Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**Курсовая работа**

по дисциплине МДК 01.01

«Разработка программного модуля для учета выдачи форменной одежды сотрудникам предприятия»

**Выполнил:**

Ульянов Андрей

студент группы

ИСП.20А 3 курса

09.02.07 Информационные системы и программирование

очной формы обучения

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_ \_\_\_\_\_\_\_\_\_\_/

Оглавление

[**Введение** 3](#_Toc131505626)

[**1.** **Назначение разработки** 4](#_Toc131505627)

[**2.** **Требования к программному модулю** 4](#_Toc131505628)

[2.1 Требования к функциональным характеристикам: 4](#_Toc131505629)

[2.2 Требования к составу и параметрам технических средств 7](#_Toc131505630)

[2.3 Требования к информационной и программной совместимости 8](#_Toc131505631)

[3. Руководство программиста 8](#_Toc131505632)

[3.1 Разработка базы данных 8](#_Toc131505633)

[3.2 Разработка программного модуля 10](#_Toc131505634)

[3.3 Текст программы 15](#_Toc131505635)

[4. Руководство пользователя 54](#_Toc131505636)

[4.1 Выполнение программного модуля 54](#_Toc131505637)

[4.2 Сообщения пользователю 63](#_Toc131505638)

[5. Тестирование программного модуля 64](#_Toc131505639)

[5.1 Модульное тестирование 64](#_Toc131505640)

[5.2 Тестовые сценарии 64](#_Toc131505641)

[5.2Тестовые сценарии 65](#_Toc131505642)

[6. Отладка программного модуля 67](#_Toc131505643)

[Заключение 67](#_Toc131505644)

**Введение**

Программный модуль разработан для учета выдачи форменной одежды сотрудникам предприятия, предназначен для автоматизации процесса учета и контроля над выдачей и заменой форменной одежды. Он может быть использован в различных организациях, где необходимо учетное ведение выдачи специальной одежды или других материальных ценностей. Программный модуль позволяет упростить и ускорить процесс выдачи форменной одежды, уменьшить вероятность ошибок, связанных с ручным ведением учета, а также повысить контроль за использованием форменной одежды. Объектом, в котором может использоваться данный программный модуль, является организация, предоставляющая своим сотрудникам специальную одежду в рамках их трудовой деятельности.

**Работник** – это специалист, работающий с различными видами недвижимости, посредник между ее покупателем и продавцом.

Задача риелтора – найти наиболее оптимальный вариант для своего клиента, выбрать ему дом или квартиру, отвечающую всем заявленным требованиям, совершив безопасную сделку, или быстро продать по максимально выгодной цене предложенную заказчиком недвижимость.

**Работник склада** **-**  физическое или юридическое лицо, заключившее с собственником какого-либо имущества договор аренды и получившее во временное владение и пользование такое имущество.

**Склад** - физическое или юридическое лицо, которому принадлежит та или иная движимая или недвижимая собственность.

**Предприятие**- это [договорное](https://translated.turbopages.org/proxy_u/en-ru.ru.bb4b3ecb-63ebc070-8ec89a77-74722d776562/https/en.wikipedia.org/wiki/Contract) соглашение, предусматривающее, что пользователь (именуемый арендатором) должен платить владельцу (именуемому [собственником](https://translated.turbopages.org/proxy_u/en-ru.ru.bb4b3ecb-63ebc070-8ec89a77-74722d776562/https/en.wikipedia.org/wiki/Lessor_(leasing))) за использование [актива](https://translated.turbopages.org/proxy_u/en-ru.ru.bb4b3ecb-63ebc070-8ec89a77-74722d776562/https/en.wikipedia.org/wiki/Asset). [Имущество](https://translated.turbopages.org/proxy_u/en-ru.ru.bb4b3ecb-63ebc070-8ec89a77-74722d776562/https/en.wikipedia.org/wiki/Property), здания и транспортные средства являются общ

ими активами, которые сдаются в аренду.

**Форменная одежда -** это гражданско-правовой договор аренды, заключаемый между владельцем имущества и лицом, желающим временно владеть и пользоваться имуществом. Является возмездным, взаимным и консенсуальным договором. Владелец имущества, сдающий его в аренду, называется [собственником](https://translated.turbopages.org/proxy_u/en-ru.ru.bb4b3ecb-63ebc070-8ec89a77-74722d776562/https/en.wikipedia.org/wiki/Lessor_(leasing)), а лицо, берущее это имущество в аренду, называется арендатором.

1. **Назначение разработки**

Разработка проводится на основании приказа о курсовом проектировании. Функциональное назначение программы, разработанной для учета выдачи форменной одежды сотрудникам предприятия, заключается в автоматизации процесса учета и контроля над выдачей и возвратом форменной одежды. Конкретные функции программы включают в себя создание электронной базы данных о сотрудниках, учет количества выданной одежды, генерацию отчетов о выдаче, а также возможность редактирования и удаления данных.

1. **Требования к программному модулю**

## ****2.1 Требования к функциональным характеристикам:****

Программный модуль для программы «Агентство Недвижимости», должен помочь сотрудникам риэлтерского учреждения автоматизировать их деятельность по обработке данных с приписанной к фирме недвижимостью. Также он должен упростить оформление документации по внутренним процессам с недвижимостью и между клиентами.

**Обрабатываемые данные:**

Данные БД:

1. О Виде(одежды): код здания, адрес, цена здания (тыс. руб.), Площадь, код собственника, цена за день аренды, доступность
2. Об одежде: код собственника, Фамилия, Имя, Отчество, Паспортные данные, Номер телефона.
3. Сотрудники: код сотрудника, Фамилия, Имя, Отчество, Номер телефона.
4. О должностях: код арендатора, Фамилия, Имя, Отчество, Паспортные данные, Номер телефона.
5. О цехах: код Договора, код здания, код сотрудника, код покупателя, дата начала аренды, дата окончания аренды, код Пользователя

Приложение должно выполнять следующие функции:

1. Добавление
2. Удаление
3. Сортировка
4. Фильтры
5. Поиск по названию
6. Редактирование
7. Вывод документации на печать

**Диаграммы:**

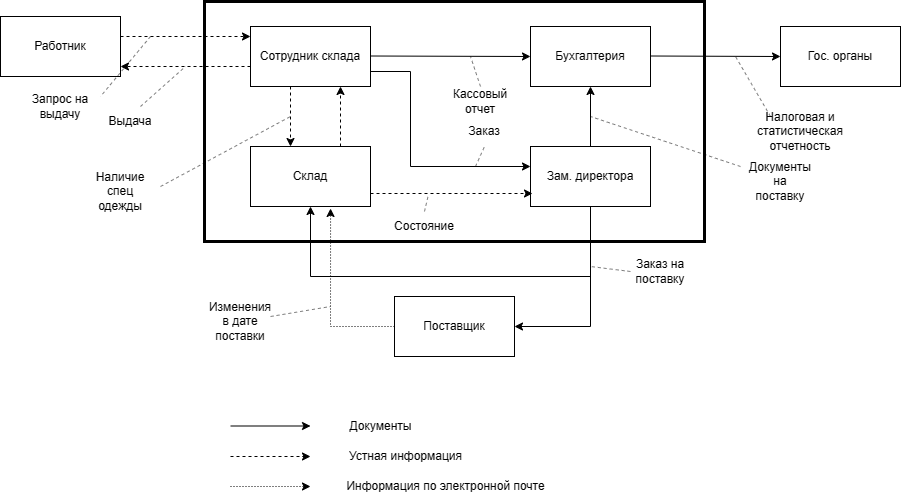


Рис. 1 «Информационная модель»

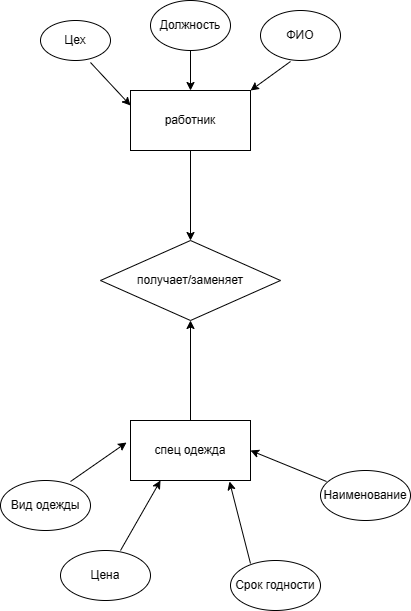


Рис. 2. «Диаграмма "сущность-связь"»

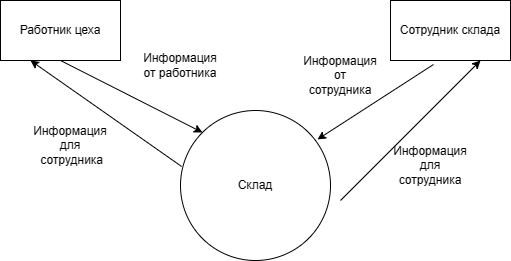


Рис. 3. «Начальная контекстная диаграмма»



Рис. 4. «Диаграмма структур данных»



Рис. 5. «UML диаграмма»

## ****2.2 Требования к составу и параметрам технических средств****

Минимальные характеристики компьютера

|  |  |
| --- | --- |
| Процессор | Intel® Core(TM) i5-6400 |
| Память | От 4 ГБ |
| Разрешение экрана | От 1250x600 |
| Устройства ввода | Клавиатура, мышь |
|  |  |

Оптимальные характеристики компьютера

|  |  |
| --- | --- |
| Процессор | Intel® Core(TM) i5-6400 |
| Память | От 8 ГБ |
| Разрешение экрана | От 1920x1080 |
| Устройства ввода | Клавиатура, мышь |
|  |  |

## ****2.3 Требования к информационной и программной совместимости****

Для корректной работы программы необходимо:

1) Microsoft Office 2016 - это офисный пакет, финальная версия которого вышла в свет 23 сентября 2015 года.

Microsoft Office Word 2016, текстовой процессор, предназначенный для создания, просмотра, а также редактирования различных текстов.

2) Windows 10 - Операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

# Руководство программиста

## 3.1 Разработка базы данных

В данной работе используется база данных, созданная в SQLite.

В базе данных имеется пять таблиц.

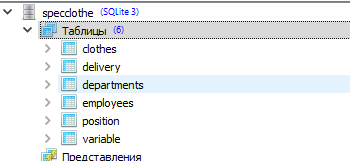


Рис. 6 «Диаграмма базы данных»

Для соединения таблиц использовались следующие типы связей:

 - один ко многим вида: not-**identifying relationship**

Таблица Clothes ings в ней хранится информация о зданиях.

Таблица. 1 «Таблица Clothes»

|  |  |  |  |
| --- | --- | --- | --- |
| **Clothes** | | | |
| Поле | Тип | Атрибуты | Описание |
| idclothes | INT | PK AI | Уникальный номер Одежды |
| clothe | nchar (250) | NN | Название |
| price | Money | NN | Цена |
| term | INT | NN | Срок годности |
| Variable\_idVariable | INT | NN | Номер вида одежды |

Таблица Delivery в ней хранится информация о владельцах зданий.

Таблица. 2 «Таблица Delivery»

|  |  |  |  |
| --- | --- | --- | --- |
| **Delivery** | | | |
| Поле | Тип | Атрибуты | Описание |
| iddelivery | INT | PK AI | Уникальный номер выдачи |
| datato | DATE | NN | Дата выдачи |
| datatrade | DATE | NN | Дата замены |
| Price | INT |  | Итоговая сумма |
| clothes\_idclothes | INT | NN | Номер одежды |
| Employees\_idEmployees | INT | NN | Номер работника |

Таблица Departments в ней хранится информация арендатор.

Таблица. 3 «Таблица Departments»

|  |  |  |  |
| --- | --- | --- | --- |
| **Departments** | | | |
| Поле | Тип | Атрибуты | Описание |
| iddepartment | INT | PK AI | Уникальный номер цеха |
| department | nchar (50) | NN | Наименование цеха |

Таблица Employees в ней хранится информация о сотрудниках.

Таблица. 4 «Таблица Employees»

|  |  |  |  |
| --- | --- | --- | --- |
| **Employees** | | | |
| Поле | Тип | Атрибуты | Описание |
| idEmployees | INT | PK AI | Уникальный номер Сотрудника |
| FIO | nchar (50) | NN | ФИО работника |
| departments\_iddepartment | nchar (50) | NN | Номер цеха |
| position\_idposition | nchar (50) | NN | Номер должности |

Таблица Position в ней хранится информация о договорах.

Таблица. 5 «Таблица Position»

|  |  |  |  |
| --- | --- | --- | --- |
| **Position** | | | |
| Поле | Тип | Атрибуты | Описание |
| idposition | INT | PK AI | Уникальный номер должности |
| discount | INT | NN | Скидка |
| posi | INT | NN | Наименование должности |

Таблица Variable в ней хранится информация о договорах.

Таблица. 5 «Таблица Variable»

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | | | |
| Поле | Тип | Атрибуты | Описание |
| idVariable | INT | PK AI | Уникальный номер вида |
| variable | INT | NN | Наименование вида |

## 3.2 Разработка программного модуля

Для проекта используется программа Visual Studio (WPF). Для подключения базы данных используется технология Entity Framework.

Обозреватель решений структура проекта

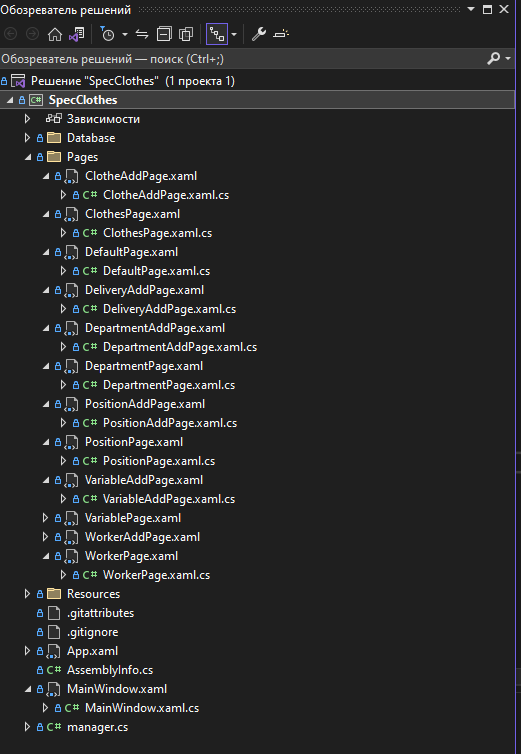


Рис. 7 «Структура проекта»

В проекте было создано три основных папки: Database, Pages, Images.

Папка Database содержит в себе:

1. Класс SpecclotheContext, отвечает за взаимодействие с базой данных.

public partial class SpecclotheContext : DbContext

{

private static SpecclotheContext \_context;

public SpecclotheContext()

{

}

public static SpecclotheContext GetContext()

{

if (\_context == null)

\_context = new SpecclotheContext();

return \_context;

}

public SpecclotheContext(DbContextOptions<SpecclotheContext> options)

: base(options)

{

}

public virtual DbSet<Clothe> Clothes { get; set; }

public virtual DbSet<Delivery> Deliveries { get; set; }

public virtual DbSet<Department> Departments { get; set; }

public virtual DbSet<Employee> Employees { get; set; }

public virtual DbSet<Position> Positions { get; set; }

public virtual DbSet<Variable> Variables { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see http://go.microsoft.com/fwlink/?LinkId=723263.

=> optionsBuilder.UseSqlite("Data Source=specclothe.db");

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Clothe>(entity =>

{

entity.HasKey(e => e.Idclothes);

entity.ToTable("clothes");

entity.Property(e => e.Idclothes).HasColumnName("idclothes");

entity.Property(e => e.Clothe1).HasColumnName("clothe");

entity.Property(e => e.Price).HasColumnName("price");

entity.Property(e => e.Term).HasColumnName("term");

entity.Property(e => e.VariableIdVariable).HasColumnName("Variable\_idVariable");

entity.HasOne(d => d.VariableIdVariableNavigation).WithMany(p => p.Clothes)

.HasForeignKey(d => d.VariableIdVariable)

.OnDelete(DeleteBehavior.ClientSetNull);

});

modelBuilder.Entity<Delivery>(entity =>

{

entity.HasKey(e => e.Iddelivery);

entity.ToTable("delivery");

entity.Property(e => e.Iddelivery).HasColumnName("iddelivery");

entity.Property(e => e.ClothesIdclothes).HasColumnName("clothes\_idclothes");

entity.Property(e => e.Datato).HasColumnName("datato");

entity.Property(e => e.Datatrade).HasColumnName("datatrade");

entity.Property(e => e.EmployeesIdEmployees).HasColumnName("Employees\_idEmployees");

entity.HasOne(d => d.ClothesIdclothesNavigation).WithMany(p => p.Deliveries)

.HasForeignKey(d => d.ClothesIdclothes)

.OnDelete(DeleteBehavior.ClientSetNull);

entity.HasOne(d => d.EmployeesIdEmployeesNavigation).WithMany(p => p.Deliveries)

.HasForeignKey(d => d.EmployeesIdEmployees)

.OnDelete(DeleteBehavior.ClientSetNull);

});

modelBuilder.Entity<Department>(entity =>

{

entity.HasKey(e => e.Iddepartment);

entity.ToTable("departments");

entity.Property(e => e.Iddepartment).HasColumnName("iddepartment");

entity.Property(e => e.Department1).HasColumnName("department");

});

modelBuilder.Entity<Employee>(entity =>

{

entity.HasKey(e => e.IdEmployees);

entity.ToTable("employees");

entity.Property(e => e.IdEmployees).HasColumnName("idEmployees");

entity.Property(e => e.DepartmentsIddepartment).HasColumnName("departments\_iddepartment");

entity.Property(e => e.Fio).HasColumnName("FIO");

entity.Property(e => e.PositionIdposition).HasColumnName("position\_idposition");

entity.HasOne(d => d.DepartmentsIddepartmentNavigation).WithMany(p => p.Employees)

.HasForeignKey(d => d.DepartmentsIddepartment)

.OnDelete(DeleteBehavior.ClientSetNull);

entity.HasOne(d => d.PositionIdpositionNavigation).WithMany(p => p.Employees)

.HasForeignKey(d => d.PositionIdposition)

.OnDelete(DeleteBehavior.ClientSetNull);

});

modelBuilder.Entity<Position>(entity =>

{

entity.HasKey(e => e.Idposition);

entity.ToTable("position");

entity.Property(e => e.Idposition).HasColumnName("idposition");

entity.Property(e => e.Discount).HasColumnName("discount");

entity.Property(e => e.Posi).HasColumnName("posi");

});

modelBuilder.Entity<Variable>(entity =>

{

entity.HasKey(e => e.IdVariable);

entity.ToTable("variable");

entity.Property(e => e.IdVariable).HasColumnName("idVariable");

entity.Property(e => e.Variable1).HasColumnName("variable");

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

1. Класс Clothe, представляет собой обьект одежды:

public partial class Clothe

{

public long Idclothes { get; set; }

public string Clothe1 { get; set; } = null!;

public double Price { get; set; }

public double Term { get; set; }

public long VariableIdVariable { get; set; }

public virtual ICollection<Delivery> Deliveries { get; } = new List<Delivery>();

public virtual Variable VariableIdVariableNavigation { get; set; } = null!;

}

3) Класс Delivery, представляет собой объект выдачи:

{

public long Iddelivery { get; set; }

public string Datato { get; set; } = null!;

public string Datatrade { get; set; } = null!;

public double Price { get; set; }

public long ClothesIdclothes { get; set; }

public long EmployeesIdEmployees { get; set; }

public virtual Clothe ClothesIdclothesNavigation { get; set; } = null!;

public virtual Employee EmployeesIdEmployeesNavigation { get; set; } = null!;

}

4) Класс Department, представляет собой объект Цеха:

public partial class Department

{

public long Iddepartment { get; set; }

public string Department1 { get; set; } = null!;

public virtual ICollection<Employee> Employees { get; } = new List<Employee>();

}

5) Класс Employee, представляет собой объект работника:

public partial class Employee

{

public long IdEmployees { get; set; }

public string Fio { get; set; } = null!;

public long DepartmentsIddepartment { get; set; }

public long PositionIdposition { get; set; }

public virtual ICollection<Delivery> Deliveries { get; } = new List<Delivery>();

public virtual Department DepartmentsIddepartmentNavigation { get; set; } = null!;

public virtual Position PositionIdpositionNavigation { get; set; } = null!;

}

6) Класс Position, представляет собой объект должности:

public partial class Position

{

public long Idposition { get; set; }

public double Discount { get; set; }

public string Posi { get; set; } = null!;

public virtual ICollection<Employee> Employees { get; } = new List<Employee>();

}

7) Класс Variable, представляет собой объект вида одежды:

public partial class Variable

{

public long IdVariable { get; set; }

public string Variable1 { get; set; } = null!;

public virtual ICollection<Clothe> Clothes { get; } = new List<Clothe>();

} }

Код дописанный в контекст подключения C#:

private static RealtorEntities \_context;

public static RealtorEntities GetContext()

{

if (\_context == null)

\_context = new RealtorEntities();

return \_context;

}

Код C# вывод списков таблиц в лист страницы:

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.ToList();

Код C# кнопки удаления строк из таблиц страницы:

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

{

var personForRemoving = ListOwner.SelectedItems.Cast<Owner>().ToList();

var resMessage = MessageBox.Show("Удалить запись?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (resMessage == MessageBoxResult.Yes)

{

try

{

RealtorEntities.GetContext().Owners.RemoveRange(personForRemoving);

RealtorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Данный клиент арендует помещение", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);}}}

Код C# поиска по списку страницы:

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.SurName.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.FirstName.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.Patron.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.PhoneNumber.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.Passport.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

Код C# фильтра поиска по списку страницы:

private void CmbFiltrSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (RbDown.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Passport).ToList();

}

if (RbUp.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Passport).ToList();}}

Код C# сортировки списка страницы

По убыванию:

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Passport).ToList();}

По возврастанию:

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Passport).ToList();

}

Код С# страницы «MainPage»

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

ClassFrame.frmObj = frmMain;

frmMain.Navigate(new Pages.MenuPage());

}

private void frmMain\_ContentRendered(object sender, EventArgs e)

{

if (frmMain.CanGoBack)

{

Back.Visibility = Visibility.Visible;

}

else

{

Back.Visibility = Visibility.Hidden;

}

}

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

{

frmMain.GoBack();}}}

Код xaml страницы «MainPage»

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="6\*"/>

</Grid.RowDefinitions>

<Image Source="/Images/framelog.jpg" Stretch="Fill"/>

<StackPanel Orientation="Horizontal"

VerticalAlignment="Center" HorizontalAlignment="Left" Grid.Column="0" Grid.Row="0">

<Button x:Name="Back" Content="Вернутся" Click="Back\_Click" Width="130" Height="30" Margin="30,10, 10,10" Style="{StaticResource buttonstyle2}"> </Button>

<TextBlock Text="\*Некоторые функции выполняются через контекстное меню списков." VerticalAlignment="Center" HorizontalAlignment="Center" Margin="10" FontStyle="Italic" FontSize="22" Foreground="White" Opacity="0.8"/>

</StackPanel>

<Frame Grid.Row="1"

Name="frmMain"

ContentRendered="frmMain\_ContentRendered"

NavigationUIVisibility="Hidden">

</Frame>

</Grid>

</Window>

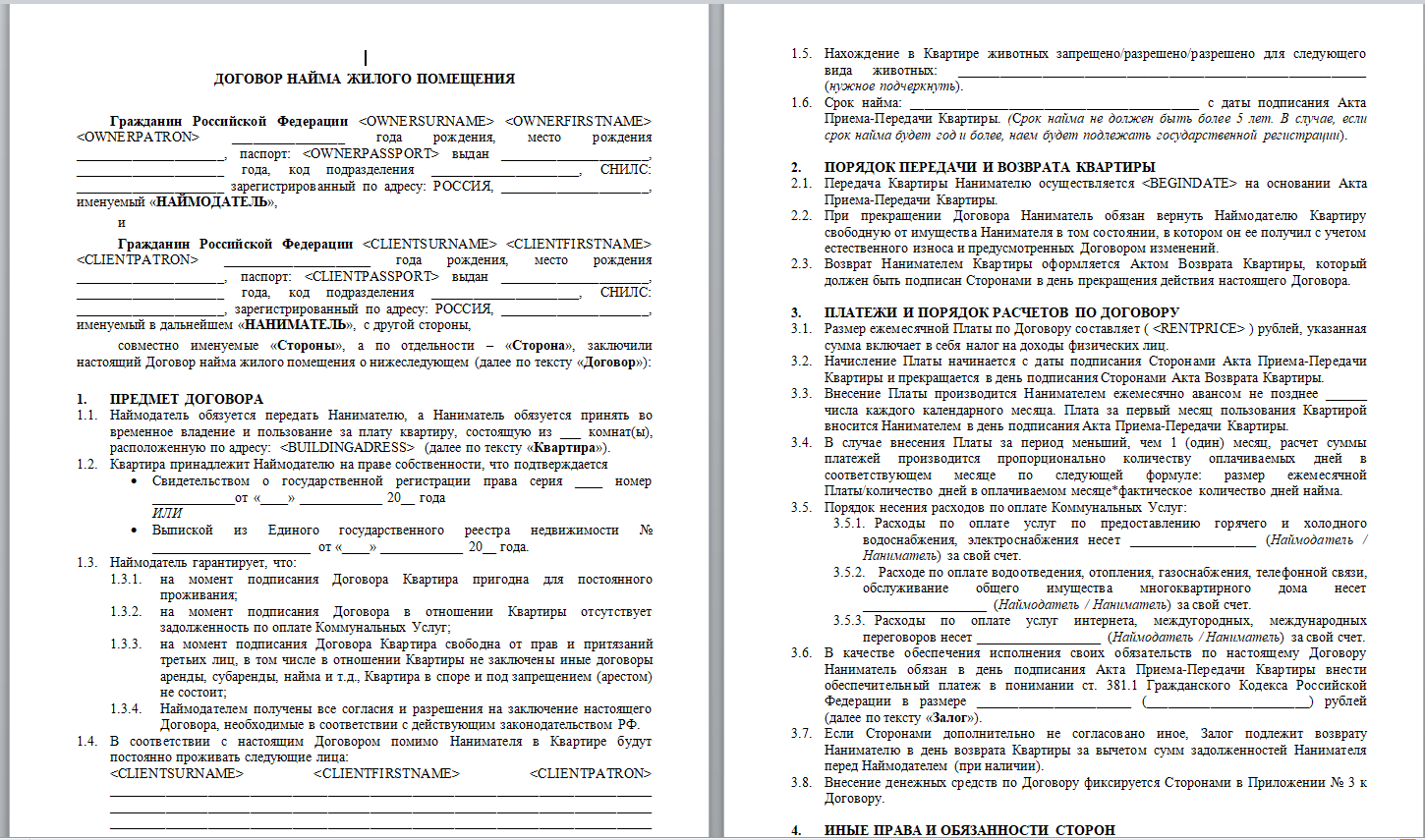


Рис. 9 «Шаблонный Документ Word»

## 3.3 Текст программы

Код страницы «MenuPage»

Xaml:

<Page x:Class="Immovables.Pages.MenuPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="700" d:DesignWidth="800"

Title="MenuPage">

<Grid >

<Grid.RowDefinitions>

<RowDefinition Height="11\*"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Image Source="/Images/LoginBackGround.jpg" Stretch="Fill" />

<StackPanel Grid.Row="0">

<StackPanel Margin="30" Width="350" Height="400">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<TextBlock Text=""/>

<Button x:Name="ToClients" Content="Клиенты" Click="ToClients\_Click" Width="130" Height="30" Margin="10,40, 10,10"></Button>

<Button x:Name="ToBuildings" Content="Помещения" Click="ToBuildings\_Click" Width="130" Height="30" Margin="10,10, 10,10"></Button>

<Button x:Name="ToEmployes" Content="Сотрудники" Click="ToEmployes\_Click" Width="130" Height="30" Margin="10,10, 10,10"></Button>

<Button x:Name="ToOwners" Content="Собственники" Click="ToOwners\_Click" Width="130" Height="30" Margin="10,10, 10,10"></Button>

<Button x:Name="ToContracts" Content="Контракты" Click="ToContracts\_Click" Width="130" Height="30" Margin="10,10, 10,10"></Button>

<Button x:Name="exit" Click="exit\_Click" Width="130" Height="30" Content="Выйти" Margin="10,10, 10,10" />

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1">

<Image Source="/Images/framelog.jpg" Stretch="Fill"/>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Логика взаимодействия для MenuPage.xaml

/// </summary>

public partial class MenuPage : Page

{

public MenuPage()

{

InitializeComponent();

}

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

{

this.NavigationService.Navigate(new BuildingsPage());

}

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

{

this.NavigationService.Navigate(new EmployeesPage());

}

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

{

this.NavigationService.Navigate(new OwnersPage());

}

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

{

this.NavigationService.Navigate(new ContractsPage());

}

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

{

this.NavigationService.Navigate(new ClientsPage());

}

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

{

Application.Current.Shutdown();

}

}

}

Страница «ClientsPage»:

Xaml:

<Page x:Class="Immovables.Pages.ClientsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ClientsPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/11222.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="8\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" >

<TextBlock Text="Поиск"

Width="130"

Height="20"

Margin="30,5" />

<TextBox Width="130"

Height="25"

Name="TxtSearch"

TextChanged="TxtSearch\_TextChanged"/>

<TextBlock Text="Способ поиска и сортировки:" Margin="-10,5" Width="164"

Height="20"/>

<ComboBox x:Name="CmbFiltrSort"

Height="25"

Width="125"

SelectionChanged="CmbFiltrSort\_SelectionChanged"

Margin="0,10,0,10">

<ComboBoxItem Content="По фамилии"/>

<ComboBoxItem Content="По имени"/>

<ComboBoxItem Content="По отчеству"/>

<ComboBoxItem Content="По номеру телефона"/>

<ComboBoxItem Content="По паспорту"/>

</ComboBox>

<StackPanel Margin="10,0,0,0">

<TextBlock Text="Сортировка по" Margin="10,10,0,0" Width="164"/>

<TextBlock Text="возврастанию/убыванию" Margin="10,0,0,10" Width="164"/>

<RadioButton Content="В порядке убывания"

Margin="20,5"

Name="RbUp"

Width="164"

Checked="RbUp\_Checked"/>

<RadioButton Content="В порядке возрастания"

Margin="20,5"

Name="RbDown"

Width="164"

Checked="RbDown\_Checked"/>

</StackPanel>

<StackPanel>

<Button Name="AddButton" Click="AddButton\_Click" Content="Добавить" Margin="-10, 30,0,0" Width="125" Height="25"/>

</StackPanel>

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

<ListView Name="ListClients" Grid.Column="1" SelectionChanged="ListClients\_SelectionChanged" >

<ListView.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</ListView.Background>

<ListView.ContextMenu>

<ContextMenu >

<MenuItem Header="Изменить" Name="editclient" Click="editclient\_Click" />

<MenuItem Header="Удалить" Name="DeleteClient" Click="DeleteClient\_Click" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.View>

<GridView AllowsColumnReorder="true" ColumnHeaderToolTip="Employee Information">

<GridViewColumn DisplayMemberBinding="{Binding Path=SurName}" Header="Фамилия" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=FirstName}" Width="100">

<GridViewColumnHeader>Имя

</GridViewColumnHeader>

</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=Patron}" Header="Отчество" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=PhoneNumber}" Header="Номер телефона" Width="120"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=Passport}" Header="Паспорт" Width="100"/>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for ClientsPage.xaml

/// </summary>

public partial class ClientsPage : Page

{

public ClientsPage()

{

InitializeComponent();

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.ToList();

}

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

{

Classes.ClassFrame.frmObj.Navigate(new ClientsAddEditPage((Client)ListClients.SelectedItem));

}

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

{

var personForRemoving = ListClients.SelectedItems.Cast<Client>().ToList();

var resMessage = MessageBox.Show("Удалить запись?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (resMessage == MessageBoxResult.Yes)

{

try

{

RealtorEntities.GetContext().Clients.RemoveRange(personForRemoving);

RealtorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Данный клиент арендует помещение", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

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

{

this.NavigationService.Navigate(new Pages.ClientsAddEditPage(null));

}

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.SurName.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.FirstName.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.Patron.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.PhoneNumber.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.Passport.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

private void CmbFiltrSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (RbDown.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Passport).ToList();

}

if (RbUp.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Passport).ToList();

}

}

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Passport).ToList();

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Passport).ToList();

}

private void ListClients\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

}

}

Страница «ClientsAddEditPage»:

Xaml:

<Page x:Class="Immovables.Pages.ClientsAddEditPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ClientsAddEditPage">

<Grid >

<Grid.Background>

<ImageBrush ImageSource="/Images/11222.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<StackPanel Name="Gridedit" Grid.Column="0" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel>

<TextBlock Text="Фамилия:" Margin="50, 50, 0, 0"/>

<TextBox Name="SurNametxt" Margin="70, 10, 0, 10" Text="{Binding SurName}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Имя:" Margin="50, 0, 0, 0"/>

<TextBox Name="FirstNametxt" Margin="70, 10, 0, 10" Text="{Binding FirstName}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Отчество:" Margin="50, 0, 0, 0"/>

<TextBox Name="Patrontxt" Margin="70, 10, 0, 10" Text="{Binding Patron}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Номер телефона:" Margin="50, 0, 0, 0"/>

<TextBox Name="PhoneNumbertxt" Margin="70, 10, 0, 10" Text="{Binding PhoneNumber}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Паспорт:" Margin="50, 0, 0, 0"/>

<TextBox Name="Passporttxt" Margin="70, 10, 0, 10" Text="{Binding Passport}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<Button Name="Savebtn" Content="Сохранить" Margin="10" Width="100" Height="25" Click="Savebtn\_Click"/>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for ClientsAddEditPage.xaml

/// </summary>

public partial class ClientsAddEditPage : Page

{

private Client \_currentperson = new Client(); //экземпляр добавляемого пользователя

public ClientsAddEditPage(Client selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser;

DataContext = \_currentperson;

}

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

{

StringBuilder error = new StringBuilder();// Объект для сообщение об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentperson.SurName))

error.AppendLine("Укажите Фамилию");

if (string.IsNullOrWhiteSpace(\_currentperson.FirstName))

error.AppendLine("Укажите Имя");

if (string.IsNullOrWhiteSpace(\_currentperson.PhoneNumber))

error.AppendLine("Укажите Номер телефона формата +7(999)999-99-99");

if (string.IsNullOrWhiteSpace(\_currentperson.Passport))

error.AppendLine("Укажите Паспортные данные");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentperson.IDClient == 0)

RealtorEntities.GetContext().Clients.Add(\_currentperson);//Добавить в контекст

try

{

RealtorEntities.GetContext().SaveChanges();// Сохранить изменения

MessageBox.Show("Данные сохраненны");

this.NavigationService.Navigate(new ClientsPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

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

{

}

}

}

Страница «BuildingsPage»

Xaml:

<Page x:Class="Immovables.Pages.BuildingsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="500" d:DesignWidth="1000"

Title="BuildingsPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/4124214.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2.4\*"/>

<ColumnDefinition Width="8\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" >

<TextBlock Text="Поиск"

Width="130"

Height="20"

Margin="30,5" />

<TextBox Width="130"

Height="25"

Name="TxtSearch"

TextChanged="TxtSearch\_TextChanged"/>

<TextBlock Text="Способ поиска и сортировки:" Margin="-10,5" Width="164"

Height="20"/>

<ComboBox x:Name="CmbFiltrSort"

Height="25"

Width="125"

SelectionChanged="CmbFiltrSort\_SelectionChanged"

Margin="0,10,0,10">

<ComboBoxItem Content="По адресу"/>

<ComboBoxItem Content="По цене аренды(руб.)"/>

<ComboBoxItem Content="По площади(м3)"/>

<ComboBoxItem Content="По собственнику"/>

<ComboBoxItem Content="По цене"/>

</ComboBox>

<StackPanel Margin="10,0,0,0">

<TextBlock Text="Сортировка по" Margin="10,10,0,0" Width="164"/>

<TextBlock Text="возврастанию/убыванию" Margin="10,0,0,10" Width="164"/>

<RadioButton Content="В порядке убывания"

Margin="20,5"

Name="RbUp"

Width="164"

Checked="RbUp\_Checked"/>

<RadioButton Content="В порядке возрастания"

Margin="20,5"

Name="RbDown"

Width="164"

Checked="RbDown\_Checked"/>

</StackPanel>

<StackPanel>

<Button Name="AddButton" Click="AddButton\_Click" Content="Добавить" Margin="-10, 30,0,0" Width="125" Height="25"/>

</StackPanel>

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

<ListView Name="ListBuildings" Grid.Column="1">

<ListView.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</ListView.Background>

<ListView.ContextMenu>

<ContextMenu >

<MenuItem Header="Изменить" Name="editclient" Click="editclient\_Click" />

<MenuItem Header="Удалить" Name="DeleteClient" Click="DeleteClient\_Click" />

<MenuItem Header="Подробнее" Name="BuildingDetails" Click="BuildingDetails\_Click" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.View>

<GridView AllowsColumnReorder="true" ColumnHeaderToolTip="Employee Information">

<GridViewColumn DisplayMemberBinding="{Binding Path=Adress}" Header="Адрес" Width="350"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=Price}" Width="100">

<GridViewColumnHeader>Цена

</GridViewColumnHeader>

</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=ApartmentArea}" Header="Площадь(м3)" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=RentPrice}" Header="Цена аренды (руб.)" Width="120"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=Owner.SurName}" Header="Владелец" Width="120"/>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for BuildingsPage.xaml

/// </summary>

public partial class BuildingsPage : Page

{

public BuildingsPage()

{

InitializeComponent();

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.ToList();

}

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

{

Classes.ClassFrame.frmObj.Navigate(new BuildingsAddEditPage((Building)ListBuildings.SelectedItem));

}

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

{

var personForRemoving = ListBuildings.SelectedItems.Cast<Building>().ToList();

var resMessage = MessageBox.Show("Удалить запись?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (resMessage == MessageBoxResult.Yes)

{

try

{

RealtorEntities.GetContext().Buildings.RemoveRange(personForRemoving);

RealtorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Данное помещение нельзя удалить! Перед удалением помещения удалите арендатора и собственника помещения!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

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

{

this.NavigationService.Navigate(new Pages.BuildingsAddEditPage(null));

}

private void CmbFiltrSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (RbDown.IsChecked == true)

{

if (CmbFiltrSort.Text == "По адресу")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.Adress).ToList();

if (CmbFiltrSort.Text == "По цене аренды(руб.)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.RentPrice).ToList();

if (CmbFiltrSort.Text == "По площади(м3)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.ApartmentArea).ToList();

if (CmbFiltrSort.Text == "По собственнику")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.Owner.SurName).ToList();

if (CmbFiltrSort.Text == "По цене")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.Price).ToList();

}

if (RbUp.IsChecked == true)

{

if (CmbFiltrSort.Text == "По адресу")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.Adress).ToList();

if (CmbFiltrSort.Text == "По цене аренды(руб.)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.RentPrice).ToList();

if (CmbFiltrSort.Text == "По площади(м3)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.ApartmentArea).ToList();

if (CmbFiltrSort.Text == "По собственнику")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.Owner.SurName).ToList();

if (CmbFiltrSort.Text == "По цене")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.Price).ToList();

}

}

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CmbFiltrSort.Text == "По адресу")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.Where(x => x.Adress.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По цене аренды(руб.)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.Where(x => x.RentPrice.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По площади(м3)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.Where(x => x.ApartmentArea.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По собственнику")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.Where(x => x.Owner.SurName.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По цене")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.Where(x => x.Price.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По адресу")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.Adress).ToList();

if (CmbFiltrSort.Text == "По цене аренды(руб.)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.RentPrice).ToList();

if (CmbFiltrSort.Text == "По площади(м3)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.ApartmentArea).ToList();

if (CmbFiltrSort.Text == "По собственнику")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.Owner.SurName).ToList();

if (CmbFiltrSort.Text == "По цене")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderBy(x => x.Price).ToList();

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По адресу")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.Adress).ToList();

if (CmbFiltrSort.Text == "По цене аренды(руб.)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.RentPrice).ToList();

if (CmbFiltrSort.Text == "По площади(м3)")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.ApartmentArea).ToList();

if (CmbFiltrSort.Text == "По собственнику")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.Owner.SurName).ToList();

if (CmbFiltrSort.Text == "По цене")

ListBuildings.ItemsSource = RealtorEntities.GetContext().Buildings.OrderByDescending(x => x.Price).ToList();

}

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

{

Classes.ClassFrame.frmObj.Navigate(new BuildingDetailsPage((Building)ListBuildings.SelectedItem));

}

}

}

Страница «BuildingAddEditPage»:

Xaml:

<Page x:Class="Immovables.Pages.BuildingsAddEditPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="BuildingsAddEditPage">

<Grid >

<Grid.Background>

<ImageBrush ImageSource="/Images/4124214.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.6\*"/>

<ColumnDefinition Width="0.4\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel Width="300" Height="300" Margin="0 30 0 0" Background="White">

<Image x:Name="imagebuilding" Width="300" Height="300" Source="{Binding Image, TargetNullValue={StaticResource House}}" Stretch="UniformToFill"/>

</StackPanel>

<Button Content="Добавить" Name="LoadImage" Width="150" Height="20" Margin="0 10 0 0" Click="LoadImage\_Click"/>

<Button Content="Очистить" Name="ClearImage" Width="150" Height="20" Margin="0 15 0 0" Click="ClearImage\_Click"/>

</StackPanel>

<StackPanel Name="Gridedit" Grid.Row="0" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel>

<TextBlock Text="Адрес:" Margin="50, 50, 0, 0"/>

<TextBox Name="AdresTxt" Margin="70, 10, 0, 10" Text="{Binding Adress}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Цена:" Margin="50, 0, 0, 0"/>

<TextBox Name="PriceTxt" Margin="70, 10, 0, 10" Text="{Binding Price}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Площадь(м3):" Margin="50, 0, 0, 0"/>

<TextBox Name="Areatxt" Margin="70, 10, 0, 10" Text="{Binding ApartmentArea}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Владелец:" Margin="50, 0, 0, 0"/>

<ComboBox x:Name="CmbFiltrOwner"

Height="25"

HorizontalAlignment="Left"

Width="297"

SelectionChanged="CmbFiltrOwner\_SelectionChanged"

SelectedValue="{Binding IDOwner}" Margin="70,10,0,10"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Цена Аренды:" Margin="50, 0, 0, 0"/>

<TextBox Name="renttxt" Margin="70, 10, 0, 10" Text="{Binding RentPrice}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<Button Name="Savebtn" Content="Сохранить" Margin="15,30" Width="100" Height="25" Click="Savebtn\_Click" VerticalAlignment="Center" HorizontalAlignment="Right"/>

</StackPanel>

<StackPanel Grid.Column="1" Grid.Row="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for BuildingsAddEditPage.xaml

/// </summary>

public partial class BuildingsAddEditPage : Page

{

string imgLoc = "пусто";

private Building \_currentperson = new Building(); //экземпляр добавляемого пользователя

public BuildingsAddEditPage(Building selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser;

DataContext = \_currentperson;

CmbFiltrOwner.ItemsSource = RealtorEntities.GetContext().Owners.ToList();

CmbFiltrOwner.SelectedValuePath = "IDOwner";

CmbFiltrOwner.DisplayMemberPath = "SurName";

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentperson.Adress)) error.AppendLine("Укажите Адрес");

if (string.IsNullOrWhiteSpace(PriceTxt.Text)) error.AppendLine("Укажите Цену");

if (string.IsNullOrWhiteSpace(Areatxt.Text)) error.AppendLine("Укажите Площадь(м3)");

if (string.IsNullOrWhiteSpace(CmbFiltrOwner.Text)) error.AppendLine("Укажите Собственника");

if (string.IsNullOrWhiteSpace(renttxt.Text)) error.AppendLine("Укажите Цену аренды");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentperson.IDBuilding == 0)

{

RealtorEntities.GetContext().Buildings.Add(\_currentperson);

try

{

RealtorEntities.GetContext().SaveChanges();// Сохранить изменения

MessageBox.Show("Данные сохраненны");

this.NavigationService.Navigate(new BuildingsPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

\_currentperson.Image = img;

}

if (imgLoc == "пусто") \_currentperson.Image = null;

RealtorEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new BuildingsPage());

MessageBox.Show("Данные о помещении измененны");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void CmbFiltrOwner\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите изображение помещения"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

imagebuilding.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

imagebuilding.Source = (ImageSource)imagebuilding.FindResource("House");

imgLoc = "пусто";

}

}

}

Страница «BuildingDetailsPage»:

Xaml:

<Page x:Class="Immovables.Pages.BuildingDetailsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="BuildingDetailsPage">

<Grid >

<Grid.Background>

<ImageBrush ImageSource="/Images/4124214.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.6\*"/>

<ColumnDefinition Width="0.4\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel Width="300" Height="300" Margin="0 30 10 0" Background="White">

<Image x:Name="imagebuilding" Width="300" Height="300" Source="{Binding Image, TargetNullValue={StaticResource House}}" Stretch="UniformToFill"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Column="1" Grid.Row="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

<StackPanel Name="Gridedit" Grid.Row="0" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel Margin="30, 30, 0 ,0">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Адрес:" Margin="50, 20, 0, 0"/>

<TextBlock Name="AdresTxt" Margin="20, 20, 0, 0" Text="{Binding Adress}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Цена:" Margin="50, 20, 0, 0"/>

<TextBlock Name="PriceTxt" Margin="20, 20, 0, 0" Text="{Binding Price}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Площадь(м3):" Margin="50, 20, 0, 0"/>

<TextBlock Name="Areatxt" Margin="20, 20, 0, 0" Text="{Binding ApartmentArea}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<TextBlock Text="Владелец:" Margin="50, 20, 0, 0"/>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фамилия:" Margin="50, 10, 0, 0"/>

<TextBlock HorizontalAlignment="Left"

Width="350"

Text="{Binding Owner.SurName}" Margin="10, 10, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Имя:" Margin="50, 10, 0, 0"/>

<TextBlock HorizontalAlignment="Left"

Width="350"

Text="{Binding Owner.FirstName}" Margin="10, 10, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Отчество:" Margin="50, 10, 0, 0"/>

<TextBlock HorizontalAlignment="Left"

Width="350"

Text="{Binding Owner.Patron}" Margin="10, 10, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Номер телефона:" Margin="50, 10, 0, 0"/>

<TextBlock HorizontalAlignment="Left"

Width="350"

Text="{Binding Owner.PhoneNumber}" Margin="10, 5, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Цена Аренды:" Margin="50, 20, 0, 0"/>

<TextBlock Name="renttxt" Margin="20, 20, 0, 0" Text="{Binding RentPrice}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for BuildingDetailsPage.xaml

/// </summary>

public partial class BuildingDetailsPage : Page

{

private Building \_currentperson = new Building(); //экземпляр добавляемого пользователя

public BuildingDetailsPage(Building selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser;

DataContext = \_currentperson;

}

}

}

Страница «EmployeesPage»:

Xaml:

<Page x:Class="Immovables.Pages.EmployeesPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="EmployeesPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/FbD5WU6GrUA.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2.4\*"/>

<ColumnDefinition Width="8\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" >

<TextBlock Text="Поиск"

Width="130"

Height="20"

Margin="30,5" />

<TextBox Width="130"

Height="25"

Name="TxtSearch"

TextChanged="TxtSearch\_TextChanged"/>

<TextBlock Text="Способ поиска и сортировки:" Margin="-10,5" Width="164"

Height="20"/>

<ComboBox x:Name="CmbFiltrSort"

Height="25"

Width="125"

SelectionChanged="CmbFiltrSort\_SelectionChanged"

Margin="0,10,0,10">

<ComboBoxItem Content="По фамилии"/>

<ComboBoxItem Content="По имени"/>

<ComboBoxItem Content="По отчеству"/>

<ComboBoxItem Content="По номеру телефона"/>

</ComboBox>

<StackPanel Margin="10,0,0,0">

<TextBlock Text="Сортировка по" Margin="10,10,0,0" Width="164"/>

<TextBlock Text="возврастанию/убыванию" Margin="10,0,0,10" Width="164"/>

<RadioButton Content="В порядке убывания"

Margin="20,5"

Name="RbUp"

Width="164"

Checked="RbUp\_Checked"/>

<RadioButton Content="В порядке возрастания"

Margin="20,5"

Name="RbDown"

Width="164"

Checked="RbDown\_Checked"/>

</StackPanel>

<StackPanel>

<Button Name="AddButton" Click="AddButton\_Click" Content="Добавить" Margin="-10, 30,0,0" Width="125" Height="25"/>

</StackPanel>

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

<ListView Grid.Column="1" Name="ListEmployees" >

<ListView.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</ListView.Background>

<ListView.ContextMenu>

<ContextMenu >

<MenuItem Header="Изменить" Name="editEmployee" Click="editEmployee\_Click" />

<MenuItem Header="Удалить" Name="DeleteEmployee" Click="DeleteEmployee\_Click" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.View>

<GridView AllowsColumnReorder="true" ColumnHeaderToolTip="Employee Information">

<GridViewColumn DisplayMemberBinding="{Binding Path=SurName}" Header="Фамилия" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=FirstName}" Width="100">

<GridViewColumnHeader>Имя

</GridViewColumnHeader>

</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=Patron}" Header="Отчество" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=PhoneNumber}" Header="Номер телефона" Width="120"/>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for EmployeesPage.xaml

/// </summary>

public partial class EmployeesPage : Page

{

public EmployeesPage()

{

InitializeComponent();

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.ToList();

}

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

{

this.NavigationService.Navigate(new Pages.EmployeeAddEditPage(null));

}

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

{

Classes.ClassFrame.frmObj.Navigate(new EmployeeAddEditPage((Employee)ListEmployees.SelectedItem));

}

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

{

var personForRemoving = ListEmployees.SelectedItems.Cast<Employee>().ToList();

var resMessage = MessageBox.Show("Удалить запись?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (resMessage == MessageBoxResult.Yes)

{

try

{

RealtorEntities.GetContext().Employees.RemoveRange(personForRemoving);

RealtorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Данный клиент арендует помещение", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.Where(x => x.SurName.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По имени")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.Where(x => x.FirstName.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.Where(x => x.Patron.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.Where(x => x.PhoneNumber.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

private void CmbFiltrSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (RbDown.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.PhoneNumber).ToList();

}

if (RbUp.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.PhoneNumber).ToList();

}

}

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderBy(x => x.PhoneNumber).ToList();

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListEmployees.ItemsSource = RealtorEntities.GetContext().Employees.OrderByDescending(x => x.PhoneNumber).ToList();

}

}

}

Страница «EmployeesAddEditPage»:

Xaml:

<Page x:Class="Immovables.Pages.EmployeeAddEditPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="EmployeeAddEditPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/FbD5WU6GrUA.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<StackPanel Name="Gridedit" Grid.Row="0">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel>

<TextBlock Text="Фамилия:" Margin="50, 50, 0, 0"/>

<TextBox Name="SurNametxt" Margin="70, 10, 0, 10" Text="{Binding SurName}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Имя:" Margin="50, 0, 0, 0"/>

<TextBox Name="FirstNametxt" Margin="70, 10, 0, 10" Text="{Binding FirstName}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Отчество:" Margin="50, 0, 0, 0"/>

<TextBox Name="Patrontxt" Margin="70, 10, 0, 10" Text="{Binding Patron}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Номер телефона:" Margin="50, 0, 0, 0"/>

<TextBox Name="PhoneNumbertxt" Margin="70, 10, 0, 10" Text="{Binding PhoneNumber}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<Button Name="Savebtn" Content="Сохранить" Margin="10,30" Width="100" Height="25" Click="Savebtn\_Click" VerticalAlignment="Center" HorizontalAlignment="Center"/>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Логика взаимодействия для EmployeeAddEditPage.xaml

/// </summary>

public partial class EmployeeAddEditPage : Page

{

private Employee \_currentperson = new Employee(); //экземпляр добавляемого пользователя

public EmployeeAddEditPage(Employee selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser;

DataContext = \_currentperson;

}

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

{

StringBuilder error = new StringBuilder();// Объект для сообщение об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentperson.SurName))

error.AppendLine("Укажите Фамилию");

if (string.IsNullOrWhiteSpace(\_currentperson.FirstName))

error.AppendLine("Укажите Имя");

if (string.IsNullOrWhiteSpace(\_currentperson.PhoneNumber))

error.AppendLine("Укажите Номер телефона формата +7(999)999-99-99");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentperson.IdEmployee == 0)

RealtorEntities.GetContext().Employees.Add(\_currentperson);//Добавить в контекст

try

{

RealtorEntities.GetContext().SaveChanges();// Сохранить изменения

MessageBox.Show("Данные сохраненны");

this.NavigationService.Navigate(new EmployeesPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}}}}

Страница «OwnersPage»:

Xaml:

<Page x:Class="Immovables.Pages.OwnersPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="OwnersPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/2685941.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="8\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" >

<TextBlock Text="Поиск"

Width="130"

Height="20"

Margin="30,5" />

<TextBox Width="130"

Height="25"

Name="TxtSearch"

TextChanged="TxtSearch\_TextChanged"/>

<TextBlock Text="Способ поиска и сортировки:" Margin="-10,5" Width="164"

Height="20"/>

<ComboBox x:Name="CmbFiltrSort"

Height="25"

Width="125"

SelectionChanged="CmbFiltrSort\_SelectionChanged"

Margin="0,10,0,10">

<ComboBoxItem Content="По фамилии"/>

<ComboBoxItem Content="По имени"/>

<ComboBoxItem Content="По отчеству"/>

<ComboBoxItem Content="По номеру телефона"/>

<ComboBoxItem Content="По паспорту"/>

</ComboBox>

<StackPanel Margin="10,0,0,0">

<TextBlock Text="Сортировка по" Margin="10,10,0,0" Width="164"/>

<TextBlock Text="возврастанию/убыванию" Margin="10,0,0,10" Width="164"/>

<RadioButton Content="В порядке убывания"

Margin="20,5"

Name="RbUp"

Width="164"

Checked="RbUp\_Checked"/>

<RadioButton Content="В порядке возрастания"

Margin="20,5"

Name="RbDown"

Width="164"

Checked="RbDown\_Checked"/>

</StackPanel>

<StackPanel>

<Button Name="AddButton" Click="AddButton\_Click" Content="Добавить" Margin="-10, 30,0,0" Width="125" Height="25"/>

</StackPanel>

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

<ListView Grid.Column="1" Name="ListOwner" >

<ListView.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</ListView.Background>

<ListView.ContextMenu>

<ContextMenu >

<MenuItem Header="Изменить" Name="editOwner" Click="editOwner\_Click" />

<MenuItem Header="Удалить" Name="DeleteClient" Click="DeleteClient\_Click" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.View>

<GridView AllowsColumnReorder="true" ColumnHeaderToolTip="Employee Information">

<GridViewColumn DisplayMemberBinding="{Binding Path=SurName}" Header="Фамилия" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=FirstName}" Width="100">

<GridViewColumnHeader>Имя

</GridViewColumnHeader>

</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path=Patron}" Header="Отчество" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=PhoneNumber}" Header="Номер телефона" Width="120"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=Passport}" Header="Паспорт" Width="120"/>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for OwnersPage.xaml

/// </summary>

public partial class OwnersPage : Page

{

public OwnersPage()

{

InitializeComponent();

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.ToList();

}

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

{

this.NavigationService.Navigate(new Pages.OwnersAddEditPage(null));

}

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

{

var personForRemoving = ListOwner.SelectedItems.Cast<Owner>().ToList();

var resMessage = MessageBox.Show("Удалить запись?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (resMessage == MessageBoxResult.Yes)

{

try

{

RealtorEntities.GetContext().Owners.RemoveRange(personForRemoving);

RealtorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Данный клиент арендует помещение", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

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

{

Classes.ClassFrame.frmObj.Navigate(new OwnersAddEditPage((Owner)ListOwner.SelectedItem));

}

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.SurName.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.FirstName.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.Patron.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.PhoneNumber.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.Where(x => x.Passport.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

private void CmbFiltrSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (RbDown.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Passport).ToList();

}

if (RbUp.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Passport).ToList();

}

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderByDescending(x => x.Passport).ToList();

}

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListOwner.ItemsSource = RealtorEntities.GetContext().Owners.OrderBy(x => x.Passport).ToList();

}

}

Страница « OwnerAddEditPage »:

Xaml:

<Page x:Class="Immovables.Pages.OwnersAddEditPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="OwnersAddEditPage">

<Grid >

<Grid.Background>

<ImageBrush ImageSource="/Images/2685941.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<StackPanel Name="Gridedit" Grid.Column="0" >

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel>

<TextBlock Text="Фамилия:" Margin="50, 50, 0, 0"/>

<TextBox Name="SurNametxt" Margin="70, 10, 0, 10" Text="{Binding SurName}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Имя:" Margin="50, 0, 0, 0"/>

<TextBox Name="FirstNametxt" Margin="70, 10, 0, 10" Text="{Binding FirstName}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Отчество:" Margin="50, 0, 0, 0"/>

<TextBox Name="Patrontxt" Margin="70, 10, 0, 10" Text="{Binding Patron}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Номер телефона:" Margin="50, 0, 0, 0"/>

<TextBox Name="PhoneNumbertxt" Margin="70, 10, 0, 10" Text="{Binding PhoneNumber}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Паспорт:" Margin="50, 0, 0, 0"/>

<TextBox Name="Passporttxt" Margin="70, 10, 0, 10" Text="{Binding Passport}" Width="350" HorizontalAlignment="Left"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<Button Name="Savebtn" Content="Сохранить" Margin="10" Width="100" Height="25" Click="Savebtn\_Click"/>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for OwnersAddEditPage.xaml

/// </summary>

public partial class OwnersAddEditPage : Page

{

private Owner \_currentperson = new Owner(); //экземпляр добавляемого пользователя

public OwnersAddEditPage(Owner selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser;

DataContext = \_currentperson;

}

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

{

StringBuilder error = new StringBuilder();// Объект для сообщение об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentperson.SurName))

error.AppendLine("Укажите Фамилию");

if (string.IsNullOrWhiteSpace(\_currentperson.FirstName))

error.AppendLine("Укажите Имя");

if (string.IsNullOrWhiteSpace(\_currentperson.PhoneNumber))

error.AppendLine("Укажите Номер телефона формата +7(999)999-99-99");

if (string.IsNullOrWhiteSpace(\_currentperson.Passport))

error.AppendLine("Укажите Паспортные данные");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentperson.IDOwner == 0)

RealtorEntities.GetContext().Owners.Add(\_currentperson);//Добавить в контекст

try

{

RealtorEntities.GetContext().SaveChanges();// Сохранить изменения

MessageBox.Show("Данные сохраненны");

this.NavigationService.Navigate(new OwnersPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

Страница «ContractsPage»:

Xaml:

<Page x:Class="Immovables.Pages.ContractsPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ContractsPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/3b2.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="8\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Column="0" >

<TextBlock Text="Поиск"

Width="130"

Height="20"

Margin="30,5" />

<TextBox Width="130"

Height="25"

Name="TxtSearch"

TextChanged="TxtSearch\_TextChanged"/>

<TextBlock Text="Способ поиска и сортировки:" Margin="-10,5" Width="164"

Height="20"/>

<ComboBox x:Name="CmbFiltrSort"

Height="25"

Width="125"

SelectionChanged="CmbFiltrSort\_SelectionChanged"

Margin="0,10,0,10">

<ComboBoxItem Content="По адресу"/>

<ComboBoxItem Content="По цене аренды(руб.)"/>

<ComboBoxItem Content="По площади(м3)"/>

<ComboBoxItem Content="По сотруднику"/>

<ComboBoxItem Content="По собственнику"/>

<ComboBoxItem Content="По арендатору"/>

<ComboBoxItem Content="По дате начала аренды"/>

<ComboBoxItem Content="По дате окончания аренды"/>

</ComboBox>

<StackPanel Margin="10,0,0,0">

<TextBlock Text="Сортировка по" Margin="10,10,0,0" Width="164"/>

<TextBlock Text="возврастанию/убыванию" Margin="10,0,0,10" Width="164"/>

<RadioButton Content="В порядке убывания"

Margin="20,5"

Name="RbUp"

Width="164"

Checked="RbUp\_Checked"/>

<RadioButton Content="В порядке возрастания"

Margin="20,5"

Name="RbDown"

Width="164"

Checked="RbDown\_Checked"/>

</StackPanel>

<StackPanel>

<Button Name="AddButton" Click="AddButton\_Click" Content="Добавить" Margin="-10, 30,0,0" Width="125" Height="25"/>

</StackPanel>

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

</StackPanel>

<ListView Grid.Column="1" Name="ListContracts">

<ListView.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.8" Stretch="Fill"></ImageBrush>

</ListView.Background>

<ListView.ContextMenu>

<ContextMenu >

<MenuItem Header="Удалить" Name="DeleteClient" Click="DeleteClient\_Click" />

<MenuItem Header="Изменить" Name="editclient" Click="editclient\_Click" />

<MenuItem Header="Печать" Name="printbtn" Click="printbtn\_Click" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.View>

<GridView AllowsColumnReorder="true" ColumnHeaderToolTip="Employee Information">

<GridViewColumn DisplayMemberBinding="{Binding Path = Building.Adress}" Header="Адрес" Width="400"/>

<GridViewColumn DisplayMemberBinding="{Binding Path = Building.RentPrice}" Width="120">

<GridViewColumnHeader>Цена аренды (руб.)</GridViewColumnHeader>

</GridViewColumn>

<GridViewColumn DisplayMemberBinding="{Binding Path = Building.ApartmentArea}" Header="Площадь(м3)" Width="100"/>

<GridViewColumn DisplayMemberBinding="{Binding Path = Employee.SurName}" Header="Сотрудник" Width="120"/>

<GridViewColumn DisplayMemberBinding="{Binding Path = Building.Owner.SurName}" Header="Собственник" Width="120"/>

<GridViewColumn DisplayMemberBinding="{Binding Path = Client.SurName}" Header="Арендатор" Width="120"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=BeginDate}" Header="Дата начала аренды" Width="auto"/>

<GridViewColumn DisplayMemberBinding="{Binding Path=EndDate}" Header="Дата окончания аренды" Width="auto"/>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for ClientsPage.xaml

/// </summary>

public partial class ClientsPage : Page

{

public ClientsPage()

{

InitializeComponent();

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.ToList();

}

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

{

Classes.ClassFrame.frmObj.Navigate(new ClientsAddEditPage((Client)ListClients.SelectedItem));

}

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

{

var personForRemoving = ListClients.SelectedItems.Cast<Client>().ToList();

var resMessage = MessageBox.Show("Удалить запись?", "Подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (resMessage == MessageBoxResult.Yes)

{

try

{

RealtorEntities.GetContext().Clients.RemoveRange(personForRemoving);

RealtorEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.ToList();

}

catch (Exception ex)

{

MessageBox.Show("Данный клиент арендует помещение", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

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

{

this.NavigationService.Navigate(new Pages.ClientsAddEditPage(null));

}

private void TxtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.SurName.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.FirstName.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.Patron.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.PhoneNumber.ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.Where(x => x.Passport.ToString().ToLower().Contains(TxtSearch.Text.ToLower())).ToList();

}

private void CmbFiltrSort\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (RbDown.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Passport).ToList();

}

if (RbUp.IsChecked == true)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Passport).ToList();

}

}

private void RbUp\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderBy(x => x.Passport).ToList();

}

private void RbDown\_Checked(object sender, RoutedEventArgs e)

{

if (CmbFiltrSort.Text == "По фамилии")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.SurName).ToList();

if (CmbFiltrSort.Text == "По имени")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.FirstName).ToList();

if (CmbFiltrSort.Text == "По отчеству")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Patron).ToList();

if (CmbFiltrSort.Text == "По номеру телефона")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.PhoneNumber).ToList();

if (CmbFiltrSort.Text == "По паспорту")

ListClients.ItemsSource = RealtorEntities.GetContext().Clients.OrderByDescending(x => x.Passport).ToList();

}

private void ListClients\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

}

}

Страница «ContractsAddEditPage»:

Xaml:

<Page x:Class="Immovables.Pages.ContractsAddEditPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ContractsAddEditPage">

<Grid >

<Grid.Background>

<ImageBrush ImageSource="/Images/3b2.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<StackPanel Name="Gridedit" Grid.Column="0">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel>

<TextBlock Text="Адрес здания:" Margin="50, 50, 0, 0"/>

<ComboBox x:Name="CmbFiltrAdress"

Height="25"

HorizontalAlignment="Left"

Width="297"

SelectionChanged="CmbFiltrAdress\_SelectionChanged"

SelectedValue="{Binding IDBuilding}" Margin="70,10,0,10"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Сотрудник:" Margin="50, 0, 0, 0"/>

<ComboBox x:Name="CmbFiltrEmployee"

Height="25"

HorizontalAlignment="Left"

Width="297"

SelectionChanged="CmbFiltrEmployee\_SelectionChanged"

SelectedValue="{Binding IDEmployee}" Margin="70,10,0,10"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Арендатор:" Margin="50, 0, 0, 0"/>

<ComboBox x:Name="CmbFiltrClient"

Height="25"

HorizontalAlignment="Left"

Width="297"

SelectionChanged="CmbFiltrClient\_SelectionChanged"

SelectedValue="{Binding IDClient}" Margin="70,10,0,10"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Дата начала аренды:" Margin="50, 0, 0, 0"/>

<DatePicker Name="datePicker1" HorizontalAlignment="Left" Width="297" SelectedDate="{Binding BeginDate}" IsDropDownOpen="False" SelectedDateFormat="Short" Margin="70,10,0,10"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Дата окончания аренды:" Margin="50, 0, 0, 0"/>

<DatePicker Name="datePicker2" HorizontalAlignment="Left" Width="297" SelectedDate="{Binding EndDate}" IsDropDownOpen="False" SelectedDateFormat="Short" Margin="70,10,0,10"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<Button Name="Savebtn" Content="Сохранить" Margin="10" Width="100" Height="25" Click="Savebtn\_Click"/>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Interaction logic for ContractsAddEditPage.xaml

/// </summary>

public partial class ContractsAddEditPage : Page

{

private Contract \_currentperson = new Contract(); //экземпляр добавляемого пользователя

public ContractsAddEditPage(Contract selectedUser)

{

InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser;

DataContext = \_currentperson;

CmbFiltrAdress.ItemsSource = RealtorEntities.GetContext().Buildings.ToList();

CmbFiltrAdress.SelectedValuePath = "IDBuilding";

CmbFiltrAdress.DisplayMemberPath = "Adress";

CmbFiltrEmployee.ItemsSource = RealtorEntities.GetContext().Employees.ToList();

CmbFiltrEmployee.SelectedValuePath = "IdEmployee";

CmbFiltrEmployee.DisplayMemberPath = "SurName";

CmbFiltrClient.ItemsSource = RealtorEntities.GetContext().Clients.ToList();

CmbFiltrClient.SelectedValuePath = "IDClient";

CmbFiltrClient.DisplayMemberPath = "SurName";

datePicker1.Text = DateTime.Now.Date.ToString();

}

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

{

StringBuilder error = new StringBuilder();// Объект для сообщение об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(CmbFiltrEmployee.Text))

error.AppendLine("Укажите сотрудника!");

if (string.IsNullOrWhiteSpace(CmbFiltrClient.Text))

error.AppendLine("Укажите клиента!");

if (string.IsNullOrWhiteSpace(datePicker1.Text))

error.AppendLine("Укажите дату начала аренды!");

if (string.IsNullOrWhiteSpace(CmbFiltrAdress.Text))

error.AppendLine("Укажите адрес!");

if (string.IsNullOrWhiteSpace(datePicker2.Text))

error.AppendLine("Укажите дату окончания аренды!");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentperson.IDContract == 0)

RealtorEntities.GetContext().Contracts.Add(\_currentperson);//Добавить в контекст

try

{

RealtorEntities.GetContext().SaveChanges();// Сохранить изменения

MessageBox.Show("Данные сохраненны");

this.NavigationService.Navigate(new ContractsPage());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

private void CmbFiltrAdress\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void CmbFiltrEmployee\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void CmbFiltrClient\_SelectionChanged(object sender, SelectionChangedEventArgs e){}}}

Страница «PrintPage»:

Xaml:

<Page x:Class="Immovables.Pages.PrintPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Immovables.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PrintPage">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/Images/3b2.jpg" Opacity="0.7" Stretch="Fill"></ImageBrush>

</Grid.Background>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.3\*"/>

</Grid.RowDefinitions>

<StackPanel Name="Gridedit" Grid.Column="0">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Адрес здания:" Margin="50, 20, 0, 0"/>

<TextBlock Text="Цена аренды:" Margin="270, 20, 0, 0"/>

<TextBlock Text="Площадь:" Margin="150, 20, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Name="Adress"

Height="25"

HorizontalAlignment="Left"

Width="297"

Text="{Binding Building.Adress}" Margin="70,10,0,10"/>

<TextBlock

Name="RentPrice"

Height="25"

HorizontalAlignment="Left"

Width="100"

Text="{Binding Building.RentPrice}" Margin="70,10,0,10"/>

<TextBlock

Name="ApartmentArea"

Height="25"

HorizontalAlignment="Left"

Width="297"

Text="{Binding Building.ApartmentArea}" Margin="125,10,0,10"/>

</StackPanel>

</StackPanel>

<StackPanel>

<TextBlock Text="Сотрудник:" Margin="300, 0, 0, 0"/>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фио:" Margin="60, 10, 0, 0"/>

<TextBlock Text="Номер телефона:" Margin="360, 10, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock

Height="25"

Width="125"

Text="{Binding Employee.SurName}" Margin="70,10,0,10"/>

<TextBlock

Height="25"

Width="125"

Text="{Binding Employee.FirstName}" />

<TextBlock

Height="25" Width="125"

Text="{Binding Employee.Patron}" />

<TextBlock

Height="25" Width="125"

Text="{Binding Employee.PhoneNumber}" />

</StackPanel>

</StackPanel>

<StackPanel>

<TextBlock Text="Арендатор:" Margin="300, 0, 0, 0"/>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фио:" Margin="60, 10, 0, 0"/>

<TextBlock Text="Номер телефона:" Margin="360, 10, 0, 0"/>

<TextBlock Text="Паспорт:" Margin="125, 10, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Name="ClientSurname"

Height="25"

Width="125"

Text="{Binding Client.SurName}" Margin="70,10,0,10"/>

<TextBlock Name="ClientFirstName"

Height="25"

Width="125"

Text="{Binding Client.FirstName}" />

<TextBlock Name="ClientPatron"

Height="25"

Width="125"

Text="{Binding Client.Patron}" />

<TextBlock Name="ClientPhoneNumber"

Height="25"

Text="{Binding Client.PhoneNumber}" />

<TextBlock Name="ClientPassport"

Height="25"

Text="{Binding Client.Passport}" Margin="125, 10, 0, 0" />

</StackPanel>

</StackPanel>

<StackPanel>

<TextBlock Text="Собственник:" Margin="300, 0, 0, 0"/>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фио:" Margin="60, 10, 0, 0"/>

<TextBlock Text="Номер телефона:" Margin="360, 10, 0, 0"/>

<TextBlock Text="Паспорт:" Margin="125, 10, 0, 0"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Name="OwnerSurname"

Height="25"

Width="125"

Text="{Binding Building.Owner.SurName}" Margin="70,10,0,10"/>

<TextBlock Name="OwnerFirstName"

Height="25"

Width="125"

Text="{Binding Building.Owner.FirstName}" />

<TextBlock Name="OwnerPatron"

Height="25"

Width="125"

Text="{Binding Building.Owner.Patron}" />

<TextBlock Name="OwnerPhoneNumber"

Height="25"

Text="{Binding Building.Owner.PhoneNumber}" />

<TextBlock Name="OwnerPassport"

Height="25"

Text="{Binding Building.Owner.Passport}" Margin="115, 10, 0, 0" />

</StackPanel>

</StackPanel>

<StackPanel>

<TextBlock Text="Дата начала аренды:" Margin="50, 0, 0, 0"/>

<TextBlock Name="datePicker1" HorizontalAlignment="Left" Width="297" Text="{Binding BeginDate}" Margin="70,10,0,10"/>

</StackPanel>

<StackPanel>

<TextBlock Text="Дата окончания аренды:" Margin="50, 0, 0, 0"/>

<TextBlock Name="datePicker2" HorizontalAlignment="Left" Width="297" Text="{Binding EndDate}" Margin="70,10,0,10"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="1">

<StackPanel.Background>

<ImageBrush ImageSource="/Images/color.png" Opacity="0.7" Stretch="Fill"></ImageBrush>

</StackPanel.Background>

<Button Name="printbtn" Content="Печать" Margin="0,35,0,0" Width="100" Height="25" Click="Printbtn\_Click"/>

</StackPanel>

</Grid>

</Page>

С#:

namespace Immovables.Pages

{

/// <summary>

/// Логика взаимодействия для PrintPage.xaml

/// </summary>

public partial class PrintPage : Page

{

private Contract \_currentperson = new Contract();

private FileInfo \_fileInfo;

public PrintPage(string fileName)

{

if (File.Exists(fileName))

{

\_fileInfo = new FileInfo(fileName);

}

else

{

MessageBox.Show("Файл не найден", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private Contract \_currentOrdersGoods = new Contract();

public PrintPage(Contract selectedUser)

{ InitializeComponent();

if (selectedUser != null)

\_currentperson = selectedUser; DataContext = \_currentperson;

}

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

{

var helper = new PrintPage("C:\\Users\\meleg\\Desktop\\Immovables3-master\\Documents\\arendadogovor.docx");

var items = new Dictionary<string, string>

{

{ "<CLIENTSURNAME>", ClientSurname.Text },

{ "<CLIENTFIRSTNAME>", ClientFirstName.Text},

{ "<CLIENTPATRON>", ClientPatron.Text},

{ "<CLIENTPASSPORT>", ClientPassport.Text },

{ "<OWNERSURNAME>", OwnerSurname.Text },

{ "<OWNERFIRSTNAME>", OwnerFirstName.Text},

{ "<OWNERPATRON>", OwnerPatron.Text},

{ "<OWNERPASSPORT>", OwnerPassport.Text },

{ "<BUILDINGADRESS>", Adress.Text},

{ "<BEGINDATE>", datePicker1.Text},

{ "<RENTPRICE>", RentPrice.Text },

{ "<CLIENTPHONENUMBER>", ClientPhoneNumber.Text},

{ "<OWNERPHONENUMBER>", OwnerPhoneNumber.Text},

};

helper.Process(items);

}

internal bool Process(Dictionary<string, string> items)

{

try

{

var app = new Word.Application();

Object file = \_fileInfo.FullName;

Object missing = Type.Missing;

app.Documents.Open(file);

foreach (var item in items)

{

Word.Find find = app.Selection.Find;

find.Text = item.Key;

find.Replacement.Text = item.Value;

Object wrap = Word.WdFindWrap.wdFindContinue;

Object replace = Word.WdReplace.wdReplaceAll;

find.Execute(FindText: Type.Missing,

MatchCase: false,

MatchWholeWord: false,

MatchWildcards: false,

MatchSoundsLike: missing,

MatchAllWordForms: false,

Forward: true,

Wrap: wrap,

Format: false,

ReplaceWith: missing, Replace: replace);

}

return true;

}

catch (Exception ex) { Console.WriteLine(ex.Message); }

return false;}}}

# 4. Руководство пользователя

## 4.1 Выполнение программного модуля

Для того чтобы воспользоваться представленной программой, необходимо произвести следующие действия: установить программу Visual Studio и СУБД SQL Server Microsoft.

Страница «MenuPage» выполняет роль главного меню и содержит кнопки переходов на другие страницы.

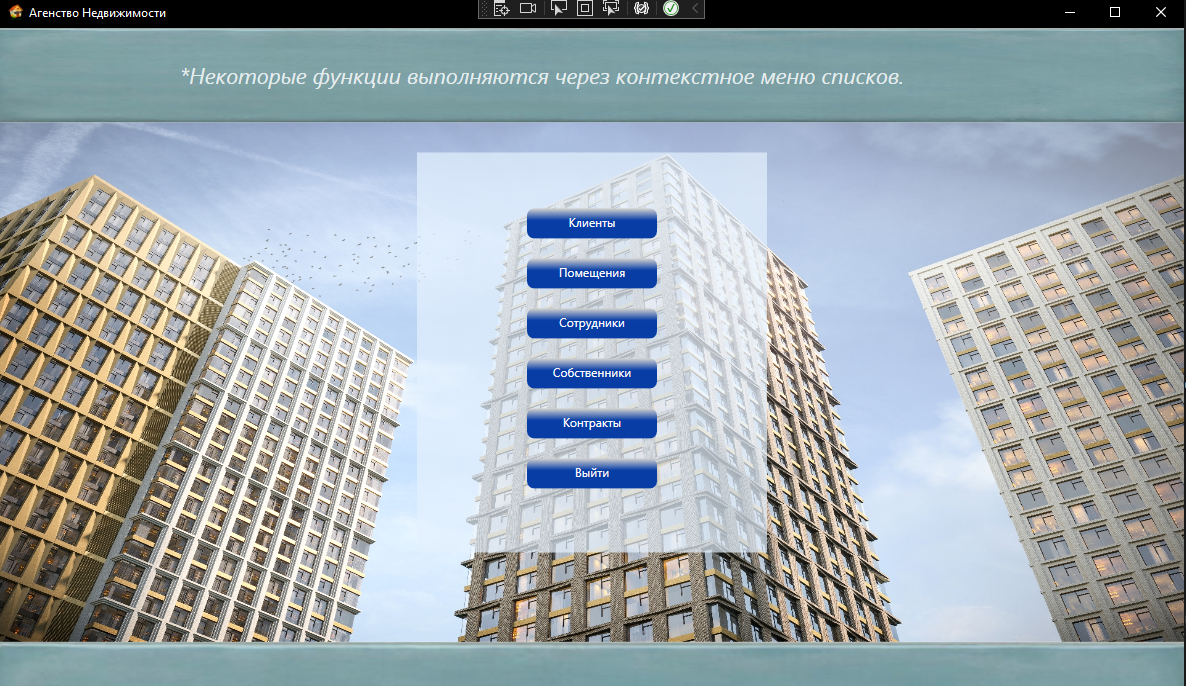


Рис. 10 «Страница MenuPage»

Страница «ClientsPage*»* выводит данные о клиентах(арендаторах), можно вернуться в главное меню по кнопке «Вернуться». Можно найти арендаторов по всем возможным видам данных выбрав «Способ поиска и сортировки» на нужный, также можно фильтровать в алфавитном порядке по способу сортировки. Есть возможность добавлять клиентов, редактировать и удалять.

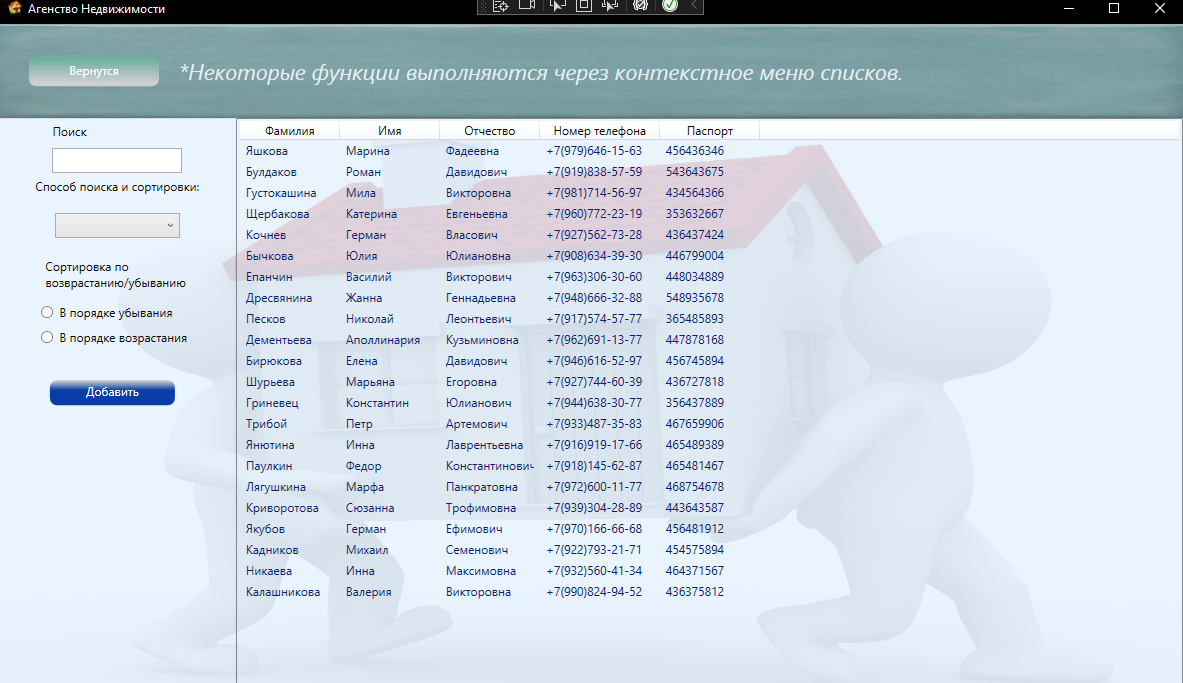


Рис. 11 «Страница ClientsPage»

Страница «ClientsAddEditPage*»* в ней вводятся данные о клиенте, которые затем сохраняются в базу данных.

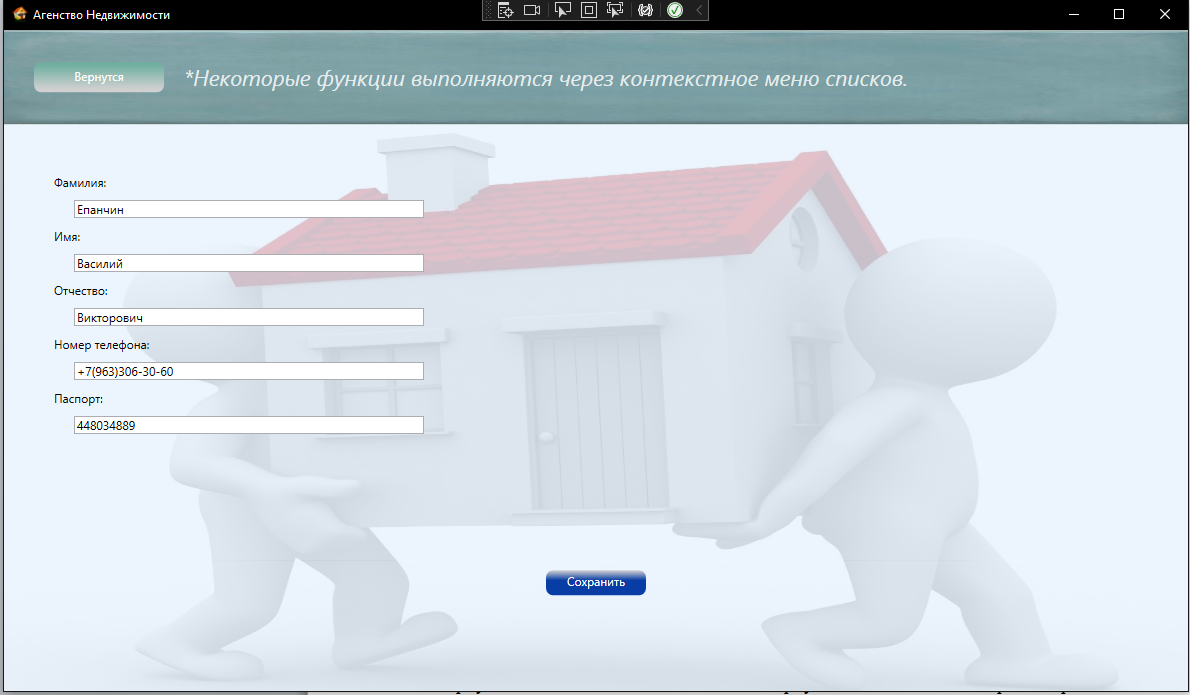


Рис. 12 «Страница ClientsAddEditPage »

Страница «BuildingsPage*»* выводит данные о зданиях, можно вернуться в главное меню по кнопке «Вернуться». Можно найти здания по всем возможным видам данных изменив «Способ поиска и сортировки» на нужный, также можно фильтровать в алфавитном порядке, по способу сортировки. Есть возможность добавлять здания, редактировать и удалять. Так же имеется возможность посмотреть более подробную информацию о здании.

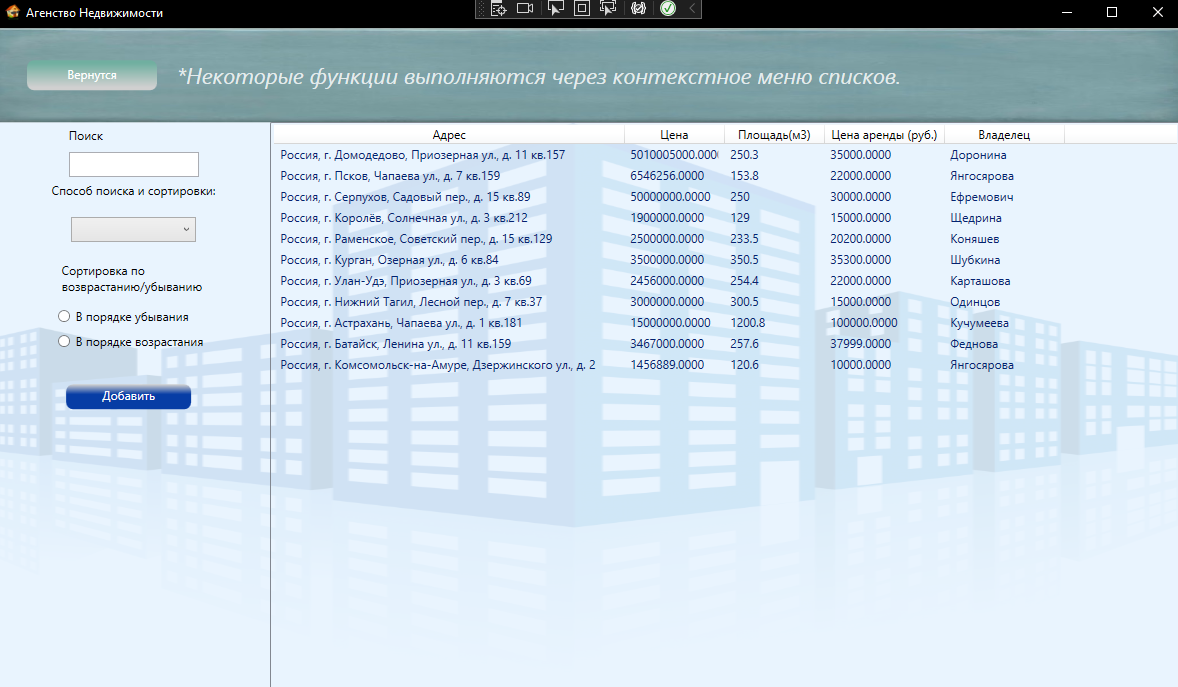


Рис. 13 «Страница BuildingsPage»

Страница «BuildingAddEditPage*»,* в ней вводятся данные о здании, которые затем сохраняются в базу данных.

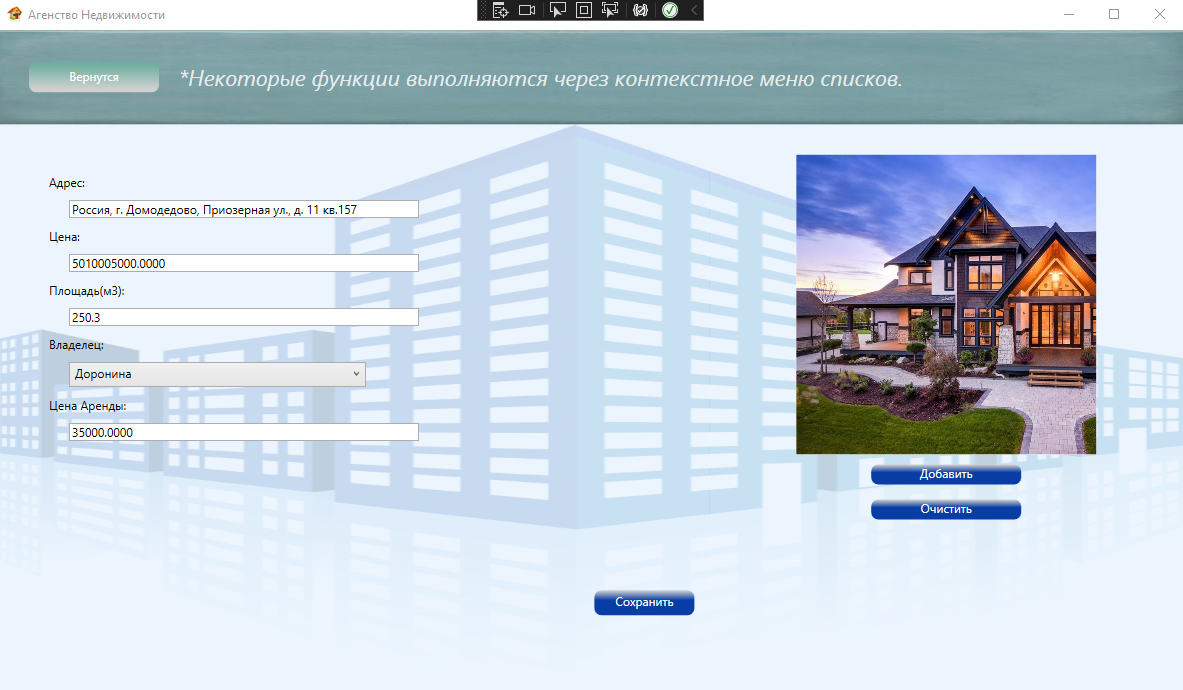


Рис. 14 «Страница BuildingAddEditPage»

Страница «BuildingDetailsPage*»,* выводит подробную информацию о здании

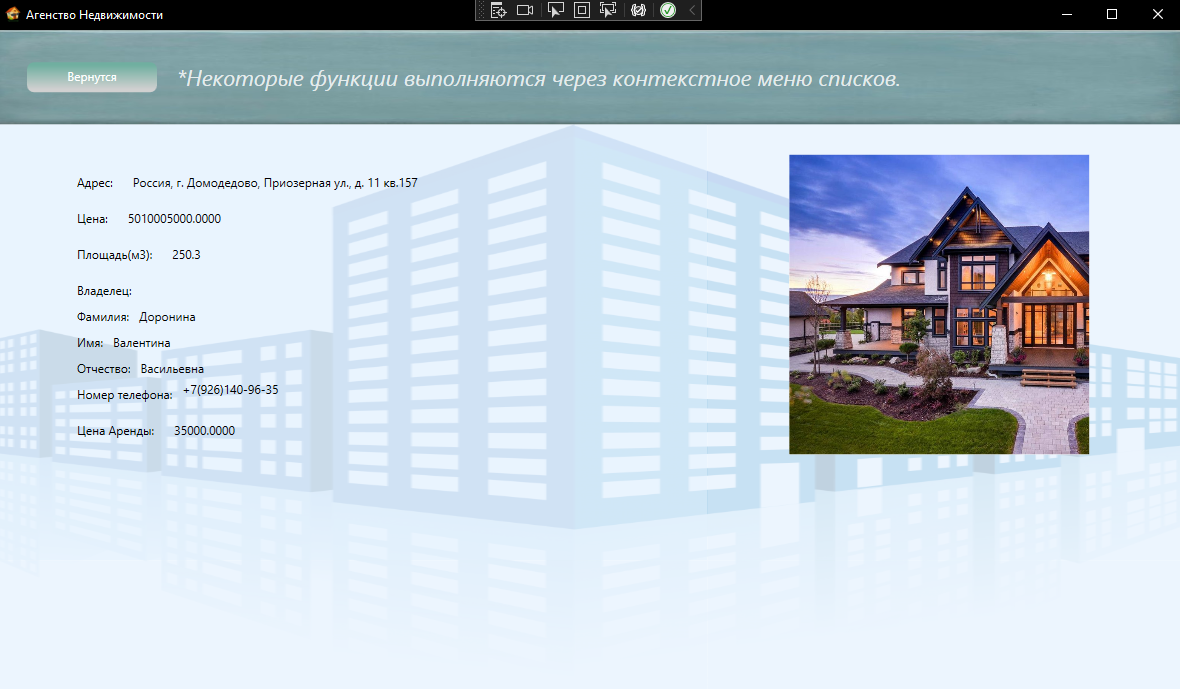


Рис. 15 «Страница BuildingDetailsPage»

Страница «EmployeesPage*»* выводит данные о сотрудниках, можно вернуться в главное меню по кнопке «Вернуться». Можно найти сотрудников по всем возможным видам данных изменив «Способ поиска и сортировки» на нужный, также можно фильтровать в алфавитном порядке, по способу сортировки. Есть возможность добавлять здания, редактировать и удалять.

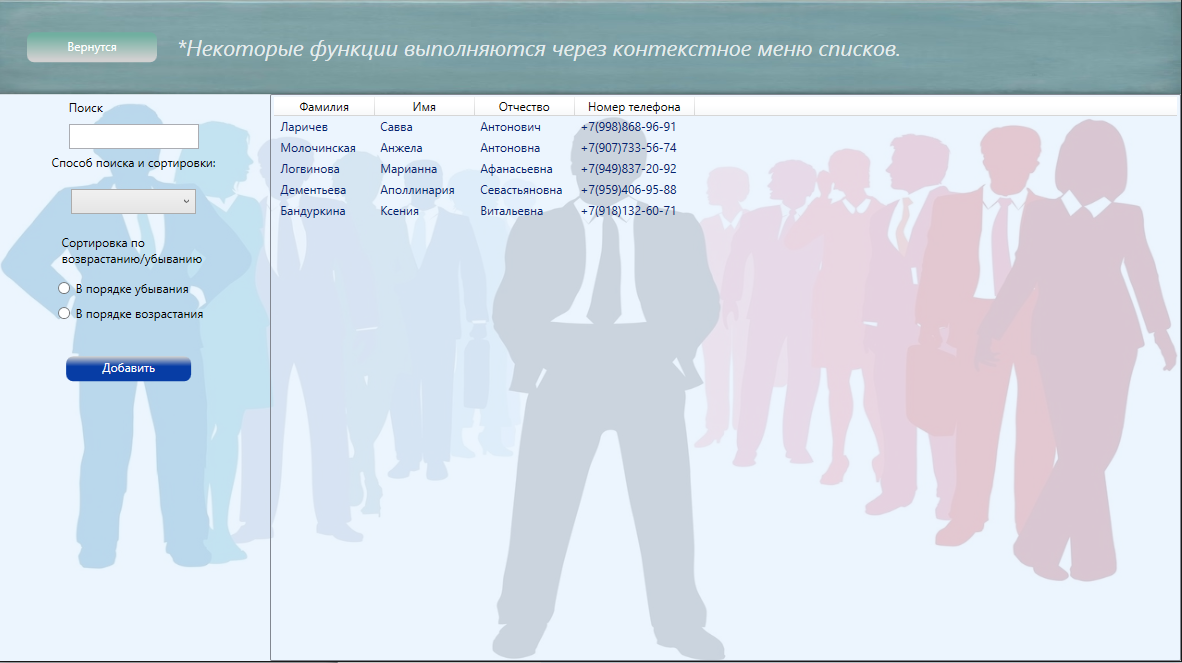


Рис. 16 «Страница EmployeesPage»

Страница «EmployeesAddEditPage*»,* в ней вводятся данные о сотруднике, которые затем сохраняются в базу данных.

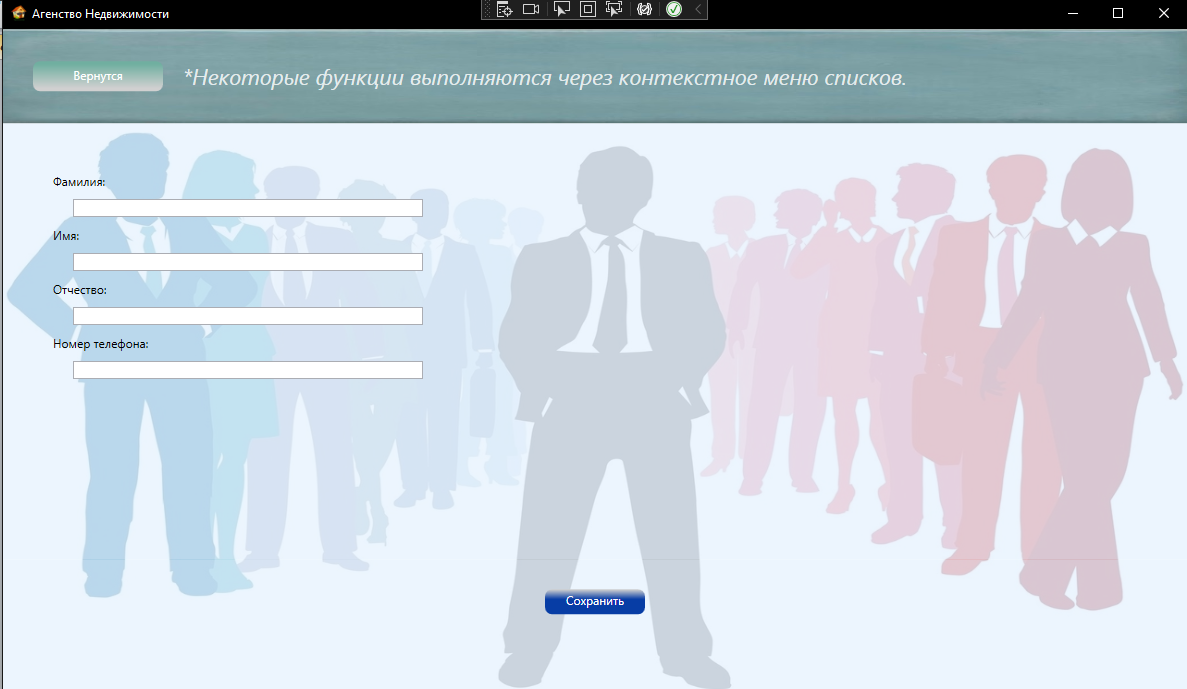


Рис. 17 «Страница EmployeesAddEditPage»

Страница «OwnersPage*»* выводит данные о владельцах, можно вернуться в главное меню по кнопке «Вернуться». Можно найти владельцев по всем возможным видам данных изменив «Способ поиска и сортировки» на нужный, также можно фильтровать в алфавитном порядке, по способу сортировки. Есть возможность добавлять здания, редактировать и удалять.

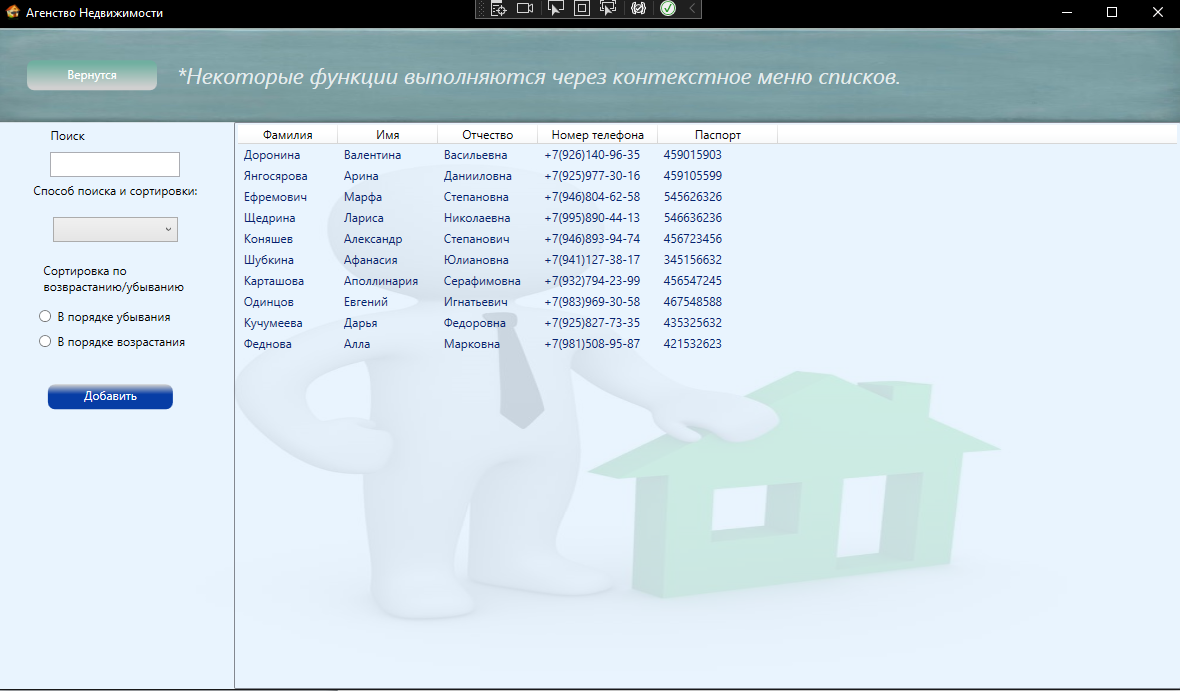


Рис. 18 «Страница OwnersPage»

Страница «OwnerAddEditPage*»,* в ней вводятся данные о владельце, которые затем сохраняются в базу данных.

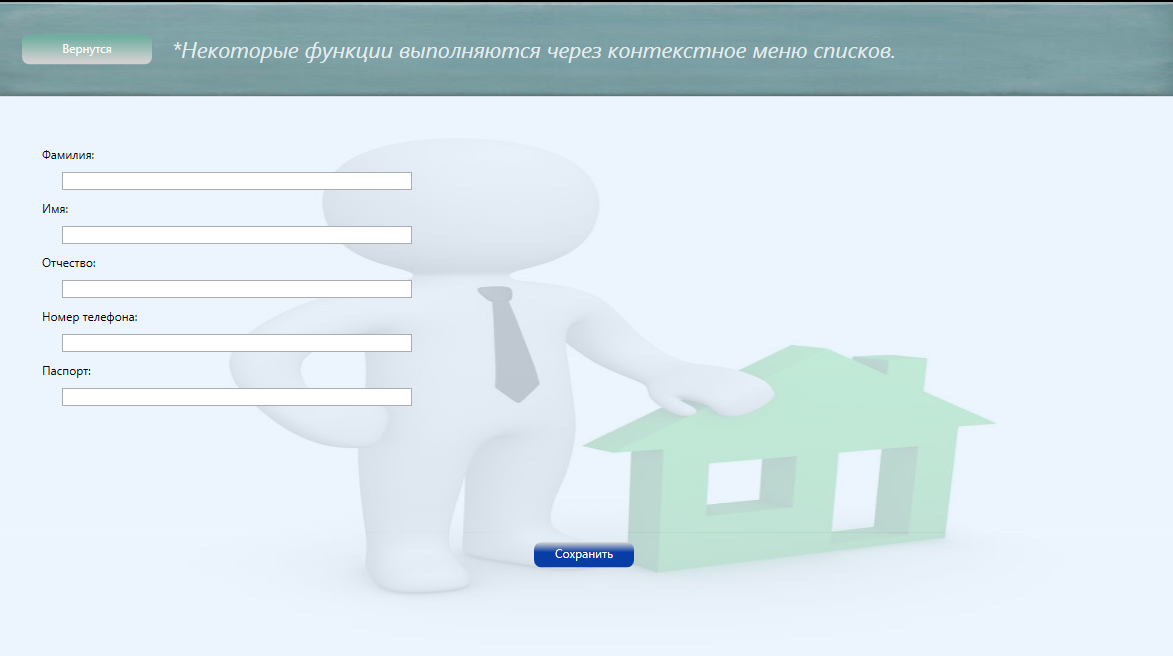


Рис. 19 «Страница OwnerAddEditPage»

Страница «ContractsPage*»* выводит данные о договорах, можно вернуться в главное меню по кнопке «Вернуться». Можно найти договорах по всем возможным видам данных изменив «Способ поиска и сортировки» на нужный, также можно фильтровать в алфавитном порядке, по способу сортировки. Есть возможность добавлять договора, редактировать и удалять, так же имеется возможность вывода данных в виде документа Office Word.



Рис. 20 «Страница ContractsPage»

Страница «ContractsAddEditPage*»,* в ней вводятся данные о контракте, которые затем сохраняются в базу данных.



Рис. 21 «Страница ContractsAddEditPage»

Страница «PrintPage», позволяет вывести контракт в шаблон Office Word

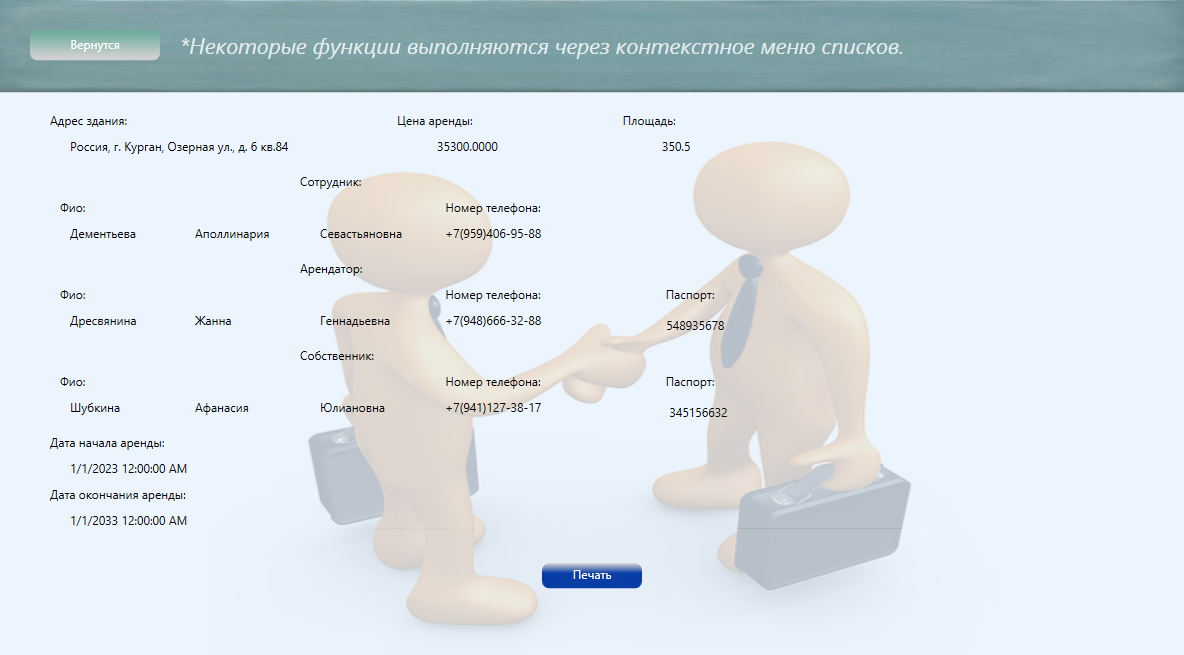


Рис. 22 «Страница PrintPage»

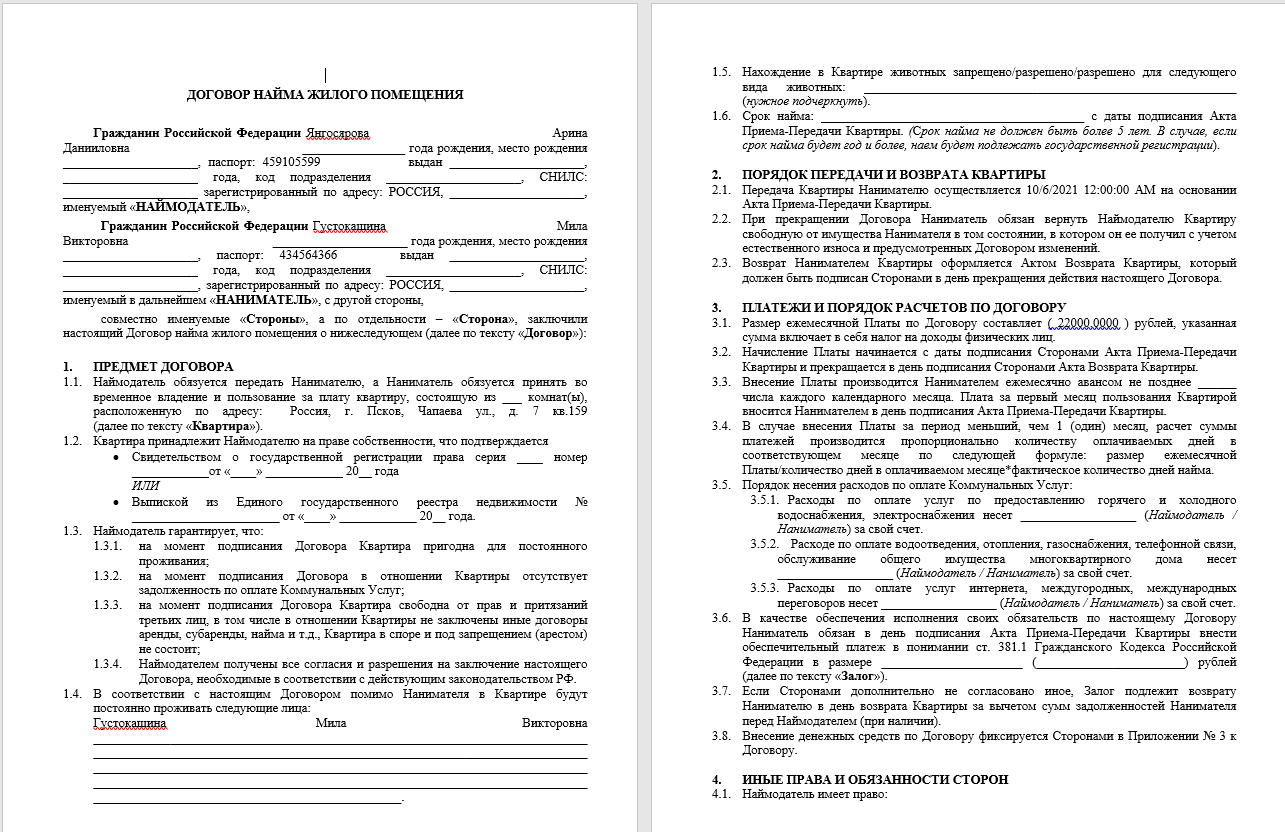


Рис. 23 «Шаблонный Документ Word»

Выпадающий список с вариантами сортировки и поиска.

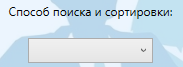


Рис. 24 «Список поиска и сортировки»

Данный элемент управления позволяет выбрать вариант сортировки данных, перед сортировкой сначала необходимо выбрать способ из списка вышеуказанного списка.

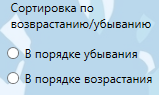


Рис. 25 «Сортировка»

Данный элемент управления позволяет найти необходимый элемент в таблице с данными по его названию, перед поиском сначала необходимо выбрать способ из списка вышеуказанного списка.



Рис. 26 «Поиск»

Кнопка «Добавить» при нажатии переводит на страницу добавления новой записи в базу данных.



Рис. 27 «Кнопка добавления»

Кнопка «Сохранить» при нажатии сохраняет изменения текущей записи в базу данных.



Рис. 28 «Кнопка сохранения»

Контекстное меню списка вызывается нажатием правой кнопки мыши по списку, в контекстном меню имеются дополнительные функции для взаимодействия с данными.

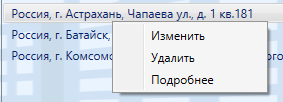


Рис. 29 «Контекстное меню»

Кнопка «Изменить» в контекстном меню переводит на страницу редактирования элемента списка, на котором было вызванно контекстное меню.

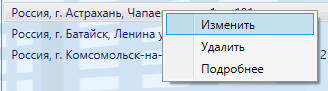


Рис. 30 «Кнопка измения в контекстном меню»

Кнопка «Удалить» в контекстном меню производит удаление элемента списка, на котором было вызванно контекстное меню.

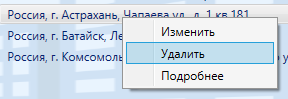


Рис. 31 «Кнопка удаления в контекстном меню»

Кнопка «Подробнее» в контекстном меню переводит на страницу с более подробной информацией о элементе списка, на котором было вызванно контекстное меню.

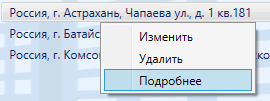


Рис. 32 «Кнопка подробнее в контекстном меню»

Кнопка «Печать» в контекстном меню переводит на страницу для вывода на печать с информацией о элементе списка, на котором было вызванно контекстное меню.

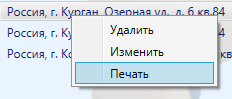


Рис. 33 «Кнопка печати в контекстном меню»

Кнопка «Печать» при нажатии переносит данные записи в ранее созданный документ Office Word, для печати в бумажный вариант.



Рис. 34 «Кнопка печати»

## 4.2 Сообщения пользователю

Оповещение в случае удачного добавления/редактирования записи.

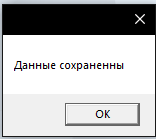


Рис. 35 «Добавление нового рейса»

При удалении записи, потребуется подтверждение действия от пользователя, он должен нажать «Yes» в случае согласия, и «No» в ином случае.

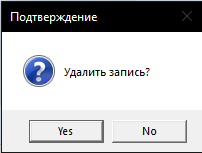


Рис. 36 «Удаление (Выбор)»

Оповещение пользователя о удалении данных из базы данных.

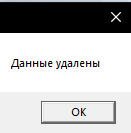


Рис. 37 «Удаление рейса»

В случае если пользователь введёт неверные или неполные данные в поля ввода он получит сообщение с инструкцией.

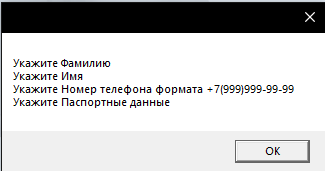


Рис. 38 «Предупреждение»

# 5. Тестирование программного модуля

## 5.1 Модульное тестирование

## Тестовые сценарии

1)Код С# модульных тестов

В качестве образца возьмем номер телефона: +7(917)574-57-77. В данной ситуации тест пройдет успешо

1. Длинна номера телефона должа быть равна 16

[TestMethod]

public static int GetNumberStrength(string PhoneNumber)

{

int expected = 1;

PhoneNumber = "+7(917)574-57-77";

int result = 0;

if (PhoneNumber.Length == 16)

{

result++;

}

double actual = result;

Assert.AreEqual(expected, actual);

return result;

}

В качестве образца возьмем номер телефона: +7(917)574-57-77. В данной ситуации тест пройдет успешо

1. Номер телефона не должен содержать буквы

[TestMethod]

public static int GetNumberLetter(string PhoneNumber)

{

int expected = 1;

PhoneNumber = "+7(917)574-57-77";

int result = 0;

if (!Regex.Match(PhoneNumber, "[a-z]").Success)

{

result += 1;

}

double actual = result;

Assert.AreEqual(expected, actual);

return result;

}

В качестве образца возьмем номер телефона: +7(917)574-57-77. В данной ситуации тест пройдет успешо

3. Данный тест проверяет номер телефона на наличие цифр.

[TestMethod]

public static int GetNameNumber(string name)

{

int expected = 1;

PhoneNumber = "+7(917)574-57-77";

int result = 0;

if (Regex.Match(name, " ").Success)

{

result++;

}

double actual = result;

Assert.AreEqual(expected, actual);

return result;

}

Итоги тестов:

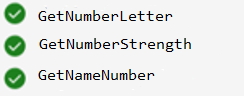


Рис. 39 «Итог тестов»

## 5.2Тестовые сценарии

Таблица№9 «Аннотация теста»

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Название проекта | Immovables |
| Рабочая версия | 1.0 |
| Имя тестирующего | Мелега Алексей Валерьевич |
| Дата(ы) теста | 03.04.2023 |

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | GetNumberStrength |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Проверка длинны символов |
| Краткое изложение теста | При сохранении информации номера телефона, должна быть проверка количества символов. Длинна номера телефона должа быть равна 16 |
| Этапы теста | Зайти на страницу добавления, указать следующие значения: ФИО, Номер телефона, пасспорт |
| Тестовые данные | ФМО: Яшкова, Марина Фадеевна. Номер телефона: +7(979)646-15-63 Паспорт: 456436346 |
| Ожидаемый результат | Сообщение отсутствует |
| Фактический результат | Запись сохранена |
| Предварительное условие | Должна быть кнопка для перехода на страницу добавления и сохранения данных, элементы управления куда будут заполняться данные. |
| Постусловие | Данные были добавлены в таблицу. |
| Статус(Зачет/Незачет) | Зачёт |

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | GetNumberLetter |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Проверка наличия букв |
| Краткое изложение теста | При сохранении информации номера телефона, должна быть проверка наличия букв в строке. Номер телефона не должен содержать буквы |
| Этапы теста | Зайти на страницу добавления, указать следующие значения: ФИО, Номер телефона, пасспорт |
| Тестовые данные | ФМО: Яшкова, Марина Фадеевна. Номер телефона: +7(979)646-15-63 Паспорт: 456436346 |
| Ожидаемый результат | Сообщение отсутствует |
| Фактический результат | Запись сохранена |
| Предварительное условие | Должна быть кнопка для перехода на страницу добавления и сохранения данных, элементы управления куда будут заполняться данные. |
| Постусловие | Данные были добавлены в таблицу. |
| Статус(Зачет/Незачет) | Зачёт |

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | GetNameNumber |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Низкий |
| Заголовок/название теста | Проверка на проверку пробелов |
| Краткое изложение теста | При сохранении информации номера телефона, должна быть проверка наличия пробелов в строке. Данный тест проверяет номер телефона на наличие пробелов. |
| Этапы теста | Зайти на страницу добавления, указать следующие значения: ФИО, Номер телефона, пасспорт |
| Тестовые данные | ФМО: Яшкова, Марина Фадеевна. Номер телефона: +7(979)646-15-63 Паспорт: 456436346 |
| Ожидаемый результат | Сообщение отсутствует |
| Фактический результат | Запись сохранена |
| Предварительное условие | Должна быть кнопка для перехода на страницу добавления и сохранения данных, элементы управления куда будут заполняться данные. |
| Постусловие | Данные были добавлены в таблицу. |
| Статус(Зачет/Незачет) | Зачёт |

# 6. Отладка программного модуля

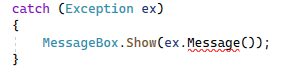
1) Было введено неправильное название таблицы базы данных



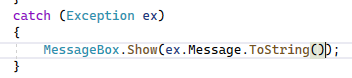
Решение:



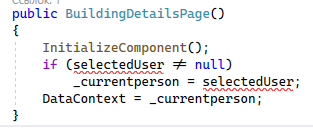
2) Неверный формат вывода сообщения об ошибке



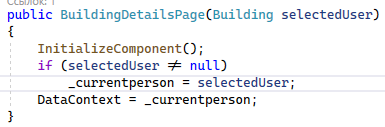
Решение:



3) Отсутствует упоминание в методе, экземпляра таблицы



Решение:



# Заключение

В заключение хочу сказать, что представленное программное приложение способно успешно осуществлять выполнение поставленных перед ним задач, однако его функциональные возможности могут быть расширены. В будущем в программном модуле можно было бы добавить функции, связанные с бухгалтерскими рассчётами. Необходимо расширить количество данных в базе данных, для больших подробностей. В процессе разработки данного программного приложения, я углубил свои знания и навыки в области информационных технологий, изучив множество новых и полезных методик и технологий, которые я успешно применил в ходе работы.