МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения 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 Программное обеспечение информационных технологий (программирование интернет-приложений)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

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

Тема «Менеджер автомобилей »

Исполнитель

студент 2 курса группы 5 Грунковский Роман Иванович

(Ф.И.О.)

Руководитель работы асс. Панченко О.Л.

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

Курсовой проект защищен с оценкой

Председатель Пацей Н.В.

(подпись)

Минск 2019

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc8724622)

[1. Аналитический обзор литературы 4](#_Toc8724623)

[1.1. Аналитический обзор источников 4](#_Toc8724624)

[1.2. Обзор аналогов 4](#_Toc8724625)

[2. Анализ требований к программному средству и разработка функциональных требований 8](#_Toc8724626)

[2.1. Спецификация функциональных требований 8](#_Toc8724627)

[2.1.1. Требования к составу и параметрам технических и программных средств 8](#_Toc8724628)

[2.1.2. Требования к организации входных данных 8](#_Toc8724629)

[2.1.3. Требования к организации выходных данных 8](#_Toc8724630)

[2.1.4. Требования к надежности 8](#_Toc8724631)

[2.2. Описание функциональности программного средства 9](#_Toc8724632)

[3. Проектирование программного средства 10](#_Toc8724633)

[3.1. Проектирование базы данных 10](#_Toc8724634)

[3.2. Проектирование архитектуры проекта 11](#_Toc8724635)

[4. Создание программного средства 15](#_Toc8724636)

[4.1. Класс Garage 15](#_Toc8724637)

[4.2. Класс Cars 15](#_Toc8724638)

[4.3. Класс Image 15](#_Toc8724639)

[4.4. Классы Add, Balance, Count 15](#_Toc8724640)

[4.5. Класс Deals 15](#_Toc8724641)

[4.6. Класс Rcommands 16](#_Toc8724642)

[4.7. Класс Sell 16](#_Toc8724643)

[4.8. Класс User 16](#_Toc8724644)

[4.9. Валидация данных 16](#_Toc8724645)

[5. Тестирование 17](#_Toc8724646)

[6. Руководство 20](#_Toc8724647)

[ЗАКЛЮЧЕНИЕ 26](#_Toc8724648)

[СПИСОК ЛИТЕРАТУРЫ 27](#_Toc8724649)

[ПРИЛОЖЕНИЕ А 28](#_Toc8724650)

[ПРИЛОЖЕНИЕ Б 29](#_Toc8724651)

# Введение

Менеджер автомобилей – это программа на ПК, позволяющая управлять своими автомобилями, а именно покупать новый, продавать старый. В век информационных технологий уже становится не модно ходить по автосалону. Ведь это затраты на поездку до автосалона, время и прочие факторы, которые усложняют данный процесс.

Иногда данная идея используется в онлайн играх, чтобы игрокам было проще манипулировать сделками, своим автомобилем.

Цель моего курсового проекта разработать менеджер автомобилей для удобства пользователя. Чтобы он мог купить интересующий автомобиль, продать какому-то конкретному пользователю, а данный пользователь в свою очередь приобрести данный автомобиль, так как это будет явно дешевле, чем в автосалоне, или отказаться от сделки.

При использовании данного мессенджера, каждый пользователь компьютера сможет обмениваться сообщениями с другими пользователями в режиме реального времени. Язык разработки проекта – C#. При выполнении курсового проекта будут использованы принципы и приемы ООП. Также будут использоваться технологии Windows Presentation Foundation(WPF).

1. **Аналитический обзор литературы**
   1. **Аналитический обзор источников**

В ходе подготовки пояснительной записки была изучена специальная техническая, учебно-методическая и справочная литература, статьи и материалы, опубликованные в сети интернет.

Общие теоретические и практические знания были получены из книги «Pro WPF 4.5 in C#: Windows Presentation». В книге приведены примеры работы с WPF, введены основные понятия для работы с WPF, описано взаимодействие окон [1].

Принцип создания окон для регистрации и авторизации был взят из статьи «WPF – система авторизации и регистрации». В статье были рассмотрены алгоритм работы системы авторизации и регистрации и пример создания окон регистрации и авторизации [2].

Алгоритм работы окна авторизации:

1. выполняется проверка заполняемости всех полей;
2. выполняется проверка существования записи в базе данных;
3. если запись существует, пользователь авторизуется.

Алгоритм работы окна регистрации:

1. выполняется проверка заполняемости всех полей;
2. выполняется проверка существования записи с таким же логином в базе данных;
3. если записи не существует, добавляется новая запись.

Принцип работы с SQL были получены из статьи «Подключение к базе данных». В статье было рассмотрено подключение необходимых библиотек, работа с SQL [3].

Концепция создания отчета была получена из статьи «Report-Driven Design». В статье была рассмотрена общая информация о создании отчёта и пример создания отчёта [4].

* 1. **Обзор аналогов**

Для создания принципиально нового решения в виде программного продукта для решения вопросов автоматизации деятельности интернет магазина необходимо ознакомиться с существующими аналогами в данной сфере. Анализ достоинств и недостатков этих аналогов позволит сформировать требования к проектируемому программному средству, учитывающие опыт существующих разработок и внести в них улучшения или изменения.

В качестве исследуемых аналогов были выбраны программные продукты, связанные с деятельностью приемных комиссий либо сферой образования, как наиболее близкие по области применения к разрабатываемому программному средству. Помимо этого были проанализированы продукты, связанные с электронной подачей заявлений на оказание различных видов услуг. Источником информации послужили электронные базы в сети Интернет.

В результате поиска были обнаружены следующие ресурсы:

* автосервис. Решение предназначено для автоматизации деятельности в небольших автосервисах, автомойках, станциях технического обслуживания автомобилей. 1С:Автосервис. Клиентская лицензия на 1 рабочее место — программный продукт от компании «1С», который предназначен для автоматизации бизнеса. Изначально «1С:Предприятие» было предназначено для автоматизации управленческого и бухгалтерского учётов, но в настоящее время этот продукт находит своё применение в различных сферах деятельности, далёких от бухгалтерских задач. 1С:Автосервис. Клиентская лицензия на 1 рабочее место позволит автоматизировать процессы в бизнесе и упростить повседневные задачи [5].

Программные продукты этой серии обладают следующими функциями:

1. функции CRM-системы: Ведение базы клиентов. Фиксация всех контактов с клиентами: входящие и исходящие звонки, письма, встречи и прочее. Отправка e-mail и SMS сообщений непосредственно из программы;
2. автосервис:
   1. использование специфичных для автосервиса справочников: модели автомобилей, нормо-часы, цеха, виды ремонта;
   2. интеграция с каталогами работ и норм времени (загрузка данных);
   3. предварительная запись на ремонт и планирование загрузки автосервиса.
   4. учет выполнения ремонта документом "Заказ-наряд", включающий: прием автомобиля, передачу автомобиля в цех, выполнение дефектовки, подбор запасных частей, распределение исполнителей работ, выдачу готового автомобиля клиенту.
   5. контроль деятельности автосервиса из любой точки земного шара через Интернет, в том числе и с помощью планшетных компьютеров или коммуникаторов;
   6. снабжение и закупки: различные схемы пополнения товарных запасов автосервиса – покупка, приобретение подотчетным лицом, прием на реализацию от комиссионера. Учет дополнительных расходов на приобретение;
   7. полный цикл операций складского учета при торговле товарами в автосервисе – от поступления до выбытия. Раздельный учет собственных запасов; запасов, принятых и переданных на реализацию, ответственное хранение. Реализован механизм "ордерного склада";
   8. торговля: автоматизированы хозяйственные операции по оптовым продажам и оптовому приобретению товаров и услуг, розничная торговля – рабочее место кассира (РМК) и работа с торговым оборудованием;
   9. деньги: учет движения денежных средств на банковских счетах и в кассах, взаимодействие с системами "Клиент-банк", контроль остатков и исполнения финансовых обязательств, платежный календарь, бюджет движения денежных средств;
   10. зарплата и персонал: планирование и учет рабочего времени, кадровый учет (оформление приема на работу, кадровые перемещения, увольнение персонала), управленческий расчет заработной платы, анализ эффективности работы персонала;
   11. учета имущества (внеоборотных активов) – основных средств и нематериальных активов автосервиса: принятие к учету, изменение параметров, расчет амортизации, продажа и списание;
   12. управленческий (финансовый) учет: учет доходов, расходов, прибыли и убытков, движения денежных средств на расчетных счетах, финансовое планирование (бюджетирование), управленческий баланс и много другое;
   13. специальные комплекты отчетов "Монитор руководителя" и "Мониторы основных показателей" позволяют быстро и удобно получать самую важную информацию по основным участкам деятельности предприятия, своевременно выявлять точки роста или негативную динамику;
   14. разграничение прав доступа к функциям программы, контроль действий персонала;
   15. контроль деятельности автосервиса из любой точки земного шара через Интернет, в том числе и с помощью планшетных компьютеров или коммуникаторов;
   16. интеграция с "1С:Бухгалтерия 8": ведение бухгалтерского и налогового учета с использованием уже накопленных данных о работе автосервиса, что позволяет исключить повторный ввод информации;
   17. широкий спектр подключаемого торгового оборудования.

* "Альфа-авто". Современный программный продукт «Альфа-Авто» позволяет объединить в рамках единой базы данных все подразделения автомобильного холдинга (автосалоны, автосервисы, склады запчастей, бухгалтерии, финансовые службы, службы маркетинга) и управлять основными бизнес-процессами компании [6].

Типовая конфигурация позволяет:

* оформлять заказы и продажи автомобилей;
* выполнять предпродажную подготовку автомобилей;
* оказывать услуги по ремонту и обслуживанию автомобилей;
* вести оптовую и розничную торговлю запасными частями;
* организовать работу склада;
* учитывать оплаты и отслеживать состояние взаиморасчетов с покупателями и поставщиками;
* «1С:Розница». Больше не надо тратить время на подбор автозапчастей. Решение от «1С» сделает это за вас. Встроенная система поиска автоматически найдет нужный товар по заданным вами параметрам. А удобный интерфейс облегчит начало работы с программой. 1С:Розница. Магазин автозапчастей — программный продукт от компании «1С», который предназначен для автоматизации бизнеса. Изначально «1С:Предприятие» было предназначено для автоматизации управленческого и бухгалтерского учётов, но в настоящее время этот продукт находит своё применение в различных сферах деятельности, далёких от бухгалтерских задач. 1С:Розница. Магазин автозапчастей позволит автоматизировать процессы в бизнесе и упростить повседневные задачи [7].

Функционал:

* оформление прихода товаров от контрагента на склады предприятия, в том числе в двухфазовом (ордерном) режиме;
* оформление реализации товаров и услуг контрагенту, в том числе в двухфазовом (ордерном) режиме;
* оформление перемещения товаров между магазинами, внутренними складами магазинов, магазинами и складами предприятия, в том числе в двухфазовом (ордерном) режиме;
* торговля комплектами товаров, созданными как в момент продажи товара, так и с подготовкой комплекта (операция "комплектация");
* оформление возвратов товаров от покупателей (реализованы механизмы автоматического создания необходимых документов при возврате "Не День в День" в режиме РМК);
* оформление документов инвентаризации товаров ("Инвентаризация товаров", "Списание товаров", "Оприходование товаров");
* оформление приходных и расходных кассовых ордеров непосредственно в магазинах;
* оформление документов перемещения денежных средств между магазинами, внутренними кассами магазинов, магазинами и кассами предприятия;
* оформление чеков продажи, и по окончании смены сводного отчета по контрольно-кассовой машине, с учетом возвращенных товаров в смену;
* работа с эквайринговыми системами, учет оплат товаров по платежным картам, учет договоров эквайринга и условия возвращения или невозвращения торговой уступки эквайрером при возврате товаров; оплата товаров кредитами;
* возможность использования процентных скидок по дисконтным картам (накопительные скидки), скидки с разделением по магазинам, скидки контрагентам, скидки на сумму чека, скидки по времени действия, по количеству товара, по виду оплаты;
* поддержка торгового оборудования: фискальные регистраторы, терминалы сбора данных, сканеры штрихкодов, дисплеи покупателя, платежные терминалы, ридеры магнитных карт.

1. **Анализ требований к программному средству и разработка функциональных требований**
   1. **Спецификация функциональных требований**
      1. **Требования к составу и параметрам технических и программных средств**

Программное средство может функционировать на ЭВМ со следующими минимальными характеристиками:

1. процессор Intel® Core™ i3 с тактовой частотой 2.4 ГГц и лучше;
2. оперативная память 256 МБ и более;
3. операционная система: Windows 10;
4. сетевая карта Ethernet 10/100 Мбит.
   * 1. **Требования к организации входных данных**

Входные данные для программного средства должны быть представлены в виде вводимого пользователем с клавиатуры текста: логин, пароль. После аутентификации пользователю предоставляется возможность работы с системой.

Данные, вводимые пользователем, должны проверяться на корректность в процессе аутентификации.

* + 1. **Требования к организации выходных данных**

В качестве выходных данных будут выступать файлы, отображающие данные и результаты их обработки в виде отчёта.

* + 1. **Требования к надежности**

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

1. надёжное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:
2. организацией бесперебойного питания технических средств,   
   выполнением требований «ГОСТ 31078-2002. Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство»;
3. необходимым уровнем квалификации сотрудников профильных подразделений.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

* 1. **Описание функциональности программного средства**

На основании анализа исходных данных для проектируемого программного средства можем выделить, что основной целью является создание качественного программного продукта, позволяющего решить существующие проблемы таких как:

* автоматизация процесса покупки автомобиля;
* автоматизация продажи автомобиля.

Диаграмма классов представлена в Приложении А.

Класс Balance содержит метод для пополнения баланса пользователя.

Класс Cats содержит методы для работы автосалона.

Класс Count содержит метод для изменения количества доступных автомобилей в автосалоне.

Класс Deals содержит методы, которые необходимы для проведения сделок между пользователями.

Класс Add содержит метод для добавления автомобиля в автосалон администратором.

Класс Garage содержит методы для домашней странички пользователя.

Класс Images содержит метод для вывода фотографий автомобилей.

Класс Login содержит методы обеспечения успешного входа, а Register методы для успешной регистрации и валидацию вводимых данных.

Класс Sell содержит метод для продажи автомобиля пользователю.

Класс User содержит метод для шифрования пароля и поля для работы с пользователем.

Класс RCommands содержит реализацию вызова справки сочетанием клавиш ALT+F1.

Класс MainWindow содержит реализацию перехода между страницами и метод для работы с базой данных.

# Проектирование программного средства

# 3.1. Проектирование базы данных

В данной курсовом проекте была спроектирована база данных, которая получила название KP. Для управления базой данных используется MS SQL Server 2012. Логическая схема данных изображена на рисунке 3.1.



Рисунок 3.1 – Логическая схема данных

Всего в базе данных 4 таблицы. В таблице users хранятся все пользователи, зарегистрированные в приложении, Deals – все сделки, Cars ­– список автомобилей в автосалоне и Garage – список автомобилей у конкретного пользователя.

Таблица users состоит из столбцов:

* id;
* login;
* password.

Столбец login содержит имя пользователя, password – захешированный пароль алгоритмом MD5.

Таблица Deals состоит из столбцов:

* id;
* seller\_id;
* customer\_id;
* car\_id;
* cost;
* status.

Столбец seller\_id содержит id пользователя, который предложил продажу. Столбец customer\_id – id покупателя, car\_id – id машины, cost – стоимость предложения, status – статус сделки (1 – сделка принята, 0 – сделка отклонена, null – решение не принято).

Таблица Garage:

* id;
* car\_id;
* user\_id.

В столбце car\_id хранится id машины, который владеет пользователь с id из столбца user\_id.

Таблица Cars включает столбцы:

* id;
* name;
* cost;
* description;
* count;
* img.

Столбец name – название автомобиля, cost – стоимость, description – описание, count – количество, img – изображение в двоичном формате.

# 3.2. Проектирование архитектуры проекта

На рисунке 3.2 изображена диаграмма вариантов использования данного приложения.

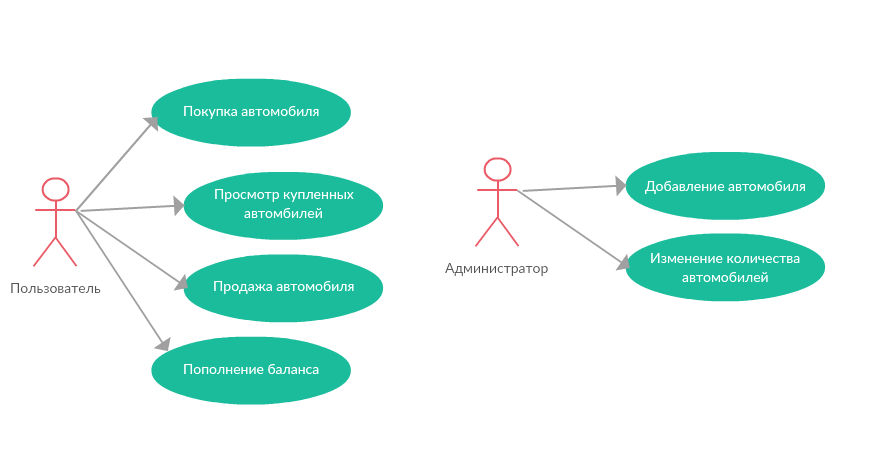


Рисунок 3.2 – Диаграмма использования

На рисунке 3.3 была спроектирована диаграмма последовательности для авторизации и регистрации. На диаграмме последовательности отображаются только те объекты, которые непосредственно принимают участие во взаимодействии.

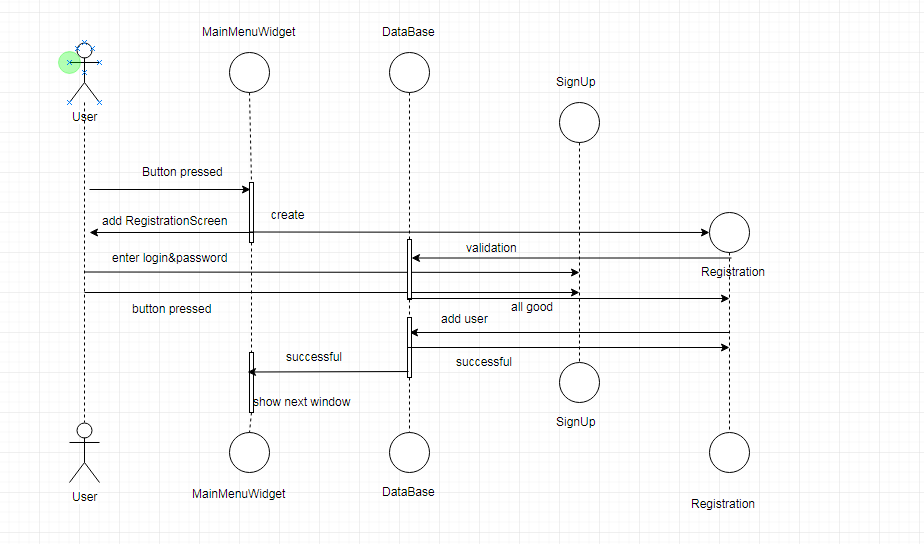


Рисунок 3.3 – Диаграмма последовательности для авторизации и регистрации.

На рисунке 3.4 изображена диаграмма компонентов нашего приложения и изображены связи между этими компонентами.

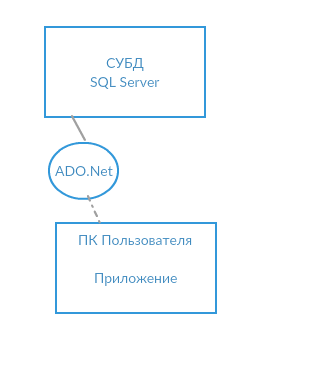


Рисунок 3.4 – Диаграмма компонентов

На рисунках 3.5 и 3.6 представлены блок-схемы для алгоритмов авторизации и регистрации пользователя.



Рисунок 3.4 – Схема алгоритма авторизации



Рисунок 3.5 – Схема алгоритма регистрации

В Приложении Б представлен программный код, который реализует алгоритм авторизации регистрации.

Переход между страницами реализован в классе MainWindow. А также метод, который используется в других классах для написания и выполнения запросов на языке Transact-SQL к базе данных KP. Для написания запросов и создания подключения была использована технология ADO.NET (ActiveX Data Object для .NET). Также для доступа только к базе данных KP был создан специальный пользователь с именем KP. Сделано это также для большей безопасности приложения. Программный код, реализующий переход между страницами представлен ниже:

public enum pages

{ login, register, cars, add, garage, balance, sell, deals, image, counts } public void OpenPage(pages pages)

{

switch (pages)

{

case pages.login:

frame.Navigate(new Login(this));

break;

case pages.register:

frame.Navigate(new Register(this));

break;

case pages.cars:

frame.Navigate(new Cars(this));

break;

case pages.add:

frame.Navigate(new Add(this));

break;

case pages.garage:

frame.Navigate(new Garage(this));

break;

case pages.balance:

frame.Navigate(new Balance(this));

break;

case pages.sell:

frame.Navigate(new Sell(this));

break;

case pages.deals:

frame.Navigate(new Deals(this));

break;

case pages.image:

frame.Navigate(new Images(this));

break;

case pages.counts:

frame.Navigate(new Count(this));

break;

default:

break;

}

}

# Создание программного средства

Для более детального изучения программного средства в данной главе остановимся поподробнее на некоторых классах. Весь графический интерфейс определён в .xaml файлах. Общие сведения о классах представлены в п. 2.2.

# Класс Garage

Данный класс отвечает за нашу домашнюю страницу. В нём реализованы методы, которые реализуют:

* отображение имени текущего пользователя;
* отображение баланса пользователя;
* отображение списка автомобиля пользователя;
* быструю продажу автомобиля;
* получение уведомлений о доступных сделках;
* переход на различные страницы.

# Класс Cars

Этот класс реализует методы, которые необходимы для работы автосалона и администратора:

* отображение доступных автомобилей;
* покупка автомобиля;
* добавление автомобиля в автосалон (для администратора);
* изменение количества автомобилей в автоcалоне;
* переход на домашнюю страницу.

# Класс Image

Данный класс отвечает за вывод изображения, которое загрузил администратор в базу данных, когда добавлял автомобиль в автосалон, и переход к предыдущей странице, в зависимости, откуда была выведена картинка.

# Классы Add, Balance, Count

Класс Add реализует методы, которые позволяют добавлять автомобиль в автосалон.

Класс Balance реализует методы, которые необходимы для корректного ввода пользователем информации при пополнении своего баланса.

Класс Count позволяет изменять количество доступных автомобилей администратору.

# Класс Deals

Данный класс отвечает за «вишенку» курсового проекта: сделки с пользователями. Здесь реализованы методы, которые позволяют приобрести или отказаться от приобретения автомобиля у некоторого пользователя. А также метод, который выбирает только те сделки, которые доступны только пользователю, а не чужие.

# Класс Rcommands

Этот статический класс нужен для вызова справки сочетанием клавиш ALT+F1, после нажатия которых появляется небольшая информация о данном программном средстве.

# Класс Sell

Класс, которому посвящен данный подраздел, отвечает за продажу автомобиля какому-то из доступных пользователей (кроме администратора).

# Класс User

Данный класс служит как вспомогательный. В нём присутствует метод, который хеширует входную строку (пароль) с помощью алгоритма хеширования MD5.

MD5 ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Message Digest 5) — 128-битный алгоритм [хеширования](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), разработанный профессором [Рональдом Л. Ривестом](https://ru.wikipedia.org/wiki/%D0%A0%D0%B8%D0%B2%D0%B5%D1%81%D1%82,_%D0%A0%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B4) из [Массачусетского технологического института](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%81%D0%B0%D1%87%D1%83%D1%81%D0%B5%D1%82%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82) (Massachusetts Institute of Technology, MIT) в [1991 году](https://ru.wikipedia.org/wiki/1991_%D0%B3%D0%BE%D0%B4). Предназначен для создания «отпечатков» или [дайджестов сообщения](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%B9%D0%B4%D0%B6%D0%B5%D1%81%D1%82_%D1%81%D0%BE%D0%BE%D0%B1%D1%89%D0%B5%D0%BD%D0%B8%D1%8F) произвольной длины и последующей проверки их подлинности. Широко применялся для проверки [целостности информации](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8) и хранения хешей паролей.

В курсовой работе данный алгоритм выбран по двум причинам: быстродействие и необходимая надежность. Был рассмотрен вариант хеширования с помощью алгоритма SHA-512. Ведь, как известно, чем больше бит, тем более надежнее будут зашифрованы данные, однако страдает быстродействие всего программного средства. А здесь достаточно будет и MD5.

# Валидация данных

Дабы избежать ошибок пользователя и падения всего приложения, были предусмотрены некоторые правила, которые пользователь не может обойти. А в случае ошибки, выдается окно с сообщением, которое информирует пользователя об ошибке. Например, в классе Login порядок проверки таков: проверка введён ли логин, введён ли пароль, после чего уже осуществляется запрос к базе данных, который соответствует введенным ранее данным. В случае успешного поиска, пользователь переходит на домашнюю страницу. В процессе ввода данных есть некоторые запрещенные символы: символы кавычек, слеши и т.д. В процессе регистрации при наведении на соответствующие поля пользователю выдается подсказка, которая напоминает ему, что разрешено использовать в том или ином поле. Аналогичные проверки сделаны и в других классах.

# Тестирование

При тестировании данного приложения были проведены следующие тесты на ситуации, которые могли бы привести к ошибке:

* тесты на валидацию авторизации и регистрации;
* тесты на отсутствие выбора автомобиля перед осуществлением операций.

В момент регистрации возможна ситуация, когда пользователь вводит уже существующий логин. Обработка данного исключения продемонстрирована на рисунке 5.1.

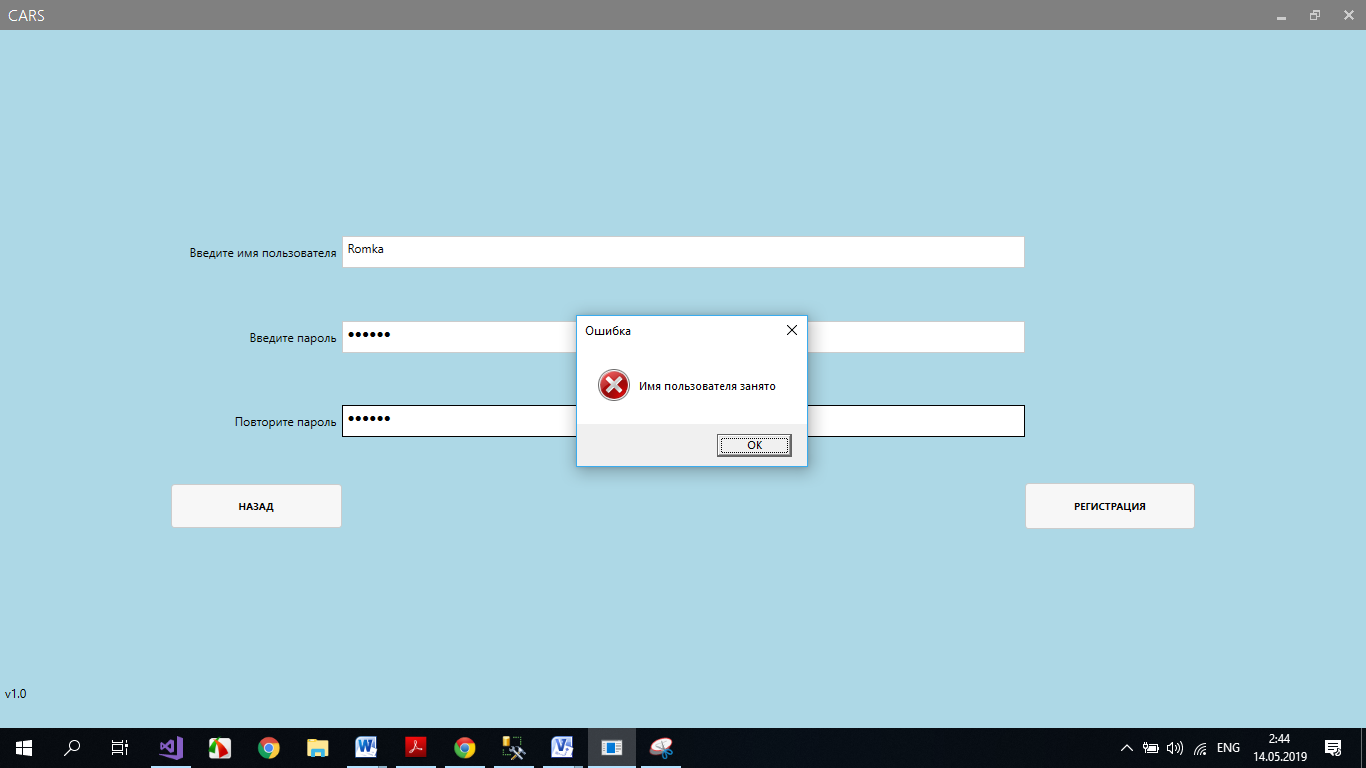


Рисунок 5.1 ­– Имя пользователя занято

Предусмотрено, что пользователь не ввел одно из полей, тогда программное средство напомнит ему об этом и не даст зарегистрироваться, пока пользователь не исправит все свои ошибки и не заполнит все поля. В случае ввода пользователем каких-то запрещенных символов ничего не произойдёт, так как при нажатии на клавишу вызывается функция проверки, которая представлена ниже:

private void TextBox\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

string inputSymbol = e.Text.ToString();

if (!Regex.Match(inputSymbol, @"[a-zA-zа-яА-Я0-9\_,.@$#!\*]").Success)

{

e.Handled = true;

}

}

Также пользователь может ввести неверно повторный пароль. Этот случай также предусмотрен и продемонстрирован на рисунке 5.2.

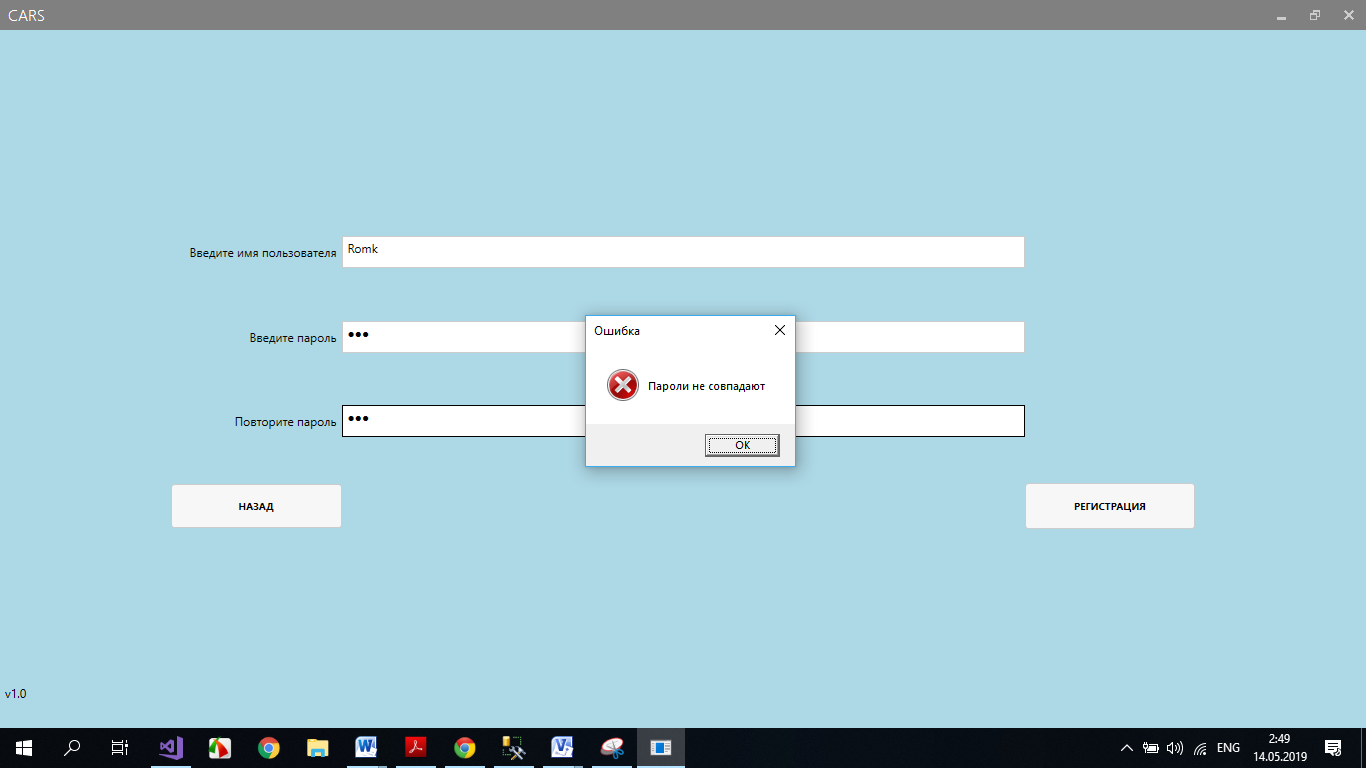


Рисунок 5.2 – Различные пароли

После авторизации пользователь попадает на домашнюю страницу с автомобилями, где может продать какой-либо автомобиль или просмотреть фотографию автомобиля. Чтобы пользователь не совершил каких-либо действий случайно, часто будет необходимо подтверждение пользователя, пример которого представлен на рисунке 5.3. Подобные запросы будут часто встречаться.

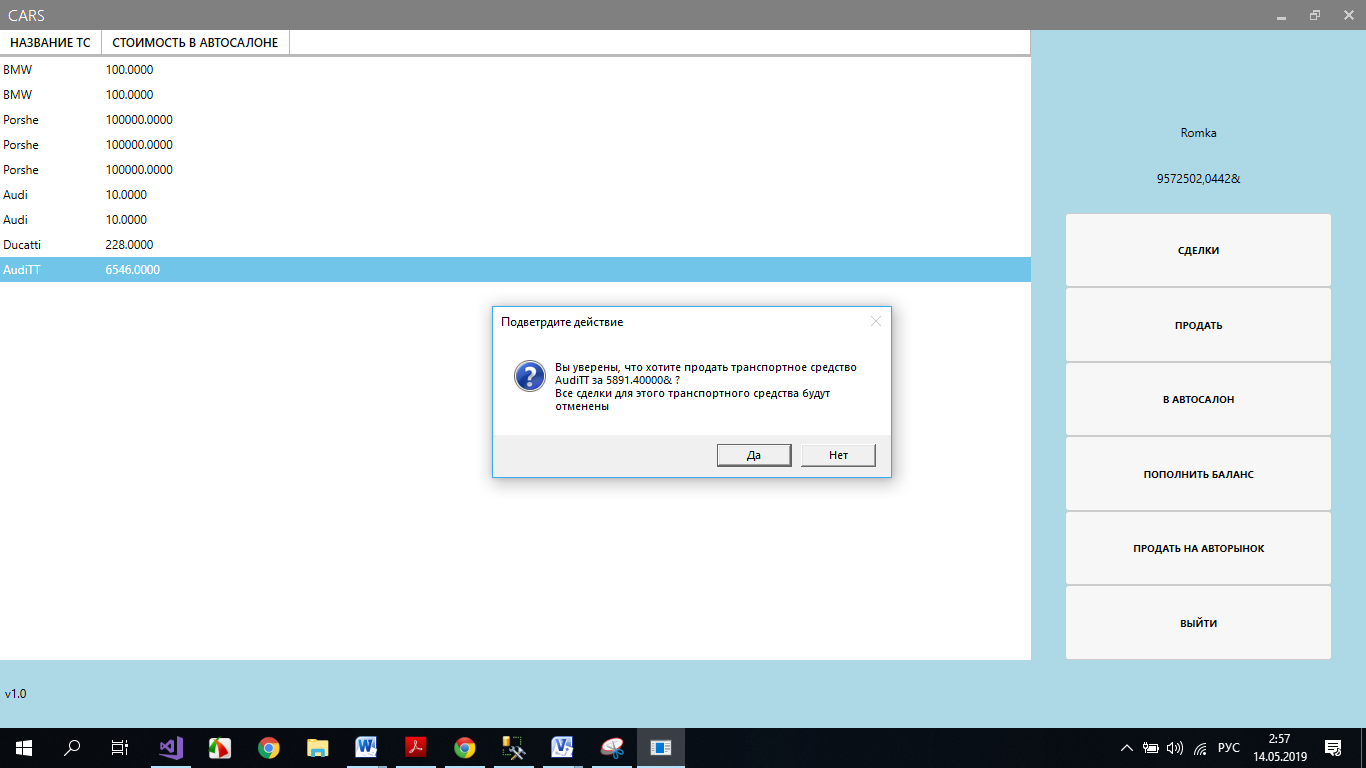


Рисунок 5.3 – Окно подтверждения

Предусмотрена ситуация, когда пользователь хочет продать свой автомобиль, но он его даже не выбрал. Результат метода проверки на рисунке 5.4.

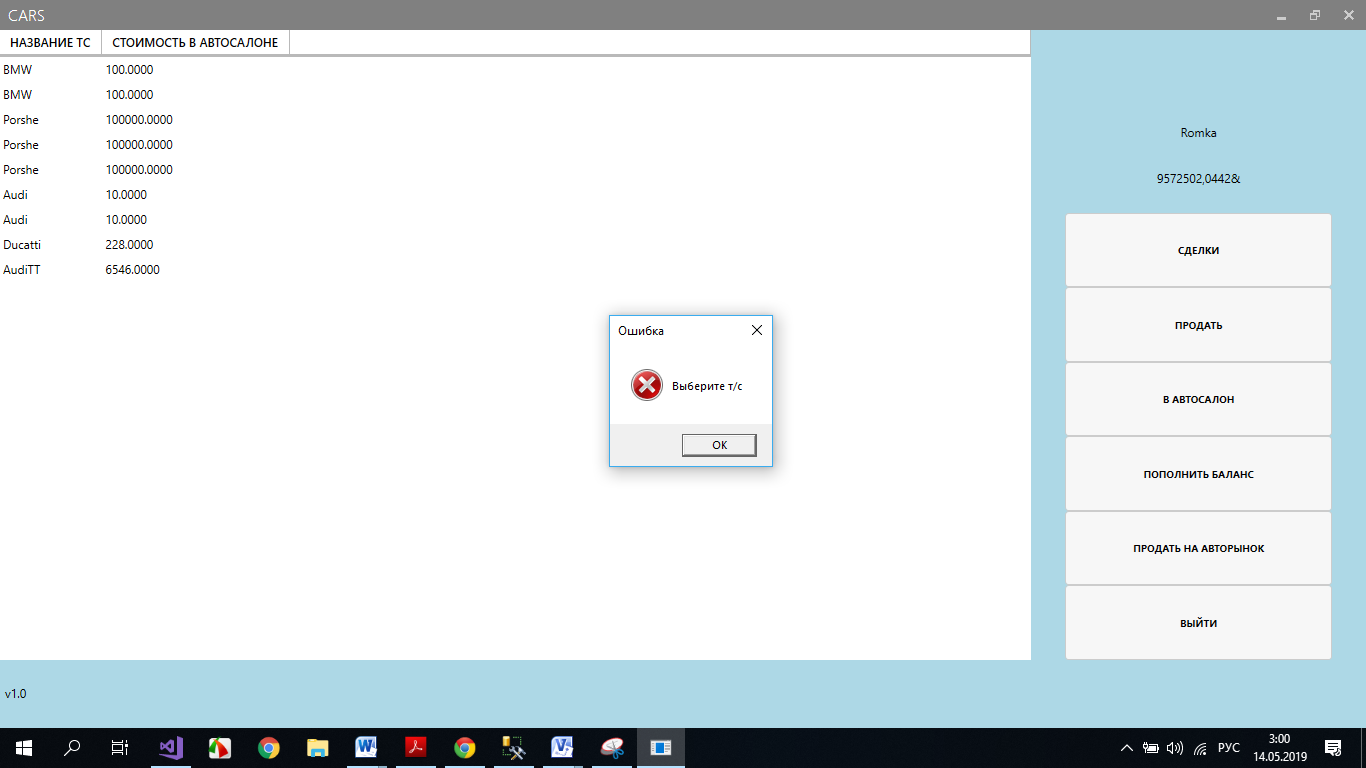


Рисунок 5.4 – Случайное нажатие

Также в некоторых полях, где необходимо вводить только цифры предусмотрен запрет на ввод букв при помощи метода, которой попросту не даст пользователю ввести что-либо кроме цифр и знаков точки, запятой, который необходимы для указания дробной составляющей в сумме пополнения баланса или продажи автомобиля:

private void Money\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

string inputSymbol = e.Text.ToString();

if (!Regex.Match(inputSymbol, @"[0-9\*.,]").Success)

{

e.Handled = true;

}

}

В левом нижнем углу пользователя всегда будет сопровождать надпись v1.0, которая намекает пользователю, что данное программное средство будет дорабатываться с учетом пожеланий пользователей.

И напоследок проверим работоспособность нашей информации о приложении, результат виден на рисунке 5.5.

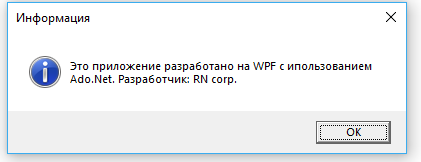


Рисунок 5.5 ­– Информация

# Руководство

На первоначальной странице приложения расположены два поля для ввода данных пользователя и две кнопки. Первая кнопка «Войти» позволяет выполнять вход пользователям, у которых уже есть учетная запись. Если учетной записи нет, то можно создать при помощи кнопки «Регистрация». Будьте внимательны, ведь восстановление и смена пароля в данной версии приложения не предусмотрена. Первоначальная страница изображена на рисунке 6.1.

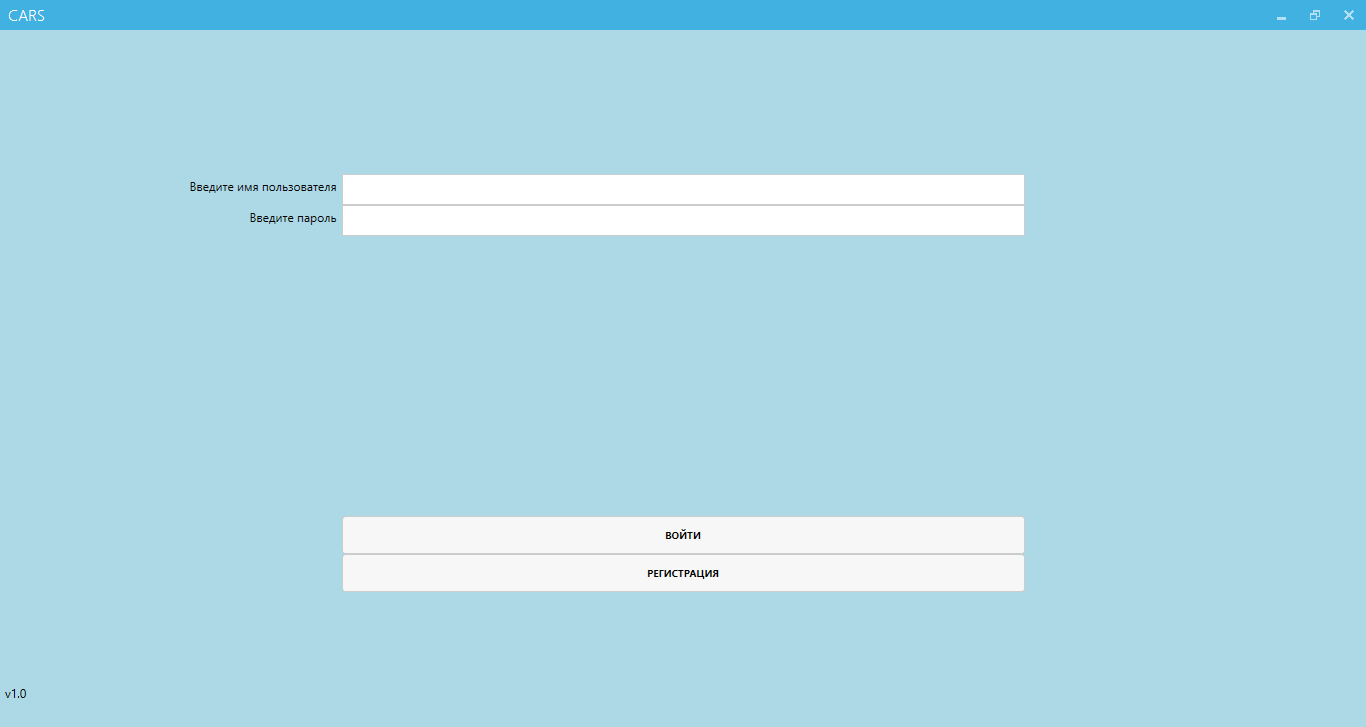


Рисунок 6.1 – Первоначальная страница

Перейдем на страницу регистрации с помощью кнопки «Регистрация» на первоначальной странице. Здесь присутствуют 3 поля для ввода: имя пользователя, пароль, поле для повторения пароля. А также 2 кнопки. Кнопка «Регистрация» служит для регистрации пользователя, если он заполнил все поля и соблюдены все правила, которые были описаны в главе 4.9. Страница регистрации представлена на рисунке 6.2.

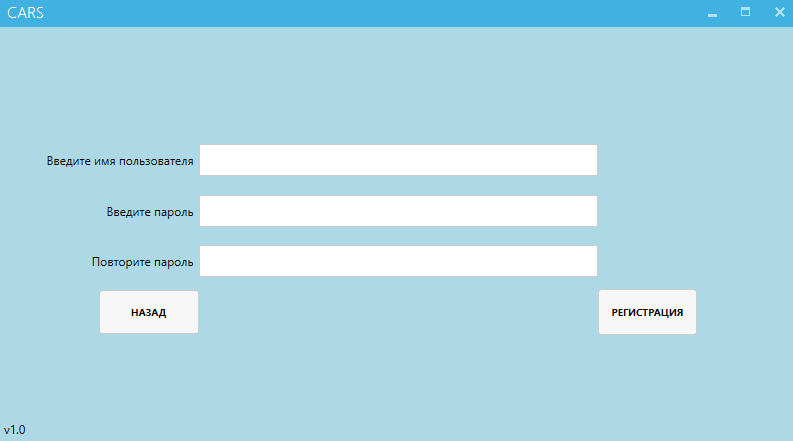


Рисунок 6.2 – Страница регистрации

В приложении предусмотрено две роли: пользователь и администратор. Роль пользователь выдаётся всем новым пользователям по умолчанию, а роль администратор есть в базе данных. Выдача пользователю прав администратора не предусмотрена.

После успешной регистрации автоматически открывается страница авторизации пользователя. Далее, если пользователь успешно авторизовался, открывается домашняя страница, на которой отображаются все названия и стоимость автомобилей, которые принадлежат данному пользователю, имя пользователя и его баланс. А также есть кнопки, которые позволяют сделать следующее:

* продать автомобиль быстро, но по уменьшенной цене;
* продать автомобиль пользователю за свою цену:
* перейти на страницу со списком доступных автомобилей:
* перейти на страницу с предложенными сделками от других пользователей;
* перейти на страницу пополнения баланса;
* перейти на страницу авторизации для смены пользователя.

Вид домашней страницы представлен на рисунке 6.3.

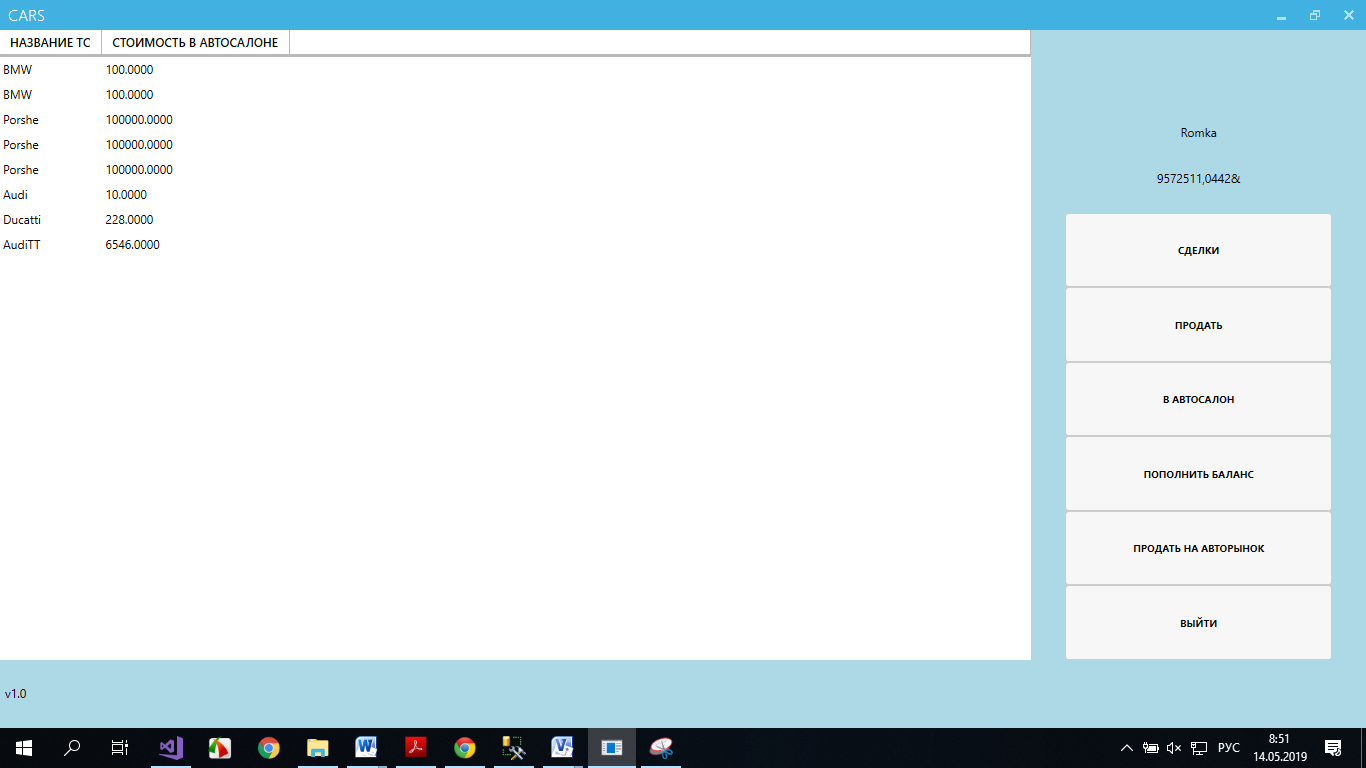


Рисунок 6.3 – Домашняя страница

Перейдём на страницу сделок при помощи кнопки «Сделки», которая расположена на домашней странице. На странице сделок пользователь может просмотреть все предложенные ему сделки от других пользователей, вернуться на домашнюю страницу, отклонить или принять предложение о покупке автомобиля за цену, которую предложил другой пользователь ранее. Страница сделок представлена на рисунке 6.4.

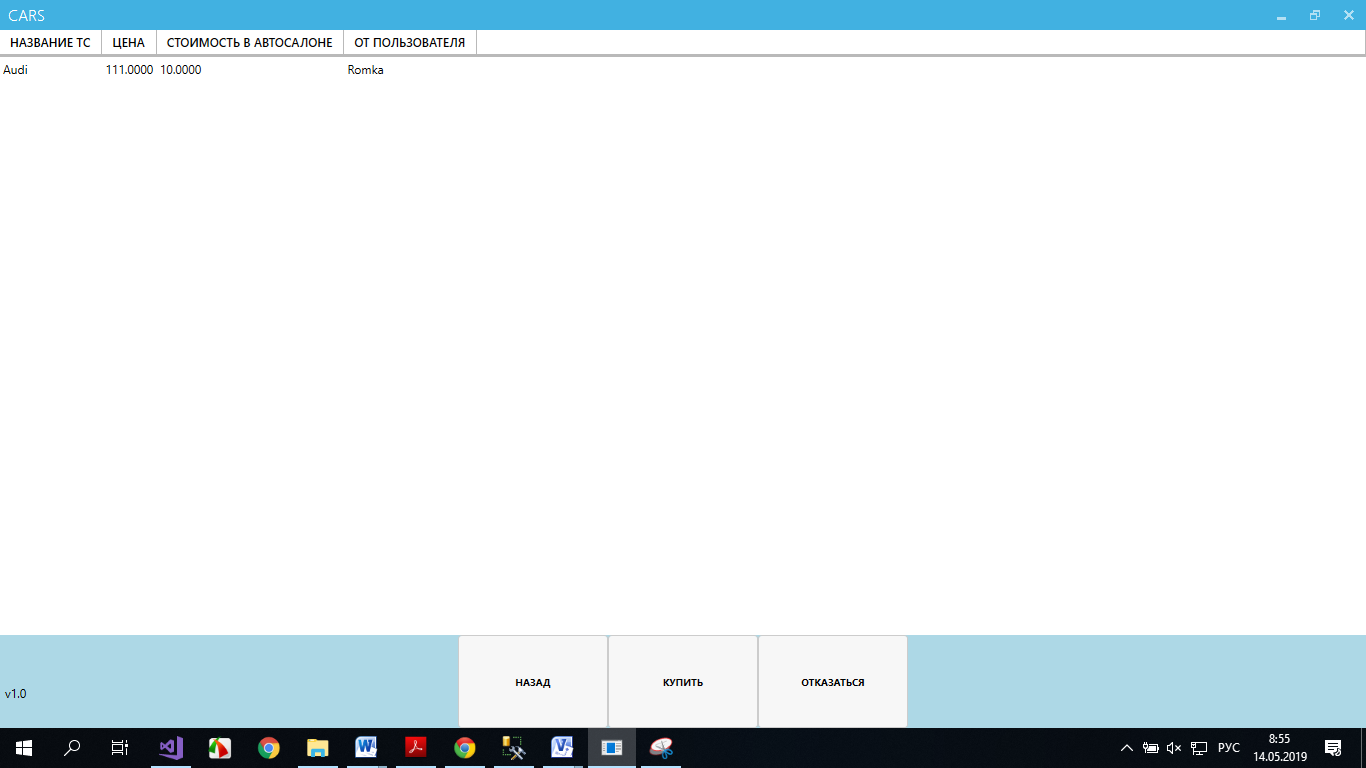


Рисунок 6.4 – Страница сделок

Вернёмся к домашней странице и рассмотрим кнопку «Продать». Данная кнопка позволяет предложить одному пользователю отправить предложение о покупке другому пользователю. Если выбран автомобиль для продажи, то пользователя переносит на страницу продажи автомобиля, где пользователь из списка других пользователя выбирает того, кому он хочет отправить предложение. А также необходимо ввести цену. Страница продажи автомобиля расположена на странице рисунке 6.5.

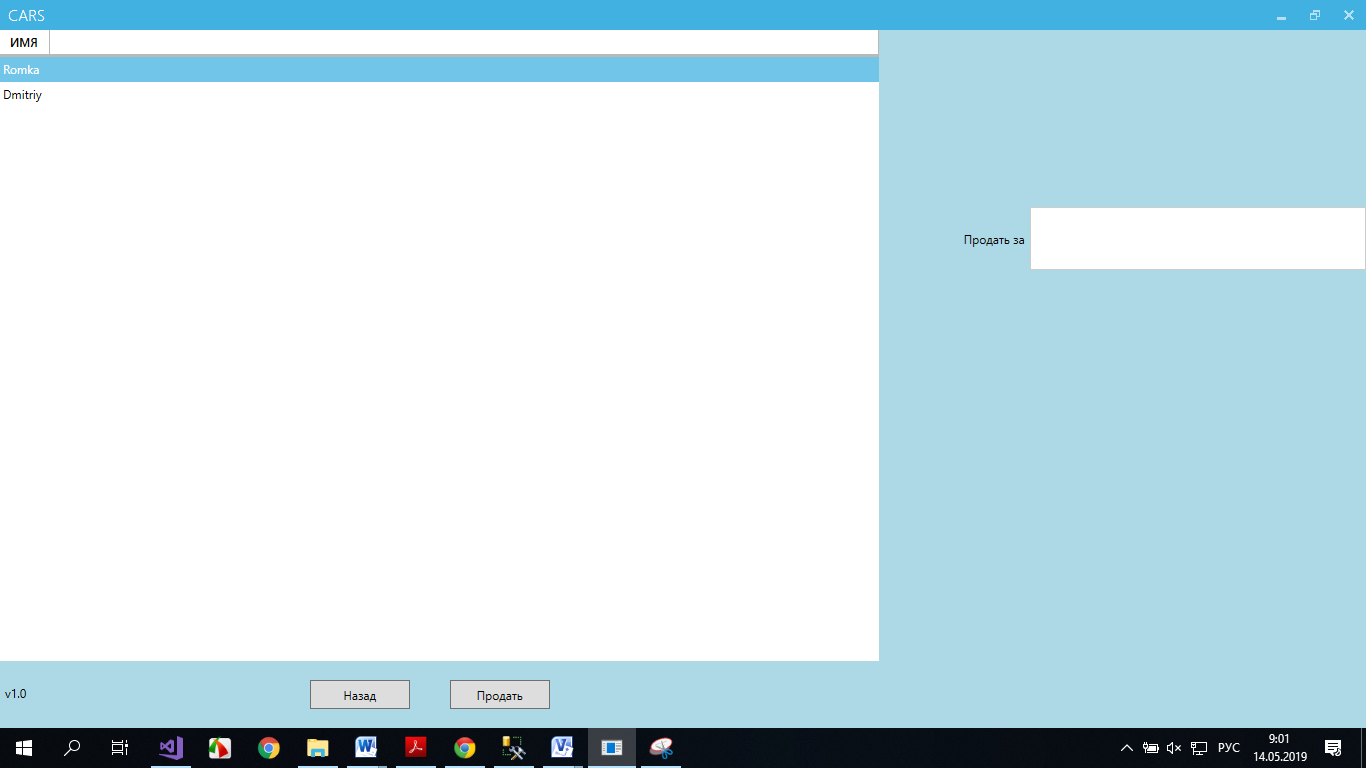


Рисунок 6.5 – Страница продажи

Перейдём обратно к домашней странице пользователя при помощи кнопки «Назад». Рассмотрим кнопку «В автосалон». Эта кнопка выполняет переход на страницу, где пользователь может приобрести интересующий его автомобиль или просмотреть изображения автомобиля, если это доступно, двойным нажатием на автомобиль, это также возможно сделать на домашней странице и на странице со сделками. На странице автосалона, в зависимости от роли пользователя, возможно отображение четырех кнопок:

* назад;
* приобрести автомобиль;
* добавить автомобиль (доступно администратору);
* изменить количество автомобилей (доступно администратору).

Вид страницы, которую видит пользователь с ролью администратор, автосалона представлен на рисунке 6.5.

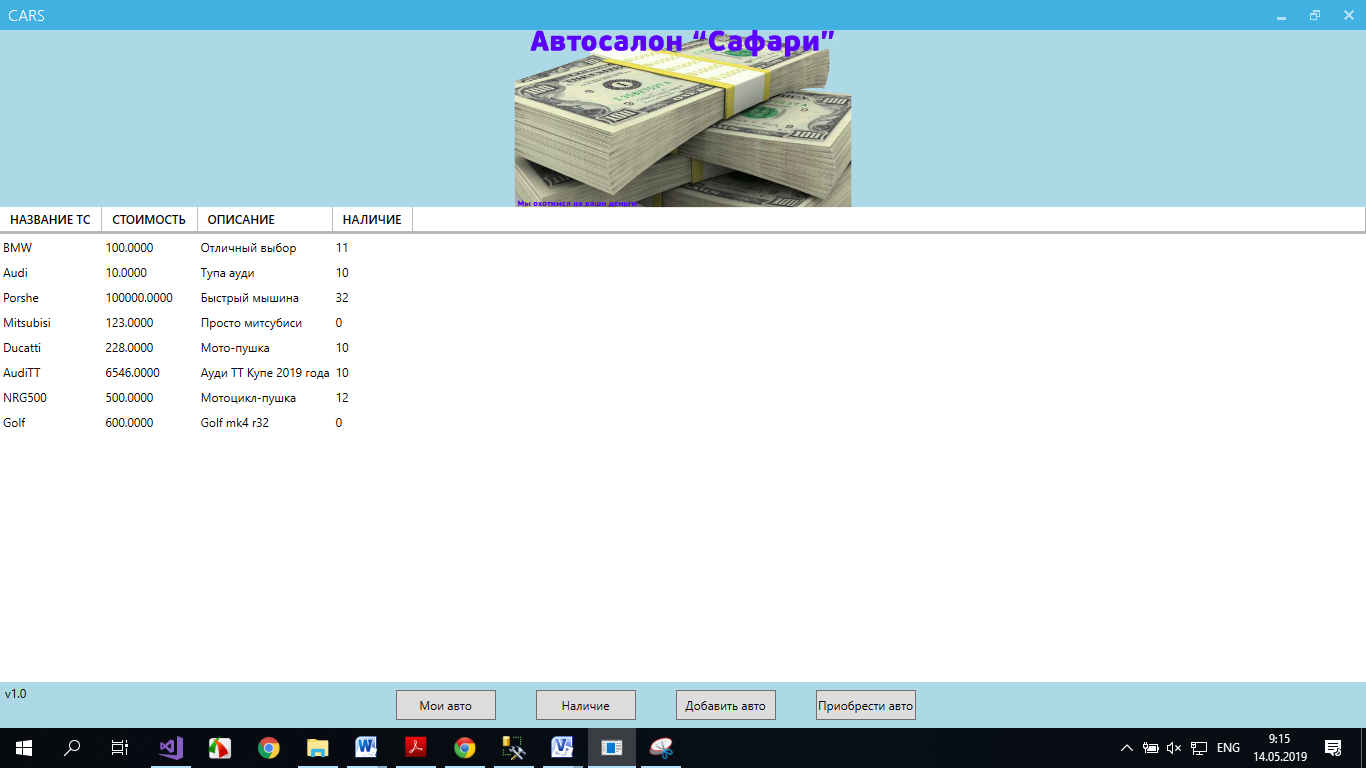


Рисунок 6.5 – Страница автосалона

При нажатии на кнопку «Приобрести авто» пользователь приобретает выбранный автомобиль, если он есть в наличии, и у него с баланса списывается сумма, которая указана в стоимости автомобиля.

При нажатии на кнопку «Мои авто» осуществляется переход на домашнюю страницу, а при нажатии на кнопку «Наличие» открывается страница, которая позволяет изменять количество доступных автомобилей для выбранного ранее администратором.

В случае нажатия кнопки «Добавить авто» администратор переходит на страницу, где необходимо ввести данные автомобиля и при нажатии кнопки «Добавить» администратору необходимо выбрать изображение, которое пользователь потом сможет просмотреть. Вид страницы расположен на рисунке 6.6.

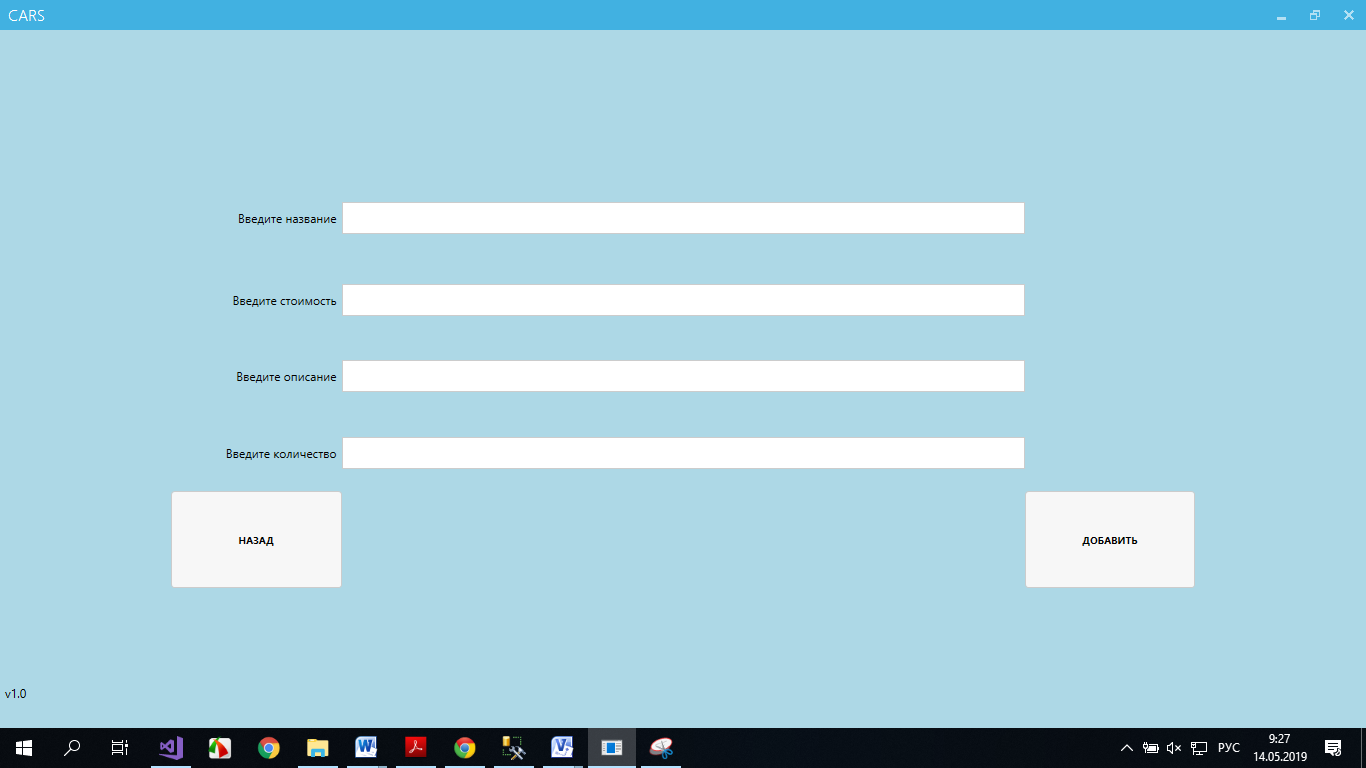


Рисунок 6.6 – Добавление автомобиля

Снова возвращаемся на домашнюю страницу. Сейчас рассмотрим элемент управления, который имеет вид кнопки с надписью «Пополнить баланс». Данная кнопка осуществляет переход на страницу пополнения баланса, где пользователю необходимо ввести номер карты, который состоит только из цифр, и сумму пополнения. Здесь можно ввести только цифры. Данная страница представлена ниже на рисунке 6.7

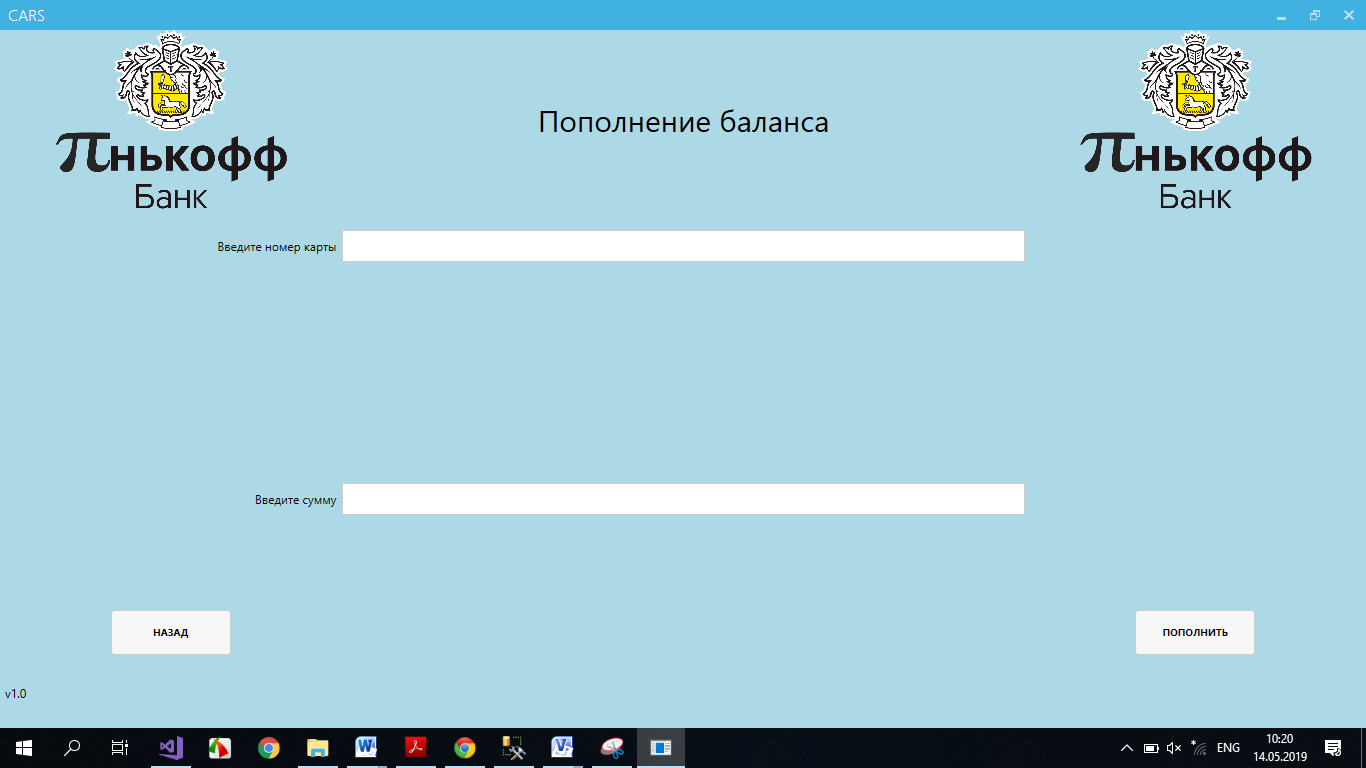


Рисунок 6.7 – Страница пополнения баланса

Вот мы уже добрались до описания предпоследней кнопки под названием «Продать на авторынок». Данная кнопка позволяет продать автомобиль, который выбрал пользователь быстро. Однако пользователь продаёт его не за ту цену, которая была указана в автосалоне, а за 90% от этой цены. Здесь также, как и в многих других функциях будет необходимо подтверждение пользователя, что он хочет продать автомобиль. Окошко подтверждения изображено ниже на рисунке 6.8.

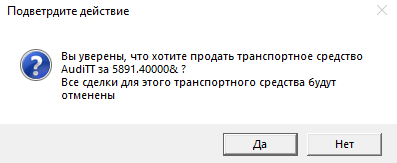


Рисунок 6.8 – Окошко подтверждения

И наконец, последняя кнопка, которая осталась в данной версии приложения. Кнопка «Выйти» выполняет переход на страницу авторизации и позволяет пользователю сменить аккаунт.

Также хочется отметить, что приложение имеет свойство адаптивности. Это значит, что размеры элементов изменяются вместе с размерами окна. И у приложения простой, спокойный дизайн без использования каких-то ресурсозатратных библиотек и прочего, что позволяет приложению работать быстро даже на самых слабых устройствах.

На этом мы закончим наше руководство, так как в данной версии приложения больше нет функционала, который мы не рассмотрели.

# ЗАКЛЮЧЕНИЕ

В самом начале была поставлена задача: реализовать некое программное средство с использованием ADO.NET и WPF, которое в будущем получило название «Менеджер автомобилей». Перед началом были изучены некоторые аспекты WPF и ADO.NET, которые и позволили реализовать то, что было задумано изначально.

В процессе решения поставленной задачи была достигнута поставленная цель по созданию приложения «Менеджер автомобилей». При разработке были выполнены все пункты из указанного списка предполагаемого основного функционала приложения. Таким образом, была достигнута цель и создан менеджер.

В программном средстве были реализованы следующие функции:

* авторизация и регистрация;
* пополнение баланса пользователя;
* покупка автомобиля;
* просмотр фотографии автомобиля;
* добавление автомобиля;
* продажа автомобиля различными способами;
* другие функции для удобств пользователя.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

# СПИСОК ЛИТЕРАТУРЫ

[1] Мэтью Мак-Дональд «Pro WPF 4.5 in C#: Windows Presentation» –Apress, 2012 –1095 с.

[2] Статья “WPF – система авторизации и регистрации” [Электронный ресурс]. – Электронные данные. – https://you-hands.ru/2018/08/31/wpf-sistema-avtorizacii-i-registracii/

[3] Статья “Подключение к базе данных” [Электронный ресурс]. – Электронные данные. – https://metanit.com/sharp/wpf/21.1.php

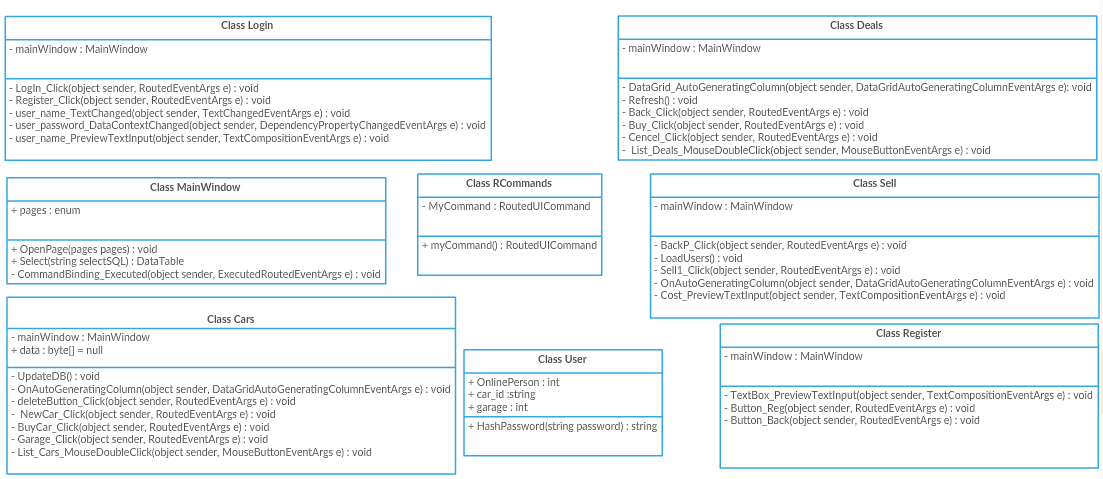
[4] Статья ”Report-Driven Design” [Электронный ресурс]. – Электронные данные. – https://habr.com/ru/company/devexpress/blog/119876/

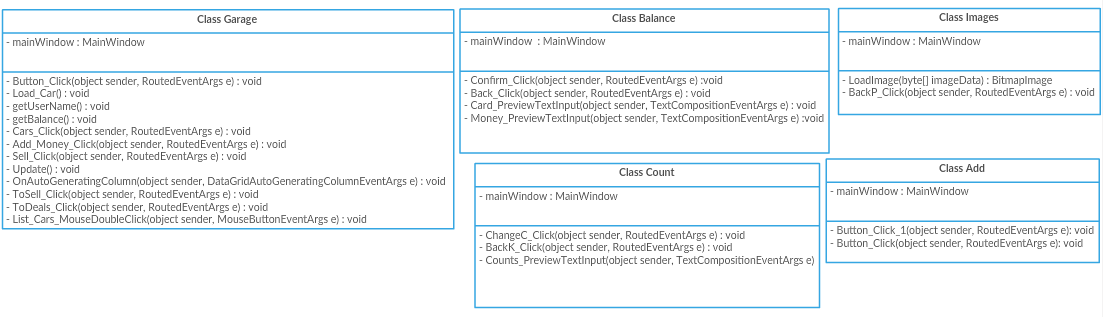
[5] Программа Автосервис [Электронный ресурс]. – Электронные данные. – <https://flagman-it.ru/product/1s:avtoservis_klientskaya_licenziya_na_1_rabochee_mesto>

[6] Система программ "Альфа-авто" [Электронный ресурс]. – Электронные данные. – <https://flagman-it.ru/product/alfa_avto:_avtosalon_avtoservis_avtozapchasti_prof>

[7] Программа 1С:Розница [Электронный ресурс]. – Электронные данные. – <https://flagman-it.ru/product/1s:roznica_8_magazin_avtozapchastey>

# ПРИЛОЖЕНИЕ А





# ПРИЛОЖЕНИЕ Б

Листинг 1. Класс Login:

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media;

using System.Data;

using System.Text.RegularExpressions;

namespace KP

{

/// <summary>

/// Логика взаимодействия для login.xaml

/// </summary>

public partial class Login : Page

{

private MainWindow mainWindow;

public Login(MainWindow \_mainWindow)

{

InitializeComponent();

mainWindow = \_mainWindow;

}

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

{

int user\_id=0;

if (user\_name.Text.Length > 0) // проверяем введён ли логин

{

if (user\_password.Password.Length > 0) // проверяем введён ли пароль

{

// ищем в базе данных пользователя с такими данными

string pass = User.HashPassword(user\_password.Password);

DataTable dt\_user = mainWindow.Select("SELECT Id FROM [dbo].[users] WHERE [login] = '" + user\_name.Text +

"' AND [password] = '" + pass + "'");

foreach (DataRow row in dt\_user.Rows)

{

var Ids = row.ItemArray;

foreach (int id in Ids)

user\_id = id;

}

if (dt\_user.Rows.Count > 0) // если такая запись существует

{

MessageBox.Show("Пользователь авторизовался", "Информация", MessageBoxButton.OK, MessageBoxImage.Asterisk); // говорим, что авторизовался

User.OnlinePerson = user\_id;

mainWindow.OpenPage(MainWindow.pages.garage);

}

else MessageBox.Show("Проверьте корректность введенных данных", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); // выводим ошибку

}

else // выводим ошибку

{

user\_password.BorderBrush = System.Windows.Media.Brushes.Red;

MessageBox.Show("Введите Пароль", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); // выводим ошибку

}

}

else

{

user\_name.BorderBrush = System.Windows.Media.Brushes.Red;

MessageBox.Show("Введите логин", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error); // выводим ошибку

}

}

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

{

mainWindow.OpenPage(MainWindow.pages.register);

}

private void user\_name\_TextChanged(object sender, TextChangedEventArgs e)

{

user\_name.BorderBrush = Brushes.Gray;

}

private void user\_password\_DataContextChanged(object sender, DependencyPropertyChangedEventArgs e)

{

user\_password.BorderBrush = Brushes.Gray;

}

private void user\_name\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

string inputSymbol = e.Text.ToString();

if (!Regex.Match(inputSymbol, @"[a-zA-zа-яА-Я0-9\_,.@$#!\*]").Success)

{

e.Handled = true;

}

}

}

}

Листинг 2. Класс Register:

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Data;

using System.Text.RegularExpressions;

namespace KP

{

/// <summary>

/// Логика взаимодействия для Page1.xaml

/// </summary>

public partial class Register : Page

{

private MainWindow mainWindow;

public Register(MainWindow \_mainWindow)

{

InitializeComponent();

mainWindow = \_mainWindow;

}

private void Button\_Reg(object sender, RoutedEventArgs e)

{

if (textBox\_login.Text.Length > 0) // проверяем логин

{

if (password.Password.Length > 0) // проверяем пароль

{

if (password\_Copy.Password.Length > 0) // проверяем второй пароль

{

if (password.Password == password\_Copy.Password) // проверка на совпадение паролей

{

DataTable dt\_user = mainWindow.Select($"SELECT Id FROM [dbo].[users] WHERE [login] = '{textBox\_login.Text}'");

if (dt\_user.Rows.Count <= 0)

{

string pass = User.HashPassword(password.Password);

//int pass = password.Password.GetHashCode();

mainWindow.Select($"INSERT INTO [dbo].[users](login,password) VALUES ('{textBox\_login.Text}', '{pass}')");

MessageBox.Show("Пользователь зарегистрирован", "Информация", MessageBoxButton.OK, MessageBoxImage.Asterisk);

mainWindow.OpenPage(MainWindow.pages.login);

}

else MessageBox.Show("Имя пользователя занято", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else MessageBox.Show("Пароли не совподают", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else MessageBox.Show("Повторите пароль", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else MessageBox.Show("Укажите пароль", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else MessageBox.Show("Укажите логин", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

private void Button\_Back(object sender, RoutedEventArgs e)

{

mainWindow.OpenPage(MainWindow.pages.login);

}

private void TextBox\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

string inputSymbol = e.Text.ToString();

if (!Regex.Match(inputSymbol, @"[a-zA-zа-яА-Я0-9\_,.@$#!\*]").Success)

{

e.Handled = true;

}

}

}

}