Министерство образования и молодежной политики Свердловской области государственное автономное профессиональное образовательное учреждение Свердловской области

«Уральский радиотехнический колледж им. А.С. Попова»

Разработка автоматизированной информационной системы для организации работы каршеринга "Fast & Furious"

Пояснительная записка к курсовому проекту

По Пм.02 Участие в разработке информационных систем

По МДК 02.01 Информационные технологии и платформы разработки информационных систем

РК 09.02.04.411 08 ПЗ

Нормоконтролер

\_\_\_\_\_\_\_\_\_\_\_\_В.А. Шамшеев

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А. Шамшеев

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Разработчик

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н. Е. Котёлкин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Содержание

[Введение 3](#_Toc155710836)

[1 Постановка задачи 5](#_Toc155710837)

[2 Системный проект 7](#_Toc155710838)

[2.1 Описание предметной области 7](#_Toc155710839)

[2.2 Диаграммы вариантов использования 9](#_Toc155710840)

[2.3 Сценарии вариантов использования 10](#_Toc155710841)

[2.4 Спецификация системы 22](#_Toc155710851)

[2.5 Логическая структура базы данных 25](#_Toc155710852)

[3 Технический проект 27](#_Toc155710853)

[3.1 Выбор состава технических и программных средств 27](#_Toc155710854)

[3.2 Физическая структура базы данных 30](#_Toc155710855)

[3.3 Физическая структура программы 32](#_Toc155710856)

[4 Экспериментальная часть 41](#_Toc155710857)

[4.1 Тестирование приложения 41](#_Toc155710858)

[4.2 Руководство пользователя 46](#_Toc155710859)

[Заключение 59](#_Toc155710860)

[Список использованных источников 60](#_Toc155710861)

# Введение

Автоматизированная информационная система (АИС) представляет собой комплекс взаимосвязанных компонентов и программных средств, предназначенных для сбора, хранения, обработки, анализа и передачи информации в рамках определенной деятельности или предприятия. Она автоматизирует различные бизнес-процессы, упрощает работу с данными и обеспечивает эффективное взаимодействие между участниками системы.

Каршеринг – это система предоставления временного доступа к автомобилю по запросу, обычно на короткий срок, с оплатой за фактическое использование. Эта услуга предоставляется компаниями каршеринга, которые поддерживаю автопарк и обеспечивают пользователей транспортными средствами, доступными для аренды посредством мобильных приложений, веб-платформ и других систем бронирования.

Существует подобная система, которая называется кикшеринг. Отличается она тем, что предоставляет доступ не к автомобилям, а к средствам индивидуальной мобильности, таким как электросамокаты, электровелосипеды и прочее.

Актуальность курсового проекта обусловлена увеличением урбанизации и ростом числа городских жителей, что порождает необходимость в эффективных и устойчивых системах транспорта. Каршеринг представляет собой альтернативу традиционным средствам передвижения, таким как общественный транспорт или личный автотранспорт. Также он предоставляет возможность сократить расходы на содержание личного автомобиля, поскольку отпадают расходы на обслуживание, страхование и парковку. В частности, каршеринг способствует снижению затрат на топливо, стоимость которого постоянно увеличивается.

Тема курсового проекта также актуальна с бизнес-позиции, поскольку в настоящее время на рынке каршеринга практически отсутствуют конкуренты, что, безусловно, представляет собой положительный аспект.

Программный комплекс АИС, который разрабатывается для каршеринга, представляет собой отдельную экосистему, состоящую из приложения для администрирования и приложения для пользователей.

# 1 Постановка задачи

Целью курсового проекта является разработка автоматизированной информационной системы для каршернига «Fast & Furious».

Для осуществления обозначенной цели служит решение следующих задач:

1. Анализ предметной области и постановка задачи;
2. Анализ информационных потребностей;
3. Выбор состава технических и программных средств;
4. Проектирование диаграмм вариантов использования;
5. Проектирование сценариев вариантов использования;
6. Проектирование Логической структуры базы данных;
7. Проектирование Физической структуры базы данных;
8. Разработка базы данных для автоматизированной информационной системы;
9. Разработка приложения для АИС каршеринга «Fast & Furious»;
10. Тестирование приложения;
11. Создание Руководство пользователя.

Для работы с информационной системой каршеринга «Fast & Furious» были созданы следующие роли:

* Администратор – это лицо, обладающее полным доступом к системе и выполняющее обслуживающие функции для АИС, такие как обновление автопарка, регулирование цен, обновление списка пользователей, а также предоставление прав администратора другим пользователям при необходимости.
* Клиент – это лицо, обладающее ограниченным доступом к системе, которое использует предоставляемые услуги.

В информационной системе должен быть следующий функционал:

Данная ИС будет хранить информацию:

* Об автомобилях (Марка, модель, государственный номер, стоимость аренды);
* Информацию о пользователях (Имя, фамилия, номер телефона. адрес электронной почты, серия и номер водительского удостоверения, логин, пароль).

Эта информационная система будет обладать такими функциями как:

* Редактирование автопарка (Добавление и удаление автомобилей);
* Редактирование пользователей;
* Создание новых пользователей;
* Удаление пользователей;
* Формирование отчётов;
* Просмотр доступных автомобилей на карте;
* Просмотр доступных вариантов оформления платной подписки;

2 Системный проект

2.1 Описание предметной области

Принцип работы предприятия до внедрения автоматизированной информационной системы:

Клиенты, желающие воспользоваться услугами каршеринга, должны были бы связаться с оператором через телефон или личное посещение офиса. Заявки на бронирование принимались бы вручную, а оператор занимался проверкой наличия доступных автомобилей в системе.

Также операторы были бы ответственны за установку тарифов и акций, а также их обновление вручную. Регистрация новых клиентов и обновление их данных также проводились бы вручную. Оплата услуг и транзакции проводились бы в офисе или через банковский перевод. Операторы должны были бы следить за финансовыми операциями, вручную обрабатывать платежи и контролировать актуальность оплаты.

Мониторинг состояния автомобилей, их техническое обслуживание, учет расхода топлива, а также выявление и устранение неисправностей требовали бы регулярных проверок и вмешательства технического персонала.

Анализ данных и составление отчетов о работе автопарка и использовании клиентами требовало бы временных и трудовых затрат на сбор и обработку информации.

Принцип работы предприятия после внедрения автоматизированной информационной системы:

Пользователь авторизуется в системе и если он является администратором, то всплывает окно «Запустить окно администрирования?»

Если пользователь выбирает вариант «Да», то открывается окно администрирования, в котором он может редактировать автопарк, например, добавить новый автомобиль или удалить старый. Также он может отредактировать данные пользователей, изменить их роли, уровни подписки и удалить пользователя из системы. Авторизованный как администратор пользователь может сформировать два отчета: по пользователям и по автомобилям.

Если пользователь выбирает вариант «Нет» или не обладает правами администратора, то для него открывается окно клиента. В этом окне он может просмотреть доступные автомобили, выбрать подходящий тарифный план для платной подписки или отредактировать свои данные.

2.2 Диаграммы вариантов использования

На диаграмме вариантов вариантов использования (Рисунок 2.2.1), описан процесс работы каршеринга. Изображены основные составляющие деятельности каршеринга. Пользователь проходит авторизацию в приложении после определяется его роль. В зависимости от роли открываются различные возможности взаимодействия с автоматизированной информационной системой: такие как просмотр доступных автомобилей, просмотр тарифного плана платной подписки, редактирование профиля у обычного пользователя, администратор может управлять пользователями, автопарком и формировать отчёты.

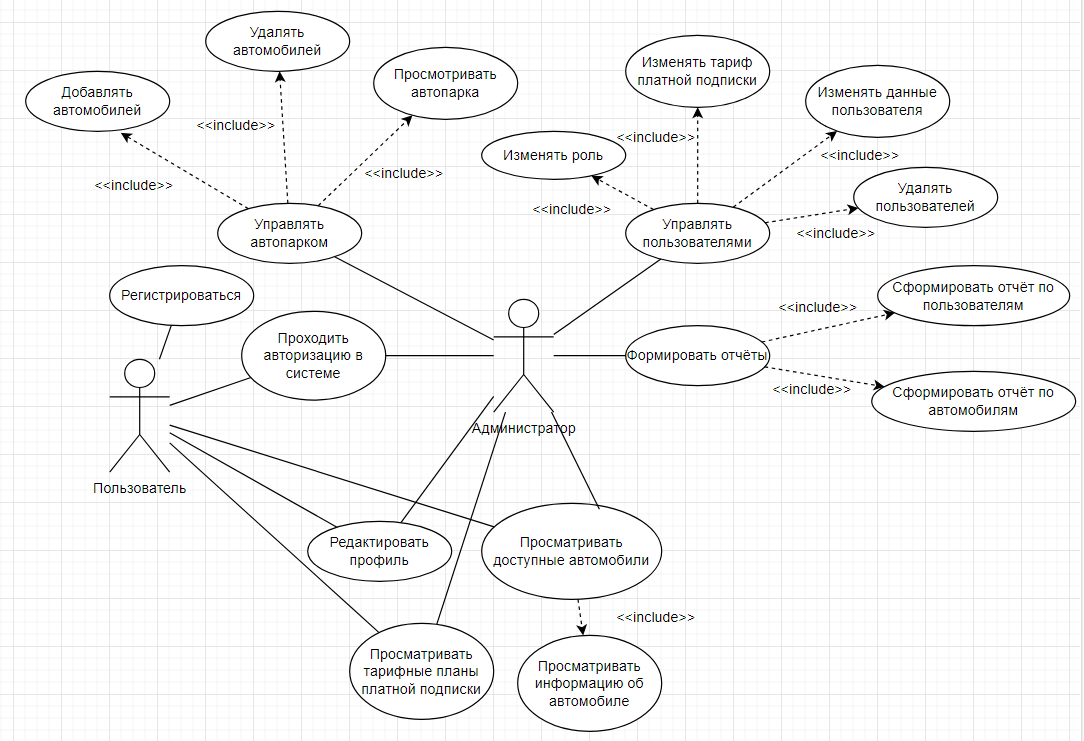


Рисунок 2.2.1 – Диаграмма вариантов использования

2.3 Сценарии вариантов использования

2.3.1 Авторизация

Данный сценарий выполняется, когда пользователь отправляет запрос входа в систему.

Основной поток событий:

1. Система генерирует окно авторизации.
2. Пользователь заполняет поля «Логин», «Пароль» и отправляет запрос входа в систему.
3. Система проверяет данные и авторизует пользователя.

Альтернативные потоки:

Если пользователь вводит некорректные данные или оставляет пустые поля, то система оповещает его об этом и предлагает исправить недочеты. Пользователь может заново ввести данные или отказаться от заполнения.

Если пользователь является администратором, то система предлагает открыть окно администрирования. Пользователь может выбрать авторизоваться ли ему как администратор или использовать пользовательское окно.

Предусловия:

Пользователь не должен быть авторизован.

2.3.2 Регистрация пользователя

Данный сценарий выполняется, когда пользователь отправляет запрос регистрации.

Основной поток событий:

1. Система генерирует окно регистрации.
2. Пользователь заполняет поля «Логин», «Пароль», «Номер телефона» и отправляет запрос регистрации.
3. Система добавляет данные в базу данных и даёт пользователю роль «user».

Альтернативные потоки:

Если пользователь оставляет пустые поля, то система оповещает его об этом и предлагает исправить недочеты.

Если система обнаружит, что такой логин или телефон уже имеются в базе данных, то оповещает об этом пользователя и предлагает исправить недочёты.

Если система обнаружит, что пароль пользователя не подходит под условия создания пароля, то оповещает об этом пользователя и предлагает исправить недочеты.

Пользователь может заново ввести данные или отказаться от заполнения.

Предусловия:

Пользователь не должен быть авторизован.

2.3.3 Просмотр автопарка

Данный сценарий выполняется, когда пользователь отправляет запрос просмотра автопарка.

Основной поток событий:

1. Система генерирует страницу просмотра автопарка.
2. Пользователь просматривает автопарк и видит необходимые ему данные.

Альтернативные потоки:

Если в базе данных не числятся автомобили, то администратору необходимо их добавить.

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «admin».

2.3.4 Добавление автомобиля в автопарк

Данный сценарий выполняется, когда пользователь отправляет запрос добавления автомобиля в автопарк.

Основной поток событий:

1. Система генерирует страницу добавления автомобиля в автопарк.
2. Пользователь заполняет поля «Марка», «Модель», «Государственный номер», «Цена», «Широта», «Долгота» и отправляет запрос добавления автомобиля в автопарк.
3. Система добавляет данные в базу данных.

Альтернативные потоки:

Если пользователь оставляет пустые поля, то система оповещает его об этом и предлагает исполнить недочеты.

Пользователь может заново ввести данные или отказаться от заполнения.

Предусловия:

Пользователь должен быть авторизован под аккаунтом c уровня доступа «admin».

2.3.5 Удаление автомобилей из автопарка

Данный сценарий выполняется, когда пользователь отправляет запрос удаления автомобилей из автопарка.

Основной поток событий:

1. Система генерирует страницу удаления автомобилей из автопарка.
2. Пользователь отправляет запрос удаления автомобиля из автопарка.
3. Система удаляет автомобиль из базы данных.

Альтернативный поток:

Если в системе нету автомобилей, то пользователю необходимо добавить в систему автомобили.

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «admin».

2.3.6 Просмотр списка администраторов

Данный сценарий выполняется, когда пользователь отправляет запрос просмотра списка администраторов.

Основной поток событий:

1. Система генерирует страницу просмотра списка администраторов.
2. Пользователь просматривает список администраторов и видит необходимые ему данные.

Альтернативные потоки:

Пользователь может отредактировать аккаунт из списка администраторов, изменить следующие поля: «Имя», «Фамилия», «Пароль», «Уровень доступа».

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «admin».

2.3.7 Просмотр списка клиентов

Данный сценарий выполняется, когда пользователь отправляет запрос просмотра списка клиентов.

Основной поток событий:

1. Система генерирует страницу просмотра списка клиентов.
2. Пользователь просматривает список клиентов и видит необходимые ему данные.

Альтернативные потоки:

Пользователь может отредактировать аккаунт из списка клиентов, изменить следующие поля: «Имя», «Фамилия», «Пароль», «Уровень доступа», «Уровень подписки».

Пользователь может удалить аккаунт из списка клиентов.

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «admin».

2.3.8 Просмотр статистики по пользователям

Данный сценарий выполняется, когда пользователь отправляет запрос просмотра статистики по пользователям.

Основной поток событий:

1. Система генерирует страницу просмотра статистики по пользователям.
2. Пользователь просматривает список администраторов и список пользователей.

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «admin».

2.3.9 Просмотр статистики по автопарку

Данный сценарий выполняется, когда пользователь отправляет запрос просмотра статистики по автопарку.

Основной поток событий:

1. Система генерирует страницу просмотра статистики по автопарку.
2. Пользователь просматривает статистику по автопарку и видит необходимые ему данные.

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «admin».

2.3.10 Просмотр тарифных планов платной подписки

Данный сценарий выполняется, когда пользователь отправляет запрос просмотра тарифных планов платной подписки.

Основной поток событий:

1. Система генерирует страницу просмотра тарифных планов платной подписки.
2. Пользователь просматривает список тарифов и выбирает подходящий ему тарифный план.
3. Пользователь оплачивает выбранный тарифный план.
4. Система проверяет поступление оплаты и устанавливает пользователю выбранный уровень подписки.

Альтернативные потоки:

Пользователь просматривает список тарифов и ничего не выбирает.

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «user» или «admin».

2.3.11 Редактирование профиля

Данный сценарий выполняется, когда пользователь отправляет запрос редактирования профиля.

Основной поток событий:

1. Система генерирует страницу личного кабинета.
2. Пользователь отправляет запрос редактирования профиля.
3. Система генерирует страницу редактирования профиля.
4. Пользователь заполняет поля «Имя», «Фамилия», «Электронная почта», «Водительское удостоверение».
5. Система вносит данные в базу данных.

Альтернативные потоки:

Если пользователь не заполнил поле «Водительское удостоверение» или заполнил неверно, то система оповещает его об этом и предлагает исправить недочеты.

Пользователь может не заполнять поля «Имя», «Фамилия», «Электронная почта».

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «user» или «admin».

* + 1. Аренда автомобиля

Данный сценарий выполняется, когда пользователь отправляет запрос аренды автомобиля.

Основной поток событий:

1. Система генерирует страницу карты и маркеры с местонахождением свободных автомобилей.
2. Пользователь выбирает автомобиль.
3. Система генерирует страницу информации об автомобиле.
4. Пользователь отправляет запрос аренды автомобиля.
5. Система изменяет статус автомобиля на «Занята» и даёт пользователю доступ к автомобилю.

Альтернативные потоки:

Пользователь выбирает автомобиль и отправляет запрос на его бронирование. Система изменяет статус автомобиля на «Забронирована» и ожидает начала аренды

Предусловия:

Пользователь должен быть авторизован под аккаунтом с уровнем доступа «user» или «admin».

2.4 Спецификация системы

Таблица 1 – Сущность «Автомобили»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| Id автомобиля | Числовой | Id автомобиля | Первичный ключ |
| Марка | Текстовый | Марка | Максимальный размер 50 |
| Модель | Текстовый | Модель | Максимальный размер 100 |
| Государственный номер | Текстовый | Государственный номер | Максимальный размер 9 |
| Цена | Числовой | Цена аренды | Вещественный |
| Долгота | Числовой | Координата по оси Y | Вещественный |
| Широта | Числовой | Координата по оси X | Вещественный |
| Id статуса | Числовой | Статус автомобиля | Внешний ключ к сущности «Статусы» |

Таблица 2 – Сущность «Статусы»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| Id статуса | Числовой | Id статуса | Первичный ключ |
| Название | Текстовый | Название статуса | Максимальный размер 50 |

Таблица 3 – Сущность «Пользователи»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| Id пользователя | Числовой | Id пользователя | Первичный ключ |
| Имя | Текстовый | Имя пользователя | Максимальный размер 255 |
| Фамилия | Текстовый | Фамилия пользователя | Максимальный размер 255 |
| Логин | Текстовый | Логин пользователя | Максимальный размер 50 |
| Электронная почта | Текстовый | Электронная почта пользователя | Максимальный размер 255 |
| Телефон | Текстовый | Номер телефона | Максимальный размер 255 |
| Пароль | Текстовый | Пароль пользователя | Максимальный размер 32 |
| Водительское удостоверение | Текстовый | Водительское удостоверение пользователя | Максимальный размер 12 |
| Id роли | Числовой | Роль в системе | Внешний ключ к сущности «Роли» |
| Id подписки | Числовой | Тариф платной подписки | Внешний ключ к сущности «Подписки» |

Таблица 4 – Сущность «Подписки»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| Id подписки | Числовой | Id подписки | Первичный ключ |
| Название | Текстовый | Название подписки | Максимальный размер 255 |

Таблица 5 – Сущность «Роли»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| Id роли | Числовой | Id роли | Первичный ключ |
| Название | Текстовый | Название роли | Максимальный размер 255 |

Таблица 6 – Сущность «Забронированные автомобили»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Описание | Примечание |
| 1 | 2 | 3 | 4 |
| Id брони | Числовой | Id брони | Первичный ключ |
| Id пользователя | Числовой | Id пользователя | Внешний ключ к сущности «Пользователи» |
| Id автомобиля | Числовой | Id автомобиля | Внешний ключ к сущности «Автомобили» |
| Id статуса | Числовой | Id статуса | Внешний ключ к сущности «Статусы» |

2.5 Логическая структура базы данных

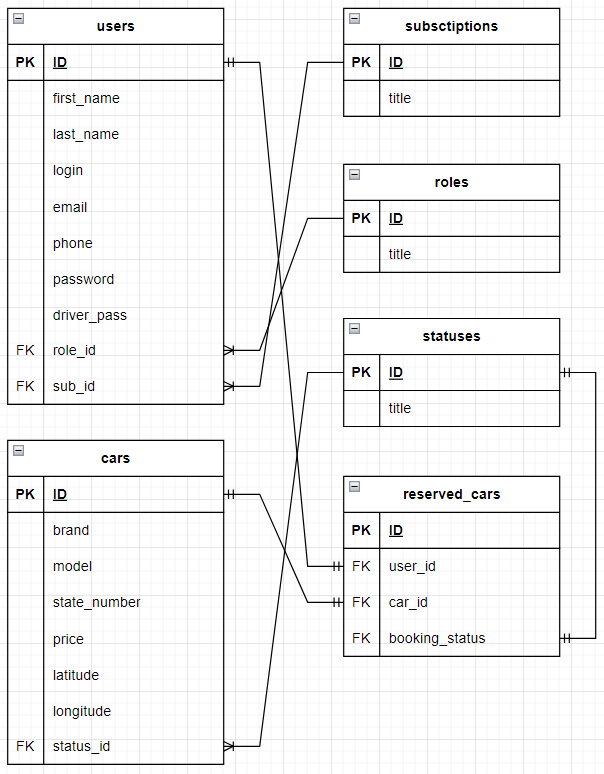


Рисунок 2.5.1 – Логическая структура базы данных.

Проведя анализ соответствия модели автоматизируемых процессов и функциональной модели информационной системы, можно сделать вывод о том, что существуют процессы:

* Учет пользователей
* Учет автомобилей
* Учет статусов автомобилей
* Учет забронированных автомобилей
* Учет подписок
* Учет ролей

3 Технический проект

3.1 Выбор состава технических и программных средств

Согласно задачам, поставленным в курсовой работе, необходимо разработать программу, предназначенную для выполнения на операционной системе Windows. Учитывая, что разработанный алгоритм предполагает использование графических возможностей и параллельных процессов для взаимодействия с пользователем, было принято решение использовать следующие программные средства:

* Среда программирования Visual Studio Community 2022;
* Язык программирования C#;
* Операционная система Windows 10;
* Платформа Windows Presentation Foundation (WPF);
* Система управления базами данных PostgreSQL;
* Веб-интерфейс для работы с PostgreSQL PgAdmin4;

Visual Studio – это интегрированная среда разработки (IDE), предназначенная для создания разнообразных приложений. Она обеспечивает широкие возможности для разработки программного обеспечения под различные платформы, включая Windows, Android, iOS, и веб-приложения. Visual Studio предоставляет удобный интерфейс для написания, отладки и тестирования кода, а также инструменты для создания графических интерфейсов, управления версиями, исследования производительности и другие возможности, упрощающие разработку приложений. Включает в себя поддержку множества языков программирования, интегрированный отладчик и обширную библиотеку ресурсов.

C# (C-Sharp) - высокоуровневый объектно-ориентированный язык программирования, созданный Microsoft. Его ключевые особенности включают:

* Синтаксис, аналогичный C++ и Java, что облегчает переход между ними.
* Поддержка принципов ООП: наследование, инкапсуляция, полиморфизм, способствующая созданию модульного кода.
* Компиляция в Common Intermediate Language (CIL), делая его языком с управляемым кодом с автоматической сборкой мусора и безопасностью типов через CLR (Common Language Runtime).
* Интеграция с .NET: тесная связь с .NET Framework для использования его библиотеки классов.
* Мощная система обработки событий для упрощения асинхронного программирования и взаимодействия с GUI.
* Поддержка LINQ для удобной работы с данными из различных источников.
* Встроенная поддержка асинхронного программирования для эффективной обработки операций ввода/вывода и асинхронных задач.
* Средства для создания и управления многопоточными приложениями, повышающие производительность и отзывчивость программ.

Windows Presentation Foundation (WPF) - технология для создания графических пользовательских интерфейсов (GUI) в Windows. Использует декларативный подход с языком разметки XAML, поддерживает векторную графику, стили, анимации, привязку данных и обеспечивает гибкую композицию элементов интерфейса.

PostgreSQL - мощная система управления базами данных с открытым исходным кодом, широко применяемая в разработке приложений. Основные характеристики:

* Открытый исходный код: Пользователи могут свободно использовать, изменять и распространять исходный код.
* Реляционная база данных: Основана на принципах реляционной модели данных, обеспечивая структурированное хранение и эффективный доступ к данным.
* Многопользовательская и многозадачная: Поддерживает одновременное обслуживание множества пользователей и выполнение множества задач.
* ACID-совместимость: Гарантирует надежность и целостность данных через свойства ACID (Atomicity, Consistency, Isolation, Durability).
* Расширяемость: Пользователи могут создавать пользовательские функции, типы данных и языки программирования для расширения функциональности.
* Поддержка структурированных и неструктурированных данных: Обеспечивает хранение и обработку разнообразных данных, включая географические и текстовые.
* Множество расширений: Активное сообщество разработчиков предлагает разнообразные расширения и дополнения для PostgreSQL.
* Кросс-платформенность: Поддерживает различные операционные системы, включая Windows, Linux и macOS.

PgAdmin 4 - бесплатный веб-интерфейс для управления базами данных PostgreSQL. Имеет интуитивный интерфейс, поддерживает многоплатформенность, многоподключение и включает в себя SQL-редактор. Обеспечивает мониторинг, отчетность, управление пользователями и безопасность баз данных. Поддерживает расширения для дополнительной функциональности.

3.2 Физическая структура базы данных

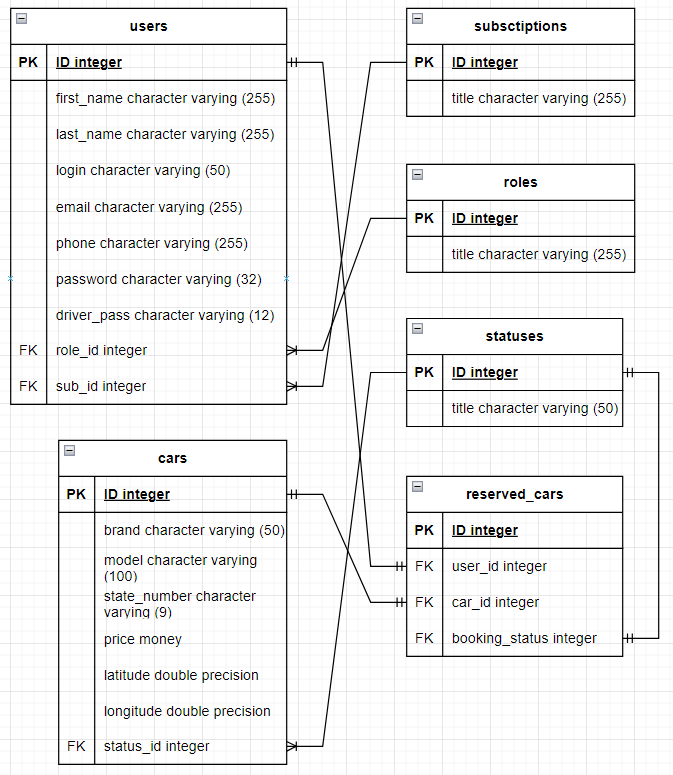


Рисунок 3.2.1 – Физическая структура базы данных

База данных включает в себя несколько таблиц:

1. «users» с атрибутами «id» (первичный ключ), «first\_name», «last\_name», «login», «email», «phone», «password», «driver\_pass», «role\_id»

(внешний ключ, связанный с «roles.id»), «sub\_id» (внешний ключ, связанный с «subscriptions.id»);

1. «cars» с атрибутами «id» (первичный ключ), «brand», «model», «state\_number», «price», «latitude», «longitude», «status\_id» (внешний ключ, связанный с «statuses.id»);
2. «statuses» с атрибутами «id» (первичный ключ), «title»;
3. «roles» с атрибутами «id» (первичный ключ), «title»;
4. «subscriptions» с атрибутами «id» (первичный ключ), «title»;
5. «reserved\_cars» с атрибутами «id» (первичный ключ), «users\_id» (внешний ключ, связанный с «users.id»), «car\_id» (внешний ключ, связанный с «cars.id»), «booking\_status» (внешний ключ, связанный с «statuses.id»);

3.3 Физическая структура программы

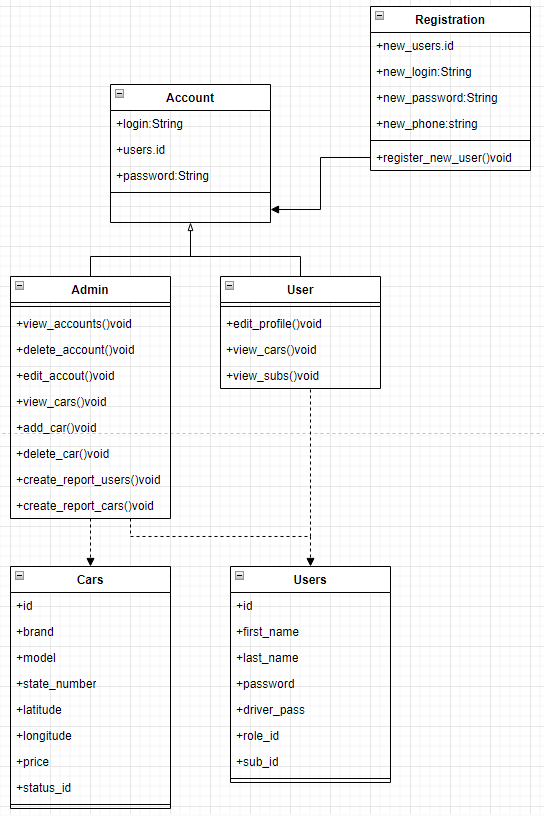


Рисунок 3.3.1 – Физическая структура программы.

Отношения между классами:

Класс «Admin» содержит функции: +view\_accounts()void, +delete\_account()void, +edit\_account()void, +view\_cars()void, +add\_car()void, +delete\_car()void, +create\_report\_users()void.

Класс «User» содержит функции: +edit\_profile()void, +view\_cars()void, +view\_subs()void.

Класс «Registration» содержит функцию: +register\_new\_user()void и поля: +new\_users.id, +new\_login, +new\_password, +new\_phone.

Класс «Account» содержит поля: +login, +users.id, +password.

Класс «Users» содержит поля: +id, +first\_name, +last\_name, +password, +driver\_pass, +role\_id, +sub\_id.

Класс «Cars» содержит поля: +id, +brand, +model, +state\_number, +latitude, +longitude, +price, +status\_id.

Все файлы, нужные для работы программы, находятся в папке CarSharing

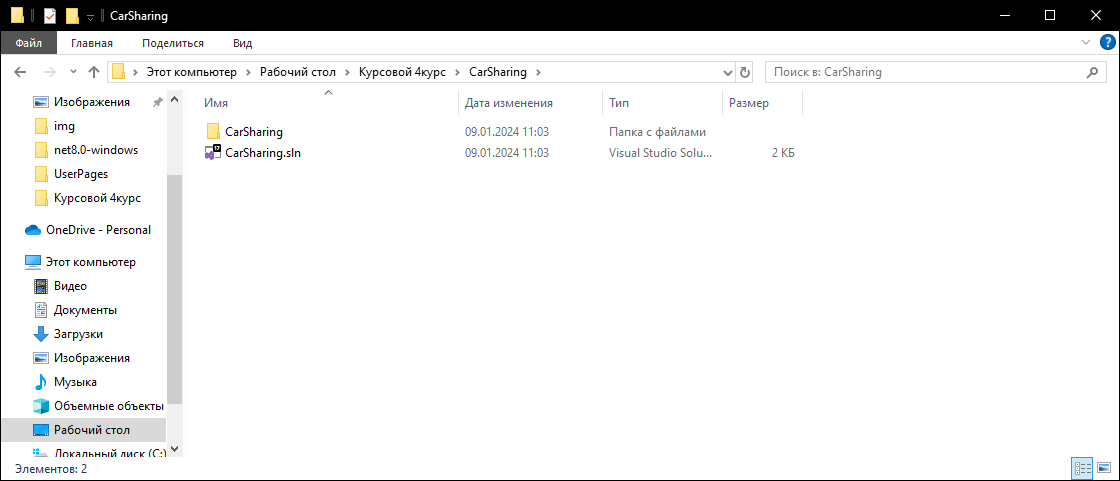


Рисунок 3.3.2 – Физическая структура программы. Папка CarSharing.

В папке находится файл CarSharing.sln, необходимый для запуска конструктора в Visual Studio. Также там присутствует папка CarSharing (Рисунок 3.3.3), в которой содержатся все необходимые файлы для работы приложения, такие как внешняя оболочка, скрипты программы и установленные пакеты.

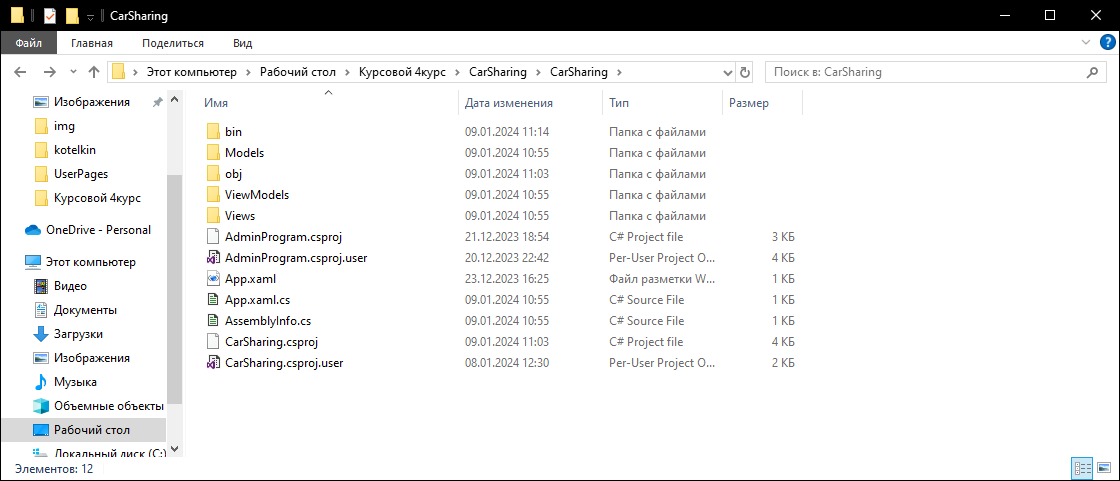


Рисунок 3.3.3 – Физическая структура программы. Папка CarSharing.

На рисунке 3.3.3 показана физическая структура программы, в папке bin находятся двоичные файлы, которые являются фактическим исполняемым кодом нашего приложения.

Также видим папку Models, в которой хранятся объекты, которыми манипулируем в программе (Рисунок 3.3.4).

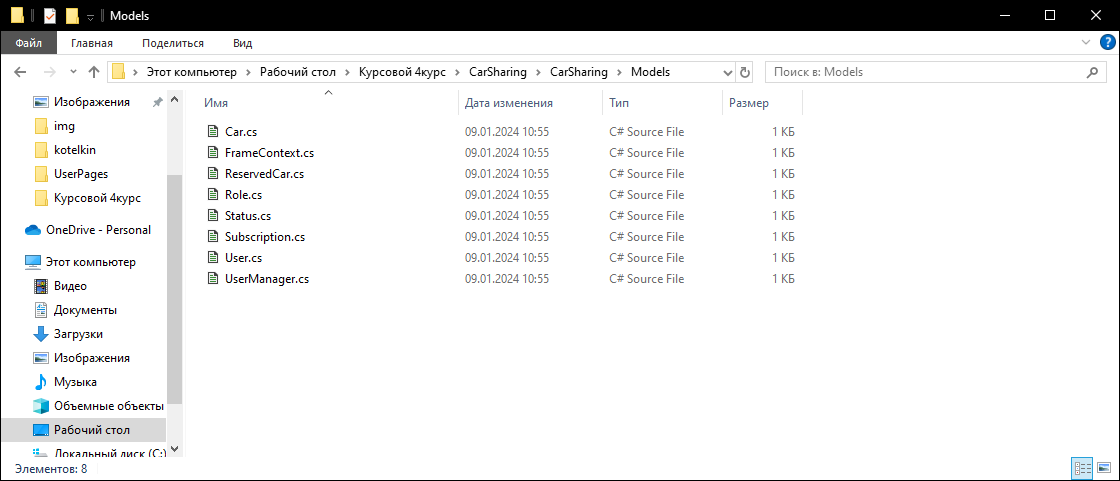


Рисунок 3.3.4 – Физическая структура программы. Папка Models.

Папка obj (Рисунок 3.3.5) содержит объектные или промежуточные файлы, которые представляют собой скомпилированные двоичные файлы, которые ещё не были связаны. Это фрагменты, которые будут объединены для создания окончательного исполняемого файла.

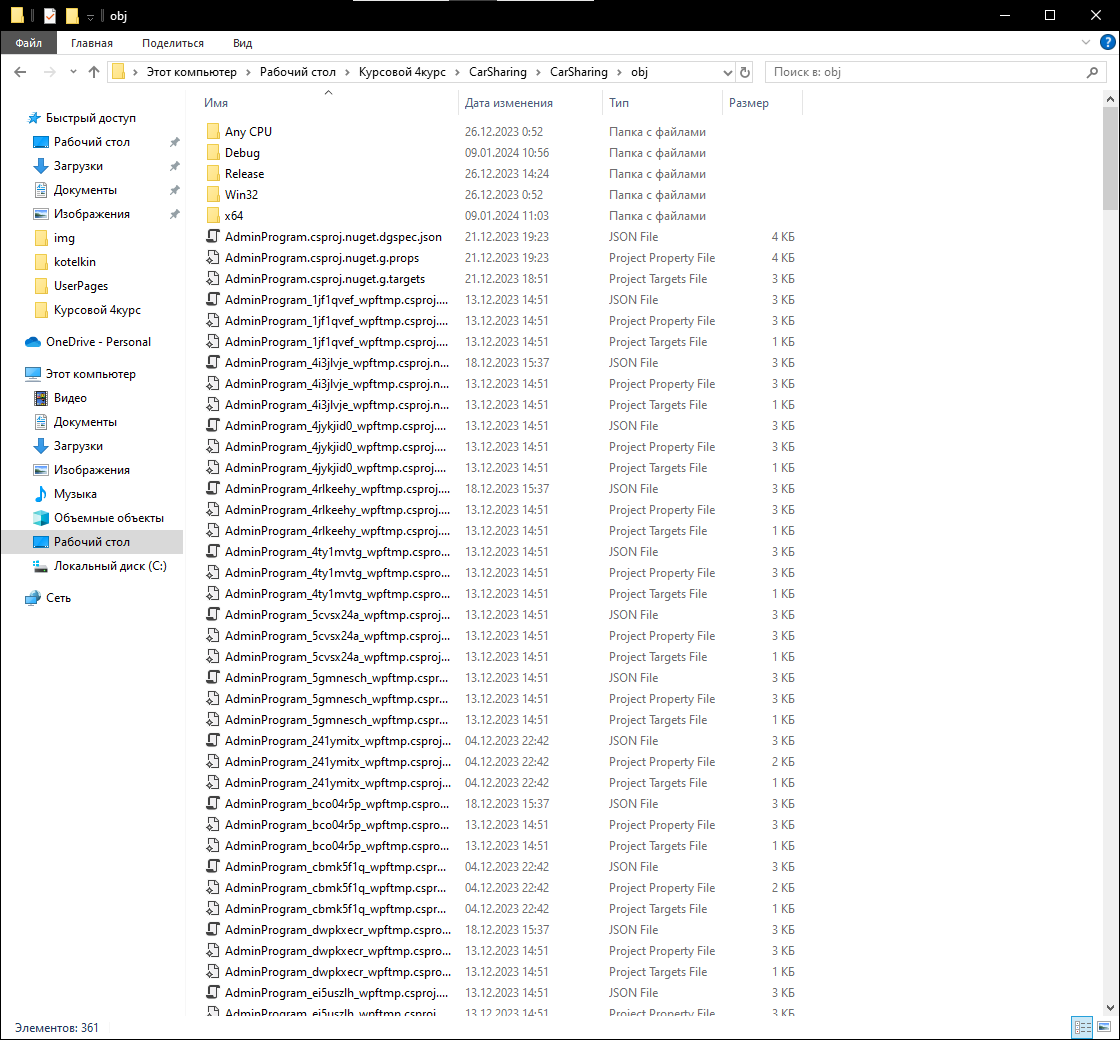


Рисунок 3.3.5 – Физическая структура программы. Папка obj.

Папка ViewModels (Рисунок 3.3.6) содержит файлы, для работы с базой данных. Они отвечают за подключение к базе данных и контроль над ней.

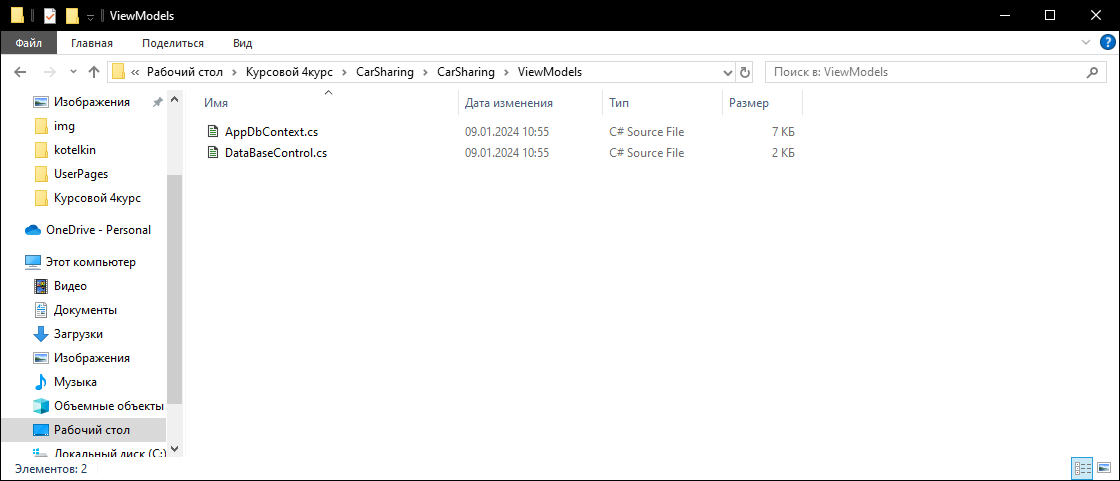


Рисунок 3.3.6 – Физическая структура программы. Папка ViewModels.

Папка Views (Рисунок 3.3.7) предназначена для хранения разметки страниц и изображений, используемых на них. Внутри есть подпапки: AdminWindow (Рисунок 3.3.9) для разметки окна администратора, UserWindow (Рисунок 3.3.х) для разметки окна пользователя, и img (Рисунок 3.3.8) для изображений, используемых в приложении. Также здесь находится разметка и скрипты для окон авторизации и регистрации.

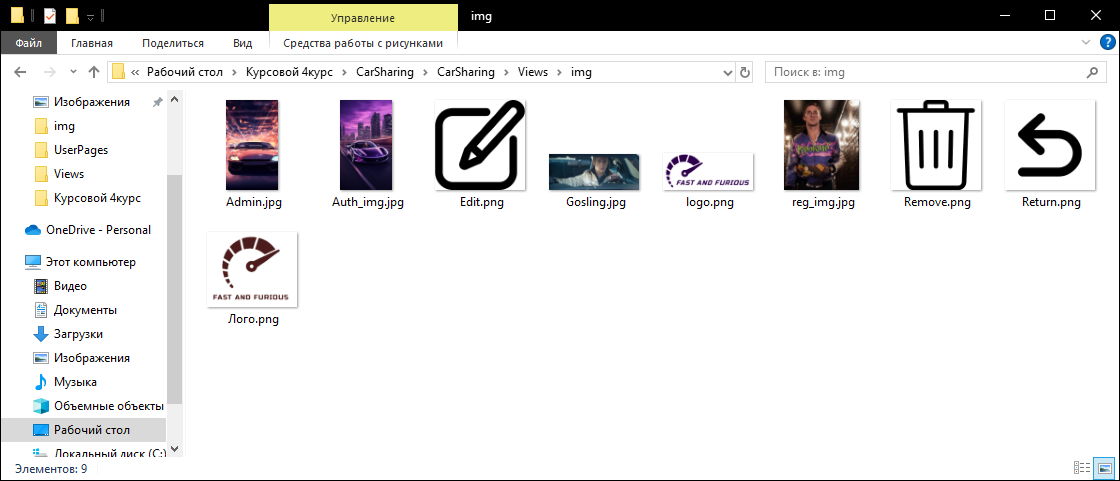


Рисунок 3.3.8 – Физическая структура программы. Папка img.

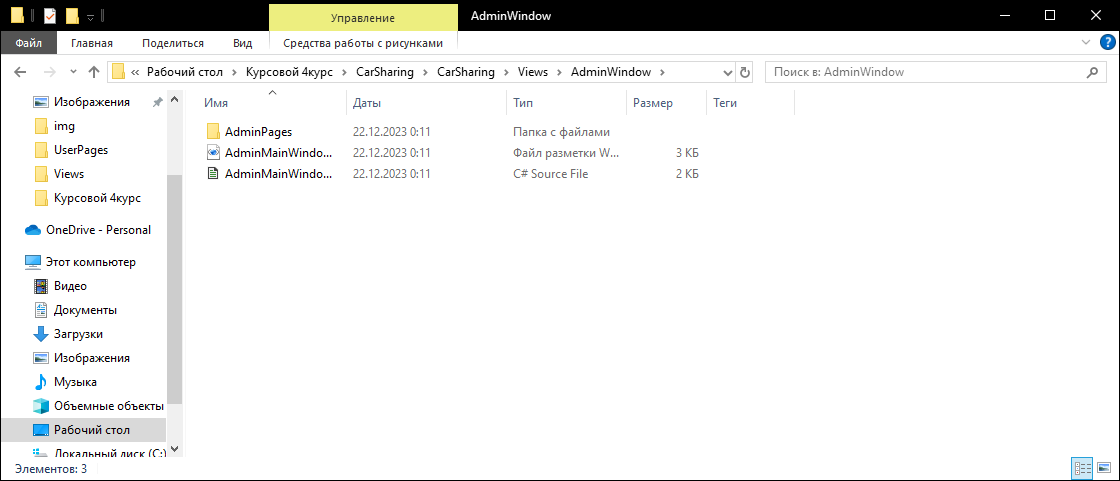


Рисунок 3.3.9 – Физическая структура приложения. Папка AdminWindow.

Папка AdminWindow (Рисунок 3.3.9) содержит оболочку окна администратора, а также включает в себя папку с разметкой для страниц AdminPages (Рисунок 3.3.10).

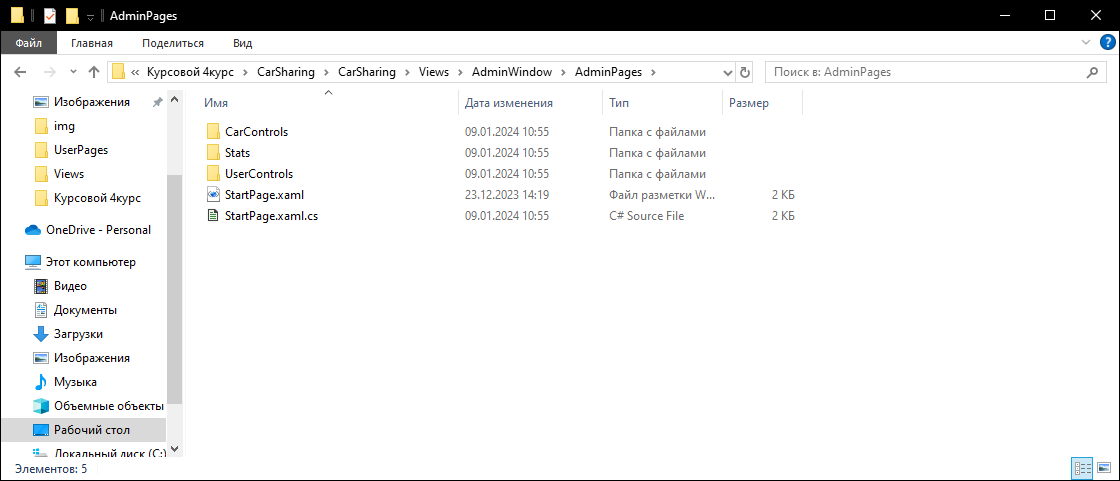


Рисунок 3.3.10 – Физическая структура приложения. Папка AdminPages.

Папка AdminPages (Рисунок 3.3.10) содержит оболочку стартовой страницы окна AdminWindow, а также содержит подпапки с разметкой для страниц CarControls (Рисунок 3.3.11), Stats (Рисунок 3.3.12), UserControls (Рисунок 3.3.13).

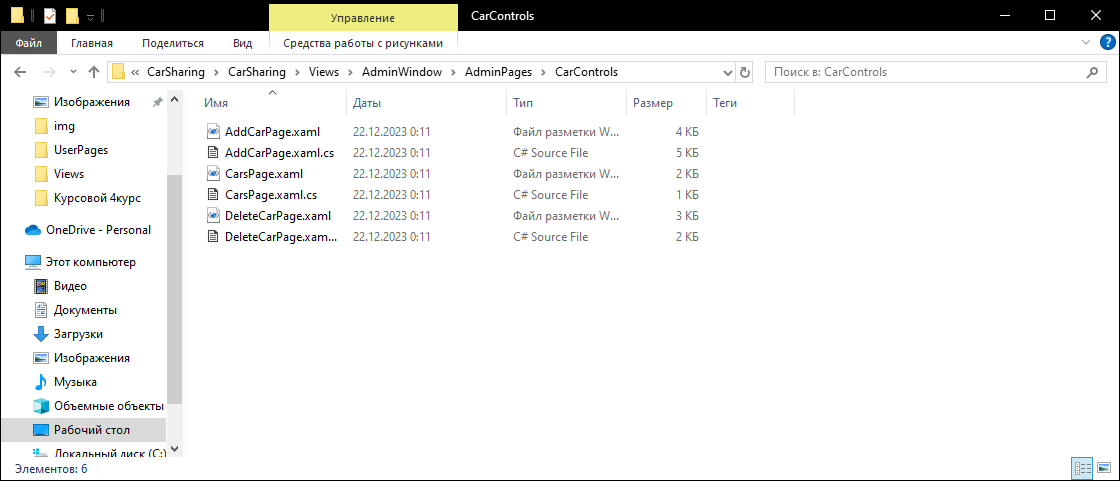


Рисунок 3.3.11 – Физическая структура приложения. Папка CarContorls.

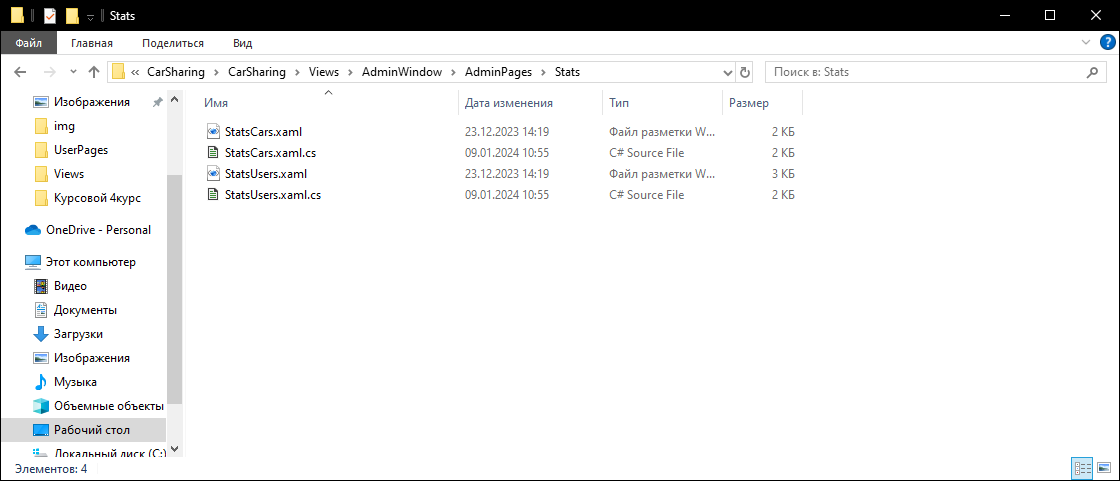


Рисунок 3.3.12 – Физическая структура программы. Папка Stats.

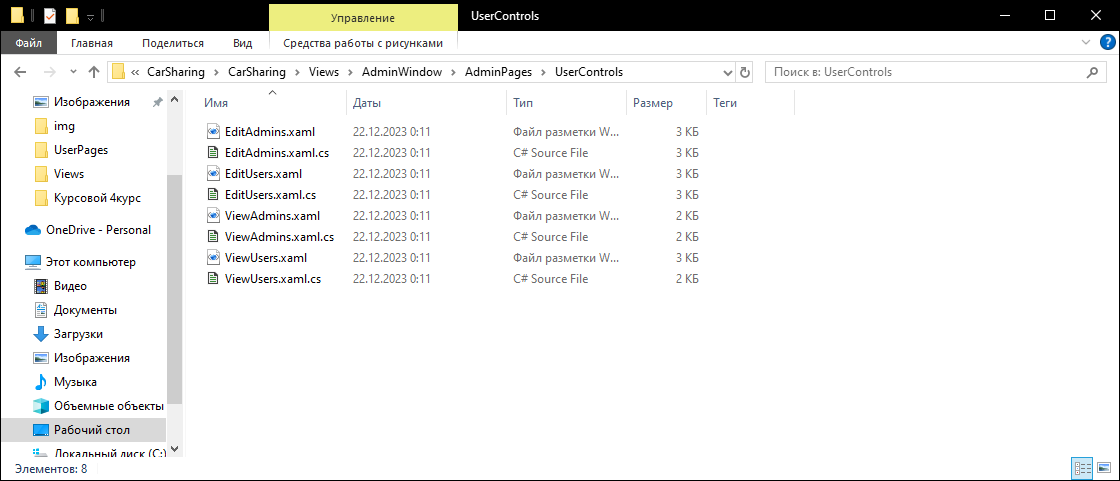


Рисунок 3.3.13 – Физическая структура приложения. Папка UserControls.

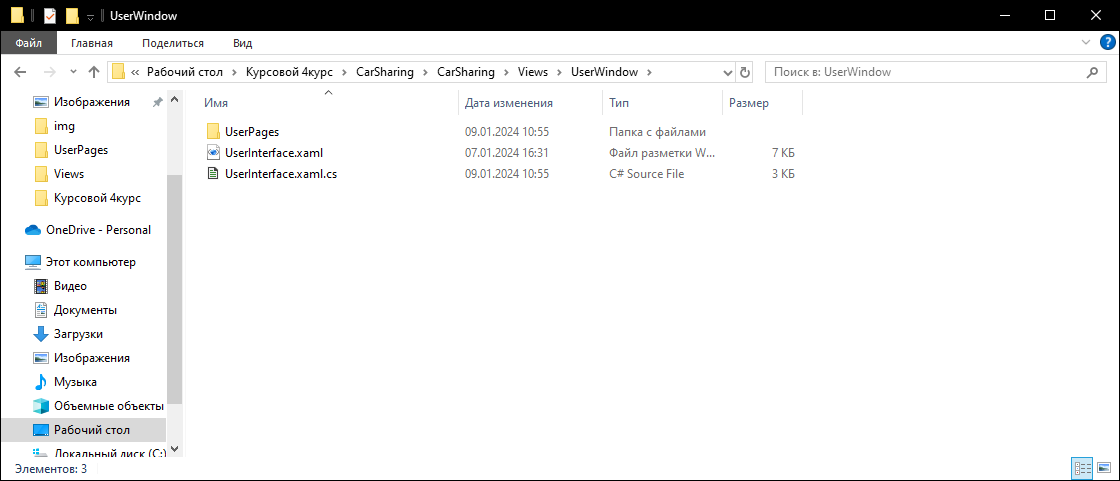


Рисунок 3.3.14 – Физическая структура приложения. Папка UserWindow.

В папке UserWindow (Рисунок 3.3.14) находится разметка окна пользователя, а также подпапка UserPages (Рисунок 3.3.15) с разметкой страниц и необходимыми изображениями для их работы.

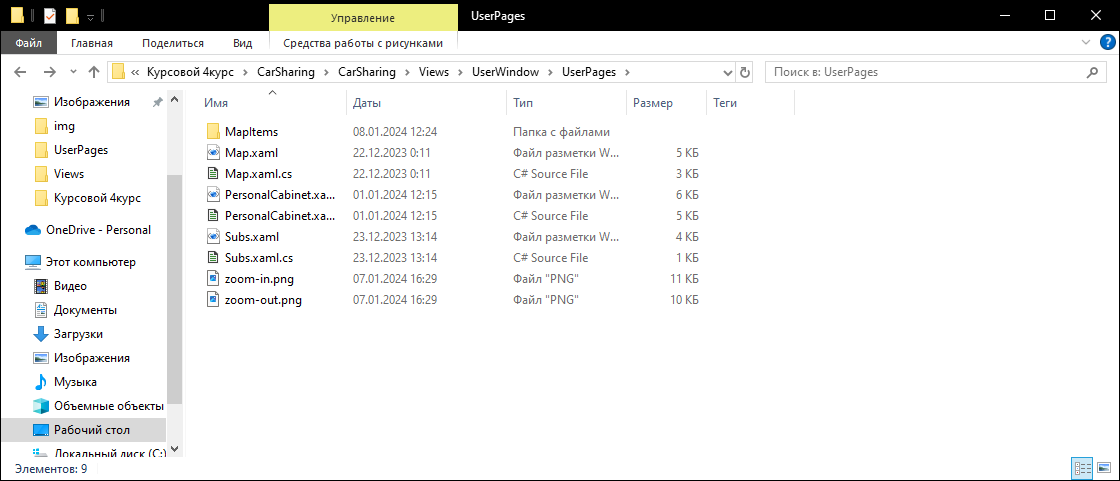


Рисунок 3.3.15 – Физическая структура программы. Папка UserPages.

Помимо разметки и скриптов страниц в папке UserPages (Рисунок 3.3.15) также присутствует подпапка MapItems (Рисунок 3.3.16), содержащая файлы, необходимые для работы с метками на карте.

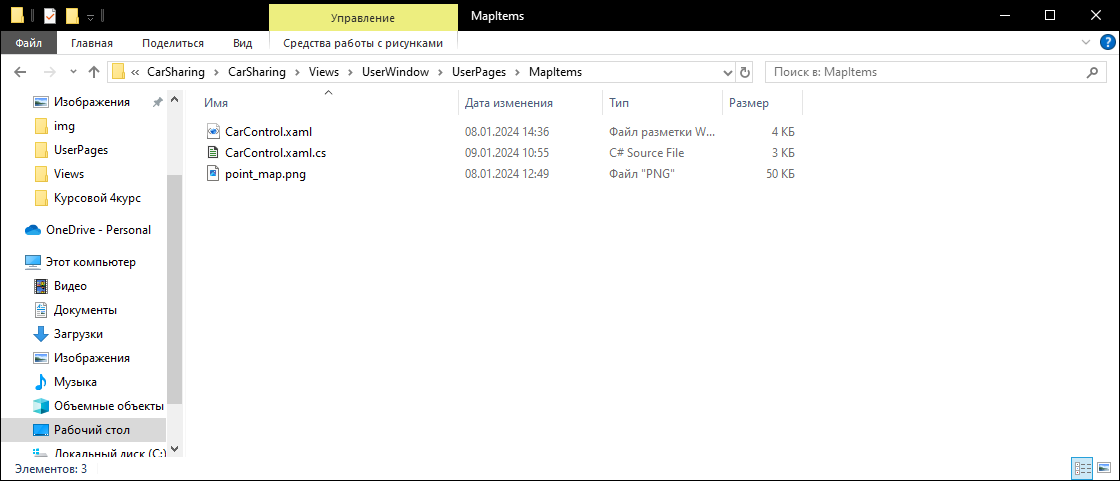


Рисунок 3.3.16 – Физическая структура программы. Папка MapItems.

4 Экспериментальная часть

4.1 Тестирование приложения

Тест: Вход в систему.

Цель теста: проверить корректность процесса входа в систему.

Шаги теста:

1. Открыть приложение.
2. Ввести корректные учетные данные (логин и пароль).
3. Нажать кнопку «Авторизоваться».

Ожидаемый результат: Пользователь успешно авторизован и перенаправлен в окно основное окно приложения. В окне отображаются элементы интерфейса, соответствующие успешной авторизации в зависимости от аккаунта (окно обычного пользователя, окно администратора).

Альтернативный результат: У пользователя возникают проблемы с авторизацией: появляется окно с ошибкой подключения к базе данных.

При пустых полях выводится сообщение «Пожалуйста, заполните все поля», а при неправильных данных «Неверный логин или пароль».

Варианты проверки:

1. Ввод некорректного логина или пароля.
2. Попытка входа без заполнения полей.
3. Попытка входа с выключенной базой данных.

Тест: Регистрация

Цель теста: проверить корректность процесса регистрации в системе.

Шаги теста:

1. Открыть приложение.
2. Нажать кнопку «Регистрация».
3. Ввести корректные данные (логин, номер телефона, пароль).

Ожидаемый результат: Пользователь успешно зарегистрирован.

Альтернативный результат: У пользователя возникают проблемы с регистрацией: появляется окно с ошибкой подключения к базе данных.

При некорректных данных выводятся сообщения: «Пользователь с таким логином существует», «Номер телефона занят», «Неправильный формат номера телефона. Пример номера +1 (123) 456-78-90», «Пароль должен содержать хотя-бы один спецсимвол, символ верхнего или нижнего регистра, длинной от 8 до 32 символов», в зависимости от полученной ошибки.

При пустых полях выводится сообщение «Заполнены не все поля».

Варианты проверки:

1. Ввод некорректного логина, пароля или номера телефона.
2. Попытка регистрации без заполнения полей.
3. Попытка регистрации с выключенной базой данных.
4. Попытка ввода логина и номера телефона, которые уже есть в базе данных.

Тест: Добавление автомобиля

Цель теста: проверить корректность процесса добавления автомобилей в системе.

Шаги теста:

1. Войти в приложение как администратор.
2. Нажать на кнопку «Добавить новую машину»
3. Ввести название марки, модели, государственный номер автомобиля, цену, широту и долготу.

Ожидаемый результат: Автомобиль успешно добавлен.

Альтернативный результат: У пользователя возникают проблемы с добавлением автомобиля в систему: всплывает окно с ошибкой подключения к базе данных.

Если в базе данных уже существует такой государственный номер автомобиля, то всплывает окно «Данный государственный номер занят».

Если в поле «Государственный номер» данные введены в некорректном формате, то выводится сообщение «Неверный формат госномера. Пожалуйста, введите верный госномер».

Если в поля «Широта» или «Долгота» данные введены в некорректном формате, то выводится сообщение «Введите корректные координаты!».

Если в поле «Цена» данные введены в некорректном формате, то выводится сообщение «Введите корректную стоимость!».

При пустых полях выводится сообщение «Заполнены не все поля».

Варианты проверки:

1. Ввод пустых полей марки, модели, государственного номера, цены, широты, долготы.
2. Попытка добавления автомобиля без заполнения полей.
3. Ввод некорректных данных.
4. Попытка добавления автомобиля с выключенной базой данных.
5. Попытка ввода государственного номера, который уже есть в базе данных.

Тест: Редактирование пользователей

Цель теста: Проверить корректность процесса редактирования пользователей в системе.

Шаги теста:

1. Войти в приложение как администратор.
2. Выбор редактируемого пользователя.
3. Изменение полей «Имя», «Фамилия», «Пароль», «Уровень доступа», «Уровень подписки».

Ожидаемый результат:

Успешное сохранение обновлённых данных.

Альтернативный результат: У пользователя возникают проблемы с добавлением автомобиля в систему: всплывает окно с ошибкой подключения к базе данных.

Если поле «Пароль» пустое, то выводится сообщение «Поле пароль не должно быть пустым».

Если поле «Пароль» заполнено некорректно, то выводится сообщение «Пароль должен содержать хотя-бы один спецсимвол, символ верхнего или нижнего регистра, длинной от 8 до 32 символов».

Варианты проверки:

1. Некорректное заполнение поля «Пароль».
2. Не заполнение поля «Пароль».
3. Попытка сохранения данных с выключенной базой данных.

Тест: Редактирование профиля.

Цель теста: проверить корректность процесса редактирования профиля.

Шаги теста:

1. Войти в приложение как пользователь.
2. Нажать кнопку «Меню» в правом верхнем углу.
3. Нажать кнопку «Личный кабинет».
4. Нажать кнопку «Заполнить данные».
5. Ввести корректные данные «Имя», «Фамилия», «Электронная почта», «Водительское удостоверение».
6. Нажать «Сохранить».

Ожидаемый результат: Пользователь успешно отредактировал профиль.

Альтернативный результат: У пользователя возникают проблемы с редактированием профиля: всплывает окно с ошибкой подключения к базе данных.

Если пользователь вводит «Водительское удостоверение» некорректно, всплывает окно «Проверьте правильность ввода водительского удостоверения. Формат должен быть: 0000 000000».

Если пользователь оставляет поле «Водительское удостоверение» пустым, то всплывает окно «Введите номер водительского удостоверения».

Варианты проверки:

1. Попытка сохранения данных с выключенной базой данных.
2. Попытка сохранения данных с некорректными данными в поле «Водительское удостоверение».
3. Попытка сохранения данных с пустым полем «Водительское удостоверение».

4.2 Руководство пользователя

Разработанный программный продукт предназначен для формализации деятельности каршеринга (учет автомобилей, пользователей, просмотр свободных автомобилей, просмотр тарифных планов платных подписок, редактирование профиля). Позволяет выполнять ввод данных, поиск данных, удаление данных.

Программное обеспечение:

* ОС Windows версии 7 / 8 / 10 / 11
* NET.Framework 3.5 или выше.

Требования к аппаратуре:

* Компьютер совместимый с операционной системой Windows.

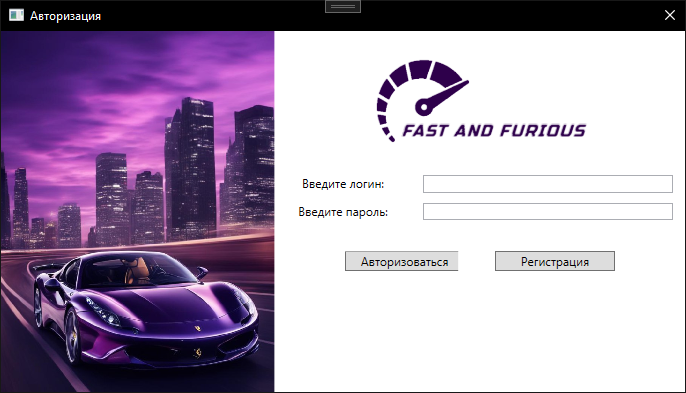
Других требований к системе не имеется.

Выполнение программы.

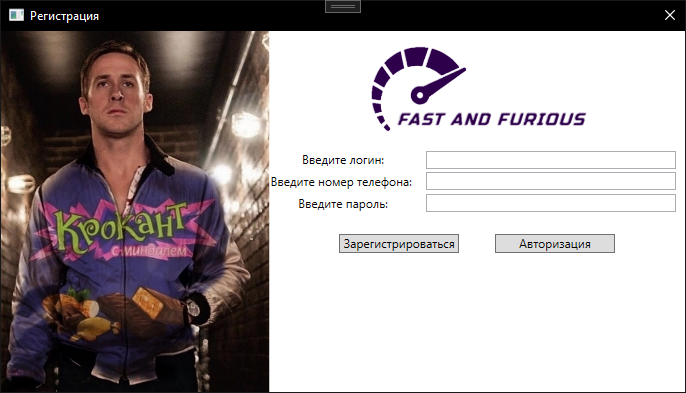
Для запуска программы необходимо запустить файл CarSharing

При запуске программа отображает окно авторизации каршеринга «Fast & Furious», имеется возможность выполнять следующие действия (Рисунок 4.2.1):

* Авторизоваться.
* Регистрация.
* Закрытие программы (выход).
* Свернуть.

  
Рисунок 4.2.1 – Окно «Авторизация».

При нажатии на кнопку «Регистрация» открывается окно «Регистрация» (Рисунок 4.2.2), при нажатии на кнопку «Авторизация» происходит авторизация пользователя, если поля «Логин» и «Пароль» были заполнены. Если пользователь является администратором, то открывается окно выбора режима работы с приложением «Выбор окна» (Рисунок 4.2.4)

  
Рисунок 4.2.2 – Окно регистрации.

При нажатии на кнопку «Зарегистрироваться» всплывает окно «Пользователь успешно зарегистрирован!» (Рисунок 4.2.3), после чего открывается окно авторизации (Рисунок 4.2.1).

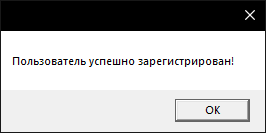


Рисунок 4.2.3 – Окно «Пользователь успешно зарегистрирован!».

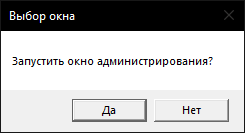


Рисунок 4.2.4 – Окно выбора режима работы с приложением «Выбор окна».

Если пользователь нажимает кнопку «Да», то открывается окно «Fast and Furious: администрирование» (Рисунок 4.2.5). Если пользователь нажимает кнопку «Нет», то открывается окно «Fast and Furious» (Рисунок 4.2.х).

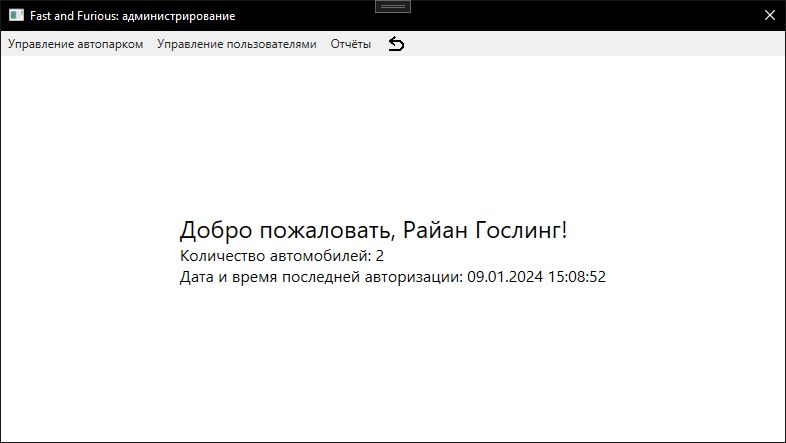
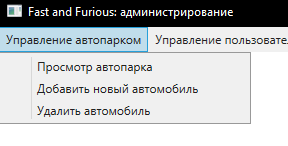
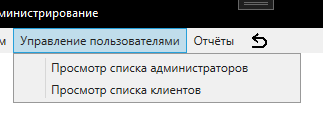


Рисунок 4.2.5 – Окно «Fast and Furious: администрирование».

При открытии окна «Fast and Furious: администрирование» (Рисунок 4.2.5) видно приветствие авторизованного пользователя, текущее количество автомобилей, а также время последней авторизации. Также в панели управления присутствует вкладки «Управление автопарком», «Управление пользователями», «Отчёты», а также кнопка для возвращения на стартовый экран.

При наведении курсора на «Управление автопарком» открывается список доступных функций (Рисунок 4.2.6). При наведении курсора на «Управление пользователями» открывается список доступных функций (Рисунок 4.2.7). При наведении курсора на «Отчёты» открывается список доступных функций (Рисунок 4.2.8).

  
Рисунок 4.2.6 – Список функций «Управление автопарком».

  
Рисунок 4.2.7 – Список функций «Управление пользователями».

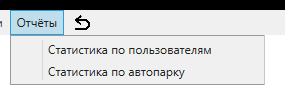
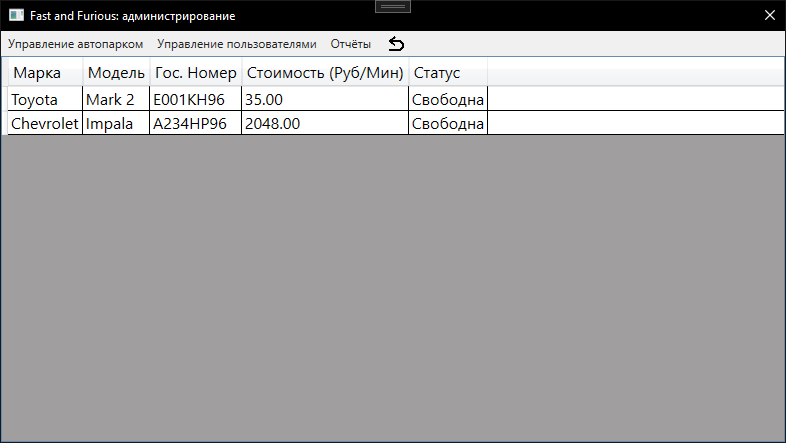
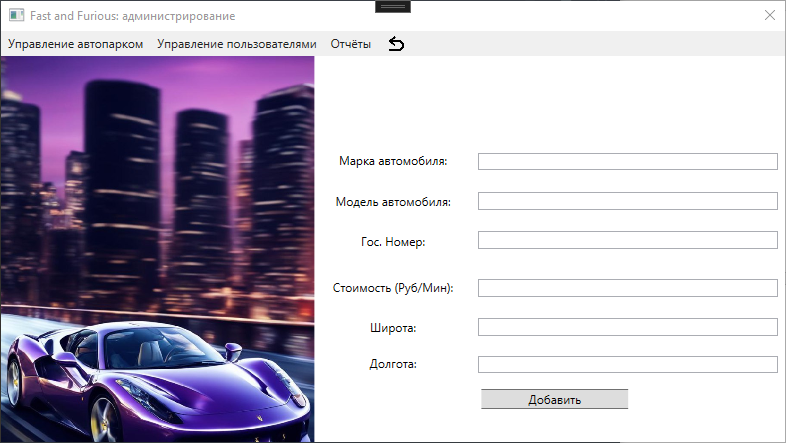


Рисунок 4.2.8 – Список функций «Отчёты».

При нажатии кнопки «Просмотр автопарка» открывается страница «Просмотр автопарка» (Рисунок 4.2.9), на которой можно посмотреть информацию об автомобилях, которые добавлены в систему.

  
Рисунок 4.2.9 – Страница «Просмотр автопарка».

При нажатии кнопки «Добавить новый автомобиль» открывается окно «Добавить автомобиль» (Рисунок 4.2.10), где можно добавить новый автомобиль в систему и указать его координаты и стоимость.

  
Рисунок 4.2.10 – Страница «Добавление автомобиля».

При нажатии кнопки «Удалить автомобиль» открывается страница «Удаление автомобилей» (Рисунок 4.2.11), где можно удалить любой автомобиль из системы.

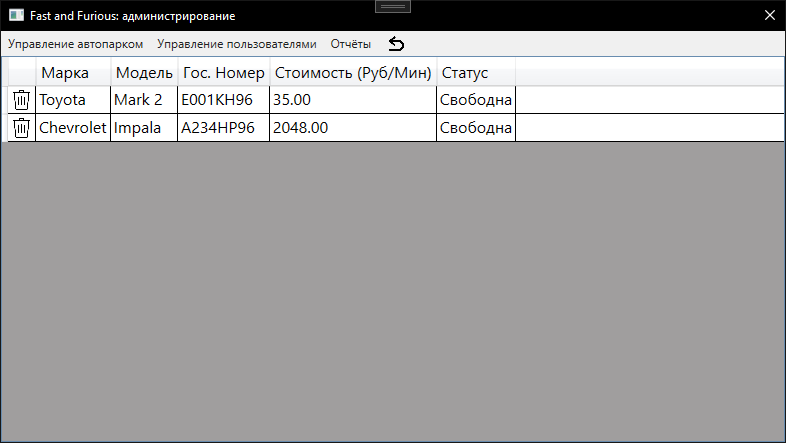


Рисунок 4.2.11 – Страница «Удаление автомобилей».

При нажатии на кнопку «Просмотр списка администраторов» открывается страница «Список администраторов» (Рисунок 4.2.12), где можно отредактировать список администраторов или при нажатии кнопки «Редактировать» откроется окно «Редактирование» (Рисунок 4.2.13), где можно отредактировать данные в аккаунте администратора.

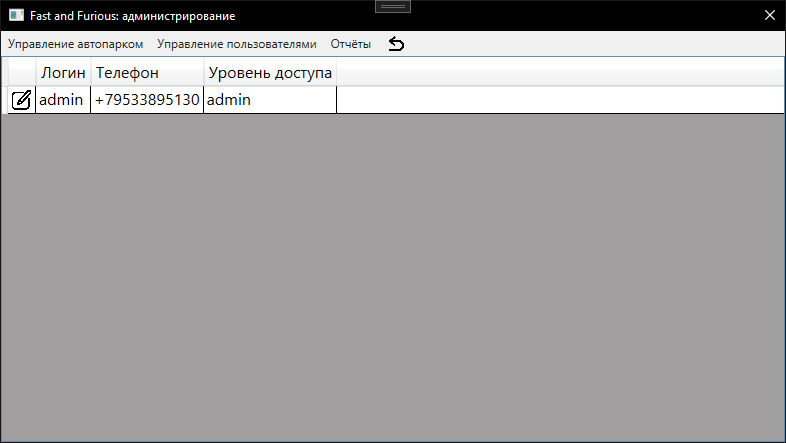


Рисунок 4.2.12 – Страница «Список администраторов».

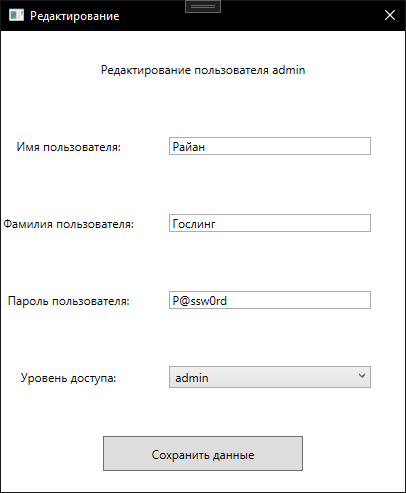


Рисунок 4.2.13 – Окно «Редактирование».

При нажатии кнопки «Просмотр списка клиентов» открывается страница «Список клиентов» (Рисунок 4.2.14), где можно удалить или отредактировать профиль клиента. При нажатии кнопки «Редактировать» открывается окно «Редактирование» (Рисунок 4.2.15).

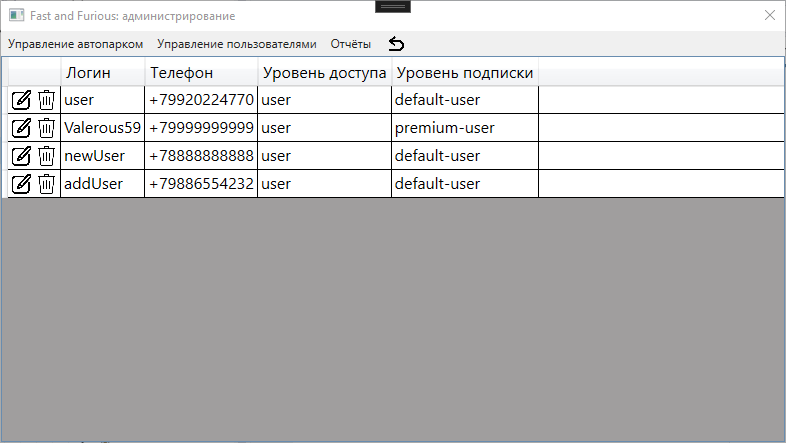


Рисунок 4.2.14 – Страница «Список клиентов».

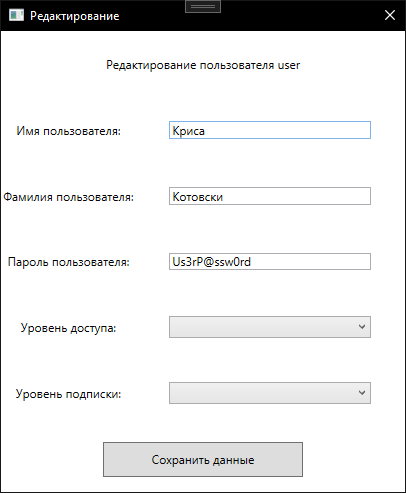


Рисунок 4.2.15 – Окно «Редактирование».

При нажатии кнопки «Статистика по пользователям» открывается страница «Статистика: пользователи» (Рисунок 4.2.16).

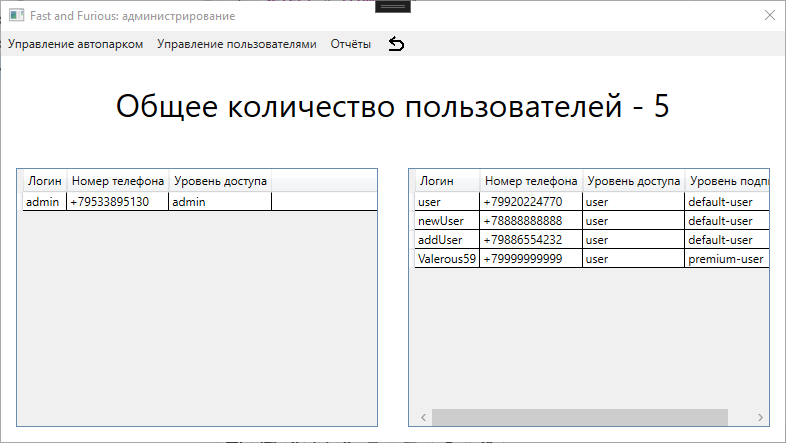


Рисунок 4.2.16 – Страница «Статистика: пользователи».

При нажатии кнопки «Статистика по автопарку» открывается страница «Статистика: автопарк» (Рисунок 4.2.17).

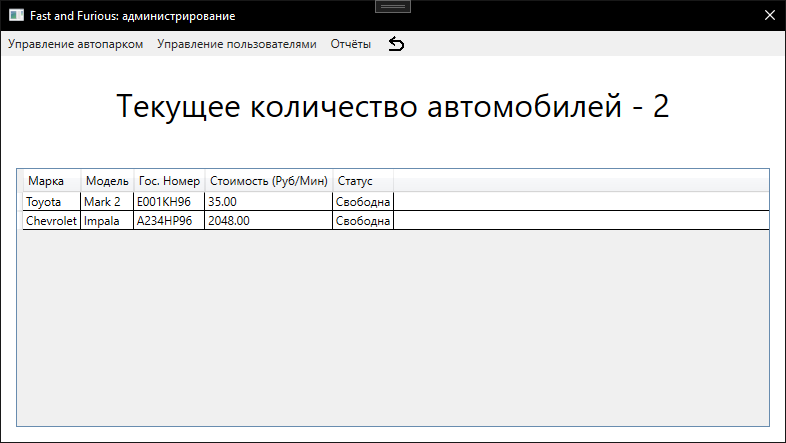


Рисунок 4.2.17 – Страница «Статистика: автопарк».

Если пользователь не является администратором или выбрал в окне "Выбор окна" (Рисунок 4.2.4) режим работы "Нет", открывается окно "Fast and Furious" (Рисунок 4.2.18), при запуске которого автоматически отображается карта города с доступными автомобилями, обозначенными маркерами.

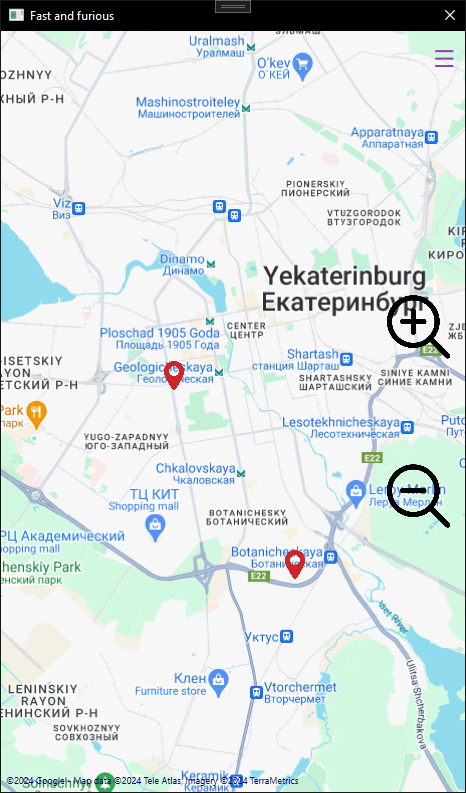
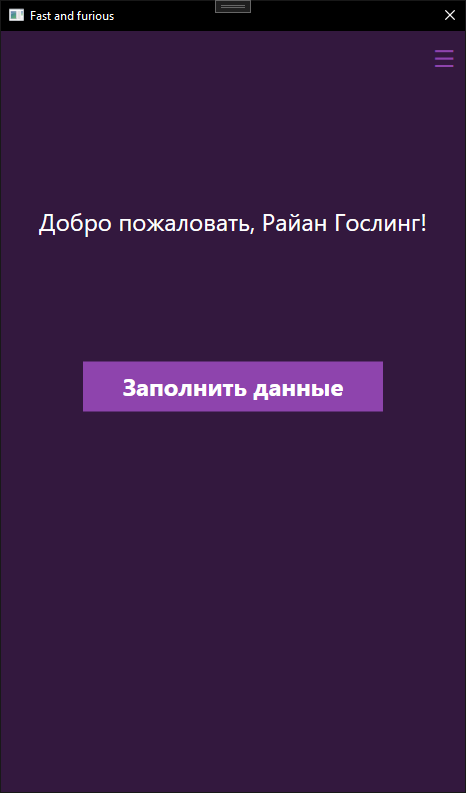


Рисунок 4.2.18 – Окно «Fast and Furious». Страница «Карта»

Если пользователь нажимает кнопку в правом верхнем углу страницы, то открывается меню (Рисунок 4.2.19). При нажатии кнопки «Личный кабинет», то откроется страница «Личный кабинет» (Рисунок 4.2.20), где пользователь может отредактировать данные своего профиля (Рисунок 4.2.21). Если нажать кнопку «Подписка», то откроется страница с тарифными планами платной подписки (Рисунок 4.2.22), где пользователь может выбрать подходящий ему тарифный план. При нажатии кнопки «Карта», пользователя вернёт на начальную страницу (Рисунок 4.2.18).



Рисунок 4.2.19 – Меню.

  
Рисунок 4.2.20 – Страница «Личный кабинет».

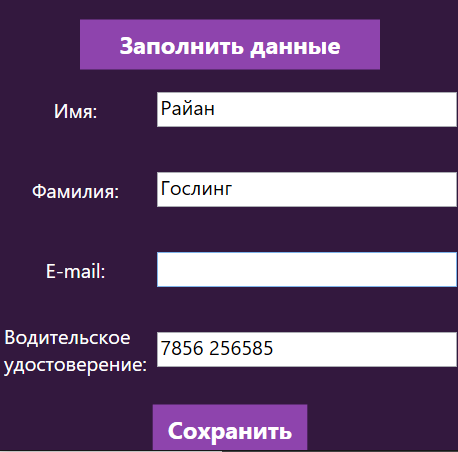


Рисунок 4.2.21 – Редактирование профиля.

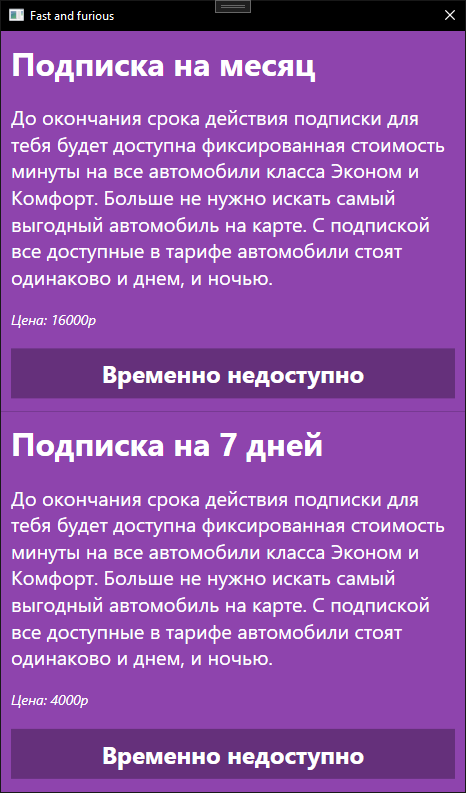


Рисунок 4.2.22 – Страница «Тарифы платной подписки»

При нажатии на маркер, обозначающий автомобиль на карте, отображается информация о данном транспортном средстве (Рисунок 4.2.23).

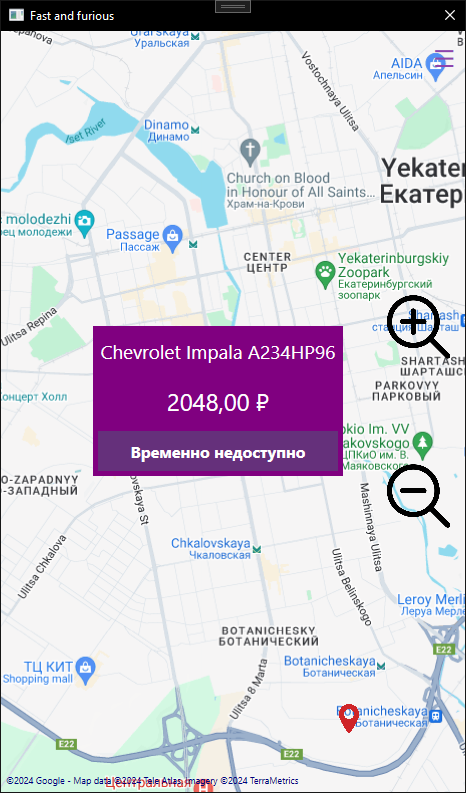


Рисунок 4.2.23 – Информация об автомобиле.

Заключение

В ходе выполнения курсового проекта по разработке автоматизированной информационной системы для каршеринга "Fast & Furious" были успешно решены поставленные задачи. Анализ предметной области и постановка задачи позволили четко определить требования к системе. Анализ информационных потребностей помог выявить основные функции и роли пользователей.

Выбор технических и программных средств, проектирование диаграмм и сценариев использования, а также разработка логической и физической структуры базы данных сделали возможным создание эффективной и устойчивой информационной системы. Разработка базы данных и приложения для АИС каршеринга "Fast & Furious" прошли успешно, учитывая необходимость хранения и обработки информации об автомобилях и пользователях.

Определены роли в системе: администратор и клиент, каждая из которых обладает своим функционалом. Реализован функционал редактирования автопарка, управления пользователями, создания новых учетных записей, формирования отчетов, просмотра доступных автомобилей на карте и вариантов оформления платной подписки.

Осуществленный тестированный функционал позволяет предположить эффективность и надежность работы информационной системы. Создано руководство пользователя, которое поможет пользователям легко освоиться с системой.

Таким образом, курсовой проект по разработке АИС каршеринга "Fast & Furious" был успешно завершен, достигнуты поставленные цели и решены задачи, предусмотренные в постановке.

Список использованных источников

1. Николаев, М.И. Метрология, стандартизация, сертификация и управление качеством [Электронный ресурс] : учебное пособие / М.И. Николаев. — Электрон, дан. — Москва : , 2016. — 115 с. — Режим доступа: https://e.lanbook.com/book/100255. — Загл. с экрана.
2. Мякишев, Д.В. Принципы и методы создания надежного программного обеспечения АСУТП [Электронный ресурс] : учебно-методическое пособие / Д.В. Мякишев. — Электрон, дан. — Вологда : «Инфра-Инженерия», 2018. — 144 с. — Режим доступа: https://e.lanbook.com/book/108700. — Загл. с экрана.
3. Лямин, А.В. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие / А.В. Лямин, Е.Н. Череповская. — Электрон, дан. — Санкт-Петербург : НИУ ИТМО, 2017. — 143 с. — Режим доступа: https://e.lanbook.com/book/110457. — Загл. с экрана.
4. Виноградов, В.И. Постреляционные модели данных и языки запросов [Электронный ресурс]: учебное пособие / В.И. Виноградов, М.В. Виноградова. — Электрон, дан. — Москва: МГТУ им. Н.Э. Баумана, 2017. — 96 с. — Режим доступа: https://e.lanbook.com/book/103533. — Загл. с экрана.
5. GMap.NET. Репозиторий на GitHub [Электронный ресурс] : <https://github.com/judero01col/GMap.NET>. — Загл. с экрана.
6. Работа с картами. GMap C# [Электронный ресурс] : <https://habr.com/ru/articles/585734/>. — Загл. с экрана.