-pasluga.by/services/gai-transport>

Достоинства:

- Возможность подачи заявлений и документов онлайн, что экономит время и упрощает процесс взаимодействия с ГАИ.
- Интуитивно понятный интерфейс, который позволяет легко находить нужные услуги и информацию.
- Широкий спектр услуг, включая регистрацию транспортных средств, получение водительских удостоверений, оплату штрафов и другие.

Недостатки:

- Возможные технические сбои или ошибки, которые могут замедлить процесс получения услуги.

Особенности:

интегрированность и многофункциональность.

1.2.2 Госавтоинспекция МВД России:

Ссылка: <https://xn--80aebkobnwfncsfk1e0h.xn--p1ai/>

Достоинства:

- Сайт предоставляет централизованный доступ ко всем необходимым услугам и информации, связанным с деятельностью Госавтоинспекции, включая правила дорожного движения, новости, и отчеты о безопасности на дорогах. Интеграция с каналами продаж и системами-партнерами
- Пользователи могут легко находить нужные услуги, включая проверку штрафов, регистрацию транспортных средств, получение справок и других документов, благодаря интуитивно понятному интерфейсу.
- Сайт предлагает разнообразные электронные услуги, такие как подача заявлений и документов, оплата штрафов, запись на прием и многое другое, что упрощает взаимодействие с органами ГИБДД.

Недостатки:

- Людям, которые не обладают достаточными навыками работы с компьютером или интернетом, может быть сложно использовать сайт без посторонней помощи.

Особенности:

Сайт предлагает интерактивные сервисы, такие как калькулятор штрафов, тесты на знание ПДД, и онлайн-консультации, а также имеет мобильную версию.

1.2.3 ГИБДД МВД России:

Ссылка: <https://xn--b1aew.xn--p1ai/%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D1%8B-%D0%B3%D0%B8%D0%B1%D0%B4%D0%B4?ysclid=m4rhdje51a558798178>

Достоинства:

- Сайт предоставляет удобный доступ к разнообразным услугам ГИБДД, таким как проверка штрафов, регистрация транспортных средств, получение справок и многие другие. Интеграция с популярными сайтами бронирования
- Возможность подачи заявлений и документов онлайн, что экономит время и упрощает взаимодействие с ГИБДД. Онлайн-оплата штрафов и госпошлин, что ускоряет процесс и делает его более удобным для пользователей.

Недостатки:

- Не все услуги могут быть доступны онлайн, и пользователи могут быть вынуждены посещать физические отделения для решения определенных вопросов.

Особенности:

Возможность проверки и оплаты штрафов, а также подачи заявлений и получения справок через интегрированные системы. Сайт имеет мобильную версию

					МИВУ 10.03.01-03ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		13

2. Разработка моделей данных

Разработка моделей данных включает 3 этапа:

Первый этап — это разработка концептуальной модели, которая включает в себя следующее:

- определение сущностей (объектов, понятий) предметной области и их взаимосвязей;
- выявление атрибутов сущностей и описание их характеристик;
- построение диаграммы сущность-связь (ER-диаграммы) для визуального представления модели.

Второй этап — это разработка логической модели, которая включает в себя следующее:

- трансформация концептуальной модели в структуру, подходящую для конкретной СУБД;
- определение типов данных, ключей, индексов, ограничений целостности;
- нормализация данных для устранения избыточности и аномалий;
- спецификация отношений между сущностями (таблицами).

Третий этап — это разработка физической модели, которая включает в себя следующее:

- оптимизация логической модели для конкретной СУБД и аппаратной платформы;
- определение физического хранения данных: файлов, томов, буферов и т.д.;
- результат - физическая схема базы данных, готовая для реализации.[1]

2.1 Концептуальная модель данных

Концептуальная модель данных является первым и наиболее абстрактным уровнем проектирования базы данных. На этом этапе определяются ключевые сущности предметной области, их атрибуты и взаимосвязи.

Сущность «Авто» имеет следующие атрибуты: марка, модель, год выпуска авто, государственный регистрационный номер, страховое свидетельство, свидетельство о регистрации транспортного средства, владелец.

Сущность «Водитель» имеет следующие атрибуты: ФИО, дата рождения, серия и номер паспорта, телефон, адрес, водительское удостоверение, свидетельство о регистрации транспортного средства.

Сущность «Нарушение» имеет следующие атрибуты: номер протокола, дата нарушения, госномер, размер штрафа(руб.), краткое описание, статус штрафа, водитель, сотрудник ГАИ(кто выписал протокол).

Сущность «Регистрация» имеет следующие атрибуты: владелец, госномер, СТС, дата регистрации тс.

Сущность «Сотрудник ГИБДД» имеет следующие атрибуты: ФИО, должность, звание, серия и номер паспорта, телефон, адрес проживания, дата рождения, выписанный протокол.

Можно выделить следующие связи между сущностями:

- Между сущностями «Авто» и «Водители» существует связь «многие-к-одному», так как один водитель может владеть несколькими автомобилями;

- Между сущностями «Нарушения» и «Сотрудник ГИБДД» - связь «многие-к-одному», так как один сотрудник ГАИ может выписать несколько протоколов о нарушениях;

- Между сущностями «Регистрация» и «Авто» - связь «многие-к-одному», так как одно транспортное средство может быть зарегистрировано несколько раз на различных владельцев;

- Между сущностями «Регистрация» и «Водители» - связь «многие-к-одному», так как один водитель может зарегистрировать несколько автомобилей;
- Между сущностями «Водители» и «Нарушения» - связь «многие-к-многим», так как один водитель может совершить несколько нарушений, и одно нарушение может быть связано с несколькими водителями (например, если один водитель был пассажиром в автомобиле другого водителя)

На рисунке 1 изображена концептуальная модель базы данных.

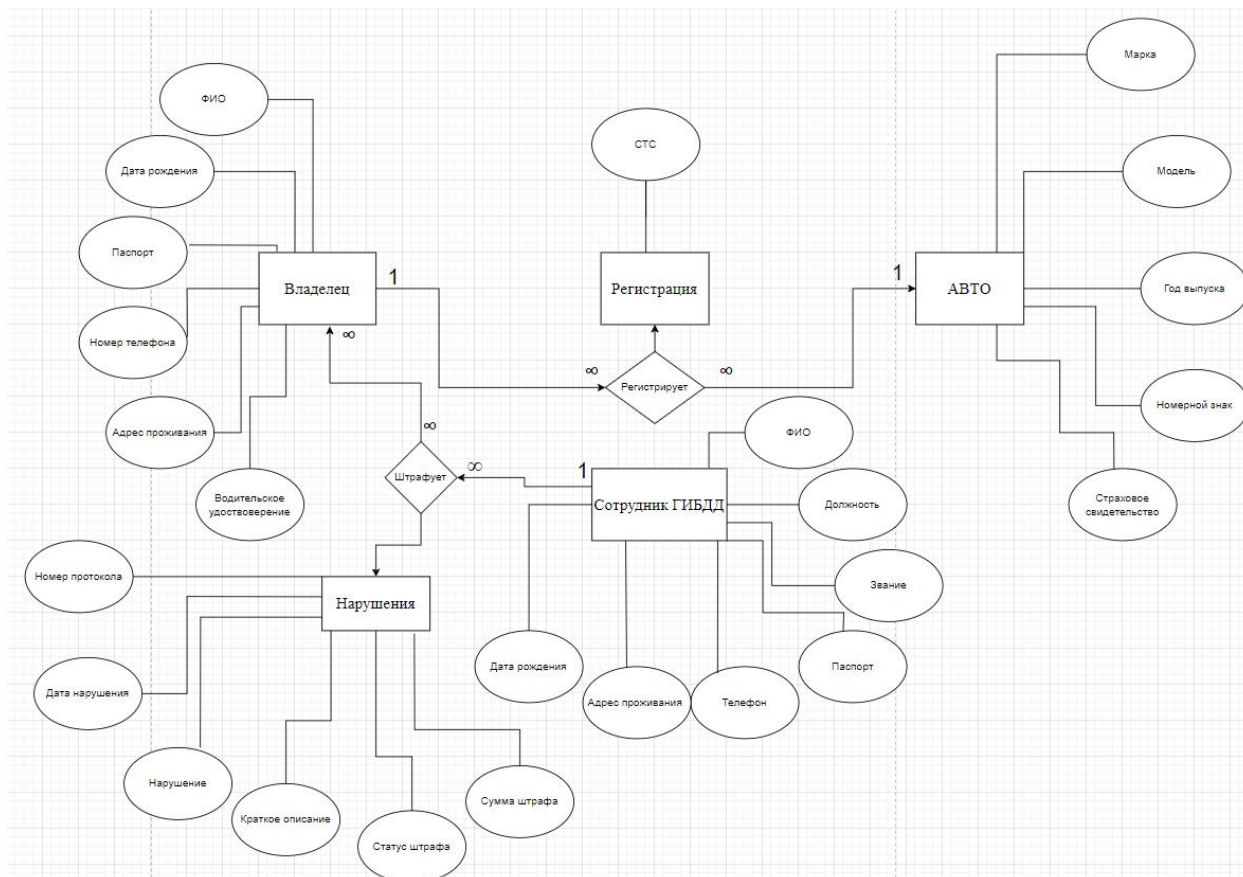


Рисунок 1 – Концептуальная модель базы данных

2.2 Логическая модель данных

Логическая модель данных является следующим этапом после концептуальной модели. На этом этапе концептуальная модель трансформируется в структуру, подходящую для конкретной системы управления базами данных (СУБД)[4].

Основные сущности и их атрибуты:

2.2.1 Авто:

- ID - уникальный идентификатор автомобиля
- Марка – марка автомобиля
- Модель – модель автомобиля
- Год – год выпуска автомобиля
- Номерной знак - государственный регистрационный номер
- Страховое свидетельство - страховой полис автомобиля
- Свидетельство о регистрации - Свидетельство о регистрации

транспортного средства

- Владелец – ФИО владельца авто

2.2.2 Водитель:

- ID - Уникальный идентификатор водителя
- ФИО – фамилия имя отчество водителя
- Дата рождения
- Паспорт – серия и номер паспорта
- Телефон – номер телефона водителя
- Адрес проживания водителя
- Водительское удостоверение
- Свидетельство о регистрации - Свидетельство о регистрации транспортного средства

2.2.3 Нарушения:

- ID - уникальный идентификатор нарушения
- Авто_Id - Идентификатор автомобиля
- ГАИ_Сотрудник_ID - Идентификатор сотрудника ГАИ
- Номер протокола нарушения
- Дата нарушения
- Нарушение
- Штраф – сумма штрафа

- Описание – краткое описание нарушения
- Статус штрафа – статус оплаты
- Водитель – ФИО
- Выписал протокол - ФИО сотрудника, выписавшего протокол

2.2.4 Регистрация:

- ID - уникальный идентификатор регистрации
- Водитель_ID – идентификатор водителя
- Авто_ID – идентификатор автомобиля
- Номерной знак - государственный регистрационный номер
- Свидетельство о регистрации - Свидетельство о регистрации транспортного средства
- Страхование свидетельство - страховой полис автомобиля
- Дата регистрации
- Водитель – ФИО водителя

2.2.5 Сотрудник ГИБДД:

- ID - уникальный идентификатор сотрудника
- ФИО
- Должность
- Звание
- Паспорт – серия и номер паспорта
- Телефон – номер телефона сотрудника
- Адрес проживания
- Дата рождения
- Выписанный протокол - протокол, выписанный сотрудником

Данная логическая модель данных отражает основные сущности, связанные с ГАИ: Водители(владельцы), автомобили, сотрудники ГАИ, регистрация и нарушения. Она позволяет хранить и управлять данными, необходимыми для эффективного функционирования ГИБДД(ГАИ).

Такая логическая модель данных представлена на рисунке 2.

					МИВУ 10.03.01-03ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		18

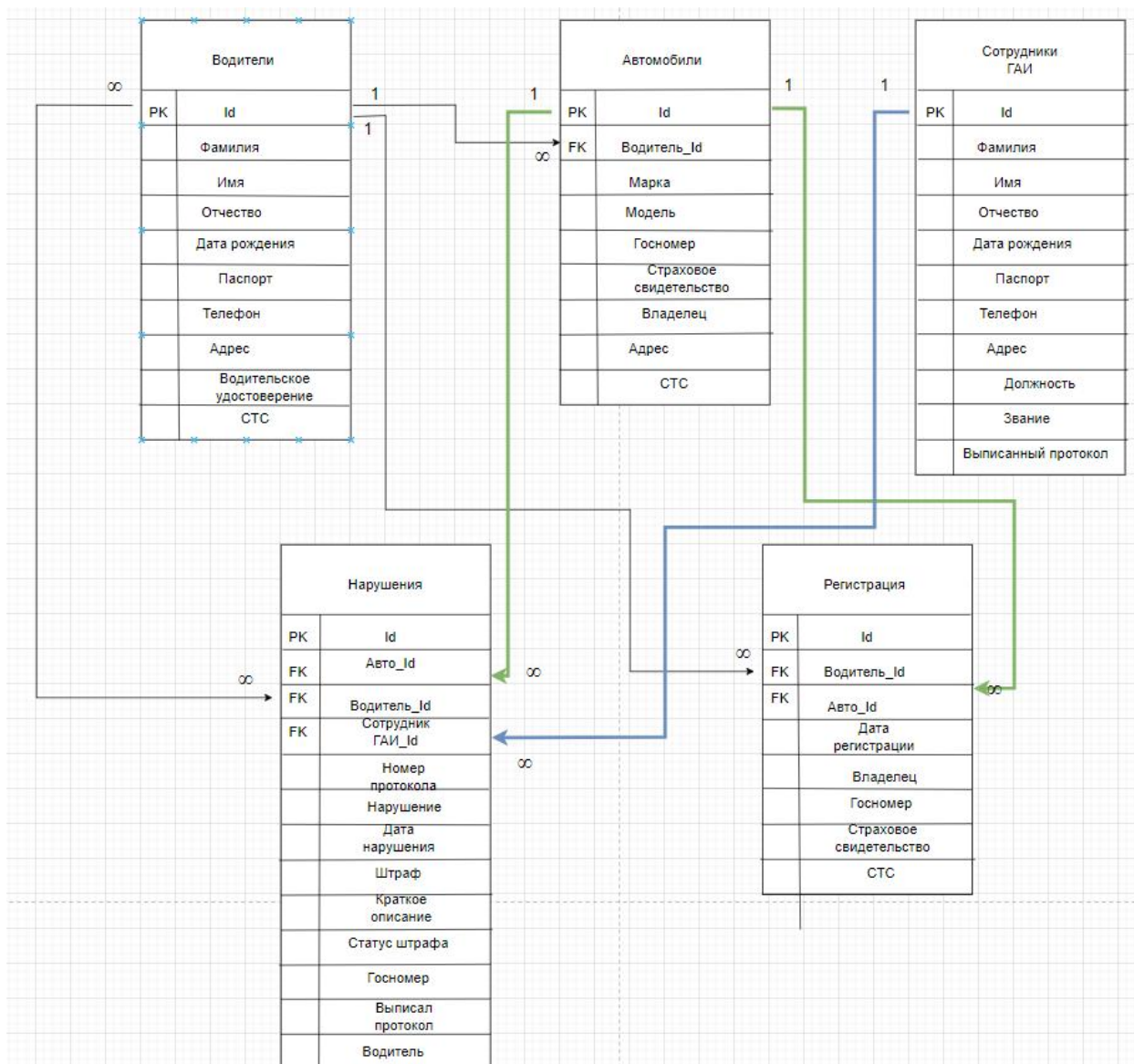


Рисунок 2 – Логическая модель данных

2.3 Физическая модель данных

Физическая модель данных — это этап проектирования базы данных предполагает реализацию логической модели с учетом конкретной системы управления базами данных (СУБД) и требований к производительности системы.[3]

К основным задачам физической модели данных относится:

1) Определение физических характеристик таблиц и полей:

- Типы данных для каждого атрибута (integer, varchar, date и т.д.)
- Размеры полей (длина строк)

- Ограничения целостности данных (первичные/внешние ключи, уникальность, проверка значений)

2) Проектирование физической структуры базы данных:

- Определение физического хранения данных (файлы, тома, блоки)
- Настройка индексов для ускорения доступа к данным
- Решения по партиционированию и кластеризации таблиц
- Механизмы резервного копирования и восстановления данных

3) Оптимизация производительности:

- Анализ запросов и создание оптимальных индексов
- Денормализация данных для повышения скорости выборки
- Использование материализованных представлений
- Настройка буферизации, кэширования, параллелизма

На рисунке 3 изображена физическая модель данных.

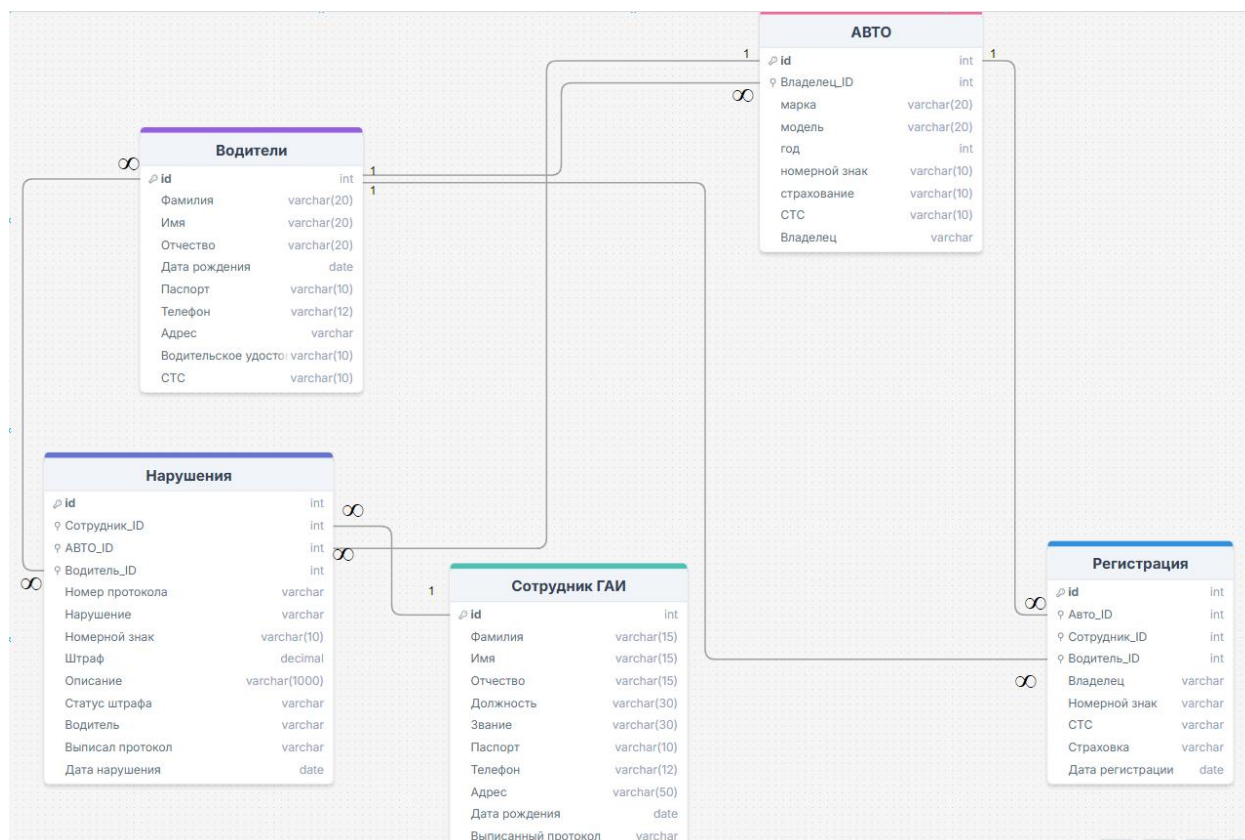


Рисунок 3 – Физическая модель данных

Физическая модель данных состоит из следующих таблиц:

2.3.1 Таблица "Авто":

- Id: integer, primary key, autoincrement
- Марка: varchar(20), NOT NULL
- Модель: varchar(20), NOT NULL
- Год: integer, NOT NULL, CHECK (Год >= 1886)
- НомернойЗнак: varchar(10), NOT NULL, UNIQUE
- Страхование: varchar(10)
- СвидетельствоОРегистрации: varchar(10)
- Владелец: varchar(50)
- Владелец_ID: integer, внешний ключ на таблицу Водители.

Для данной таблицы могут быть определены:

- первичный ключ Id для уникальной идентификации записей
- уникальный индекс по полю НомернойЗнак для предотвращения

дублирования;

- ограничения целостности для полей Год, Марка и Модель.

2.3.2 Таблица "Водители":

- Id: integer, primary key, autoincrement
- Фамилия: varchar(20), NOT NULL
- Имя: varchar(20), NOT NULL
- Отчество: varchar(20)
- ДатаРождения: date, NOT NULL
- Паспорт: varchar(10), NOT NULL, UNIQUE
- Телефон: varchar(12)
- Адрес: varchar(100)
- Водительское удостоверение: varchar(10)
- Свидетельство о регистрации: varchar(10)

Ключевые особенности:

- первичный ключ Id для уникальной идентификации записей;;
- уникальный индекс по полю НомерПаспорта;

- ограничения целостности для полей Телефон и НомерПаспорта.

2.3.3 Таблица "Нарушения":

- ID: integer, primary key, autoincrement
- НомерПротокола: varchar(20), NOT NULL
- ДатаНарушения: date, NOT NULL
- Нарушение: varchar(255), NOT NULL
- Номерной знак: varchar(10), NOT NULL
- Штраф: decimal, NOT NULL
- Описание: varchar(1000), NOT NULL
- СтатусШтрафа: varchar(10), NOT NULL
- Водитель: varchar(50)
- Авто_Id: integer, внешний ключ на таблицу Авто
- ГАИ_Сотрудник_ID: integer, внешний ключ на таблицу СотрудникГАИ
- ВыписалПротокол: varchar(50)

Ключевые особенности:

- связь с таблицами Авто и СотрудникГАИ через внешние ключи;
- уникальный индекс по полю НомерПротокола;
- граничения целостности для полей СтатусШтрафа и НомерПротокола.

2.3.4 Таблица "Регистрация":

- Id: integer, primary key, autoincrement
- Владелец: varchar(50)
- НомернойЗнак: varchar(10)
- СвидетельствоОРегистрации: varchar(10)
- Страховка: varchar(10)
- ДатаРегистрации: date, NOT NULL
- Водитель_ID: integer, внешний ключ на таблицу Водители(Id)
- Авто_ID: integer, внешний ключ на таблицу Авто(Id)
- Сотрудник_ID: integer, внешний ключ на таблицу СотрудникГАИ(Id)

Ключевые особенности:

- связь с таблицами Водители, Авто и СотрудникГАИ через внешние ключи;

- индексы для ускорения поиска по полям Водитель_ID и Авто_ID;
- ограничения целостности для всех полей.

2.3.5 Таблица "СотрудникГАИ":

- Id: integer, primary key, autoincrement
- Фамилия: varchar(15), NOT NULL
- Имя: varchar(15), NOT NULL
- Отчество: varchar(15), NOT NULL
- Должность: varchar(30), NOT NULL
- Звание: varchar(30), NOT NULL
- НомерПаспорта: varchar(10), NOT NULL
- Телефон: varchar(12), NOT NULL
- Адрес: varchar(50), NOT NULL
- ДатаРождения: date, NOT NULL
- ВыписанныйПротокол: varchar(20), NOT NULL,

Ключевые особенности:

- первичный ключ Id для уникальной идентификации записей;
- индексы для ускорения поиска по полям НомерПаспорта и Телефон;
- ограничения целостности для всех полей.

3. Разработка и реализация АИС

3.1 SQL запросы

SQL (Structured Query Language) - это стандартный язык для работы с реляционными базами данных, который позволяет выполнять широкий спектр операций: от создания и изменения структуры базы данных до манипулирования данными и их извлечения. Использование SQL-запросов является ключевым аспектом разработки и внедрения автоматизированных информационных систем, поскольку они обеспечивают взаимодействие с базой данных и выполняют различные операции с данными.[2]

3.1.1 Запрос создания таблицы (CREATE TABLE):

Запрос, который создает новую таблицу с наименованием Авто.

```
CREATE TABLE Авто (  
    Id INTEGER PRIMARY KEY AUTOINCREMENT,  
    Марка TEXT NOT NULL  
    CHECK (LENGTH(Марка) <= 20),  
    Модель TEXT NOT NULL  
    CHECK (LENGTH(Модель) <= 20),  
    Год INTEGER NOT NULL  
    CHECK (Год >= 1886),  
    НомернойЗнак TEXT NOT NULL  
    UNIQUE  
    CHECK (LENGTH(НомернойЗнак) <= 10),  
    Страхование TEXT CHECK (LENGTH(Страхование) = 10),  
    СвидетельствоОРегистрации TEXT CHECK  
(LENGTH(СвидетельствоОРегистрации) = 10),  
    Владелец TEXT,
```

```

Владелец_ID INTEGER,
FOREIGN KEY (
Владелец_ID
)
REFERENCES Водители (Id) ON DELETE CASCADE
);

```

Этот запрос делает следующие действия:

- CREATE TABLE Авто создает новую таблицу с именем Авто.
- Id INTEGER PRIMARY KEY AUTOINCREMENT: Создает столбец Id, который является первичным ключом (уникальный идентификатор для каждой записи) и автоматически увеличивается при добавлении новых записей..
- Марка TEXT NOT NULL CHECK (LENGTH(Марка) <= 20): Создает столбец Марка типа TEXT, который не может быть пустым (NOT NULL) и длина значения которого не должна превышать 20 символов.
- Модель TEXT NOT NULL CHECK (LENGTH(Модель) <= 20): Создает столбец Модель типа TEXT, который не может быть пустым и длина значения которого не должна превышать 20 символов.
- Год INTEGER NOT NULL CHECK (Год >= 1886): Создает столбец Год типа INTEGER, который не может быть пустым и значение которого должно быть больше или равно 1886 (год изобретения автомобиля).
- НомернойЗнак TEXT NOT NULL UNIQUE CHECK (LENGTH(НомернойЗнак) <= 10): Создает столбец НомернойЗнак типа TEXT, который не может быть пустым, должен быть уникальным и длина значения которого не должна превышать 10 символов.
- Страхование TEXT CHECK (LENGTH(Страхование) = 10): Создает столбец Страхование типа TEXT, длина значения которого должна быть равна 10 символам (например, номер страхового полиса).

- СвидетельствоОРегистрации TEXT CHECK
(LENGTH(СвидетельствоОРегистрации) = 10): Создает столбец
СвидетельствоОРегистрации типа TEXT, длина значения которого должна быть
равна 10 символам.

- Владелец TEXT: Создает столбец Владелец типа TEXT, который может
содержать имя владельца.

- Владелец_ID INTEGER: Создает столбец Владелец_ID типа INTEGER,
который будет использоваться в качестве внешнего ключа.

3.1.2 Запрос обновления обновление(UPDATE) данных в таблице:

(Создан через триггер)

UPDATE Авто

SET Владелец = (

SELECT

Фамилия || ' ' ||

SUBSTR(Имя, 1, 1) || '. ' ||

IFNULL(SUBSTR(Отчество, 1, 1) || '.', '')

FROM Водители

WHERE Водители.Id = NEW.Владелец_ID

)

WHERE Id = NEW.Id;

Этот запрос делает следующие действия:

- UPDATE Нарушения SET: Начало команды обновления, указывающее, что
мы будем обновлять данные в таблице Нарушения.

- Водитель = (SELECT Владелец FROM Авто WHERE Авто.Id =
NEW.Авто_Id);

- Подзапрос (SELECT), который выбирает значение столбца Владелец из
таблицы Авто.
- Условие WHERE Авто.Id = NEW.Авто_Id указывает, что выбирается
значение для автомобиля с идентификатором, соответствующим

значению NEW.Авто_Id. NEW ссылается на новые данные, которые только что были вставлены или обновлены в записи.

Это действие обновляет столбец Водитель в таблице Нарушения, устанавливая значение в соответствии с именем владельца автомобиля из таблицы Авто.

- ВыписалПротокол = (SELECT Звание || ' ' || Фамилия || ' ' || SUBSTR(Имя, 1, 1) || '. ' || IFNULL(SUBSTR(Отчество, 1, 1) || '.', '') FROM СотрудникГАИ WHERE СотрудникГАИ.Id = NEW.ГАИ_Сотрудник_ID);

- Подзапрос (SELECT), который объединяет значения из столбцов Звание, Фамилия, Имя, и Отчество из таблицы СотрудникГАИ.
- Условие WHERE СотрудникГАИ.Id = NEW.ГАИ_Сотрудник_ID указывает, что выбирается значение для сотрудника ГАИ с идентификатором, соответствующим значению NEW.ГАИ_Сотрудник_ID.

Это действие обновляет столбец ВыписалПротокол в таблице Нарушения, устанавливая значение в соответствии с данными сотрудника ГАИ, который выписал протокол.

3.1.3 Запросы вставки в таблицу (INSERT):

1) Запрос для вставки новой записи в таблицу Нарушения:

INSERT INTO Нарушения (НомерПротокола, ДатаНарушения, Нарушение, НомернойЗнак, Штраф, Описание, СтатусШтрафа, Авто_Id, ГАИ_Сотрудник_ID) VALUES ('№001', '2023-03-01', 'Превышение скорости', 'A111BC77', 5000, 'Превышение скорости на 20 км/ч', 'не оплачен', 1, 1);

Этот запрос выполняет следующие действия:

- INSERT INTO Нарушения (НомерПротокола, ДатаНарушения, Нарушение, НомернойЗнак, Штраф, Описание, СтатусШтрафа, Авто_Id, ГАИ_Сотрудник_ID) VALUES: Определяет таблицу Нарушения и столбцы, в которые будут вставляться значения: НомерПротокола, ДатаНарушения, Нарушение, НомернойЗнак, Штраф, Описание, СтатусШтрафа, Авто_Id, ГАИ_Сотрудник_ID.

- ('№001', '2023-03-01', 'Превышение скорости', 'A111BC77', 5000, 'Превышение скорости на 20 км/ч', 'не оплачен', 1, 1): вставляет значения для записи.

3.1.4 Запросы выборки из таблицы (SELECT):

- SELECT Владелец: указывает, что мы хотим выбрать значение столбца Владелец из таблицы Авто.

- FROM Авто: указывает, что выборка будет производиться из таблицы Авто.

- WHERE Авто.Id = NEW.Авто_Id: условие, определяющее, какие строки будут выбраны. В данном случае мы выбираем строки, где значение столбца Id в таблице Авто соответствует значению NEW.Авто_Id.

3.1.5 Запросы удаления записи из таблицы (DELETE)

DELETE FROM Нарушения

WHERE НомерПротокола = '№001';

Этот запрос удалит все записи из таблицы Нарушения, у которых номер протокола равен №001.

3.2 Руководство пользователя

При открытии приложения первым шагом для пользователя является окно авторизации, показанное на рисунке 4, которое служит для аутентификации пользователей.

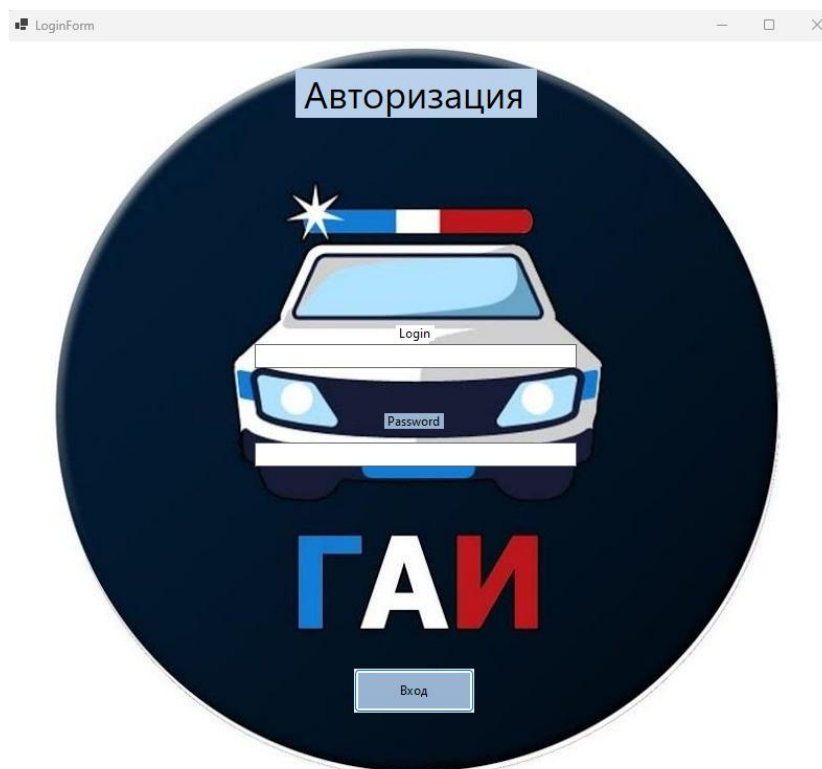


Рисунок 4 – окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, что показано на рисунках 5, 6 и соответственно если логин и пароль правильные, то он попадает в рабочее окно, что показано рисунках 5,6,7.

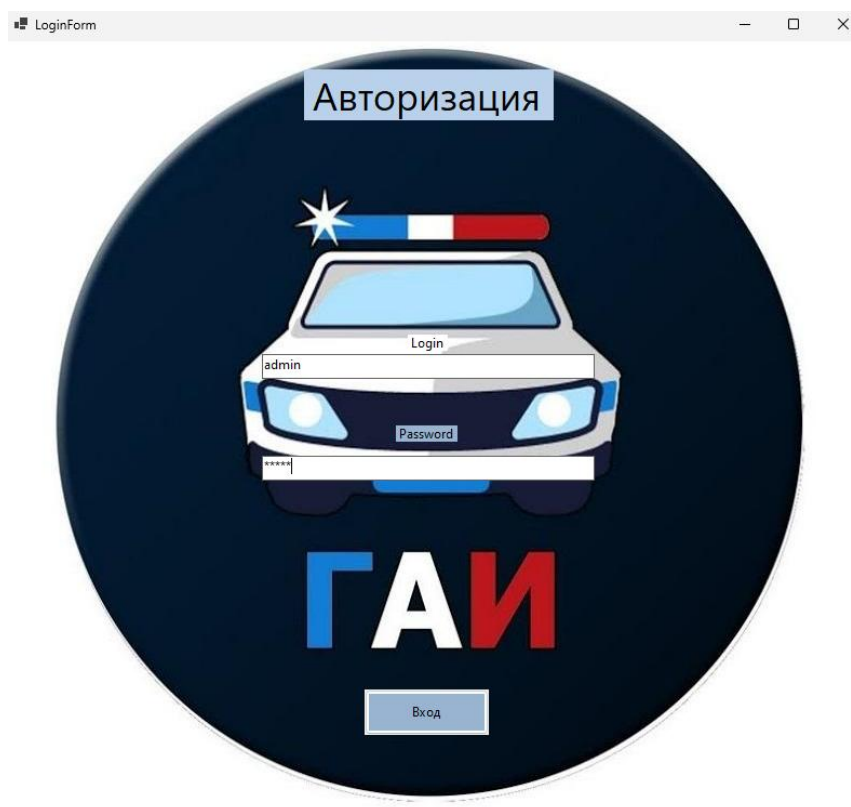


Рисунок 5 – ввод данных от учетной записи администратора

Изм.	Лист	№ докум	Подпись	Дата

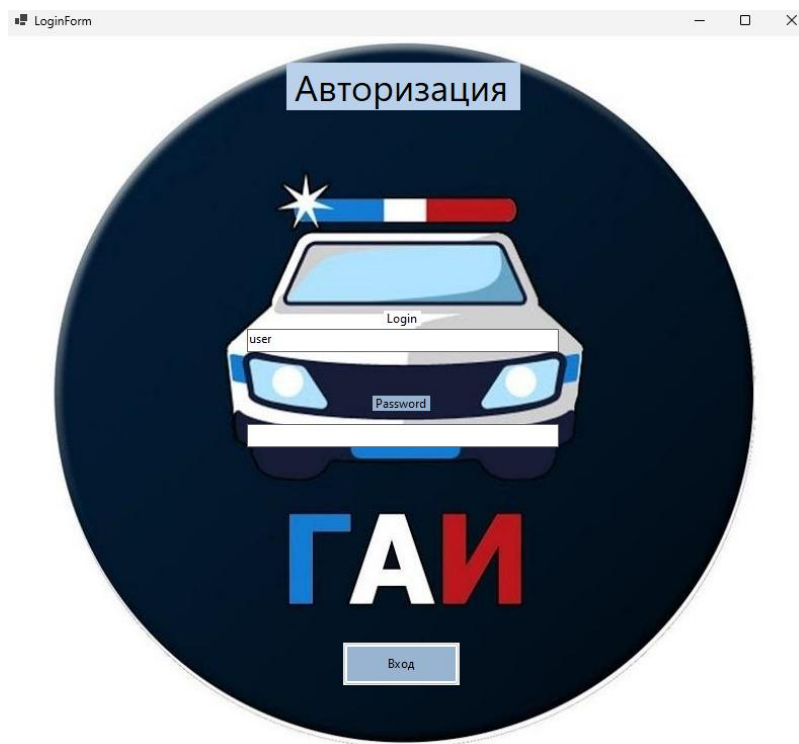


Рисунок 6 – ввод данных от учетной записи пользователей

АИС ГАИ (Администратор)

Меню

☐ Использовать дату

Фамилия Имя Отчество Паспорт

Телефон Адрес СТС Водительское удостоверение

Дата рождения
21 декабря 2024 г.

Марка Модель Гос. номер Год выпуска авто

СТС Страховое свидетельство ФИО

Фамилия Имя Отчество Паспорт

Телефон Адрес Должность Звание

Дата рождения Выписанные протоколы
21 декабря 2024 г.

Правонарушения Регистрация

Номер протокола Нарушение Гос. номер Статус штрафа

Штраф Краткое описание Дата нарушения
21 декабря 2024 г.

ФИО сотрудника ГАИ ФИО водителя

Фамилия	Имя	Отчество	ДатаРождения	НомерПаспорта	Телефон	Адрес
Иванов	Иван	Иванович	01.01.1980	1234567890	+79001234567	ул. Ленина, 1
Петров	Петр	Петрович	02.02.1985	2345678901	+79002345678	ул. Пушкина, 2
Сидоров	Сидор	Сидорович	03.03.1990	3456789012	+79003456789	ул. Чехова, 3

Поиск водителей: добавить изменить удалить Обновить

Марка	Модель	Год	НомернойЗнак	Страхование	СвидетельствоО	Владелец
Toyota	Camry	2020	A123BC45	1234567890	1234567890	Иванов И. И.
Ford	mondeo	2018	B456G D78	2345678901	2345678901	Петров П. П.
Honda	Civic	2019	C789EJ90	3456789012	3456789012	Сидоров С. С.

Поиск транспортного средства: добавить изменить удалить Обновить

Фамилия	Имя	Отчество	Должность	Звание	НомерПаспорта	Телефон
Смирнов	Алексей	Петрович	Инспектор	Лейтенант	1234567890	+79001234567
Зинцов	Иван	Алексеевич	Старший инсп...	Старший лейт...	2345678901	+79002345678
Петров	Петр	Иванович	Капитан	Капитан	3456789012	+79003456789

Поиск сотрудников ГАИ: добавить изменить удалить Обновить

НомерПротокол	ДатаНарушения	Нарушение	НомернойЗнак	Штраф	Описание	СтатусШтрафа
001	21.12.2024	Нарушение ск...	A123BC45	500	Превышение с...	не оплачен
002	10.07.2024	Проезд на кра...	B456G D78	1000	Проезд на кра...	оплачен
003	15.08.2024	Парковка в не...	C789EJ90	300	Парковка на т...	не оплачен

Поиск нарушения: добавить изменить удалить Обновить

Рисунок 7 – рабочее окно администратора

Рабочее окно администратора, которое показано на рисунке 7, имеет следующие функции:

1) Меню, в котором мы можем очистить формочки или выйти обратно в окно авторизации, что показано на рис. 8.

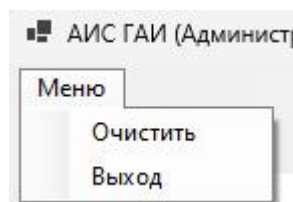


Рис. 8 – Меню в окне Администратора

2) Кнопки поиска Водителей, Авто, Сотрудников ГАИ(ГИБДД), нарушений, а также регистрации, которая показана на рисунке 12. Поиск происходит по любым параметрам из тех элементов, которые можно ввести в клавиатуру, но также мы можем использовать дату для поиска, а можем и не использовать. Данная функция представлена рис. 7, прямо под меню.

Владелец	НомернойЗнак	СвидетельствоО	Страховка	ДатаРегистрации
Иванов И. И.	A123BC45	1234567890	1234567890	21.12.2024
Петров П. П.	B456ГД78	2345678901	2345678901	22.12.2024
Сидоров С. С.	C789ЕЖ90	3456789012	3456789012	23.12.2024

Рис. 9 - Раздел Регистрация

3) Также представлены кнопки добавления, удаления, редактирования (изменить) и обновления.

4) При нажатии на любой элемент из строчки, мы можем перенести элементы из таблицы в поля, которые мы вводим. Реализация данной функции представлена на рис. 10.

Фамилия	Имя	Отчество
Кузнецов	Алексей	Алексеевич
Смирнов	Андрей	Андреевич
Воробьев	Никита	Романович

Рис. 10 – «Копирование» элементов из таблицы в элементы ввода

5) Функция перехода на раздел регистрация или правонарушения, которая представлена на рис. 11.

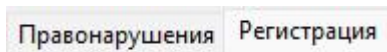


Рис. 11 – переход между разделами

6) При нажатии на любой элемент в строчке из таблицы двойным кликом, информация из таблицы будет перенесена в элементы для ввода. Данная функция представлена рис. 10.

Фамилия	Имя	Отчество	ДатаРождения	НомерПаспорта	Телефон	Адрес
Воробьев	Никитос	Романович	20.06.2004	1724024576	+79004750443	Муром, ул. Во...

Рис. 12 – работа кнопки поиск

Фамилия	Имя	Отчество	ДатаРождения	НомерПаспорта	Телефон	Адрес
Иванов	Иван	Иванович	01.01.1980	1234567890	+79001234567	ул. Ленина, 1
Петров	Петр	Петрович	02.02.1985	2345678901	+79002345678	ул. Пушкина, 2
Сидоров	Сидор	Сидорович	03.03.1990	3456789012	+79003456789	ул. Чехова, 3

Марка	Модель	Год	НомернойЗнак	Страхование	СвидетельствоО	Владелец
Toyota	Camry	2020	A123BC45	1234567890	1234567890	Иванов И. И.
Ford	mondeo	2018	B456FД78	2345678901	2345678901	Петров П. П.
Honda	Civic	2019	C789EЖ30	3456789012	3456789012	Сидоров С. С.

НомерПротокола	ДатаНарушения	Нарушение	НомернойЗнак	Штраф	Описание	СтатусШтрафа
001	21.12.2024	Нарушение ск...	A123BC45	500	Превышение с...	не оплачен
002	10.07.2024	Проезд на кра...	B456FД78	1000	Проезд на кра...	оплачен

Рисунок 13 – форма пользователя

Форма пользователя, которая показана на рисунке 13 имеет следующие функции:

- 1) Меню, как в окне администратора, в котором мы можем очистить формочки или выйти обратно в окно авторизации, что показано на рис. 8.
- 2) Кнопки поиска Водителей, Авто и нарушений. Поиск происходит по любым параметрам из тех элементов, которые можно ввести в клавиатуру, но также мы можем использовать дату для поиска, а можем и не использовать. Данная функция представлена рис. 7, прямо под меню.
- 3) Кнопка обновления таблицы.
- 4) При нажатии на любой элемент в строчке из таблицы двойным кликом, информация из таблицы будет перенесена в элементы для ввода. Данная функция представлена рис. 14.

Марка	Модель	Гос. номер	Год выпуска авто
ВАЗ	2113	M241XA33	2011

СТС	Страховое свидетельство	ФИО
1724024576	1724024576	Воробьев Н. Р.

Марка	Модель	Год	НомернойЗнак	Страхование	СвидетельствоО	Владелец
Chevrolet	Malibu	2021	D012ЖЗ34	4567890123	4567890123	Кузнецов А. А.
Nissan	Altima	2022	E345IM56	5678901234	5678901234	Смирнов А. А.
ВАЗ	2113	2011	M241XA33	1724024576	1724024576	Воробьев Н. Р.

Поиск транспортного средства Обновить

Рисунок 14 – «Копирование» элементов из таблицы в элементы ввода

3.3 Руководство программиста

3.3.1 Введение

Проект написан на языке программирования C# в среде Visual Studio с использованием базы данных SQLite. Проект включает в себя формы для различных типов пользователей, таких как пользователи(клиенты), администраторы.

3.3.2 Установка и настройка системы

Требования к системе:

- Visual Studio 2019 или новее
- .NET Framework 4.7.2 или новее[5]
- Библиотека SQLite

Установка необходимых инструментов и библиотек:

- Откройте Visual Studio.
- Создайте или откройте проект.
- Установите пакет SQLite через NuGet Package Manager: Install-Package System.Data.SQLite[6]

3.3.3 Структура проекта

Проект состоит из следующих файлов и директорий:

- Form1.cs: Форма для администратора.
- From2.cs: Форма авторизации.
- User.cs: Форма для пользователя.

3.3.4 Описание форм

1) Form1.cs:

- Конструктор Form1: инициализирует форму, устанавливает соединение с базой данных и загружает данные из БД.
- Метод LoadData: открывает соединение с базой данных, выполняет SQL-запрос для получения данных из БД, заполняет DataTable данными и отображает их в DataGridView, также метод состоит из: LoadDrivers(), LoadAuto(), LoadPoliceman(), LoadViolations(), LoadRegistr(). Другие 5 методов открывают соединения с таблицами из БД.
- Метод driversDataGridView_CellDoubleClick: заполняет текстовые поля данными из выбранной строки DataGridView при двойном щелчке.
- Метод searchdriver_Click_1: выполняет поиск водителей по различным критериям, выполняет SQL-запрос для поиска и отображает результаты в DataGridView.
- Метод Adddriver_Click_1: добавляет нового водителя в базу данных, если все поля заполнены и водитель с такими данными не существует, выполняет SQL-запрос для добавления в базу данных водителя и обновляет данные в DataGridView.
- Метод Editdriver_Click_1: обновляет информацию о выбранном водителе в базе данных, если строка выбрана в DataGridView, выполняет SQL-запрос для обновления данных водителя в базе данных и обновляет данные в DataGridView.
- Метод Deletedriver_Click_1: удаляет выбранные записи водителей из базы данных после подтверждения пользователя, выполняет SQL-запрос для удаления из БД и обновляет DataGridView.
- Метод button3_Click_1: выполняет поиск автомобилей по различным критериям и отображает результаты в DataGridView, выполняет SQL-запрос для поиска.
- Метод AddCar_Click_1: добавляет новый автомобиль в базу данных, если все поля заполнены корректно, выполняет SQL-запрос для добавления в базу данных и обновляет данные в DataGridView.

- Метод EditCar_Click_1: обновляет данные о выбранном автомобиле в базе данных, если строка выбрана в DataGridView, выполняет SQL-запрос для обновления данных автомобиля в базе данных и обновляет данные в DataGridView.
- Метод DeleteCar_Click_1: удаляет выбранные автомобили из базы данных после подтверждения пользователя и обновляет данные в DataGridView, выполняет SQL-запрос для удаления из БД и обновляет DataGridView.
- Метод ObnoveAuto_Click_1: перезагружает данные о автомобилях из базы данных и обновляет данные в DataGridView.
- Метод Obnovedrив_Click_1: перезагружает данные о водителях из базы данных и обновляет данные в DataGridView.
- Метод AutoDataGridView_CellDoubleClick: заполняет текстовые поля данными из выбранной строки DataGridView при двойном щелчке.
- Метод SearchPolice_Click: выполняет поиск сотрудников ГАИ по различным критериям и отображает результаты в DataGridView, выполняет SQL-запрос для поиска.
- Метод Addpolice_Click_1: добавляет нового сотрудника ГАИ в базу данных, если все поля заполнены корректно и данные уникальны, выполняет SQL-запрос для добавления в базу данных и обновляет данные в DataGridView.
- Метод EditPolice_Click_1: обновляет данные о выбранном сотруднике ГАИ в базе данных, если строка выбрана в DataGridView, выполняет SQL-запрос для обновления данных сотрудника ГАИ(ГИБДД) в базе данных и обновляет данные в DataGridView.
- Метод DeletePolice_Click_1: удаляет выбранных сотрудников ГАИ из базы данных после подтверждения пользователя, выполняет SQL-запрос для удаления из БД и обновляет DataGridView.
- Метод ObnovePolice_Click_1: перезагружает данные о сотрудниках ГАИ из базы данных и обновляет данные в DataGridView.
- Метод PolicemanDataGridView_CellDoubleClick: заполняет текстовые поля данными из выбранной строки DataGridView при двойном щелчке.

- Метод SearchViol_Click_1: выполняет поиск нарушений по различным критериям и отображает результаты в DataGridView, выполняет SQL-запрос для поиска.
- Метод button5_Click_1: добавляет новое Нарушение в базу данных, если все поля заполнены корректно и данные уникальны, выполняет SQL-запрос для добавления в базу данных и обновляет данные в DataGridView.
- Метод button4_Click_1: обновляет информацию о выбранных нарушениях в базе данных, если строка выбрана в DataGridView, выполняет SQL-запрос для обновления данных нарушений в базе данных и обновляет данные в DataGridView.
- Метод DelViol_Click_1: удаляет выбранные нарушения из базы данных после подтверждения пользователя, выполняет SQL-запрос для удаления из БД и обновляет DataGridView.
- Метод ObnoveViol_Click_1: перезагружает данные о нарушениях из базы данных и обновляет данные в DataGridView.
- Метод violationsDataGridView_CellDoubleClick: заполняет текстовые поля формы данными из выбранной строки DataGridView при двойном щелчке.
- Метод searchRegistr_Click: выполняет поиск данных регистрации по различным критериям и отображает результаты в DataGridView, выполняет SQL-запрос для поиска.
- Метод addRegister_Click: добавляет новую запись регистрации автомобиля в базу данных, все поля заполнены корректно и данные уникальны, выполняет SQL-запрос для добавления в базу данных и обновляет данные в DataGridView.
- Метод EditRegister_Click: обновляет информацию о выбранной записи регистрации в базе данных, если строка выбрана в DataGridView, выполняет SQL-запрос для обновления данных регистраций в БД и обновляет данные в DataGridView.
- Метод ObnoveRegistr_Click: перезагружает данные о регистрациях из базы данных и обновляет данные в DataGridView.

- Метод `RegistrdataGridView_CellDoubleClick`: заполняет текстовые поля формы данными из выбранной строки `DataGridView` при двойном щелчке.
- Метод `DeleteRegistr_Click`: удаляет выбранные записи регистрации из базы данных после подтверждения пользователя, выполняет SQL-запрос для удаления из БД и обновляет `DataGridView`.
- Метод `очиститьToolStripMenuItem_Click`: используется для очистки всех `TextBox` и `ComboBox` на форме, а также всех элементов внутри `TabControl`.
- Метод `выходToolStripMenuItem_Click`: используется для выхода из текущей формы с подтверждением пользователя.

2) Form2.cs:

- Конструктор `GuestForm`: инициализирует форму, устанавливает обработчик события закрытия и символ пароля для текстового поля.
- Метод `button1_Click`: обрабатывает нажатие кнопки для входа, проверяет введенные данные и открывает соответствующую форму или показывает сообщение об ошибке.

3) User.cs:

- Конструктор `Form1`: инициализирует форму, устанавливает соединение с базой данных и загружает данные из БД.
- Метод `LoadData`: открывает соединение с базой данных, выполняет SQL-запрос для получения данных из БД, заполняет `DataTable` данными и отображает их в `DataGridView`, также метод состоит из: `LoadDrivers()`, `LoadAuto()`, `LoadViolations()`. Другие 3 метода открывают соединения с таблицами из БД.
- Метод `driversDataGridView_CellDoubleClick`: заполняет текстовые поля данными из выбранной строки `DataGridView` при двойном щелчке.
- Метод `searchdriver_Click`: выполняет поиск водителей по различным критериям, выполняет SQL-запрос для поиска и отображает результаты в `DataGridView`.
- Метод `Obnovedriv_Click`: перезагружает данные о водителях из базы данных и обновляет данные в `DataGridView`.

- Метод ObnoveAuto_Click: перезагружает данные о автомобилях из базы данных и обновляет данные в DataGridView.
- Метод AutoDataGridView_CellDoubleClick: заполняет текстовые поля данными из выбранной строки DataGridView при двойном щелчке.
- Метод button3_Click: выполняет поиск автомобилей по различным критериям и отображает результаты в DataGridView, выполняет SQL-запрос для поиска.
- Метод ObnoveViol_Click: перезагружает данные о нарушениях из базы данных и обновляет данные в DataGridView.
- Метод violationsDataGridView_CellDoubleClick: заполняет текстовые поля формы данными из выбранной строки DataGridView при двойном щелчке.
- Метод SearchViol_Click_1: выполняет поиск нарушений по различным критериям и отображает результаты в DataGridView, выполняет SQL-запрос для поиска.
- Метод очиститьToolStripMenuItem_Click: используется для очистки всех TextBox и ComboBox на форме, а также всех элементов внутри TabControl.
- Метод выходToolStripMenuItem_Click: используется для выхода из текущей формы с подтверждением пользователя.

4. Тестирование АИС

Тестирование автоматизированной информационной системы (АИС) является неотъемлемой частью процесса разработки, направленной на обеспечение ее корректной работы, надежности и безопасности. При открытии приложения первым шагом для пользователя является окно авторизации, показанное на рисунке 15, которое служит для идентификации и аутентификации пользователей. Это окно играет важную роль в защите данных и управлении доступом к функционалу системы, что делает его критически важным элементом для тестирования.

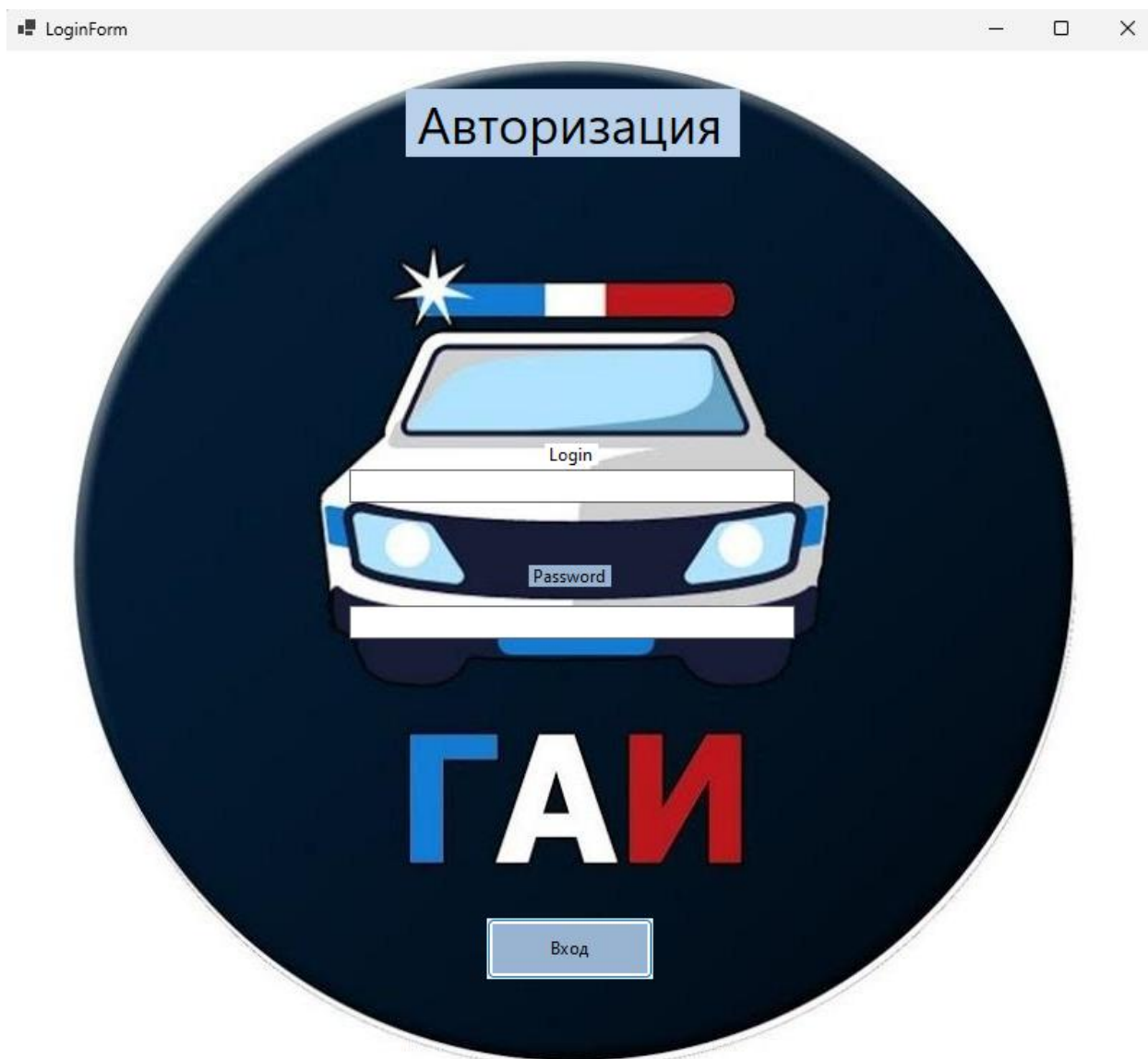


Рисунок 15 – окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, если логин и пароль правильные, то пользователь попадает в рабочее окно, а если логин или пароль, который был введен не корректен, то приложение выдаст ошибку, что логин и пароль не правильный, это показано на рисунке 16.

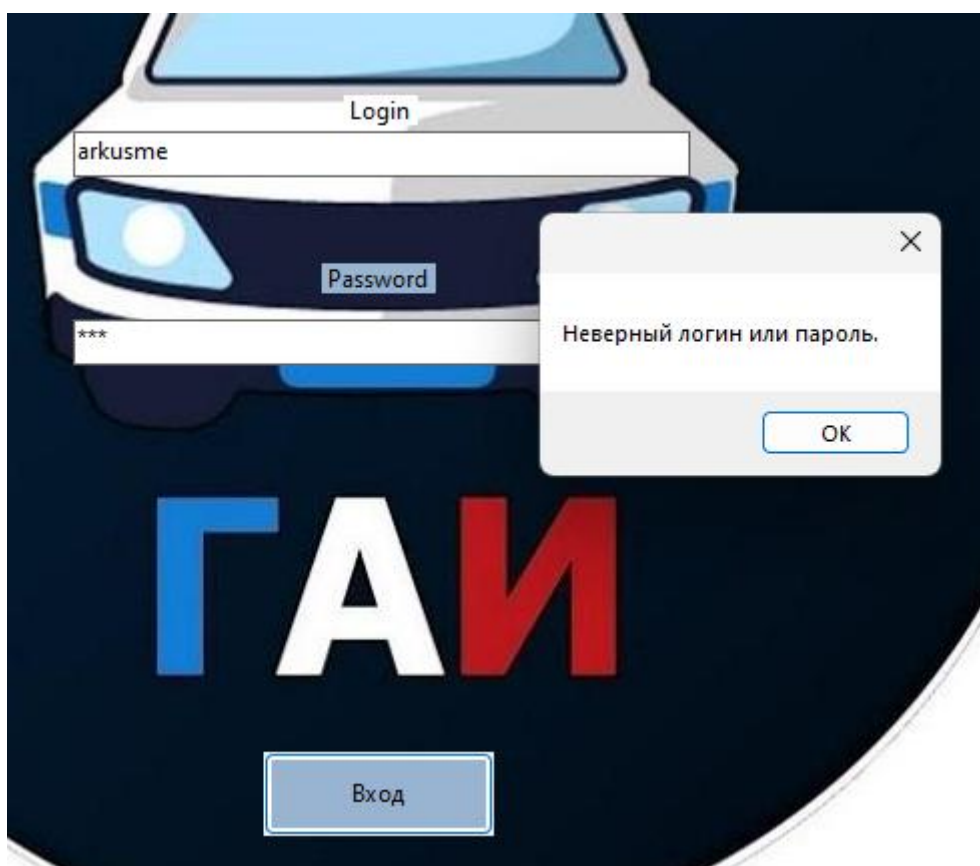


Рисунок 16 – ошибка входа, если логин и/или пароль указан не правильный

Если в таблице Водители, таблице Авто, таблице сотрудники ГАИ(ГИБДД), таблице Нарушения и таблице Регистрация необходимо отредактировать запись или ее удалить, и пользователь не выделит запись, которую необходимо отредактировать или удалить, но нажмет на кнопку «редактировать» или «удалить», то получит следующие сообщения, показанные на рисунках 17, 18, 19, 20, 21, 22, 23, 24, 25, 26.

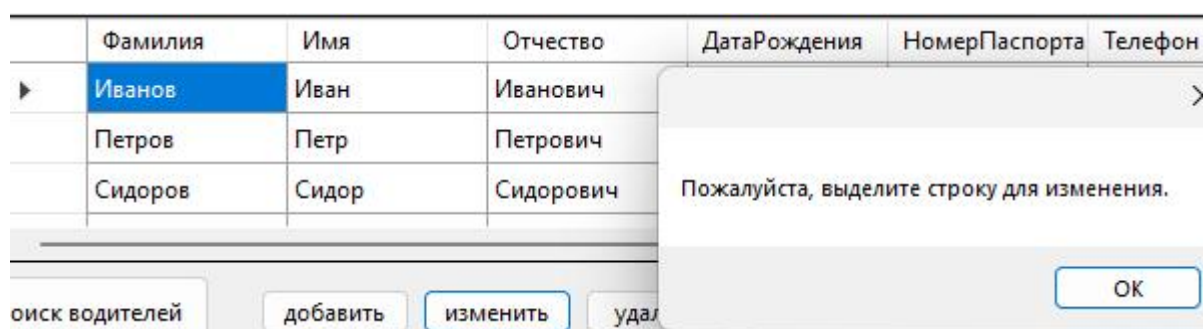


Рисунок 17 – уведомление о том, что не выбрана запись для изменения в таблице Водители

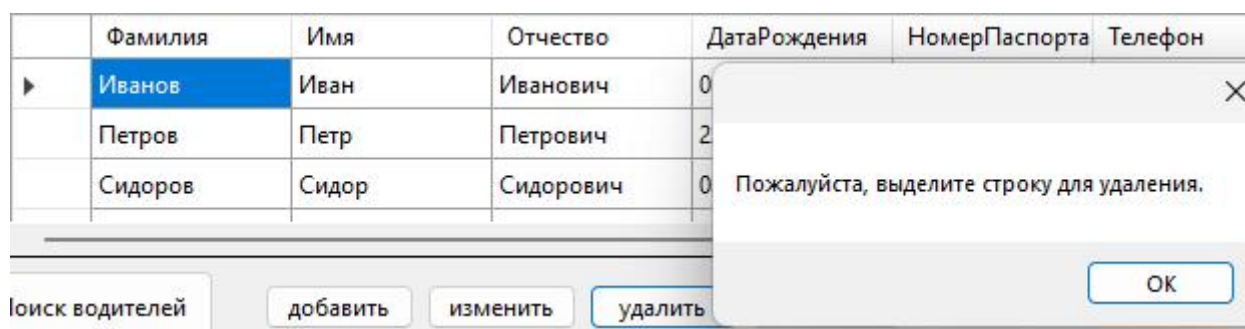


Рисунок 18 – уведомление о том, что не выбрана запись для удаления в таблице
Водители

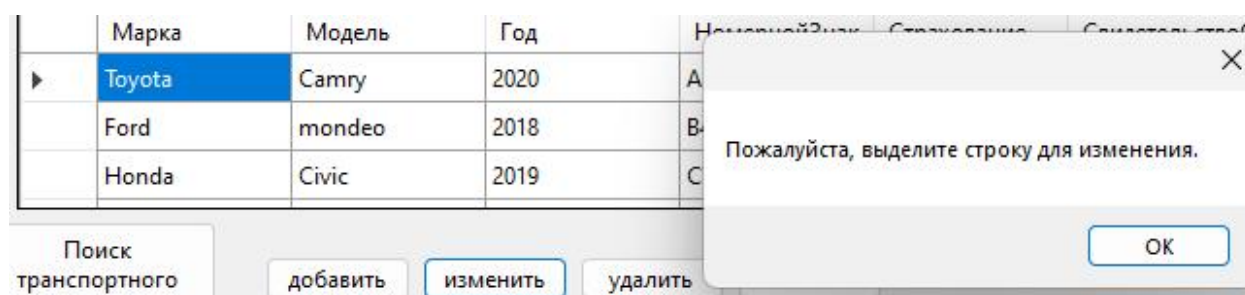


Рисунок 19 – уведомление о том, что не выбрана запись для изменения в таблице
Авто

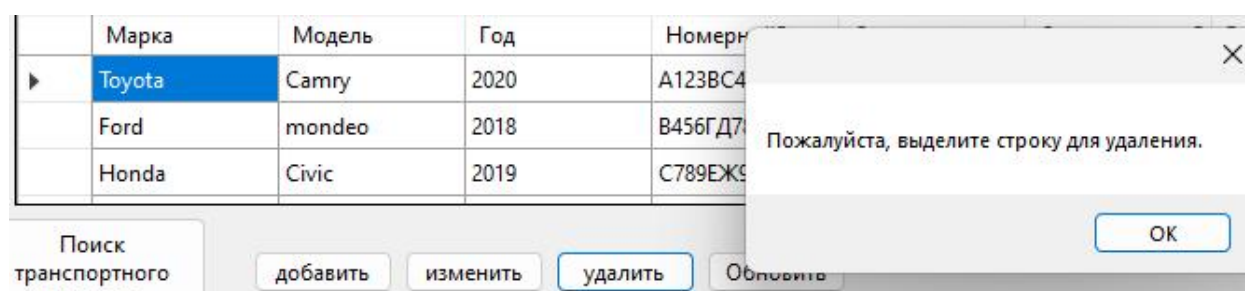


Рисунок 20 – уведомление о том, что не выбрана запись для удаления в таблице
Авто

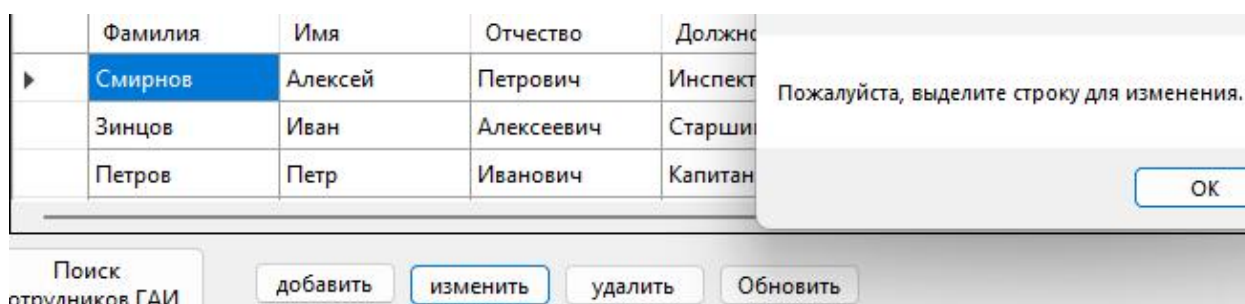


Рисунок 21 – уведомление о том, что не выбрана запись для изменения в таблице
сотрудники ГАИ

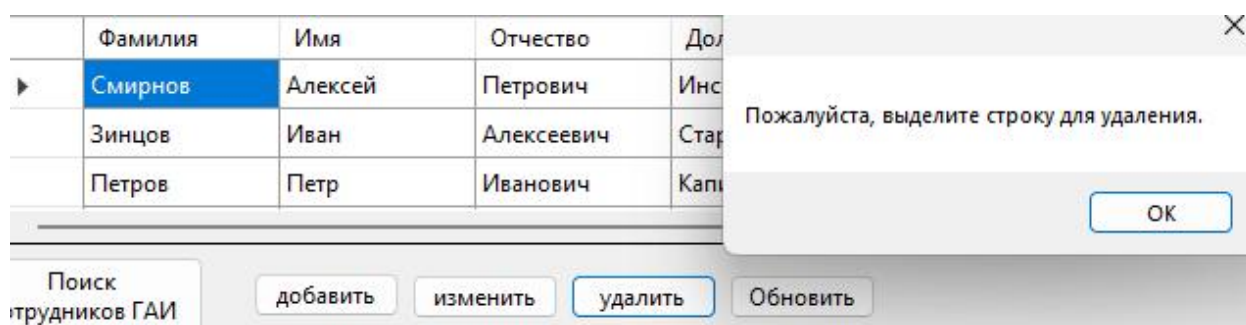


Рисунок 22 – уведомление о том, что не выбрана запись для удаления в таблице сотрудники ГАИ

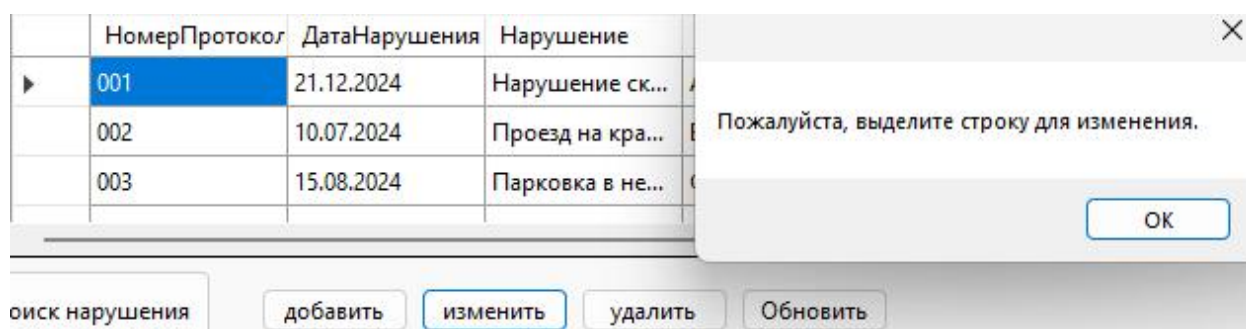


Рисунок 23 – уведомление о том, что не выбрана запись для изменения в таблице Нарушения

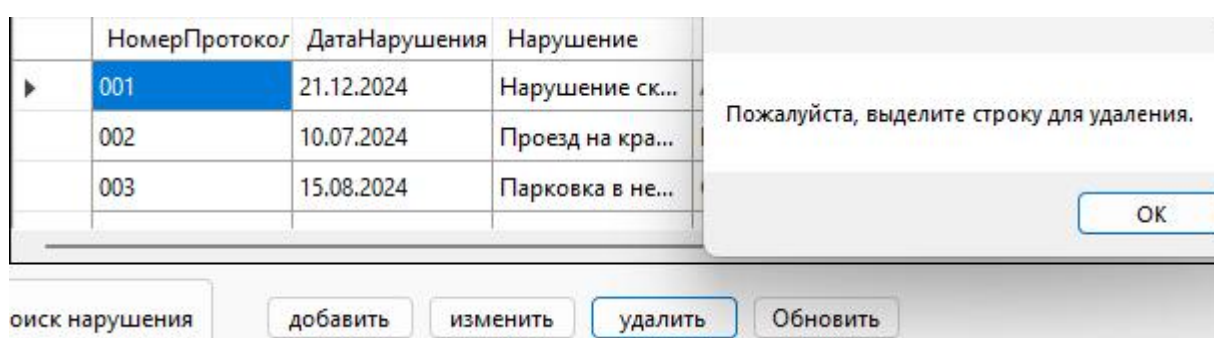


Рисунок 24 – уведомление о том, что не выбрана запись для удаления в таблице Нарушения

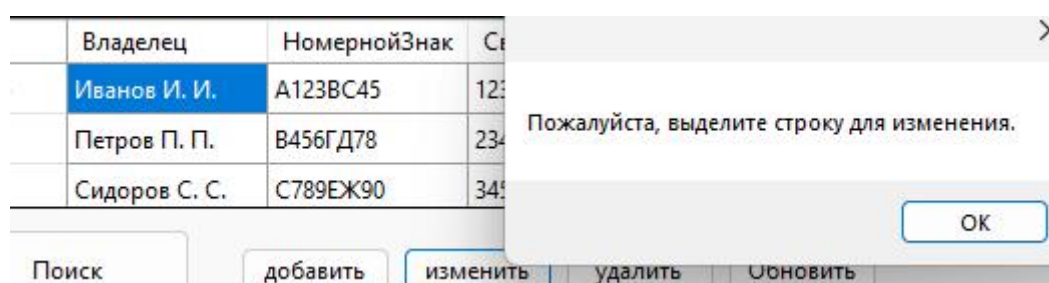


Рисунок 25 – уведомление о том, что не выбрана запись для изменения в таблице Регистрация

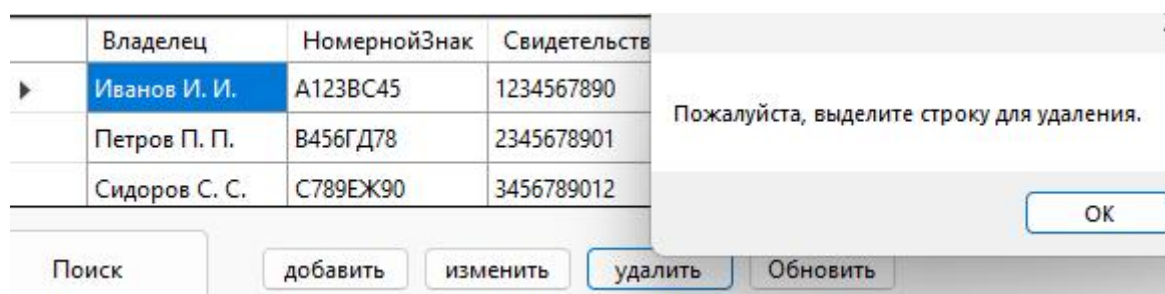


Рисунок 26 – уведомление о том, что не выбрана запись для удаления в таблице Регистрация

Когда администратор удаляет запись, появляется окно подтверждения удаления записи, показанное на рисунках 27, 28.

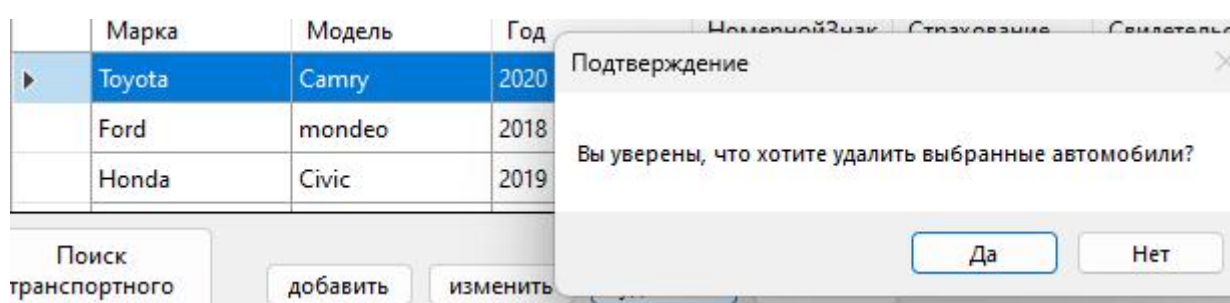


Рисунок 27 – окно подтверждения удаления записи в таблице Авто

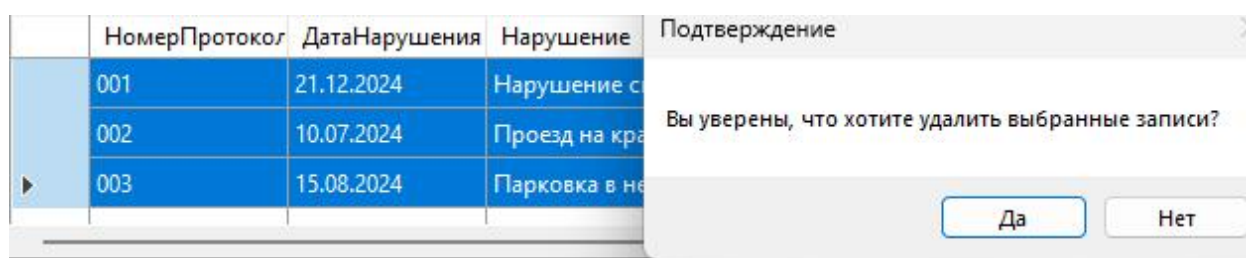


Рисунок 28 – окно подтверждения удаления записи в таблице Нарушения

После нажатия на кнопку из меню «Выход», пользователь получит уведомление, точно он хочет выйти из учетной записи или нет, данное окно показано на рисунке 29.

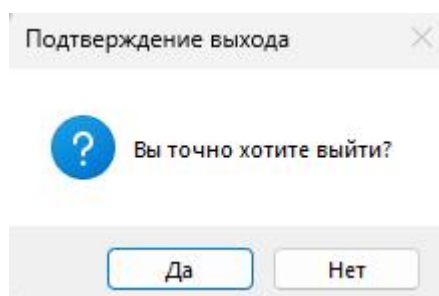


Рисунок 29 – окно подтверждения выхода из учтённой записи

Когда пользователь добавляет записи о нарушениях, то может получить следующее уведомление, показанное на рисунке 30, которое означает, что такой номер протокола уже существует.

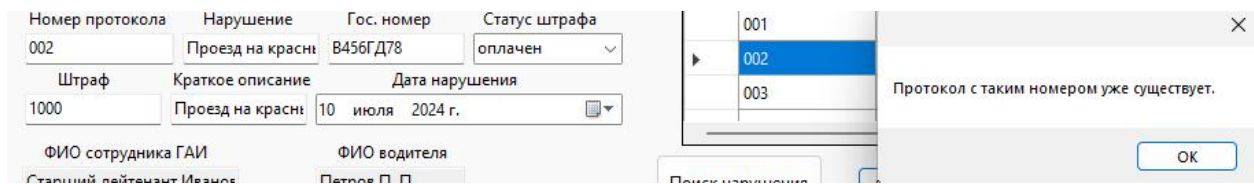


Рисунок 30 – уведомление о том, что такой протокол уже существует

Когда пользователь добавляет записи о водителях, то может получить следующее уведомление, показанное на рисунке 31, которое означает, что такой водитель уже существует, потому что ФИО и серия, и номер паспорта совпадают.

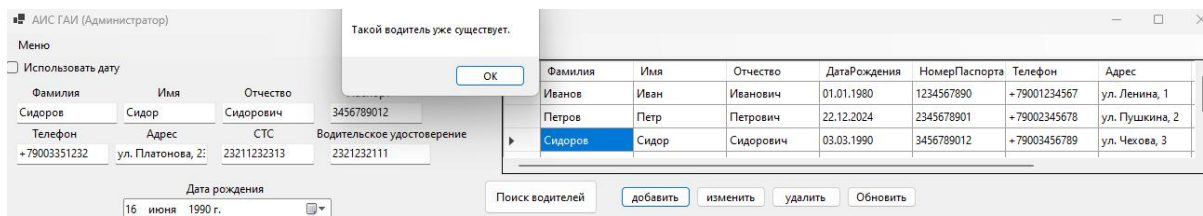


Рис. 31 – уведомление о том, что такой водитель уже существует

Точно также у нас и с сотрудниками ГИБДД(ГАИ).

Когда пользователь добавляет записи об авто, то может получить следующее уведомление, показанное на рисунке 32, которое означает, что такой автомобиль уже существует.

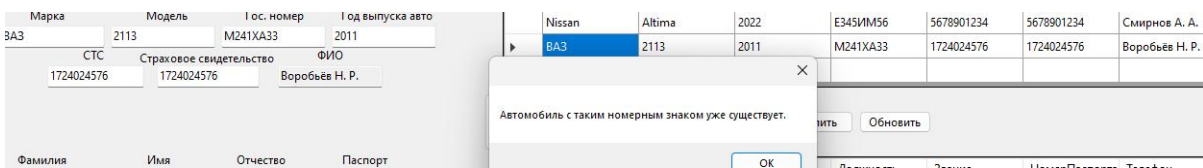


Рис. 32 - уведомление о том, что такой автомобиль уже существует

Когда пользователь добавляет записи о регистрации, то может получить следующее уведомление, показанное на рисунке 33, которое означает, что такой автомобиль уже зарегистрировали.

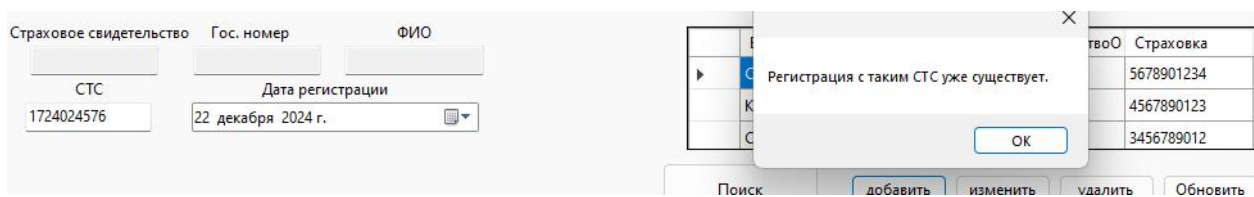


Рис. 33 - уведомление о том, что такой автомобиль уже зарегистрировали

Также, когда пользователь добавляет в таблицу Водители, Авто, Сотрудник ГАИ, Нарушения и Регистрация, он получает следующее уведомление показанное на рисунке 34, 35, 36, 37, 38.

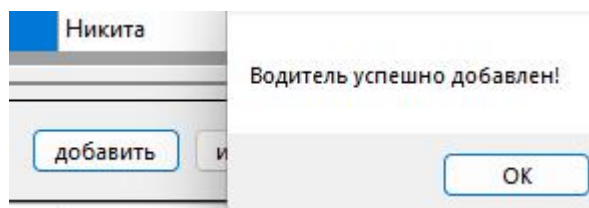


Рис. 34 – уведомление о том, что водитель добавлен успешно

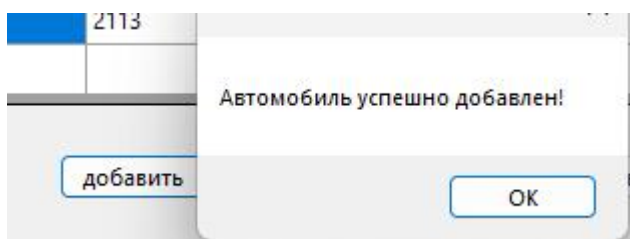


Рис. 35 – уведомление о том, что автомобиль добавлен успешно

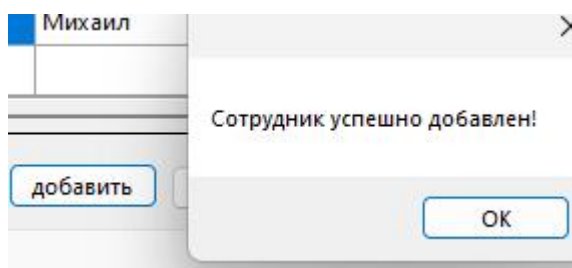


Рис. 36 – уведомление о том, что сотрудник ГАИ(ГИБДД) добавлен успешно

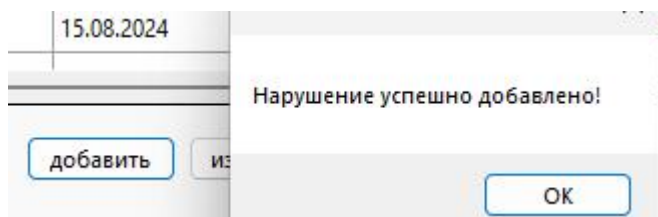


Рис. 37 – уведомление о том, что нарушение добавлено успешно

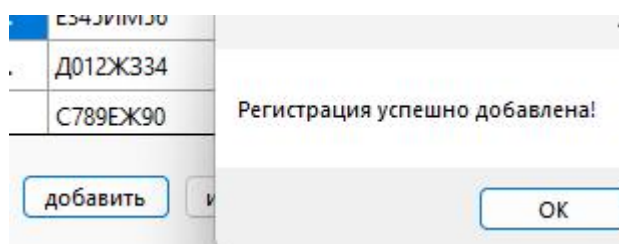


Рис. 38 – уведомление о том, что регистрация добавлена успешно

Заключение

В рамках данной курсовой работы была разработана автоматизированная информационная система (АИС) ГИБДД с использованием SQLite и языка программирования C# в среде Windows Forms. Рассмотрены основные компоненты системы, их назначение и принципы работы.

В процессе разработки был выполнен анализ, который показал, что АИС ГИБДД играет ключевую роль в автоматизации процессов регистрации автомобилей, учета нарушений ПДД и контроля транспортных средств и водителей.

					МИВУ 10.03.01-03ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		47

Применение SQLite в качестве СУБД обеспечило легкость интеграции и достаточную производительность для решения поставленных задач. Использование языка C# и платформы WindowsForms позволило создать функциональное и визуально приятное приложение.

Проведенное тестирование подтвердило работоспособность и надежность разработанной системы. Основные функции, такие как регистрация ТС, введение учета нарушения ПДД, хранение в базе данных водителей и ТС, показали стабильные результаты.

Также хочется отметить, что система обладает большим потенциалом для дальнейшего развития и усовершенствования, включая интеграцию с другими информационными системами и использование новых технологий.

Таким образом, АИС ГИБДД представляет собой важный инструмент для повышения эффективности работы Государственной инспекции безопасности дорожного движения и улучшения качества предоставляемых услуг. Дальнейшее развитие системы, ее адаптация к изменяющимся условиям и внедрение новых технологий позволят еще более повысить ее значимость и эффективность.

Список литературы

1. Гарсия-Молина, Х., Ульман, Дж. Д., Видом, Дж. "Основы систем баз данных." – Москва: Вильямс, 2020. – 336с.
2. Грин, М. "Пособие по SQLite: основы и практическое применение." – СПб.: БХВ-Петербург, 2018. – 369с.
3. Кортес, Р., Барчулевич, Е. В., Артемьев, А. А. "Базы данных. Теория и практика." – Москва: Научный мир, 2019. – 459с.
4. Кауфман, О. "SQLite для профессионалов." – СПб.: Питер, 2020. – 576с.

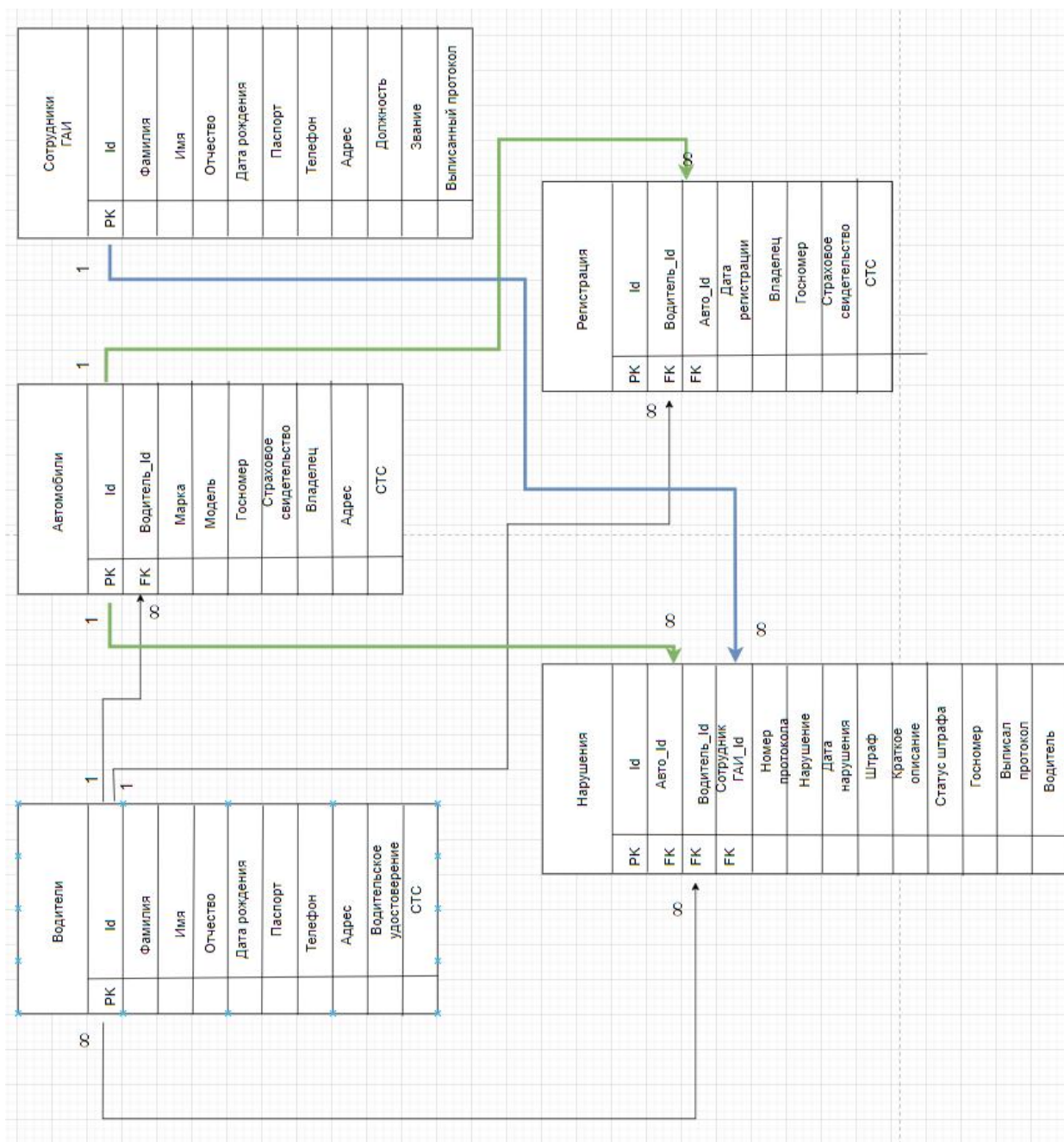
5. Троелсен, Э., Джепперсен, Ф. "Язык программирования C# 8.0 и платформы .NET Core 3.0." – СПб.: Питер, 2020. – 819с.

6. Хортон, М. "C# и .NET: руководство для начинающих." – СПб.: Символ-Плюс, 2019. – 413с.

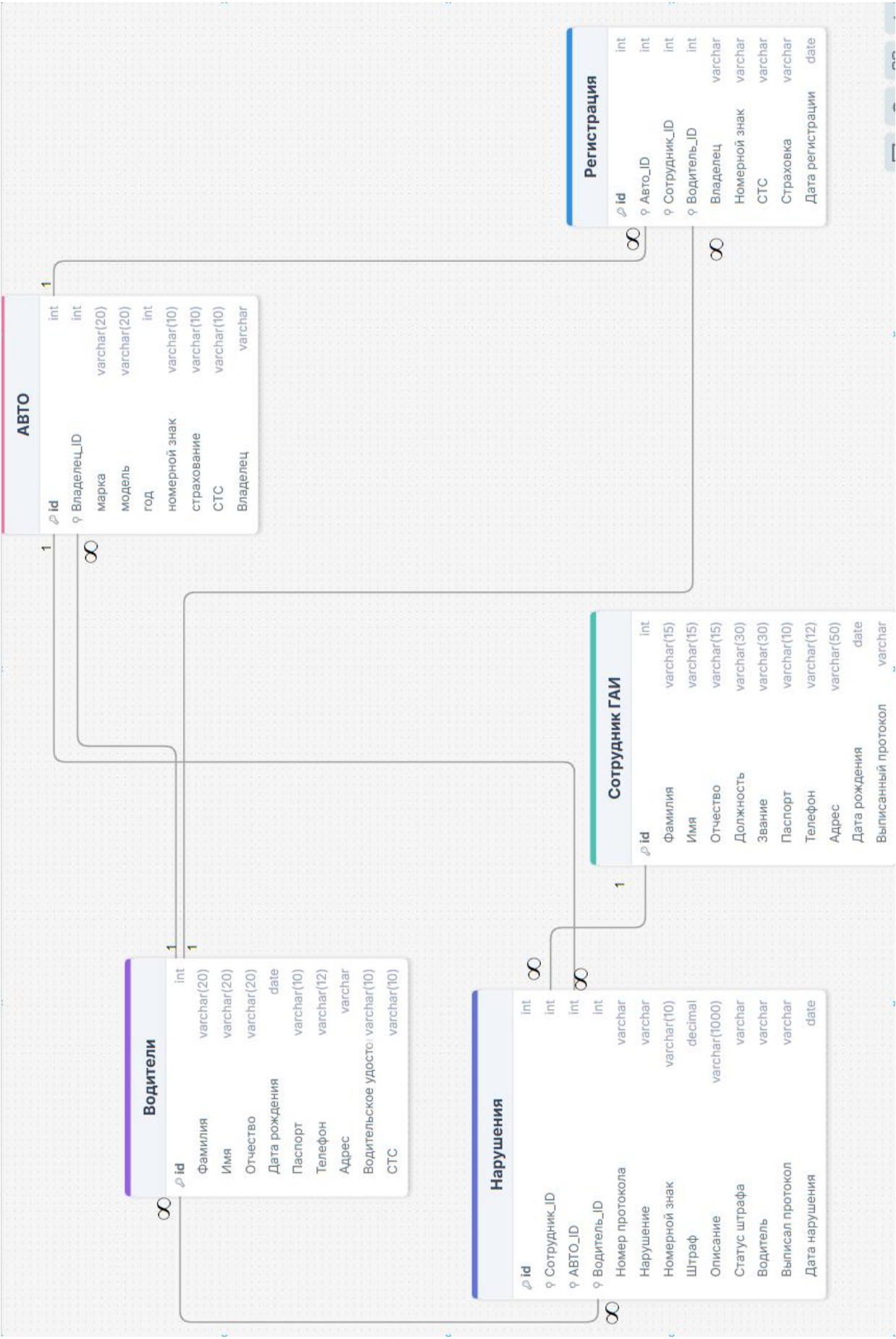
Приложение 1. Модели данных

1. Концептуальная модель

					МИВУ 10.03.01-03ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		49



3. Физическая модель



Приложение 2. Текст программы

Вся программа расположена на Git: [Arkusme/WinFormsApp5 at gai_lastversionAA](https://github.com/Arkusme/WinFormsApp5_at_gai_lastversionAA)



Рисунок 39 – окно авторизации

АИС ГАИ (Администратор)

Меню

☐ Искать по дате

Фамилия

Имя

Отчество

Паспорт

Телефон

Адрес

СТС

Водительское удостоверение

Дата рождения

22 декабря 2024 г.

Марка

Модель

Гос. номер

Год выпуска авто

СТС

Страховое свидетельство

ФИО

Фамилия

Имя

Отчество

Паспорт

Телефон

Адрес

Должность

Звание

Дата рождения

22 декабря 2024 г.

Выписанные протоколы

Правонарушения

Регистрация

Номер протокола

Нарушение

Гос. номер

Статус штрафа

Штраф

Краткое описание

Дата нарушения

22 декабря 2024 г.

ФИО сотрудника ГАИ

ФИО водителя

Поиск водителей

добавить изменить удалить Обновить

Фамилия	Имя	Отчество	ДатаРождения	НомерПаспорта	Телефон	Адрес
▶ Иванов	Иван	Иванович	01.01.1980	1234567890	+79001234567	ул. Ленина, 1
Петров	Петр	Петрович	22.12.2024	2345678901	+79002345678	ул. Пушкина, 2
Сидоров	Сидор	Сидорович	03.03.1990	3456789012	+79003456789	ул. Чехова, 3

Поиск транспортных средства

добавить изменить удалить Обновить

Марка	Модель	Год	НомернойЗнак	Страхование	СвидетельствоО	Владелец
▶ Toyota	Camry	2020	A123BC45	1234567890	1234567890	Иванов И. И.
Ford	mondeo	2018	B456GД78	2345678901	2345678901	Петров П. П.
Honda	Civic	2019	C789EЖ90	3456789012	3456789012	Сидоров С. С.

Поиск сотрудников ГАИ

добавить изменить удалить Обновить

Фамилия	Имя	Отчество	Должность	Звание	НомерПаспорта	Телефон
▶ Смирнов	Алексей	Петрович	Инспектор	Лейтенант	1234567890	+79001234567
Зинцов	Иван	Алексеевич	Старший инсп...	Старший лейт...	2345678901	+79002345678
Петров	Петр	Иванович	Капитан	Капитан	3456789012	+79003456789

Поиск нарушений

добавить изменить удалить Обновить

НомерПротокол	ДатаНарушения	Нарушение	НомернойЗнак	Штраф	Описание	СтатусШтрафа
▶ 001	21.12.2024	Нарушение с...	A123BC45	500	Превышение с...	не оплачен
002	10.07.2024	Презд на кра...	B456ГД78	1000	Презд на кра...	оплачен
003	15.08.2024	Парковка в не...	C789EЖ90	300	Парковка на т...	не оплачен

User

Меню

☐ Искать по дате

Фамилия

Имя

Отчество

Паспорт

Телефон

Адрес

СТС

Водительское удостоверение

Дата рождения

22 декабря 2024 г.

Марка

Модель

Гос. номер

Год выпуска авто

СТС

Страховое свидетельство

ФИО

Поиск водителей

Обновить

Фамилия	Имя	Отчество	ДатаРождения	НомерПаспорта	Телефон	Адрес
Иванов	Иван	Иванович	01.01.1980	1234567890	+79001234567	ул. Ленина, 1
Петров	Петр	Петрович	22.12.2024	2345678901	+79002345678	ул. Пушкина, 2
Сидоров	Сидор	Сидорович	03.03.1990	3456789012	+79003456789	ул. Чехова, 3

Поиск транспортных средств

Обновить

Марка	Модель	Год	НомернойЗнак	Страхование	СвидетельствоО	Владелец
Toyota	Camry	2020	A123BC45	1234567890	1234567890	Иванов И. И.
Ford	mondeo	2018	B456F.D78	2345678901	2345678901	Петров П. П.
Honda	Civic	2019	C789EJK30	3456789012	3456789012	Сидоров С. С.

Правонарушения

Номер протокола

Нарушение

Гос. номер

Статус штрафа

Штраф

Краткое описание

Дата нарушения

ФИО сотрудника ГАИ

ФИО водителя

НомерПротокол

ДатаНарушения

Нарушение

НомернойЗнак

Штраф

Описание

СтатусШтрафа

001	21.12.2024	Нарушение сс...	A123BC45	500	Превышение с...	не оплачен
002	10.07.2024	Проезд на кра...	B456F.D78	1000	Проезд на кра...	оплачен

Поиск нарушения

Обновить

Правонарушения

Регистрация

сотрудников | АИИ

Страховое свидетельство

Гос. номер

ФИО

СТС

Дата регистрации

22 декабря 2024 г.

	Владелец	НомернойЗнак	СвидетельствоО	Страховка	ДатаРегистрации
▶	Смирнов А. А.	E345IM56	5678901234	5678901234	05.05.2023
	Кузнецов А. А.	D012ЖЗ34	4567890123	4567890123	10.04.2023
	Сидоров С. С.	C789ЕЖ90	3456789012	3456789012	25.03.2023

Поиск

добавить

изменить

удалить

Обновить

Правонарушения

Регистрация

сотрудников ГАИ

добавить

изменить

удалить

Обновить

Номер протокола

Нарушение

Гос. номер

Статус штрафа

Штраф

Краткое описание

Дата нарушения

ФИО сотрудника ГАИ

ФИО водителя

22 декабря 2024 г.

Поиск нарушения

добавить

изменить

удалить

Обновить

	НомерПротокол	ДатаНарушения	Нарушение	НомернойЗнак	Штраф	Описание	СтатусШтрафа
▶	001	21.12.2024	Нарушение ск...	A123BC45	500	Превышение с...	не оплачен
	002	10.07.2024	Проезд на кра...	B456ГД78	1000	Проезд на кра...	оплачен
	003	15.08.2024	Парковка в не...	C789ЕЖ90	300	Парковка на т...	не оплачен

Рисунок 43 – окно раздела Правонарушения