**Оглавление**

[Аннотация 4](#_Toc72446649)

[Введение 5](#_Toc72446650)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ 6](#_Toc72446651)

[1.1 Основное функциональное назначение программного средства 6](#_Toc72446652)

[1.2 Полное наименование программного средства 6](#_Toc72446653)

[1.3 Условное обозначение программного средства 6](#_Toc72446654)

[1.4 Разработчики программного средства 6](#_Toc72446655)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc72446656)

[2.1 Основание для разработки 7](#_Toc72446657)

[2.2 Назначение разработки 7](#_Toc72446658)

[2.3 Требование к программному средству 7](#_Toc72446659)

[2.4 Требования к программной документации 8](#_Toc72446660)

[2.5 Требования к эргономике и технической эстетике 8](#_Toc72446661)

[2.6 Стадии и этапы разработки 9](#_Toc72446662)

[2.7 Порядок контроля и приемки 10](#_Toc72446663)

[3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ 11](#_Toc72446664)

[3.1 Декомпозиция поставленной задачи 11](#_Toc72446665)

[3.2 Общая архитектура программного средства 12](#_Toc72446666)

[3.3 Разработка алгоритма решения задачи 15](#_Toc72446667)

[3.4 Реализация функционального назначения программного средства 16](#_Toc72446668)

[3.5 Структурная организация данных 16](#_Toc72446669)

[3.6 Разработка интерфейса ПС 22](#_Toc72446670)

[3.7 Описание структуры выходной информации 35](#_Toc72446671)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 36](#_Toc72446672)

[4.1 Назначение программного средства 36](#_Toc72446673)

[4.2 Условия выполнения программного средства 36](#_Toc72446674)

[4.3 Эксплуатация программного средства 36](#_Toc72446675)

[4.4 Сообщения пользователю 50](#_Toc72446676)

[5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА 51](#_Toc72446677)

[5.1 Определение верхней границы количества тестов 51](#_Toc72446678)

[5.2 Разработка тестовых вариантов 52](#_Toc72446679)

[5.3 Составление отчета о тестировании 54](#_Toc72446680)

[ЗАКЛЮЧЕНИЕ 60](#_Toc72446681)

[Оценка качества программного средства с помощью метрик 60](#_Toc72446682)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 63](#_Toc72446683)

[Приложение А Текст кода программного средства 67](#_Toc72446684)

[Приложение Б Спецификация 110](#_Toc72446685)

[Приложение B Диск с программным средством 112](#_Toc72446686)

**Аннотация**

На дипломную работу студента Смирнова Н.В. на тему: Автоматизированная информационная система «Каршеринговая компания MaxДрайв». Дипломная работа представлена пояснительной запиской на 112 листов машинописного текста. Состоит из введения, 5 глав, заключения и списка литературных источников.

В главе 1 содержится общие сведения о программном средстве АИС MaxДрайв. В главе 2 приведено техническое задание на проектирование БД согласно ГОСТ 19.201. Глава 3 содержит пояснительную записку к программному продукту согласно ГОСТ 19.404-79. В главе 4 содержится руководство пользователя согласно ГОСТ 19.504-79. Глава 5 содержит обоснование количества необходимых тестов и тестовые пакеты для оценки работоспособности программного продукта. В главе 6 приведены результаты ручного и автоматического тестирования программного продукта и выполнена оценка качества с помощью метрик.

В работе содержится общее описание программного средства, предназначенного для объекта проектирования – автоматизированной иинформационной системы «Каршеринговая компания MaxДрайв», изложены требования к программному средству, описаны его архитектура, реализация функций, интерфейс, приведено руководство пользователя, обоснованы тестовые пакеты и выполнена оценка качества программного продукта с помощью метрик.

**Введение**

В дипломной работе была поставлена задача разработки автоматизированной информационной системы «Каршеринговая компания MaxДрайв», которая позволяет автоматизировать деятельность каршеринговой компании.

Для достижения поставленной цели был составлен основной алгоритм работы программы, а также разработаны алгоритмы вспомогательных процедур.

Для реализации алгоритмов был выбран язык программирования C#. Выбор обусловлен тем, что возможностей этого языка достаточно для достижения поставленной цели моей дипломной работы.

При разработке интерфейса программы, я учитывал наиболее нужные и важные функциональные возможности, которые должна выполнять данная программа, а также тип пользователей, которые с ней будут работать.

**1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ**

**1.1 Основное функциональное назначение программного средства**

Программа предназначена для автоматизации компании «MaxДрайв», которая позволяет бронировать автомобили. Программа не требует владения современными информационными технологиями для эффективного использования данного программного средства, т.е. подходит для любого пользователя.

**1.2 Полное наименование программного средства**

Автоматизированная информационная система «Каршеринговая компания MaxДрайв».

**1.3 Условное обозначение программного средства**

«CarsharingApp»

**1.4 Разработчики программного средства**

Программное средство разработал студент Зеленодольского Механического колледжа группы 195 Смирнов Н.В.

**2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**2.1 Основание для разработки**

Работа ведётся на основании задания на дипломную работу по специальности 09.02.03 — «Программирование в компьютерных системах».

**2.2 Назначение разработки**

Программа «CarsharingApp» работает под управление операционной системы Windows 7, Windows 8, Windows 10. Программа предназначена для автоматизации деятельности компании «MaxДрайв». Цель создания программы – учёт бронирования автомобилей. Программа позволяет:

* создавать и редактировать базу автомобилей;
* создавать базу клиентов;
* бронирование выбранного автомобиля;
* создавать и редактировать базу дополнительных услуг.

**2.3 Требование к программному средству**

«CarsharingApp» устанавливается на персональный компьютер, имеющий представленные минимальные системные требования:

Оперативная память: 4 Гб;

Процессор: Intel Core i / Phenom II X4 / AMD Athlon64 X2 4600 + 2.4 ГГц;

Видео карта: nVidia GeForce 8600 / ATI Radeon HD 2400;

Звуковая карта: Совместимая с DirectX;

Программный продукт «CarsharingApp» является кроссплатформенным продуктом, что позволяет не ограничиваться в работе с определенной версией ОС.

«CarsharingApp» написана на объектно-ориентированном языке программировании C# (си-шарп) в интегрированной среде разработки Visual Studio 2022. В качестве СУБД используется MS SQL. Программа не может подлежать сторонней оптимизации и редактированию программного кода.

**2.4 Требования к программной документации**

Состав программной документации должен включать в себя:

1. техническое задание;

2. сроки выполнения работ;

3. программу и методику испытаний;

4. эксплуатационные инструкции пользователю;

**2.5 Требования к эргономике и технической эстетике**

Система обеспечивает удобный для пользователей системный интерфейс, отвечающий следующим требованиям:

- единый стиль оформления для пользовательских интерфейсов;

- удобная, интуитивно понятная навигация в интерфейсе пользователя;

- взаимодействие пользователя с системой осуществляется на двух языках по выбору (русский и английский), для наибольшего удобства

Пользовательские интерфейсы системы спроектированы и разработаны с применением единых принципов графического представления информации и организации доступа к функциональным возможностям и сервисам. Разработан графический дизайн пользовательских интерфейсов, цветовые, шрифтовые и композиционные решения для отображения текстов, изображений, таблиц, гиперссылок, управляющих и навигационных элементов (меню, кнопок, форм и т.п.).

Экранные формы спроектированы с учетом требований унификации:

- все экранные формы пользовательского интерфейса выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

- для обозначения сходных операций использованы сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении унифицированы;

- внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) реализованы одинаково для однотипных элементов.

**2.6 Стадии и этапы разработки**

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4. определение стадий, этапов и сроков разработки программы и документации на нее;

5. выбор языков программирования;

6. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с пунктом Предварительный состав программной документации настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение порядка и методики испытаний;

2. проведение испытаний;

3. корректировка программы и программной документации по результатам испытаний.

**2.7 Порядок контроля и приемки**

Сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом. Сдача-приемка осуществляется комиссией, в состав которой входят представители ГАПОУ и работодатель. По результатам приемки подписывается акт приемочной комиссии.

В указанном подразделе, согласно таблице «Сроки выполнения работ» технического задания, будет описаны перечень участвующих организаций, место и сроки проведения работ, согласно п. 2.8 ГОСТ 34.602-89.

Порядок согласования и утверждения приемочной документации должен регламентироваться организационно-распорядительной документацией организации, принимающей участие в создании программного продукта. Согласно разделу «Приемка результатов разработки» ГОСТ 15.001-88 для согласования и утверждения приемочной документации создается приемочная комиссия (приказом).

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

**3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ**

**3.1 Декомпозиция поставленной задачи**

Для достижения цели работы была поставлена следующая задача:

Создать модуль “CodeDatabase” отвечающий за сохранение БД. Результат работы модуля - сохранение БД.

Поставленные задачи наглядно отображаются на диаграмме деятельности.

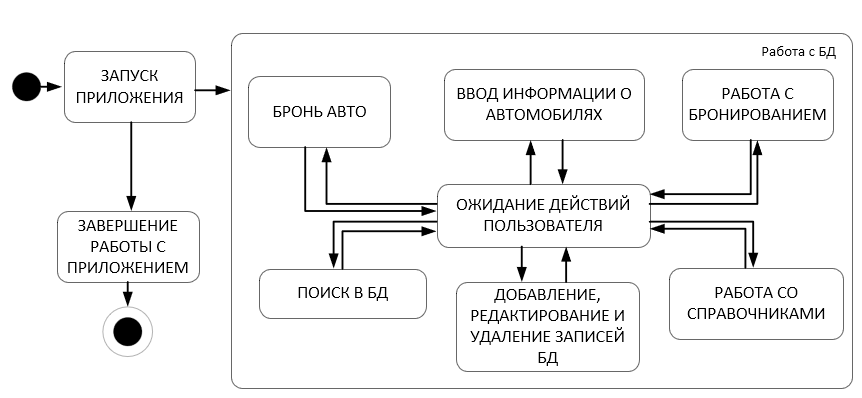


Рис.1 Декомпозиция поставленной задачи

**3.2 Общая архитектура программного средства**

Объектом моделирования является Автоматизированная информационная система «Каршеринговая компания MaxДрайв».

Рассматриваемые процессы: учёт брони.

Объекты моделирования представлены на диаграмме классов, рис.2.

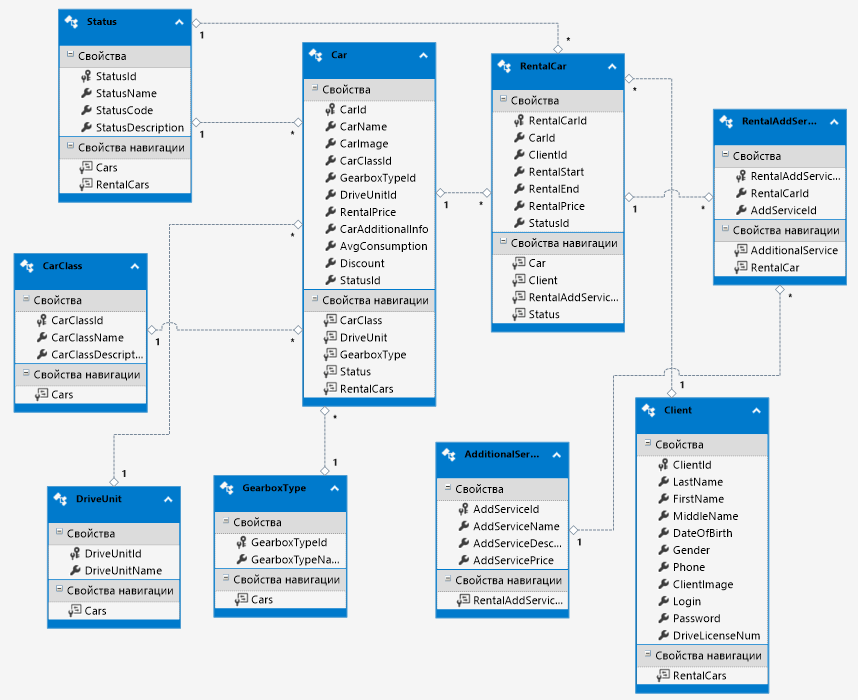


Рис.2 Диаграмма классов

По полученной декомпозиции задач была спроектирована следующая архитектура программного средства.

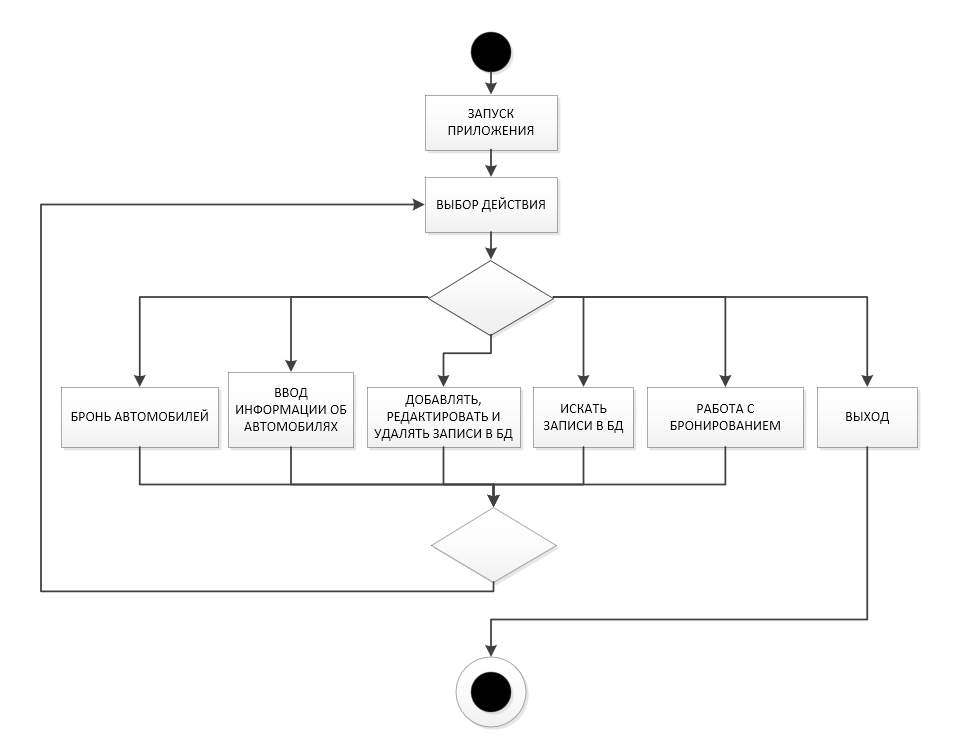


Рис.3 Схема событийно-ориентированной архитектуры

В «CarsharingApp» используется событийно-ориентированная архитектура.(Рис.3) Роль агента (источник событий) в программе выполняют: пользователи программы, за роль стоков (потребители событий) отвечают таблицы, входящие в базу данных. Например, когда пользователь выбирает определенное действие: добавить, удалить, редактировать, вывести в Excel и т.д. система осуществит выбранные действия, и база отреагирует соответствующим образом: запись добавлена, удалена, отредактирована, либо отчет выведен в Excel. Системная архитектура пользователя рассматривает это изменение состояния как событие, создаваемое, публикуемое, определяемое и потребляемое различными приложениями в составе архитектуры.

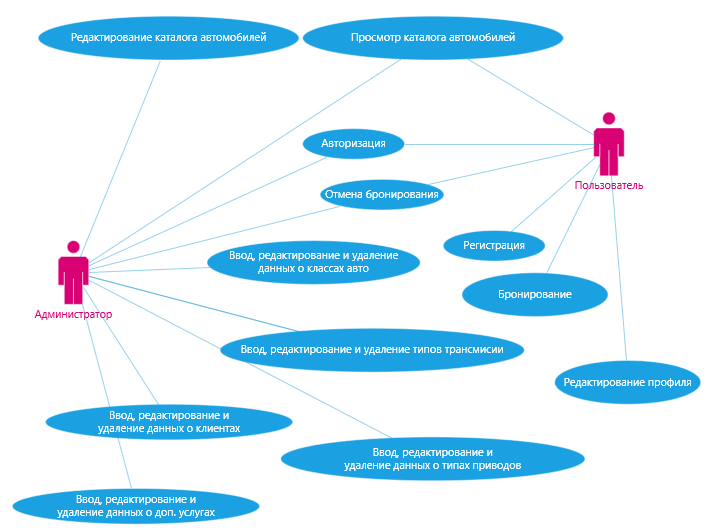
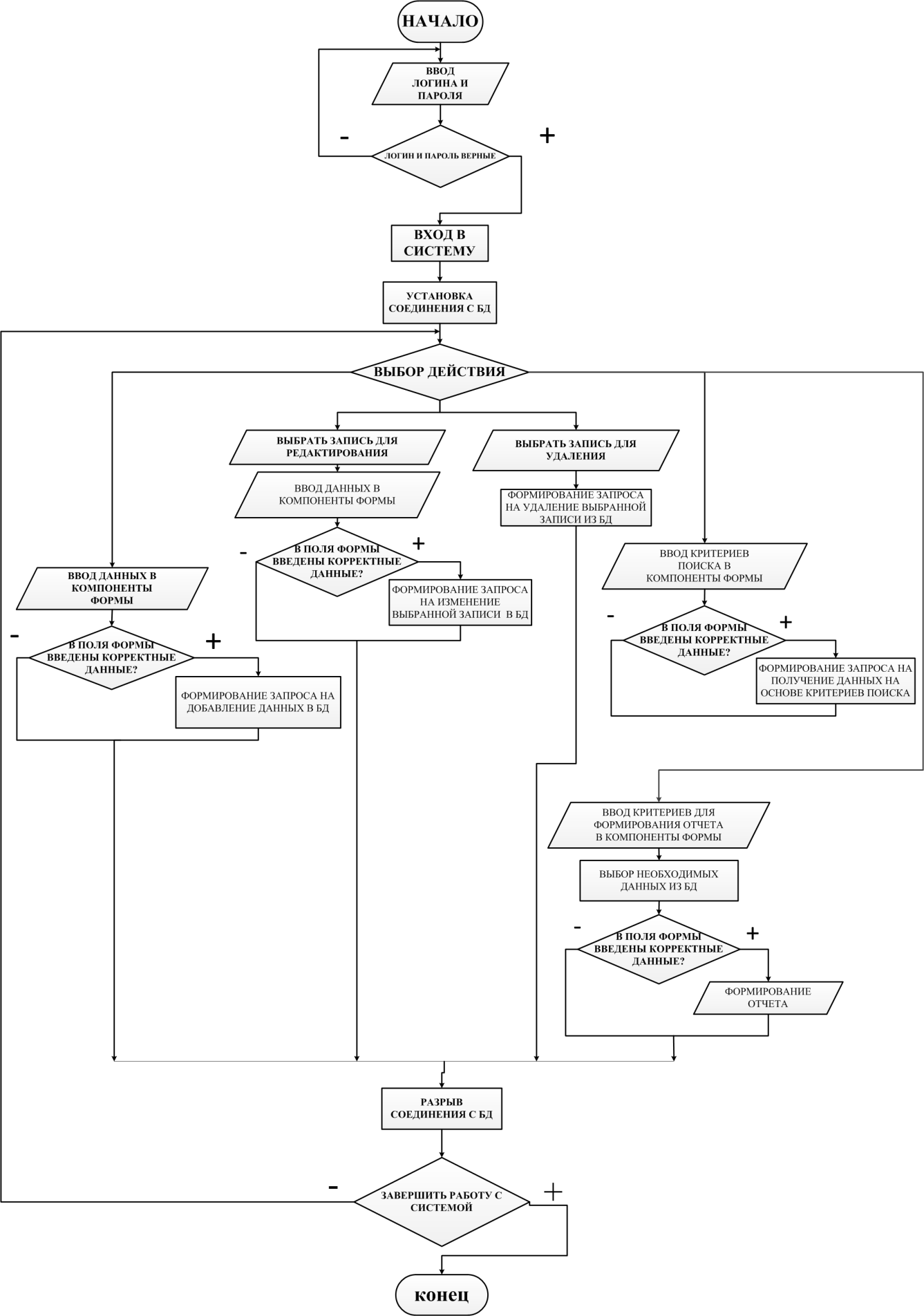


Рис.4 Диаграмма вариантов использования

**3.3 Разработка алгоритма решения задачи**

**Общий алгоритм**

****Рис.5 Алгоритм программы

**3.4 Реализация функционального назначения программного средства**

Программа «CarsharingApp» имеет следующий набор входных данных, такие как: каталог автомобилей, бронь автомобилей.

Данные вводятся администратором или пользователем в соответствующие поля ввода, снабженные всплывающими подсказками.

Выходными данными являются: каталог автомобилей, профиль пользователя, и офоромление брони.

Данные вводятся пользователем в соответствующие поля вывода, снабженные всплывающими подсказками.

Выходные данные редактировать вручную невозможно, т.к. они служат в качестве отчетной информации для пользователя.

**3.5 Структурная организация данных**

Для создания БД необходимо определиться с данными, которые необходимы для полноценного функционирования системы. Все эти данные указаны в реляционной модели представленной на рисунке. Любая реляционная база данных и называется реляционной, что характеризуется отношениями (relation) между таблицами. На рисунке изображены таблицы моей базы данных. При этом одна таблица является родительской (главной), а вторая – дочерней (подчиненной). Главной таблицей является «Бронь автомобиля». Реляционная модель автоматизированной системы соответствует всем 12 правилам Кодда. (рис.6)

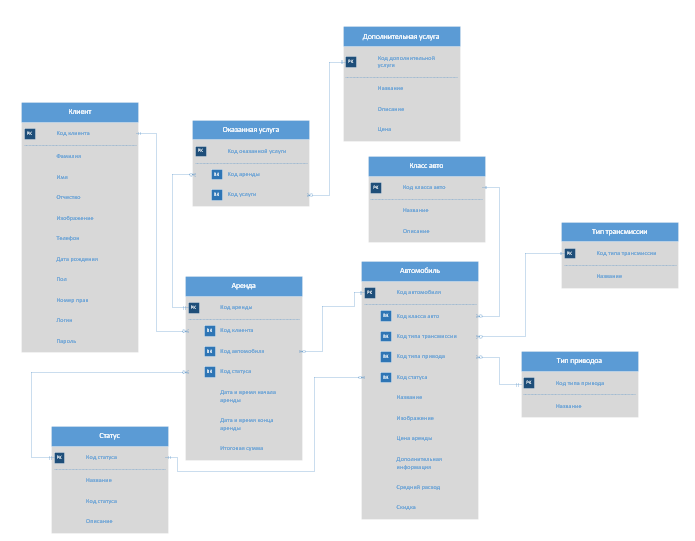


Рис.6 Реляционная модель

Первичный ключ в базе уникален, используется для организации отношений между таблицами, который не может иметь пустых и повторяющихся значений. Первичными ключами в базе являются поля: «Код клиента» (таблица «Клиент»), «Код статуса» (Таблица «Статус»), «Код аренды» (Таблица «Аренда»), «Код автомобиля» (Таблица «Автомобиль»), «Код типа привода» (Таблица «Тип привода»), «Код типа трансмиссии» (Таблица «Тип трансмиссии»), «Код оказанной услуги» (Таблица «Оказанная услуга»), «Код класса авто» (Таблица «Класс авто»), «Код дополнительной услуги» (Таблица «Дополнительная услуга). Остальные ID-ключи являются внешними ключами.

Для организации более эффективной обработки данных применяется нормализация. Таблицы моей БД находятся в 3НФ:

* БД находится в форме -1НФ потому, что
  + Таблица не имеет повторяющихся записей;
  + Каждый атрибут отношения хранит одно-единственное значение и не является списком, ни множеством значений;
  + Таблица не имеет повторяющихся групп полей.
    - Вторая нормальная форма(2НФ):
      * Устранены атрибуты, зависящие только от части уникального (первичного) идентификатора, т.е. ID.
        + Третья нормальная форма(3НФ):

Отсутствуют атрибуты, зависящие от атрибутов, не входящих в уникальный (первичный) ключ.

На основе реляционной модели базы данных мною в MS SQL была построена следующая база данных. (Рис.7) В ней первичным ключом является поле таблицы “Cars” – “CarId”. Связь таблиц “Cars” и “GearboxTypes” по полю “GearboxTypeId” один ко многим (1:М). Связь таблиц “Cars” и “DriveUnits” по полю “DriveUnitId” один ко многим (1:1). Связь таблиц “Cars” и “CarClasses” по полю “CarClassId” один ко многим (1:М). Связь таблиц “Cars” и “Statuses” по полю “StatusId” один ко многим (1:М). Связь таблиц “RentalCars” и “Cars” по полю “CarId” один ко многим (1:М). Связь таблиц “RentalCars” и “Clients” по полю “ClientId” один ко многим (1:М). Связь таблиц “RentalAddServices” и “RentalCars” по полю “RentalCarId” один ко многим (1:М). Связь таблиц “RentalAddServices” и “AdditionalServices” по полю “AdditionalServiceId” один ко многим (1:М). Все связи между таблицами базы данных типа один ко многим.



Рис.7 Схема базы данных

Таблица “DriveUnits”(Типы приводов)

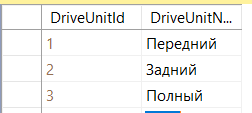


Рис.8 Таблица DriveUnits

Таблица “GearboxTypes”(Типы трансмиссий)

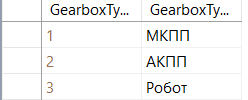


Рис.9 Таблица GearboxTypes

Таблица “CarClasses”(Классы авто )

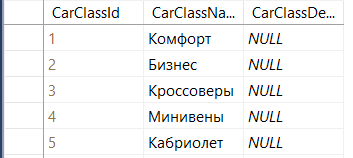


Рис.10 Таблица CarClasses

Таблица “AddServices”(Дополнительные услуги)



Рис.11 Таблица AddServices

Таблица “Cars”(Автомобили)

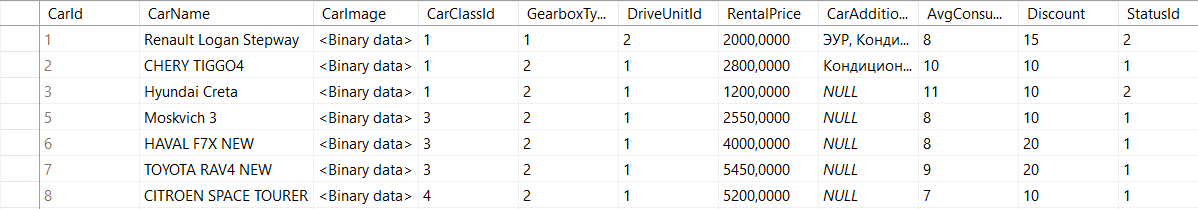


Рис.12 Таблица Cars

Таблица “Clients”(Клиенты)

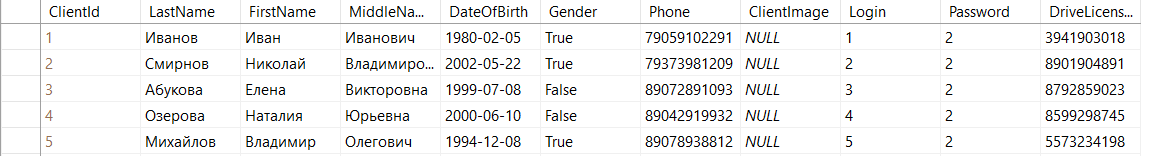


Рис.13 Таблица Clients

Таблица “RentalAddServices”(Список выбранных дополнительных услуг)

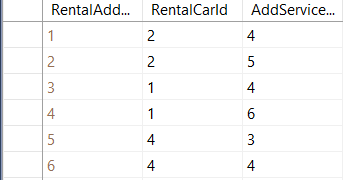


Рис.14 Таблица RentalAddServices

Таблица “RentalCar”(Аренда автомобилей)

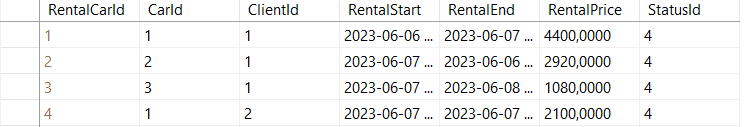


Рис.15 Таблица RentalCar

Таблица “Statuses”(Статусы)

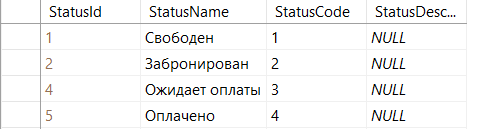


Рис.16 Таблица Statuses

**3.6 Разработка интерфейса ПС**

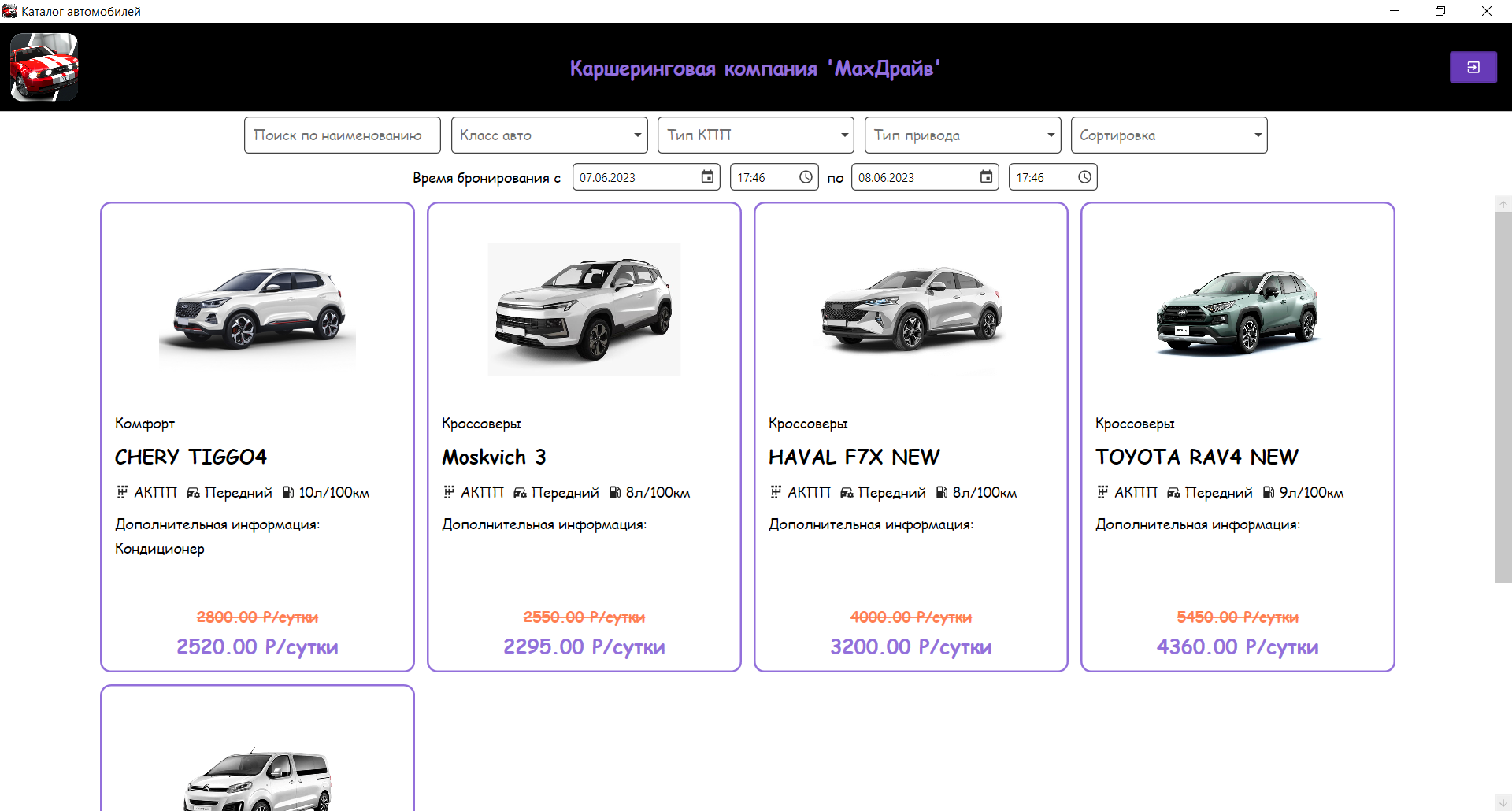
Приложение предназначено для двух типов пользователей: администратора и клиент. Для входа в панель администратора необходимо ввести свои учётные данные. Для запуска приложения достаточно щелкнуть дважды левой кнопкой мыши по ярлыку.

После запуска программы пользователь приступит к работе с главной формой программы, на которой отбражается каталог автомобилей в виде плиточного интерфейса. «CarsharingApp» обладает удобным интерфейсом, так что даже неопытному в обращение с пользовательскими компьютерными программами пользователю не составит труда быстро привыкнуть к работе с данным программным средством.

На рабочем экране расположены несколько кнопок для комфортной работы с приложением.

**Форма Автоматизированная инфомационная система «Каршеринговая компания MaxДрайв»»**. Основная форма приложения (Рис.17). При разработке интерфейса этой формы было использовано множество различныx компонентов:

1. ComboBox – выпадающие список для фильтрации;
2. DatePicker – для выбора даты аренды;
3. TimePicker – для выбора времени аренды;
4. Buttons – кнопки для перехода на другие формы;
5. TextBox – поля для ввода



Button

ComboBox

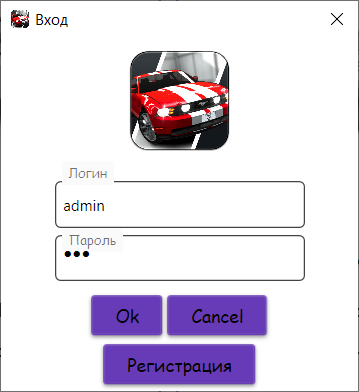
DatePicker

TextBox

TimePicker

Рис.17 Автоматизированная информационная система «Каршеринговая компания MaxДрайв»

**Форма входа в систему**. Интерфейс данной формы состоит из компонентов: Button – кнопки ОК, Cancel, Регистрация, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода логина, PasswordBox – поле для ввода пароля и TextBlock – текстовые метки. (Рис.18)



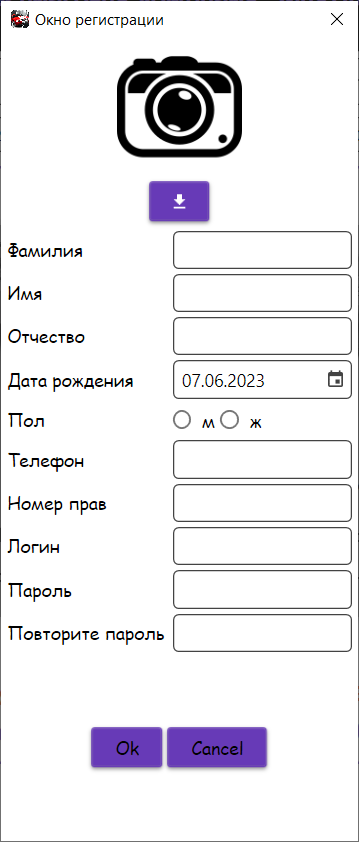
Buttons

TextBox

PasswordBox

Рис.18 Форма входа в систему

**Форма Регистрации**. Интерфейс данной формы состоит из компонентов: Button – кнопки Загрузить, ОК, Cancel, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода логина, PasswordBox – поле для ввода пароля, RadioButton – для выбора пола, DatePicker – для выбора даты рождения и TextBlock – текстовые метки. (Рис.19)



RadioButton

DatePicker

Buttons

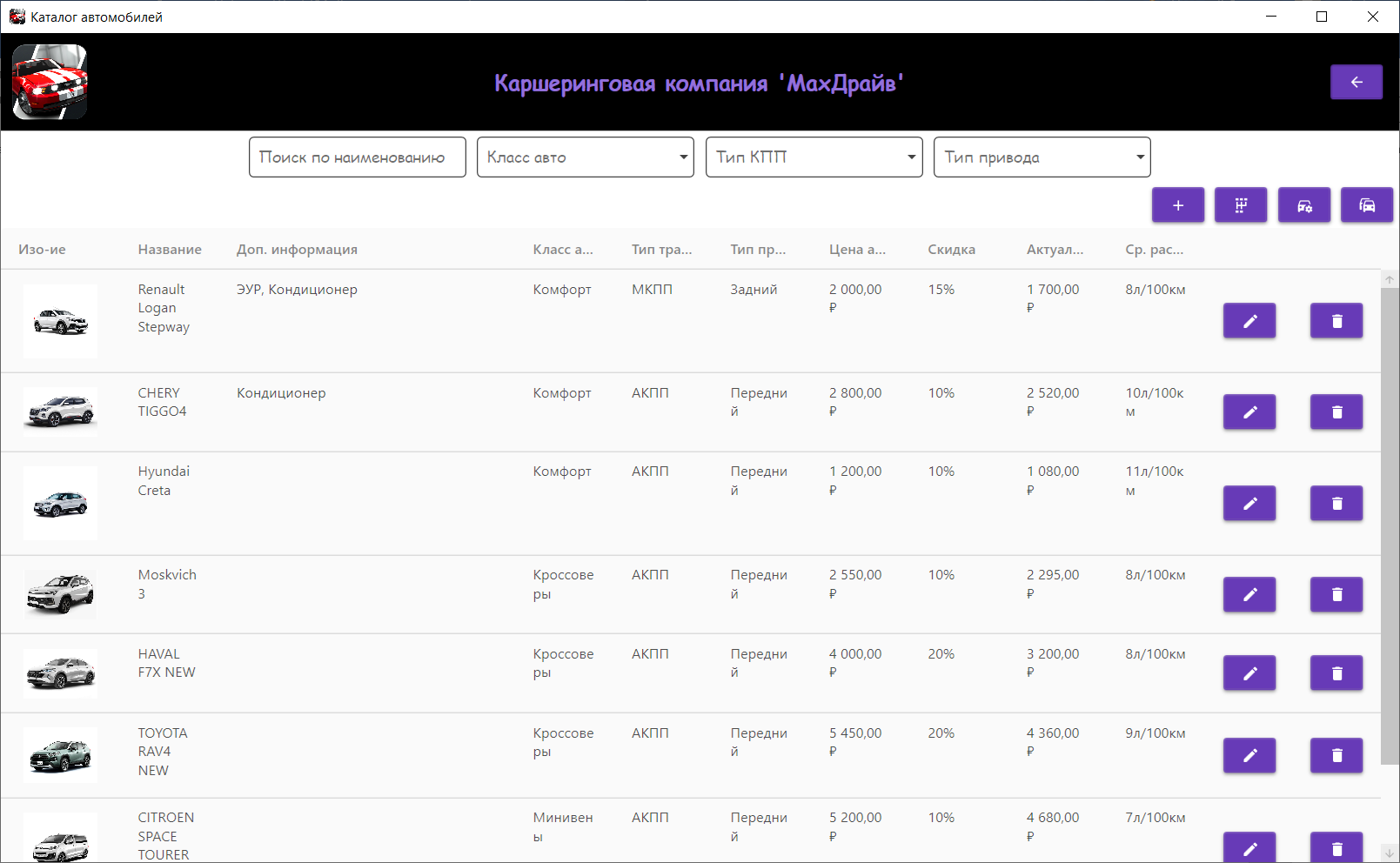
TextBox

PasswordBox

Рис.19 Форма Регистрации

**Форма Автомобили (**Рис.20**)**. При разработке интерфейса этой формы были использованы компоненты:

1. DataGrid – отображение списка автомобилей в виде таблицы;
2. Image – для отображения изображения;
3. ComboBox – для фильтрации автомобилей;
4. Buttons – кнопки для перехода к справочникам, добавления, редактирования и удаления записей;
5. TextBlock – текстовая метка;



ComboBox

Button

DataGrid

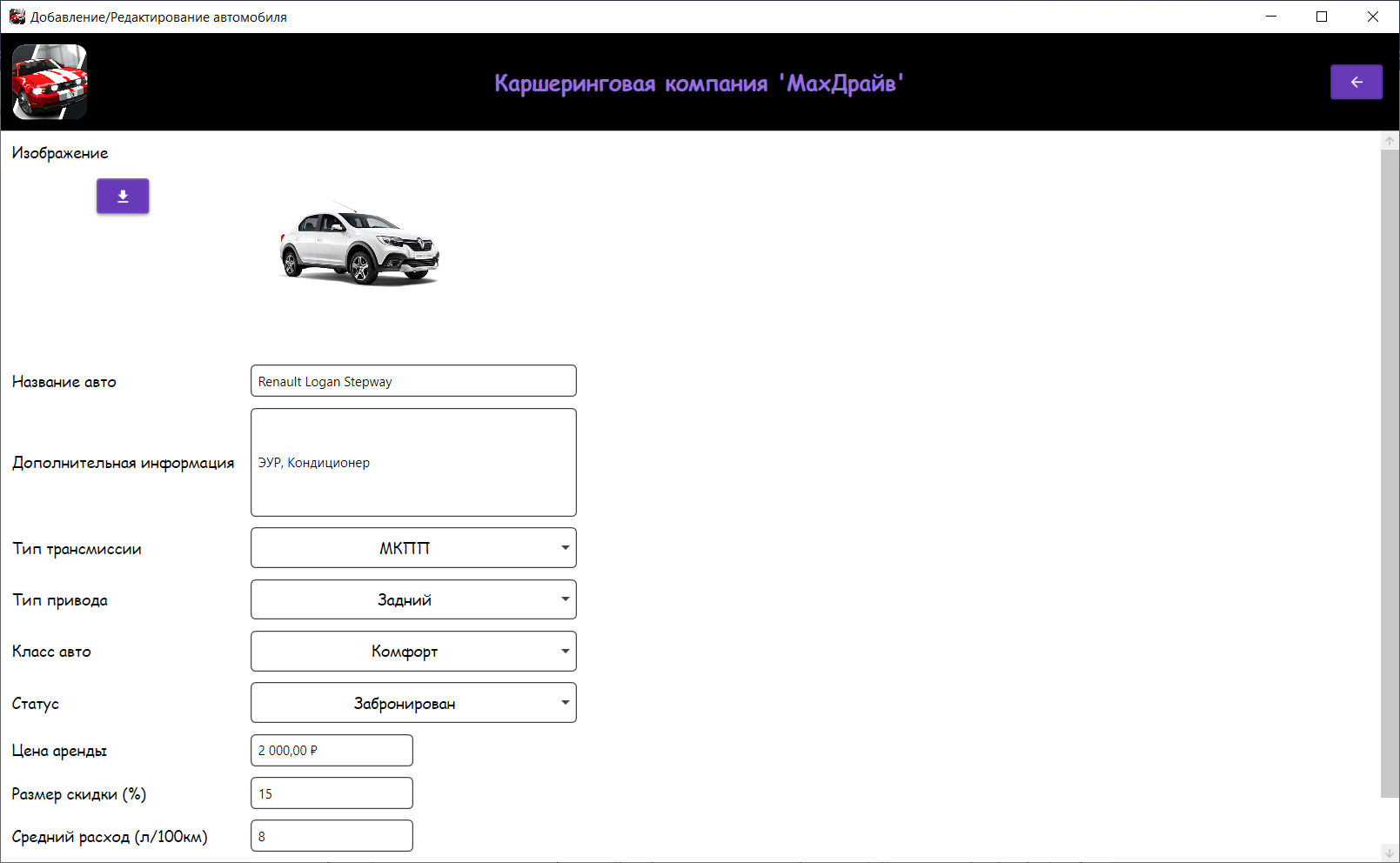
Image

TextBlock

Рис.20 Форма Автомобили

**Страница добавления и редактирования (**Рис.21**)**. При разработке интерфейса этой формы были использованы компоненты:

1. ComboBox – выпадающий список;
2. Buttons – кнопки загрузки и перехода назад;
3. TextBox – поля для ввода текста
4. Image – для отображения изображения;
5. TextBlock – текстовые метки;



Button

Image

TextBlock

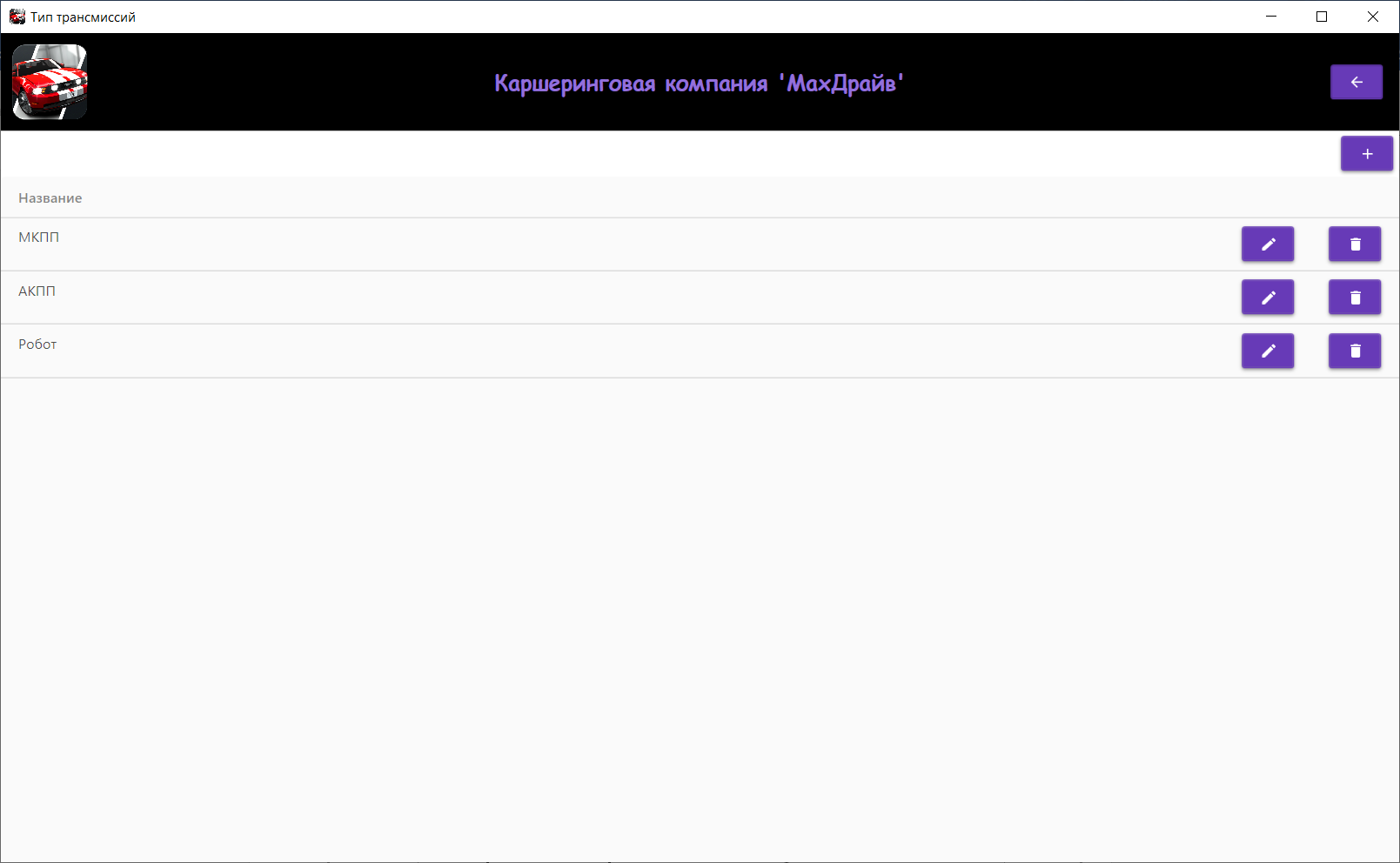
ComboBox

TextBox

Button

Рис.21 Страница добавления и редактирования

**Страница Тип трансмиссий**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.22)



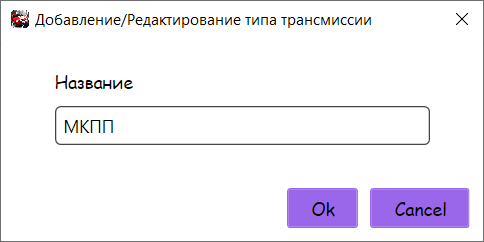
Button

Button

DataGrid

Рис.22 Тип трансмиссий

**Окно Добавление/Редактирование типа трансмиссии**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода и кнопок Button. (Рис.23)

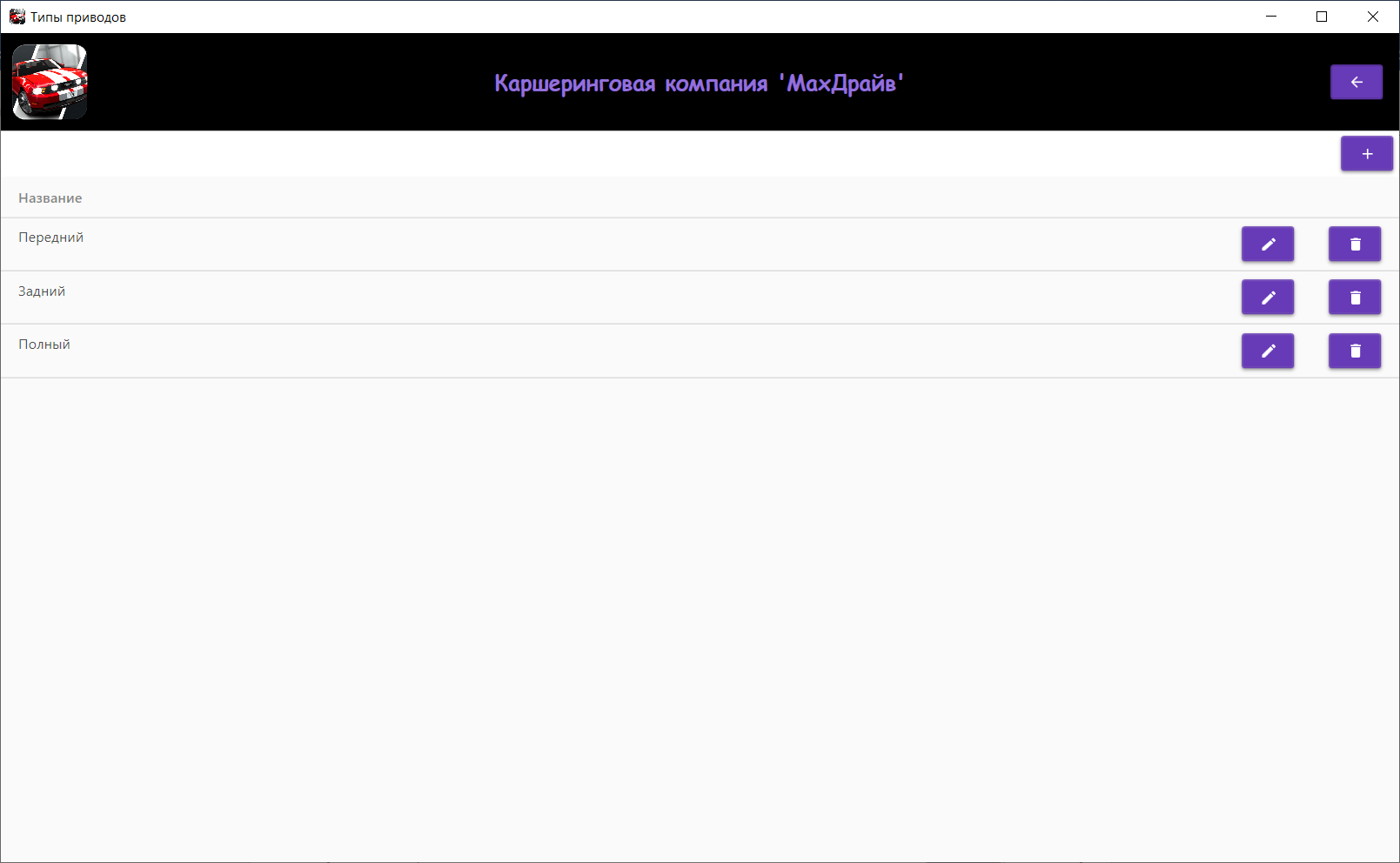


Button

TextBox

Рис.23 Окно Добавление/Редактирование типа трансмиссии

**Страница Типы приводов**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.24)



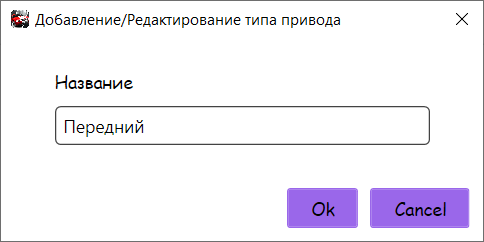
Button

Button

DataGrid

Рис.24 Типы приводов

**Окно Добавление/Редактирование типа привода**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода и кнопок Button. (Рис.25)

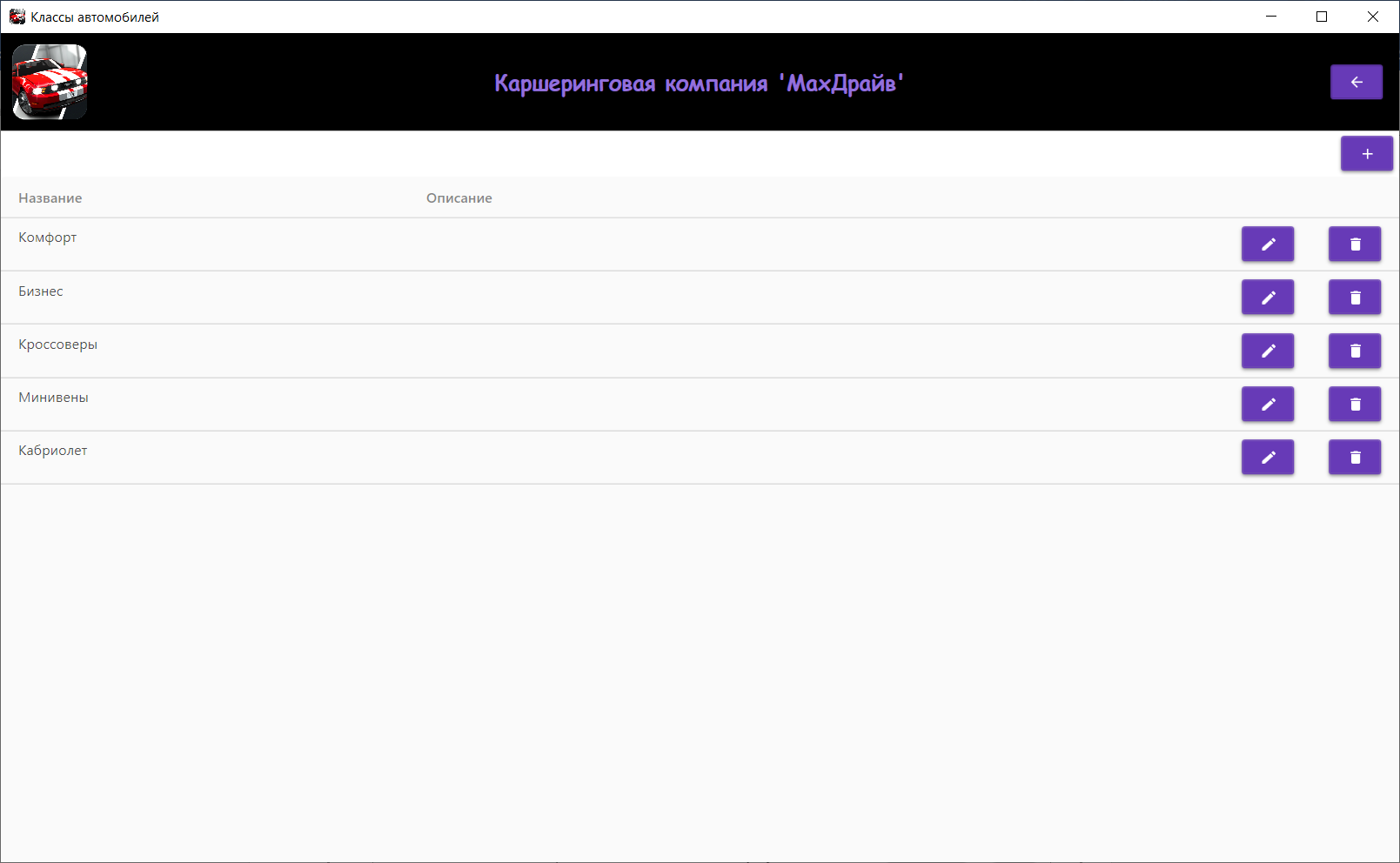


Button

TextBox

Рис.25 Окно Добавление/Редактирование типа привода

**Страница Классы автомобилей**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.26)



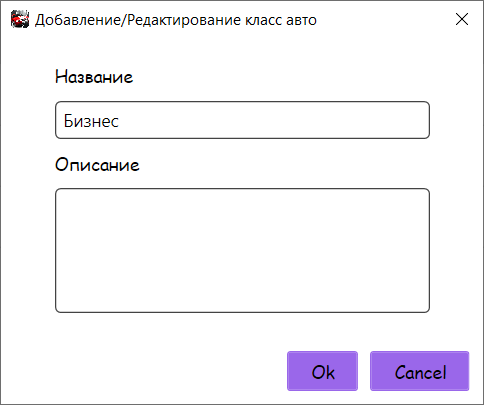
Button

Button

DataGrid

Рис.26 Классы автомобилей

**Окно Добавление/Редактирование класс авто**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода и кнопок Button. (Рис.27)

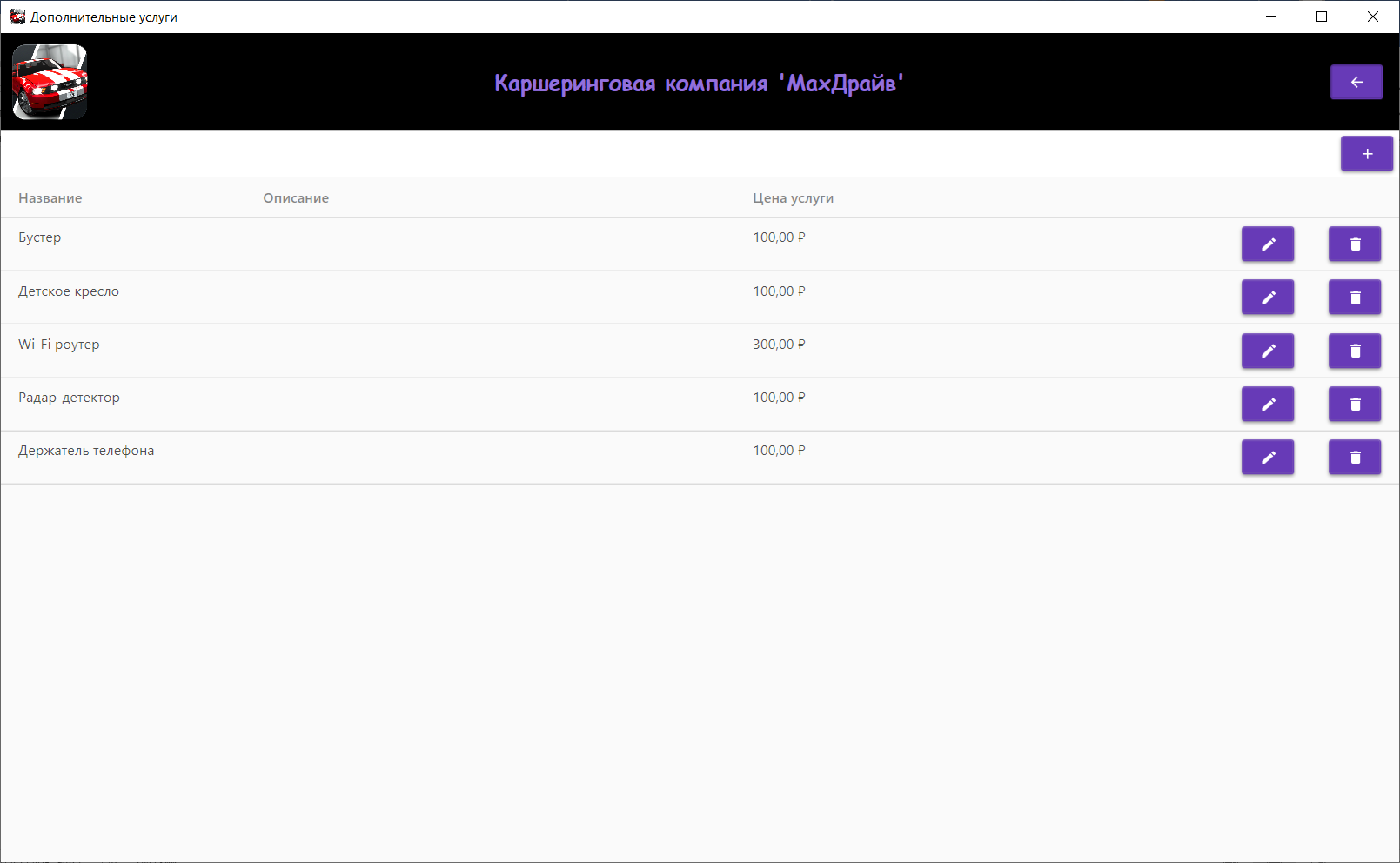


Button

TextBox

Рис.27 Окно Добавление/Редактрование класс авто

**Страница Дополнительные услуги**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.28)



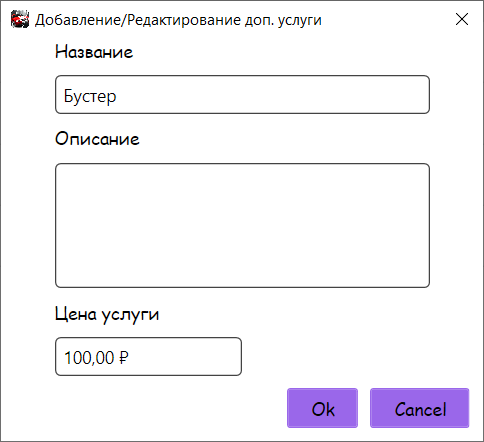
Button

Button

DataGrid

Рис.28 Дополнительные услуги

**Окно Добавление/Редактирование доп.услуги**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода и кнопок Button. (Рис.29)

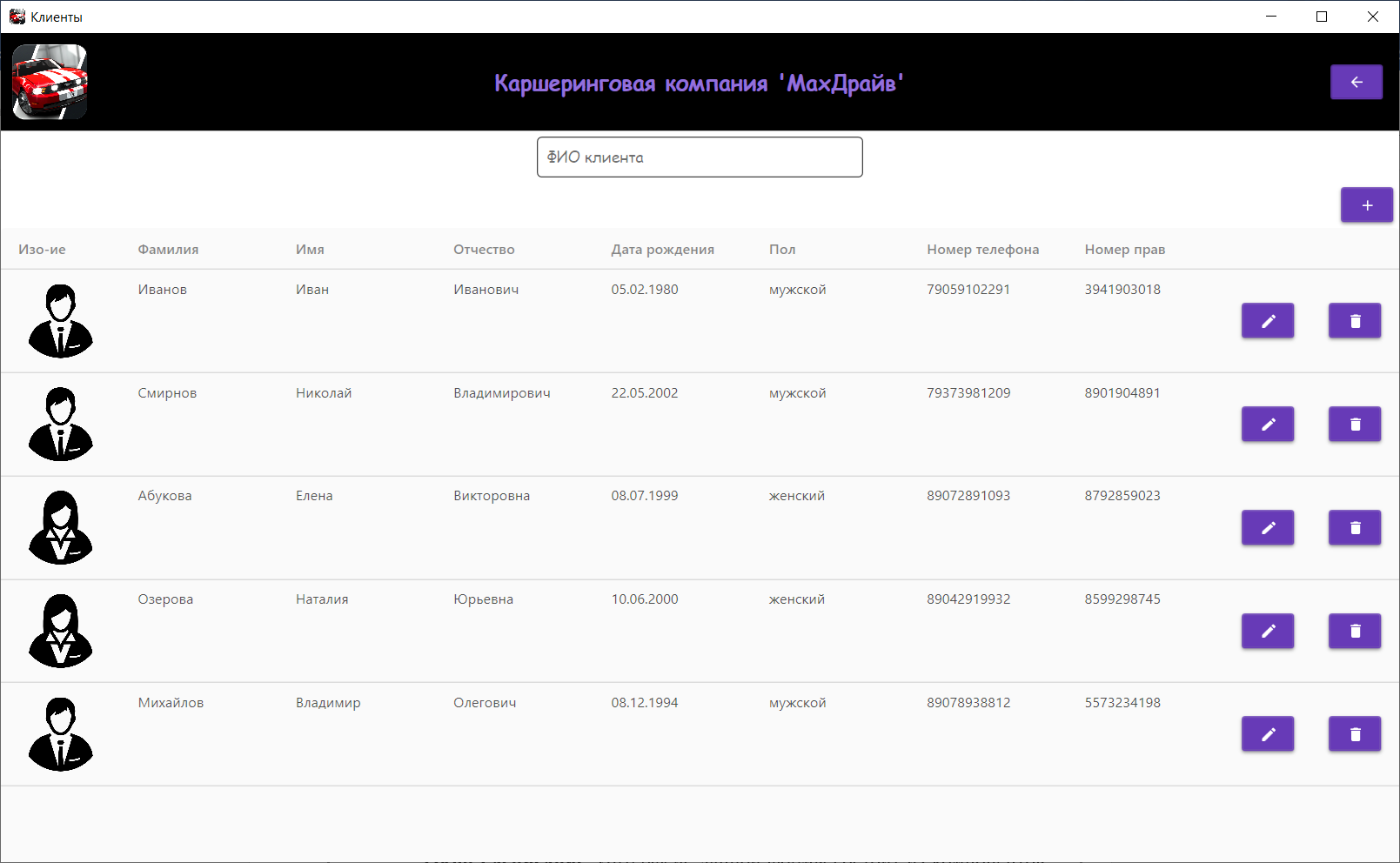


Button

TextBox

Рис.29 Окно Добавление/Редактрование доп. услуги

**Страница Клиенты**. Интерфейс данной формы состоит из компонентов: DataGrid, TextBox, и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.30)



TextBox

Button

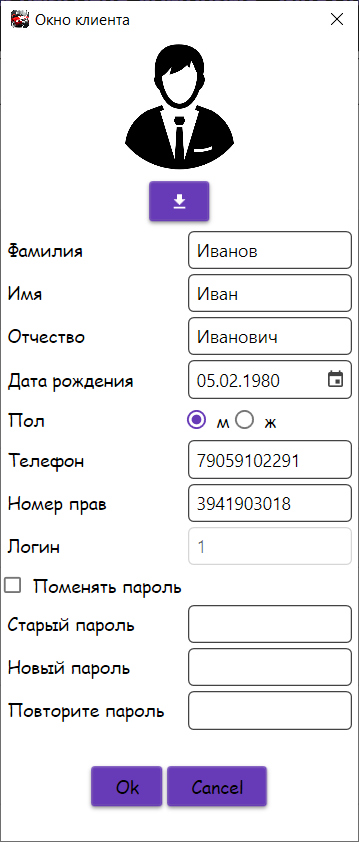
Button

Image

DataGrid

Рис.30 Страница Клиенты

**Окно Окно клиента**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода, DatePicker – для ввода даты рождения, RadioButton – для выбора пола, CheckBox – флаг для смены пароля и кнопок Button. (Рис.31)



CheckBox

RadioButton

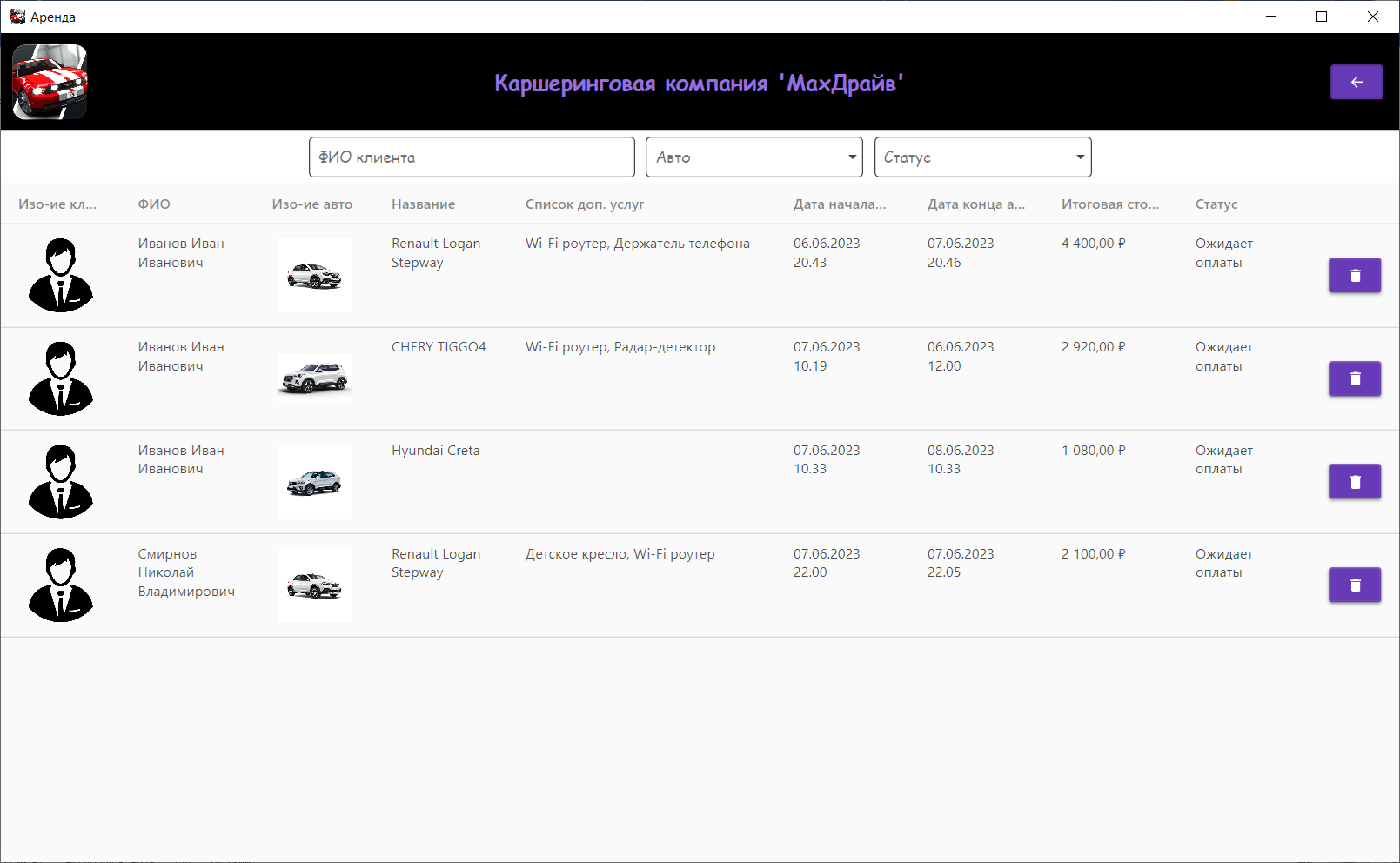
Buttons

TextBox

PasswordBox

Рис.31 Окно Окно клиента

**Страница Аренда**. Интерфейс данной формы состоит из компонентов: DataGrid, Combobox, и кнопок: Удалить, Назад. (Рис.32)



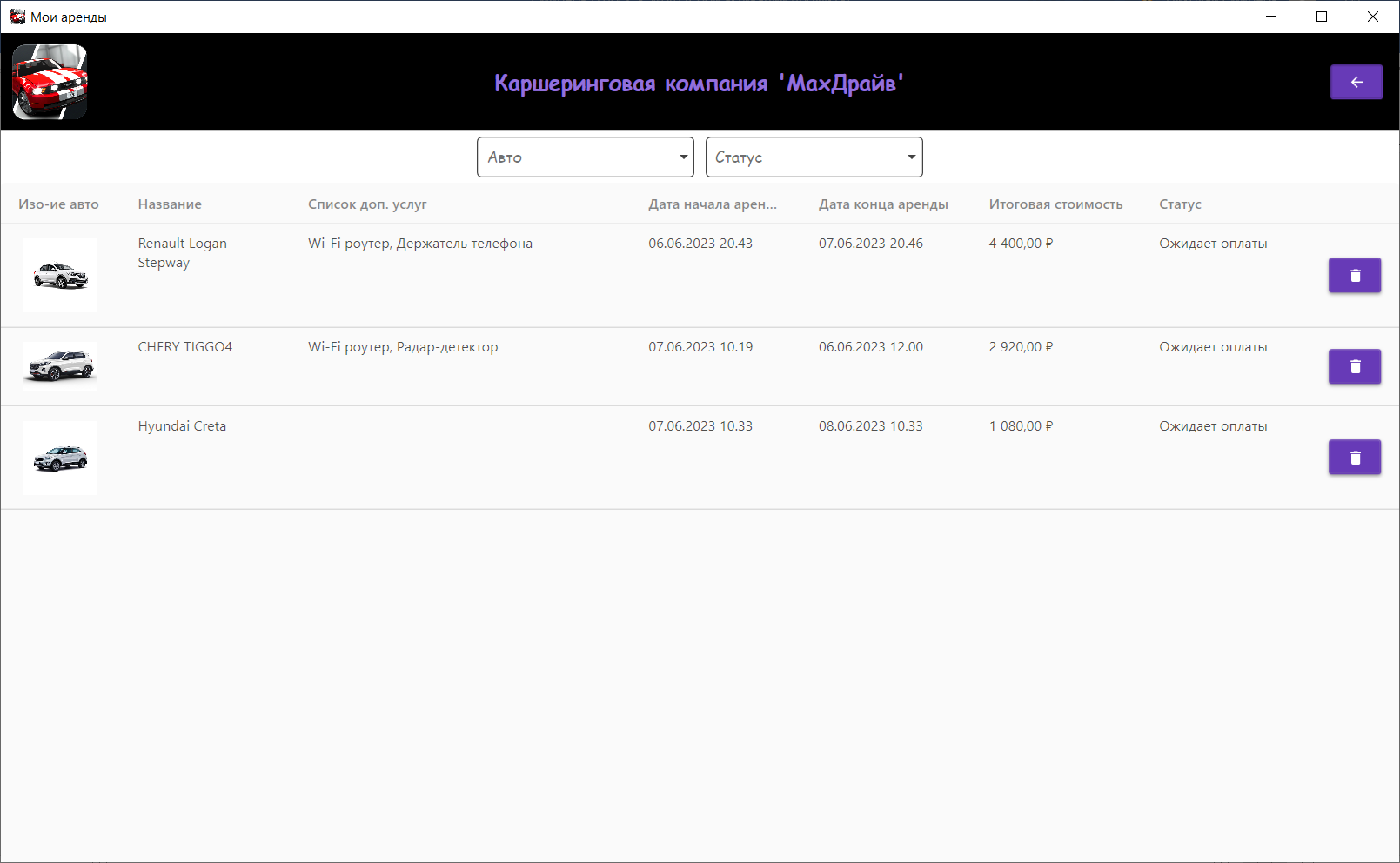
Button

DataGrid

ComboBox

Рис.32 Аренда

**Страница Мои аренды**. Интерфейс данной формы состоит из компонентов: DataGrid, Combobox, и кнопок: Отменить, Назад. (Рис.33)



Button

DataGrid

ComboBox

Рис.33 Мои аренды

**3.7 Описание структуры выходной информации**

Выходными данными являются список авомобилей.

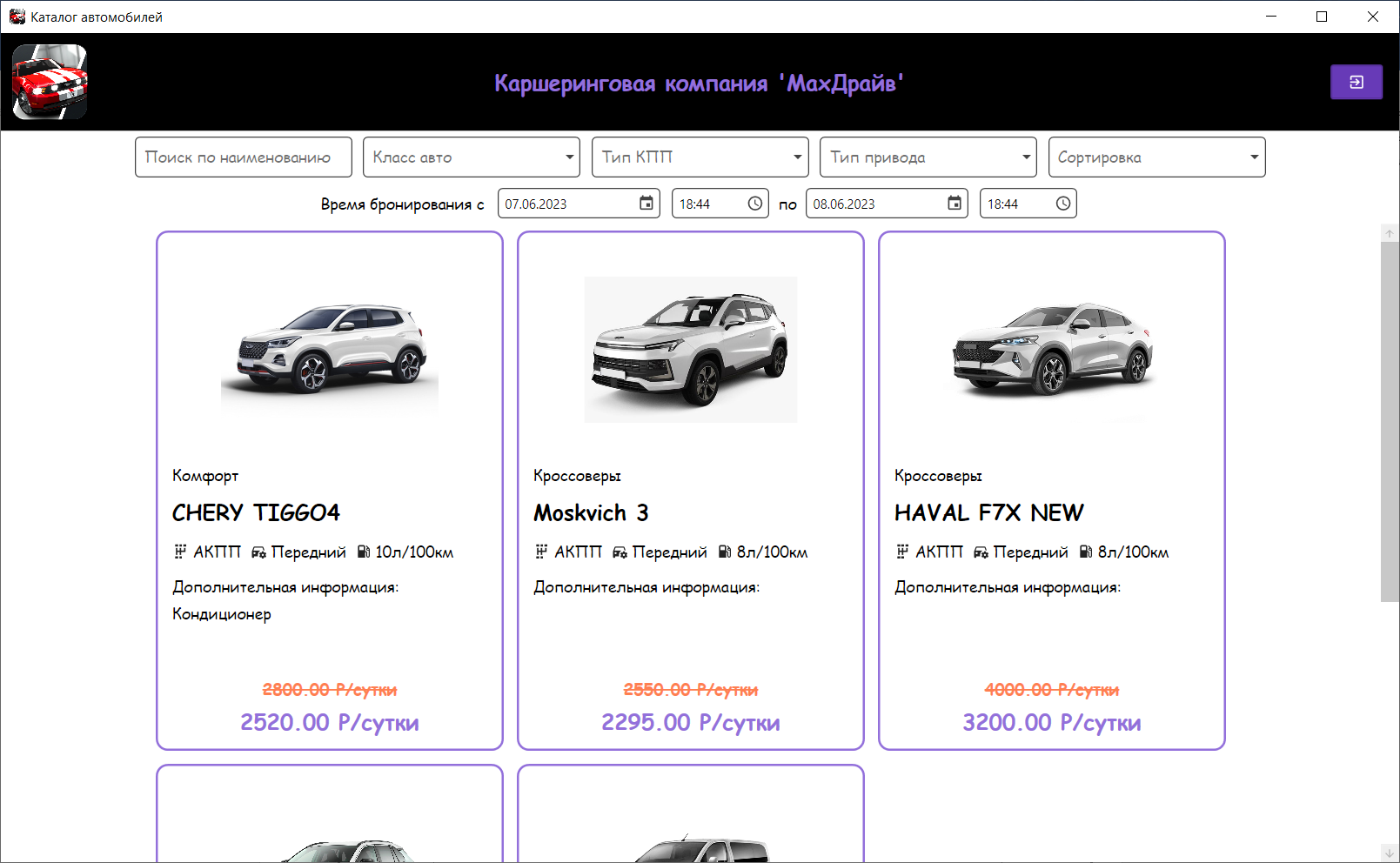


Рис.34 Каталог автомобилей

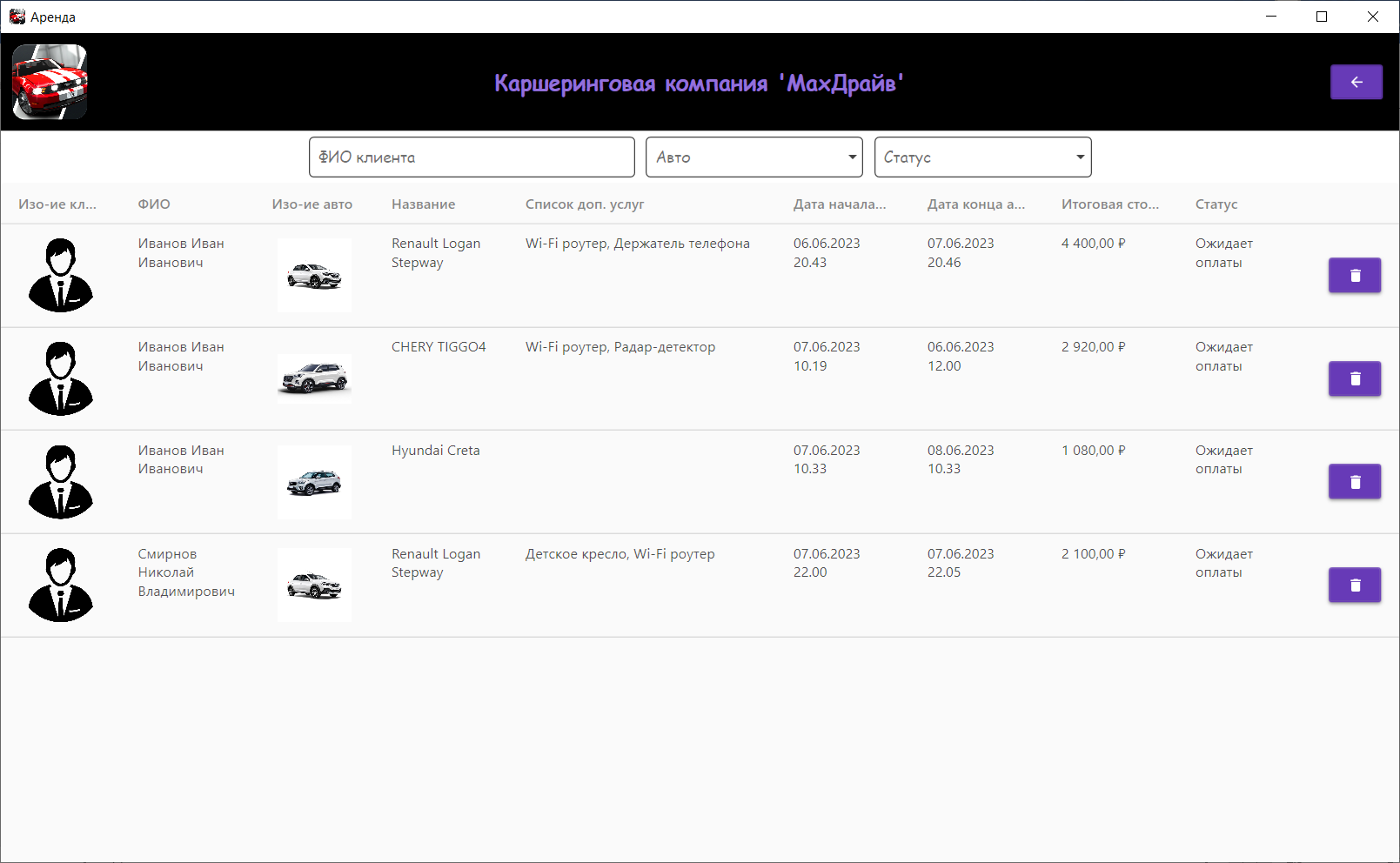


Рис.35 Аренда

**4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**4.1 Назначение программного средства**

Назначением программного средства является автоматизация деятельности компании MaxДрайв.

**4.2 Условия выполнения программного средства**

Условия, при соблюдении которых возможно выполнение работы с программным продуктом:

-Компьютер с установленной на нём «CarsharingApp».

Подготовительные действия:

-На иконке «CarsharingApp» рабочего стола произвести двойной щелчок левой кнопкой мыши.

**4.3 Эксплуатация программного средства**

Этот раздел содержит обзор приложений «CarsharingApp» и ее основных параметров.

Активировать копию программы CarsharingApp можно сразу после её установки, когда появится окно подтверждения подлинности Вашей версии. Ключ активации расположен внутри упаковки программного продукта. Для успешного выполнения активации ключ активации необходимо вводить именно в том виде, в котором он предоставлен.

Вводные сведения об интерфейсе пользователя

После запуска приложения отображается стартовое окно приложения (Рис.36). Главная форма представляет собой каталог автомобилей в виде плиточного интерфейса. Каждый автомобиль отображается в виде карточки с краткой информацией: название и фотография. Пользователь имеет возможность отфильтровать список по следующим критериям: класс авто, тип КПП и тип привода.

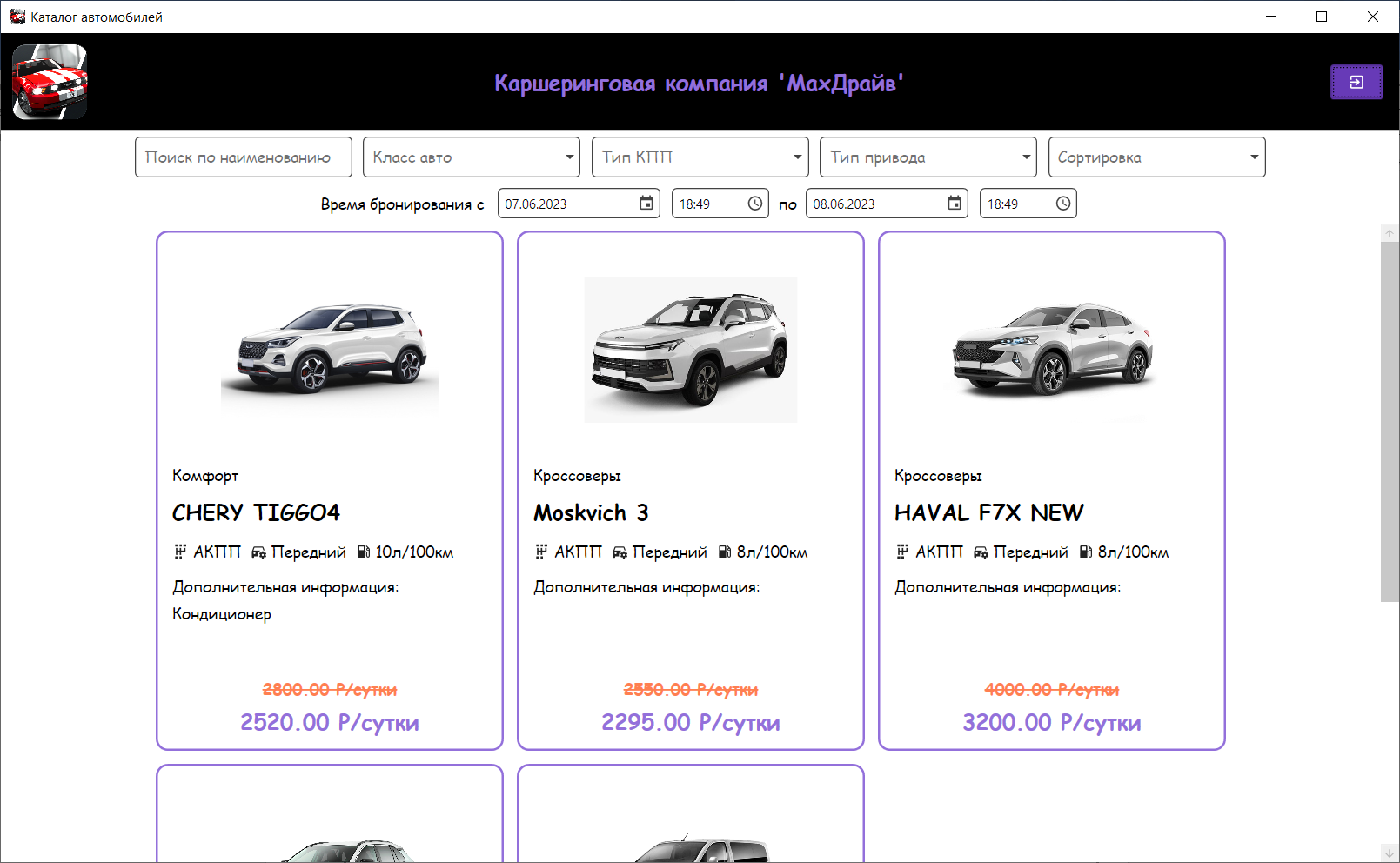


Рис.36 Окно Автоматизированная информационная система «Каршеринговая компания MaxДрайв»

Кликнув по кнопке «Закрыть» программа потребует подтверждение. (Рис.37)

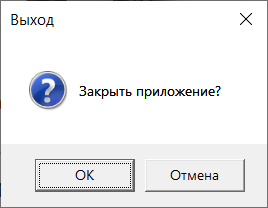


Рис.37 Сообщение программы

Для входа в систему необходимо кликнуть на кнопку . Ввести логин и пароль администратора или логин и пароль пользователя и нажать OK(Рис. 38).

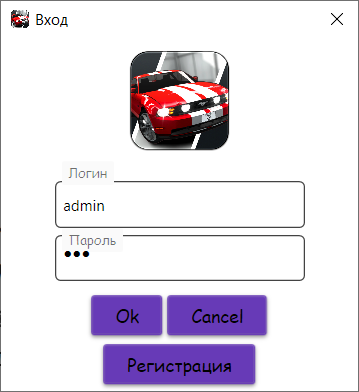


Рис.38 Форма входа в систему

Если вы являетесь новым пользователем в системе, то необходимо нажать на кнопку Регистрация. В появившейся форме требуется заполнить обязательные поля. При регистрации роль нового пользователя всегда клиент (Рис. 39).

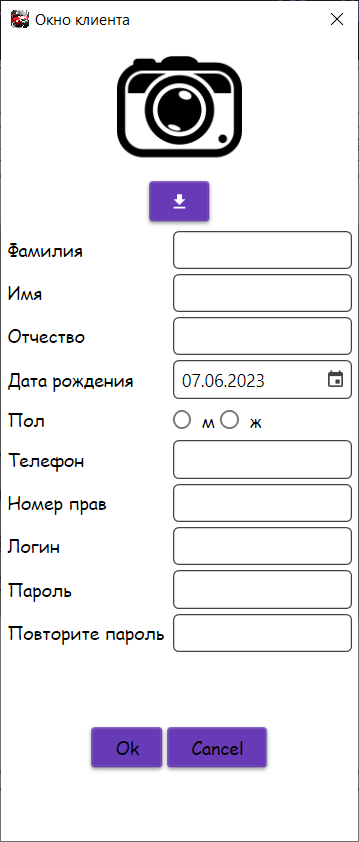


Рис.39 Форма регистрации

После входа в систему появятся кнопки для доступа к различным компонентам системы.

Для того чтобы выйти из системы нажмите на кнопку , программа потребует потдверждения(Рис.40)

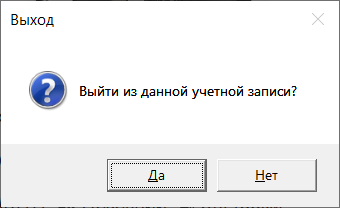


Рис.40 Сообщение программы

Ниже описаны пункты меню управления и информации.

Страница «Автомобили». Эта форма доступна только администратору. Для того, чтобы открыть эту форму нужно нажать на кнопку «Автомобили» , расположенную на главной форме приложения. Данная страница предназначена для отображения информации о автомобилях в табличном виде, открытия страницы редактирования, добавления, удаления выделенной записи, а также для перехода на другие страницы приложения. (Рис.41)

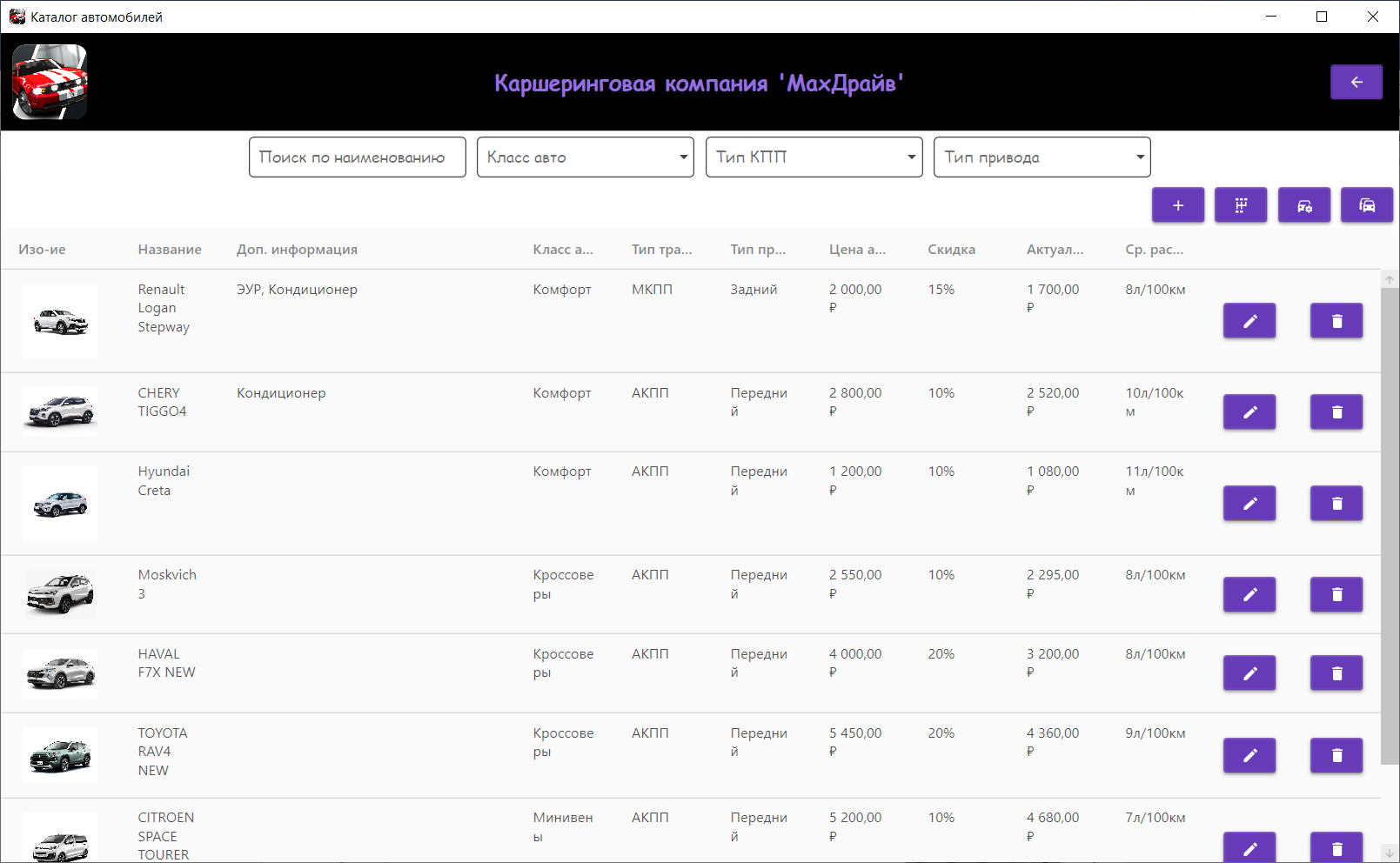


Рис.41 Страница Автомобили

Страница Добавление и редактирование. (Рис.42) Для открытия этой страницы нужно на странице «Автомобили» выбрать запись и нажать на кнопку Редактировать. Также данную страницу можно открыть в режиме добавления, нажав на соотвестсвующую кнопку на предыдущей странице. Эта страница предназначена для добавления или редактирования сведений о автомобилях.

 Рис.42 Страница Добавление и редактирование

«Тип трансмиссий»: эта страница предназначена для просмотра и удаления информации о типах трансмиссий. Добавление и редактирование осуществляется через дополнительную форму. (Рис.44)

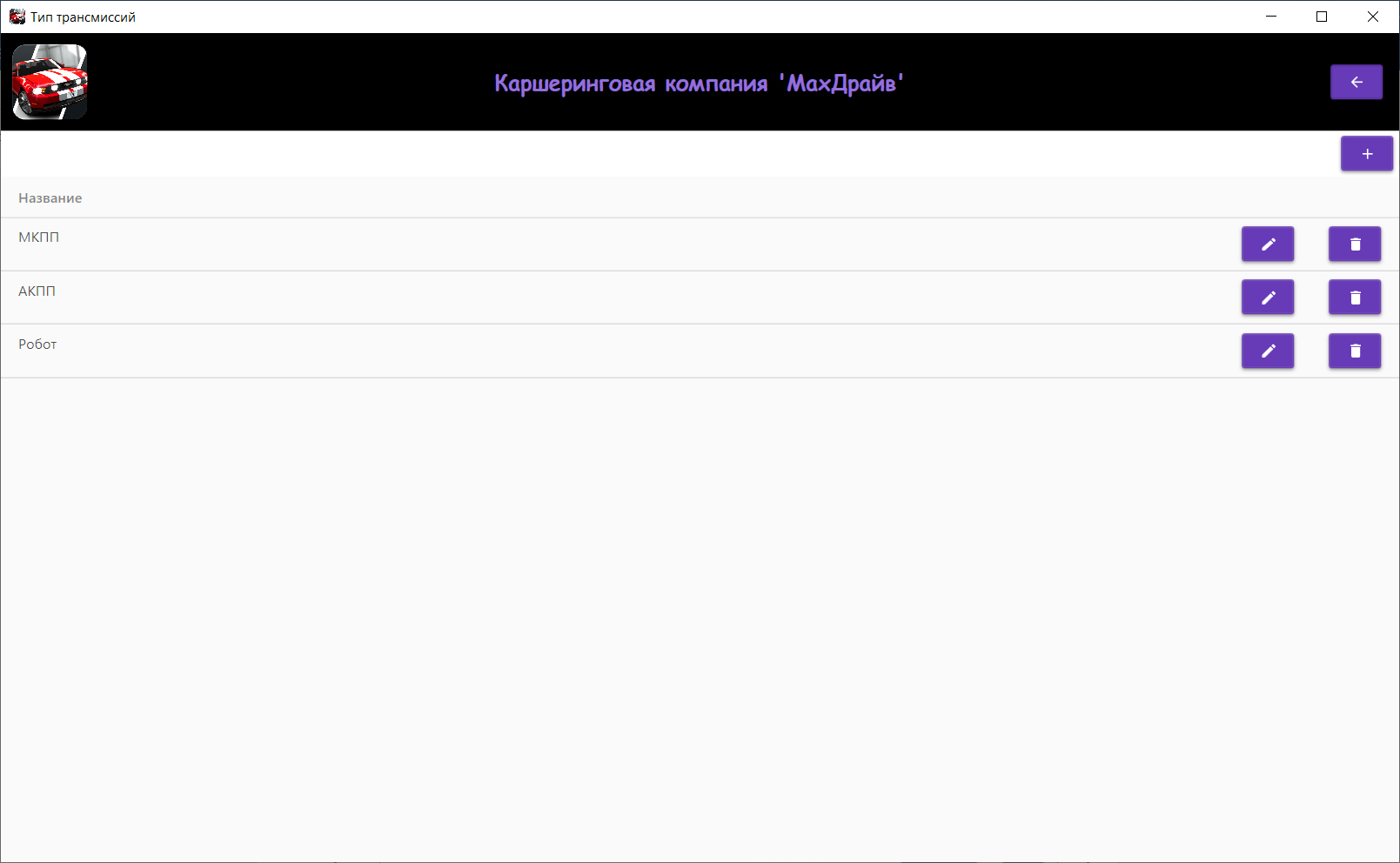


Рис.43 Тип трансмиссий

«Добавление/Редактирование типа трансмиссии»: эта форма предназначена для добавления и редактирования записи. (Рис.44)

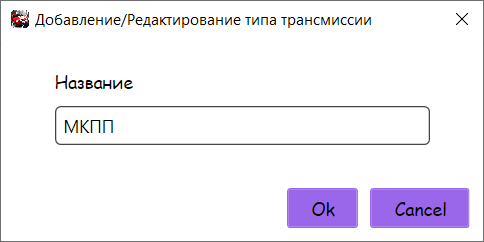


Рис.44 Добавление/Редактирование типа трансмиссии

«Типы приводов»: эта страница предназначена для просмотра и удаления информации о типах приводов. Добавление и редактирование осуществляется через дополнительную форму. (Рис.46)

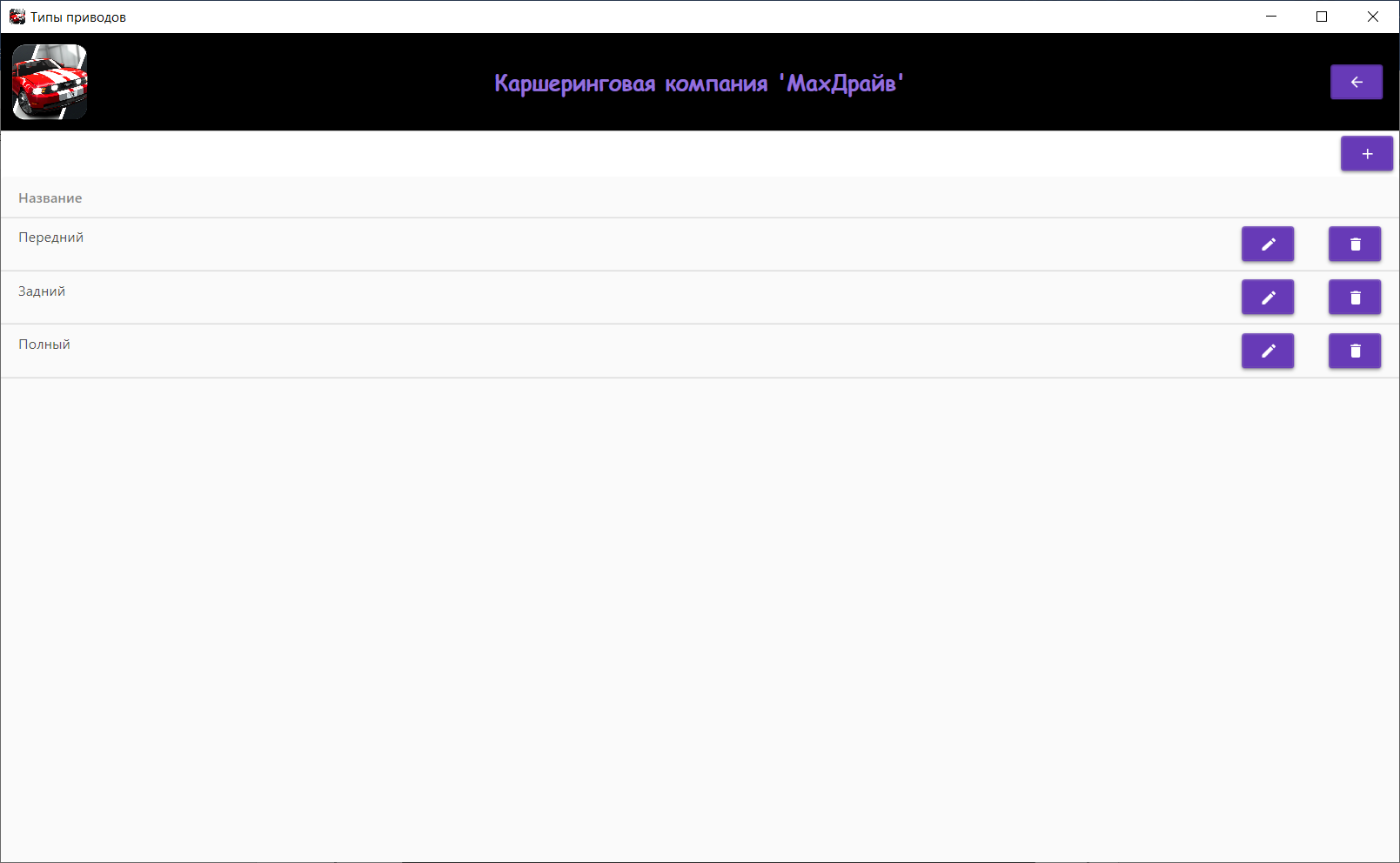


Рис.45 Категории квестов

«Добавление/Редактирование типа привода»: эта форма предназначена для добавления и редактирования записи. (Рис.46)

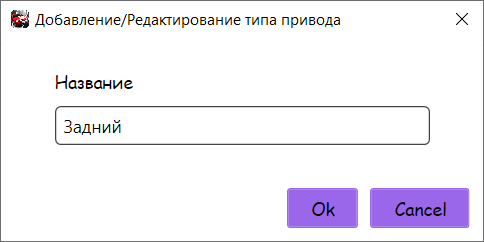


Рис.46 Добавление/Редактирование типа привода

«Классы автомобилей»: эта страница предназначена для просмотра и удаления информации о классах автомобилей. Добавление и редактирование осуществляется через дополнительную форму. (Рис.48)

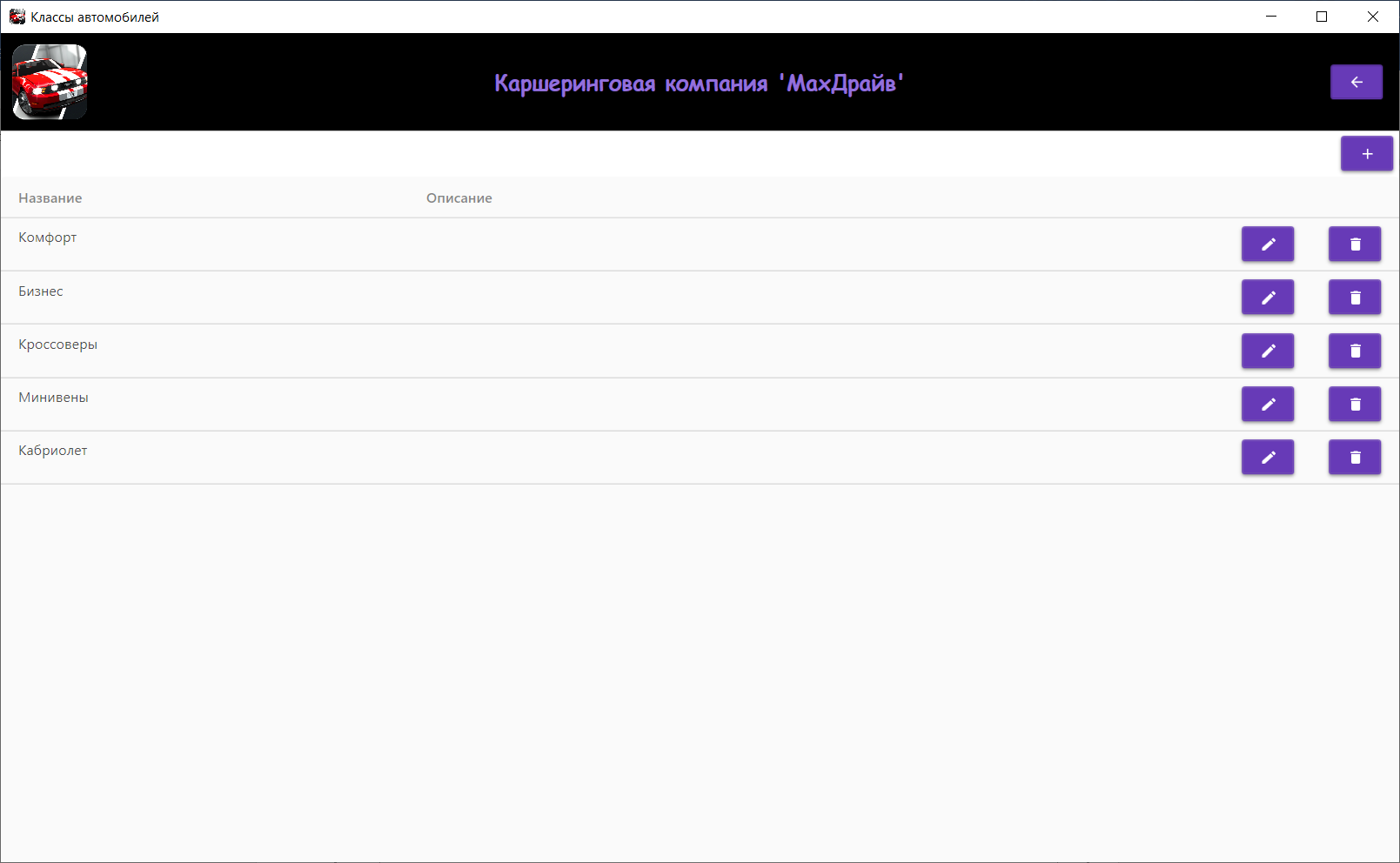


Рис.47 Классы автомобилей

«Добавление/Редактирование класс авто»: эта форма предназначена для добавления и редактирования записи. (Рис.48)

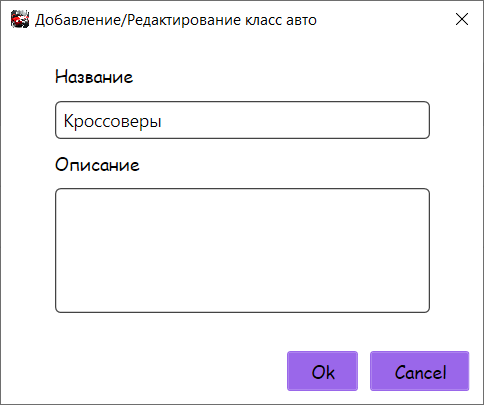


Рис.48 Добавление/Редактирование класс авто

«Клиенты»: эта страница предназначена для просмотра и удаления информации о клиентах. Добавление и редактирование осуществляется через дополнительную форму. (Рис.50)

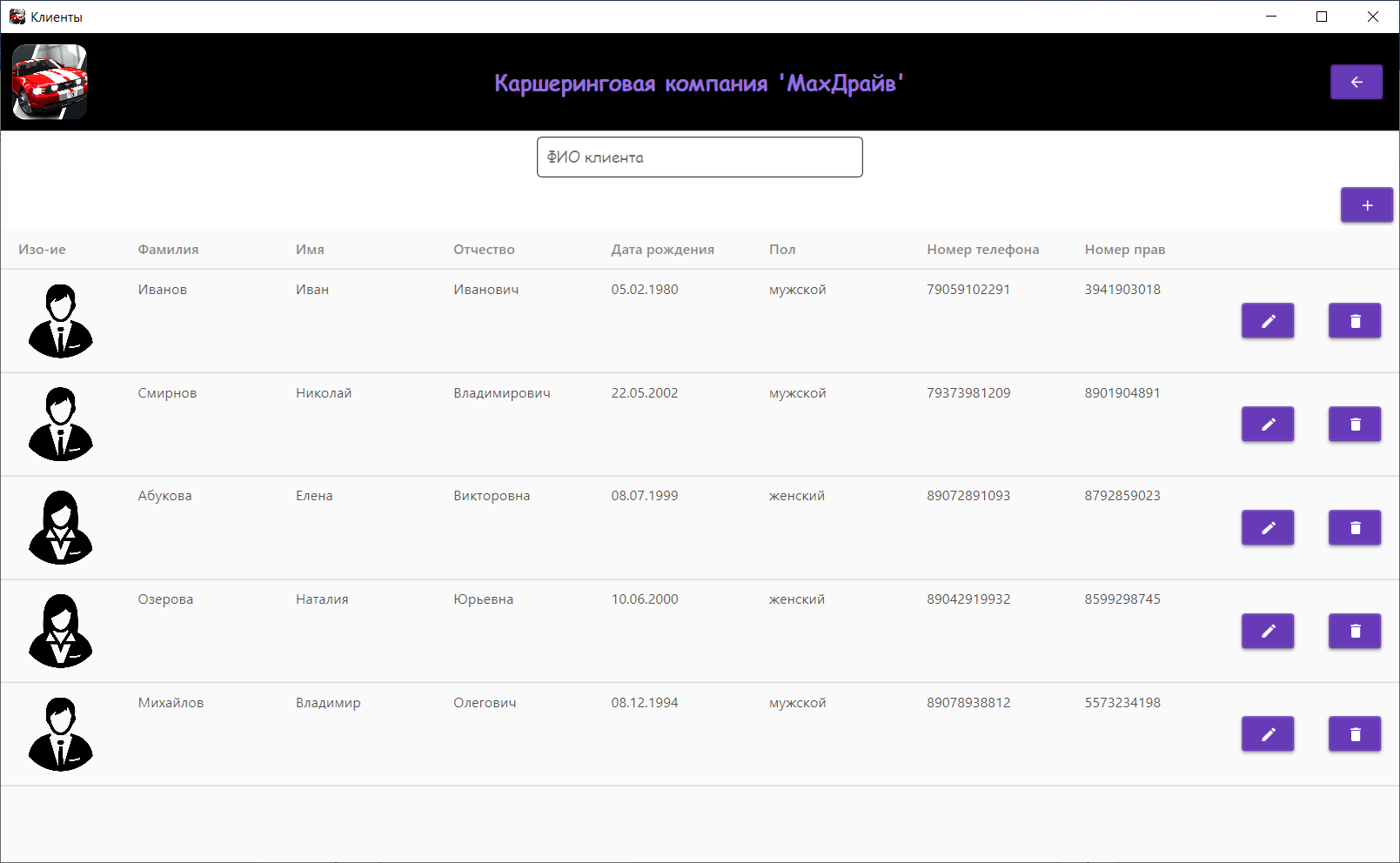


Рис.49 Страница Клиенты

«Окно клиента»: эта форма предназначена для добавления и редактирования записи о клиенте. (Рис.50)

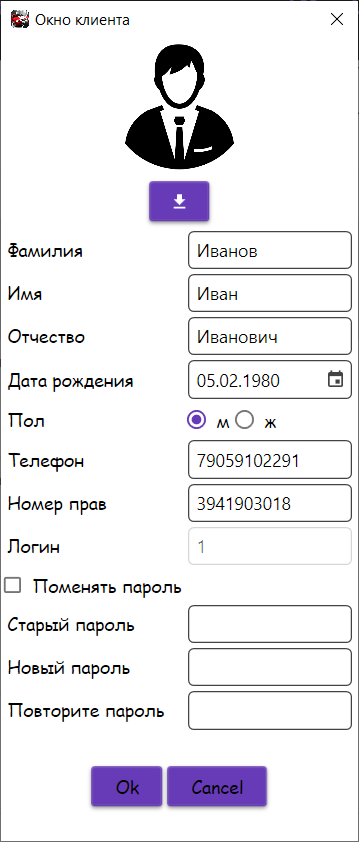


Рис.50 Форма Окно клиента

«Дополнительные услуги»: эта страница предназначена для просмотра и удаления информации о дополнительных услугах. Добавление и редактирование осуществляется через дополнительную форму. (Рис.52)

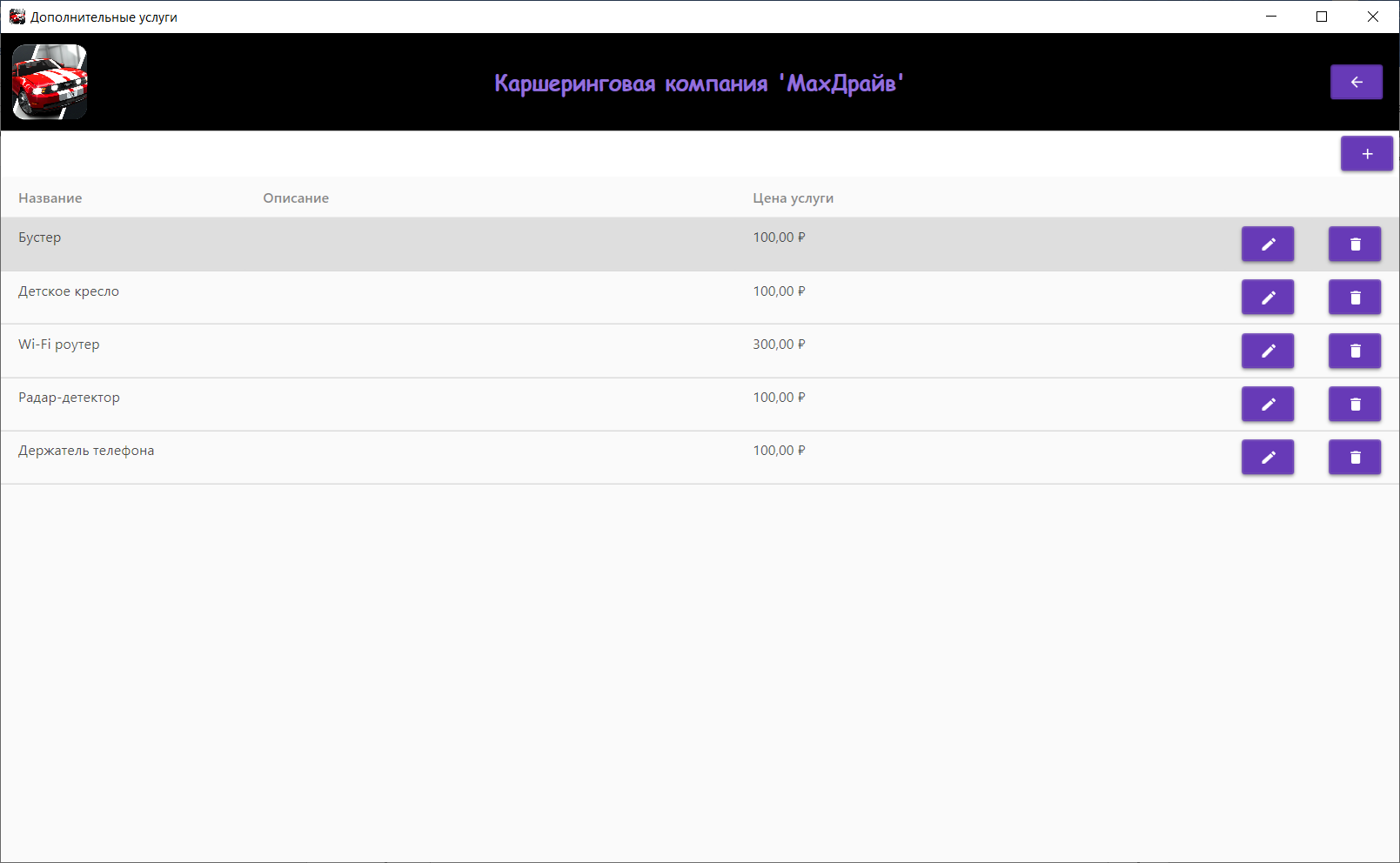


Рис.51 Дополнительные услуги

«Добавление/Редактирование класс авто»: эта форма предназначена для добавления и редактирования записи. (Рис.52)

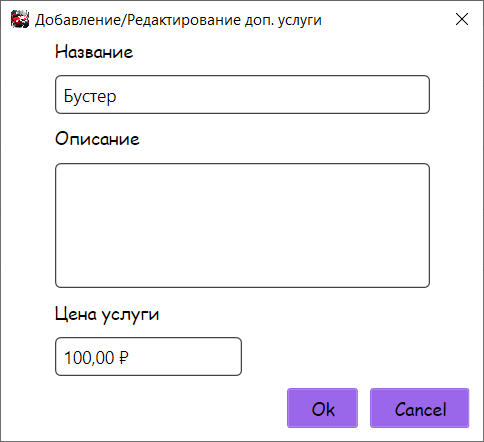


Рис.52 Добавление/Редактирование класс авто

«Аренда»: эта страница предназначена для просмотра аренды и удаления информации. (Рис.53)



Рис.53 Страница Аренда

«Аренда автомобиля»: эта форма предназначена для аренды автомобиля. (Рис.54)

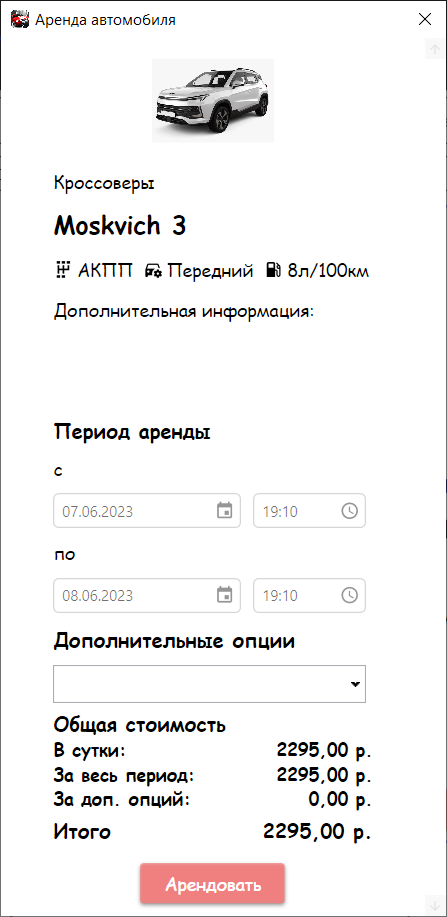


Рис.54 Форма Аренда автомобиля

«Мои аренды»: эта форма предназначена для просмотра и отмены аренды ( бронирования ) автомобилей. (Рис.55)



Рис.55 Мои аренды

Работа с CarsharingApp

После запуска программы Вы перейдете на главную форму программы. В формах «Автомобили», «Типы приводов», «Тип трансмиссии», «Класс автомобиля», «Дополнительные услуги», заполняем все необходимые для работы пункты.

После этого кликаем по кнопке «Ок».

Если Вы желаете удалить любую введенную ранее информацию, Вам необходимо кликнуть по нужной кнопке.

Для того чтобы изменить запись, введите в редактируемые поля данные и нажмите кнопку Ок. Данные формы редактированию и управлению со стороны пользователя не подлежат.

После того, как Вы закончите сеанс работы с программой, если Вы хотите выйти из CarsharingApp кликните на соответствующей кнопке «Выход».

Основные кнопки CarsharingApp

Кнопки управления «Добавить запись», «Удалить запись», «Сохранить» расположены на каждой конкретной форме в базе данных.

Действия этих кнопок приведены в таблице 1:

Таблица 1 – Действие кнопок

|  |  |
| --- | --- |
| Добавить запись | После нажатия на эту кнопку выйдет вспомогательная форма для добавления записей |
| Удалить запись | Кнопка «Удалить запись» позволяет удалить выделенную запись. |
| Сохранить данные | Кнопка «Сохранить данные» добавляет запись введенную пользователем в БД, либо обновляет существующую. |
| Изменить | Открывает форму в режиме редактирования информации |
|  | Позволяет Загрузить файл |
|  | Переход на предыдущую страницу |

**4.4 Сообщения пользователю**

Для уверенной и комфортной работы пользователю важно понимать поведение программы, а программе, в свою очередь, поведение пользователя. Для информирования и уточнения намерений пользователей в программах используются сообщения.

При попытке удалить запись, пользователю выводится сообщение с запросом на подтверждение или отмену действия. (Рис.56)

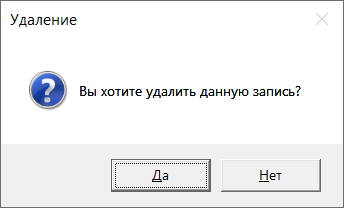


Рис.56 Сообщение Подтверждение удаления

После добавления новой записи, пользователю выводится сообщение. (Рис.57)

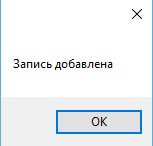


Рис.57 Сообщение Запись добавлена

После обновления записи, пользователю выводится сообщение. (Рис.58)

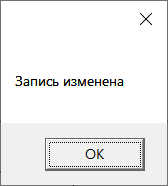


Рис.58 Сообщение Запись изменена

При попытке удалить связанные записи, программа выводит пользователю сообщение о невозможности действия. (Рис.59)

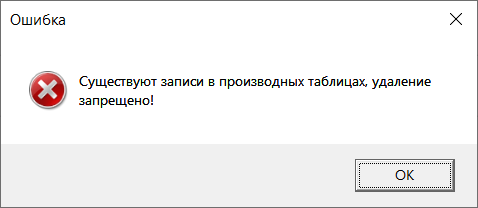


Рис.59 Сообщение Ошибка удаления

**5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА**

**5.1 Определение верхней границы количества тестов**

Для разработанного алгоритма был построен граф.

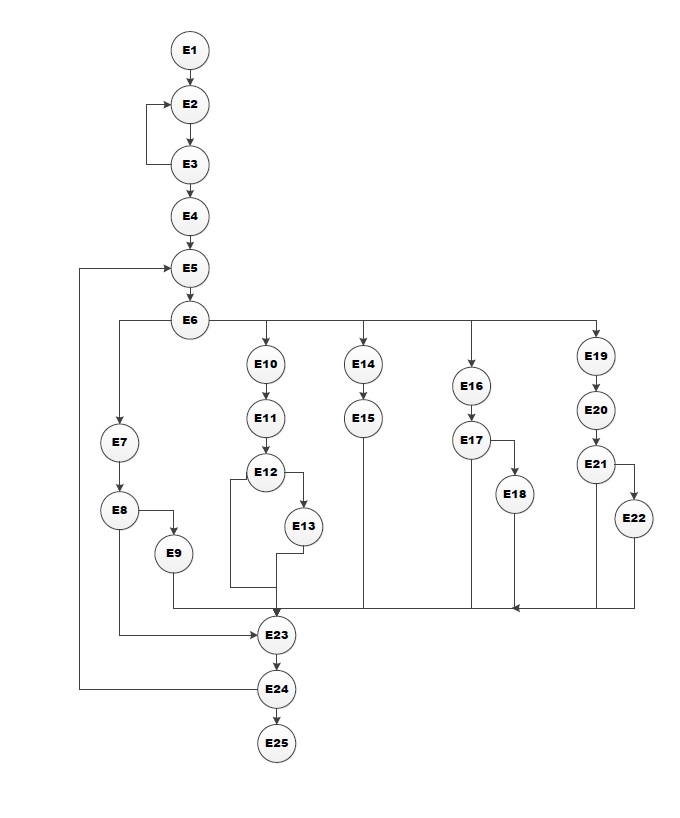
****

Рис.65 Потоковый граф

Необходимое количество тестов определяется по формуле V(G) = E-N+2, где E – это число ребер, а N–это число вершин данного графа. Согласно представленному графу E=34, N=25. Далее по формуле рассчитаем необходимое количество тестовых сценариев (тест-кейсов).

V(G)=34-25+2=11.

Для данного программного средства необходимо разработать не менее 11 тестовых сценариев.

Чтобы определить количество тестовых прогонов для программного продукта CarsharingApp необходимо выделить требования. Из требований к приложению выделим поддержку 2-х операционных систем с одним основным языком локализации и выполнения 6-ти основных функций. Кроме того, приложение позволяет выполнять 5 функций по взаимодействию с системой (запуск приложения, выход из приложения, сохранение результатов в файл, работа с буфером и т.п.).

Таким образом, полное покрытие требований задаёт набор из 2\*1\*(6+5) =22 тестовых прогонов.

**5.2 Разработка тестовых вариантов**

Таблица 2 – Тестирование корректного добавления информации в Дополнительные услуги

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Доп. услуги | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Доп. условия | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = Тест  Описание = Описание1  Цена = 100 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Доп. условия открыта | | Отображается форма Доп. условия |  |

Таблица 3 – Тестирование некорректного добавления информации в таблицу Дополнительные условия

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Доп. условия | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Доп. условия | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название =  Описание = Описание1  Цена = 100 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение “Введите название доп. условия” | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Доп. условия открыта | | Отображается форма Доп. условия |  |

Таблица 4 – Тестирование корректного редактирования информации в таблице Дополнительные условия

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Доп. условия | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Доп. условия | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название = Название1  Описание = Описание1  Цена = 100 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Доп. условия  открыта | | Отображается форма Доп. условия |  |

Таблица 5 – Тестирование некорректного редактирования информации в таблице Дополнительные условия

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Доп. условия | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Доп. условия | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название =  Описание = Описание1  Цена = 100 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение “Введите название доп. условия” | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Доп. условия  открыта | | Отображается форма Доп. условия |  |

Таблица 6 – Тестирование корректного добавления информации в таблицу Автомобили

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Автомобили | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Автомобили | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название авто = Тест  Тип трансмиссии = МКПП  Тип привода = передний  Класс авто = Комфорт  Статус = Свободен  Цена аренды = 1200  Размер скидки = 0  Средний расход = 10 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Автомобили открыта | | Отображается форма Автомобили |  |

Таблица 7 – Тестирование некорректного добавления информации в таблицу Автомобили

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Автомобили | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Автомобили | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название авто =  Тип трансмиссии = МКПП  Тип привода = передний  Класс авто = Комфорт  Статус = Свободен  Цена аренды = 1200  Размер скидки = 0  Средний расход = 10 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение “Введите название автомобиля” | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Автомобили  открыта | | Отображается форма Автомобили |  |

Таблица 8 – Тестирование корректного редактирования информации в таблице Автомобили

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Автомобили | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Автомобили | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название авто = Ниссан  Тип трансмиссии = МКПП  Тип привода = передний  Класс авто = Комфорт  Статус = Свободен  Цена аренды = 1200  Размер скидки = 0  Средний расход = 10 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Автомобили открыта | | Отображается форма Автомобили |  |

Таблица 9 – Тестирование некорректного редактирования информации в таблице Автомобили

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Автомобили | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Автомобили | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название авто =  Тип трансмиссии = МКПП  Тип привода = передний  Класс авто = Комфорт  Статус = Свободен  Цена аренды = 1200  Размер скидки = 0  Средний расход = 10 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение “Введите название автомобиля” | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Автомобили открыта | | Отображается форма Автомобили |  |

Таблица 10 – Тестирование корректного добавления информации в Классы автомобилей

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Классы автомобилей | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Классы автомобилей | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = Комфорт  Описание = Описание1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Классы автомобилей открыта | | Отображается форма Классы автомобилей |  |

Таблица 11 – Тестирование некорректного добавления информации в таблицу Классы автомобилей

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Классы автомобилей | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Классы автомобилей | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название =  Описание = Описание1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение “Введите название класса автомобиля” | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Классы автомобилей открыта | | Отображается форма Классы автомобилей |  |

Таблица 12 – Тестирование корректного редактирования информации в таблице Классы автомобилей

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Классы автомобилей | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Классы автомобилей | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название = Тест  Описание = Описание1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Классы автомобилей открыта | | Отображается форма Классы автомобилей |  |

Таблица 13 – Тестирование некорректного редактирования информации в таблице Классы автомобилей

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Классы автомобилей | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Классы автомобилей | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название =  Описание = Описание1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Сообщение “Введите название класса автомобиля” | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Классы автомобилей открыта | | Отображается форма Классы автомобилей |  |

**5.3 Составление отчета о тестировании**

Для форм проекта были созданы и проведены тесты, результаты сведены в таблицу:

Таблица 14 – Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Прохождение | | |
| Форма | Число ТП | пройдено | | не пройдено |
| Главная | 4 | 4 | | 0 |
| Квест | 4 | 4 | | 0 |
| Категории | 4 | 4 | | 0 |
| Расписание | 4 | 4 | | 0 |
| Бронь | 4 | 4 | | 0 |
| Отзывы | 4 | 4 | | 0 |
| Организаторы | 4 | 4 | | 0 |
| Профиль | 4 | 4 | | 0 |
| ИТОГО | 32 | 36 | 0 | |

По результатам тестирования были выявлены и устранены следующие проблемы:

При выполнении теста таблица №9 Тестирование формы «Автомобили», Некорректный запрос

**Ошибка**

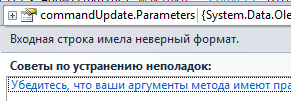


Рис.66 Ошибка теста №8

**Причина**

Поле цена аренды имеет тип вещественное число, было введено строковое значение.

**Принятые меры**

Ошибка устранена. В поле цена аренды можно ввести только цифры.

**ЗАКЛЮЧЕНИЕ**

**Оценка качества программного средства с помощью метрик**

Оценка качества программного средства проводится исходя из стандарта оценки качества ISО 9126.

На компьютере следующей конфигурации:

Установленная операционная система – Windows 10 Professional.

Процессор: Intel Pentium G2030 3.0 GHz.

ОЗУ: 4 Gb

Таблица 15 – Требования к количественным характеристикам качества программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Надежность** |  | | |
| *Завершенность:*  *-* наработка на отказ при отсутствии рестарта; | Часы | 10 | 10 |
| *Устойчивость:*  - относительные ресурсы на обеспечение надежности и рестарта. | % | 0,5 | 0,5 |
| *Восстанавливаемость:*  -длительность восстановления. | Минуты | 2 | 2 |
| *Доступность-готовность:*  -относительное время работоспособного функционирования. | Вероятность | 0,998 | 0,998 |
| **Эффективность** |  | | |
| *Временная эффективность:*  *-*время отклика - получение результатов на типовое задание;  - пропускная способность число типовых заданий, исполняемых в единицу времени. | Секунды  Число в минуту | 3  60 | 3  60 |
| *Используемость ресурсов:*  *-* относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства. |  | 0,001 | 0,001 |

Таблица 16 – Требования к качественным характеристикам программного средства.

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Практичность** |  | | |
| *Простота использования:*  **-**среднее время ввода заданий;  -среднее время отклика на задание. | Секунды  Секунды | 10  3 | 10  3 |
| *Изучаемость :*  - трудоемкость изучения применения ПС;  -продолжительность изучения;  -объем эксплуатационной документации; | Чел.-часы  Часы  Страницы | 5  2  15 | 5  2  15 |
| **Сопровождаемость** |  | | |
| *Изменяемость:*  -трудоемкость подготовки изменений;  -длительность подготовки изменений. | Чел.-часы  Часы | 5  5 | 5  5 |
| *Тестируемость:*  *-трудоемкость тестирования изменений;*  *- длительность тестирования изменений.* | Чел.-часы  Часы | 3  3 | 3  3 |
| **Мобильность** |  | | |
| *Адаптируемость:*  - трудоемкость адаптации;  -длительность адаптации. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Простота установки:*  -трудоемкость инсталляции;  -длительность инсталляции. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Замещаемость:*  -трудоемкость замены компонентов;  - длительность замены компонентов. | Чел.-часы  Часы | 0,2  0,2 | 0,2  0,2 |

Таблица 17 – Сложность

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип объекта | Количество | Вес | | | Итого |
|  |  | Простой | Средний | Сложный |  |
| Экран | 17 | x1 | x2 | **x3** | =51 |
| Отчет | 3 | x2 | x5 | **x8** | =24 |

Таблица 18 – Сложность экрана

|  |  |  |  |
| --- | --- | --- | --- |
| Экраны | Количество клиентских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | Всего <8 | **Всего >8** |
|  | (<2 срв, <3 клт) | (2-3 срв, 3-5 клт) | (>3 срв, >5 клт) |
| <3 | Простой | Простой | Средний |
| 3-7 | Простой | Средний | Сложный |
| **>8** | Средний | Сложный | **Сложный** |

Таблица 19 – Сложность отчета

|  |  |  |  |
| --- | --- | --- | --- |
| Отчеты | Количество Возвратских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | Всего <8 | Всего >8 |
|  | (<2 срв, <3 клт) | (2-3 срв, 3-5 клт) | **(>3 срв, >5 клт)** |
| 0 или 1 | Простой | Простой | Средний |
| **2 или 3** | Простой | Средний | **Сложный** |
| >4 | Средний | Сложный | Сложный |

Производительность = 6000 тысяч строк кода/1 месяца=6000 тысяч строк кода/месяц

Качество = 30 ошибок /6 тысяч строк кода=5 ошибок/тысяч строк кода

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Г. Шилдт C#: Учебный курс. - Спб.: Питер, 2021. - 512с

2. Либерти, Д. Программирование на Си Шарп. - М.: Символ-плюс, 2018. - 684 с.

3. Ватсон К. Си Шарп/К.Ватсон.-М.:Лори, 2021.-862 с

4. Балена Ф., Димауро Д. Современная практика программирования на Visual Basic и Visual C#.-М.:Русская редакция, 2021,604с.

5. Енин А., Енин Н. Локальная СУБД своими руками. Учимся на примерах.-М.:СОЛОН-ПРЕСС, 2021,464с.,СД

6. Лабор В.В. Си Шарп: СоВозврат приложений для Windows/-Мн.: Харвест, 2021-384с.

7. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2021-512с.

8. ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению

9. ГОСТ 19.102-77.ЕСПД. Стадии разработки

10. ГОСТ 19.505-79.ЕСПД. Руководство оператора. Требования к содержанию и оформлению

11. ГОСТ 19.002-80.ЕСПД. Схемы алгоритмов и программ. Правила выполнения

12. ГОСТ 19.404-79.ЕСПД. Пояснительная записка. Требования к содержанию и оформлению

13. Владимир Биллинг. Основы программирования на С#. 2018 г.

[Электронный ресурс http://mexalib.com/view/42671]

14. Никита Культин. Основы программирования в MicrosoftVisualC# 2019. 2019 г.

[Электронный ресурс http://www.litres.ru/nikita-kultin/osnovy-programmirovaniya-v-microsoft-visual-c-2010-2]

15. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2021.

[Электронный ресурс http://www.twirpx.com/file/27852]

16. Йен Гриффитс. Программирование на С#. 2020г.

[Электронный ресурс http://rutracker.org/forum/viewtopic.php?t=4772853]

**Приложение А**

Автоматизированная информационная система

«Каршеринговая компания MaxДрайв»

Текст кода программного средства

460.ЗМК.00016-01 12 01

**Текст кода программного средства**

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<configSections>

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />

</configSections>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8" />

</startup>

<connectionStrings>

<add name="CarsharingEntities" connectionString="metadata=res://\*/Entities.Model1.csdl|res://\*/Entities.Model1.ssdl|res://\*/Entities.Model1.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=(localdb)\MSSQLLocalDB;initial catalog=Carsharing;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

<entityFramework>

<defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />

</providers>

</entityFramework>

</configuration>

<Application x:Class="CarsharingApp.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:CarsharingApp" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

StartupUri="/Windows/MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<!-- This is the current way to setup your app's initial theme -->

<materialDesign:BundledTheme BaseTheme="Inherit"

ColorAdjustment="{materialDesign:ColorAdjustment}"

PrimaryColor="DeepPurple"

SecondaryColor="Lime" />

<!-- If you would prefer to use your own colors there is an option for that as well -->

<!--<materialDesign:CustomColorTheme BaseTheme="Light" PrimaryColor="Aqua" SecondaryColor="DarkGreen" />-->

<!--You can also use the built in theme dictionaries as well-->

<!--<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Light.xaml" />-->

<!--<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.DeepPurple.xaml" />-->

<!--<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.Lime.xaml" />-->

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml"/>

</ResourceDictionary.MergedDictionaries>

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="Comic Sans MS"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="Margin" Value="3"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

<Style TargetType="ListViewItem" BasedOn="{StaticResource MaterialDesignListBoxItem}" x:Key="ListViewItem">

<Setter Property="BorderBrush" Value="MediumPurple"/>

<Setter Property="BorderThickness" Value="2"/>

<Setter Property="Margin" Value="3"/>

<Style.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="10"/>

</Style>

</Style.Resources>

</Style>

<Style TargetType="TextBox" BasedOn="{StaticResource MaterialDesignOutlinedTextBox}">

<Setter Property="Padding" Value="5"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

</Style>

<Style TargetType="ComboBox" BasedOn="{StaticResource MaterialDesignOutlinedComboBox}">

<Setter Property="Padding" Value="5"/>

<Setter Property="Margin" Value="5"/>

</Style>

<Style TargetType="TextBlock" x:Key="textWrap">

<Setter Property="TextWrapping" Value="Wrap"/>

</Style>

</ResourceDictionary>

</Application.Resources>

</Application>

<Window x:Class="CarsharingApp.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

Title="{Binding ElementName=MainFrame, Path=Content.Title}" Height="800" Width="800" WindowStartupLocation="CenterScreen"

WindowState="Maximized" MinHeight="800" MinWidth="1300" Closed="Window\_Closed" Closing="Window\_Closing">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="90"/>

<RowDefinition />

</Grid.RowDefinitions>

<Grid Background="Black"/>

<StackPanel Orientation="Horizontal">

<Image Source="/Assets/car.png" Margin="5" HorizontalAlignment="Left"/>

<StackPanel Orientation="Horizontal" x:Name="StackPanelAdmin" Visibility="Collapsed">

<Button x:Name="BtnCar" Content="{materialDesign:PackIcon Kind=Car}" ToolTip="Автомобили" Margin="5" Click="BtnCar\_Click"/>

<Button x:Name="BtnClient" Content="{materialDesign:PackIcon Kind=User}" ToolTip="Клиенты" Margin="5" Click="BtnClient\_Click"/>

<Button x:Name="BtnRental" Content="{materialDesign:PackIcon Kind=ListBox}" ToolTip="Аренды" Margin="5" Click="BtnRental\_Click"/>

<Button x:Name="BtnAddItionalService" Content="{materialDesign:PackIcon Kind=CardPlus}" ToolTip="Дополнинительные услуги" Margin="5" Click="BtnAddItionalService\_Click"/>

</StackPanel>

<StackPanel x:Name="StackPanelUser" Orientation="Horizontal" Margin="5" HorizontalAlignment="Left" Visibility="Collapsed">

<Button x:Name="BtnMyProfile" Content="{materialDesign:PackIcon Kind=User}" ToolTip="Мой профиль" Margin="5" Click="BtnMyProfile\_Click"/>

<Button x:Name="BtnMyRental" Content="{materialDesign:PackIcon Kind=ListBox}" ToolTip="Мои аренды" Margin="5" Click="BtnMyRental\_Click"/>

</StackPanel>

</StackPanel>

<TextBlock Text="Каршеринговая компания 'MaxДрайв'" HorizontalAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" FontSize="20" Foreground="MediumPurple"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" Margin="0,0,10,0">

<Button x:Name="BtnLogin" Content="{materialDesign:PackIcon Kind=LoginVariant}" ToolTip="Авторизация" Margin="5" Click="BtnLogin\_Click"/>

<Button x:Name="BtnBack" Content="{materialDesign:PackIcon Kind=ArrowBack}" ToolTip="Назад" Margin="5" Click="BtnBack\_Click"/>

<Button x:Name="BtnExit" Content="{materialDesign:PackIcon Kind=ExitRun}" ToolTip="Выход" Margin="5" Click="BtnExit\_Click"/>

</StackPanel>

<Frame Grid.Row="1" x:Name="MainFrame" NavigationUIVisibility="Hidden" ContentRendered="MainFrame\_ContentRendered" />

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using CarsharingApp.Entities;

using CarsharingApp.Pages;

using CarsharingApp.Windows;

namespace CarsharingApp

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Content = new CarPage();

Manager.MainFrame = MainFrame;

}

private void MainFrame\_ContentRendered(object sender, EventArgs e)

{

UpdateButtons();

}

private void UpdateButtons()

{

if (MainFrame.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

BtnExit.Visibility = Visibility.Collapsed;

BtnLogin.Visibility = Visibility.Collapsed;

StackPanelAdmin.Visibility = Visibility.Collapsed;

StackPanelUser.Visibility = Visibility.Collapsed;

}

else

{

BtnBack.Visibility = Visibility.Collapsed;

BtnLogin.Visibility = Visibility.Visible;

BtnExit.Visibility = Visibility.Collapsed;

if (Manager.Client != null)

{

StackPanelUser.Visibility = Visibility.Visible;

BtnLogin.Visibility = Visibility.Collapsed;

BtnExit.Visibility = Visibility.Visible;

}

else if (Manager.AdminLogin)

{

StackPanelAdmin.Visibility = Visibility.Visible;

BtnLogin.Visibility = Visibility.Collapsed;

BtnExit.Visibility = Visibility.Visible;

}

else

{

StackPanelAdmin.Visibility = Visibility.Collapsed;

StackPanelUser.Visibility = Visibility.Collapsed;

}

}

}

private void BtnCar\_Click(object sender, RoutedEventArgs e)

{

MainFrame.Content = new CatalogCarPage();

}

private void BtnClient\_Click(object sender, RoutedEventArgs e)

{

MainFrame.Content = new ClientPage();

}

private void BtnRental\_Click(object sender, RoutedEventArgs e)

{

MainFrame.Content = new RentalPage();

}

private void BtnMyProfile\_Click(object sender, RoutedEventArgs e)

{

RegClientWindow regClientWindow = new RegClientWindow(Manager.Client);

regClientWindow.ShowDialog();

}

private void BtnMyRental\_Click(object sender, RoutedEventArgs e)

{

MainFrame.Content = new MyRentalPage();

}

private void BtnLogin\_Click(object sender, RoutedEventArgs e)

{

LoginWindow loginWindow = new LoginWindow();

if (loginWindow.ShowDialog() == true)

{

UpdateButtons();

MainFrame.NavigationService.Refresh();

}

}

private void BtnBack\_Click(object sender, RoutedEventArgs e)

{

if (MainFrame.CanGoBack)

MainFrame.GoBack();

}

private void BtnExit\_Click(object sender, RoutedEventArgs e)

{

if (MessageBox.Show("Выйти из данной учетной записи?", "Выход", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

Manager.AdminLogin = false;

Manager.Client = null;

UpdateButtons();

MainFrame.NavigationService.Refresh();

}

}

private void Window\_Closed(object sender, EventArgs e)

{

App.Current.Shutdown();

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

if (MessageBox.Show("Закрыть приложение?","Выход", MessageBoxButton.OKCancel, MessageBoxImage.Question) == MessageBoxResult.Cancel)

e.Cancel = true;

}

private void BtnAddItionalService\_Click(object sender, RoutedEventArgs e)

{

MainFrame.Content = new AdditionalServicePage();

}

}

}

<Window x:Class="CarsharingApp.Windows.LoginWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

Title="Вход" Height="320" Width="300" MinWidth="300" MinHeight="320" WindowStartupLocation="CenterScreen" ResizeMode="NoResize">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="100"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Image Source="/Assets/car.png" Margin="5" HorizontalAlignment="Center" />

<StackPanel Grid.Row="1" VerticalAlignment="Center">

<TextBox Grid.Row="1" Width="200" VerticalAlignment="Center" x:Name="TbLogin" materialDesign:HintAssist.Hint="Логин" Text="admin"/>

<PasswordBox Style="{StaticResource MaterialDesignOutlinedPasswordBox}" Padding="5" Grid.Row="1" Width="200" VerticalAlignment="Center" x:Name="TbPassword"

materialDesign:HintAssist.Hint="Пароль" Password="777"/>

</StackPanel>

<StackPanel Grid.Row="2" HorizontalAlignment="Center" Orientation="Horizontal">

<Button x:Name="BtnOk" Content="Ok" Margin="2" Click="BtnOk\_Click"/>

<Button x:Name="BtnCancel" Content="Cancel" Margin="2" Click="BtnCancel\_Click"/>

</StackPanel>

<Button x:Name="BtnReg" Grid.Row="3" Content="Регистрация" Margin="5" HorizontalAlignment="Center" Click="BtnReg\_Click"/>

</Grid>

</Window>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using CarsharingApp.Entities;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для LoginWindow.xaml

/// </summary>

public partial class LoginWindow : Window

{

public LoginWindow()

{

InitializeComponent();

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

try

{

List<Client> clients = CarsharingEntities.GetContext().Clients.ToList();

Client client = clients.FirstOrDefault(p => p.Login == TbLogin.Text && p.Password == TbPassword.Password);

if (client != null)

{

Manager.Client = client;

DialogResult = true;

}

else if (TbLogin.Text == "admin" && TbPassword.Password == "777")

{

Manager.AdminLogin = true;

DialogResult = true;

}

else

MessageBox.Show("Не верный логин или пароль");

} catch(Exception ex) { ex.Message.ToString(); }

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private void BtnReg\_Click(object sender, RoutedEventArgs e)

{

RegClientWindow regClientWindow = new RegClientWindow(null);

regClientWindow.ShowDialog();

}

}

}

<Page x:Class="CarsharingApp.Pages.CarPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:CarsharingApp.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Каталог автомобилей" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel HorizontalAlignment="Center">

<TextBox Width="200" x:Name="TbCarName" materialDesign:HintAssist.Hint="Поиск по наименованию" TextChanged="TbCarName\_TextChanged"/>

<ComboBox Width="200" Name="ComboCarClasses" materialDesign:HintAssist.Hint="Класс авто" DisplayMemberPath="CarClassName" SelectionChanged="ComboCarClasses\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboGearboxTypes" materialDesign:HintAssist.Hint="Тип КПП" DisplayMemberPath="GearboxTypeName" SelectionChanged="ComboGearboxTypes\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboUnitTypes" materialDesign:HintAssist.Hint="Тип привода" DisplayMemberPath="DriveUnitName" SelectionChanged="ComboUnitTypes\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboSort" materialDesign:HintAssist.Hint="Сортировка" SelectionChanged="ComboSort\_SelectionChanged">

<ComboBoxItem>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Стоимость аренды"/>

<materialDesign:PackIcon Kind="ArrowUp" Margin="5,0" VerticalAlignment="Center"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Стоимость аренды"/>

<materialDesign:PackIcon Kind="ArrowDown" Margin="5,0" VerticalAlignment="Center"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Наиименование авто "/>

<materialDesign:PackIcon Kind="ArrowUp" Margin="5,0" VerticalAlignment="Center"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Наименование авто"/>

<materialDesign:PackIcon Kind="ArrowDown" Margin="5,0" VerticalAlignment="Center"/>

</StackPanel>

</ComboBoxItem>

</ComboBox>

</WrapPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Время бронирования с "/>

<DatePicker x:Name="DatePickerStart" Width="150" Style="{StaticResource MaterialDesignOutlinedDatePicker}" Margin="5" Padding="5" SelectedDateChanged="DatePickerStart\_SelectedDateChanged"/>

<materialDesign:TimePicker x:Name="TimePickerStart" Width="90" Style="{StaticResource MaterialDesignOutlinedTimePicker}"

Is24Hours="True" Margin="5" Padding="5" SelectedTimeChanged="TimePickerStart\_SelectedTimeChanged"/>

<TextBlock Text="по"/>

<DatePicker x:Name="DatePickerEnd" Width="150" Style="{StaticResource MaterialDesignOutlinedDatePicker}" Margin="5" Padding="5" SelectedDateChanged="DatePickerEnd\_SelectedDateChanged"/>

<materialDesign:TimePicker x:Name="TimePickerEnd" Width="90" Style="{StaticResource MaterialDesignOutlinedTimePicker}" Is24Hours="True" Margin="5" Padding="5"

SelectedTimeChanged="TimePickerEnd\_SelectedTimeChanged"/>

</StackPanel>

<ListView Grid.Row="2" x:Name="ListViewCars" ItemContainerStyle="{StaticResource ListViewItem}" ScrollViewer.HorizontalScrollBarVisibility="Disabled">

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Width="300">

<Grid.RowDefinitions>

<RowDefinition Height="200"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Image Width="200" Height="200" Stretch="Uniform" HorizontalAlignment="Center">

<Image.Source>

<Binding Path="CarPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<StackPanel Grid.Row="1">

<TextBlock Text="{Binding CarClass.CarClassName}" HorizontalAlignment="Left" Margin="5"/>

<TextBlock Text="{Binding CarName}" HorizontalAlignment="Left" Margin="5" FontSize="20" FontWeight="Bold"/>

<StackPanel Orientation="Horizontal" Margin="5" >

<materialDesign:PackIcon Kind="CarTransmission" VerticalAlignment="Center"/>

<TextBlock Text="{Binding GearboxType.GearboxTypeName}" VerticalAlignment="Center"/>

<materialDesign:PackIcon Kind="CarCog" VerticalAlignment="Center" Margin="5,0,0,0"/>

<TextBlock Text="{Binding DriveUnit.DriveUnitName}" VerticalAlignment="Center"/>

<materialDesign:PackIcon Kind="GasStation" VerticalAlignment="Center" Margin="5,0,0,0"/>

<TextBlock Text="{Binding AvgConsumption,StringFormat={}{0}л/100км}" VerticalAlignment="Center"/>

</StackPanel>

<TextBlock Text="Дополнительная информация:" HorizontalAlignment="Left" Margin="5"/>

<TextBlock Text="{Binding CarAdditionalInfo}" HorizontalAlignment="Left" Margin="5,0,5,5" Height="60" TextWrapping="Wrap"/>

<TextBlock Text="{Binding RentalPrice,StringFormat={}{0:f2} Р/сутки}" FontWeight="Bold" FontSize="15" Foreground="Coral" TextDecorations="Strikethrough"/>

<TextBlock Text="{Binding ActualPrice,StringFormat={}{0:f2} Р/сутки}" FontWeight="Bold" FontSize="20" Foreground="MediumPurple"/>

<Button x:Name="BtnRental" Background="LightCoral" BorderBrush="LightCoral" Foreground="White" Click="BtnRental\_Click" Visibility="{Binding RentalVisible}">

<TextBlock Text="Арендовать" Foreground="White"/>

</Button>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Grid>

</Page>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using CarsharingApp.Entities;

using CarsharingApp.Pages;

using CarsharingApp.Windows;

using Xceed.Wpf.Toolkit;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для CarPage.xaml

/// </summary>

public partial class CarPage : Page

{

private DateTime dateTimeStart = DateTime.Today;

private DateTime dateTimeEnd = DateTime.Today;

public CarPage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

ListViewCars.ItemsSource = CarsharingEntities.GetContext().Cars.ToList();

DatePickerStart.SelectedDate = TimePickerStart.SelectedTime = dateTimeStart = DateTime.Now;

DatePickerEnd.SelectedDate = TimePickerEnd.SelectedTime = dateTimeEnd = DateTime.Now.AddDays(1);

var carClasses = CarsharingEntities.GetContext().CarClasses.ToList();

carClasses.Insert(0, new CarClass { CarClassName = "Все категории" });

ComboCarClasses.ItemsSource = carClasses;

var gearboxTypes = CarsharingEntities.GetContext().GearboxTypes.ToList();

gearboxTypes.Insert(0, new GearboxType { GearboxTypeName = "Все категории" });

ComboGearboxTypes.ItemsSource = gearboxTypes;

var unitTypes = CarsharingEntities.GetContext().DriveUnits.ToList();

unitTypes.Insert(0, new DriveUnit { DriveUnitName = "Все категории" });

ComboUnitTypes.ItemsSource = unitTypes;

UpdateDate();

}

private void BtnRental\_Click(object sender, RoutedEventArgs e)

{

AddRentalWindow addRentalWindow = new AddRentalWindow(((Button)sender).DataContext as Car, dateTimeStart, dateTimeEnd);

addRentalWindow.ShowDialog();

UpdateDate();

}

private void ComboCarClasses\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void ComboGearboxTypes\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void ComboUnitTypes\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void TbCarName\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateDate();

}

private void UpdateDate()

{

List<Car> cars = CarsharingEntities.GetContext().Cars.ToList();

List<RentalCar> rentalCars = CarsharingEntities.GetContext().RentalCars.ToList();

// загрузка свободных машин

cars = cars.Where(p => p.RentalCars.FirstOrDefault(k => !((k.RentalStart <= dateTimeStart && k.RentalEnd <= dateTimeStart) && (k.RentalStart <= dateTimeEnd && k.RentalEnd <= dateTimeEnd))) == null).ToList();

// фильтрация

if (ComboCarClasses.SelectedIndex > 0 && ComboCarClasses.SelectedItem is CarClass carClass)

cars = cars.Where(p => p.CarClassId == carClass.CarClassId).ToList();

if (ComboGearboxTypes.SelectedIndex > 0 && ComboGearboxTypes.SelectedItem is GearboxType gearboxType)

cars = cars.Where(p => p.GearboxTypeId == gearboxType.GearboxTypeId).ToList();

if (ComboUnitTypes.SelectedIndex > 0 && ComboUnitTypes.SelectedItem is DriveUnit driveUinit)

cars = cars.Where(p => p.DriveUnitId == driveUinit.DriveUnitId).ToList();

switch (ComboSort.SelectedIndex)

{

case 0:

cars = cars.OrderBy(p => p.ActualPrice).ToList();

break;

case 1:

cars = cars.OrderByDescending(p => p.ActualPrice).ToList();

break;

case 2:

cars = cars.OrderBy(p => p.CarName).ToList();

break;

case 3:

cars = cars.OrderByDescending(p => p.CarName).ToList();

break;

}

// поиск автомобиля

cars = cars.Where(p => p.CarName.ToLower().Contains(TbCarName.Text.ToLower())).ToList();

ListViewCars.ItemsSource = cars;

}

private void DatePickerStart\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

if (DatePickerStart.SelectedDate != null && TimePickerStart.SelectedTime != null)

{

DateTime dateTime = DatePickerStart.SelectedDate.Value;

DateTime time = TimePickerStart.SelectedTime.Value;

dateTimeStart = new DateTime(dateTime.Year, dateTime.Month, dateTime.Day, time.Hour, time.Minute, time.Second);

}

UpdateDate();

}

private void TimePickerStart\_SelectedTimeChanged(object sender, RoutedPropertyChangedEventArgs<DateTime?> e)

{

if (DatePickerStart.SelectedDate != null && TimePickerStart.SelectedTime != null)

{

DateTime dateTime = DatePickerStart.SelectedDate.Value;

DateTime time = TimePickerStart.SelectedTime.Value;

dateTimeStart = new DateTime(dateTime.Year, dateTime.Month, dateTime.Day, time.Hour, time.Minute, time.Second);

}

UpdateDate();

}

private void DatePickerEnd\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

if (DatePickerEnd.SelectedDate != null && TimePickerEnd.SelectedTime != null)

{

DateTime dateTime = DatePickerEnd.SelectedDate.Value;

DateTime time = TimePickerEnd.SelectedTime.Value;

dateTimeEnd = new DateTime(dateTime.Year, dateTime.Month, dateTime.Day, time.Hour, time.Minute, time.Second);

}

UpdateDate();

}

private void TimePickerEnd\_SelectedTimeChanged(object sender, RoutedPropertyChangedEventArgs<DateTime?> e)

{

if (DatePickerEnd.SelectedDate != null && TimePickerEnd.SelectedTime != null)

{

DateTime dateTime = DatePickerEnd.SelectedDate.Value;

DateTime time = TimePickerEnd.SelectedTime.Value;

dateTimeEnd = new DateTime(dateTime.Year, dateTime.Month, dateTime.Day, time.Hour, time.Minute, time.Second);

}

UpdateDate();

}

private void ComboSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

}

}

<Page x:Class="CarsharingApp.Pages.CatalogCarPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:CarsharingApp.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Каталог автомобилей" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel HorizontalAlignment="Center">

<TextBox Width="200" x:Name="TbCarName" materialDesign:HintAssist.Hint="Поиск по наименованию" TextChanged="TbCarName\_TextChanged"/>

<ComboBox Width="200" Name="ComboCarClasses" materialDesign:HintAssist.Hint="Класс авто" DisplayMemberPath="CarClassName" SelectionChanged="ComboCarClasses\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboGearboxTypes" materialDesign:HintAssist.Hint="Тип КПП" DisplayMemberPath="GearboxTypeName" SelectionChanged="ComboGearboxTypes\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboUnitTypes" materialDesign:HintAssist.Hint="Тип привода" DisplayMemberPath="DriveUnitName" SelectionChanged="ComboUnitTypes\_SelectionChanged"/>

</WrapPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="BtnAdd" Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Margin="5" Click="BtnAdd\_Click"/>

<Button x:Name="BtnGearboxType" Content="{materialDesign:PackIcon Kind=CarTransmission}" ToolTip="Типы трансмиссиий" Margin="5" Click="BtnGearboxType\_Click"/>

<Button x:Name="BtnDriveUnit" Content="{materialDesign:PackIcon Kind=CarCog}" ToolTip="Типы приводов" Margin="5" Click="BtnDriveUnit\_Click"/>

<Button x:Name="BtnCarClass" Content="{materialDesign:PackIcon Kind=CarMultiple}" ToolTip="Классы автомобилей" Margin="5" Click="BtnCarClass\_Click"/>

</StackPanel>

<DataGrid Grid.Row="2" x:Name="DataGridCars" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTemplateColumn Width="110" Header="Изо-ие">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="CarPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding CarName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Доп. информация" Width="3\*" Binding="{Binding CarAdditionalInfo}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Класс авто" Width="\*" Binding="{Binding CarClass.CarClassName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Тип трансмиссии" Width="\*" Binding="{Binding GearboxType.GearboxTypeName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Тип привода" Width="\*" Binding="{Binding DriveUnit.DriveUnitName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Цена аренды" Width="\*" Binding="{Binding RentalPrice,StringFormat='c', ConverterCulture='ru-RU'}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Скидка" Width="\*" Binding="{Binding Discount,StringFormat={}{0}%}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Актуальная цена" Width="\*" Binding="{Binding ActualPrice,StringFormat='c', ConverterCulture='ru-RU'}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Ср. расход" Width="\*" Binding="{Binding AvgConsumption,StringFormat={}{0}л/100км}" ElementStyle="{StaticResource textWrap}"/>

<!--<DataGridTextColumn Header="Забронировано" Width="\*" Binding="{Binding IsPrecense}" ElementStyle="{StaticResource textWrap}"/>-->

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для CatalogCarPage.xaml

/// </summary>

public partial class CatalogCarPage : Page

{

public CatalogCarPage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridCars.ItemsSource = CarsharingEntities.GetContext().Cars.ToList();

var carClasses = CarsharingEntities.GetContext().CarClasses.ToList();

carClasses.Insert(0, new CarClass { CarClassName = "Все категории" });

ComboCarClasses.ItemsSource = carClasses;

var gearboxTypes = CarsharingEntities.GetContext().GearboxTypes.ToList();

gearboxTypes.Insert(0, new GearboxType { GearboxTypeName = "Все категории" });

ComboGearboxTypes.ItemsSource = gearboxTypes;

var unitTypes = CarsharingEntities.GetContext().DriveUnits.ToList();

unitTypes.Insert(0, new DriveUnit { DriveUnitName = "Все категории" });

ComboUnitTypes.ItemsSource = unitTypes;

}

private void TbCarName\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateDate();

}

private void ComboCarClasses\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void ComboGearboxTypes\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void ComboUnitTypes\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Content = new AddOrEdittCarPage(null);

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Content = new AddOrEdittCarPage(((Button)sender).DataContext as Car);

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите удалить данную запись?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes && DataGridCars.SelectedItem is Car car)

{

try

{

if (car.RentalCars.Count > 0)

throw new Exception("Существуют записи в производных таблицах, удаление запрещено!");

CarsharingEntities.GetContext().Cars.Remove(car);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void UpdateDate()

{

List<Car> cars = CarsharingEntities.GetContext().Cars.ToList();

// фильтрация

if (ComboCarClasses.SelectedIndex > 0 && ComboCarClasses.SelectedItem is CarClass carClass)

cars = cars.Where(p => p.CarClassId == carClass.CarClassId).ToList();

if (ComboGearboxTypes.SelectedIndex > 0 && ComboGearboxTypes.SelectedItem is GearboxType gearboxType)

cars = cars.Where(p => p.GearboxTypeId == gearboxType.GearboxTypeId).ToList();

if (ComboUnitTypes.SelectedIndex > 0 && ComboUnitTypes.SelectedItem is DriveUnit driveUinit)

cars = cars.Where(p => p.DriveUnitId == driveUinit.DriveUnitId).ToList();

// поиск автомобиля

cars = cars.Where(p => p.CarName.ToLower().Contains(TbCarName.Text.ToLower())).ToList();

DataGridCars.ItemsSource = cars;

}

private void BtnGearboxType\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Content = new GearboxTypePage();

}

private void BtnDriveUnit\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Content = new DriveUnitPage();

}

private void BtnCarClass\_Click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Content = new CarClassPage();

}

}

}

<Page x:Class="CarsharingApp.Pages.AdditionalServicePage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Дополнительные услуги" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="BtnAdd" Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Margin="5" Click="BtnAdd\_Click"/>

</StackPanel>

<DataGrid Grid.Row="2" x:Name="DataGridAddServices" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding AddServiceName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Описание" Width="2\*" Binding="{Binding AddServiceDescription}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Цена услуги" Width="2\*" Binding="{Binding AddServicePrice, StringFormat='c', ConverterCulture='ru-RU'}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using CarsharingApp.Windows;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для AdditionalServicePage.xaml

/// </summary>

public partial class AdditionalServicePage : Page

{

public AdditionalServicePage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridAddServices.ItemsSource = CarsharingEntities.GetContext().AdditionalServices.ToList();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

if (DataGridAddServices.SelectedItem is AdditionalService additionalService)

{

AddOrEditAddServiceWindow addServiceWindow = new AddOrEditAddServiceWindow(additionalService);

addServiceWindow.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите удалить данную запись?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DataGridAddServices.SelectedItem is AdditionalService additionalService)

{

try

{

if (additionalService.RentalAddServices.Count > 0)

throw new Exception("Существуют записи в производных таблицах, удаление запрещено!");

CarsharingEntities.GetContext().AdditionalServices.Remove(additionalService);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

AddOrEditAddServiceWindow addServiceWindow = new AddOrEditAddServiceWindow(null);

addServiceWindow.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

}

<Page x:Class="CarsharingApp.Pages.AddOrEdittCarPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Добавление/Редактирование автомобиля" Background="White">

<Page.Resources>

</Page.Resources>

<ScrollViewer>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<StackPanel>

<TextBlock Text="Изображение" Margin="10" HorizontalAlignment="Left"/>

<Button x:Name="BtnLoad" HorizontalAlignment="Center" Content="{materialDesign:PackIcon Kind=Download}" ToolTip="Назад" Margin="5" Click="BtnLoad\_Click"/>

</StackPanel>

<TextBlock Text="Название авто" Grid.Row="1" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Дополнительная информация" Grid.Row="2" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Тип трансмиссии" Grid.Row="3" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Тип привода" Grid.Row="4" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Класс авто" Grid.Row="5" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Статус" Grid.Row="6" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Цена аренды" Grid.Row="7" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Размер скидки (%)" Grid.Row="8" Margin="10" HorizontalAlignment="Left"/>

<TextBlock Text="Средний расход (л/100км)" Grid.Row="9" Margin="10" HorizontalAlignment="Left"/>

<Image x:Name="ImageCar" MaxWidth="200" MaxHeight="200" Stretch="Uniform" HorizontalAlignment="Left" Margin="5" Grid.Column="1">

<Image.Source>

<Binding Path="CarPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBox Text="{Binding CarName}" HorizontalAlignment="Left" Width="300" Grid.Column="1" Grid.Row="1"/>

<TextBox Text="{Binding CarAdditionalInfo}" HorizontalAlignment="Left" Width="300" TextWrapping="Wrap" Height="100" Grid.Column="1" Grid.Row="2"/>

<ComboBox x:Name="ComboGearboxTypes" Width="300" HorizontalAlignment="Left" Grid.Column="1" Grid.Row="3" DisplayMemberPath="GearboxTypeName" SelectedItem="{Binding GearboxType}"/>

<ComboBox x:Name="ComboDriveUnits" Width="300" HorizontalAlignment="Left" Grid.Column="1" Grid.Row="4" DisplayMemberPath="DriveUnitName" SelectedItem="{Binding DriveUnit}"/>

<ComboBox x:Name="ComboCarClasses" Width="300" HorizontalAlignment="Left" Grid.Column="1" Grid.Row="5" DisplayMemberPath="CarClassName" SelectedItem="{Binding CarClass}"/>

<ComboBox x:Name="ComboStatuses" Width="300" HorizontalAlignment="Left" Grid.Column="1" Grid.Row="6" DisplayMemberPath="StatusName" SelectedItem="{Binding Status}"/>

<TextBox Text="{Binding RentalPrice,StringFormat='c', ConverterCulture='ru-RU'}" HorizontalAlignment="Left" Width="150" Grid.Column="1" Grid.Row="7"/>

<TextBox Text="{Binding Discount}" HorizontalAlignment="Left" Width="150" Grid.Column="1" Grid.Row="8"/>

<TextBox Text="{Binding AvgConsumption}" HorizontalAlignment="Left" Width="150" Grid.Column="1" Grid.Row="9"/>

<Button Grid.Row="10" Margin="10" x:Name="BtnSave" Content="Сохранить" HorizontalAlignment="Center" Grid.ColumnSpan="2" Click="BtnSave\_Click"/>

</Grid>

</ScrollViewer>

</Page>

using CarsharingApp.Entities;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для AddOrEdittCarPage.xaml

/// </summary>

public partial class AddOrEdittCarPage : Page

{

private Car \_currentCar = new Car();

private string textInfo = "изменена";

public AddOrEdittCarPage(Car car)

{

InitializeComponent();

if (car != null)

{

\_currentCar = car;

}

DataContext = \_currentCar;

ComboCarClasses.ItemsSource = CarsharingEntities.GetContext().CarClasses.ToList();

ComboDriveUnits.ItemsSource = CarsharingEntities.GetContext().DriveUnits.ToList();

ComboGearboxTypes.ItemsSource = CarsharingEntities.GetContext().GearboxTypes.ToList();

ComboStatuses.ItemsSource = CarsharingEntities.GetContext().Statuses.ToList();

}

private void BtnSave\_Click(object sender, RoutedEventArgs e)

{

try

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentCar.CarId == 0)

{

textInfo = "добавлена";

CarsharingEntities.GetContext().Cars.Add(\_currentCar);

}

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись " + textInfo);

Manager.MainFrame.GoBack();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void BtnLoad\_Click(object sender, RoutedEventArgs e)

{

try

{

OpenFileDialog op = new OpenFileDialog() { Title = "Загрузить", Filter = "Image Files|\*.png;\*.jpg;\*.jpeg;\*.gif" };

if (op.ShowDialog() == true)

{

// проверка на размер файла

FileInfo fileInfo = new FileInfo(op.FileName);

if (fileInfo.Length > 1024 \* 1024 \* 2)

throw new Exception("Размер файла не больше 2 мб");

ImageCar.Source = new BitmapImage(new Uri(op.FileName));

Stream stream = File.OpenRead(op.FileName);

byte[] binaryImage = new byte[stream.Length];

stream.Read(binaryImage, 0, (int)stream.Length);

\_currentCar.CarImage = binaryImage;

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentCar.CarName))

error.AppendLine("Введите название авто");

if (ComboGearboxTypes.SelectedItem == null)

error.AppendLine("Выберите тип трансмисии");

if (ComboDriveUnits.SelectedItem == null)

error.AppendLine("Выберите тип привода");

if (ComboCarClasses.SelectedItem == null)

error.AppendLine("Выберите класс авто");

if (ComboStatuses.SelectedItem == null)

error.AppendLine("Укажите статус авто");

if (\_currentCar.RentalPrice < 0)

error.AppendLine("Цена аренды не может быть меньше 0");

if (\_currentCar.AvgConsumption < 0)

error.AppendLine("Средний расход не может быть меньше 0");

if (\_currentCar.Discount < 0 || \_currentCar.Discount > 100)

error.AppendLine("Размер скидки в диапазоне от 0 до 100");

return error;

}

}

}

<Page x:Class="CarsharingApp.Pages.CarClassPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Классы автомобилей" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="BtnAdd" Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Margin="5" Click="BtnAdd\_Click"/>

</StackPanel>

<DataGrid Grid.Row="2" x:Name="DataGridCarClasses" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding CarClassName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Описание" Width="2\*" Binding="{Binding CarClassDescription}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using CarsharingApp.Windows;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для CarClassPage.xaml

/// </summary>

public partial class CarClassPage : Page

{

public CarClassPage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridCarClasses.ItemsSource = CarsharingEntities.GetContext().CarClasses.ToList();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

if (DataGridCarClasses.SelectedItem is CarClass carClass)

{

AddOrEditCarClassWindow addOrEditCarClassWindow = new AddOrEditCarClassWindow(carClass);

addOrEditCarClassWindow.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите удалить данную запись?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DataGridCarClasses.SelectedItem is CarClass carClass)

{

try

{

if (carClass.Cars.Count > 0)

throw new Exception("Существуют записи в производных таблицах, удаление запрещено!");

CarsharingEntities.GetContext().CarClasses.Remove(carClass);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

AddOrEditCarClassWindow addOrEditCarClassWindow = new AddOrEditCarClassWindow(null);

addOrEditCarClassWindow.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

}

<Page x:Class="CarsharingApp.Pages.ClientPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Клиенты" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel HorizontalAlignment="Center">

<TextBox Width="300" x:Name="TbFIOClient" materialDesign:HintAssist.Hint="ФИО клиента" TextChanged="TbFIOClient\_TextChanged"/>

</WrapPanel>

<Button Grid.Row="1" x:Name="BtnAdd" HorizontalAlignment="Right" Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Margin="5" Click="BtnAdd\_Click"/>

<DataGrid Grid.Row="2" x:Name="DataGridClients" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTemplateColumn Width="110" Header="Изо-ие">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="ClientPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="Фамилия" Width="\*" Binding="{Binding LastName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Имя" Width="\*" Binding="{Binding FirstName}"

ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Отчество" Width="\*" Binding="{Binding MiddleName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Дата рождения" Width="\*" Binding="{Binding DateOfBirth,StringFormat={}{0:dd.MM.yyyy}}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Пол" Width="\*" Binding="{Binding GenderName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Номер телефона" Width="\*" Binding="{Binding Phone}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Номер прав" Width="\*" Binding="{Binding DriveLicenseNum}" ElementStyle="{StaticResource textWrap}"/>

<!--<DataGridTextColumn Header="Забронировано" Width="\*" Binding="{Binding IsPrecense}" ElementStyle="{StaticResource textWrap}"/>-->

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using CarsharingApp.Windows;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для ClientPage.xaml

/// </summary>

public partial class ClientPage : Page

{

public ClientPage()

{

InitializeComponent();

}

private void TbFIOClient\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateDate();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridClients.ItemsSource = CarsharingEntities.GetContext().Clients.ToList();

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

RegClientWindow regClientWindow = new RegClientWindow(null);

regClientWindow.ShowDialog();

UpdateDate();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

if (DataGridClients.SelectedItem is Client client)

{

RegClientWindow regClientWindow = new RegClientWindow(client);

regClientWindow.ShowDialog();

UpdateDate();

}

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите удалить данную запись?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (result == MessageBoxResult.Yes && DataGridClients.SelectedItem is Client client)

{

try

{

if (client.RentalCars.Count > 0)

throw new Exception("Существуют записи в производных таблицах, удаление запрещено!");

CarsharingEntities.GetContext().Clients.Remove(client);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void UpdateDate()

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

List<Client> clients = CarsharingEntities.GetContext().Clients.ToList();

// поиск клиента

clients = clients.Where(p => p.LastName.ToLower().Contains(TbFIOClient.Text.ToLower()) || p.FirstName.ToLower().Contains(TbFIOClient.Text.ToLower()) ||

p.MiddleName.ToLower().Contains(TbFIOClient.Text.ToLower())).ToList();

DataGridClients.ItemsSource = clients;

}

}

}

<Page x:Class="CarsharingApp.Pages.DriveUnitPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Типы приводов" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="BtnAdd" Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Margin="5" Click="BtnAdd\_Click"/>

</StackPanel>

<DataGrid Grid.Row="2" x:Name="DataGridDriveUnits" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding DriveUnitName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using CarsharingApp.Windows;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для DriveUnitPage.xaml

/// </summary>

public partial class DriveUnitPage : Page

{

public DriveUnitPage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridDriveUnits.ItemsSource = CarsharingEntities.GetContext().DriveUnits.ToList();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

if (DataGridDriveUnits.SelectedItem is DriveUnit driveUnit)

{

AddOrEditDriveUnitWindow addOrEditDriveUnitWindow = new AddOrEditDriveUnitWindow(driveUnit);

addOrEditDriveUnitWindow.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите удалить данную запись?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DataGridDriveUnits.SelectedItem is DriveUnit driveUnit)

{

try

{

if (driveUnit.Cars.Count > 0)

throw new Exception("Существуют записи в производных таблицах, удаление запрещено!");

CarsharingEntities.GetContext().DriveUnits.Remove(driveUnit);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

AddOrEditDriveUnitWindow addOrEditDriveUnitWindow = new AddOrEditDriveUnitWindow(null);

addOrEditDriveUnitWindow.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

}

<Page x:Class="CarsharingApp.Pages.GearboxTypePage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Тип трансмиссий" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="BtnAdd" Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Margin="5" Click="BtnAdd\_Click"/>

</StackPanel>

<DataGrid Grid.Row="2" x:Name="DataGridGearboxTypes" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding GearboxTypeName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using CarsharingApp.Windows;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для GearboxTypePage.xaml

/// </summary>

public partial class GearboxTypePage : Page

{

public GearboxTypePage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridGearboxTypes.ItemsSource = CarsharingEntities.GetContext().GearboxTypes.ToList();

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

if (DataGridGearboxTypes.SelectedItem is GearboxType gearboxType)

{

AddOrEditGearboxWindow addOrEditGearbox = new AddOrEditGearboxWindow(gearboxType);

addOrEditGearbox.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите удалить данную запись?", "Удаление", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DataGridGearboxTypes.SelectedItem is GearboxType gearboxType)

{

try

{

if (gearboxType.Cars.Count > 0)

throw new Exception("Существуют записи в производных таблицах, удаление запрещено!");

CarsharingEntities.GetContext().GearboxTypes.Remove(gearboxType);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

AddOrEditGearboxWindow addOrEditGearbox = new AddOrEditGearboxWindow(null);

addOrEditGearbox.ShowDialog();

Page\_IsVisibleChanged(null, default);

}

}

}

<Page x:Class="CarsharingApp.Pages.MyRentalPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Мои аренды" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel HorizontalAlignment="Center">

<ComboBox Width="200" Name="ComboCars" materialDesign:HintAssist.Hint="Авто" DisplayMemberPath="CarName" SelectionChanged="ComboCars\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboStatuses" materialDesign:HintAssist.Hint="Статус" DisplayMemberPath="StatusName" SelectionChanged="ComboStatuses\_SelectionChanged"/>

</WrapPanel>

<DataGrid Grid.Row="2" x:Name="DataGridRentals" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTemplateColumn Width="110" Header="Изо-ие авто">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="Car.CarPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding Car.CarName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Список доп. услуг" Width="2\*" Binding="{Binding RentalServicesName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Дата начала аренды" Width="\*" Binding="{Binding RentalStart,StringFormat={}{0:dd.MM.yyyy HH.mm}}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Дата конца аренды" Width="\*" Binding="{Binding RentalEnd,StringFormat={}{0:dd.MM.yyyy HH.mm}}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Итоговая стоимость" Width="\*" Binding="{Binding RentalPrice,StringFormat='c', ConverterCulture='ru-RU'}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Статус" Width="\*" Binding="{Binding Status.StatusName}" ElementStyle="{StaticResource textWrap}"/>

<!--<DataGridTextColumn Header="Забронировано" Width="\*" Binding="{Binding IsPrecense}" ElementStyle="{StaticResource textWrap}"/>-->

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Отменить" Click="BtnDell\_Click" Visibility="{Binding BtnEditVisible}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для MyRentalPage.xaml

/// </summary>

public partial class MyRentalPage : Page

{

public MyRentalPage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridRentals.ItemsSource = CarsharingEntities.GetContext().RentalCars.Where(p => p.ClientId == Manager.Client.ClientId).ToList();

var cars = CarsharingEntities.GetContext().Cars.ToList();

cars.Insert(0, new Car { CarName = "Все категории" });

ComboCars.ItemsSource = cars;

var statuses = CarsharingEntities.GetContext().Statuses.ToList();

statuses.Insert(0, new Status { StatusName = "Все статусы" });

ComboStatuses.ItemsSource = statuses;

}

private void ComboCars\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void ComboStatuses\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите отменить данную запись?", "Отмена", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DataGridRentals.SelectedItem is RentalCar rentalCar)

{

try

{

CarsharingEntities.GetContext().RentalAddServices.RemoveRange(rentalCar.RentalAddServices);

CarsharingEntities.GetContext().RentalCars.Remove(rentalCar);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void UpdateDate()

{

List<RentalCar> rentalCars = CarsharingEntities.GetContext().RentalCars.Where(p => p.ClientId == Manager.Client.ClientId).ToList();

// фильтрация

if (ComboCars.SelectedIndex > 0 && ComboCars.SelectedItem is Car car)

rentalCars = rentalCars.Where(p => p.CarId == car.CarId).ToList();

if (ComboStatuses.SelectedIndex > 0 && ComboStatuses.SelectedItem is Status status)

rentalCars = rentalCars.Where(p => p.StatusId == status.StatusId).ToList();

DataGridRentals.ItemsSource = rentalCars;

}

}

}

<Page x:Class="CarsharingApp.Pages.RentalPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:CarsharingApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Аренда" Background="White" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel HorizontalAlignment="Center">

<TextBox Width="300" x:Name="TbFIOClient" materialDesign:HintAssist.Hint="ФИО клиента" TextChanged="TbFIOClient\_TextChanged"/>

<ComboBox Width="200" Name="ComboCars" materialDesign:HintAssist.Hint="Авто" DisplayMemberPath="CarName" SelectionChanged="ComboCars\_SelectionChanged"/>

<ComboBox Width="200" Name="ComboStatuses" materialDesign:HintAssist.Hint="Статус" DisplayMemberPath="StatusName" SelectionChanged="ComboStatuses\_SelectionChanged"/>

</WrapPanel>

<DataGrid Grid.Row="2" x:Name="DataGridRentals" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTemplateColumn Width="110" Header="Изо-ие клиента">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="Client.ClientPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="ФИО" Width="\*" Binding="{Binding Client.ClientFullName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTemplateColumn Width="110" Header="Изо-ие авто">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="Car.CarPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="Название" Width="\*" Binding="{Binding Car.CarName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Список доп. услуг" Width="2\*" Binding="{Binding RentalServicesName}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Дата начала аренды" Width="\*" Binding="{Binding RentalStart,StringFormat={}{0:dd.MM.yyyy HH.mm}}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Дата конца аренды" Width="\*" Binding="{Binding RentalEnd,StringFormat={}{0:dd.MM.yyyy HH.mm}}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Итоговая стоимость" Width="\*" Binding="{Binding RentalPrice,StringFormat='c', ConverterCulture='ru-RU'}" ElementStyle="{StaticResource textWrap}"/>

<DataGridTextColumn Header="Статус" Width="\*" Binding="{Binding Status.StatusName}" ElementStyle="{StaticResource textWrap}"/>

<!--<DataGridTextColumn Header="Забронировано" Width="\*" Binding="{Binding IsPrecense}" ElementStyle="{StaticResource textWrap}"/>-->

<!--<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" ToolTip="Редактировать" Click="BtnEdit\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>-->

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDell" Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Click="BtnDell\_Click"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace CarsharingApp.Pages

{

/// <summary>

/// Логика взаимодействия для RentalPage.xaml

/// </summary>

public partial class RentalPage : Page

{

public RentalPage()

{

InitializeComponent();

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

CarsharingEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DataGridRentals.ItemsSource = CarsharingEntities.GetContext().RentalCars.ToList();

var cars = CarsharingEntities.GetContext().Cars.ToList();

cars.Insert(0, new Car { CarName = "Все категории" });

ComboCars.ItemsSource = cars;

var statuses = CarsharingEntities.GetContext().Statuses.ToList();

statuses.Insert(0, new Status { StatusName = "Все статусы" });

ComboStatuses.ItemsSource = statuses;

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

}

private void BtnEdit\_Click(object sender, RoutedEventArgs e)

{

}

private void BtnDell\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы хотите отменить данную запись?", "Отмена", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (DataGridRentals.SelectedItem is RentalCar rentalCar)

{

try

{

CarsharingEntities.GetContext().RentalAddServices.RemoveRange(rentalCar.RentalAddServices);

CarsharingEntities.GetContext().RentalCars.Remove(rentalCar);

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись удалена!");

Page\_IsVisibleChanged(null, default);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void ComboCars\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

private void UpdateDate()

{

List<RentalCar> rentalCars = CarsharingEntities.GetContext().RentalCars.ToList();

// фильтрация

if (ComboCars.SelectedIndex > 0 && ComboCars.SelectedItem is Car car)

rentalCars = rentalCars.Where(p => p.CarId == car.CarId).ToList();

if (ComboStatuses.SelectedIndex > 0 && ComboStatuses .SelectedItem is Status status)

rentalCars = rentalCars.Where(p => p.StatusId == status.StatusId).ToList();

// поиск клиента

rentalCars = rentalCars.Where(p => p.Client.LastName.ToLower().Contains(TbFIOClient.Text.ToLower()) || p.Client.FirstName.ToLower().Contains(TbFIOClient.Text.ToLower()) ||

p.Client.MiddleName.ToLower().Contains(TbFIOClient.Text.ToLower())).ToList();

DataGridRentals.ItemsSource = rentalCars;

}

private void TbFIOClient\_TextChanged(object sender, TextChangedEventArgs e)

{

UpdateDate();

}

private void ComboStatuses\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateDate();

}

}

}

<Window x:Class="CarsharingApp.Windows.AddOrEditAddServiceWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows"

mc:Ignorable="d"

Title="Добавление/Редактирование доп. услуги" Height="360" Width="400" MinHeight="360" MinWidth="400" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Название" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="300" Text="{Binding AddServiceName}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}" />

<TextBlock Text="Описание" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="300" Text="{Binding AddServiceDescription}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}"

TextWrapping="Wrap" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Height="100"/>

<TextBlock Text="Цена услуги" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="150" Text="{Binding AddServicePrice,StringFormat='c', ConverterCulture='ru-RU'}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right" Margin="10">

<Button x:Name="BtnOk" Content="Ok" Click="BtnOk\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}"/>

<Button x:Name="BtnCancel" Content="Cancel" Click="BtnCancel\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}" Margin="10,0,0,0"/>

</StackPanel>

</Grid>

</Window>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrEditAddServiceWindow.xaml

/// </summary>

public partial class AddOrEditAddServiceWindow : Window

{

private AdditionalService \_currentAddService = new AdditionalService();

private string textInfo = "изменена";

public AddOrEditAddServiceWindow(AdditionalService addService)

{

InitializeComponent();

if (addService != null)

\_currentAddService = addService;

DataContext = \_currentAddService;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

try

{

if (\_currentAddService.AddServiceId == 0)

{

textInfo = "добавлена";

CarsharingEntities.GetContext().AdditionalServices.Add(\_currentAddService);

}

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись " + textInfo);

Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentAddService.AddServiceName))

error.AppendLine("Введите название доп. услуги");

if (\_currentAddService.AddServicePrice < 0)

error.AppendLine("Цена услуги не может быть отрицательной");

return error;

}

}

}

<Window x:Class="CarsharingApp.Windows.AddOrEditCarClassWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows"

mc:Ignorable="d"

Title="Добавление/Редактирование класс авто" Height="330" Width="400" MinHeight="330" MinWidth="400" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Название" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="300" Text="{Binding CarClassName}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}" />

<TextBlock Text="Описание" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="300" Text="{Binding CarClassDescription}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}"

TextWrapping="Wrap" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Height="100"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right" Margin="10">

<Button x:Name="BtnOk" Content="Ok" Click="BtnOk\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}"/>

<Button x:Name="BtnCancel" Content="Cancel" Click="BtnCancel\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}" Margin="10,0,0,0"/>

</StackPanel>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using CarsharingApp.Entities;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrEditCarClassWindow.xaml

/// </summary>

public partial class AddOrEditCarClassWindow : Window

{

private CarClass \_currentCarClass = new CarClass();

private string textInfo = "изменена";

public AddOrEditCarClassWindow(CarClass carClass)

{

InitializeComponent();

if (carClass != null)

\_currentCarClass = carClass;

DataContext = \_currentCarClass;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

try

{

if (\_currentCarClass.CarClassId == 0)

{

textInfo = "добавлена";

CarsharingEntities.GetContext().CarClasses.Add(\_currentCarClass);

}

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись " + textInfo);

Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentCarClass.CarClassName))

error.AppendLine("Введите название класса авто");

return error;

}

}

}

<Window x:Class="CarsharingApp.Windows.AddOrEditDriveUnitWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows"

mc:Ignorable="d"

Title="Добавление/Редактирование типа привода" Height="200" Width="400" MinHeight="200" MinWidth="400" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Название" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="300" Text="{Binding DriveUnitName}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}" />

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right" Margin="10">

<Button x:Name="BtnOk" Content="Ok" Click="BtnOk\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}"/>

<Button x:Name="BtnCancel" Content="Cancel" Click="BtnCancel\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}" Margin="10,0,0,0"/>

</StackPanel>

</Grid>

</Window>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrEditDriveUnitWindow.xaml

/// </summary>

public partial class AddOrEditDriveUnitWindow : Window

{

private DriveUnit \_currentDriveUnit = new DriveUnit();

private string textInfo = "изменена";

public AddOrEditDriveUnitWindow(DriveUnit driveUnit)

{

InitializeComponent();

if (driveUnit != null)

\_currentDriveUnit = driveUnit;

DataContext = \_currentDriveUnit;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

try

{

if (\_currentDriveUnit.DriveUnitId == 0)

{

textInfo = "добавлена";

CarsharingEntities.GetContext().DriveUnits.Add(\_currentDriveUnit);

}

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись " + textInfo);

Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentDriveUnit.DriveUnitName))

error.AppendLine("Введите тип привода");

return error;

}

}

}

<Window x:Class="CarsharingApp.Windows.AddOrEditGearboxWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows"

mc:Ignorable="d"

Title="Добавление/Редактирование типа трансмиссии" Height="200" Width="400" MinHeight="200" MinWidth="400" ResizeMode="NoResize" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Название" HorizontalAlignment="Left" Margin="5"/>

<TextBox Width="300" Text="{Binding GearboxTypeName}" FontSize="14" Padding="5" Margin="5" Style="{StaticResource MaterialDesignOutlinedTextBox}" />

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Right" Margin="10">

<Button x:Name="BtnOk" Content="Ok" Click="BtnOk\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}"/>

<Button x:Name="BtnCancel" Content="Cancel" Click="BtnCancel\_Click" Style="{StaticResource MaterialDesignFlatLightBgButton}" Margin="10,0,0,0"/>

</StackPanel>

</Grid>

</Window>

using CarsharingApp.Entities;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для AddOrEditGearboxWindow.xaml

/// </summary>

public partial class AddOrEditGearboxWindow : Window

{

private GearboxType \_currentGearboxType = new GearboxType();

private string textInfo = "изменена";

public AddOrEditGearboxWindow(GearboxType gearboxType)

{

InitializeComponent();

if (gearboxType != null)

\_currentGearboxType = gearboxType;

DataContext = \_currentGearboxType;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

try

{

if (\_currentGearboxType.GearboxTypeId == 0)

{

textInfo = "добавлена";

CarsharingEntities.GetContext().GearboxTypes.Add(\_currentGearboxType);

}

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись " + textInfo);

Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentGearboxType.GearboxTypeName))

error.AppendLine("Введите тип трансмиссии");

return error;

}

}

}

<Window x:Class="CarsharingApp.Windows.AddRentalWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:xceed="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

Title="Аренда автомобиля" Height="740" Width="370" MinHeight="740" MinWidth="370" WindowStartupLocation="CenterScreen" ResizeMode="NoResize">

<ScrollViewer>

<Grid >

<Grid.RowDefinitions>

<RowDefinition Height="100"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Image MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center">

<Image.Source>

<Binding Path="CarPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<StackPanel Grid.Row="1" HorizontalAlignment="Center">

<TextBlock Text="{Binding CarClass.CarClassName}" HorizontalAlignment="Left" Margin="5"/>

<TextBlock Text="{Binding CarName}" HorizontalAlignment="Left" Margin="5" FontSize="20" FontWeight="Bold"/>

<StackPanel Orientation="Horizontal" Margin="5" >

<materialDesign:PackIcon Kind="CarTransmission" VerticalAlignment="Center"/>

<TextBlock Text="{Binding GearboxType.GearboxTypeName}" VerticalAlignment="Center"/>

<materialDesign:PackIcon Kind="CarCog" VerticalAlignment="Center" Margin="5,0,0,0"/>

<TextBlock Text="{Binding DriveUnit.DriveUnitName}" VerticalAlignment="Center"/>

<materialDesign:PackIcon Kind="GasStation" VerticalAlignment="Center" Margin="5,0,0,0"/>

<TextBlock Text="{Binding AvgConsumption,StringFormat={}{0}л/100км}" VerticalAlignment="Center"/>

</StackPanel>

<TextBlock Text="Дополнительная информация:" HorizontalAlignment="Left" Margin="5" TextWrapping="NoWrap"/>

<TextBlock Text="{Binding CarAdditionalInfo}" HorizontalAlignment="Left" Margin="5,0,5,5" Height="60" TextWrapping="Wrap"/>

<TextBlock Text="Период аренды" HorizontalAlignment="Left" Margin="5" FontSize="16" FontWeight="Bold"/>

<TextBlock Text="c" HorizontalAlignment="Left" Margin="5" />

<StackPanel Orientation="Horizontal">

<DatePicker x:Name="DatePickerStart" Width="150" Style="{StaticResource MaterialDesignOutlinedDatePicker}" Margin="5" Padding="5" IsEnabled="False" SelectedDateChanged="DatePickerStart\_SelectedDateChanged"/>

<materialDesign:TimePicker x:Name="TimePickerStart" Width="90" Style="{StaticResource MaterialDesignOutlinedTimePicker}"

Is24Hours="True" Margin="5" Padding="5" SelectedTimeChanged="TimePickerStart\_SelectedTimeChanged" IsEnabled="False" />

</StackPanel>

<TextBlock Text="по" HorizontalAlignment="Left" Margin="5" />

<StackPanel Orientation="Horizontal">

<DatePicker x:Name="DatePickerEnd" Width="150" Style="{StaticResource MaterialDesignOutlinedDatePicker}" Margin="5" Padding="5" SelectedDateChanged="DatePickerEnd\_SelectedDateChanged" IsEnabled="False" />

<materialDesign:TimePicker x:Name="TimePickerEnd" Width="90" Style="{StaticResource MaterialDesignOutlinedTimePicker}" Is24Hours="True" Margin="5" Padding="5" IsEnabled="False"

SelectedTimeChanged="TimePickerEnd\_SelectedTimeChanged"/>

</StackPanel>

<TextBlock Text="Дополнительные опции" HorizontalAlignment="Left" Margin="5" FontSize="16" FontWeight="Bold"/>

<xceed:CheckComboBox x:Name="ComboAddServices" Width="250" DisplayMemberPath="AddServiceNamePice" Margin="5" ItemSelectionChanged="ComboAddServices\_ItemSelectionChanged"

HorizontalAlignment="Left" SelectedMemberPath="AddServiceSelected"/>

<TextBlock Text="Общая стоимость" HorizontalAlignment="Left" Margin="5,0" FontSize="16" FontWeight="Bold"/>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="В сутки: " HorizontalAlignment="Left" Margin="5,0" FontSize="14" FontWeight="Bold"/>

<TextBlock x:Name="TbDay" Text="" Grid.Column="1" HorizontalAlignment="Right" Margin="5,0" FontSize="14" FontWeight="Bold"/>

<TextBlock Text="За весь период: " Grid.Row="1" HorizontalAlignment="Left" Margin="5,0" FontSize="14" FontWeight="Bold"/>

<TextBlock x:Name="TbPeriod" Text="" Grid.Row="1" Grid.Column="1" HorizontalAlignment="Right" Margin="5,0" FontSize="14" FontWeight="Bold"/>

<TextBlock Text="За доп. опций: " Grid.Row="2" HorizontalAlignment="Left" Margin="5,0" FontSize="14" FontWeight="Bold"/>

<TextBlock x:Name="TbAddService" Text="" Grid.Row="2" Grid.Column="1" HorizontalAlignment="Right" Margin="5,0" FontSize="14" FontWeight="Bold"/>

<TextBlock Text="Итого" HorizontalAlignment="Left" Grid.Row="3" Margin="5" FontSize="16" FontWeight="Bold"/>

<TextBlock x:Name="TbGeneral" Text="" Grid.Row="3" Grid.Column="1" HorizontalAlignment="Right" Margin="5" FontSize="16" FontWeight="Bold"/>

</Grid>

</StackPanel>

<Button x:Name="BtnRental" Grid.Row="2" Margin="10" Background="LightCoral" BorderBrush="LightCoral" Foreground="White" Click="BtnRental\_Click" HorizontalAlignment="Center">

<TextBlock Text="Арендовать" Foreground="White"/>

</Button>

</Grid>

</ScrollViewer>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using CarsharingApp.Entities;

using CarsharingApp.Windows;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для AddRentalWindow.xaml

/// </summary>

public partial class AddRentalWindow : Window

{

private Car \_currentCar = new Car();

RentalCar rentalCar = new RentalCar();

//private RentalCar \_currentRental = new RentalCar();

private decimal rentalPricePeriod = 0;

private decimal rentalPriceService = 0;

private decimal rentalPriceGeneral = 0;

private DateTime dateTimeStart = DateTime.Today;

private DateTime dateTimeEnd = DateTime.Today;

TimeSpan timeInterval = TimeSpan.Zero;

List<AdditionalService> \_additionalServices = CarsharingEntities.GetContext().AdditionalServices.ToList();

public AddRentalWindow(Car car, DateTime dateTimeStart, DateTime dateTimeEnd)

{

InitializeComponent();

if (car != null)

\_currentCar = car;

DataContext = \_currentCar;

ComboAddServices.ItemsSource = \_additionalServices;

DatePickerStart.SelectedDate = TimePickerStart.SelectedTime = this.dateTimeStart = dateTimeStart;

DatePickerEnd.SelectedDate = TimePickerEnd.SelectedTime = this.dateTimeEnd = dateTimeEnd;

//DatePickerStart.SelectedDate = TimePickerStart.SelectedTime = DateTime.Now;

//DatePickerEnd.SelectedDate = TimePickerEnd.SelectedTime = DateTime.Now.AddDays(1);

TbDay.Text = $"{car.ActualPrice:f2} p.";

}

private void BtnRental\_Click(object sender, RoutedEventArgs e)

{

try

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

rentalCar = new RentalCar

{

CarId = \_currentCar.CarId,

ClientId = Manager.Client.ClientId,

RentalStart = dateTimeStart,

RentalEnd = dateTimeEnd,

RentalPrice = rentalPriceGeneral,

StatusId = CarsharingEntities.GetContext().Statuses.FirstOrDefault(p => p.StatusCode == 3).StatusId

};

CarsharingEntities.GetContext().RentalCars.Add(rentalCar);

\_currentCar.StatusId = 2;

CarsharingEntities.GetContext().SaveChanges();

SaveAdditionalServices();

MessageBox.Show("Ожидайте, ваша заявка обрабатывается");

Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (DatePickerStart.SelectedDate == null)

error.AppendLine("Укажите дату начала аренды");

if (TimePickerStart.SelectedTime == null)

error.AppendLine("Укажите время начала аренды");

if (DatePickerEnd.SelectedDate == null)

error.AppendLine("Укажите дату сдачи автомобиля");

if (TimePickerEnd.SelectedTime == null)

error.AppendLine("Укажте время сдачаи автомобиля");

return error;

}

private void SaveAdditionalServices()

{

try

{

List<RentalAddService> rentalAddServices = new List<RentalAddService>();

foreach (var i in \_additionalServices)

if (i.AddServiceSelected)

rentalAddServices.Add(new RentalAddService

{

RentalCarId = rentalCar.RentalCarId,

AddServiceId = i.AddServiceId

});

CarsharingEntities.GetContext().RentalAddServices.AddRange(rentalAddServices);

CarsharingEntities.GetContext().SaveChanges();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void ComboAddServices\_ItemSelectionChanged(object sender, Xceed.Wpf.Toolkit.Primitives.ItemSelectionChangedEventArgs e)

{

//AdditionalService additionalService = e.Item as AdditionalService;

//if (e.IsSelected)

// rentalPriceService += additionalService.AddServicePrice;

//else

// rentalPriceService += additionalService.AddServicePrice;

//if (timeInterval.Days > 0 && timeInterval.Minutes == 0)

// TbAddService.Text = $"{rentalPriceService \* timeInterval.Days:f2} p.";

//else

// TbAddService.Text = $"{rentalPriceService \* (timeInterval.Days + 1):f2} p.";

UpdateInfo();

}

private void TimePickerStart\_SelectedTimeChanged(object sender, RoutedPropertyChangedEventArgs<DateTime?> e)

{

UpdateInfo();

}

private void DatePickerStart\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

UpdateInfo();

}

private void DatePickerEnd\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

UpdateInfo();

}

private void TimePickerEnd\_SelectedTimeChanged(object sender, RoutedPropertyChangedEventArgs<DateTime?> e)

{

UpdateInfo();

}

private void UpdateInfo()

{

//if (DatePickerStart.SelectedDate != null && TimePickerStart.SelectedTime != null)

//{

// DateTime dateTime = DatePickerStart.SelectedDate.Value;

// DateTime time = TimePickerStart.SelectedTime.Value;

// dateTimeStart = new DateTime(dateTime.Year, dateTime.Month, dateTime.Day, time.Hour, time.Minute, time.Second);

//}

//if (DatePickerEnd.SelectedDate != null && TimePickerEnd.SelectedTime != null)

//{

// DateTime dateTime = DatePickerEnd.SelectedDate.Value;

// DateTime time = TimePickerEnd.SelectedTime.Value;

// dateTimeEnd = new DateTime(dateTime.Year, dateTime.Month, dateTime.Day, time.Hour, time.Minute, time.Second);

//}

rentalPriceService = 0;

foreach (var i in \_additionalServices)

if (i.AddServiceSelected)

rentalPriceService += i.AddServicePrice;

timeInterval = dateTimeEnd.Subtract(dateTimeStart);

if (timeInterval.Days > 0 && timeInterval.Minutes == 0)

{

rentalPricePeriod = \_currentCar.ActualPrice \* timeInterval.Days;

rentalPriceService = rentalPriceService \* timeInterval.Days;

}

else

{

rentalPriceService = rentalPriceService \* (timeInterval.Days + 1);

rentalPricePeriod = \_currentCar.ActualPrice \* (timeInterval.Days + 1);

}

rentalPriceGeneral = rentalPricePeriod + rentalPriceService;

TbPeriod.Text = $"{rentalPricePeriod:f2} p.";

TbAddService.Text = $"{rentalPriceService:f2} p.";

TbGeneral.Text = $"{rentalPriceGeneral:f2} p.";

}

}

}

<Window x:Class="CarsharingApp.Windows.RegClientWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:CarsharingApp.Windows" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

Title="Окно клиента" Height="680" Width="300" MinWidth="300" MinHeight="680" WindowStartupLocation="CenterScreen" ResizeMode="NoResize">

<Grid>

<Grid.Resources>

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="Comic Sans MS"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="Margin" Value="5"/>

<Setter Property="HorizontalAlignment" Value="Left"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

<Style TargetType="TextBox" BasedOn="{StaticResource MaterialDesignOutlinedTextBox}">

<Setter Property="Margin" Value="2,2,5,2"/>

<Setter Property="Padding" Value="5"/>

<Setter Property="FontSize" Value="14"/>

</Style>

</Grid.Resources>

<Grid.RowDefinitions>

<RowDefinition Height="110"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="auto"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Image x:Name="ClientImage" MaxWidth="100" MaxHeight="100" Stretch="Uniform" HorizontalAlignment="Center" Margin="5" Grid.ColumnSpan="2">

<Image.Source >

<Binding Path="ClientPhoto">

<Binding.TargetNullValue>

<ImageSource>/Assets/picture.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<Button Grid.Row="1" Grid.ColumnSpan="2" x:Name="BtnLoad" HorizontalAlignment="Center" Content="{materialDesign:PackIcon Kind=Download}" ToolTip="Назад" Margin="5" Click="BtnLoad\_Click"/>

<TextBlock Text="Фамилия" Grid.Row="2" />

<TextBlock Text="Имя" Grid.Row="3" />

<TextBlock Text="Отчество" Grid.Row="4" />

<TextBlock Text="Дата рождения" Grid.Row="5" />

<TextBlock Text="Пол" Grid.Row="6" />

<TextBlock Text="Телефон" Grid.Row="7" />

<TextBlock Text="Номер прав" Grid.Row="8" />

<TextBlock Text="Логин" Grid.Row="9" />

<TextBlock x:Name="TbOldPass" Text="Пароль" Grid.Row="11" />

<TextBlock x:Name="TextBNewPass" Text="Новый пароль" Grid.Row="12" Visibility="Collapsed"/>

<TextBlock Text="Повторите пароль" Grid.Row="13"/>

<TextBox Grid.Row="2" Grid.Column="1" Text="{Binding LastName}"/>

<TextBox Grid.Row="3" Grid.Column="1" Text="{Binding FirstName}"/>

<TextBox Grid.Row="4" Grid.Column="1" Text="{Binding MiddleName}"/>

<DatePicker Name="DateOfBirthPicker" Grid.Row="5" Grid.Column="1" Style="{StaticResource MaterialDesignOutlinedDatePicker}" Padding="5" FontSize="14" Margin="2,2,5,2"

SelectedDate="{Binding DateOfBirth}"/>

<StackPanel Orientation="Horizontal" Grid.Row="6" Grid.Column="1">

<RadioButton x:Name="RButtonM" GroupName="Gender" Content="м" Checked="RButtonM\_Checked"/>

<RadioButton x:Name="RButtonF" GroupName="Gender" Content="ж" Checked="RButtonF\_Checked"/>

</StackPanel>

<TextBox Grid.Row="7" Grid.Column="1" Text="{Binding Phone}"/>

<TextBox Grid.Row="8" Grid.Column="1" Text="{Binding DriveLicenseNum}"/>

<TextBox x:Name="TbLogin" Grid.Row="9" Grid.Column="1" Text="{Binding Login}"/>

<CheckBox x:Name="PassEditCheck" Content="Поменять пароль" Grid.Row="10" Visibility="Collapsed"/>

<PasswordBox x:Name="TbPass" Grid.Row="11" Grid.Column="1" Style="{StaticResource MaterialDesignOutlinedPasswordBox}" FontSize="14" Padding="5" Margin="2,2,5,2"/>

<PasswordBox x:Name="TbNewPass" Grid.Row="12" Grid.Column="1" Style="{StaticResource MaterialDesignOutlinedPasswordBox}" FontSize="14" Padding="5" Margin="2,2,5,2" Visibility="Collapsed"/>

<PasswordBox x:Name="TbPassRepeat" Grid.Row="13" Grid.Column="1" Style="{StaticResource MaterialDesignOutlinedPasswordBox}" FontSize="14" Padding="5" Margin="2,2,5,2"/>

<StackPanel Grid.Row="14" Margin="20" Grid.ColumnSpan="2" HorizontalAlignment="Center" Orientation="Horizontal">

<Button x:Name="BtnOk" Content="Ok" Margin="2" Click="BtnOk\_Click"/>

<Button x:Name="BtnCancel" Content="Cancel" Margin="2" Click="BtnCancel\_Click"/>

</StackPanel>

</Grid>

</Window>

using CarsharingApp.Entities;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace CarsharingApp.Windows

{

/// <summary>

/// Логика взаимодействия для RegClientWindow.xaml

/// </summary>

public partial class RegClientWindow : Window

{

private Client \_currentClient = new Client();

string textInfo = "изменена";

public RegClientWindow(Client client)

{

InitializeComponent();

if (client != null)

{

\_currentClient = client;

if (\_currentClient.Gender)

RButtonM.IsChecked = true;

else

RButtonF.IsChecked = true;

TbOldPass.Text = "Старый пароль";

TbLogin.IsEnabled = false;

PassEditCheck.Visibility = Visibility.Visible;

TbNewPass.Visibility = Visibility.Visible;

TextBNewPass.Visibility = Visibility.Visible;

}

else

{

\_currentClient.DateOfBirth = DateTime.Today;

}

DataContext = \_currentClient;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{

try

{

StringBuilder error = CheckField();

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentClient.ClientId == 0)

{

textInfo = "добалена";

\_currentClient.Password = TbPass.Password;

CarsharingEntities.GetContext().Clients.Add(\_currentClient);

}

CarsharingEntities.GetContext().SaveChanges();

MessageBox.Show("Запись " + textInfo);

Close();

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private void BtnLoad\_Click(object sender, RoutedEventArgs e)

{

try

{

OpenFileDialog op = new OpenFileDialog() { Title = "Загрузить", Filter = "Image Files|\*.png;\*.jpg;\*.jpeg;\*.gif" };

if (op.ShowDialog() == true)

{

// проверка на размер файла

FileInfo fileInfo = new FileInfo(op.FileName);

if (fileInfo.Length > 1024 \* 1024 \* 2)

throw new Exception("Размер файла не больше 2 мб");

ClientImage.Source = new BitmapImage(new Uri(op.FileName));

Stream stream = File.OpenRead(op.FileName);

byte[] binaryImage = new byte[stream.Length];

stream.Read(binaryImage, 0, (int)stream.Length);

\_currentClient.ClientImage = binaryImage;

}

}

catch (Exception ex) { MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); }

}

private StringBuilder CheckField()

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentClient.LastName))

error.AppendLine("Введите фамилию");

if (string.IsNullOrWhiteSpace(\_currentClient.FirstName))

error.AppendLine("Введите имя");

if (RButtonM.IsChecked == null && RButtonM.IsChecked == null)

error.AppendLine("Выберите пол");

if (string.IsNullOrWhiteSpace(\_currentClient.Phone))

error.AppendLine("Введите номер телефона");

if (string.IsNullOrWhiteSpace(\_currentClient.DriveLicenseNum))

error.AppendLine("Введите номер прав");

if (string.IsNullOrWhiteSpace(\_currentClient.Login))

error.AppendLine("Введите логин");

else

{

Client client = CarsharingEntities.GetContext().Clients.FirstOrDefault(p => p.Login == \_currentClient.Login);

if (\_currentClient.ClientId == 0 && client != null)

error.AppendLine("Указанный логин уже занят");

//else if (\_currentClient.ClientId != 0 && \_currentClient.ClientId != client.ClientId)

}

if (\_currentClient.ClientId == 0)

{

if (string.IsNullOrWhiteSpace(TbPass.Password))

error.AppendLine("Введите пароль");

if (TbPass.Password != TbPassRepeat.Password)

error.AppendLine("Введенные пароли не совпадают");

}

else if (PassEditCheck.IsChecked == true)

{

if (string.IsNullOrWhiteSpace(TbPass.Password))

error.AppendLine("Введите старый пароль");

else if (TbPass.Password != \_currentClient.Password)

error.AppendLine("Старый пароль введен неверно");

if (string.IsNullOrWhiteSpace(TbNewPass.Password))

error.AppendLine("Введите новый пароль");

else if (TbPassRepeat.Password != TbNewPass.Password)

error.AppendLine("Введенные пароли не совпадают");

else

\_currentClient.Password = TbNewPass.Password;

}

return error;

}

private void RButtonM\_Checked(object sender, RoutedEventArgs e)

{

\_currentClient.Gender = true;

}

private void RButtonF\_Checked(object sender, RoutedEventArgs e)

{

\_currentClient.Gender = false;

}

}

}

**Приложение Б**

Автоматизированная информационная система

«Каршеринговая компания MaxДрайв»

Спецификация

460.ЗМК.00016-01 01

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Обозначение** | | |  | | --- | | **Наименование** | | |  | | --- | | **Примечание** | |
|  | |  | | --- | | Документация | |  |
| 460.ЗМК.00016-01 12 01 | Автоматизированная информационная система «Каршеринговая компания MaxДрайв» |  |
|  | Текст программы |  |
| 460.ЗМК.00016-01 13 01 | Автоматизированная информационная система «Каршеринговая компания MaxДрайв» |  |
|  | Описание программы |  |
| 460.ЗМК.00016-01 34 01 | Автоматизированная информационная система «Каршеринговая компания MaxДрайв» |  |
|  | Руководство пользователя |  |
| 460.ЗМК.00016-01 81 01 | Автоматизированная информационная система «Каршеринговая компания MaxДрайв» |  |
|  | Пояснительная записка |  |

**Приложение B**

**Диск с программным средством**