

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Направление подготовки (шифр, название) 09.03.04 «Программная инженерия»

Дневник практики обучающегося 2 курса

Фамилия Стриженок

Имя Матвей

Отчество Иванович

Обучающийся

Стриженок М. И.

подпись

расшифровка подписи

Место практики: Научно-исследовательская лаборатория "Лаборатория мобильной и веб-разработки" ООО НПИЦИТ "Иносфера"
наименование профильной организации

Вид практики Учебная практика

Тип практики Практика по получению первичных профессиональных умений и навыков

Способ проведения практики Стационарная

Форма проведения практики Дискретная (по видам практик)

Сроки прохождения практики с 10.02.2020 по 10.05.2020

***Индивидуальное задание, содержание, планируемые результаты и рабочий график
(план) проведения практики согласованы:***

Руководитель практики
от структурного подразделения

ЮФУ

Пилипушко Е.М.

ФИО, подпись

I СОДЕРЖАНИЕ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ПРАКТИКИ

(Заполняется руководителем практики в соответствии с программой практики)

Планируемые результаты обучения при прохождении учебной практики (практики по получению первичных профессиональных умений и навыков) в рамках планируемых результатов освоения основной образовательной программы направлению подготовки 09.03.04 «Программная инженерия» (уровень бакалавриат).

<u>Код компетенции</u>	<u>Содержание компетенции</u>	<u>Результаты обучения</u>
ПК-1	готовность применять основные методы и инструменты разработки программного обеспечения	Знать: –методы и приемы формализации задач; –алгоритмы решения типовых задач, области и способы их применения; –методы и приемы алгоритмизации поставленных задач; –методы и приемы отладки программного кода; –методы повышения читаемости программного кода.
ПК-2	владением навыками использования операционных систем, сетевых технологий, средств разработки программного	Знать: –технологии разработки и отладки системных продуктов; –механизмы авторизации, аутентификации и защиты от несанкционированного доступа. Уметь:

	интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	<ul style="list-style-type: none"> – оценивать вычислительную сложность функционирования разрабатываемых программных продуктов; – применять технологии разработки компиляторов. <p>Владеть:</p> <ul style="list-style-type: none"> – технологиями администрирования ОС.
ПК-3	владением навыками использования различных технологий разработки программного обеспечения	<p>Знать:</p> <ul style="list-style-type: none"> – методологию разработки программного обеспечения; – методы и средства сборки и интеграции программных модулей и компонент; – интерфейсы взаимодействия внутренних модулей системы; – языки, утилиты и среды программирования, средства пакетного выполнения процедур; – стили написания кода.
ПК-4	владением концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том	<p>Знать:</p> <ul style="list-style-type: none"> – основные атрибуты качества ПО; – методики оценки качества ПО; – стандарты, регламентирующие требования к пользовательским интерфейсам.

	числе роли людей, процессов, методов, инструментов и технологий обеспечения качества	<p>Уметь:</p> <ul style="list-style-type: none"> – оценивать пользовательский интерфейс ПО – уметь оценивать функциональные характеристики ПО.
ПК-5	владением стандартами и моделями жизненного цикла	<p>Знать:</p> <ul style="list-style-type: none"> – процессы жизненного цикла ПО: управление проектом, управление требованиями, управление конфигурацией и изменениями, анализ и проектирование ПО, тестирование ПО. <p>Уметь:</p> <ul style="list-style-type: none"> – применять методологии и средства проектирования программного обеспечения. <p>Владеть:</p> <ul style="list-style-type: none"> – владеть методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения.
ПК-18	владением навыками моделирования, анализа и использования	<p>Знать:</p> <ul style="list-style-type: none"> – методы разработки, анализа и проектирования ПО; – знать методы и средства проектирования ИР;

	формальных методов конструирования программного обеспечения	– знать шаблоны (стили) проектирования слоев компонентов;
ПК-19	способностью оценивать временную и емкостную сложность программного обеспечения	Знать: – методы оценки сложности алгоритмов. Уметь: – анализировать алгоритмическую сложность ПО.
ПК-20	владением навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации	Знать: – методы и средства проектирования программного обеспечения; Уметь: – выбирать средства реализации требований к программному обеспечению.
ПК-21	способностью создавать программные интерфейсы	Знать: – критерии оценки юзабилити- и эргономических характеристик; – методы юзабилити- тестирования; – стандарты, регламентирующие требования к пользовательским интерфейсам. Уметь: – интерпретировать бизнес- задачи в параметры,

		характеризующие качество интерфейса.
--	--	--------------------------------------

Содержание практики. Во время практики обучающийся должен проработать ряд вопросов, которые являются общими и обязательными для всех практикантов вне зависимости от индивидуального задания на практику.

<u>№</u>	<u>Раздел ы (этапы) прохождени я практики</u>	<u>Содержание практики</u>	<u>Результат</u>	
			<u>вид отчетнос ти</u>	<u>код компе тенции по ООП</u>
1.	Подготовите льный этап.	Разъяснение этапов практики. Инструктаж по технике безопасности. Формулировка темы практики и постановка индивидуального задания на практику	Дне вник практики	ПК-1, ПК-3, ПК-5, ПК-18, ПК- 20
2.	Основн ой этап. Разработка и описание программны х систем.	Тема 1. Сбор информации по теме практики. Тема 2. Разработка требований к программной системе. Тема 3. Проектирование архитектуры программной системы. Тема 4. Разработка программной системы в соответствии с заданием.	Дне вник практики, отчет по практике	ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-18, ПК- 19, ПК-20, ПК-21

		Тема 5. Описание по применению программной системы.		
3.	Заключ ительный этап.	Написание и представление отчета по практике. Защита практики.	Дне вник практики, Отчет по практике	ПК-2, ПК-4, ПК-5, ПК-19, ПК- 20

II ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ ОБУЧАЮЩЕГОСЯ НА ПРАКТИКУ

Содержание индивидуального задания на практику

-
1. Разработка и реализация на языке программирования C# автоматизированной информационной системы для продажи автомобилей.
 2. В АИС должны размещаться объявления о продажах автомобилей.
 3. В системе реализовано три типа пользователей: администратор, зарегистрированный пользователь и незарегистрированный пользователь («гость»). Каждый пользователь будет обладать своими смысловыми функциями и возможностями.
 4. Гость может просматривать объявления.
 5. Зарегистрированный пользователь: просматривать, добавлять, редактировать и удалять объявления. Удалять он может только свои объявления.
 6. Администратор: просматривать, добавлять и удалять объявления. Может удалять любые объявления.
 7. Сделать вывод информации об объявлениях в CSV формат.
-

III РАБОЧИЙ ГРАФИК (ПЛАН) ПРОВЕДЕНИЯ ПРАКТИКИ

№	Дата	Содержание работы
1	10.02.2020	Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, правилам внутреннего распорядка
2	11.02.2020 – 15.02.2020	Анализ требований для создания АИС
3	16.02.2020 – 3.03.2020	Создание баз данных и их тестирование
4	4.03.2020 – 11.03.2020	Продумывание хода работы и вида готовой работы (интерфейса)
5	12.03.2020 – 25.03.2020	Изучение необходимого материала и начало создания системы
6	26.03.2020 – 30.04.2020	Конец реализации и изучения основных компонентов программы
7	1.05.2020 – 7.05.2020	Тестирование, корректировка и исправление ошибок
8	8.05.2020	Завершение работы над программой
9	9.05.2020	Проверка работы руководителем
10	10.05.2020	Составление отчета по практике, защита практики

**IV ИНСТРУКТАЖ ПО ОЗНАКОМЛЕНИЮ С ТРЕБОВАНИЯМИ
ОХРАНЫ ТРУДА, ТЕХНИКИ БЕЗОПАСНОСТИ, ПОЖАРНОЙ
БЕЗОПАСНОСТИ, ПРАВИЛАМ ВНУТРЕННЕГО РАСПОРЯДКА**

	Инструктаж проведен	Ознакомлен
по требованиям охраны труда	<div style="text-align: center;"> <hr style="width: 200px; margin: 0 auto;"/> — <i>(подпись руководителя практики от профильной организации)</i> «10» февраля 2020 г. </div>	<div style="text-align: center;"> <hr style="width: 200px; margin: 0 auto;"/> — <i>(подпись обучающегося)</i> «10» февраля 2020 г. </div>
по техники безопасност и		
по пожарной безопасност и		
по правилами внутреннего трудоого распорядка		

VI ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ ОТ СТРУКТУРНОГО ПОДРАЗДЕЛЕНИЯ ЮФУ

Студент 2 курса Стриженок Матвей Иванович, направления 09.03.04 «Программная инженерия», прошел в 4 семестре учебную практику в Научно-исследовательской лаборатории "Лаборатория мобильной и веб-разработки" ООО НПИЦИТ "Иносфера".

В период учебной практики Стриженок М.И. разрабатывал АИС по продаже автомобилей. Им же были решены следующие задачи:

1. Проведено исследование предметной области для использования готового программного продукта.
2. Проведен анализ требований к выполнению поставленной задачи.
3. Спроектирована архитектура программной системы.
4. Разработана база данных для хранения объявлений, информации о пользователях и продажах.
5. Разработана программная система в соответствии с заданием
6. Проведено тестирование данной АИС

За время прохождения практики Стриженок М.И. показал достаточный уровень теоретической подготовки, высокую степень умения и навыков применять и использовать знания, полученные в университете, для решения поставленных перед ним практических задач.

Стриженок М.И. проявил следующие личностные и профессиональные качества: организованность, своевременность, упорство и усидчивость в решении поставленных задач.

Считаю, что проявленные профессиональные качества полностью удовлетворяют потребностям лаборатории, программа практики выполнена в полном объеме, сроки выполнения заданий соблюдались полностью.

Рекомендуемая оценка «отлично»/ «хорошо».

Руководитель практики от
структурного подразделения
ЮФУ

Старший
преподаватель, Пилипушко
Елена Михайловна

подпись

расшифровка подписи

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Направление подготовки (шифр, название) 09.03.04 «Программная инженерия»

Отчет по практике

обучающегося 2 курса

Фамилия Стриженок

Имя Матвей

Отчество Иванович

Обучающийся

Стриженок М.И.

подпись

расшифровка подписи

Место практики Научно-исследовательская лаборатория "Лаборатория мобильной и веб-разработки" ООО НПИЦИТ "Иносфера"

наименование профильной организации

Вид практики Учебная практика

Тип практики Практика по получению первичных профессиональных умений и навыков

Способ проведения практики Стационарная

Форма проведения практики Дискретная (по видам практик)

Сроки прохождения практики с 10.02.2020 по 10.05.2020

Руководитель практики

от структурного подразделения

ЮФУ

Пилипушко Е.М.

ФИО, подпись

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	15
ВВЕДЕНИЕ.....	16
1 Анализ требований	18
2 Рабочий график (план) проведения практики	19
3 Ход выполнения работы	20
3.1 Техническое задание.....	20
3.2 Постановка задачи	29
3.3 Инструментарий.....	29
3.4 Проектирование модели БД.....	30
3.5 Интерфейс	32
3.5.1 Тестирование	39
4 Фрагменты кода программы	40
ЗАКЛЮЧЕНИЕ	51
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	52

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем документе применяются следующие обозначения и сокращения:

БД – база данных

ПО – программное обеспечение

СУБД – система управления базой данных

АИС – автоматизированная информационная система

ТЗ – техническое задание

ВВЕДЕНИЕ

Основной целью настоящей учебной практики является: расширение и закрепление теоретических знаний студентов через получение первичных профессиональных навыков, ознакомление студентов с характером и спецификой будущей деятельности.

Задачами учебной практики при данных показателях являются:

- углубление и расширение теоретических знаний, умений и навыков, полученных студентами в процессе теоретического обучения;
- получение первичных профессиональных умений и навыков проектирования и производства программного продукта;
- получение практического опыта по решению стандартных задач профессиональной деятельности;
- закрепление, углубление и расширение теоретических знаний, умений и навыков, полученных студентами в процессе теоретического обучения;
- обеспечение практической направленности в изучении дисциплин;
- овладение функционалом и инструментарием различных средств, поддерживающих создание программного обеспечения для решения практических задач;
- овладение навыками подготовки презентации и оформления научно-технических отчетов по результатам выполненной работы, а также методических материалов и пособий по применению программных систем.

В наше время пользуются популярностью различные сайты и АИС для продажи автомобилей. До сих пор появляются новые виртуальные площадки по продажам авто.

Задачей моей работы было создать программный продукт, который обеспечивает просмотр, добавление объявлений о продаже, а также ряд других возможностей. Функции администратора включают удаление любых объявлений из базы данных и ведение статистики продаж.

1 Анализ требований

К разрабатываемому ПО представлены следующие требования:

- Для хранения данных рекомендуется использовать текстовые файлы или любую СУБД. Все данные должны быть структурированы, разные файлы (таблицы) – связаны с собой по ключевым полям. Должно быть не менее 3-4 файлов (таблиц). Самостоятельное заполнение всех таблиц тестовыми данными. Кроме того, необходимо наличие каталога графических файлов, каким-то образом связанных с данными в таблицах.

- Разработанное ПО должно считывать данные из файла и выполнять выборку необходимых данных по запросу пользователя (в т. ч. используя данные одновременно из нескольких таблиц). Перечень запросов для реализации согласовывается с руководителем практики.

- Для разработки программного приложения допускается использовать только языки программирования C# или Java.

- Все экранные формы приложения должны соответствовать единому стилю. На формах должны присутствовать надписи, текстовые поля, кнопки, списки, таблицы и т. д. Приложение должно обеспечивать работу со стандартными компонентами среды разработки. Одна из форм должна работать с отображением графических данных (файлов изображений).

- Приложение должно обеспечивать возможность распечатывать выбранные данные.

- При выполнении работы должны быть самостоятельно изучены вопросы тестирования программного обеспечения. Должно быть проведено тестирование ПО.

2 Рабочий график (план) проведения практики

Совместно с руководителем был создан рабочий график проведения учебной практики, представленный в таблице см. Таблица 1.

Таблица 1 – План проведения практики

№	Дата	Содержание работы
1	10.02.2020	Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, правилам внутреннего распорядка
2	11.02.2020 – 15.02.2020	Анализ требований для создания АИС
3	16.02.2020 – 3.03.2020	Создание баз данных и их тестирование
4	4.03.2020 – 11.03.2020	Продумывание хода работы и вида готовой работы(интерфейса)
5	12.03.2020 – 25.03.2020	Изучение необходимого материала и начало создания системы
6	26.03.2020 – 30.04.2020	Конец реализации и изучения основных компонентов программы
7	1.05.2020 – 7.05.2020	Тестирование, корректировка и исправление ошибок
8	8.05.2020	Завершение работы над программой
9	9.05.2020	Проверка работы руководителем
10	10.05.2020	Составление отчета по практике, защита практики

3 Ход выполнения работы

На начальных этапах работы с проектом необходимо создать ТЗ

3.1 Техническое задание

ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Требуется разработать и реализовать на языке программирования С# автоматизированную информационную систему «Магазин по продаже автомобилей». Данная АИС нужна для размещения объявлений о продаже различных автомобилей.

ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Для хранения данных должна быть создана база данных (БД). В качестве СУБД будет использована MS SQL. В БД должна храниться следующая информация: данные о зарегистрированных пользователях, данные об автомобилях и данные о проданных автомобилях и изображения авто. Таблицы будут связываться с помощью ID.

СПЕЦИФИКАЦИЯ ДАННЫХ

Пользователи

- ID пользователя (int) PK Unique
- Имя пользователя (строка 50 символов)
- Login (строка 50 символов)
- Password (строка 25 символов)
- Тип пользователя (строка)
- Номер телефона (строка)

Автомобили

- ID объявления (int) PK Unique
- Тип кузова (строка 30 символов)
- Тип двигателя (строка 30 символов)
- Состояние (строка 50 символов)
- Марка (строка 50 символов)
- Модель (строка 50 символов)
- Цена (строка 50 символов)

- Год выпуска (строка 10 символов)
- Пробег (строка 50 символов)
- Тип привода (строка 50 символов)
- Место продажи (строка 50 символов)
- User Id (int)
- Краткое описание от владельца (строка 350 символов)

Продажа

- Id (int)
- ID продавца (внешний ключ)
- ID объявления (внешний ключ)
- Дата продажи (Строка)

Фотографии

- Фотография (массив)
- id объявления (int)

Связи между таблицами см. Рисунок 1

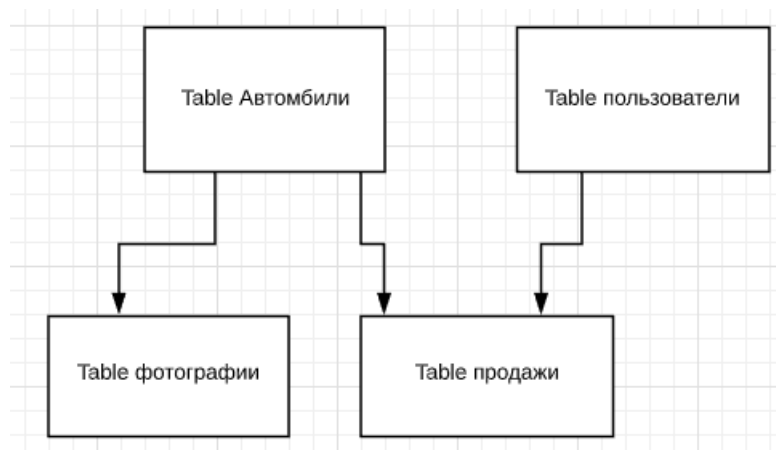


Рисунок 1 - Связи между таблицами

ОПИСАНИЕ ПРИЛОЖЕНИЯ

В АИС должны быть реализованы следующие формы: начальная, форма регистрации, форма авторизации для зарегистрированных пользователей, формы

просмотра информации об автомобилях, форма добавления объявления, форма редактирования объявления, форма удаления объявления и другие.

В формах должны присутствовать следующие элементы управления: Button, Label, TextBox, MaskedTextBox, Radiobutton, DateTimePicker, PictureBox, ComboBox, ProgressBar, MonthCalendar, CheckBox.

В формах должна быть реализована возможность: открытия файлов с помощью компонентов OpenFileDialog; сохранения файлов с помощью компонентов SaveFileDialog; вызова различных диалоговых окон при помощи MessageBox; перехода между формами/экранами; передачи данных между формами/экранами; работы с Windows Forms контейнерами; динамического добавления элементов; создания списка с простыми элементами; создания списка с составными элементами: каждый элемент списка это несколько разных компонент; работы со списками: добавление элементов, удаление, изменение, фильтрация списка; создания таблицы; работы с таблицами: добавление элементов, удаление, изменение, фильтрация таблицы по полям; экспорта данных таблицы в Excel или csv; отправки на печать текста, изображения.

Разработанное приложение должно импортировать данные из файла в приложение и выполнить обработку данных по запросу пользователя (в т. ч. используя одновременно данные из нескольких таблиц).

В приложении должны обрабатываться ошибки неправильного ввода.

Приложение должно обеспечивать возможность сохранять образ в файл.

Все экранные формы приложения должны соответствовать единому стилю и иметь логотип. На формах должны присутствовать надписи, текстовые поля, кнопки, списки, выпадающие списки, таблицы и т.д. Приложение должно обеспечивать работу со стандартными компонентами среды разработки. Приложение должно содержать не менее 12 экранных форм, имеющих связь между собой. Одна из форм должна работать с отображением графических данных (файлов изображений).

Минимальный список форм и элементов на них:

1. Главная страница (гость) – стартовая форма. Содержит элементы:

Кнопки:

- поиск (ищет автомобили по названию)
- войти (переход на форму “Вход”)
- регистрация
- закрыть приложение

Изображения:

- шапка

2. Главная страница (авторизованный пользователь). Содержит элементы:

Кнопки:

- поиск авто
- избранные объявления
- выйти из аккаунта
- фильтры по популярным авто
- добавить объявление
- мои объявления
- загрузить объявления

Изображения:

- шапка

3. Главная страница (администратор). Содержит элементы:

Кнопки:

- просмотр объявлений
- статистика
- выйти из аккаунта
- фильтры по популярным авто
- загрузить объявления
- мои объявления

Изображения:

- шапка

4. Авторизация. Содержит элементы:

Редактируемые текстовые поля:

- логин
- пароль (пароль для входа в аккаунт)

Кнопки:

- войти (переход на форму “Главная страница” с определенным типом пользователя)
- назад (переход на предыдущую форму)

5. Регистрация. Содержит элементы:

Редактируемые текстовые поля:

- имя пользователя

- номер телефона
- логин
- пароль (пароль для входа в аккаунт)
- повторить пароль (контрольный ввод пароля)
- пароль для админа

Кнопки:

- зарегистрироваться (переход на форму “Главная страница” с определенным типом пользователя)
- назад (переход на предыдущую форму)

6. Список объявлений (гость). Содержит элементы:

Текстовые поля x5:

- название автомобиля, модель
- имя продавца
- описание автомобиля
- цена
- прочие характеристики

Изображения:

- изображение машины

7. Список объявлений (авторизованный пользователь).

Текстовые поля x5:

- название автомобиля, модель
- имя продавца
- описание автомобиля
- цена
- прочие характеристики

Check box:

- добавить в избранное

Изображения:

- изображение машины

Изображения:

- фото машины

8. Список объявлений (администратор).

Тоже самое, что и авторизованный пользователь, но может удалять любые объявления

9. Редактирование объявления (администратор, авторизованный пользователь).

Редактируемые текстовые поля:

- цена
- название автомобиля

- описание автомобиля

- прочие параметры

Кнопки:

- изменить фото

- завершить правки (сохранить корректировки)

Изображения:

- изображения автомобиля

10. Мои объявления (авторизованный пользователь /администратор).

Всё тоже самое что при простом просмотре объявлений у гостя +

Содержит элементы:

Кнопки:

- удалить

- снять с публикации

- редактировать

11. Отправка на печать. Содержит элементы:

Текстовые поля:

- комментарий

Кнопки:

- отправить

12. Статистика

Текстовые поля:

- Количество проданных автомобилей
- календарь
- progress bar

13. Популярные марки (для всех типов пользователей)

Label: марка выбранного авто

Текстовые поля x5:

- название автомобиля, модель
- имя продавца
- описание автомобиля
- цена
- прочие характеристики

Изображения:

- изображение машины

ПОЛЬЗОВАТЕЛИ СИСТЕМЫ

В системе будет реализовано три типа пользователей: администратор, зарегистрированный пользователь и незарегистрированный пользователь («гость»). Каждый пользователь будет обладать своими смысловыми функциями и возможностями.

Гость: просматривать объявления

Зарегистрированный пользователь: просматривать, добавлять, редактировать и удалять объявления. Удалять и редактировать он может только свои объявления.

Администратор: просматривать, добавлять, редактировать и удалять объявления. Может удалять любые объявления.

3.2 Постановка задачи

В ходе анализа требований была сформулирована задача: разработать автоматизированную информационную систему по продаже автомобилей, в которой необходимо:

- а) - разграничить доступ для администратора и простого пользователя;
- б) - предусмотреть авторизацию администратора с возможностью удаления любых объявления;
- в) - дать возможность администратору и авторизованному пользователю добавлять и просматривать объявления;
- г) - Пользователю дать возможность редактировать свои объявления;
- д) - Позволить авторизованному пользователю выводить данные об объявлениях в формате csv [1].

3.3 Инструментарий

Для разработки и оформления отчёта были использованы:

- Microsoft Visual Studio 2019 – продукт компании Microsoft, включающий интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Для создания ПО выбран объектно-ориентированный язык C#;
- Microsoft SQL Server Management Studio 2018 – СУБД
- Paint.net – графический редактор
- Microsoft Word – текстовый редактор

3.4 Проектирование модели БД

После анализа требований и изучения работы с базой данных была спроектирована модель базы данных см. Рисунок 2 [3].

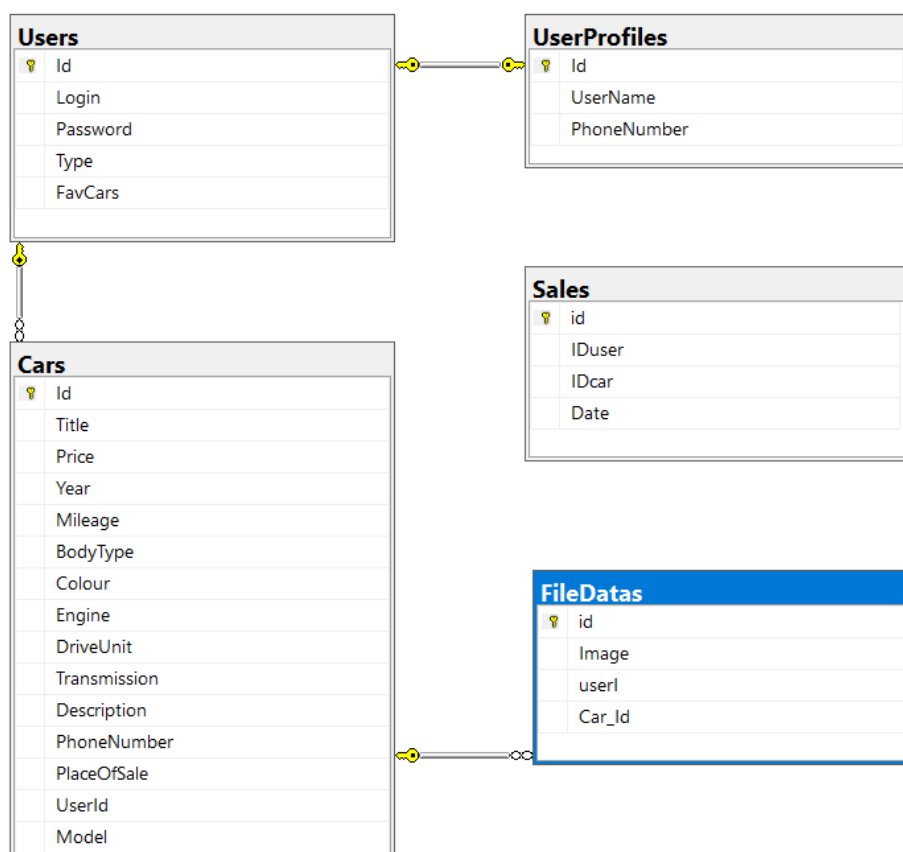


Рисунок 2– Модель связей таблиц в БД

С помощью этой модели были выявлены ключевые поля и обозначены связи между таблицами и их полями. Совместно с моделью БД были определены типы и назначение полей и таблиц (таблица 2) [1].

Таблица 1 – Описание таблиц и их содержания

Таблица	Имя поля	Тип данных	Описание
	Id	Счётчик	Ключевое поле, идентификатор объявления
	Title	Короткий текст	Поле, содержащее марку машины

Cars	Price	Короткий текст	Поле, содержащее цену
	Year	Короткий текст	Поле, содержащее год
	Mileage	Короткий текст	Поле, содержащее пробег
	BodyType	Короткий текст	Поле, содержащее тип кузова
	Colour	Короткий текст	Поле, содержащее цвет
	Engine	Короткий текст	Поле, содержащее тип двигателя
	DriveUnit	Короткий текст	Поле, содержащее тип привода
	Transmission	Короткий текст	Поле, содержащее тип коробки
	Description	Короткий текст	Поле, содержащее краткое описание
	PhoneNumber	Короткий текст	Поле, содержащее номер телефона
	PlaceOfSale	Короткий текст	Поле, содержащее место продажи
	UserId	Счётчик	Поле, содержащее Id владельца
Users	Model	Короткий текст	Поле, содержащее модель
	Id	Счётчик	Ключевое поле, идентификатор участника
	Login	Короткий текст	Поле, содержащее логин
	Password	Короткий текст	Поле, содержащее пароль
	Type	Короткий текст	Поле, содержащее тип пользователя
UserProfiles	FavCars	Короткий текст	Поле, содержащее список выбранных авто
	Id	Счётчик	Ключевое поле, идентификатор участника
	UserName	Короткий текст	Поле, содержащее имя пользователя
	Phonenumber	Короткий текст	Поле, содержащее номер телефона
	Id	Счётчик	Ключевое поле, идентификатор продажи

Sales	IDuser	Короткий текст	Поле, содержащее Id продавца
	IDcar	Короткий текст	Поле, содержащее Id проданного авто
	Date	Короткий текст	Поле, содержащее дату продажи
FileDatas	Id	Счётчик	Ключевое поле, идентификатор картинки
	Image	Массив байт	Поле, содержащее последовательность байт
	userI	Целое число	Поле, содержащее Id объявления

3.5 Интерфейс

При запуске программы открывается главная страница гостя. Гость может зарегистрироваться, войти, посмотреть список объявлений и просмотреть объявления с популярными марками см. Рисунок 3.

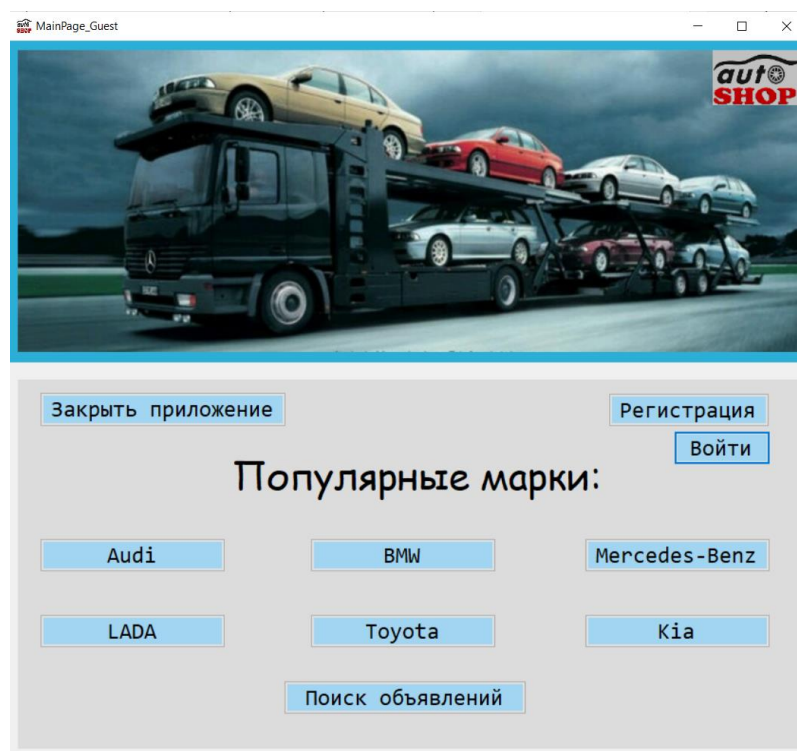


Рисунок 3 - Главная страница гостя

При авторизации необходимо ввести логин и пароль, при несовпадении данных выводятся сообщения о некорректном вводе данных см. Рисунок 4 [0].

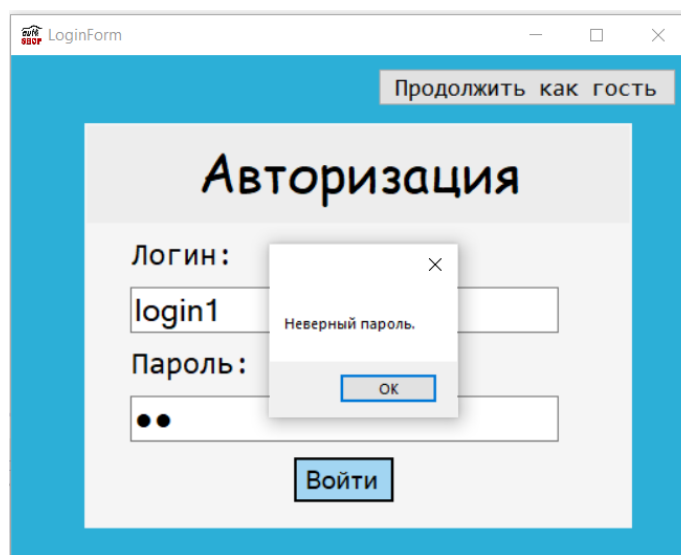


Рисунок 4 - Форма авторизация

При регистрации идёт проверка на существование логина и правильный ввод паролей, при некорректном вводе выводятся сообщения. Также если гость не хочет регистрироваться, то он может вернуться к предыдущей странице. Если регистрируется админ, то он должен ввести пароль для администратора в специальном окне см. Рисунок 5.

The image shows a web application window titled "RegistrationForm". At the top right, there is a button labeled "Продолжить как гость". The main content area is titled "Регистрация" and contains the following fields and elements:

- Имя:** A text input field containing "Никита".
- Номер телефона:** A text input field containing "+789124986".
- Логин:** A text input field containing "login1".
- Пароль:** A password input field with five dots, partially obscured by a modal dialog.
- Повторите пароль:** A password input field with five dots.
- Зарегистрироваться:** A blue button at the bottom of the registration form.
- Пароль для Администратора:** A text input field at the bottom of the window.

A modal dialog box is open over the password field, displaying the message: "Пароли не совпадают, введите заново" (Passwords do not match, enter again). It has an "ОК" button.

Рисунок 5 - Форма регистрации

При успешной регистрации, пользователю открывается ещё ряд возможностей. Пользователь может отправлять на печать краткую информацию обо всех объявлениях, добавлять объявления, просматривать свои объявления и все остальные возможности гостя см. Рисунок 6.

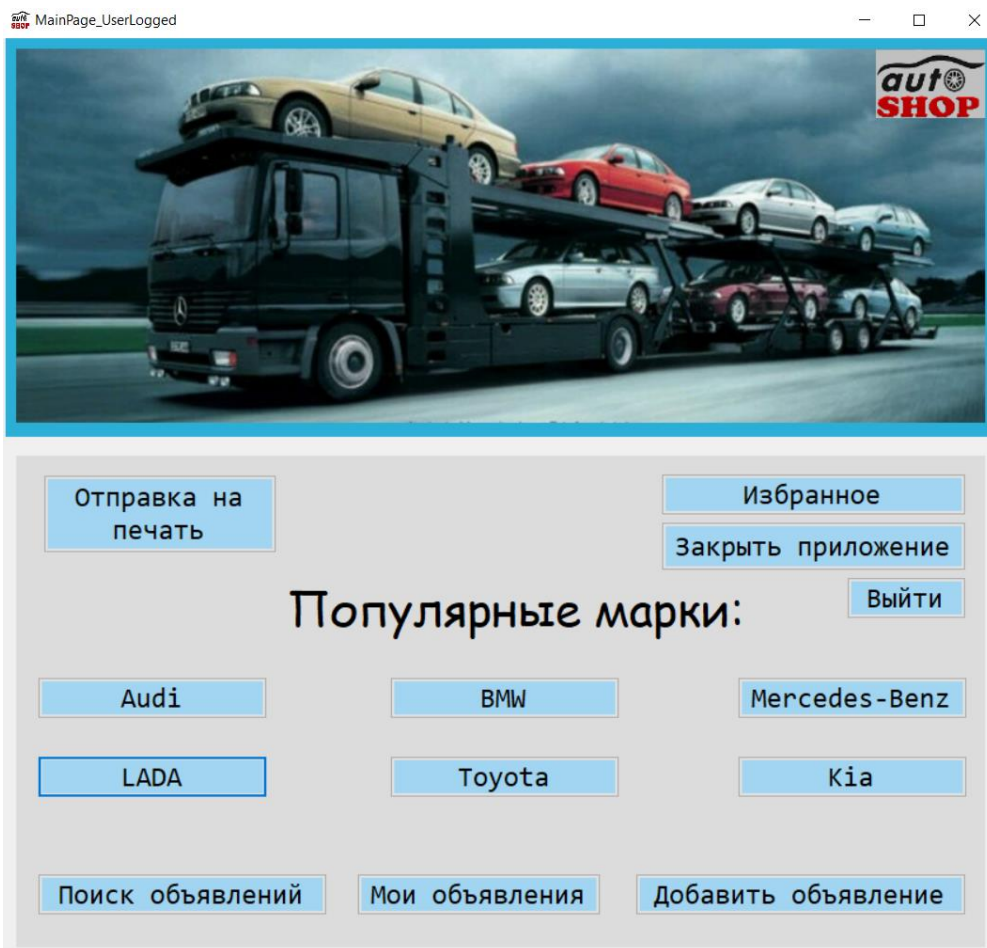


Рисунок 6 - Главная страница вошедшего пользователя

Авторизованный пользователь может просматривать свои объявления, редактировать их, удалять и снимать с публикации если продал Рисунок 7.

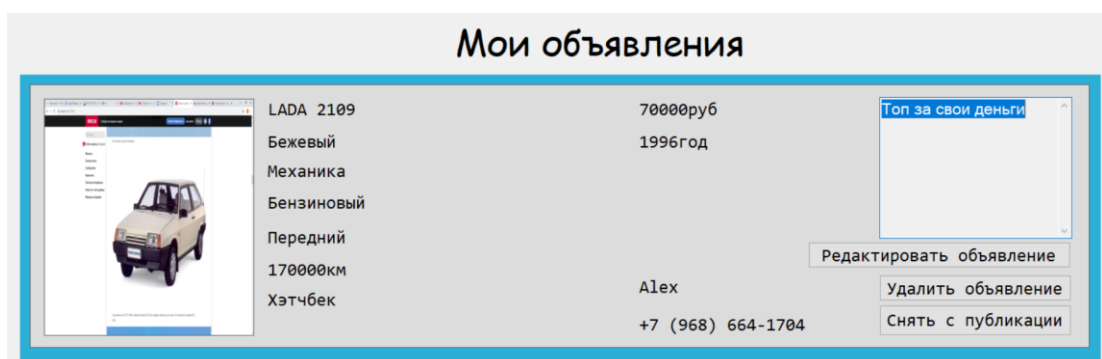


Рисунок 7 - Форма с объявлениями пользователя

Также авторизованный пользователь может добавлять объявления. Ему необходимо как можно подробнее описать характеристики авто. Для этого нужно заполнить все поля см. Рисунок 8.

ФillingAd

Добавление объявления

Марка:

Цена (Руб):

Год выпуска:

Пробег (км):

Тип кузова:

Цвет:

Тип двигателя:

Фото автомобиля:

Модель:

Тип привода:

Коробка:

Описание:

Номер телефона:

Место продажи:

Рисунок 8 - Форма добавления объявлений

При нажатии на кнопку “Отправка” появляется форма для сохранения данных. В дальнейшем он может распечатать эту краткую информацию о всех продаваемых автомобилях см. Рисунок 9 [2].

FormSendDetail

Отправка

Выберите название файла,
в него будет загружена короткая
информация о всех объявлениях.

Комментарий:

Отправить

Рисунок 9 - Форма отправки на печать

После отправки, можно посмотреть сохранённый файл см. Рисунок 10

	A	B	C	D	E	F	G
1	Марка	Модель	Номер те	Место при	Цена	Тип двигателя	
2	Lada	2109	9,88E+08	Таганрог	70000	Бензиновый	
3	Mercedes-Benz						
4	Nissan						
5	Volkswagen						
6	Volkswagen						
7	Volkswagen						
8	Nissan						
9	Nissan						
10							
11	Lada					Бензиновый	

Рисунок 10 - Пример полученного документа для печати

Главная страница администратора не сильно отличается от других пользователей, добавлена форма для просмотра статистики продаж, также при просмотре всех объявлений он может удалить любое см. Рисунок 11.

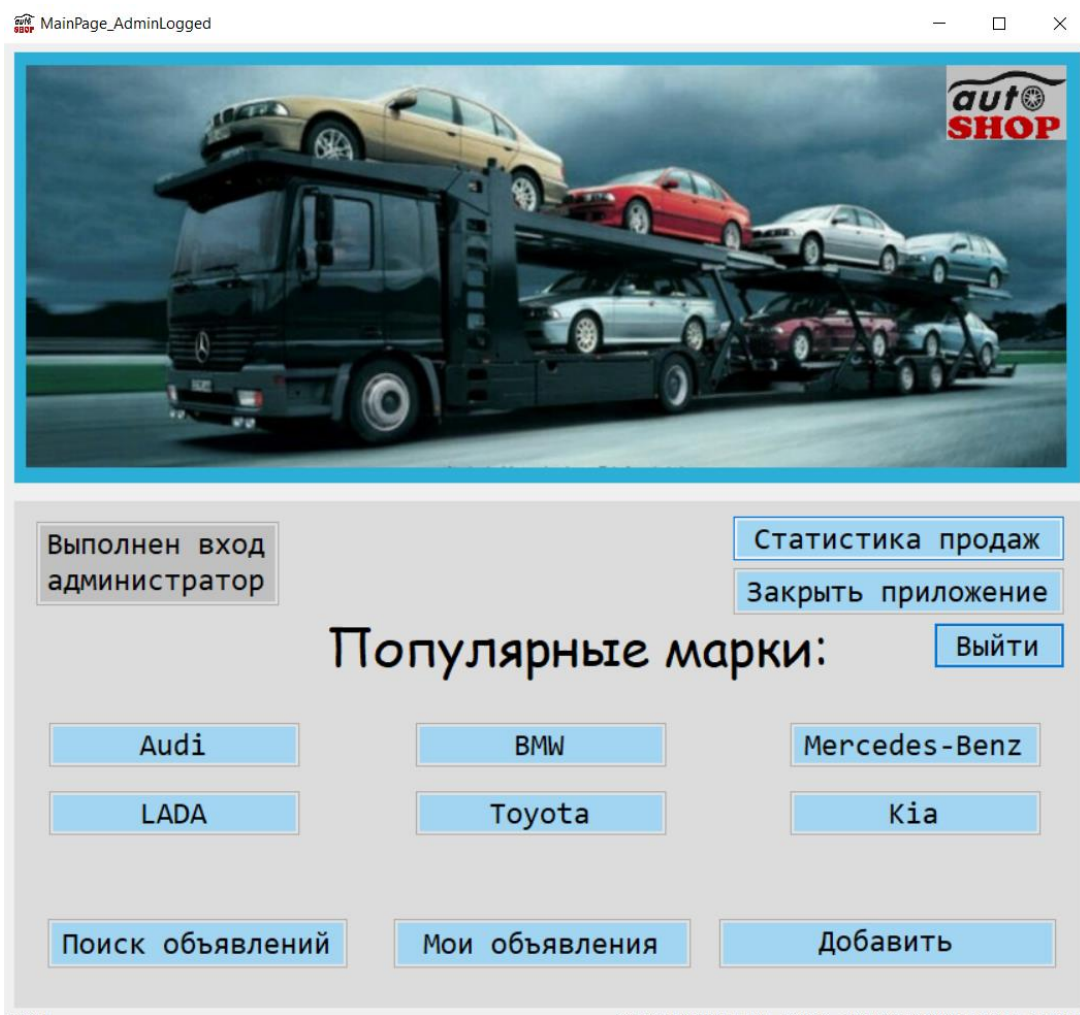


Рисунок 11 - Главная страница администратора

В псевдо-форме по статистике продаж, отображается progress bar, который обнуляется каждый раз после продажи 10 автомобилей, также присутствуют два типа календарей см. Рисунок 12.

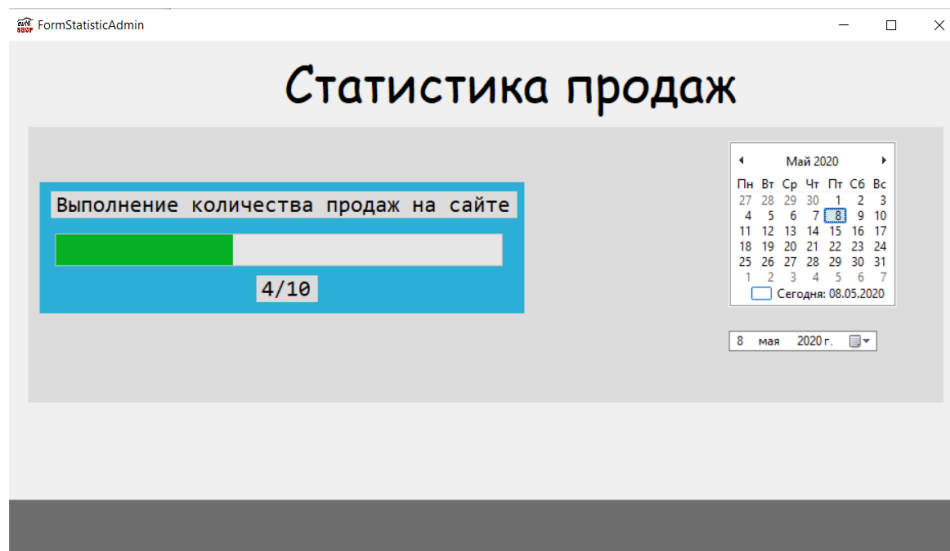


Рисунок 12 - Форма статистики продаж

В данном разделе представлены не все существующие формы, а лишь те, которые наиболее значимы.

3.5.1 Тестирование

В ходе разработки ПО был использован метод ручного тестирования. Был проверена корректная работоспособность системы. Критичных багов не выявлено.

4 Фрагменты кода программы

Код для авторизации пользователя:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CarSalesSystem
{
    public partial class LoginForm : Form
    {
        MainPage_Guest PreForm;
        public LoginForm(MainPage_Guest BackForm)
        {
            PreForm = BackForm;
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Флаг нахождения такого логина в базе
            bool LoginRight = false;
            var NeededUser = new User();

            // Сверяем с данными из БД
            using (var context = new MyDbContext())
            {
                var UserFind = context.Users.ToList();
                foreach (User userF in UserFind)
                {
                    // Если встречаем такой логин в базе, то сверяем пароль
                    if (userF.Login.Equals(LoginText.Text))
                    {
                        LoginRight = true;
                        NeededUser = userF;
                    }
                }
            }
            if (!LoginRight)
                MessageBox.Show("Неверный логин.");
            else
            {
                if (PasswordText.Text == NeededUser.Password)
                {
                    if (NeededUser.Type == "admin")
                    {
                        // Переход на форму админа
                        MainPage_AdminLogged mainFormAdmin = new MainPage_AdminLogged(NeededUser);
                        mainFormAdmin.Show();
                    }
                    else
                    {
                        // Переход на форму для авторизованного пользователя
                        MainPage_UserLogged mainFormUser = new MainPage_UserLogged(NeededUser);
                        int a = NeededUser.Id;
                        mainFormUser.Show();
                    }
                }
            }
        }
    }
}
```



```

        this.Hide(); // Скрываем эту форму
    }

    }
    else
        MessageBox.Show("Неверный пароль.");
    }
}
// Окно входа скрывается по завершению работы с ней
}

private void ContinueAsGuest_Click(object sender, EventArgs e)
{
    // Переход на форму главного экрана у гостя
    this.Close();
    PreForm.Show();
}
}
}

```

Код для регистрации новых пользователей

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CarSalesSystem
{
    public partial class RegistrationForm : Form
    {
        MainPage_Guest PreForm;
        public RegistrationForm(MainPage_Guest BackForm)
        {
            PreForm = BackForm;
            InitializeComponent();
        }

        private void buttonRegistration_Click(object sender, EventArgs e)
        {
            bool LoginRight = false; // Флаг для проверки уникальности Login
            bool PasswordMatch = false; // Флаг для проверки совпадения паролей

            LoginRight = false;
            PasswordMatch = false;

            // Проверка на совпадение двух паролей
            if (PasswordText.Text == RePasswordText.Text)
                PasswordMatch = true;
            else
                MessageBox.Show("Пароли не совпадают,\nпведите заново");
            // Поиск в базе пользователя с таким же Login (Этот параметр должен быть уникальным для каждого
            пользователя)

            bool ExistUsers = false;
            using (var context = new MyDbContext())
            {
                var UserFind = context.Users.ToList();
                foreach (User userF in UserFind)

```

```

    {
        ExistUsers = true;
        // Если встречаем такой логин в базе, то сразу выходим из цикла
        if (userF.Login.Equals(LoginText.Text))
        {
            LoginRight = false;
            break;
        }
        else
            LoginRight = true;
    }
}

// Различаем обычного пользователя от админа
bool FlagContinue = false;
string PasswordAdmin = "1111";

bool FlagAdminPassword = false;
if (PasswordForAdmin.Text != "" && PasswordForAdmin.Text != PasswordAdmin)
{
    MessageBox.Show("Неверный пароль\ндля администратора");
    FlagContinue = false;
}
else
    FlagContinue = true;

if (PasswordForAdmin.Text == PasswordAdmin)
{
    FlagContinue = true;
    FlagAdminPassword = true;
}

// Если пользователей в базе пока нет, то первый зарегистрировавшийся может выбрать любой логин
if (!ExistUsers)
    LoginRight = true;

if (LoginRight == false)
    MessageBox.Show("Данный логин уже занят,\нзамените его");

if (LoginRight && PasswordMatch && FlagContinue)
{
    User CurrUser;
    using (var context = new MyDbContext())
    {
        User user1 = new User();
        UserProfile profile1 = new UserProfile();

        user1.Login = LoginText.Text;
        user1.Password = PasswordText.Text;

        profile1.UserName = NameText.Text;
        profile1.PhoneNumber = NumberText.Text;

        if (FlagAdminPassword)
            user1.Type = "admin";

        context.Users.Add(user1);
        context.UserProfiles.Add(profile1);
        context.SaveChanges();

        CurrUser = user1;
    }
    if (!FlagAdminPassword)
    {
        // Переход на форму для авторизованного пользователя
    }
}

```

```

        //this.Hide();
        MainPage_UserLogged mainFormUser = new MainPage_UserLogged(CurrUser);
        mainFormUser.Show();
    }
    else
    {
        // Переход на форму для админа
        MainPage_AdminLogged mainFormAdmin = new MainPage_AdminLogged(CurrUser);
        mainFormAdmin.Show();
    }
}

private void ContinueAsGuest_Click(object sender, EventArgs e)
{
    // Переход на форму главного экрана у гостя
    this.Close();
    PreForm.Show();
}

}
}

```

Код для вывода объявлений

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CarSalesSystem
{
    public partial class ListOfAds_Guest : Form
    {
        // Счётчик для контроля порядка объявлений
        byte[] imageB;
        int k = 0;
        public ListOfAds_Guest()
        {
            InitializeComponent();

            // Изначально все формы невидимы, если хоть одно поле в форме заполняем, то она становится видимой
            using (var context = new MyDbContext())
            {
                Label[] L = { labelModel, labelModel1, labelModel2, labelModel3, labelModel4 };
                Label[] Col = { labelColour, labelColour1, labelColour2, labelColour3, labelColour4 };
                Label[] Trans = { labelTransmission, labelTransmission1, labelTransmission2, labelTransmission3,
labelTransmission4 };
                Label[] Eng = { labelEngine, labelEngine1, labelEngine2, labelEngine3, labelEngine4 };
                Label[] DrUnit = { LabelDriveUnit, LabelDriveUnit1, LabelDriveUnit2, LabelDriveUnit3, LabelDriveUnit4
};

                Label[] Mil = { labelMileage, labelMileage1, labelMileage2, labelMileage3, labelMileage4 };
                Label[] bType = { labelBodyType, labelBodyType1, labelBodyType2, labelBodyType3, labelBodyType4
};

                Label[] P = { labelPrice, labelPrice1, labelPrice2, labelPrice3, labelPrice4 };
                Label[] Y = { labelYear, labelYear1, labelYear2, labelYear3, labelYear4 };
                Label[] NumPhone = { labeNumberPhone, labeNumberPhone1, labeNumberPhone2, labeNumberPhone3,
labeNumberPhone4 };
            }
        }
    }
}

```

```

Label[] Name = { labelNameUser, labelNameUser1, labelNameUser2, labelNameUser3, labelNameUser4
};

TextBox[] textBoxeDescription = { tBDesc, tBDesc1, tBDesc2, tBDesc3, tBDesc4 };
Panel[] Pan = { panelAd, panelAd1, panelAd2, panelAd3, panelAd4 };
PictureBox[] PB = { pictureBoxCar, pictureBoxCar1, pictureBoxCar2, pictureBoxCar3, pictureBoxCar4 };

// Изначально, все панели невидимые
for (int i = 0; i < 5; i++)
    Pan[i].Visible = false;

var CurrentCar = context.Cars.ToList();
int tempK = k;
foreach (Car CarAd in CurrentCar)
{
    if (k < tempK + 5)
    {
        //Показываем изображение
        PB[k % 5].BorderStyle = BorderStyle.FixedSingle;
        PB[k % 5].SizeMode = PictureBoxSizeMode.StretchImage;

        var F = context.Files.ToList();
        foreach (FileData F1 in F)
        {
            if (F1.userI == CarAd.Id)
            {
                imageB = F1.Image; // Выводим фото для текущего элемента
                System.IO.MemoryStream memoryStream1 = new System.IO.MemoryStream();
                foreach (byte b1 in imageB) memoryStream1.WriteByte(b1);
                Image image = Image.FromStream(memoryStream1);
                PB[k % 5].Image = image;
                break;
            }
        }

        Pan[k % 5].Visible = true;

        L[k % 5].Text = CarAd.Title + " " + CarAd.Model;
        Col[k % 5].Text = CarAd.Colour;
        Trans[k % 5].Text = CarAd.Transmission;
        Eng[k % 5].Text = CarAd.Engine;
        DrUnit[k % 5].Text = CarAd.DriveUnit;
        Mil[k % 5].Text = CarAd.Mileage + "км";
        bType[k % 5].Text = CarAd.BodyType;
        P[k % 5].Text = CarAd.Price + "руб";
        Y[k % 5].Text = CarAd.Year + "год";
        NumPhone[k % 5].Text = CarAd.PhoneNumber;

        var CurrentSeller = context.UserProfiles.ToList();
        int tempS = 0;
        foreach (UserProfile CurS in CurrentSeller)
        {
            tempS++;
            if (tempS == CarAd.UserId)
                Name[k % 5].Text = CurS.UserName;
        }
        textBoxeDescription[k % 5].Text = CarAd.Description;
        textBoxeDescription[k % 5].ReadOnly = true;
    }
    else
        break;
    k++;
}

if (k < CurrentCar.Count)
    buttonNext.Visible = true;

```

```

else
    buttonNext.Visible = false;

if (k > 5)
    buttonBack.Visible = true;
else
    buttonBack.Visible = false;
}
}

// Если есть ещё объявления, то кнопка становится видимой
private void buttonNext_Click_1(object sender, EventArgs e)
{
    // Изначально все формы невидимы, если хоть одно поле в форме заполняем, то она становится видимой
    using (var context = new MyDbContext())
    {
        Label[] L = { labelModel, labelModel1, labelModel2, labelModel3, labelModel4 };
        Label[] Col = { labelColour, labelColour1, labelColour2, labelColour3, labelColour4 };
        Label[] Trans = { labelTransmission, labelTransmission1, labelTransmission2, labelTransmission3,
labelTransmission4 };
        Label[] Eng = { labelEngine, labelEngine1, labelEngine2, labelEngine3, labelEngine4 };
        Label[] DrUnit = { LabelDriveUnit, LabelDriveUnit1, LabelDriveUnit2, LabelDriveUnit3, LabelDriveUnit4
};

        Label[] Mil = { labelMileage, labelMileage1, labelMileage2, labelMileage3, labelMileage4 };
        Label[] bType = { labelBodyType, labelBodyType1, labelBodyType2, labelBodyType3, labelBodyType4
};

        Label[] P = { labelPrice, labelPrice1, labelPrice2, labelPrice3, labelPrice4 };
        Label[] Y = { labelYear, labelYear1, labelYear2, labelYear3, labelYear4 };
        Label[] NumPhone = { labelNumberPhone, labelNumberPhone1, labelNumberPhone2, labelNumberPhone3,
labelNumberPhone4 };
        Label[] Name = { labelNameUser, labelNameUser1, labelNameUser2, labelNameUser3, labelNameUser4
};

        TextBox[] textBoxDescription = { tBDesc, tBDesc1, tBDesc2, tBDesc3, tBDesc4 };
        Panel[] Pan = { panelAd, panelAd1, panelAd2, panelAd3, panelAd4 };
        PictureBox[] PB = { pictureBoxCar, pictureBoxCar1, pictureBoxCar2, pictureBoxCar3, pictureBoxCar4 };

        // Изначально, все панели невидимые
        for (int i = 0; i < 5; i++)
            Pan[i].Visible = false;

        var CurrentCar = context.Cars.ToList();
        int StartK = 0;
        int tempK = k;
        foreach (Car CarAd in CurrentCar)
        {
            if (StartK >= k)
            {
                if (k < tempK + 5)
                {
                    PB[k % 5].BorderStyle = BorderStyle.FixedSingle;
                    PB[k % 5].SizeMode = PictureBoxSizeMode.StretchImage;

                    //Показываем изображение
                    var F = context.Files.ToList();
                    foreach (FileData F1 in F)
                    {
                        if (F1.userI == CarAd.Id)
                        {
                            imageB = F1.Image; // Выводим фото для текущего элемента
                            System.IO.MemoryStream memoryStream1 = new System.IO.MemoryStream();
                            foreach (byte b1 in imageB) memoryStream1.WriteByte(b1);
                            Image image = Image.FromStream(memoryStream1);
                            PB[k % 5].Image = image;
                            break;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }

    Pan[k % 5].Visible = true;

    L[k % 5].Text = CarAd.Title + " " + CarAd.Model;
    Col[k % 5].Text = CarAd.Colour;
    Trans[k % 5].Text = CarAd.Transmission;
    Eng[k % 5].Text = CarAd.Engine;
    DrUnit[k % 5].Text = CarAd.DriveUnit;
    Mil[k % 5].Text = CarAd.Mileage + "км";
    bType[k % 5].Text = CarAd.BodyType;
    P[k % 5].Text = CarAd.Price + "руб";
    Y[k % 5].Text = CarAd.Year + "год";
    NumPhone[k % 5].Text = CarAd.PhoneNumber;

    var CurrentSeller = context.UserProfiles.ToList();
    int tempS = 0;
    foreach (UserProfile CurS in CurrentSeller)
    {
        tempS++;
        if (tempS == CarAd.UserId)
            Name[k % 5].Text = CurS.UserName;
    }

    textBoxDescription[k % 5].Text = CarAd.Description;
    textBoxDescription[k % 5].ReadOnly = true;

    }
    else
        break;
    k++;
}
StartK++;
}

if (k < CurrentCar.Count)
    buttonNext.Visible = true;
else
    buttonNext.Visible = false;

if (k > 5)
    buttonBack.Visible = true;
else
    buttonBack.Visible = false;
}
}

private void buttonBack_Click_1(object sender, EventArgs e)
{
    // Изначально все формы невидимы, если хоть одно поле в форме заполняем, то она становится видимой
    using (var context = new MyDbContext())
    {
        Label[] L = { labelModel, labelModel1, labelModel2, labelModel3, labelModel4 };
        Label[] Col = { labelColour, labelColour1, labelColour2, labelColour3, labelColour4 };
        Label[] Trans = { labelTransmission, labelTransmission1, labelTransmission2, labelTransmission3,
labelTransmission4 };
        Label[] Eng = { labelEngine, labelEngine1, labelEngine2, labelEngine3, labelEngine4 };
        Label[] DrUnit = { LabelDriveUnit, LabelDriveUnit1, LabelDriveUnit2, LabelDriveUnit3, LabelDriveUnit4
};

        Label[] Mil = { labelMileage, labelMileage1, labelMileage2, labelMileage3, labelMileage4 };
        Label[] bType = { labelBodyType, labelBodyType1, labelBodyType2, labelBodyType3, labelBodyType4
};

        Label[] P = { labelPrice, labelPrice1, labelPrice2, labelPrice3, labelPrice4 };
        Label[] Y = { labelYear, labelYear1, labelYear2, labelYear3, labelYear4 };
    }
}

```

```

Label[] NumPhone = { labeNumberPhone, labeNumberPhone1, labeNumberPhone2, labeNumberPhone3,
labeNumberPhone4 };
Label[] Name = { labelNameUser, labelNameUser1, labelNameUser2, labelNameUser3, labelNameUser4
};

TextBox[] textBoxeDescription = { tBDesc, tBDesc1, tBDesc2, tBDesc3, tBDesc4 };
Panel[] Pan = { panelAd, panelAd1, panelAd2, panelAd3, panelAd4 };
PictureBox[] PB = { pictureBoxCar, pictureBoxCar1, pictureBoxCar2, pictureBoxCar3, pictureBoxCar4 };

// Изначально, все панели невидимые
for (int i = 0; i < 5; i++)
    Pan[i].Visible = false;

var CurrentCar = context.Cars.ToList();
int StartK = 0;

int TempAddK = 0;
if (k % 5 != 0)
    TempAddK = k % 5;
else
    if (k > 5)
        TempAddK = 5;

k = k - 5 - TempAddK;
int tempK = k;
foreach (Car CarAd in CurrentCar)
{
    if (StartK >= k)
    {
        if (k < tempK + 5)
        {
            //Показываем изображение
            PB[k % 5].BorderStyle = BorderStyle.FixedSingle;
            PB[k % 5].SizeMode = PictureBoxSizeMode.StretchImage;

            var F = context.Files.ToList();
            foreach (FileData F1 in F)
            {
                if (F1.userI == CarAd.Id)
                {
                    imageB = F1.Image; // Выводим фото для текущего элменета
                    System.IO.MemoryStream memoryStream1 = new System.IO.MemoryStream();
                    foreach (byte b1 in imageB) memoryStream1.WriteByte(b1);
                    Image image = Image.FromStream(memoryStream1);
                    PB[k % 5].Image = image;
                    break;
                }
            }

            Pan[k % 5].Visible = true;

            L[k % 5].Text = CarAd.Title + " " + CarAd.Model;
            Col[k % 5].Text = CarAd.Colour;
            Trans[k % 5].Text = CarAd.Transmission;
            Eng[k % 5].Text = CarAd.Engine;
            DrUnit[k % 5].Text = CarAd.DriveUnit;
            Mil[k % 5].Text = CarAd.Mileage + "км";
            bType[k % 5].Text = CarAd.BodyType;
            P[k % 5].Text = CarAd.Price + "руб";
            Y[k % 5].Text = CarAd.Year + "год";
            NumPhone[k % 5].Text = CarAd.PhoneNumber;

            var CurrentSeller = context.UserProfiles.ToList();
            int tempS = 0;
            foreach (UserProfile CurS in CurrentSeller)
            {
                tempS++;
            }
        }
    }
}

```

```

        if (tempS == CarAd.UserId)
            Name[k % 5].Text = CurS.UserName;
        }

        textBoxDescription[k % 5].Text = CarAd.Description;
        textBoxDescription[k % 5].ReadOnly = true;

    }
    else
        break;
    k++;
}
StartK++;
}

if (k < CurrentCar.Count)
    buttonNext.Visible = true;
else
    buttonNext.Visible = false;

if (k > 5)
    buttonBack.Visible = true;
else
    buttonBack.Visible = false;
}
}

private void ListOfAds_Guest_Load(object sender, EventArgs e)
{
}
}
}
}

```

Реализация всех кнопок для главной страницы авторизованного пользователя

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CarSalesSystem
{
    public partial class MainPage_UserLogged : Form
    {
        // Передаём данные авторизованного пользователя в форму для создания объявления
        User currentUser;
        public MainPage_UserLogged(User NeededUser)
        {
            currentUser = NeededUser;
            InitializeComponent();

            private void AddAnnouncement_Click(object sender, EventArgs e)
            {
                //int a = currentUser.Id;
                FillingAd NewAd = new FillingAd(currentUser);
                NewAd.Show();
            }
        }
    }
}

```



```

// Разлогиниться
private void ExiteUserLogged_Click(object sender, EventArgs e)
{
    this.Hide();
    MainPage_Guest ReturnGuest = new MainPage_Guest();
    ReturnGuest.Show();
}

private void MyAds_Click(object sender, EventArgs e)
{
    FormMyAds MyAllAds = new FormMyAds(CurrentUser);
    MyAllAds.Show();
}

private void buttonFindAds_Click(object sender, EventArgs e)
{
    ListOfAds_UserLogged Ads = new ListOfAds_UserLogged(CurrentUser);
    Ads.Show();
}

private void buttonFavorites_Click(object sender, EventArgs e)
{
    FormFavCars Ads = new FormFavCars(CurrentUser);
    Ads.Show();
}

private void buttonSend_Click(object sender, EventArgs e)
{
    FormSendDetail Ads = new FormSendDetail();
    Ads.Show();
}

private void buttonCLOse_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void buttonAudi_Click(object sender, EventArgs e)
{
    ListOfAdsFiltration Ads = new ListOfAdsFiltration("Audi");
    Ads.Show();
}

private void buttonBMW_Click(object sender, EventArgs e)
{
    ListOfAdsFiltration Ads = new ListOfAdsFiltration("BMW");
    Ads.Show();
}

private void buttonMercedes_Click(object sender, EventArgs e)
{
    ListOfAdsFiltration Ads = new ListOfAdsFiltration("Mercedes");
    Ads.Show();
}

private void buttonKia_Click(object sender, EventArgs e)
{
    ListOfAdsFiltration Ads = new ListOfAdsFiltration("Kia");
    Ads.Show();
}

private void buttonToyota_Click(object sender, EventArgs e)
{
    ListOfAdsFiltration Ads = new ListOfAdsFiltration("Toyota");
}

```

```
        Ads.Show();
    }

    private void buttonLADA_Click(object sender, EventArgs e)
    {
        ListOfAdsFiltration Ads = new ListOfAdsFiltration("LADA");
        Ads.Show();
    }

    private void buttonCLose_Click_1(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
```

ЗАКЛЮЧЕНИЕ

В ходе прохождения учебной практики мной были достигнуты поставленные цели и реализованы задачи в полном объёме. Задание разработать автоматизированную информационную систему по продаже автомобилей было выполнено.

Я расширил и закрепил полученные теоретические знания через получение первичных профессиональных навыков, ознакомился с характером и спецификой будущей деятельности, овладел функционалом, инструментарием различных средств, поддерживающих создание программного обеспечения для решения практических задач, и навыками подготовки презентации и оформления научно-технических отчетов по результатам выполненной работы.

Знания, умения и навыки, приобретённые за период практики, явились отличным стимулом для активной работы в освоении профессии, что позволило получить первый профессиональный опыт работы и сформировать общие представления о специфике своей специальности.

Конечно, созданная АИС не идеальна, её можно расширять, модернизировать и искать лучшие приёмы реализации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Работа с файлами в С#. Классы StreamReader и StreamWriter [Электронный ресурс].- Режим доступа: http://mycsharp.ru/post/21/2013_06_12_rabota_s_fajlami_v_si-sharp_klassy_streamreader_i_streamwriter.html (15.04.2020).
- 2 Вывод данных в DataGridView из БД на С# [Электронный ресурс]. – Режим доступа: <http://vscode.ru/prog-lessons/vyivod-dannyih-v-datagridview-c-sharp.html> (17.04.2020)
- 3 О.Н. Евсеева, А.Б. Шамшев. Работа с базами данных на языке С#. Технология ADO .NET: учебное пособие [Текст] / О.Н. Евсеева, А.Б. Шамшев – Ульяновск: Изд-во УЛГТУ, 2009. – 170 с