СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc41408472)

[ЗАДАНИЕ 4](#_Toc41408473)

[ЛИСТ НОРМОКОНТРОЛЕРА 5](#_Toc41408474)

[1. РАЗРАБОТКА ПРОЕКТА ПРОТОТИПА ИНФОРМАЦИОНОЙ СИСТЕМЫ 6](#_Toc41408475)

[1.1 ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc41408476)

[1.2 ДЕЙСТВУЮЩИЕ ЛИЦА 6](#_Toc41408477)

[1.3 ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ 9](#_Toc41408478)

[1.4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 15](#_Toc41408479)

[1.5 МАКЕТ ПРОТОТИПА СИСТЕМЫ 17](#_Toc41408480)

[2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ 22](#_Toc41408481)

[2.1 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ КЛИЕНТА 22](#_Toc41408482)

[2.2 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ РИЭЛТОРА 25](#_Toc41408483)

[2.3 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ СЕКРЕТАРЯ 28](#_Toc41408484)

[2.4 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ АДМИНИСТРАТОРА 32](#_Toc41408485)

[ЗАКЛЮЧЕНИЕ 35](#_Toc41408486)

[СПИСОК ЛИТЕРАТУРЫ 36](#_Toc41408487)

# ВВЕДЕНИЕ

Курсовой проект предназначен для обеспечения работы риэлтерского агентства, упрощения процесса поиска недвижимости и администрирования базой данных.

Цель работы - создание прототипа информационной системы для работы риэлтерского агентства .

Задачи работы:

* Изучить предметную область риэлтерских агентств
* Определить функциональные требования к системе
* Построить макеты прототипа системы
* Построить диаграммы вариантов использования, диаграммы классов
* Представить схему базы данных в виде ER-модели
* Разработать и представить прототип информационной системы

# 1. РАЗРАБОТКА ПРОЕКТА ПРОТОТИПА ИНФОРМАЦИОНОЙ СИСТЕМЫ

## 1.1 ИЗУЧЕНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Информационная система риэлтерского агентства позволяет клиентам осуществлять поиск недвижимости выставленной на продажу, а также оставлять заявки для дальнейшей работы с агентством. Системным пользователям доступны расширенные возможности системы. В них входят удобный поиск данных по внутренним базам и редактирование информации в них. Использование информационной системы предусматривает существенное упрощение и ускорение работы риэлтерского агентства.

Похожее решение – «Идеальный вариант: Недвижимость».

Преимущества:

* Просмотр данных на веб-сайте в интернете
* Автоматическая публикация объявлений на сайте «Авито» и других порталах недвижимости
* Поддержка одновременной работы неограниченного числа пользователей
* Мощная и быстрая база данных

## 1.2 ДЕЙСТВУЮЩИЕ ЛИЦА

Будущие пользователи системы:

* риэлторы
* клиенты агентства
* секретари
* администраторы

Клиент может оставить заявку, где укажет свою фамилию, имя, отчество и номер телефона, по которому с ним в дальнейшем свяжется секретарь, а впоследствии при необходимости и риэлтор. Также клиент способен просмотреть информацию о недвижимости, которую предоставляет на продажу риэлтерское агентство.

Риэлтор может просматривать информацию о недвижимости и клиентах. После того, как секретарь передаёт информацию о клиенте, риэлтор связывается с ним для дальнейшей работы. После завершения сделки передаёт информацию секретарю.

Секретарь может просматривать информацию о недвижимости, клиентах, пользователях системы, сделках и вносить изменения в список клиентов. После того как клиент оставил заявку, секретарь связывается с ним по телефону и узнаёт какие услуги требуются клиенту. Если клиент выбирает услуги по покупке недвижимости, то секретарь подбирает подходящего риелтора и передаёт ему всю необходимую информацию о клиенте. После успешного завершения сделки передаёт информацию администратору в соответствии с документами от риэлтора.

Администратор имеет полный доступ к базе данных риэлтерского агентства. На основании предоставленной информации от секретаря он может вносить изменения в список пользователей, клиентов, недвижимости, историю сделок.

Диаграмма ролей и функций изображена на рисунке 1.1.

Рисунок 1.1 – Диаграмма ролей и функций

## 1.3 ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ

Вариант использования «Вход в систему»

Краткое описание:

Данный вариант использования описывает вход секретаря, риэлтора, администратора в систему риэлтерского агентства.

Основной поток событий:

1. Система запрашивает имя пользователя и пароль.
2. Пользователь вводит имя и пароль.
3. Система подтверждает имя и пароль, после чего открывается доступ в систему.

Альтернативные потоки:

Неправильно имя пользователя и/или пароль. Если во время выполнения основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

Вариант использования «Просмотр информации о недвижимости»

Краткое описание:

Данный вариант использования позволяет пользователям и клиенту просмотреть варианты недвижимости, которые предлагает агентство.

Основной поток событий:

1. Пользователь или клиент выбирает параметры недвижимости, которые ему интересны.
2. Система открывает выбранный раздел.
3. Пользователь или клиент просматривает информацию о недвижимости.

Альтернативные потоки:

Если окажется, что установить связь с системой каталога недвижимости невозможно, то будет выдано сообщение об ошибке. После подтверждения данного сообщения, вариант использования завершится.

Вариант использования «Изменение информации о клиентах»

Краткое описание:

Данный вариант использования позволяет изменить информацию о клиентах в базе данных риэлтерского агентства.

Основной поток событий:

1. Система подтверждает процесс изменения информации о клиенте.
2. Пользователь вводит фамилию, имя, отчество и номер телефона.
3. Система проверяет введенные данные.
4. Система изменяет данные клиента в соответствии с введенными данными.

Альтернативные потоки:

В случае введения фамилии, имени, отчества и номера телефона которых нет в базе, выдается ошибка, что клиента с такими данными нет в базе. После подтверждения ошибки вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

Вариант использования «Изменение информации о недвижимости»

Краткое описание:

Данный вариант использования позволяет администратору изменять данные о недвижимости из базы данных риэлтерского агентства.

Основной поток событий:

1. Система подтверждает процесс изменения информации о недвижимости.
2. Администратор выбирает информацию, которую нужно изменить.
3. Администратор изменяет информацию.
4. Система проверяет введенные изменения.
5. Система сохраняет измененные данные.

Альтернативные потоки:

В случае введения информации о недвижимости, которых нет в базе, выдается ошибка, что недвижимости с такими данными нет в базе. После подтверждения ошибки вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования администратор должен войти в систему.

Вариант использования «Изменение информации о пользователях системы»

Краткое описание:

Данный вариант использования позволяет администратору изменить информацию о пользователях системы в базе данных риэлтерского агентства.

Основной поток событий:

1. Система подтверждает процесс изменения информации о пользователях системы.
2. Администратор вводит фамилию, имя, отчество, номер телефона, логин, пароль, роль.
3. Система проверяет введенные данные.
4. Система изменяет данные клиента в соответствии с введенными данными.

Альтернативные потоки:

В случае введения данных, которых нет в базе, выдается ошибка, что пользователя с такими данными нет в базе. После подтверждения ошибки вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

Вариант использования «Просмотр информации о пользователях системы»

Краткое описание:

Данный вариант использования позволяет администратору и секретарю просмотреть информацию о пользователях системы.

Основной поток событий:

1. Администратор или секретарь выбирает параметры, которые ему интересны.
2. Система открывает выбранный раздел.
3. Администратор или секретарь просматривает информацию о недвижимости.

Альтернативные потоки:

Если окажется, что установить связь с системой каталога пользователей невозможно, то будет выдано сообщение об ошибке. После подтверждения данного сообщения, вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

Вариант использования «Просмотр информации о клиентах»

Краткое описание:

Данный вариант использования позволяет пользователю просмотреть информацию о клиентах.

Основной поток событий:

1. Пользователь выбирает параметры, которые ему интересны.
2. Система открывает выбранный раздел.
3. Пользователь просматривает информацию о недвижимости.

Альтернативные потоки:

Если окажется, что установить связь с системой каталога клиентов невозможно, то будет выдано сообщение об ошибке. После подтверждения данного сообщения, вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

Вариант использования «Изменение информации о сделках»

Краткое описание:

Данный вариант использования позволяет администратору изменить информацию о сделках в базе данных риэлтерского агентства.

Основной поток событий:

1. Система подтверждает процесс изменения информации о сделках.
2. Администратор вводит номер клиента, номер пользователя, дата, сумма сделки.
3. Система проверяет введенные данные.
4. Система изменяет информацию о сделках в соответствии с введенными данными.

Альтернативные потоки:

В случае введения данных, которых нет в базе, выдается ошибка, что сделок с такими данными нет в базе. После подтверждения ошибки вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

Вариант использования «Просмотр информации о сделках»

Краткое описание:

Данный вариант использования позволяет администратору и секретарю просмотреть информацию о сделках.

Основной поток событий:

1. Администратор или секретарь выбирает параметры, которые ему интересны.
2. Система открывает выбранный раздел.
3. Администратор или секретарь просматривает информацию о сделках.

Альтернативные потоки:

Если окажется, что установить связь с системой каталога сделок невозможно, то будет выдано сообщение об ошибке. После подтверждения данного сообщения, вариант использования завершится.

Предусловие:

Перед началом выполнения данного варианта использования пользователь должен войти в систему.

Вариант использования «Оставление заявки»

Краткое описание:

Данный вариант использования позволяет клиенту оставить заявку.

Основной поток событий:

1. Система подтверждает процесс оставления заявки.
2. Клиент вводит фамилию, имя, отчество и номер телефона.
3. Система проверяет введенные данные.
4. Система изменяет данные клиента в соответствии с введенными данными и выдает сообщение, что заявка принята.

Альтернативные потоки:

Если окажется, что установить связь с системой каталога клиентов невозможно, то будет выдано сообщение об ошибке. После подтверждения данного сообщения, вариант использования завершится**.**

## 1.4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

База данных состоит из 4 таблиц. Таблицы «Пользователь», «Недвижимость», «Клиент», «Сделка» вместе с их полями изображены на рисунках 1.2 и 1.3.

Таблица «Клиент» хранит фамилию, имя, отчество, номер телефона и статус. Статус определяет положение заявки клиента в системе риэлтерского агентства. Она может быть новой заявкой, заявкой, с которой проводится работа, или заявкой, которую уже обработали. В данной таблице номер клиента является первичным ключом.

Таблица «Недвижимость» хранит адрес, город, площадь, цену, номер телефона, тип недвижимости, статус. Адрес это полный и подробный адрес недвижимости. В поле город указан город, где находится недвижимость. Если недвижимость находится не в городе, то пишут город, который находится поблизости. Площадь – площадь недвижимости в кв. м. Цена указывается в рублях. Тип недвижимости можно выбрать один из представленных на рынке. В их число входят квартира, дом, дача, участок, коммерческая недвижимость и гараж. Статус определяет продана недвижимость или нет. Здесь первичным ключом является номер недвижимости.

Рисунок 1.2 – Диаграмма базы данных

Таблица «Пользователь» содержит фамилию, имя, отчество, номер телефона, роль, логин и пароль. Логин и пароль требуются пользователю для входа в систему, в среду своей работы. Роль определяет уровень доступа в системе. Первичный ключ здесь – номер пользователя.

Таблица «Сделка» является таблицей, где возможно объединить данные других таблиц и создать сделку между риэлтором, который представляет интересы агентства и клиентом, который хочет заказать услуги риэлтора. В сделке указывается номер риэлтора, номер клиента, номер недвижимости, дата сделки и сумма. Сумма сделки подразумевает оплату агентству денег за предоставленные услуги. Как правило услуги риэлтора оцениваются в 2-3% от стоимости недвижимости.

Рисунок 1.3 – ER – модель базы данных

## 1.5 МАКЕТ ПРОТОТИПА СИСТЕМЫ

На рисунке 1.4 показано стартовое окно. Здесь клиент может ввести параметры поиска недвижимости. По нажатию на кнопку «Поиск» будет показана недвижимость соответствующая условию поиска. По нажатию кнопки «Очистить» данные из полей поиска обнулятся. Если клиент желает оставить заявку, он должен нажать на кнопку “Оставить заявку”. После чего программа направит его на форму заполнения заявки, которая изображена на рисунке 1.5.

Рисунок 1.4 – Макет 1

Рисунок 1.5 – Макет 2

Кнопка «Отмена» предназначена для закрытия формы заполнения заявки. После её нажатия клиент перейдёт на стартовую форму. Если клиент ввёл в поля корректные данные и нажал кнопку «ОК», то появится сообщение «Ваша заявка принята. С вами свяжутся в ближайшее время» и эта форма закроется.

Пользователи системы для доступа к расширенным возможностям могут нажать на кнопку «Войти в систему». После этого откроется форма, как на рисунке 1.6.

Рисунок 1.6 – Макет 3

Кнопка «Отмена» закроет эту форму, и пользователь вернется к стартовому окну. После ввода логина, пароля и нажатия кнопки «ОК» система проверит правильность введенных данных. Если логин и пароль неправильный, то появится сообщение с соответствующей фразой. Если логин и пароль верный, то система открывает форму в соответствии с ролью, записанной в базе данных. Эта форма может быть трёх видов, как на рисунках 1.7, 1.8, 1.9. На рисунке 1.7 форма для работы администратора. На рисунке 1.8 форма для работы секретаря. На рисунке 1.9 форма для работы риэлтора.

В соответствии с ролью пользователь имеет разные возможности по сравнению с другими. Так администратор может вносить изменения в любую таблицу, секретарь может только изменять данные в таблице клиентов, а риэлтор может только просматривать таблицы без возможности их изменения.

Рисунок 1.7 – Макет 4

Рисунок 1.8 – Макет 5

Рисунок 1.9 – Макет 6

# 2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ КЛИЕНТА

Клиент работает со стартовой формой, изображённой на рисунке 2.1. Здесь он может найти недвижимость и оставить заявку.

Рисунок 2.1 – Стартовая форма

При запуске формы загружаются данные о недвижимости, которая находится на продаже. Доступа к проданной недвижимости у клиента нет. Это реализовано следующим образом.

var query = (from realt in realtyh

where realt.status == "Продаётся"

orderby realt.priсe

select new {realt.address, realt.priсe, realt.area, realt.type, realt.phone\_number}).ToList();

dataGridView1.DataSource = query;

По нажатию на кнопку «Очистить» срабатывает такой код.

textBox5.Text = "";

textBox4.Text = "";

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

comboBox1.SelectedIndex = 6;

comboBox2.SelectedIndex = 0;

Нажатием на кнопку «Поиск» запускается алгоритм поиска данных из таблицы недвижимости. Данные введенные в поля поиска ищут соответствия в таблице недвижимости и выводятся на экран. Вот часть кода алгоритма поиска.

dataGridView1.DataSource = (from realt in realtyh

orderby realt.priсe

where realt.status == "Продаётся"

where realt.priсe > lp

where realt.priсe < mp

where realt.area > la

where realt.area <ma

select new {realt.address,realt.priсe,realt.area,realt.type,realt.phone\_number }).ToList();

Нажатие на кнопку «Оставить заявку» открывает форм, как на рисунке 2.2. В этой форме клиент вводит данные и нажимает на кнопку «ОК». После этого срабатывает следующий алгоритм.

int s = 0;

if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") || (textBox4.Text == "")||(textBox4.Text.Length!=11))

MessageBox.Show("Заполните корректно все поля");

else

{

try

{

s = (from clien in clienth

where clien.surname == textBox1.Text

where clien.name == textBox2.Text

where clien.pathronymic == textBox3.Text

where clien.phone\_number == textBox4.Text

select clien.id\_client).Distinct().First();

}

catch

{ }

if (s == 0)

{

client new\_client = new client

{ surname = textBox1.Text, name = textBox2.Text, pathronymic = textBox3.Text, phone\_number = textBox4.Text, status = "новый" };

db.client.Add(new\_client);

db.SaveChanges();

}

if (s != 0)

{

var result = db.client.SingleOrDefault(w => w.id\_client == s);

result.status = "новый";

db.SaveChanges();

}

MessageBox.Show("Ваша заявка принята. С вами свяжутся в ближайшее время.");

Form f1 = Application.OpenForms[0];

f1.Show();

this.Close();

}

Рисунок 2.2 - Заявка

В стартовой форме клиент может попытаться войти в систему, нажав соответствующую кнопку. Откроется форма, как на рисунке 2.3. Но без знания логина и пароля клиент больше ничего сделать не может.

Рисунок 2.3 – Вход в систему

## 2.2 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ РИЭЛТОРА

Для того чтобы риэлтору были доступны его возможности, он должен войти в систему. После нажатия кнопки «Войти в систему» откроется форма из рисунка 2.3. Риэлтор вводит логин и пароль. Нажимает кнопку «ОК». И срабатывает следующий алгоритм.

string s = "";

try

{

s = (from user in usersh

where user.login == textBox1.Text

where user.password == textBox2.Text

select user.role).Distinct().First();

}

catch

{

if (s == "")

MessageBox.Show("Неправильный логин или пароль");

}

if (s == "администратор")

{

f4.Show();

this.Close();

Form f1 = Application.OpenForms[0];

f1.ShowInTaskbar = false;

f1.Hide();

}

if (s == "риэлтор")

{

f5.Show();

this.Close();

Form f1 = Application.OpenForms[0];

f1.ShowInTaskbar = false;

f1.Hide();

}

if (s == "секретарь")

{

f6.Show();

this.Close();

Form f1 = Application.OpenForms[0];

f1.ShowInTaskbar = false;

f1.Hide();

}

Система ищет в таблице введённый логин и пароль, и открывает форму работы риэлтора (рисунок 2.4), при этом все предыдущие формы закрываются.

Рисунок 2.4 – Форма для работы риэлтора

Здесь теперь у риэлтора имеется доступ к расширенной таблице недвижимости и клиентов. С помощью полей справа он может искать данные в таблице используя сортировку. Далее представлена часть кода реализации поиска.

if ((comboBox4.SelectedIndex == 0) && (comboBox5.SelectedIndex == 0))

{

if ((textBox8.Text != "") && (textBox9.Text != "") && (textBox10.Text != "") && (textBox13.Text != ""))

dataGridView2.DataSource = (from clien in clienth

orderby clien.surname

where clien.surname == textBox8.Text

where clien.name == textBox9.Text

where clien.pathronymic == textBox10.Text

where clien.phone\_number == textBox13.Text

select new { clien.id\_client, clien.surname, clien.name, clien.pathronymic, clien.phone\_number, clien.status }).ToList();}

Очистка полей поиска происходи аналогично очистке полей у клиента на стартовой форме. На этом возможности риэлтора ограничиваются.

## 2.3 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ СЕКРЕТАРЯ

Секретарь, как и риэлтор для доступа к своим возможностям должен войти в систему. Делается это аналогично риэлтору с помощью того же алгоритма, но в этот раз система откроет форму, как на рисунке 2.5

Рисунок 2.5 – Форма для работы секретаря

Для работы секретаря предоставлено 3 таблицы. Первая с недвижимостью, вторая с клиентами, третья с пользователями. Секретарь является связующим элементом между риэлтором и клиентом. В распоряжении секретаря поиск по расширенным таблицам и возможность добавлять и изменять данные клиентов. Риэлтерское агентство подразумевает под собой офис, где сидит секретарь. Когда в офис приходят клиенты, секретарь самостоятельно может добавить клиента в базу. По нажатию на кнопку «Добавить» открывается форма из рисунка 2.6.

Добавление клиента в базу происходит аналогично оставлению заявки клиентом, только в данном случае выводится сообщение «Клиент добавлен».

Нажатие на кнопку изменить вызовет форму, как на рисунке 2.7.

Рисунок 2.6 – Добавить клиента

Рисунок 2.7 – Изменить данные клиента.

Изменение данных клиента реализовано следующим кодом.

int k = 0;

if (comboBox2.SelectedIndex == 0)

k = 0;

if (comboBox2.SelectedIndex == 1)

k = 1;

if (comboBox2.SelectedIndex == 2)

k = 2;

if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") || (textBox7.Text == "") || (textBox7.Text.Length != 11))

MessageBox.Show("Заполните корректно поля");

else

{

int s = 0;

try

{

s = (from clien in clienth

where clien.surname == textBox1.Text

where clien.name == textBox2.Text

where clien.pathronymic == textBox3.Text

where clien.phone\_number == textBox7.Text

select clien.id\_client).Distinct().First();

}

catch

{ }

if (s == 0)

{

MessageBox.Show("Данный клиент не найден");

}

if (s != 0)

{

if (textBox4.Text == "")

textBox4.Text = textBox1.Text;

if (textBox5.Text == "")

textBox5.Text = textBox2.Text;

if (textBox6.Text == "")

textBox6.Text = textBox3.Text;

if (textBox8.Text == "")

textBox8.Text = textBox7.Text;

if (textBox8.Text.Length != 11)

MessageBox.Show("Заполните корректно поля");

else

{

var result = db.client.SingleOrDefault(w => w.id\_client == s);

result.surname = textBox4.Text;

result.name = textBox5.Text;

result.pathronymic = textBox6.Text;

result.phone\_number = textBox8.Text;

if (k==0)

result.status = "новый";

if (k == 1)

result.status = "обрабатывается";

if (k == 2)

result.status = "выполнено";

db.SaveChanges();

MessageBox.Show("Данные изменены");

Form f6 = Application.OpenForms[0];

f6.Enabled = true;

this.Close();

}

}

## 2.4 РЕАЛИЗАЦИЯ ВОЗМОЖНОСТЕЙ АДМИНИСТРАТОРА

Администратор, как секретарь и риэлтор, должен в войти в сеть под своим логином и паролем. Он имеет в своём распоряжении форму на рисунке 2.8. Она включает в себя таблицы с недвижимостью, с клиентами, с пользователями, с договорами. Администратор может добавлять, изменять и удалять данные из любых таблиц.

Рисунок 2.8 – Форма работы для администратора

По нажатию на кнопки «Добавить», «Удалить», «Изменить» система вызывает специальные формы похожие как на рисунке 2.6, 2.7 и 2.9.

Рисунок 2.9 – Удалить договор

Удаление реализовано следующим образом.

int s = 0;

if ((textBox1.Text == ""))

MessageBox.Show("Заполните поле");

else

{

try

{

s = (from dea in dealh

where dea.id\_deal == Convert.ToInt32(textBox1.Text)

select dea.id\_deal).Distinct().First();

}

catch

{ }

if (s == 0)

{

MessageBox.Show("Данный договор не найден");

}

if (s != 0)

{

List<deal> query = (from us in db.deal

select us).ToList();

deal item = query.First(n => n.id\_deal == s);

db.deal.Remove(item);

db.SaveChanges();

MessageBox.Show("Договор удален");

Form f6 = Application.OpenForms[0];

f6.Enabled = true;

this.Close();

}

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта был разработан прототип информационной системы «Риэлтерское агентство». Проект позволяет упростить работу риэлтерского агентства и сократить использование бумажных носителей.

Все задачи выполнены. Изучена предметная область риэлтерских агентств. Определены функциональные требования к системе. Построены макеты прототипа системы. Построены диаграммы вариантов использования, диаграммы классов. Представлена схема базы данных в виде ER –модели. Разработан и представлен прототип информационной системы.