Краевое государственное бюджетное профессиональное

образовательное учреждение

«Барнаульский государственный педагогический колледж»

Курсовая работа

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ “АПЕНДА АВТОМОБИЛЕЙ”

Основная профессиональная образовательная программа по специальности

Информационные системы и программирование

Выполнил: Гуляев П.С.

Студент 403 группы

Руководитель: Лукьянова Н.В.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись руководителя

Барнаул 2023

Оглавление

[Введение……………………………………………………………………...2](#_Toc100877309)

[Глава 1. Теоретические основыразработки….……………………………..4](#_Toc100877310)

[1.1. Анализ предметной области…….…………………………………..4](#_Toc100877311)

[1.2. Основные задачи при создании программного модуля….………..7](#_Toc100877312)

[1.3. Требования к программному обеспечению……….………………...7](#_Toc100877313)

[1.4. Выбор инструментов для разработки программного модуля….…...8](#_Toc100877314)

[1.4.1. Выбор средств разработки и проектирования СУБД…….……...8](#_Toc100877315)

[1.4.2. Выбор средств разработки программного модуля……….…….15](#_Toc100877316)

[Глава 2. Разработка программного модуля…………….………………….18](#_Toc100877317)

[2.1. Создание базы данных…………………….…………………………....18](#_Toc100877318)

[2.2. Создание программного приложения……………….………………...20](#_Toc100877319)

[Заключение…………………………….…………………………………….28](#_Toc100877320)

[Список использованных источников………………….…………………...29](#_Toc100877321)

Введение

В сегодняшнем быстро меняющемся мире можно наблюдать, как развивается технологический прогресс. Появляются новые виды инструментов, электроники, машин и транспортных средств. Наше общество превратилось в систему потребления, которому нужно сейчас в определенное время получить что-то и определенный вид продукта. Каждый человек стремится взять что ему хочется и быстро, для себя лично, для общего пользования или для бизнеса или же наоборот предоставить кому-то что-то.

На данный момент система договоров широко развита, но с появлением все новых и новых видов орудия труда или какие-то вещей, будут и появляться все новые договорные отношения.

Если не так давно можно было для личных нужд взять коня в качестве транспортного средства или орудия труда, то ограничивались или устным при свидетелях договором (иногда и письменным), то сейчас это целая процедура прописанная подробно и дающая каждой стороне (арендодателю и получателю) огромный список обязательств и требований.

Сегодня транспортное средство стало неотъемлемой частью жизни людей, а не роскошью, и для какой цели берется то или иное транспортное средство, которое может в полной мере удовлетворить специфическим требованиям желающего арендовать физического или юридического лица.

Данное направление, аренда транспортных средств, развито и развивается очень быстро и особенно в нашей стране, так как раньше общество искусственно сдерживалось, но в изменившейся политической и социальной среде быстро заполняется. Сейчас можно взять в аренду от маленького скутера до целых железнодорожных составов, пароходов, круизных лайнеров и это стало обыденностью жизни.

В будущем аренда транспортных средств еще больше войдет в нашу повседневную жизнь и с появлением все более совершенных и новых видов транспорта приобретет глобальный характер.

Будущее не за покупкой и владением транспортным средством, а за его арендой. Поэтому система договоров се более и более будет совершенствоваться и подстраиваться под специфику той или иной машины.

**Проблема исследования** заключается в поиске методов и средств для разработки программного модуля “Аренда автомобиля”.

**Объект исследования:** процесс разработки программного модуля аренда автомобиля.

**Предмет исследования:** объектно-ориентированное программирование как средство разработки программного модуля аренда автомобиля.

**Цель исследования:** разработать программный модуль “ Аренда автомобиля ” средствами объектно-ориентированного программирования.

**Задачи:**

1. Провести анализ предметной области данной проблемы;
2. Ознакомиться с требованиями к программным модулям;
3. Разработать программный модуль выдачи аренда автомобиля;
4. Проверить готовый продукт.

**Методы исследования:**

* Теоретический;
* Эмпирический;
* Исследовательский.

Глава 1. Теоретические основы разработки

## Анализ предметной области

В рамках данной курсовой работы рассматривается предметная область «Прокат автомобиля». Она обоснована развитием коммерческих организаций по прокату автомобилей, в которых необходим компьютеризированный учет имеющихся машин. Эта информация обширна и разрознена. Чтобы вести учет всех автомобилей, которые можно взять в прокат, в организации имеется потребность в структурировании данных об автомобилях. Отсутствие такой возможности приводит к проблеме утери данных и большим временным затратам на выборку данных. Нас интересует работа авто проката. Углубимся в структуру работы авто проката. Прокат автомобилей (или аренда) автомобилей - весьма распространенная и востребованная услуга. В любом городе работает множество авто прокатных контор от крупных до самых маленьких. Процедура аренды автомобиля проста. Служащему арендной компании надо предъявить паспорт и водительское удостоверение (международное). К моменту заключения договора удостоверение должно быть действительно не менее 2 лет. В некоторых странах обслуживание клиента производится только при наличии кредитной карты, в большинстве же стран оплата за услуги проводится наличными, но необходимо внести небольшой залог. В стоимость арендной платы должно входить следующее: неограниченный пробег автомобиля; доставка клиенту автомобиля в черте города; ремонт или замена автомобиля в случае технической неисправности, кроме повреждения покрышек и ветрового стекла; полная страховка на случай ДТП, произошедшего не по вине клиента; страховка, покрывающая ущерб, нанесенный автомобилю в ДТП по вине клиента, сверх определенной суммы (но если на момент ДТП водитель находился в состоянии алкогольного опьянения, страховка не выплачивается); страховка пассажиров (кроме водителя) от несчастных случаев (себя водитель может застраховать за дополнительную плату); налоги. Обычно машину доставляют с полным баком, но и вернуть в автопрокатную контору ее нужно также с полным баком. Аренду автомобиля в месте отдыха можно заказать еще во время покупки тура в своем агентстве, включив ее в пакет услуг. Многие агентства предлагают это своим клиентам, так как заказ аренды машины в своем агентстве облегчает туристу проблемы по прочтению договора на аренду на иностранном языке, гарантирует предоставление качественных и понятных услуг договором на покупку тура в целом. Автомобили застрахованы от всех рисков на условиях КАСКО и ОСАГО. В случае ДТП ответственность Клиента составляет величину залога - 200$, весь остальной ущерб, нанесённый автомобилю, покрывает страховая компания. Процесс оформления аренды автомобиля прост и понятен. От Вас требуется лишь минимальный комплект документов (паспорт и водительское удостоверение) для оформления документов аренды.

Нас интересует работа авто проката.

Углубимся в структуру работы авто проката. Прокат автомобилей (или аренда) автомобилей - весьма распространенная и востребованная услуга. В любом городе работает множество авто прокатных контор - от крупных до самых маленьких. Процедура аренды автомобиля проста. От клиента требуется, чтобы его возраст был не менее 21 года и не превышал 70 лет. Служащему арендной компании надо предъявить паспорт и водительское удостоверение (международное). К моменту заключения договора удостоверение должно быть действительно не менее 2 лет. В некоторых странах обслуживание клиента производится только при наличии кредитной карты, в большинстве же стран оплата за услуги проводится наличными, но необходимо внести небольшой залог. В стоимость арендной платы должно входить следующее:

* - неограниченный пробег автомобиля;
* - доставка клиенту автомобиля в черте города;
* - ремонт или замена автомобиля в случае технической неисправности, кроме повреждения покрышек и ветрового стекла;
* - полная страховка на случай ДТП, произошедшего не по вине клиента;
* - страховка, покрывающая ущерб, нанесенный автомобилю в ДТП по вине клиента, сверх определенной суммы (но если на момент ДТП водитель находился в состоянии алкогольного опьянения, страховка не выплачивается);
* - страховка пассажиров (кроме водителя) от несчастных случаев (себя водитель может застраховать за дополнительную плату);
* - налоги. аренда прокат автомобиль инфологический моделирование

Обычно машину доставляют с полным баком, но и вернуть в авто прокатную контору ее нужно также с полным баком. Аренду автомобиля в месте отдыха можно заказать еще во время покупки тура в своем агентстве, включив ее в пакет услуг. Автомобили застрахованы от всех рисков на условиях КАСКО и ОСАГО. В случае ДТП ответственность Клиента составляет величину залога - 200$, весь остальной ущерб, нанесённый автомобилю, покрывает страховая компания

## Основные задачи при создании программного модуля

1. Рассмотреть существующие программные модули, аналогичные программному модулю «Аренда автомобиля»;

1. Разработать программный модуль «Аренда автомобиля», реализующий автоматизацию для коммерческих организаций по прокату автомобилей;
2. Спроектировать и запрограммировать пользовательский интерфейс программного модуля «Аренда автомобиля»;
3. Сформировать отчёт о проделанной работе.

## **1.3. Требования к программному обеспечению**

Созданный программный модуль должен быть прост в освоении, чтобы любой пользователь с минимальной подготовкой смог им воспользоваться. Ввод данных должен сопровождаться контролем, чтобы вероятность ошибки пользователя была минимальной.

Приступая к созданию собственного программного модуля, были изучены существующие программные модули, чтобы проверить, облегчают ли они работу сотрудников коммерческих организаций и помогают ли наиболее эффективно справляться с поставленными задачами.

Программа должно соответствовать следующим критериям:

* Наличие дружественного и понятного интерфейса.
* Возможность получения информации о пользователях.
* Работоспособность программы после удаления, редактирования и добавления данных о пользователях в базе данных.

Программа должна быть отказоустойчива. Необходимо создать систему резервного копирования для защиты информации от сбоев и других критических ситуаций.

Минимальная аппаратная конфигурация, необходимая для работы программного продукта, перечислена ниже:

* Windows 10;
* Net Framework 4.7.2;
* Microsoft Access;
* Visual Studio 2022

## 1.4. Выбор инструментов для разработки программного модуля

### 1.4.1. **Выбор средств разработки и проектирования СУБД**

Исходя из поставленной задачи в процессе исследования необходимо разработать приложение способное взаимодействовать с базой данных.

Базами данных называют электронные хранилища информации, доступ к которым осуществляется с помощью одного или нескольких компьютеров. Обычно БД создается для хранения и доступа к данным, содержащим сведенья о некоторой предметной области, то есть некоторой области человеческой деятельности или области реального мира.

Любая база данных может быть разделена на три составляющих уровня:

Первый уровень (внутренний) – определяет способы хранение данных на физических носителях. Этот уровень для конечного пользователя не доступен для непосредственной модификации и определяется типом базы данных.

Второй уровень (внешний) – это уровень, который определяет представление хранимых данных для конечного пользователя. Этот уровень определяет интерфейс СУБД (система управления базами данных). Именно на этом уровне пользователю предоставляется доступ к функциям СУБД для непосредственного изменения хранимых данных.

Третий уровень (концептуальный) – этот уровень является переходным между двумя вышеописанными. Этот уровень представляет собой обобщенное представление хранимых данных для множества пользователей.

Система управления базами данных (СУБД) – это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования базы данных несколькими пользователями.

Обычно для обеспечения автоматизации обработки, хранимой в базе данных информации, создаются, так называемые приложения баз данных, представляющие собой программу или комплекс программ, которые позволяют обрабатывать информацию для конкретной прикладной задачи. Такие приложения могут быть созданы как в среде самой СУБД, так и вне ее с помощью систем программирования, использующих средства доступа к базам данных, например Delphi, C++ Builder, Visual C++, Visual Basic и др. Приложения, созданные в среде самой СУБД называют приложениями СУБД, а приложения созданные вне этой среды – внешними приложениями.

Для работы с базами данных часто хватает и приложений СУБД, однако, в большинстве случаев требуется создать приложение баз данных для неквалифицированных пользователей или обеспечить такую функциональность, которую в состоянии обеспечить лишь мощные системы программирования, подобные тем, которые были указаны выше.

Кроме того, в зависимости от расположения отдельных частей СУБД различают локальные и сетевые СУБД.

Все части локальной СУБД размещаются на компьютере пользователя базы данных. Чтобы с одной и той же базой данных одновременно могло работать несколько пользователей, каждый пользовательский компьютер должен иметь свою копию локальной базы данных. Существенной проблемой СУБД такого типа является соответствие разных копий базы данных друг другу – синхронизация копий. Именно поэтому для решения задач, требующих совместной работы нескольких пользователей такой вид СУБД фактически, не используется.

К сетевым относятся файл-серверные, клиент-серверные и распределенные СУБД. Непременным атрибутом таких СУБД является сеть, обеспечивающая аппаратную связь компьютеров и делающая возможной корпоративную работу множества пользователей с одними и теми же данными. Работа первых двух видов СУБД основана на работе соответствующих ИС и описана выше. Распределенные СУБД могут содержать несколько десятков и сотен серверов БД. Количество клиентских мест в них может достигать нескольких сотен тысяч. В распределенных СУБД некоторые серверы могут дублировать друг друга, с целью достижения предельно малой вероятности отказов и сбоев, которые могут исказить жизненно важную информацию. Интерес к распределенным СУБД возрос в связи со стремительным развитием Интернета.

Данные, хранимые в базах данных, имеют определенную логическую структуру, то есть описываются некоторой моделью представления данных (моделью данных), поддерживаемой СУБД. К числу традиционных моделей относятся следующие:

* Иерархическая;
* Сетевая;
* Реляционная.

Иерархическая модель данных.

В иерархической модели связи между данными можно описать с помощью упорядоченного графа (или дерева).

Вообще, иерархическая модель строго структурирована, то есть взаимосвязь между объектами подчинена строгому ранжированию. Подчинение объектов разделено на уровни. На первом уровне представлен один главный объект, которому подчиняются объекты второго уровня. Причем объект первого уровня не может напрямую управлять объектом третьего уровня, управление объектом третьего уровня производится только через объект второго уровня.

Для описания структуры (схемы) иерархической базы данных на некотором языке программирования используется тип данных «дерево» («Tree»). Тип «дерево» является составным. Он включает в себя подтипы («поддеревья»), каждый из которых, в свою очередь, является типом «дерево». Каждый из этих типов «дерево» состоит из одного «корневого типа» и упорядоченного набора (возможно пустого).

В целом тип дерево представляет собой упорядоченную совокупность экземпляров типа «запись». Иерархическая БД представляет собой упорядоченную совокупность экземпляров типа «дерево», содержащих экземпляры типа «запись». Именно поля записей и хранят собственно информацию, содержащуюся в БД (числовые, строковые и другие значения). Обход всех элементов иерархической БД обычно производится сверху вниз и слева направо.

К достоинствам иерархической модели данных относятся эффективное использование памяти компьютера и неплохие показатели времени выполнения основных операций над данными. Иерархическая модель удобна при работе с иерархически упорядоченной информацией.

Недостатками иерархической модели является ее громоздкость для обработки информации с достаточно сложными логическими связями, а также сложность понимания для обычного пользователя.

Сетевая модель данных.

Сетевая модель данных позволяет отображать разнообразные взаимосвязи элементов данных в виде произвольного графа, обобщая тем самым иерархическую модель данных.

Сетевая модель более демократична, чем иерархическая и не так структурирована. В сетевой модели отсутствует понятие главного и подчиненного объекта. Один и тот же объект может выступать как главный, так и подчиненный. Для описания схемы сетевой БД используются две группы типов: «запись» и «связь». Тип «связь» определятся для двух типов «запись» - предка и потомка. Переменные типа связь являются экземплярами связей.

Физическое размещение данных в базах сетевого типа может быть организованно практически теми же методами, что и в иерархических базах данных.

К числу важнейших операций манипулирования данными баз сетевого типа можно отнести следующие:

* Поиск записи в БД;
* Переход от предка к первому потомку;
* Переход от потомка к предку;
* Создание новой записи;
* Удаление текущей записи;
* Обновление текущей записи;
* Включение записи в связь;
* Исключение записи из связи;
* Изменение связей и т.д.

Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. В сравнении с иерархической моделью сетевая модель предоставляет большие возможности в смысле допустимости образования произвольных связей.

Недостатками сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также сложность для понимания и выполнения обработки информации обычным пользователем. Кроме того, в сетевой модели ослаблен контроль целостности связей вследствие допустимости произвольного установления связей между записями.

Реляционная модель данных.

Как уже отмечалось выше, приведенные модели данных достаточно сложны для понимания обычного пользователя. Реляционная модель, описываемая ниже, в этом смысле имеет большое преимущество перед приведенными моделями. Именно поэтому, а также по ряду других причин, эта модель в настоящее время является самой распространенной для построения баз данных. А также по этим причинам именно эта модель будет использована для решения задачи курсового проетка. Рассмотрим основы реляционной модели данных более подробно.

Реляционная модель данных некоторой предметной области представляет собой набор отношений, изменяющихся во времени. При создании информационной системы совокупность отношений позволяет хранить данные об объектах предметной области и моделировать связи между ними.

Отношение является важнейшим понятием и представляет собой двумерную таблицу, содержащую некоторые данные.

Сущность есть объект любой природы, данные о котором хранятся в базе данных. Данные о сущности хранятся в отношении.

Атрибуты представляют собой свойства, характеризующие сущность. В структуре таблицы каждый атрибут именуется и ему соответствует заголовок некоторого столбца таблицы.

Как отмечалось выше, в реляционных базах данных широко используется связывание таблиц. При связывании двух таблиц выделяют основную и подчиненную таблицы. Логическое Связывание производится при помощи ключа связи. Ключ связи, по аналогии с обычным ключом таблицы, состоит из одного или нескольких полей, которые называют полями связи (ПС).

Вообще, суть связывания состоит в установлении соответствия полей связи основной и подчиненной таблицы.

В зависимости от того, как определены поля связи основной и подчиненной таблицы (как соотносятся поля связи к ключевым полям), между двумя таблицами могут устанавливаться следующие четыре основных вида связи:

Связь «один к одному» (1:1). Связь вида 1:1 образуется в случае, когда все поля связи основной и подчиненной таблицы являются ключевыми. Поскольку значения в ключевых полях не повторяются, обеспечивается взаимо-однозначное соответствие записей из этих таблиц. Сами таблицы при использовании такого вида связи становятся равноправными. На практике связь вида 1:1 используется сравнительно редко, так как хранимую в двух таблицах информацию, в этом случае, легко объединить в одну, что позволяет сэкономить ресурсы компьютера.

Связь «один ко многим» (1: М). Связь вида 1: М имеет место, если одной записи основной таблицы ставится в соответствие несколько записей подчиненной таблицы.

Связь «многие к одному» (М:1). Этот вид связи образуется, если одной или нескольким записям основной таблицы ставится в соответствие одна запись подчиненной таблицы.

Связь «многие ко многим» (М: М). Самый общий вид связи М: М возникает в случаях, когда нескольким записям основной таблицы ставится в соответствие несколько записей подсиненной таблицы. Очевидно, аналогично связи 1:1, связь М: М, делает связываемые таблицы, по сути, равноправными.

### 1.4.2. **Выбор средств разработки программного модуля**

После проведения анализа многих языков программирования, для разработки программного модуля был выбран язык программирования C#.

C# - современный объектно-ориентированный язык программирования. Он позволяет разработчикам создавать множество типов безопасных и надежных приложений, работающих в экосистеме .NET. Это язык с C-подобным синтаксисом.

Будучи объектно-ориентированным языком, он много перенял у Java и С++. Как и Java, C# изначально предназначался для веб-разработки, и примерно 75% его синтаксических возможностей такие же, как у Java. C# также называют «очищенной версией Java». Ещё 10% позаимствовал из C++ и 5% – из Visual Basic. Оставшиеся 10% C# — это реализация собственных идей разработчиков. Объектно-ориентированный подход позволяет строить с помощью C# крупные, но в то же время гибкие, масштабируемые и расширяемые приложения.

C# уже давно поддерживает много полезных функций:

* Инкапсуляция;
* Наследование;
* Полиморфизм;
* Перегрузка операторов;
* Статическая типизация.

При этом он всё ещё активно развивается, и с каждой новой версией появляется всё больше нововведений - например лямбды, динамическое связывание, асинхронные методы и т.д.

Выбор среды разработки для программного модуля пал на Visual Studio 2019.

Visual Studio 2019 - это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства авто завершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Основные возможности Visual Studio 2019:

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (Subversion, VisualSourceSafe и Git), добавление новых наборов инструментов (для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения).

Поддержка множества языков при разработке. Visual Studio позволяет писать код на своем языке или любых других предпочитаемых языках, используя все время один и тот же интерфейс (IDE).

Интуитивный стиль кодирования. По умолчанию Visual Studio форматирует код по мере его ввода, автоматически вставляя необходимые отступы и применяя цветовое кодирование для выделения элементов типа комментариев. Такие незначительные отличия делают код более удобным для чтения и менее подверженным ошибкам. Применяемые Visual Studio автоматически параметры форматирования можно даже настраивать, что очень удобно в случаях, когда разработчик предпочитает другой стиль размещения скобок (например, стиль K&R, при котором открывающая скобка размещается на той же строке, что и объявление, которому она предшествует).

Более высокая скорость разработки. Многие из функциональных возможностей Visual Studio направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Удобные функции, вроде функции IntelliSense (которая умеет перехватывать ошибки и предлагать правильные варианты), функции поиска и замены (которая позволяет отыскивать ключевые слова как в одном файле, так и во всем проекте) и функции автоматического добавления и удаления комментариев (которая может временно скрывать блоки кода), позволяют разработчику работать быстро и эффективно.

Возможности отладки. Предлагаемые в Visual Studio инструменты отладки являются наилучшим средством для отслеживания загадочных ошибок и диагностирования странного поведения. Разработчик может выполнять свой код по строке за раз, устанавливать интеллектуальные точки прерывания, при желании сохраняя их для использования в будущем, и в любое время просматривать текущую информацию из памяти.

Visual Studio также имеет и множество других функций: возможность управления проектом; встроенная функция управления исходным кодом; возможность факторизации кода; мощная модель расширяемости.

Глава 2. Разработка программного модуля

# **2.1. Создание базы данных**

Итоговый продукт будет состоять из двух основных частей: базы данных, основанной на MS Access, и пользовательского приложения для работы с БД, созданного в среде программирования Visual Studio.

База данных содержит следующие таблицы:

* Автомобили
* Админы
* Пользователи
* Арендованные авто

Таблица «Автомобили» содержит информацию о автомобилях. Ее структура представлена ниже:



Рисунок 1 – Структура таблицы «Автомобили»

Таблица «Админы» содержит информацию об админах. Ее структура:

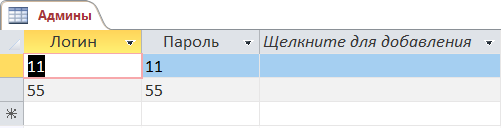


Рисунок 2 – Структура таблицы «Админах»

Таблица «Пользователи» содержит информацию об пользователях. Ее структура:

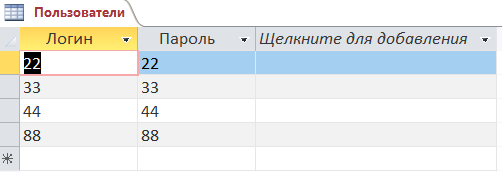


Рисунок 3 – Структура таблицы «Пользователи»

Таблица «Арендованные авто» содержит информацию об арендованном автомобиле и ФИО человека арендовавшего его:

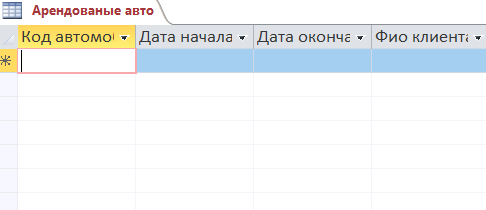


Рисунок 4 – Структура таблицы «Пользователи»

Добавляем ранее созданную базу данных в нашу программу, используя мастер настройки источника данных и компонент dataGridView.

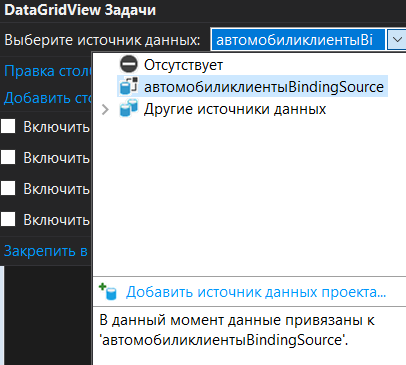


Рисунок 5 – Подключенная база данных.

# **2.2. Создание программного приложения**

Для работы с данной БД, необходимо разработать приложение. Само приложение будет содержать несколько форм, с помощью которых будет происходить работа с данными:

* Главная кнопочная форма
* Форма «Авторизация администратора»
* Форма «Кабинет администратора»
* Форма «Регистрации нового пользователя»
* Форма «Неверного ввода данных»
* Форма «Авторизация пользователя»
* Форма «Кабинет пользователя»
* Форма «Регистрации нового админа»
* Форма «Добавления нового автомобиля»

Описание форм:

При входе в приложение нас встречает, главная кнопочная форма в которой делается выбор под каким типом пользователя человек хочет зарегистрироваться.

.

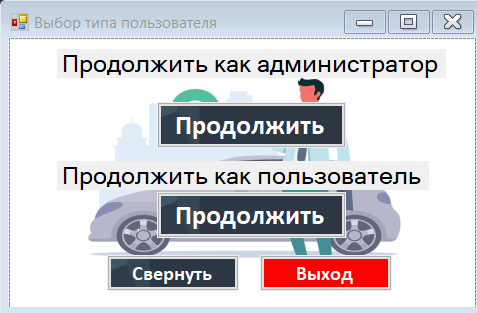


Рисунок 6 – Главная форма

Форма «Авторизация администратора» необходима для авторизации администратора.

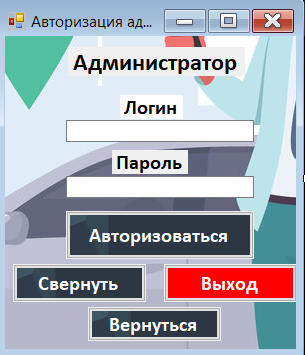


Рисунок 7 – Форма «Авторизация администратора»

Форма «кабинет администратора» С ее помощью администратор может:

1. Удалять данные клиентов, у которых закончился срок аренды автомобиля.
2. Добавлять новых администраторов.
3. Осуществлять поиск по всей таблице.
4. Обновлять данные в таблице.
5. Сохранять изменения в таблице.
6. Свернуть приложении.
7. Вернуться к авторизации.
8. Выйти из приложения.

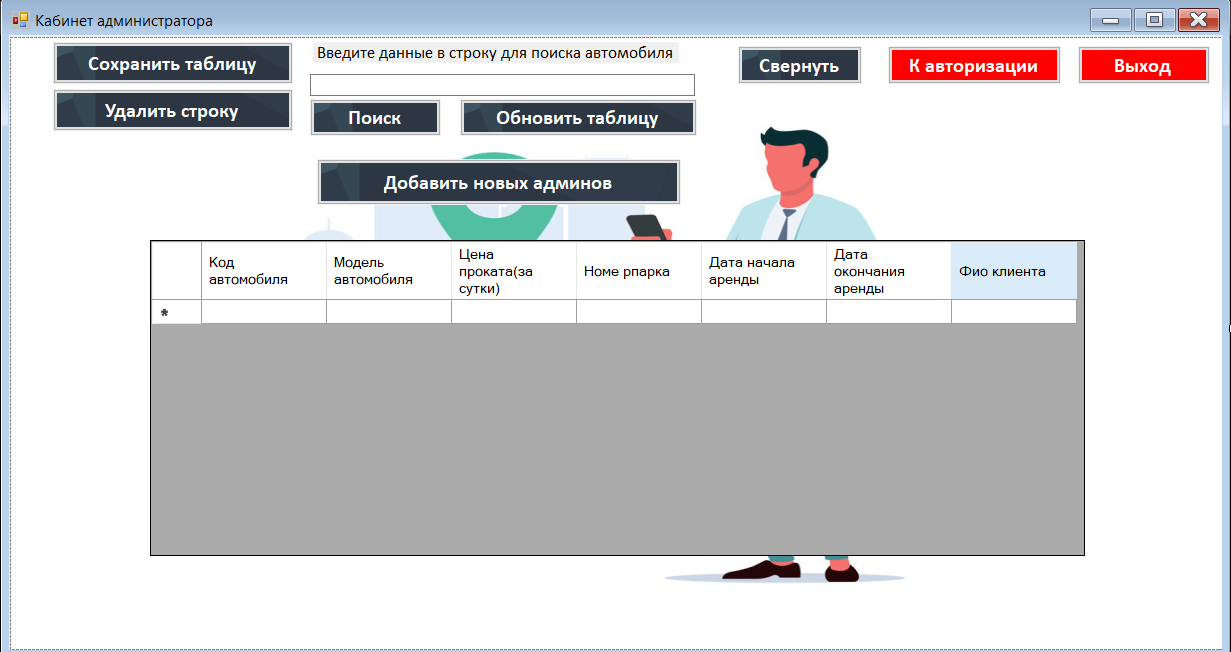


Рисунок 8 – окно “ Кабинет администратора”

Форма «Авторизация нового администратора» с ее помощью администратор может:

1. Добавить данные нового администратора.
2. Удалить данные администратора.
3. Сохранить измененные данные в базу данных.
4. Вернуться в свой кабинет.
5. Свернуть приложение.
6. Выйти из приложения.

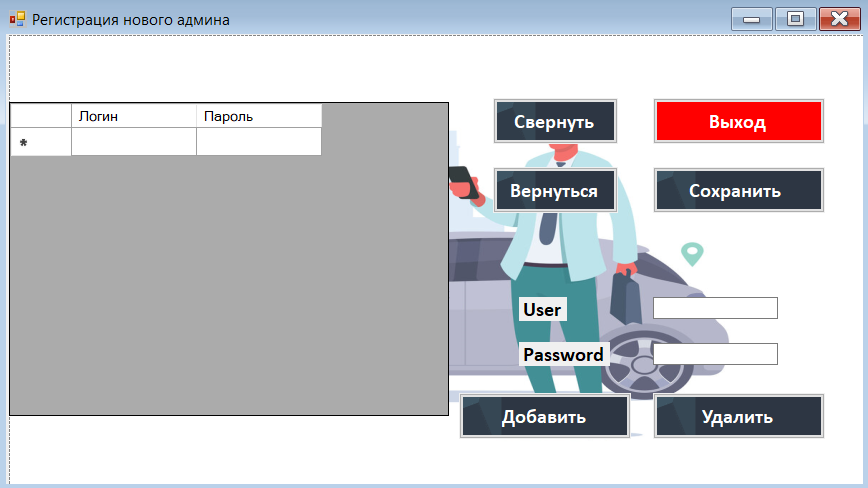


Рисунок 9 – Регистрация нового администратора

Форма «Авторизация пользователя» необходима для авторизации пользователя.

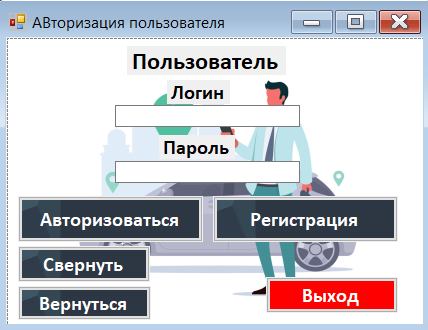


Рисунок 10 – Авторизация пользователя

Форма «кабинет администратора» c ее помощью администратор может:

1. Произвести расчет аренды выбранного ими автомобиля.
2. Осуществлять поиск по всей таблице.
3. Сохранять изменения в таблице.
4. Свернуть приложении.
5. Вернуться к авторизации.
6. Выйти из приложения.

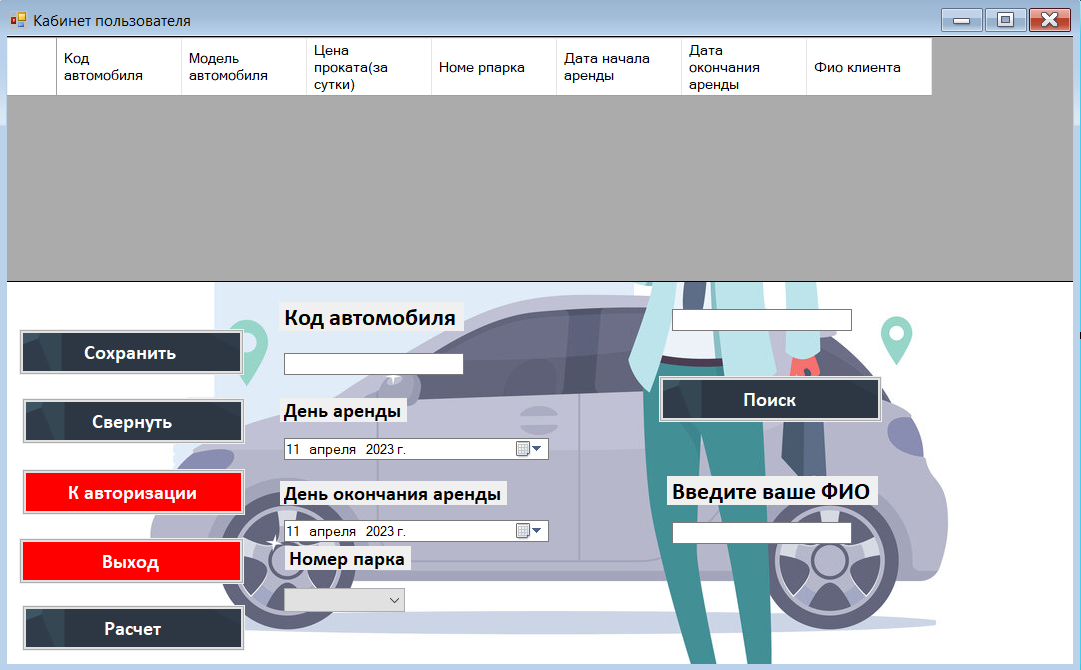


Рисунок 11 – окно “ Кабинет пользователя”

Форма «Добавить новый автомобиль» c ее помощью администратор может:

1. Произвести расчет аренды выбранного ими автомобиля.
2. Осуществлять поиск по всей таблице.
3. Сохранять изменения в таблице.
4. Свернуть приложении.
5. Вернуться к авторизации.
6. Выйти из приложения.

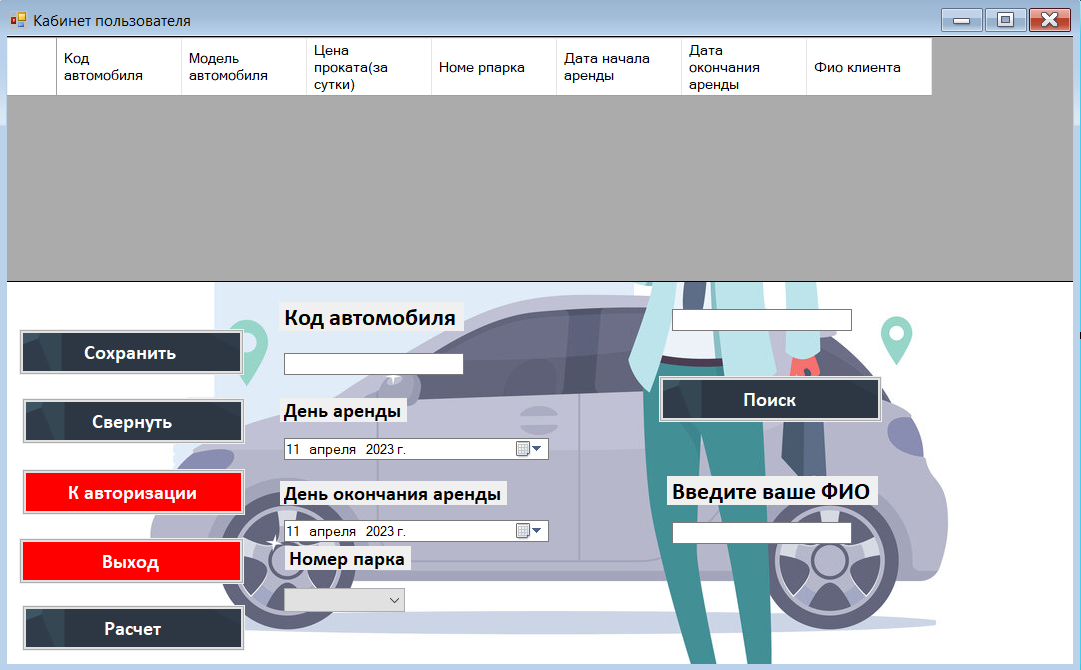


Рисунок 11 – окно “ Кабинет пользователя”

Форма «Авторизация нового пользователя» с ее помощью пользователь может:

1. Зарегистрироваться.
2. Сохранить измененные данные в базу данных.
3. Вернуться к авторизации.
4. Свернуть приложение.
5. Выйти из приложения.

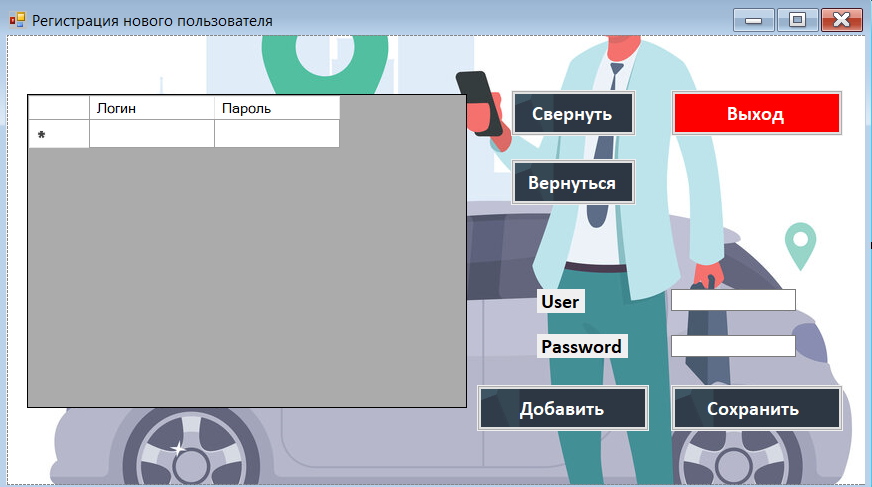


Рисунок 12 – Регистрация нового пользователя.

Форма-неверный ввод данных-если пользователь ввел данные не корректно или данные которых нет в базе данных.

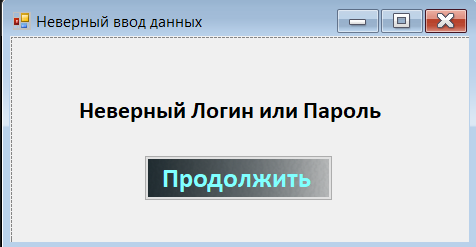


Рисунок 13 – Неверный ввод данных.

Заключение

На основе полученной информации были сформулированы задачи, по созданию программного модуля аренды автомобилей.

Удалось создать удобный пользовательский интерфейс, позволяющий быстро и без наличия специальных навыков работать с программой. Нам удалось выполнить все выявленные в начале разработки информационной системы задачи и достичь поставленных целей.

Для создания программного продукта были проанализированы и сравнены различные средства проектирования и разработки баз данных, языка программирования с#, а также Visual studio2019.

Данный программный модуль может использоваться людьми в повседневной жизни для быстрой аренды автомобиля.

Данную информационную систему в будущем можно будет модернизировать,

Исходя из того, что в процессе работы все поставленные задачи были выполнены в полной мере, можно сказать, что цель работы достигнута.

Список использованных источников

1. Астапчук, В. А. Корпоративные информационные системы: требования при проектировании / В. А. Астапчук, П. В. Терещенко. — 2-е изд. — М.: Издательство Юрайт, 2019. — 113 c. — Текст: непосредственный.
2. Васильев, А. Н. Программирование на C# для начинающих. Основные сведения / А. Н. Васильев. — Текст: электронный // AvidReaders.ru: [сайт]. — URL: https://avidreaders.ru/book/programmirovanie-na-c-dlya-nachinayuschih-osnovnye.html (дата обращения: 09.03.2022).
3. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2019. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/438444 (дата обращения: 24.03.2022).
4. Документация по C#. — Текст: электронный // Microsoft: [сайт]. — URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения: 25.03.2022).
5. Зыков, С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. — М.: Издательство Юрайт, 2022. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст: непосредственный.
6. Илюшечкин, В. М. Основы использования и проектирования баз данных: учебник для вузов / В. М. Илюшечкин. — М.: Издательство Юрайт, 2021. — 213 с. — Текст: непосредственный.
7. Казанский, А. А. Программирование на Visual C#: учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб, и доп. — М.: Издательство Юрайт, 2022. — 192 с. — Текст: непосредственный.
8. Кубенский, А. А. Функциональное программирование: учебник и практикум для вузов / А. А. Кубенский. — М.: Издательство Юрайт, 2022. — 348 с. — (Высшее образование). — ISBN 978-5-9916-9242-7. — Текст: непосредственный.
9. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — М.: Издательство Юрайт, 2020. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/456221 (дата обращения: 21.03.2022).
10. Кудрявцев, В. Б. Интеллектуальные системы: учебник и практикум для вузов / В. Б. Кудрявцев, Э. Э. Гасанов, А. С. Подколзин. — 2-е изд., исп. и доп. — М.: Издательство Юрайт, 2022. — 165 с. — (Высшее образование). — ISBN 978-5-534-07779-7. — Текст: непосредственный.
11. Малов, А. В. Концепции современного программирования: учебное пособие для вузов / А. В. Малов, С. В. Родионов. — М.: Издательство Юрайт, 2022. — 96 с. — (Высшее образование). — ISBN 978-5-534-14911-1. — Текст: непосредственный.
12. Мякишев, Д. В. Разработка программного обеспечения АСУ ТП на основе объектно-ориентированного подхода / Д. В. Мякишев. — Текст: электронный // AvidReaders.ru: [сайт]. — URL: https://avidreaders.ru/book/razrabotka-programmnogo-obespecheniya-asu-tp-na.html (дата обращения: 14.03.2022).
13. Основные принципы реляционных данных. — Текст: электронный // Microsoft: [сайт]. — URL: https://docs.microsoft.com/ru-ru/learn/modules/describe-concepts-of-relational-data/ (дата обращения: 04.03.2022).
14. Подбельский, В. В. Программирование. Базовый курс С#: учебник для среднего профессионального образования / В. В. Подбельский. — М.: Издательство Юрайт, 2022. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст: непосредственный.
15. Подключение к данным в БД Access. — Текст: электронный // Microsoft: [сайт]. — URL: https://docs.microsoft.com/ru-ru/visualstudio/data-tools/connect-to-data-in-an-access-database-windows-forms?view=vs-2022 (дата обращения: 02.03.2022).
16. Полное руководство по языку программирования С#. — Текст: электронный // Metanit.com: [сайт]. — URL: https://metanit.com/sharp/tutorial/ (дата обращения: 10.03.2022).
17. Руководство по ADO.NET и работе с базами данных в .NET 6. — Текст: электронный // Metanit.com: [сайт]. — URL: https://metanit.com/sharp/adonetcore/ (дата обращения: 29.03.2022).
18. Советов, Б. Я. Информационные технологии: учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский. — 7-е изд., перераб, и доп. — М.: Издательство Юрайт, 2022. — 327 с. — (Профессиональное образование). — ISBN 978-5-534-06399-8. — Текст: непосредственный.
19. Станкевич, Л. А. Интеллектуальные системы и технологии: учебник и практикум для вузов / Л. А. Станкевич. — М.: Издательство Юрайт, 2022. — 397 с. — (Высшее образование). — ISBN 978-5-534-02126-4. — Текст: непосредственный.
20. Стасышин, В. М. Базы данных. Технологии доступа: учебное пособие для среднего профессионального образования / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., исп. и доп. — М.: Издательство Юрайт, 2020. — 164 с. — (Профессиональное образование). — ISBN 978-5-534-09888-4. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/455863 (дата обращения: 25.02.2022).
21. Стружкин, Н. П. Базы данных: проектирование. Практикум: учебное пособие для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — М.: Издательство Юрайт, 2022. — 291 с. — (Профессиональное образование). — ISBN 978-5-534-08140-4. — Текст: непосредственный.
22. Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. — М.: Издательство Юрайт, 2022. — 206 с. — (Высшее образование). — ISBN 978-5-534-00849-4. — Текст: непосредственный.
23. Черпаков, И. В. Основы программирования: учебник и практикум для среднего профессионального образования / И. В. Черпаков. — М.: Издательство Юрайт, 2022. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст: непосредственный.