МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема Программное средство «Аренда автомобилей»

Исполнитель

студент 2 курса группы 5 Скачко Илья Александрович

(Ф.И.О.)

Руководитель работы ассистент Север А.С.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Север А.С.

(подпись)

Минск 2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc104200164)

[Глава 1. Аналитический обзор литературы и постановка задачи 4](#_Toc104200165)

[1.1 Анализ прототипов 4](#_Toc104200166)

[1.2 Постановка задачи 6](#_Toc104200167)

[Глава 2. Анализ требований к программному средству и разработка функциональных требований 7](#_Toc104200168)

[2.1 Описание средств разработки 7](#_Toc104200169)

[2.2 Определение требований к программному средству 8](#_Toc104200170)

[2.3 Описание функциональности программного средства 9](#_Toc104200171)

[Глава 3. Проектирование программного средства 10](#_Toc104200172)

[3.1 Архитектура системы 10](#_Toc104200173)

[3.2 Разработка общей схемы приложения 10](#_Toc104200174)

[3.3 Описание структуры проекта 12](#_Toc104200175)

[3.4 Проектирование базы данных 13](#_Toc104200176)

[3.5 Проектирование сервиса 16](#_Toc104200177)

[Глава 4. Создание (реализация) программного средства 17](#_Toc104200178)

[4.1 Диаграмма классов 17](#_Toc104200179)

[4.2 Реализация архитектуры MVVM 18](#_Toc104200180)

[Глава 5. Тестирование, проверка работоспособности и анализ полученных результатов 20](#_Toc104200181)

[5.1 Мануальное тестирование 20](#_Toc104200182)

[5.2 Модульное тестирование 23](#_Toc104200183)

[Глава 6. Руководство по использованию 24](#_Toc104200184)

[ЗАКЛЮЧЕНИЕ 28](#_Toc104200185)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 29](#_Toc104200186)

[Приложение А 30](#_Toc104200187)

[Приложение Б 31](#_Toc104200188)

[Приложение В 32](#_Toc104200189)

[Приложение Г 33](#_Toc104200190)

[Приложение Д 34](#_Toc104200191)

ВВЕДЕНИЕ

Данный курсовой проект посвящен разработке программного средства «Аренда автомобилей». Данное приложение содержит две роли: пользователь(user), администратор(admin). В зависимости от роли определяется функционал приложения. Основными возможностями клиента являются выполнение регистрации, авторизации, бронирование автомобиля, просмотра сделанных заказов и отмена заказа. А в функции администратора входит: обработка заявок, редактирование предоставляемых автомобилей, а именно добавление нового и удаление старого.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс – Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом – С#, разработанный для создания приложений на платформе Microsoft .NET Framework.

Хранение данных осуществляется в Microsoft SQL Server

Для успешной реализации курсового проекта необходимо:

* провести анализ соответствующей литературы;
* ознакомиться с прототипами программных средств выбранной мной темы;
* определить функциональные требования;
* продумать структуру базы данных;
* продумать структуру проекта;
* реализовать программное средство;
* протестировать программное средство;
* написать руководство пользователя.

Содержание данной пояснительной записки отражает этапы выполнения курсового проекта.

Глава 1. Аналитический обзор литературы и постановка задачи

При написании курсового проекта была использована дополнительная литература. Основная информация для реализации курсового проекта была взята с сайтов https://metanit.com/, https://proffesorweb.ru/ и официального сайта <https://docs.microsoft.com/en-us/dotnet/csharp/>. Также использовались некоторые статьи с сайта https://habr.com/ru/.

1.1 Анализ прототипов

В качестве аналога для своего курсового проекта выбрано web-приложение rentcars.ru, так как оно имеет похожий функционал, который реализуется в моём приложении.

После входа открывается основное окно приложения, которое представлено на рисунке 1.1.

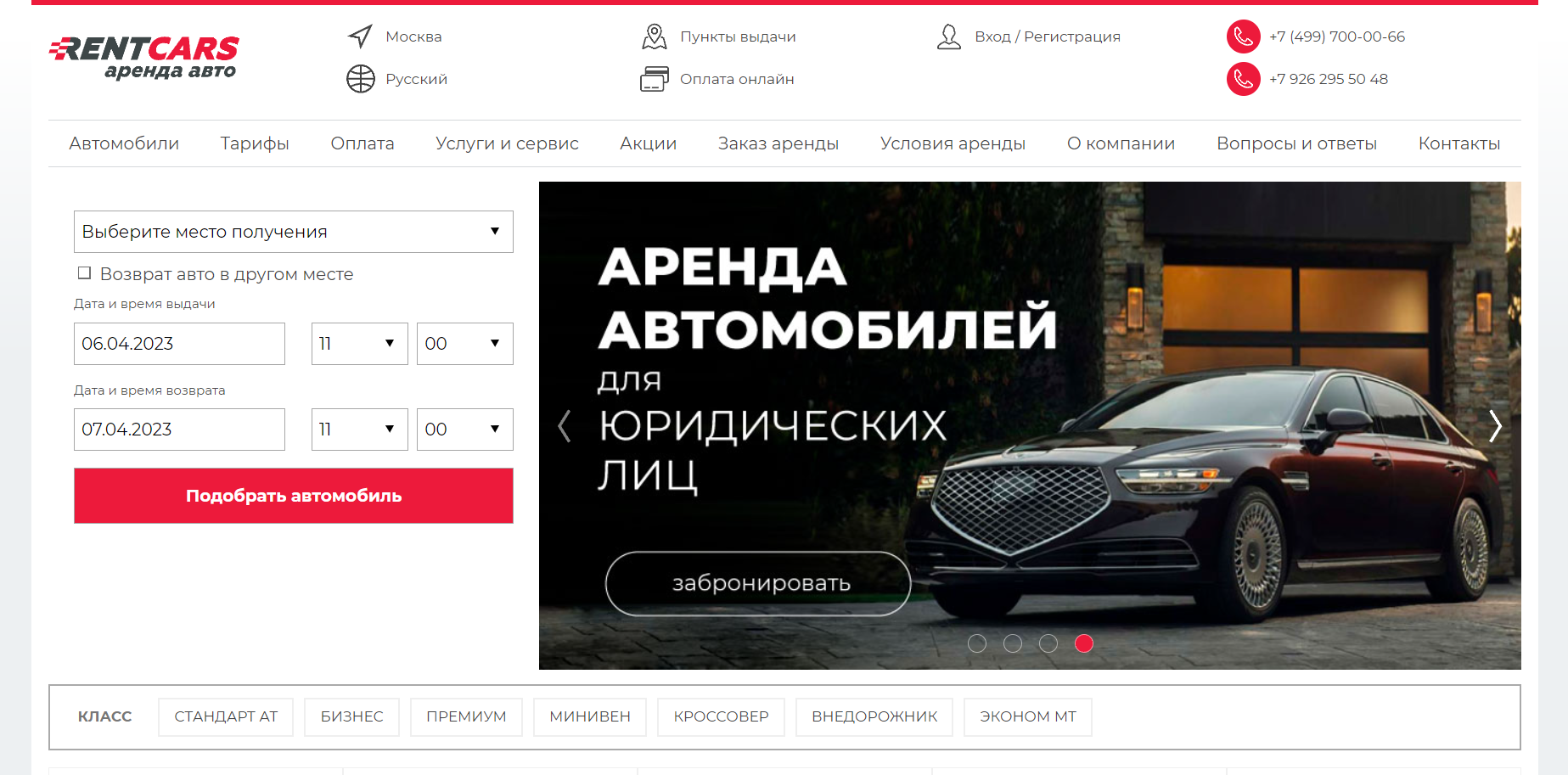


Рисунок 1.1 – Основная страница приложения

На основной странице виден список основной информации для пользователя и есть возможность заполнить форму для аренды.

В окне представлен список ссылок на страницы приложения, что позволяет быстро переходить в нужный раздел. Снизу есть поле, в котором представлены разные классы автомобилей.

Также сверху есть кнопка, при нажатии на которую отображается форма авторизации или регистрация в зависимости от выбора.

Окно регистрации представлено на рисунке 1.2.

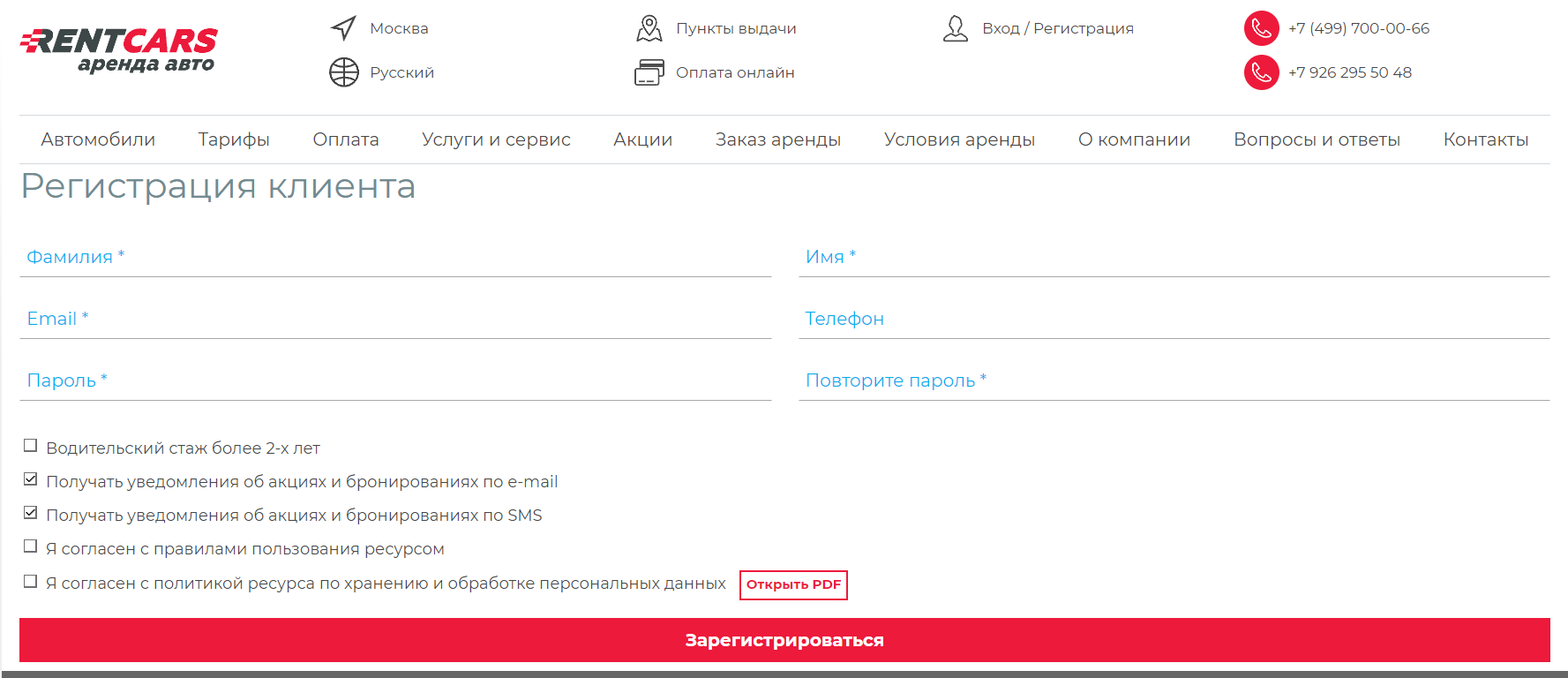


Рисунок 1.2 – Окно регистрации пользователей

Выбрать автомобиль можно перейдя по ссылке “Автомобили”. При нажатии на пункт открывается страница с автомобилями, которая представлена на рисунке 1.3.

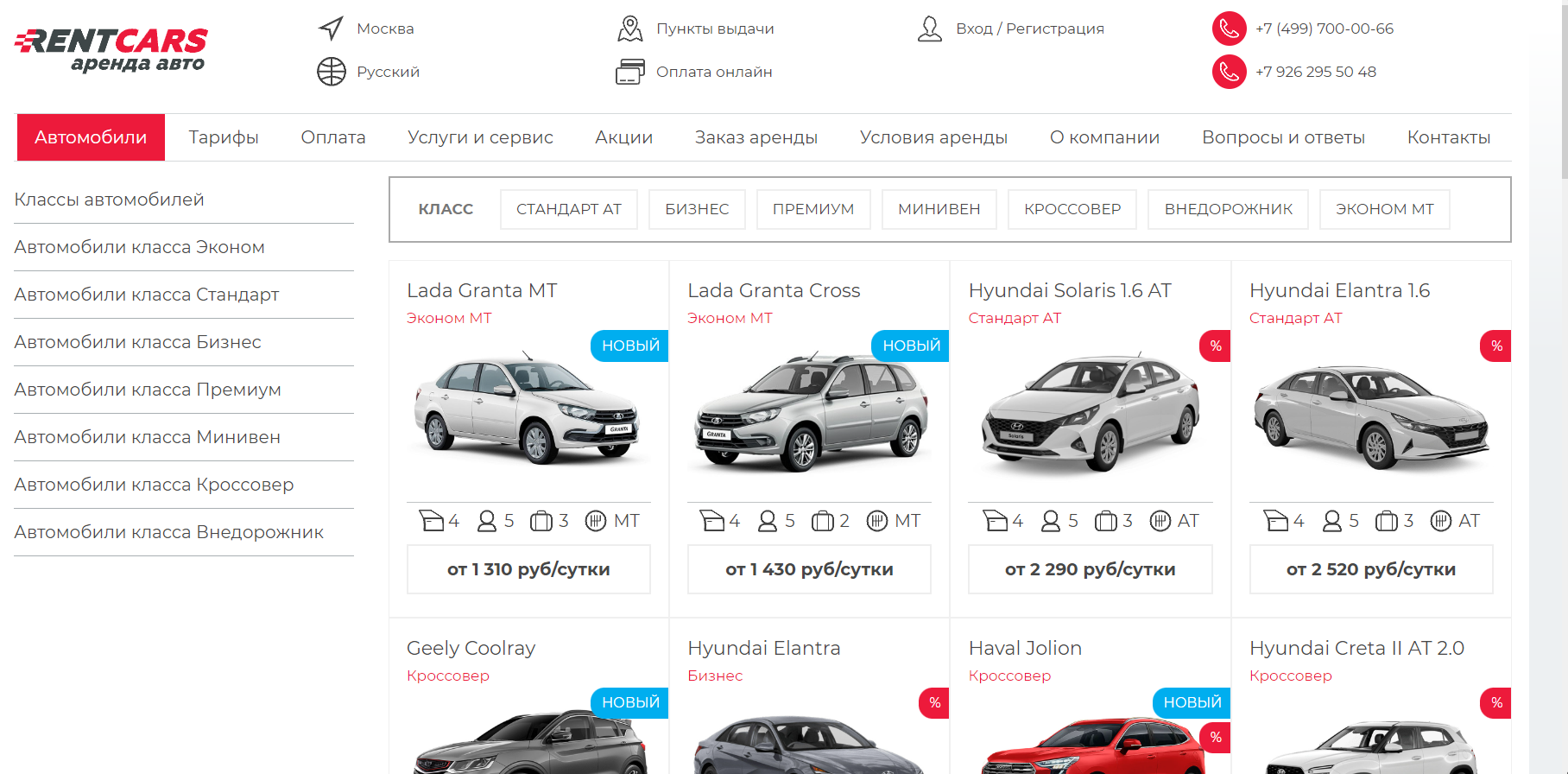


Рисунок 1.3 – Окно с каталогом автомобилей

В этом окне есть возможность выбрать класс автомобиля, нажав на соответствующую кнопку. Результат выбора класса представлен на рисунке 1.4.

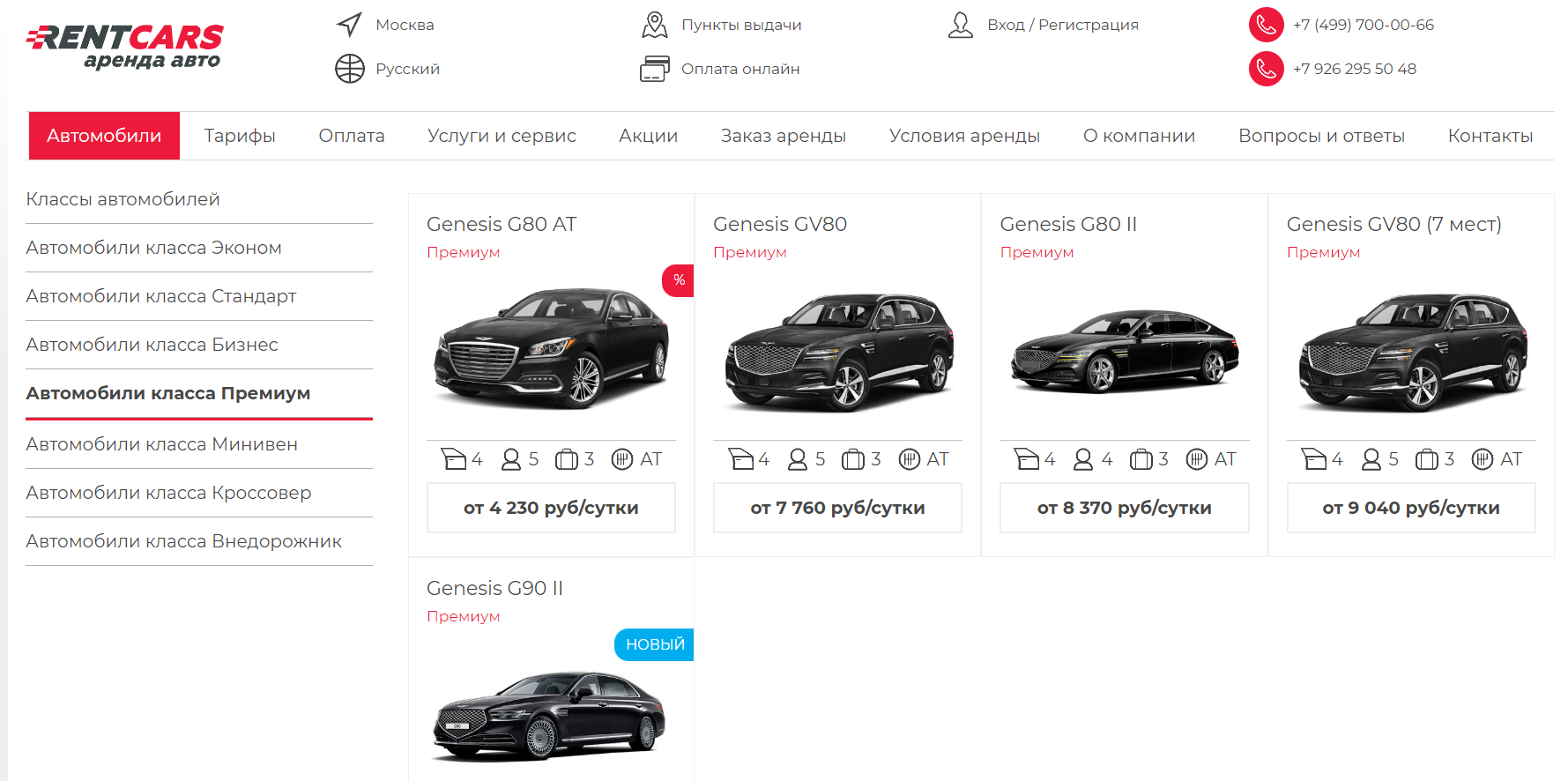


Рисунок 1.4 – Окно с автомобилями класса “Премиум”

1.2 Постановка задачи

Проанализировав прототип, были выделены основная задача и функциональные требования проекта.

Целью курсового проекта стало написание десктопного приложения, позволяющего выполнять авторизацию и регистрацию пользователей, пользователю в лице администратора просматривать заявки пользователей и обрабатывать их. Пользователю в лице клиента доступна возможность выбрать автомобиль по нужным критериям и арендовать его.

Функциональные требования описаны в главе 2.

Глава 2. Анализ требований к программному средству и разработка функциональных требований

Для начала этапа разработки необходимо сформулировать функциональные требования к программному средству.

2.1 Описание средств разработки

Задачей курсового проекта является разработка приложения, предназначенного для аренды автомобилей. Пользователи могут подать заявку на аренду автомобиля, затем администратор обрабатывает эту заявку.

Программный продукт реализован на объектно-ориентированном языке C# с использованием технологий WPF и WCF.

Windows Communication Foundation (WCF) – программный фреймворк, используемый для обмена данными между приложениями, входящий в состав .NET Framework.

Windows Presentation Foundation (WPF) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Window Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

База данных была разработана в приложении Microsoft SQL Server.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире.

SQL Server характеризуется такими особенностями как:

* Производительность. SQL Server работает очень быстро.
* Надежность и безопасность. SQL Server предоставляет шифрование данных.
* Простота. С данной СУБД относительно легко работать и вести администрирование.

Для организации баз данных MS SQL Server использует реляционную модель, которая предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language).

Клиент (например, внешняя программа) отправляет запрос на языке SQL должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Основной используемый язык запросов – Transact-SQL – реализован на структурированном языке запросов (SQL) с расширениями.

Для осуществления связи между базой данных и приложением на C# необходим посредник, которым будет являться технология Entity Framework. Она основана на платформе .NET Framework и предназначена для работы с данными.

Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Эта технология предоставляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Entity Framework предполагает три возможных способа взаимодействия с базой данных:

* database first: Entity Framework создает набор классов, которые отражают модель конкретной базы данных;
* model first: сначала разработчик создает модель базы данных, по которой затем Entity Framework создает реальную базу данных на сервере;
* code first: разработчик создает класс модели данных, которые будут храниться в базе данных, а затем Entity Framework по этой модели генерирует базу данных и ее таблицы.

В курсовом проектировании для разработки был выбран третий подход.

Приложение необходимо реализовать с использованием шаблона проектирования MVVM и с организацией взаимодействия с базой данных. В качестве клиентской части необходимо реализовать оконное приложение с использованием стандартных и иных библиотек. Приложение должно иметь возможность запускаться без использования интегрированных средств разработки.

2.2 Определение требований к программному средству

Для разработанного продукта необходимо предусмотреть следующие задачи:

* главное поле для отображения данных;
* интуитивно понятный интерфейс;
* механизм поиска автомобилей по их классу, характеристикам;

По итогу работы над продуктом, его необходимо протестировать и отладить, проанализировать его возможности и, по необходимости, внести дополнения или улучшения проекта, с возможностью дальнейшего расширения проекта.

Программное средство предоставляет пользователю следующие функциональные возможности:

* регистрация пользователей;
* авторизация пользователей;
* фильтрация автомобилей;
* просмотр характеристик конкретного автомобиля;
* отправка заявки на аренду;
* просмотр статуса заявки;
* отменить заявку;

Администратор имеет следующие возможности:

* добавление и удаление автомобилей из каталога;
* получать заявки от пользователей;
* возможность обрабатывать заявки.

2.3 Описание функциональности программного средства

Описание функциональности программного средства представлено на UML-схеме, изображённой на рисунке 2.1.

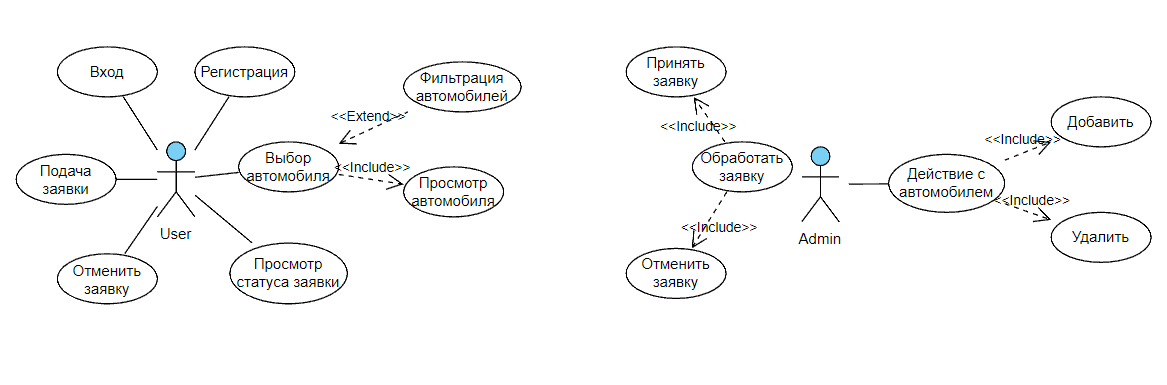


Рисунок 2.1 – UML-диаграмма возможностей