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](#_Toc131762688)

[**1.** **Назначение разработки** 4](#_Toc131762689)

[**2.** **Требования к программному модулю** 4](#_Toc131762690)

[2.1 Требования к функциональным характеристикам: 4](#_Toc131762691)

[2.2 Требования к составу и параметрам технических средств 7](#_Toc131762692)

[2.3 Требования к информационной и программной совместимости 8](#_Toc131762693)

[3. Руководство программиста 8](#_Toc131762694)

[3.1 Разработка базы данных 8](#_Toc131762695)

[3.2 Разработка программного модуля 10](#_Toc131762696)

[3.3 Текст программы 17](#_Toc131762697)

[4. Руководство пользователя 17](#_Toc131762698)

[4.1 Выполнение программного модуля 17](#_Toc131762699)

[4.2 Сообщения пользователю 25](#_Toc131762700)

[5. Тестирование программного модуля 26](#_Toc131762701)

[5.1 Модульное тестирование 26](#_Toc131762702)

[5.2 Тестовые сценарии 26](#_Toc131762703)

[5.2Тестовые сценарии 27](#_Toc131762704)

[6. Отладка программного модуля 29](#_Toc131762705)

[Заключение 30](#_Toc131762706)

[Приложение 1: 30](#_Toc131762707)

**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:**

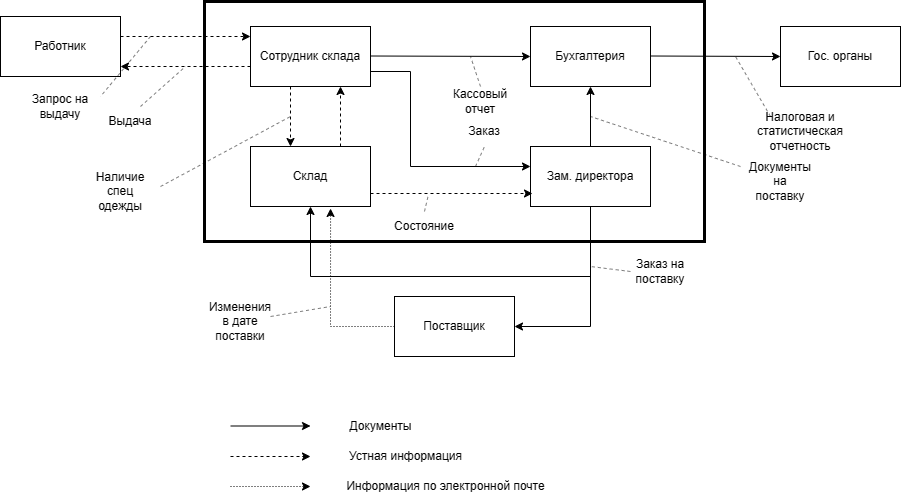


Рис. 1 «Информационная модель»

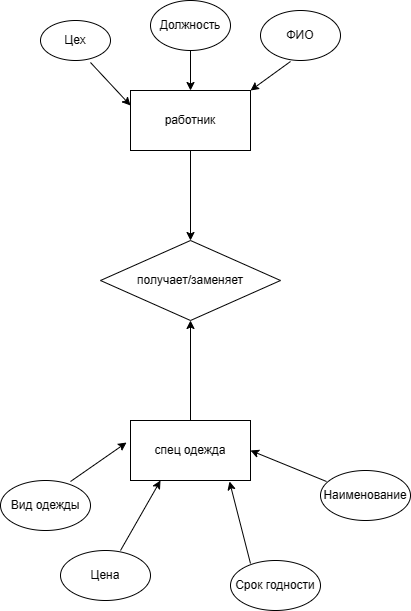


Рис. 2. «Диаграмма "сущность-связь"»

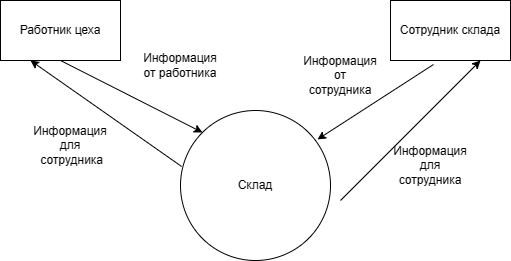


Рис. 3. «Начальная контекстная диаграмма»



Рис. 4. «Диаграмма структур данных»

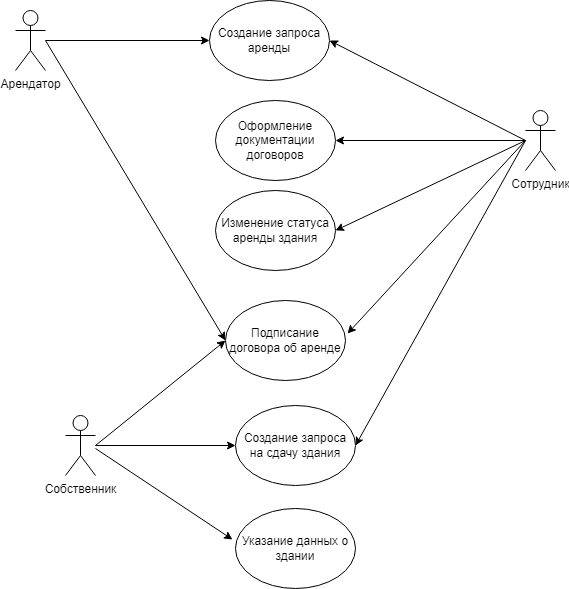


Рис. 5. «UML диаграмма»

## ****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.

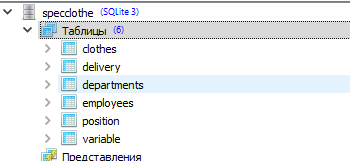


Рис. 6 «Диаграмма базы данных»

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

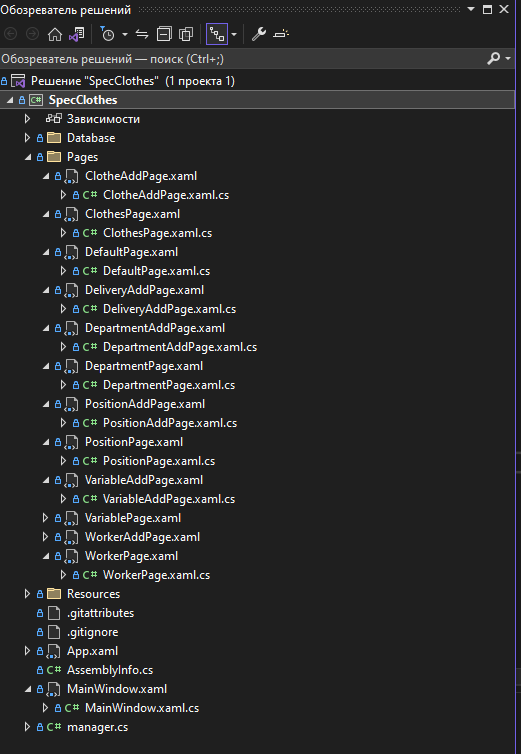


Рис. 7 «Структура проекта»

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:

List<Department> del = SpecclotheContext.GetContext().Departments.ToList();

DGrid.ItemsSource = del;

Код C# кнопки удаления строк из таблиц страницы:

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Department>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{ SpecclotheContext.GetContext().Departments.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Departments.ToList();

}

catch (Exception ex)

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

Код C# поиска записи:

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

{

//поиск

string search = Search.Text;

if (Search.Text != null)

{

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.

Where(x => x.Fio.Contains(search)

|| x.DepartmentsIddepartmentNavigation.Department1.Contains(search)

|| x.PositionIdpositionNavigation.Posi.Contains(search)

|| x.IdEmployees.ToString().Contains(search)).ToList();

}

}

Код C# фильтра поиска по списку страницы:

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

{

try

{

if (Combo.SelectedValue.ToString() == "Виды")

{

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.Count().ToString();

}

else

{

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.

Where(x => x.DepartmentsIddepartmentNavigation.Department1 == Combo.SelectedValue.ToString()).ToList();}

}

catch (Exception ex) { }

}

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 Background="#1A202C">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="9\*"/>

</Grid.ColumnDefinitions>

<Border CornerRadius="15,15,15,15" Margin="10" Background="#364153">

<StackPanel Grid.Column="0" Orientation="Vertical">

<Image Source="pack://application:,,,/Resources/logo.png" Margin="1" Width="60" Height="50" />

<Image Source="pack://application:,,,/Resources/Image1.png" Margin="5" Width="60" Height="50" MouseLeftButtonDown="Gotohomepage" />

<Image Source="pack://application:,,,/Resources/builder-22\_icon-icons.com\_60368.png" Margin="10" Width="40" Height="50" MouseLeftButtonDown="GotoWorkerPage"/>

<Image Source="pack://application:,,,/Resources/1455554770\_line-74\_icon-icons.com\_53368.png" Width="40" Height="50" Margin="10" MouseLeftButtonDown="Image\_MouseLeftButtonDown"/>

</StackPanel>

</Border>

<StackPanel Grid.Column="1">

<TextBlock Name="header" Text="Выдача спец одежды" FontWeight="Bold" FontSize="25" Foreground="GhostWhite"/>

<Frame NavigationUIVisibility="Hidden" Name="MainFrame" Height="399" BorderBrush="Wheat" BorderThickness="0"/>

</StackPanel>

</Grid>

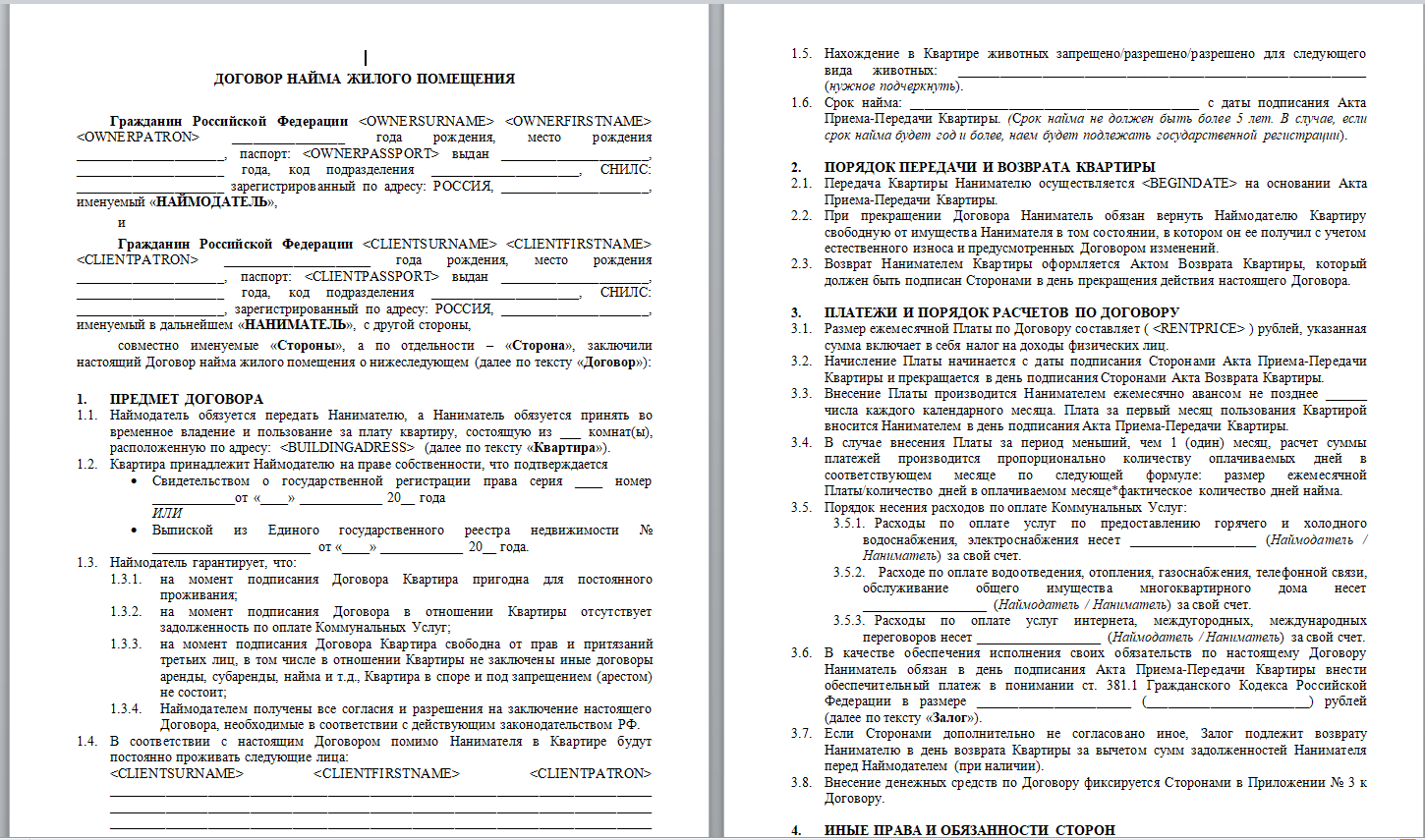
</Window>

Рис. 9 «Шаблонный Документ Word»

## 3.3 Текст программы

Приложение 1.

# 4. Руководство пользователя

## 4.1 Выполнение программного модуля

Для того чтобы воспользоваться представленной программой, необходимо произвести следующие действия: установить программу Visual Studio.

Страница «DefaultPage» выполняет роль главной страницыи содержит кнопки переходов на другие страницы.

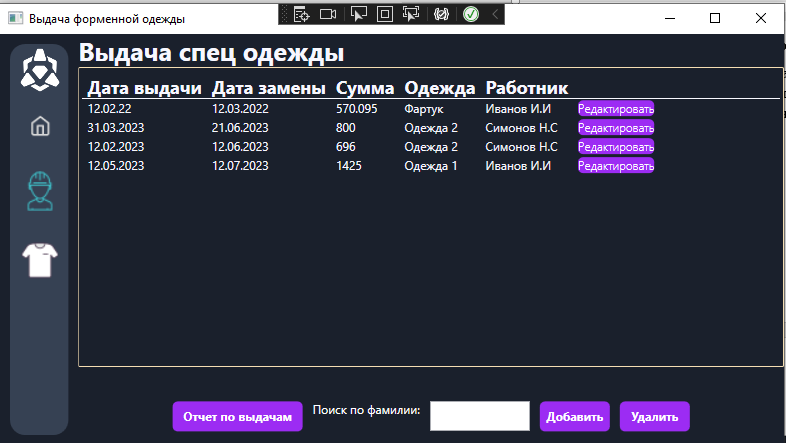


Рис. 10 «Страница DefaultPage»

Страница «WorkerPage*»* выводит данные о сотрудниках(работниках). Можно найти работника по фамилии, или же отсортировав по цеху. Есть возможность добавлять сотрудников, редактировать и удалять.

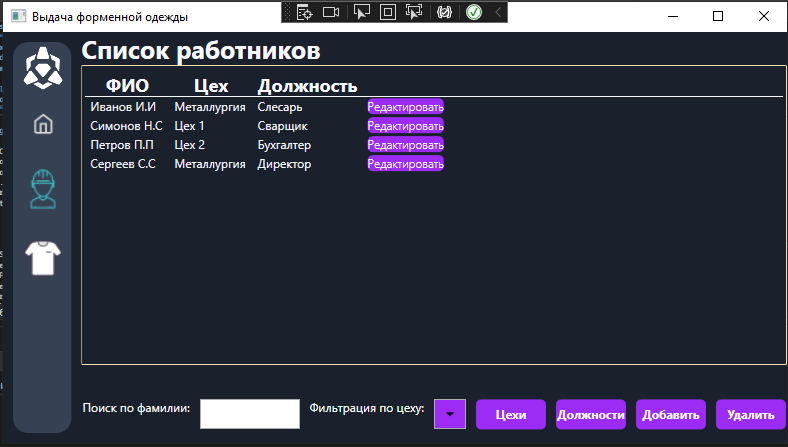


Рис. 11 «Страница ClientsPage»

Страница «WorkerAddPafge*»* в ней вводятся данные о сотруднике, которые затем сохраняются в базу данных.

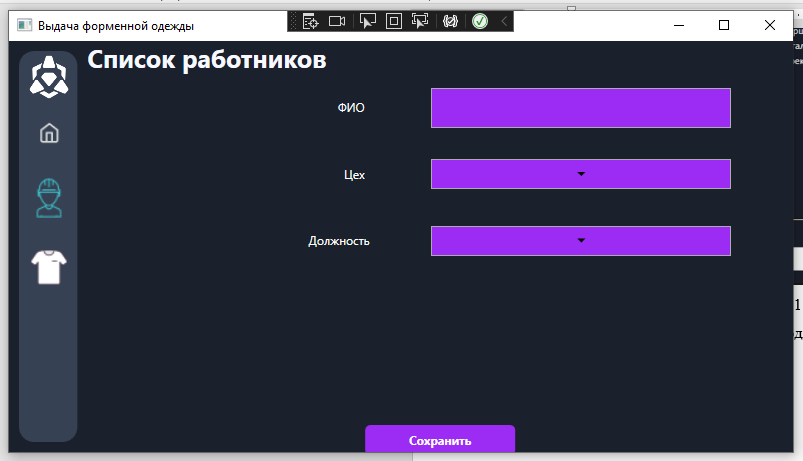


Рис. 12 «Страница ClientsAddEditPage»

Страница «DepartmentPage*»* выводит данные о цехах. Есть возможность добавлять цехи, редактировать и удалять.

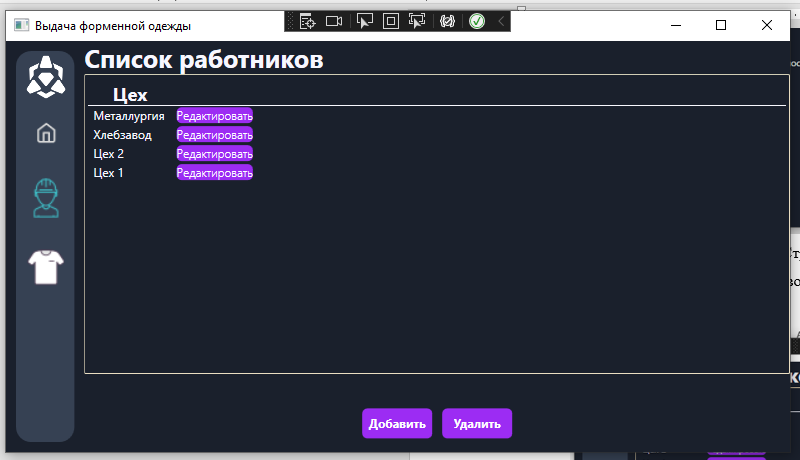


Рис. 13 «Страница BuildingsPage»

Страница «DepartmentAddPage*»,* в ней вводятся данные о цехе, которые затем сохраняются в базу данных.

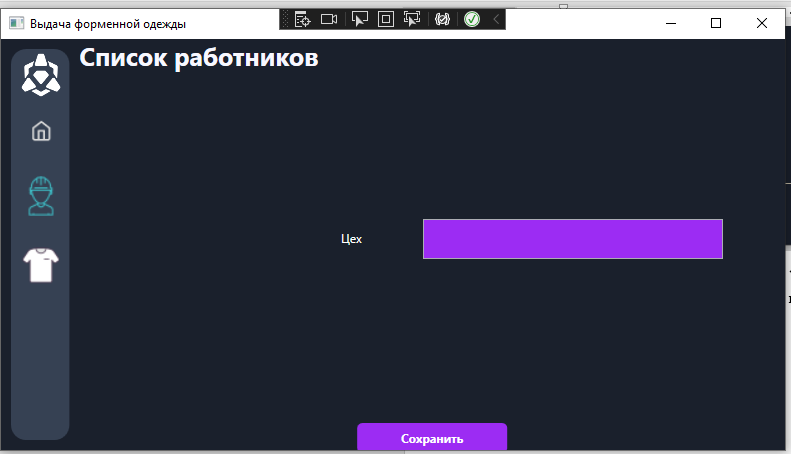


Рис. 14 «Страница BuildingAddEditPage»

Страница «PositionPage*»,* выводит данные о должностях. Есть возможность добавлять должности, редактировать и удалять.

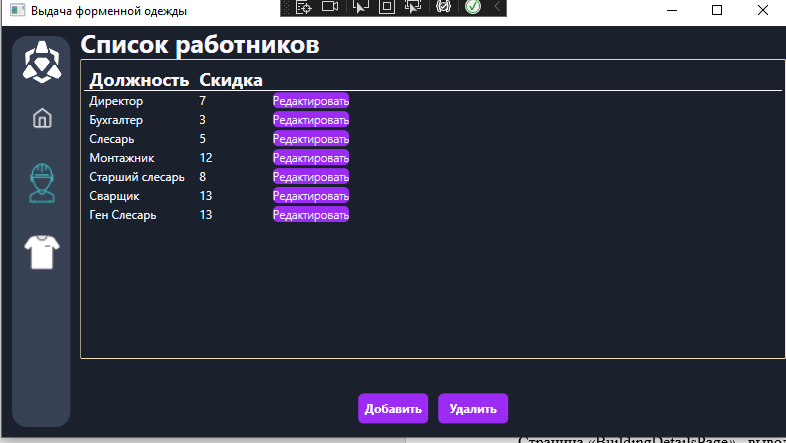


Рис. 15 «Страница BuildingDetailsPage»

Страница «PositionAddPage*»,* в ней вводятся данные о должности, которые затем сохраняются в базу данных.

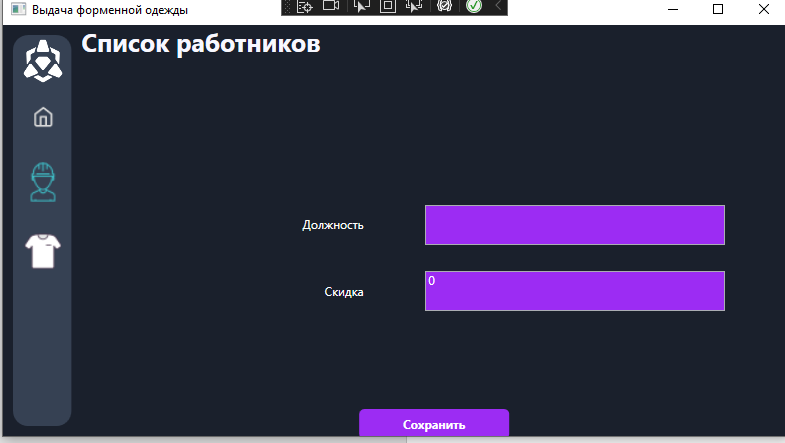


Рис. 16 «Страница EmployeesPage»

Страница «ClothePage*»* выводит данные об одежде. Есть фильтрация по виду одежды. Есть возможность добавлять сотрудников, редактировать и удалять.

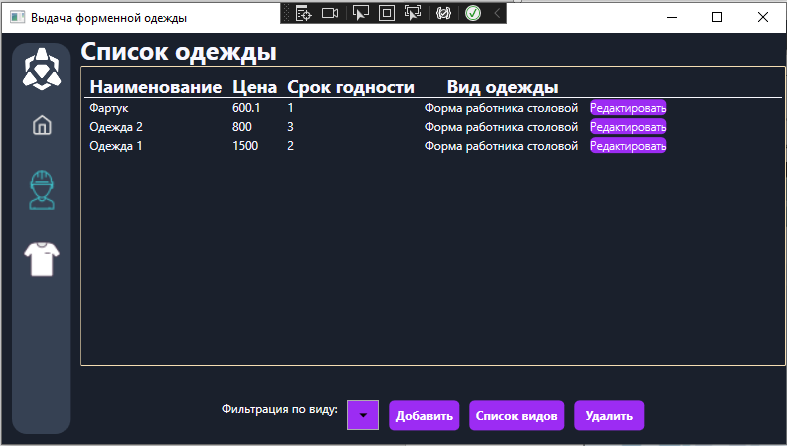


Рис. 17 «Страница EmployeesAddEditPage»

Страница «ClotheAddPage*»,* в ней вводятся данные о должности, которые затем сохраняются в базу данных.

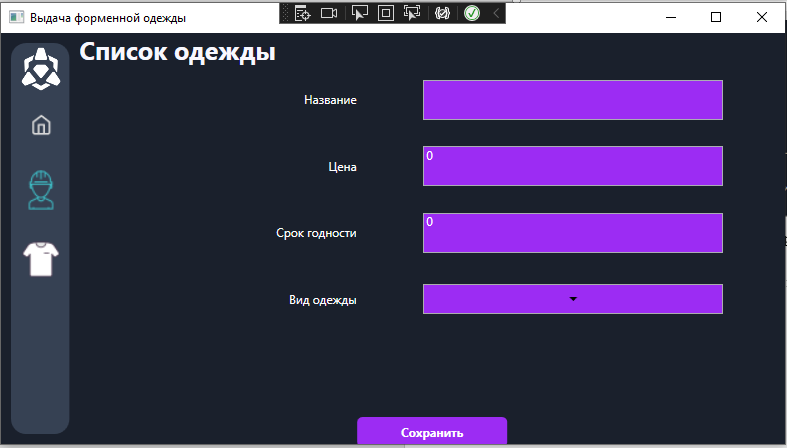


Рис. 18 «Страница OwnersPage»

Страница «VariablePage*»,* выводит данные о видах одежды. Есть возможность добавлять сотрудников, редактировать и удалять.

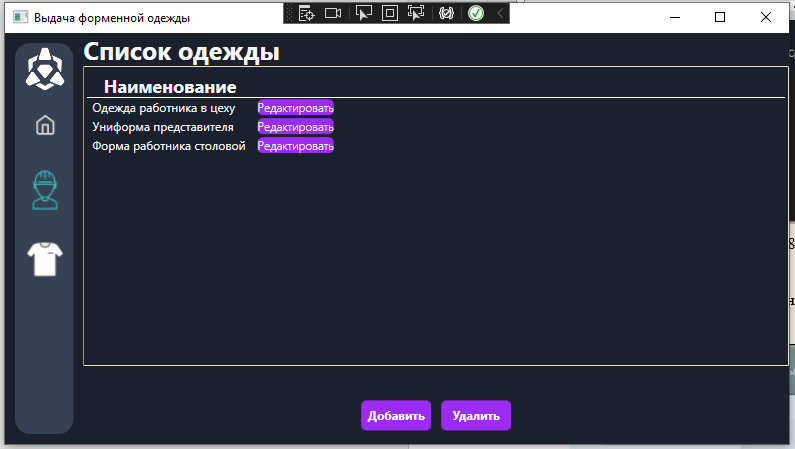


Рис. 19 «Страница OwnerAddEditPage»

Страница «VariableAddPage*»,* в ней вводятся данные видах, которые затем сохраняются в базу данных.

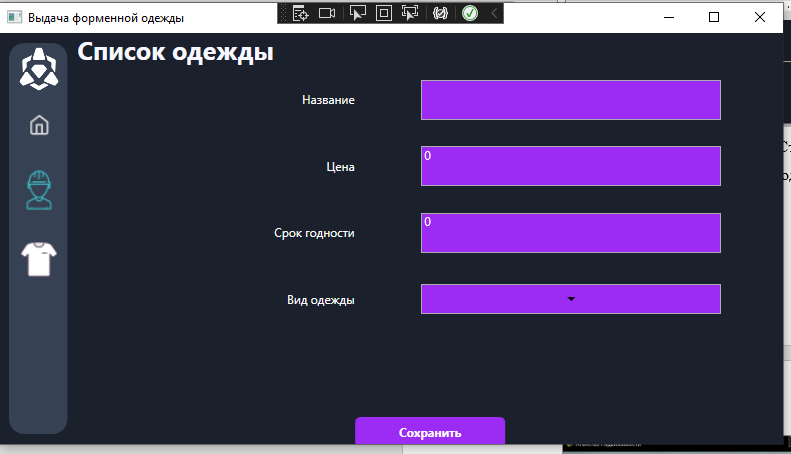


Рис. 20 «Страница ContractsPage»

Страница «DeliveryAddPage» *,* в ней вводятся данные о выдаче одежды, которые затем сохраняются в базу данных.

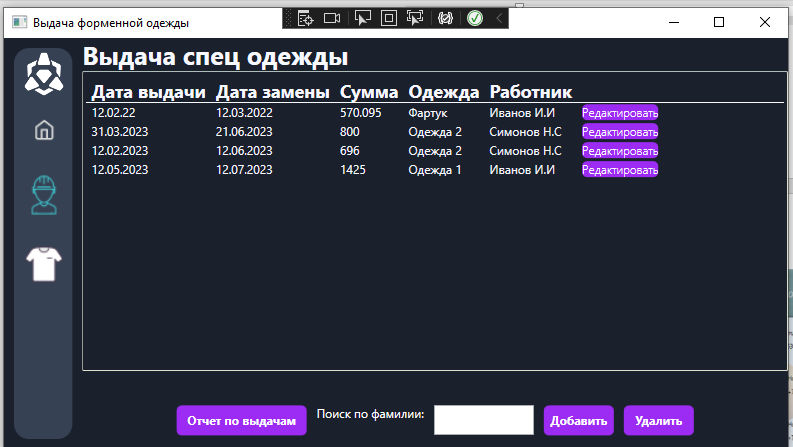


Рис. 21 «Страница PrintPage»

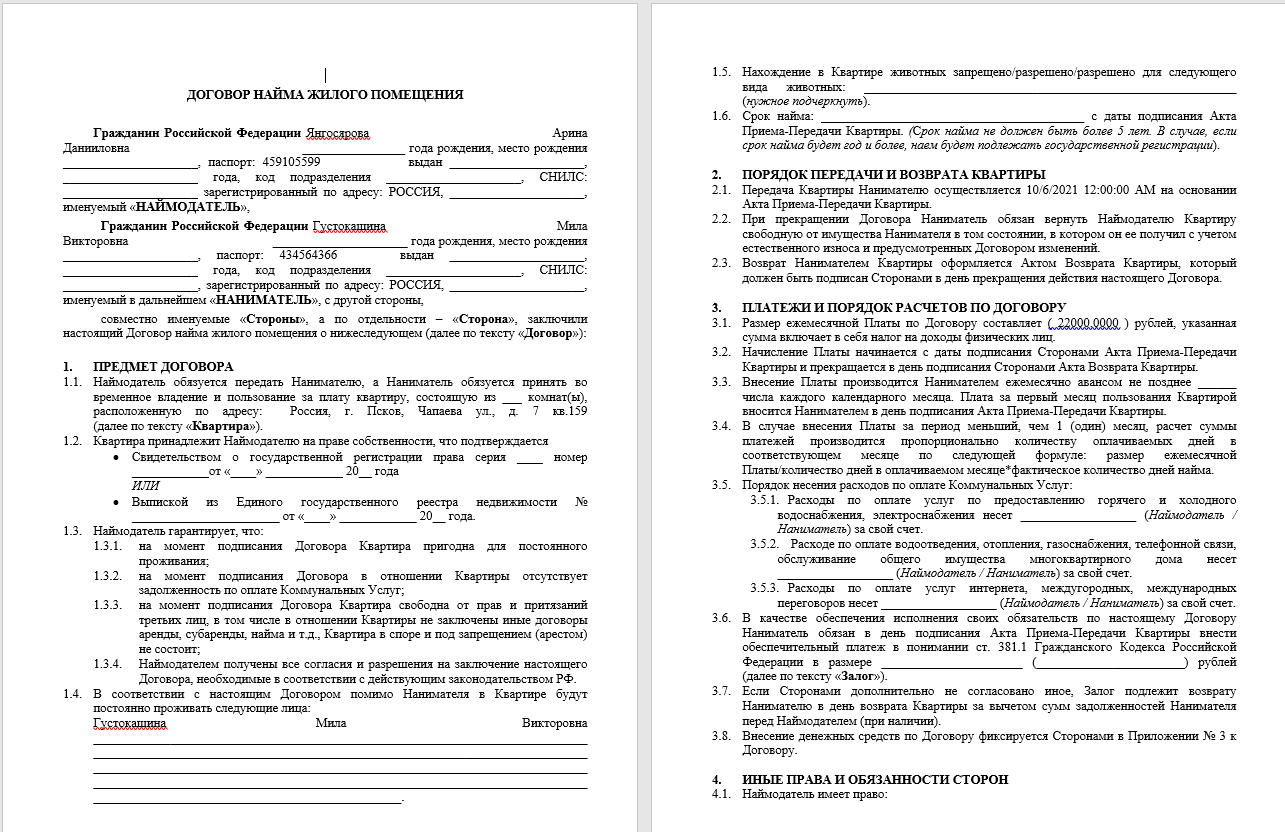


Рис. 22 «Шаблонный Документ Word»

Выпадающий список с вариантами сортировки.



Рис. 23 «Список поиска и сортировки»

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. 24 «Поиск»

Кнопка «Добавить» при нажатии переводит на страницу добавления новой записи в базу данных.



Рис. 25 «Кнопка добавления»

Кнопка «Сохранить» при нажатии сохраняет изменения текущей записи в базу данных.



Рис. 26 «Кнопка сохранения»

Кнопка «Печать» при нажатии переносит данные записи в ранее созданный документ Office Word, для печати в бумажный вариант.



Рис. 27 «Кнопка печати»

## 4.2 Сообщения пользователю

Оповещение в случае удачного добавления/редактирования записи.

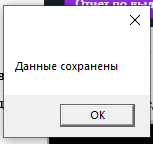


Рис. 28 «Добавление нового рейса»

При удалении записи, потребуется подтверждение действия от пользователя, он должен нажать «Yes» в случае согласия, и «No» в ином случае.

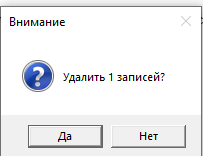


Рис. 29 «Удаление (Выбор)»

Оповещение пользователя о удалении данных из базы данных.

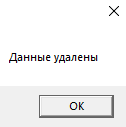


Рис. 30 «Удаление рейса»

В случае если пользователь введёт неверные или неполные данные в поля ввода он получит сообщение с инструкцией.

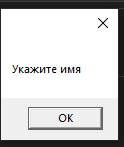


Рис. 31 «Предупреждение»

# 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:

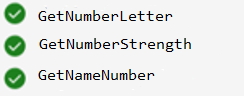


Рис. 32 «Итог тестов»

## 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:



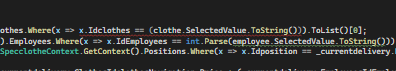
2) Неверный формат вывода сообщения об ошибке



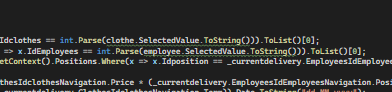
Решение:



3) Неверный тип данных используемой переменной



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.

# Приложение 1:

Код страницы «DefaultPage»

Xaml:

<Page x:Class="SpecClothes.DefaultPage"

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:SpecClothes"

mc:Ignorable="d"

d:DesignHeight="440" d:DesignWidth="800"

Title="DefaultPage">

<Grid Background="#1A202C">

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="Отчет по выдачам" Margin="5" Width="130" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Ochjet\_click"/>

<TextBlock Text="Поиск по фамилии:" Foreground="White" FontSize="12" Height="30" Margin="5"/>

<TextBox Name="Search" Width="100" Height="30" Margin="5" TextChanged="TxtSearch\_TextChanged"/>

<Button Content="Добавить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click"/>

<Button Content="Удалить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_1"/>

</StackPanel>

<Grid>

<ListView Background="#1A202C" Foreground="GhostWhite" BorderBrush="Wheat" Name="DGrid" SelectionChanged="DGridHotels\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="FontSize" Value="12"/>

<Setter Property="Background" Value="#1A202C"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Style.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

</Style.Triggers>

</Style>

</ListView.ItemContainerStyle>

<ListView.View>

<GridView>

<GridViewColumn Header="Дата выдачи" DisplayMemberBinding="{Binding Datato}" Width="Auto" />

<GridViewColumn Header="Дата замены" DisplayMemberBinding="{Binding Datatrade}" Width="Auto"/>

<GridViewColumn Header="Сумма" DisplayMemberBinding="{Binding Price}" Width="Auto"/>

<GridViewColumn Header="Одежда" DisplayMemberBinding="{Binding ClothesIdclothesNavigation.Clothe1}" Width="Auto"/>

<GridViewColumn Header="Работник" DisplayMemberBinding="{Binding EmployeesIdEmployeesNavigation.Fio}" Width="Auto"/>

<GridViewColumn>

<GridViewColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Foreground="White" FontWeight="ExtraLight" Background="#9C2CF3

" Name ="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Grid>

</Page>

С#:

public partial class DefaultPage : Page

{

public DefaultPage()

{

InitializeComponent();

List<Delivery> del = SpecclotheContext.GetContext().Deliveries.ToList();

foreach (var d in del)

{

d.ClothesIdclothesNavigation = SpecclotheContext.GetContext().Clothes.Where(x => x.Idclothes == d.ClothesIdclothes).ToList()[0];

d.EmployeesIdEmployeesNavigation = SpecclotheContext.GetContext().Employees.Where(x => x.IdEmployees == d.EmployeesIdEmployees).ToList()[0];

}

DGrid.ItemsSource = del;

}

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

{

//поиск

string search = Search.Text;

if (Search.Text != null)

{

List<Delivery> del = SpecclotheContext.GetContext().Deliveries.ToList();

foreach (var d in del)

{

d.ClothesIdclothesNavigation = SpecclotheContext.GetContext().Clothes.Where(x => x.Idclothes == d.ClothesIdclothes).ToList()[0];

d.EmployeesIdEmployeesNavigation = SpecclotheContext.GetContext().Employees.Where(x => x.IdEmployees == d.EmployeesIdEmployees).ToList()[0];

}

DGrid.ItemsSource = del.

Where(x => x.ClothesIdclothesNavigation.Clothe1.Contains(search)

|| x.EmployeesIdEmployeesNavigation.Fio.Contains(search)

|| x.Datato.Contains(search)

|| x.Datatrade.ToString().Contains(search)).ToList();

}

}

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

{

manager.MainFrame.Navigate(new DeliveryAddPage((sender as Button).DataContext as Delivery));

}

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

{

}

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

{

manager.MainFrame.Navigate(new DeliveryAddPage(null));

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Delivery>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Deliveries.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Deliveries.ToList();

}

catch (Exception ex)

{

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

}

}

private void Ochjet\_click(object sender, RoutedEventArgs e)

{

}

}

}

Код страницы «DeliveryAddPage»

Xaml:

<Page x:Class="SpecClothes.DeliveryAddPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="DeliveryAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Дата выдачи"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="0"

Grid.Column="0"

Margin="5"/>

<TextBlock Text="Одежда"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="1"

Grid.Column="0"/>

<TextBlock Text="Работник"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="2"

Grid.Column="0"/>

<TextBox Name="datato"

Background="#9C2CF3"

Foreground="White"

Text="{Binding Datato}"

Width="300"

Grid.Row="0"

Height="40"

Grid.Column="1"/>

<ComboBox Name="clothe"

Grid.Row="1"

Grid.Column="1" Width="300" Background="#9C2CF3"

SelectedValue="{Binding ClothesIdclothes}"/>

<ComboBox Name="employee"

Grid.Row="2"

Grid.Column="1" Width="300" Background="#9C2CF3"

SelectedValue="{Binding EmployeesIdEmployees}"/>

<Button Content="Сохранить"

Background="#9C2CF3"

Foreground="White"

FontWeight="Bold"

Grid.ColumnSpan="2"

Grid.Row="5"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</Grid>

</Page>

С#:

public partial class DeliveryAddPage : Page

{

private Delivery \_currentdelivery = new Delivery();

public DeliveryAddPage(Delivery selecteddelivery)

{

InitializeComponent();

clothe.ItemsSource = SpecclotheContext.GetContext().Clothes.ToList();

clothe.SelectedValuePath = "Idclothes";

clothe.DisplayMemberPath = "Clothe1";

employee.ItemsSource = SpecclotheContext.GetContext().Employees.ToList();

employee.SelectedValuePath = "IdEmployees";

employee.DisplayMemberPath = "Fio";

if (selecteddelivery != null)

\_currentdelivery = selecteddelivery;

//создаем контекст

DataContext = \_currentdelivery;

}

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

{

\_currentdelivery.ClothesIdclothesNavigation = SpecclotheContext.GetContext().Clothes.Where(x => x.Idclothes == int.Parse(clothe.SelectedValue.ToString())).ToList()[0];

\_currentdelivery.EmployeesIdEmployeesNavigation = SpecclotheContext.GetContext().Employees.Where(x => x.IdEmployees == int.Parse(employee.SelectedValue.ToString())).ToList()[0];

\_currentdelivery.EmployeesIdEmployeesNavigation.PositionIdpositionNavigation = SpecclotheContext.GetContext().Positions.Where(x => x.Idposition == \_currentdelivery.EmployeesIdEmployeesNavigation.PositionIdposition).ToList()[0];

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

\_currentdelivery.Price = \_currentdelivery.ClothesIdclothesNavigation.Price - (\_currentdelivery.ClothesIdclothesNavigation.Price \* (\_currentdelivery.EmployeesIdEmployeesNavigation.PositionIdpositionNavigation.Discount / 100));

\_currentdelivery.Datatrade =Convert.ToDateTime(\_currentdelivery.Datato).AddMonths(Convert.ToInt32(\_currentdelivery.ClothesIdclothesNavigation.Term)).Date.ToString("dd.MM.yyyy");

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentdelivery.Datato))

error.AppendLine("Укажите Дату выдачи");

if (string.IsNullOrWhiteSpace(\_currentdelivery.Datatrade.ToString()))

error.AppendLine("Укажите Дату замены");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentdelivery.Iddelivery == 0)

SpecclotheContext.GetContext().Deliveries.Add(\_currentdelivery); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new WorkerPage());

}

catch (Exception ex)

{

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

}

}

}

}

Код страницы «ClothesPage»

Xaml:

<Page x:Class="SpecClothes.ClothesPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="ClothesPage">

<Grid Background="#1A202C">

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Фильтрация по виду:" Foreground="White" FontSize="12" Height="30" Margin="5"/>

<ComboBox Name="Combo" SelectedItem="{Binding Selected}" Background="#9C2CF3" Foreground="White" Margin="5" Height="30" Width="Auto" SelectionChanged="Combo\_SelectionChanged"/>

<Button Content="Добавить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click"/>

<Button Content="Список видов" Margin="5" Width="95" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_1"/>

<Button Content="Удалить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="BtnDelete\_Click" />

</StackPanel>

<Grid>

<ListView Background="#1A202C" Foreground="GhostWhite" BorderBrush="Wheat" Name="DGrid" IsEnabled="True" SelectionChanged="DGridHotels\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="FontSize" Value="12"/>

<Setter Property="Background" Value="#1A202C"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Style.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

</Style.Triggers>

</Style>

</ListView.ItemContainerStyle>

<ListView.View>

<GridView>

<GridViewColumn Header="Наименование" DisplayMemberBinding="{Binding Clothe1}" Width="Auto" />

<GridViewColumn Header="Цена" DisplayMemberBinding="{Binding Price}" Width="Auto"/>

<GridViewColumn Header="Срок годности" DisplayMemberBinding="{Binding Term}" Width="Auto"/>

<GridViewColumn Header="Вид одежды" DisplayMemberBinding="{Binding VariableIdVariableNavigation.Variable1}" Width="Auto"/>

<GridViewColumn>

<GridViewColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Foreground="White" FontWeight="ExtraLight" Background="#9C2CF3

" Name ="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Grid>

</Page>

С#:

public partial class ClothesPage : Page

{

public ClothesPage()

{

InitializeComponent();

List<Clothe> del = SpecclotheContext.GetContext().Clothes.ToList();

foreach (var d in del)

{

d.VariableIdVariableNavigation = SpecclotheContext.GetContext().Variables.Where(x => x.IdVariable == d.VariableIdVariable).ToList()[0];

}

DGrid.ItemsSource = del;

Combo.Items.Add("Виды");

foreach (var item in SpecclotheContext.GetContext().Clothes.

Select(x => x.VariableIdVariableNavigation.Variable1).Distinct().ToList())

{

Combo.Items.Add(item);

}

}

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

{

try

{

if (Combo.SelectedValue.ToString() == "Виды")

{

List<Clothe> del = SpecclotheContext.GetContext().Clothes.ToList();

foreach (var d in del)

{

d.VariableIdVariableNavigation = SpecclotheContext.GetContext().Variables.Where(x => x.IdVariable == d.VariableIdVariable).ToList()[0];

}

DGrid.ItemsSource = del;

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.Count().ToString();

}

else

{

List<Clothe> del = SpecclotheContext.GetContext().Clothes.ToList();

foreach (var d in del)

{

d.VariableIdVariableNavigation = SpecclotheContext.GetContext().Variables.Where(x => x.IdVariable == d.VariableIdVariable).ToList()[0];

}

DGrid.ItemsSource = del;

DGrid.ItemsSource = SpecclotheContext.GetContext().Clothes.

Where(x => x.VariableIdVariableNavigation.Variable1 == Combo.SelectedValue.ToString()).ToList();

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.

// Where(x => x.LastName == CmbFiltr.SelectedValue.ToString()).Count().ToString();

}

}

catch(Exception ex) { }

}

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

{

manager.MainFrame.Navigate(new ClotheAddPage((sender as Button).DataContext as Clothe));

}

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

{

}

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

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Clothe>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Clothes.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Clothes.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

manager.MainFrame.Navigate(new ClotheAddPage(null));

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new VariablePage());

}

}

}

Код страницы «ClothesAddPage»

Xaml:

<Page x:Class="SpecClothes.ClotheAddPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="ClotheAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Название"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="0"

Grid.Column="0"

Margin="5"/>

<TextBlock Text="Цена"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="1"

Grid.Column="0"

Margin="5"/>

<TextBlock Text="Срок годности"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="2"

Grid.Column="0"

Margin="5"/>

<TextBlock Text="Вид одежды"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="3"

Grid.Column="0"

Margin="5"/>

<TextBox Name="TxtFirstName"

Background="#9C2CF3"

Foreground="White"

Text="{Binding Clothe1}"

Width="300"

Grid.Row="0"

Height="40"

Grid.Column="1"/>

<TextBox Name="TxtLastName"

Foreground="White"

Background="#9C2CF3"

Text="{Binding Price}"

Width="300"

Height="40"

Grid.Row="1"

Grid.Column="1"/>

<TextBox Name="TxtPhone"

Foreground="White"

Background="#9C2CF3"

Text="{Binding Term}"

Grid.Row="2" Height="40"

Width="300"

Grid.Column="1"/>

<ComboBox Name="CmbLogin"

Grid.Row="3"

Grid.Column="1" Width="300" Background="#9C2CF3"

SelectedValue="{Binding VariableIdVariable}"/>

<Button Content="Сохранить"

Background="#9C2CF3"

Foreground="White"

FontWeight="Bold"

Grid.ColumnSpan="2"

Grid.Row="5"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</Grid>

</Page>

С#:

public partial class ClotheAddPage : Page

{

private Clothe \_currentclothe = new Clothe();

public ClotheAddPage(Clothe selectedclothe)

{

InitializeComponent();

CmbLogin.ItemsSource = SpecclotheContext.GetContext().Variables.ToList();

CmbLogin.SelectedValuePath = "IdVariable";

CmbLogin.DisplayMemberPath = "Variable1";

if (selectedclothe != null)

\_currentclothe = selectedclothe;

//создаем контекст

DataContext = \_currentclothe;

}

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

{

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentclothe.Clothe1))

error.AppendLine("Укажите имя");

if (string.IsNullOrWhiteSpace(\_currentclothe.Term.ToString()))

error.AppendLine("Укажите фамилию");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentclothe.Idclothes == 0)

SpecclotheContext.GetContext().Clothes.Add(\_currentclothe); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new ClothesPage());

}

catch (Exception ex)

{

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

}

}

}

}

Код страницы «DepartmentPage»

Xaml:

<Page x:Class="SpecClothes.DepartmentPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="DepartmentPage">

<Grid Background="#1A202C">

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<Button Content="Добавить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click"/>

<Button Content="Удалить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_3"/>

</StackPanel>

<Grid>

<ListView Background="#1A202C" Foreground="GhostWhite" BorderBrush="Wheat" Name="DGrid" SelectionChanged="DGridHotels\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="FontSize" Value="12"/>

<Setter Property="Background" Value="#1A202C"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Style.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

</Style.Triggers>

</Style>

</ListView.ItemContainerStyle>

<ListView.View>

<GridView>

<GridViewColumn Header="Цех" DisplayMemberBinding="{Binding Department1}" Width="Auto"/>

<GridViewColumn>

<GridViewColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Foreground="White" FontWeight="ExtraLight" Background="#9C2CF3

" Name ="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Grid>

</Page>

С#:

public partial class DepartmentPage : Page

{

public DepartmentPage()

{

InitializeComponent();

List<Department> del = SpecclotheContext.GetContext().Departments.ToList();

DGrid.ItemsSource = del;

}

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

{

manager.MainFrame.Navigate(new DepartmentAddPage((sender as Button).DataContext as Department));

}

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

{

}

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

{

manager.MainFrame.Navigate(new DepartmentAddPage(null));

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new PositionPage());

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new PositionPage());

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Department>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Departments.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Departments.ToList();

}

catch (Exception ex)

{

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

}

}

}

}

}

Код страницы «DepartmentAddPage»

Xaml:

<Page x:Class="SpecClothes.DepartmentAddPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="DepartmentAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Цех"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="2"

Grid.Column="0"/>

<TextBox Name="TxtFirstName"

Background="#9C2CF3"

Foreground="White"

Text="{Binding Department1}"

Width="300"

Grid.Row="2"

Height="40"

Grid.Column="3"/>

<Button Content="Сохранить"

Background="#9C2CF3"

Foreground="White"

FontWeight="Bold"

Grid.ColumnSpan="2"

Grid.Row="5"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</Grid>

</Page>

С#:

public partial class DepartmentAddPage : Page

{

private Department \_currentdepartment = new Department();

public DepartmentAddPage(Department currentdepartment)

{

InitializeComponent();

if (currentdepartment != null)

\_currentdepartment = currentdepartment;

//создаем контекст

DataContext = \_currentdepartment;

}

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

{

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentdepartment.Department1))

error.AppendLine("Укажите имя");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentdepartment.Iddepartment == 0)

SpecclotheContext.GetContext().Departments.Add(\_currentdepartment); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new WorkerPage());

}

catch (Exception ex)

{

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

}

}

}

}

Код страницы «PositionPage»

Xaml:

<Page x:Class="SpecClothes.PositionPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="PositionPage">

<Grid Background="#1A202C">

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<Button Content="Добавить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click"/>

<Button Content="Удалить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_2"/>

</StackPanel>

<Grid>

<ListView Background="#1A202C" Foreground="GhostWhite" BorderBrush="Wheat" Name="DGrid" SelectionChanged="DGridHotels\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="FontSize" Value="12"/>

<Setter Property="Background" Value="#1A202C"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Style.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

</Style.Triggers>

</Style>

</ListView.ItemContainerStyle>

<ListView.View>

<GridView>

<GridViewColumn Header="Должность" DisplayMemberBinding="{Binding Posi}" Width="Auto"/>

<GridViewColumn Header="Скидка" DisplayMemberBinding="{Binding Discount}" Width="Auto"/>

<GridViewColumn>

<GridViewColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Foreground="White" FontWeight="ExtraLight" Background="#9C2CF3

" Name ="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Grid>

</Page>

С#:

public partial class PositionPage : Page

{

public PositionPage()

{

InitializeComponent();

List<Position> del = SpecclotheContext.GetContext().Positions.ToList();

DGrid.ItemsSource = del;

}

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

{

manager.MainFrame.Navigate(new PositionAddPage((sender as Button).DataContext as Position));

}

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

{

}

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

{

manager.MainFrame.Navigate(new PositionAddPage(null));

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new PositionPage());

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Position>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Positions.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Positions.ToList();

}

catch (Exception ex)

{

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

}

}

}

}

}

Код страницы «PositionAddPage»

Xaml:

<Page x:Class="SpecClothes.PositionAddPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="PositionAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Должность"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="2"

Grid.Column="0"/>

<TextBlock Text="Скидка"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="3"

Grid.Column="0"/>

<TextBox Name="TxtFirstName"

Background="#9C2CF3"

Foreground="White"

Text="{Binding Posi}"

Width="300"

Grid.Row="2"

Height="40"

Grid.Column="3"/>

<TextBox Name="Dis"

Background="#9C2CF3"

Foreground="White"

Text="{Binding Discount}"

Width="300"

Grid.Row="3"

Height="40"

Grid.Column="3"/>

<Button Content="Сохранить"

Background="#9C2CF3"

Foreground="White"

FontWeight="Bold"

Grid.ColumnSpan="2"

Grid.Row="5"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</Grid>

</Page>

С#:

public partial class PositionAddPage : Page

{

private Position \_currentposition = new Position();

public PositionAddPage(Position currentposition)

{

InitializeComponent();

if (currentposition != null)

\_currentposition = currentposition;

//создаем контекст

DataContext = \_currentposition;

}

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

{

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentposition.Posi))

error.AppendLine("Укажите имя");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentposition.Idposition == 0)

SpecclotheContext.GetContext().Positions.Add(\_currentposition); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new WorkerPage());

}

catch (Exception ex)

{

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

}

}

}

}

Код страницы «VariablePage»

Xaml:

<Page x:Class="SpecClothes.VariablePage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="VariablePage">

<Grid Background="#1A202C">

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<Button Content="Добавить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click"/>

<Button Content="Удалить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="BtnDelete\_Click" />

</StackPanel>

<Grid>

<ListView Background="#1A202C" Foreground="GhostWhite" BorderBrush="Wheat" Name="DGrid" IsEnabled="True" SelectionChanged="DGridHotels\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="FontSize" Value="12"/>

<Setter Property="Background" Value="#1A202C"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Style.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

</Style.Triggers>

</Style>

</ListView.ItemContainerStyle>

<ListView.View>

<GridView>

<GridViewColumn Header="Наименование" DisplayMemberBinding="{Binding Variable1}" Width="Auto" />

<GridViewColumn>

<GridViewColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Foreground="White" FontWeight="ExtraLight" Background="#9C2CF3

" Name ="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Grid>

</Page>

С#:

public partial class VariablePage : Page

{

public VariablePage()

{

InitializeComponent();

List<Variable> del = SpecclotheContext.GetContext().Variables.ToList();

DGrid.ItemsSource = del;

}

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

{

manager.MainFrame.Navigate(new VariableAddPage((sender as Button).DataContext as Variable));

}

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

{

}

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

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Variable>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Variables.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Variables.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

manager.MainFrame.Navigate(new VariableAddPage(null));

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new VariablePage());

}

}

}

Код страницы «VariableAddPage»

Xaml:

<Page x:Class="SpecClothes.VariableAddPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="VariableAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Название"

Grid.Row="2"

Grid.Column="0"/>

<TextBox Name="TxtFirstName"

Text="{Binding Variable1}"

Grid.Row="2"

Grid.Column="1"/>

<Button Content="Сохранить"

Grid.ColumnSpan="2"

Grid.Row="5"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</Grid>

</Page>

С#:

public partial class VariableAddPage : Page

{

private Variable \_currentvariable = new Variable();

public VariableAddPage(Variable selectedvariable)

{

InitializeComponent();

if (selectedvariable != null)

\_currentvariable = selectedvariable;

//создаем контекст

DataContext = \_currentvariable;

}

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

{

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentvariable.Variable1))

error.AppendLine("Укажите имя");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentvariable.IdVariable == 0)

SpecclotheContext.GetContext().Variables.Add(\_currentvariable); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new ClothesPage());

}

catch (Exception ex)

{

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

}

}

}

}

Код страницы «WorkerPage»

Xaml:

<Page x:Class="SpecClothes.WorkerPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="WorkerPage">

<Grid Background="#1A202C">

<Grid.RowDefinitions>

<RowDefinition Height="3\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Поиск по фамилии:" Foreground="White" FontSize="12" Height="30" Margin="5"/>

<TextBox Name="Search" TextChanged="TxtSearch\_TextChanged" Width="100" Height="30" Margin="5"/>

<TextBlock Text="Фильтрация по цеху:" Foreground="White" FontSize="12" Height="30" Margin="5"/>

<ComboBox Name="Combo" SelectedItem="{Binding Selected}" Background="#9C2CF3" Margin="5" Width="Auto" SelectionChanged="Combo\_SelectionChanged"/>

<Button Content="Цехи" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_3"/>

<Button Content="Должности" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_2"/>

<Button Content="Добавить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click"/>

<Button Content="Удалить" Margin="5" Width="70" Foreground="White" FontWeight="Bold" Background="#9C2CF3" Height="30" Click="Button\_Click\_4"/>

</StackPanel>

<Grid>

<ListView Background="#1A202C" Foreground="GhostWhite" BorderBrush="Wheat" Name="DGrid" SelectionChanged="DGridHotels\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="FontSize" Value="12"/>

<Setter Property="Background" Value="#1A202C"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0"/>

<Style.Triggers>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightBlue"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

</Style.Triggers>

</Style>

</ListView.ItemContainerStyle>

<ListView.View>

<GridView>

<GridViewColumn Header="ФИО" DisplayMemberBinding="{Binding Fio}"/>

<GridViewColumn Header="Цех" DisplayMemberBinding="{Binding DepartmentsIddepartmentNavigation.Department1}" Width="Auto"/>

<GridViewColumn Header="Должность" DisplayMemberBinding="{Binding PositionIdpositionNavigation.Posi}" Width="Auto"/>

<GridViewColumn>

<GridViewColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Foreground="White" FontWeight="ExtraLight" Background="#9C2CF3

" Name ="BtnEdit" Click="BtnEdit\_Click"/>

</DataTemplate>

</GridViewColumn.CellTemplate>

</GridViewColumn>

</GridView>

</ListView.View>

</ListView>

</Grid>

</Grid>

</Page>

С#:

public partial class WorkerPage : Page

{

public WorkerPage()

{

InitializeComponent();

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.PositionIdpositionNavigation = SpecclotheContext.GetContext().Positions.Where(x => x.Idposition == d.PositionIdposition).ToList()[0];

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

Combo.Items.Add("Виды");

foreach (var item in SpecclotheContext.GetContext().Employees.

Select(x => x.DepartmentsIddepartmentNavigation.Department1).Distinct().ToList())

{

Combo.Items.Add(item);

}

}

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

{

manager.MainFrame.Navigate(new WorkerAddPage((sender as Button).DataContext as Employee));

}

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

{

}

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

{

//поиск

string search = Search.Text;

if (Search.Text != null)

{

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.

Where(x => x.Fio.Contains(search)

|| x.DepartmentsIddepartmentNavigation.Department1.Contains(search)

|| x.PositionIdpositionNavigation.Posi.Contains(search)

|| x.IdEmployees.ToString().Contains(search)).ToList();

}

}

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

{

try

{

if (Combo.SelectedValue.ToString() == "Виды")

{

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.Count().ToString();

}

else

{

List<Employee> del = SpecclotheContext.GetContext().Employees.ToList();

foreach (var d in del)

{

d.DepartmentsIddepartmentNavigation = SpecclotheContext.GetContext().Departments.Where(x => x.Iddepartment == d.DepartmentsIddepartment).ToList()[0];

}

DGrid.ItemsSource = del;

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.

Where(x => x.DepartmentsIddepartmentNavigation.Department1 == Combo.SelectedValue.ToString()).ToList();

//TxbCountSearchItem.Text = dbISP19AEntities.GetContext().User.

// Where(x => x.LastName == CmbFiltr.SelectedValue.ToString()).Count().ToString();

}

}

catch (Exception ex) { }

}

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

{

manager.MainFrame.Navigate(new WorkerAddPage(null));

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new PositionPage());

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new PositionPage());

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

manager.MainFrame.Navigate(new DepartmentPage());

}

private void Button\_Click\_4(object sender, RoutedEventArgs e)

{

{// удаление нескольких пользователей

var usersForRemoving = DGrid.SelectedItems.Cast<Employee>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} записей?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

SpecclotheContext.GetContext().Employees.RemoveRange(usersForRemoving);

SpecclotheContext.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

DGrid.ItemsSource = SpecclotheContext.GetContext().Employees.ToList();

}

catch (Exception ex)

{

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

}

}

}

}

}

Код страницы «WorkerAddPage»

Xaml:

<Page x:Class="SpecClothes.WorkerAddPage"

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:SpecClothes"

mc:Ignorable="d"

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

Title="WorkerAddPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="3\*"/>

</Grid.ColumnDefinitions>

<TextBlock Text="ФИО"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="0"

Grid.Column="0"

Margin="5"/>

<TextBlock Text="Цех"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="1"

Grid.Column="0"

Margin="5"/>

<TextBlock Text="Должность"

Foreground="White"

HorizontalAlignment="Right"

VerticalAlignment="Center"

Grid.Row="2"

Grid.Column="0"/>

<TextBox Name="TxtFirstName"

Background="#9C2CF3"

Foreground="White"

Text="{Binding Fio}"

Width="300"

Grid.Row="0"

Height="40"

Grid.Column="1"/>

<ComboBox Name="CmbDep"

Grid.Row="1"

Grid.Column="1" Width="300" Background="#9C2CF3"

SelectedValue="{Binding DepartmentsIddepartment }"/>

<ComboBox Name="CmbPos"

Grid.Row="2"

Grid.Column="1" Width="300" Background="#9C2CF3"

SelectedValue="{Binding PositionIdposition}"/>

<Button Content="Сохранить"

Background="#9C2CF3"

Foreground="White"

FontWeight="Bold"

Grid.ColumnSpan="2"

Grid.Row="5"

Width="150"

Height="30"

Name="BtnSave"

Click="BtnSave\_Click"/>

</Grid>

</Page>

С#:

public partial class WorkerAddPage : Page

{

private Employee \_currentemployee = new Employee();

public WorkerAddPage(Employee selectedemployee)

{

InitializeComponent();

CmbPos.ItemsSource = SpecclotheContext.GetContext().Positions.ToList();

CmbPos.SelectedValuePath = "Idposition";

CmbPos.DisplayMemberPath = "Posi";

CmbDep.ItemsSource = SpecclotheContext.GetContext().Departments.ToList();

CmbDep.SelectedValuePath = "Iddepartment";

CmbDep.DisplayMemberPath = "Department1";

if (selectedemployee != null)

\_currentemployee = selectedemployee;

//создаем контекст

DataContext = \_currentemployee;

}

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

{

StringBuilder error = new StringBuilder(); //объект для сообщения об ошибке

//проверка полей объекта

if (string.IsNullOrWhiteSpace(\_currentemployee.Fio))

error.AppendLine("Укажите имя");

if (string.IsNullOrWhiteSpace(\_currentemployee.PositionIdposition.ToString()))

error.AppendLine("Укажите фамилию");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

//если пользователь новый

if (\_currentemployee.IdEmployees == 0)

SpecclotheContext.GetContext().Employees.Add(\_currentemployee); //добавить в контекст

try

{

SpecclotheContext.GetContext().SaveChanges(); // сохранить изменения

// dbISP19AEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

MessageBox.Show("Данные сохранены");

manager.MainFrame.Navigate(new WorkerPage());

}

catch (Exception ex)

{

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

}

}

}

}