Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Промышленно-экономический колледж ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

«Разработка программных модулей для информационной системы "Поликлиника"»

МДК 01.01 «Разработка программных модулей»

**Выполнил:**

Жаворонков Владислав Владимирович\_\_\_\_\_\_\_\_\_\_\_\_

студент группы ИСП.21.1А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

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

Подпись руководителя

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

Орехово-Зуево

2024 год

Оглавление

[**Введение.** 3](#_Toc161166424)

[**1.** **Назначение разработки.** 5](#_Toc161166425)

[**2.** **Требования к программному модулю.** 8](#_Toc161166426)

[**2.1** **Требования к функциональным характеристикам.** 8](#_Toc161166427)

[**2.2** **Требования к составу и параметрам технических средств.** 9](#_Toc161166428)

[**2.3** **Требования к информационной и программной совместимости.** 9](#_Toc161166429)

[**3.** **Руководство программиста.** 10](#_Toc161166430)

[**3.1** **Разработка базы данных.** 12](#_Toc161166431)

[**3.2** **Разработка программного модуля.** 18](#_Toc161166432)

[**3.3** **Текст программы.** 32](#_Toc161166433)

[**4.** **Руководство пользователя.** 38](#_Toc161166434)

[**4.1** **Выполнение программного модуля.** 38](#_Toc161166435)

[**4.2** **Сообщения пользователю.** 43](#_Toc161166436)

[**5.** **Тестирование и отладка программного модуля.** 50](#_Toc161166437)

[**Заключение.** 52](#_Toc161166438)

[**Список литературы:** 53](#_Toc161166439)

[**Приложение № 1 «Текст программы».** 54](#_Toc161166440)

**Введение.**

Информационная система поликлиника. Основная деятельность регистратуры поликлиники — это сбор и хранение информации о врачах и пациентах, обработка информации, выдача расписаний работы врачей.

**Поликлиника** – это многопрофильное лечебно-профилактическое учреждение, оказывающее медицинскую помощь населению на закреплённой территории на догоспитальном этапе.

**Врач** – специалист с высшим медицинским образованием, использующий свои навыки, знания и опыт в профилактике и лечении заболеваний, поддержании нормальной жизнедеятельности организма человека.

**Приём** – обращение к специалисту по поводу заболевания.

**Пациент** – любой человек, обратившийся за медицинской помощью, независимо от состояния своего здоровья или наличия какого-либо заболевания.

**Медицинская карта** (жарг. «карточка») – медицинский документ, в котором лечащими врачами ведётся запись истории болезни пациента и назначения ему лечения.

**Электронная медицинская карта** (ЭМК) – медицинская карта пациента медицинского учреждения в электронной форме.

**Диагноз** – медицинское заключение о состоянии здоровья обследуемого (пациента), а также сущности болезни и состоянии пациента, выраженное в принятой медицинской терминологии и основанное на всестороннем систематическом изучении пациента.

В карточку заносится информация о каждом посещении поликлиники пациентом: дата посещения, жалобы, предварительный диагноз, назначения, имя врача и его специальность.

Регистрация пациентов производится работниками регистратуры, который заполняют первую страницу карточки: фамилия, имя, отчество, возраст, год рождения, пол, страховой полис (номер). Информацию о болезнях и о посещениях вносит врач. Уволить врача может только заведующий поликлиникой.

Информационные технологии обеспечивают работу с базой данных через интерфейс программы, автоматизацию просмотра/добавления данных в базу данных, редактирования базы данных, удалению данных из неё с помощью интерфейса. Обеспечение удобства при работе с информационной системой и базой данных для людей посредством интуитивно понятного интерфейса.

В данной предметной области информационные технологии позволят автоматизировать деятельность по ведению электронной медицинской карты пациента со всеми необходимыми назначениями для диагностики и лечения, деятельность по записи пациента на приём.

1. **Назначение разработки.**

Разработка производится на основании приказа о курсовом проектировании.

Программный продукт «Поликлиника им. Уварова» - программа, созданная для взаимодействия с базой данных по предметной области.

Автоматизация продукта «Поликлиника им. Уварова» - процесс внедрения программно-аппаратных комплексов для облегчения и автоматизации процесса сбора, хранения и редактировании информации о врачах и пациентах, о приёмах, о диагнозах, о историях болезней, обработки информации.

Информация, хранимая в базе данных информационной системы регистратуры:

* Информация о сотрудниках.
* Сведения о приемах.
* Информация о пациентах (карточка пациента).
* Картотека возможных заболеваний.

Информация о сотрудниках:

* Фамилия.
* Имя.
* Отчество.
* Специальность.
* Стаж работы (опыт).
* Фото сотрудника.

Сведения о приёмах:

* Врач (Фамилия, Имя, Отчество).
* Дата приёма.
* Пациент (Фамилия, Имя, Отчество).
* Жалобы.
* Диагноз.
* Назначение.
* Дополнение к диагнозу.

Информация о пациенте хранится в его медицинской карте. Медицинская карта имеет номер. В медицинской карте указывается:

* Фамилия.
* Имя.
* Отчество.
* Возраст.
* Дата рождения.
* Пол.
* Домашний адрес.
* Страховой полис (номер).

Картотека возможных заболеваний:

* Название заболевания на русском языке.
* Название заболевания на латинском языке.
* Этап заболевания.
* Вид заболевания.
* Характер заболевания.

Приложение должно выполнять следующие функции:

* Вход в программу при запуске.
* Корректный просмотр информации из базы данных.
* Обеспечение авторизованным и аутентифицированным пользователям функционал по их ролям.
* Удаление, изменение информации в базе данных.
* Сортировка.
* Переходы по страницам.
* Печать итоговых данных.

1. **Требования к программному модулю.**
   1. **Требования к функциональным характеристикам.**

Программный модуль для программы «Поликлиника им. Уварова», должен помочь сотрудника поликлиники автоматизировать их ежедневную деятельность по обработке данных при работе с пациентами и сотрудниками.

Диаграмма:

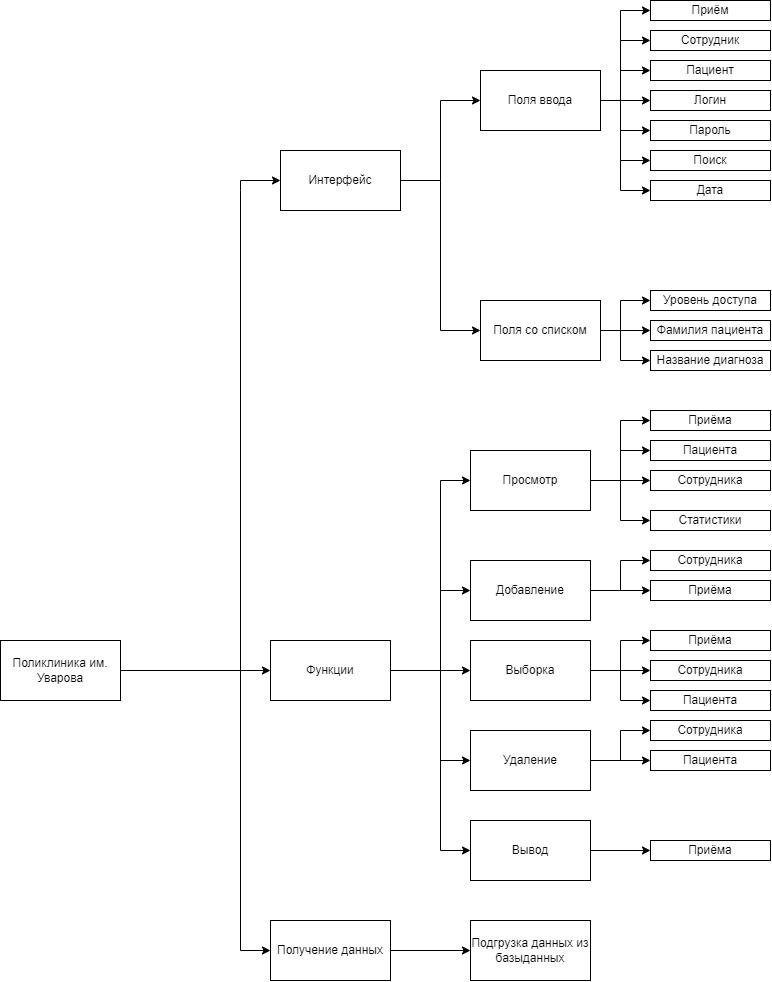


Рисунок 1. «Диаграмма “Схема приложения”»

* 1. **Требования к составу и параметрам технических средств.**

Примерные характеристики компьютера:

Таблица 1 «Примерные характеристики компьютера»

|  |  |
| --- | --- |
| Процессор | 12-Core Intel Xeon E5-2670 v3 |
| Память | 16 Гбайт |
| Разрешение экрана | 1920x1080 |
| Устройства ввода | Клавиатура, мышь |
| Дополнительно | Постоянное подключение к сети, в которой находится база данных |

* 1. **Требования к информационной и программной совместимости.**

Для корректной работы программы необходимо:

1. SQL Server Management Studio (SSMS), с установленной базой данных – это бесплатная система управления базами данных, которая включает в себя набор инструментов для управления Microsoft SQL Server.
2. Windows 10 – Операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.
3. Microsoft Excel 2019 – популярный программный продукт Microsoft, который используется для создания и анализа электронных таблиц.
4. Microsoft Visual Studio 2022 – это лучшая интегрированная среда разработки для создания многофункциональных, привлекательных кроссплатформенных приложений для Windows, Mac, Linux, iOS и Android. Создавайте многофункциональные клиентские приложения с помощью таких технологий, как WinForms, WPF, WinUI, MAUI или Xamarin.
5. MS SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных.
6. **Руководство программиста.**

При выполнении курсового проекта возникла необходимость использования базы данных. В качестве рассматриваемых вариантов были следующие:

* MySQL (СУБД).
* PostgreSQL (СУБД).
* SQLite.
* SQL Server Management Studio (СУБД).

Описание каждого варианта с плюсами и минусами:

1. MySQL - свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

Плюсы:

* Производительность.
* Кроссплатформенность.
* Бесплатная СУБД.
* Безопасность – встроенные инструменты безопасности.

Минусы:

* Работа с большими базами данных - проблемы при работе с большими объёмами данных.
* SQL - MySQL не соответствует стандарту SQL.

1. PostgreSQL - свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Плюсы:

* Написание функций на нескольких языках – он поддерживает такие языки, как: SQL, C (C++), Java, PHP, Python, Lua, Ruby.
* Кроссплатформенность.
* Работа с большими объёмами данных.

Минусы:

* Низкая эффективность репликации.
* Возможные повреждения таблиц при переносе информации между БД.

1. SQLite – это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

Плюсы:

* Хранение данных в одном файле.
* Нулевая конфигурация – не требует длительной установки или сложной настройки.
* Кроссплатформенность.
* Автономность.

Минусы:

* Ограниченная поддержка типов данных – только INTEGER, REAL, TEXT, BLOB.
* Отсутствие хранимых процедур.
* Отсутствие встроенной поддержки UNICODE.

1. SQL Server Management Studio (SSMS) – это бесплатная система управления базами данных, которая включает в себя набор инструментов для управления Microsoft SQL Server.

Плюсы:

* Тесная интеграция с ОС Windows.
* Высокая производительность, отказоустойчивость.
* Поддержка многопользовательской среды.
* Расширенные функции резервного копирования данных.
* Работа с удалённым подключением.

Минусы:

* Зависимость от ОС – работает только на ОС семейства Windows.

При первичном выборе БД варианты с MySQL и PostgreSQL были отброшены, так как: отсутствуют навыки и знания работы с ними, довольно серьёзные минусы.

При выборе между SQLite и SSMS, исходя из их плюсов и минусов, был выбран вариант – SQL Server Management Studio. Главным аргументом против SQLite стала ограниченная поддержка типов данных.

* 1. **Разработка базы данных.**
* Создание Базы данных «Hospital».

Базы данных – ПКМ.

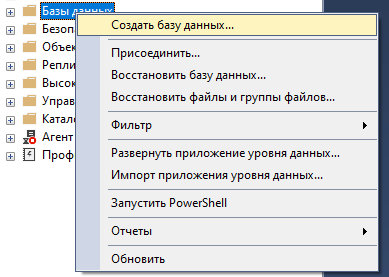


Рисунок 2. «Создание базы данных»

В поле вводим «Hospital».

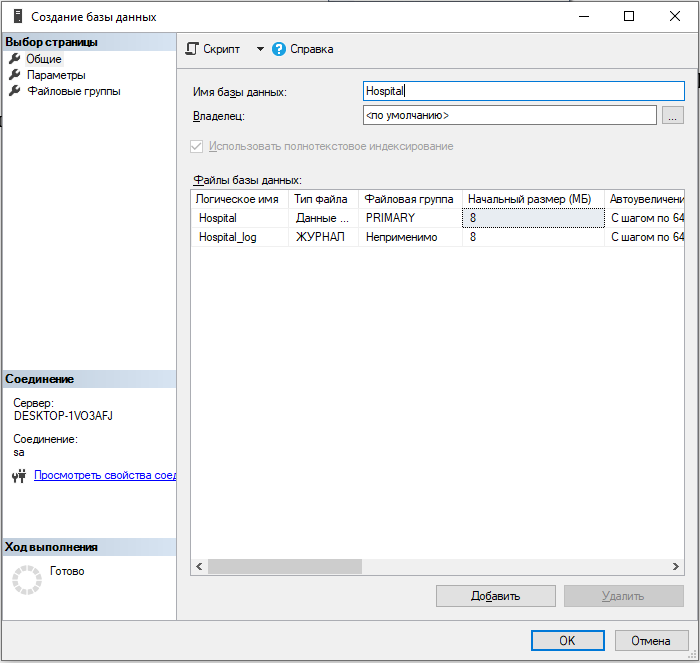


Рисунок 3. «Имя базы данных»

* Создание таблиц.

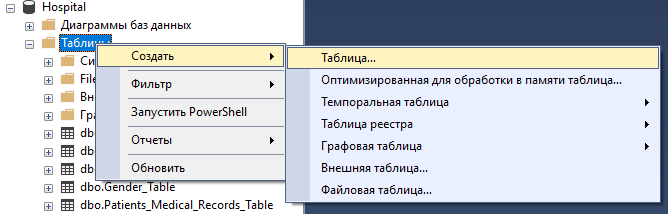


Рисунок 4. «Создание таблица»

Таблица «Access\_Level\_Table» со столбцами и типами данных:

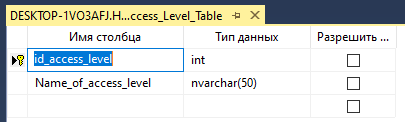


Рисунок 5. «Таблица Access\_Level\_Table»

Таблица «Diagnoses\_Table» со столбцами и типами данных:

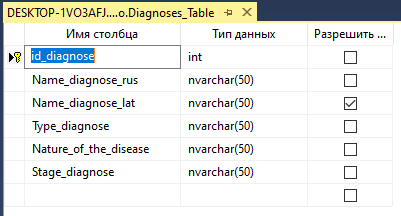


Рисунок 6. «Таблица Diagnoses\_Table»

Таблица «Doctors\_Table» со столбцами и типами данных:

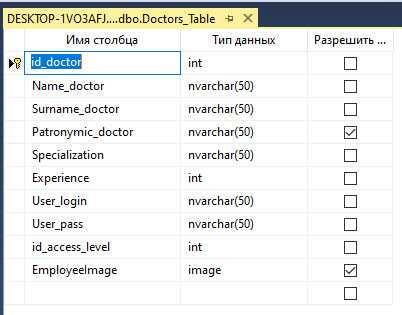


Рисунок 7. «Таблица Doctors\_Table»

Таблица «Gender\_Table» со столбцами и типами данных:

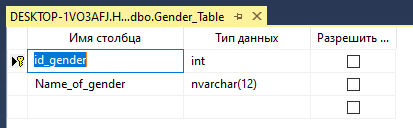


Рисунок 8. «Таблица Gender\_Table»

Таблица «Patient\_Medical\_Records\_Table» со столбцами и типами данных:

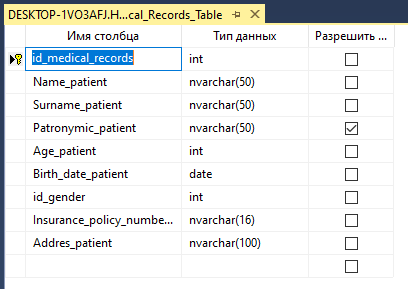


Рисунок 9. «Таблица Patient\_Medical\_Records\_Table»

Таблица «Reception\_Table» со столбцами и типами данных:

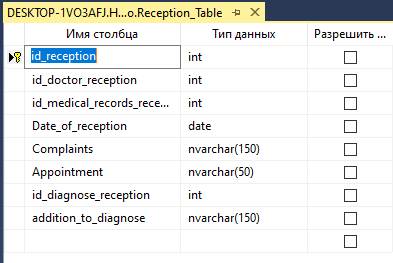


Рисунок 10. «Таблица Reception\_Table»

* Создание диаграммы базы данных:

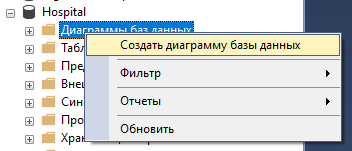


Рисунок 11. «Создание диаграммы базы данных»

Диаграмма базы данных со связями:

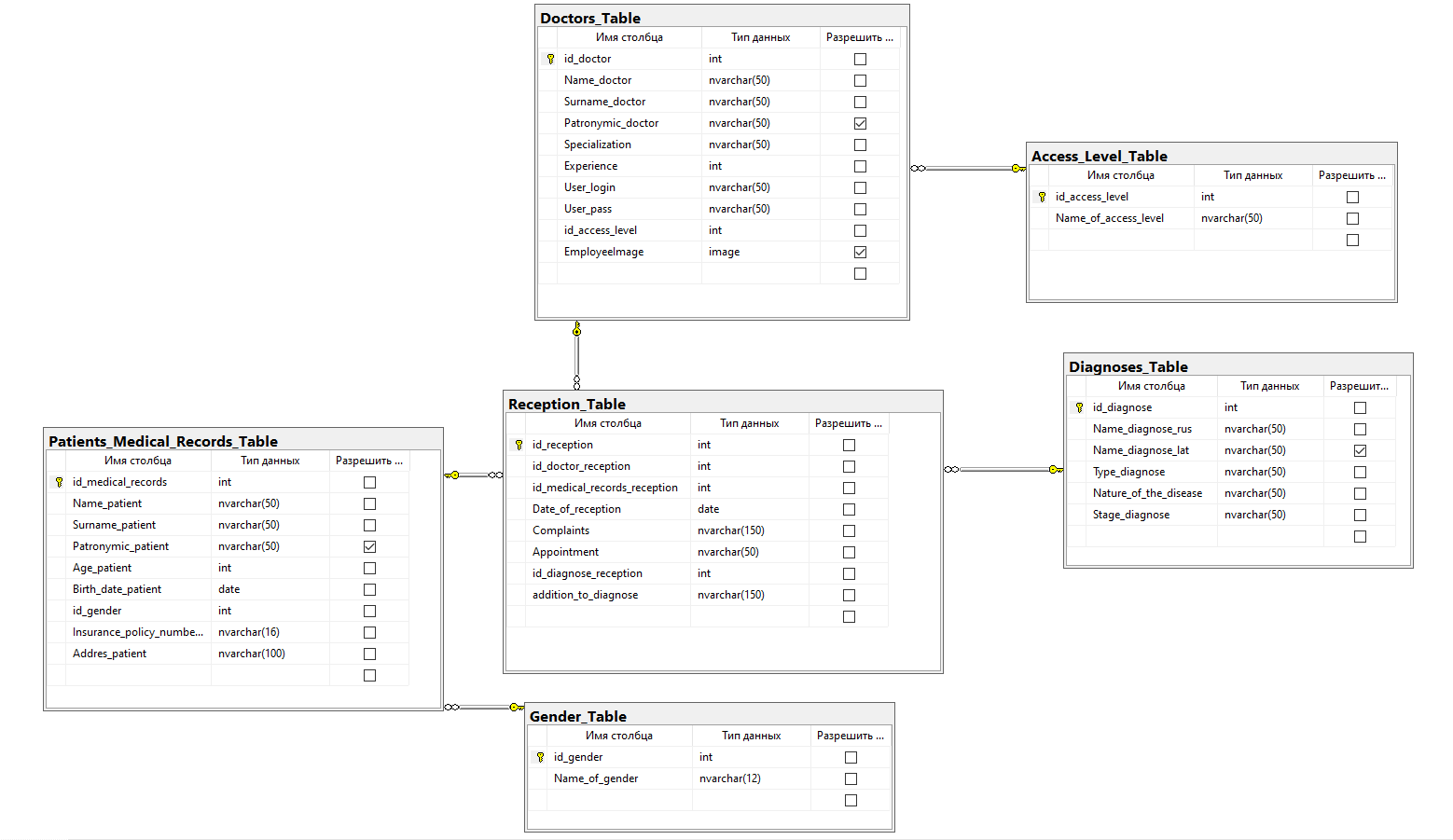


Рисунок 12. «Диаграмма базы данных со связями»

* 1. **Разработка программного модуля.**

Программа была разработана с использованием MS Visual Studio 2022.

Шаги создания программы:

1. Создание проекта:

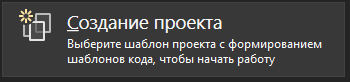


Рисунок 13. «Создание проекта»

1. Выбор типа проекта:

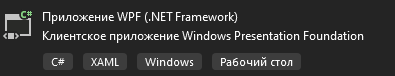


Рисунок 14. «Выбор типа проекта»

1. Подключение к базе данных с помощью ADO.NET:

В проекте создал папку Classes:

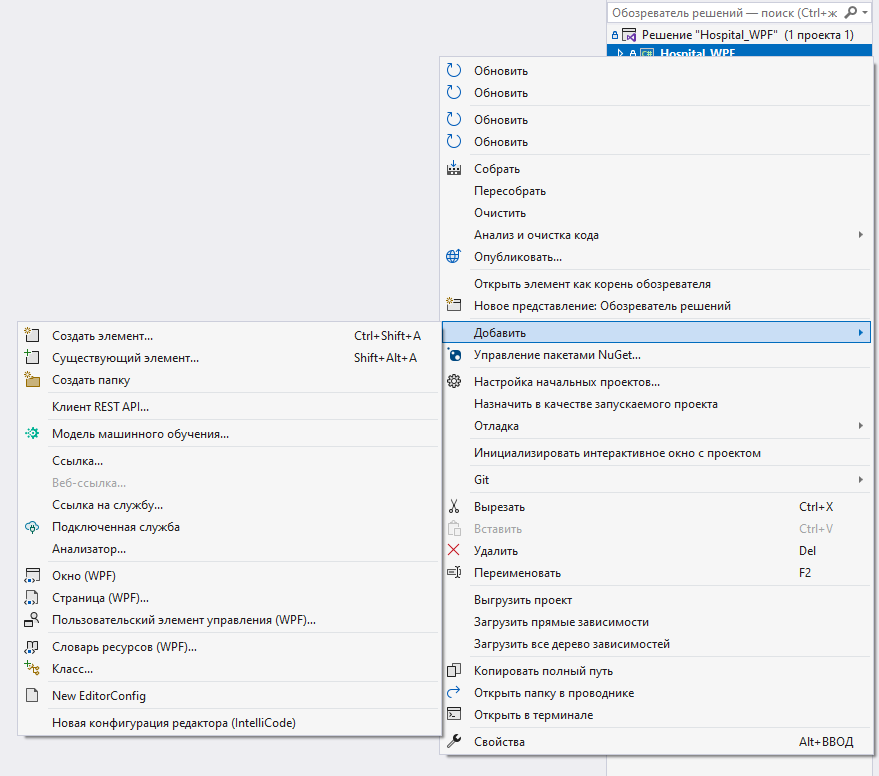


Рисунок 15. «Создание папки Classes»

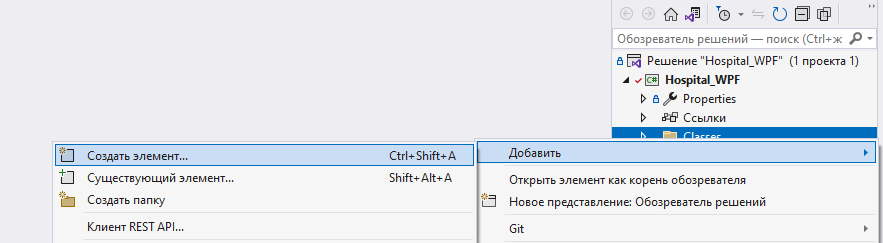


Рисунок 16. «Добавление нового элемента»

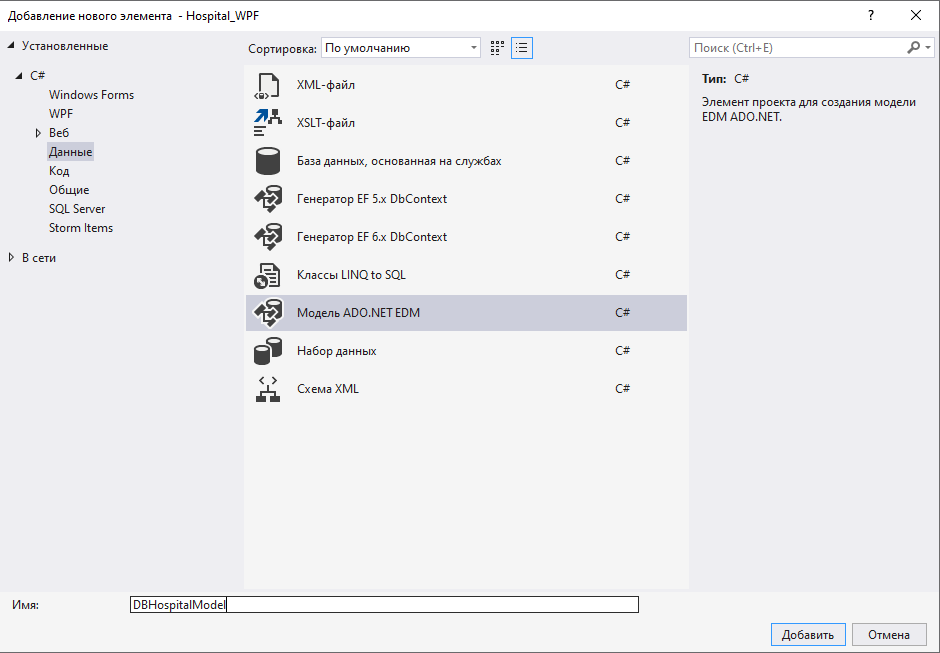


Рисунок 17. «Добавление нового элемента: Модель ADO.NET EDM»

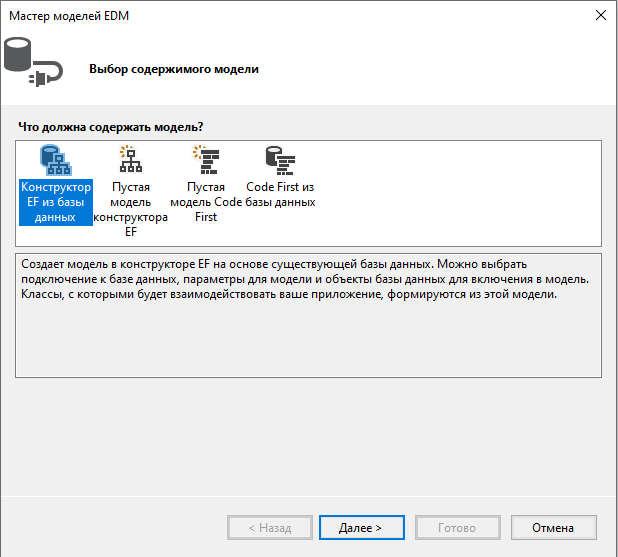


Рисунок 18. «Мастер моделей EDM: создание модели базы данных с помощью Entity First»

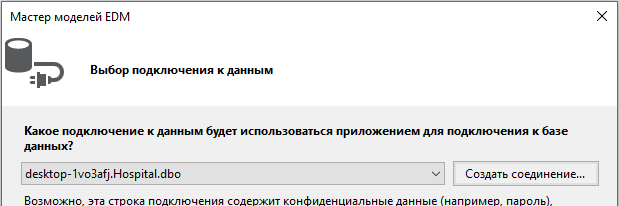


Рисунок 19. «Мастер моделей EDM»

Здесь нажать «Создать соединение».

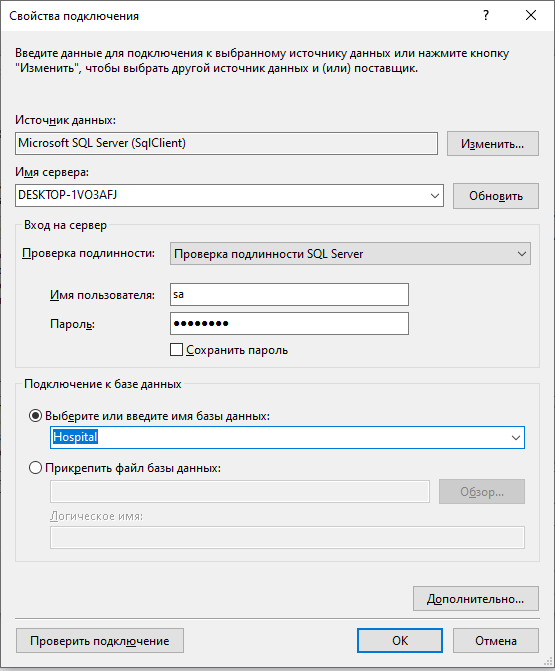


Рисунок 20. «Создание подключения к базе данных»

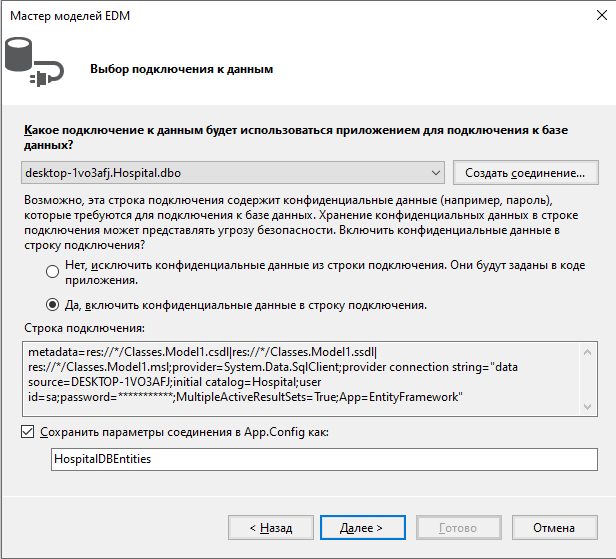


Рисунок 21. «Мастер моделей EDM: подключение к базе данных»

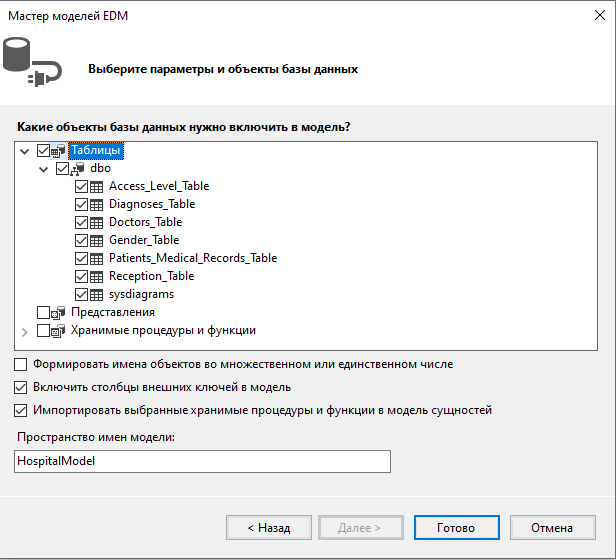


Рисунок 22. «Мастер моделей EDM: выбор таблиц»

Модель данных.

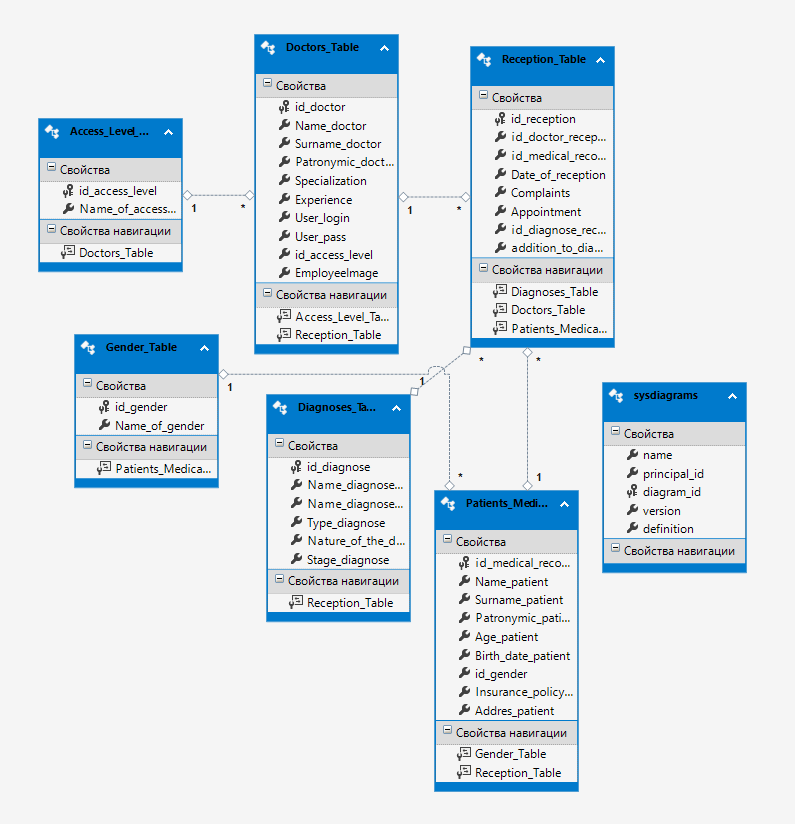


Рисунок 23. «Модель данных»

Модель базы данных в «Обозреватель решений».

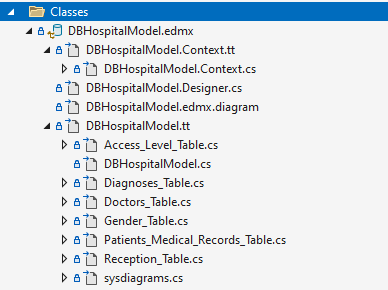


Рисунок 24. «Модель базы данных в «Обозреватель решений»»

Код дописанный в контекст подключения базы данных:

private static DBHospitalEntities \_context;

public static DBHospitalEntities GetContext()

{

if (\_context == null)

\_context = new DBHospitalEntities();

return \_context;

}

1. Создание класса FrameClass:

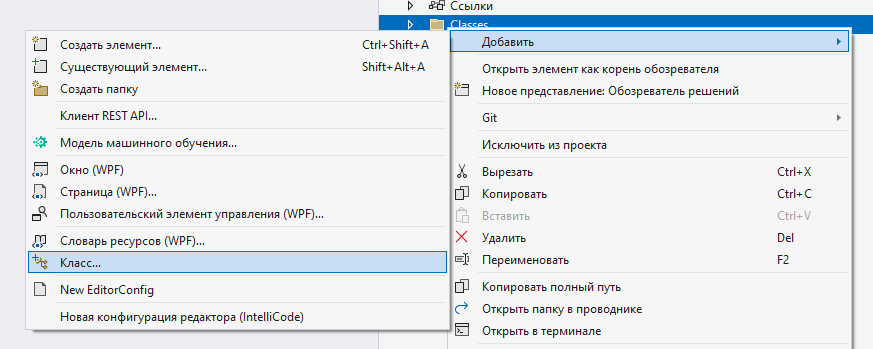


Рисунок 25. «Создание класса FrameClass»

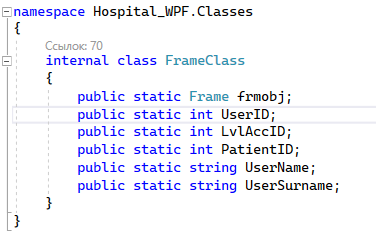


Рисунок 26. «Элементы класса FrameClass»

В классе элемент для перехода между страницами (элемент frmobj). Остальные элементы необходимы во время работы программы.

1. Создание страницы.

Создал папку Pages.

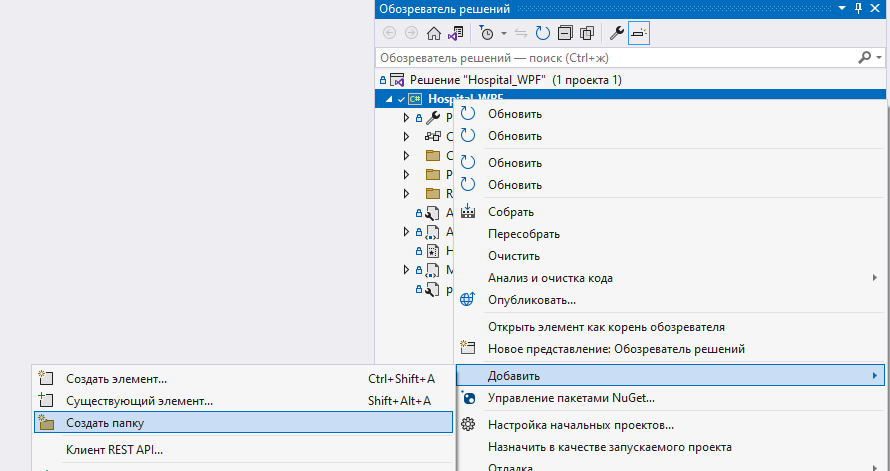


Рисунок 27. «Создание папки»

В ней страницу:

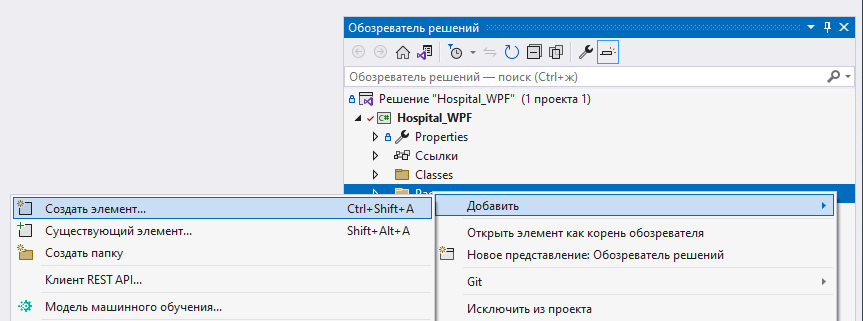


Рисунок 28. «Создание страницы: создание элемента»

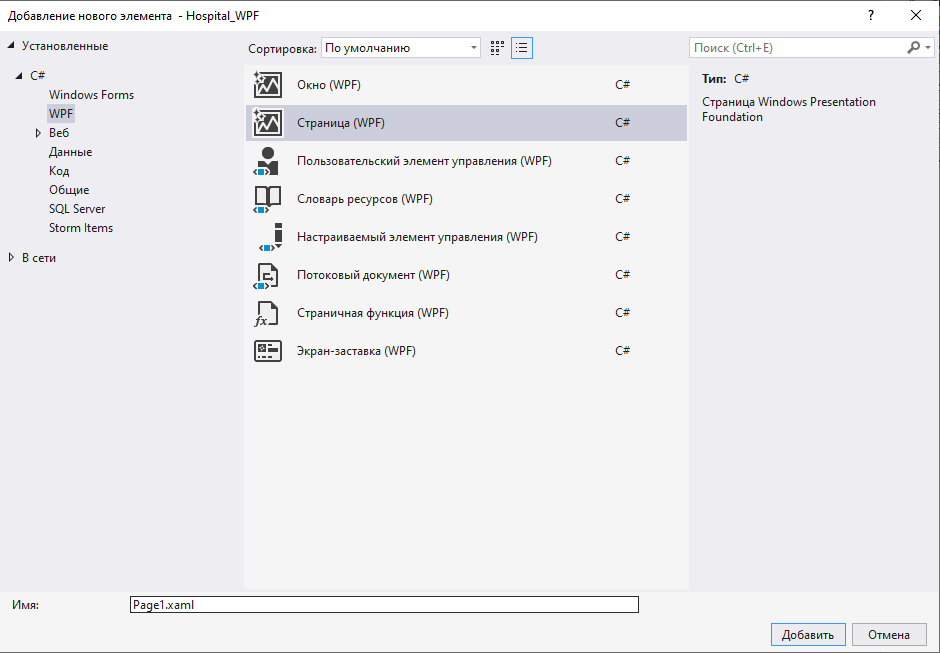


Рисунок 29. «Создание страницы: создание страницы»



Рисунок 30. «Страница MainWindow.xaml»

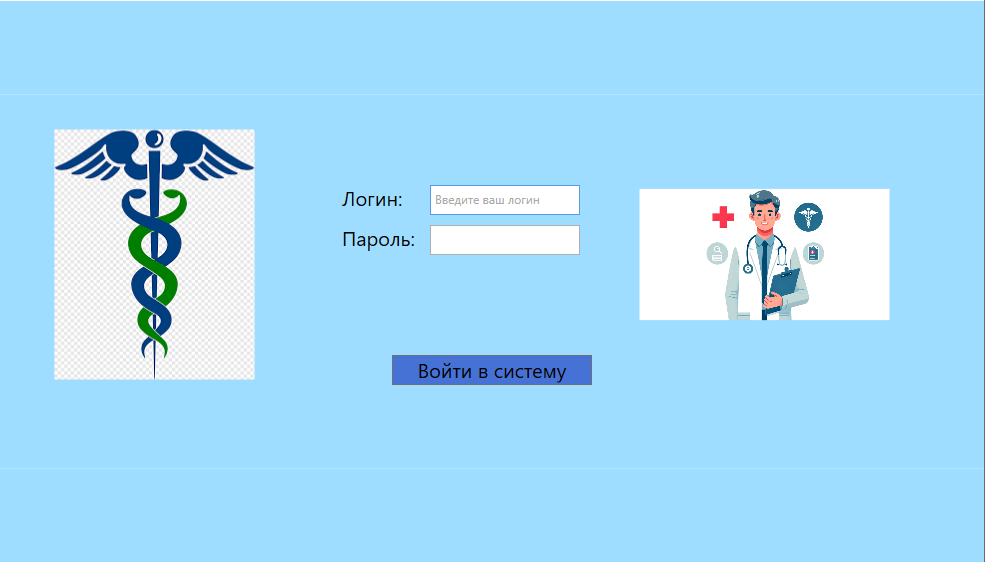


Рисунок 31. «Страница AuthorizationPage.xaml»

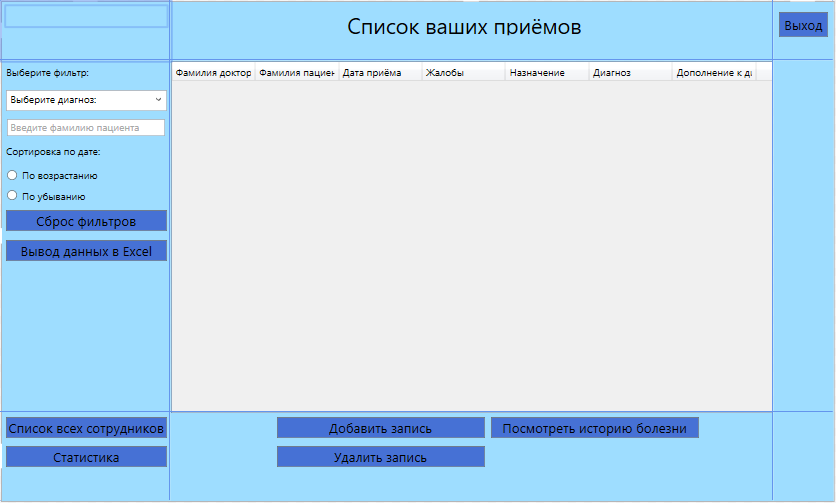


Рисунок 32. «Страница MainViewPage.xaml»

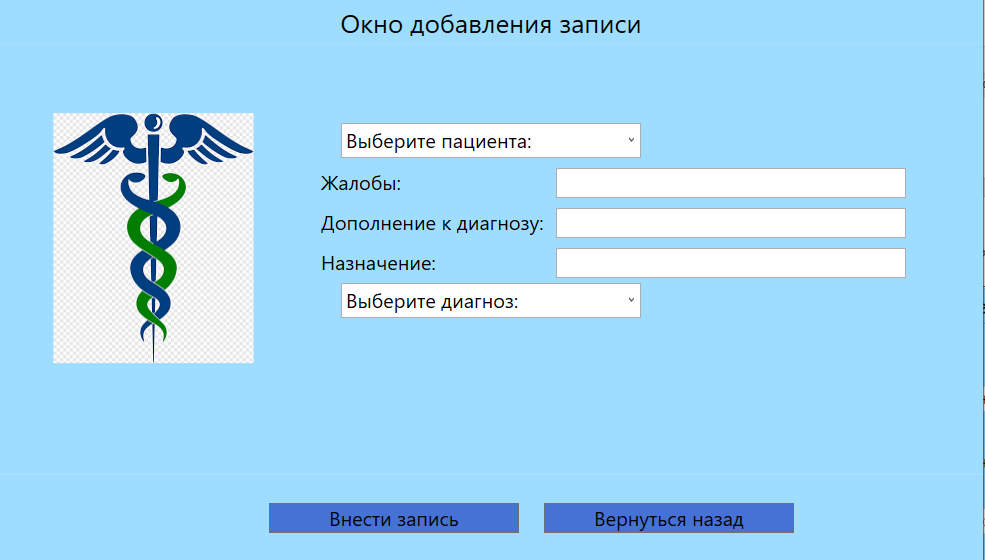


Рисунок 33. «Страница AddReceptionPage.xaml»

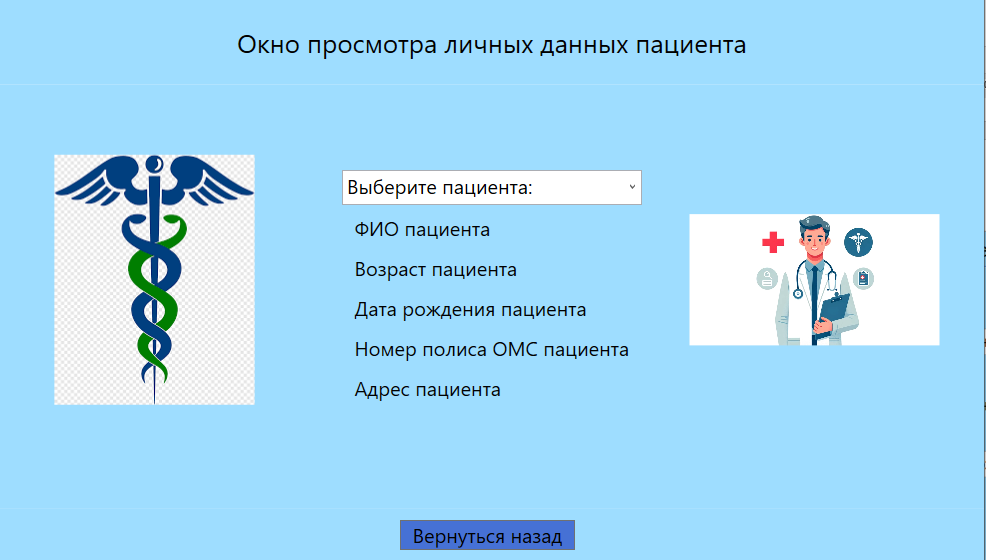


Рисунок 34. «Страница PatientViewPage.xaml»



Рисунок 35. «Страница StatsPage.xaml»

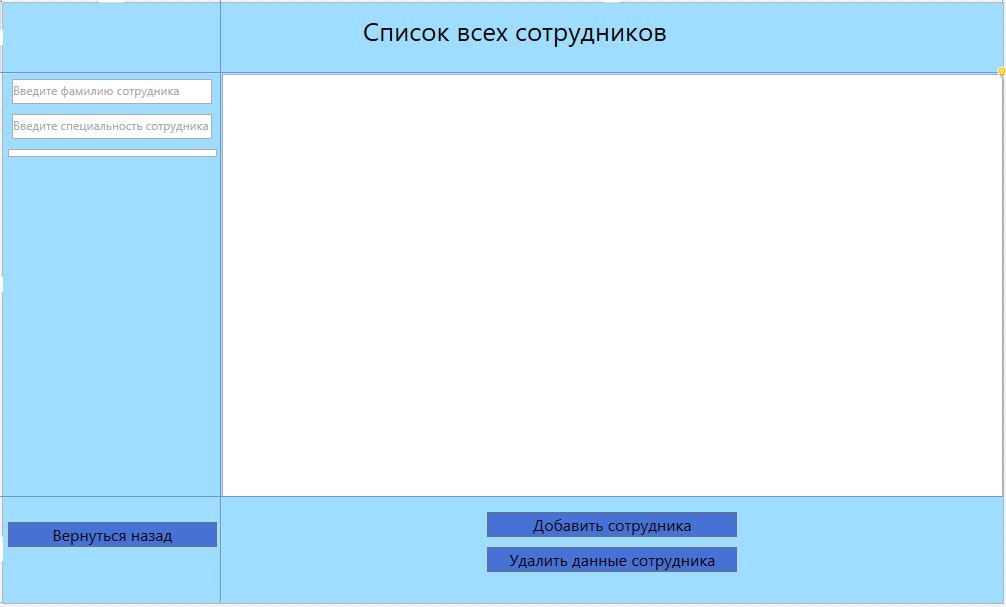


Рисунок 36. «Страница ViewEmployeePage.xaml»

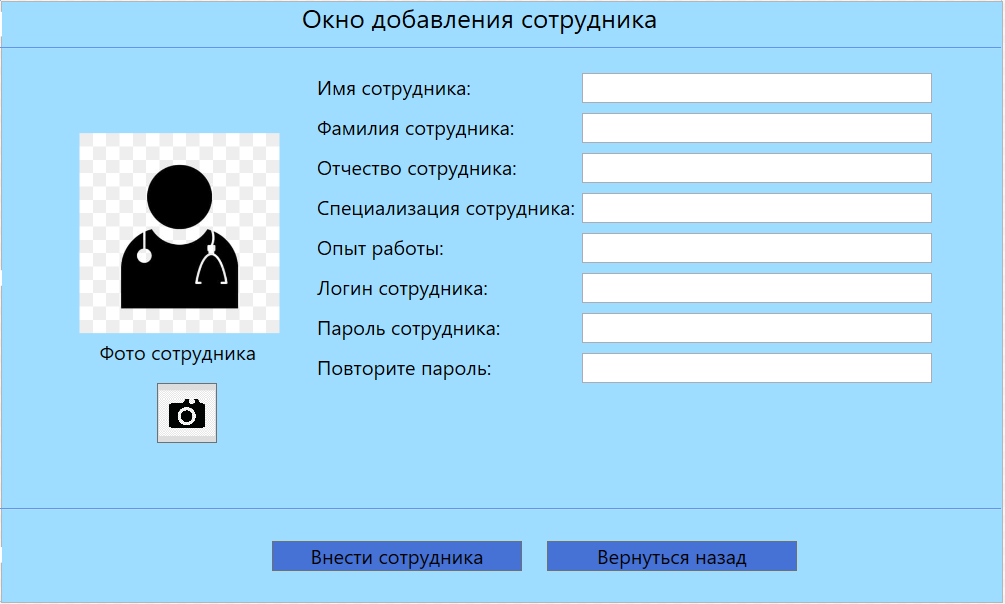


Рисунок 37. «Страница AddEmployeePage.xaml»

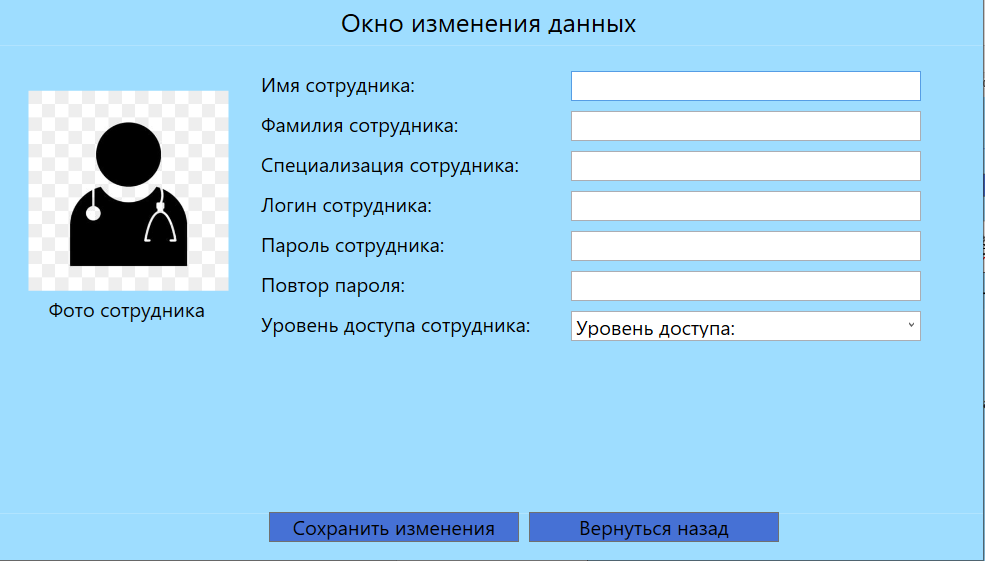


Рисунок 38. «Страница EditEmployeePage.xaml»

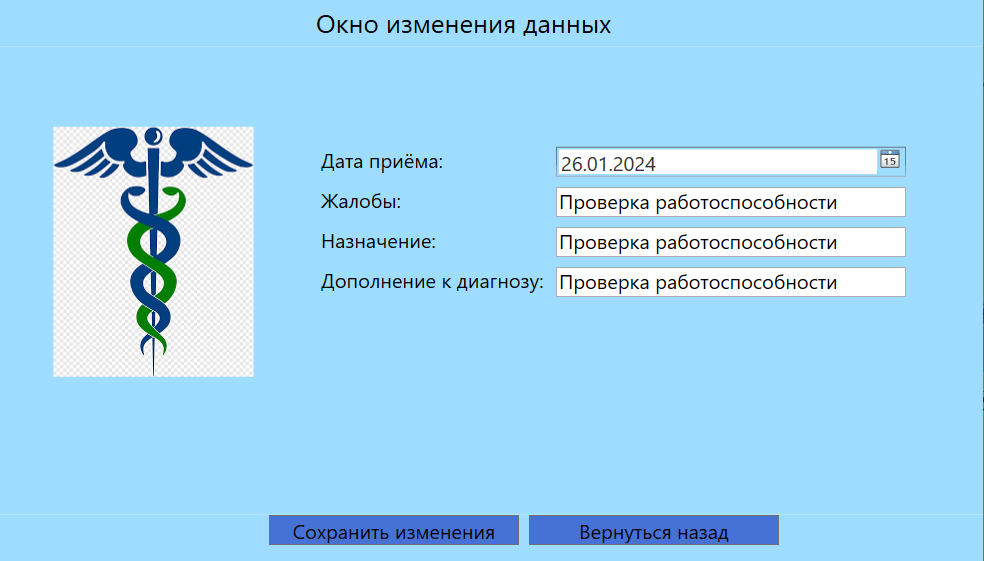


Рисунок 39. «Страница EditReceptionPage.xaml»

**3.3** **Текст программы.**

Страница MainWindow – главное окно, в которое загружаются все остальные страницы.

Xaml код:

<Window x:Class="Hospital\_WPF.MainWindow"

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

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

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

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

xmlns:local="clr-namespace:Hospital\_WPF"

mc:Ignorable="d"

Title="Поликлиника им. Уварова" Height="600" Width="1000"

ResizeMode="CanMinimize">

<Grid>

<Frame x:Name="MainFrame"

NavigationUIVisibility="Hidden"/>

</Grid>

</Window>

Cs код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

Classes.FrameClass.frmobj = MainFrame;

Classes.FrameClass.frmobj.Navigate(new Pages.AuthorizationPage());

}

}

}

Страница AuthorizationPage – используется для авторизации в программе:

<Page x:Class="Hospital\_WPF.Pages.AuthorizationPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Поликлиника">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Style="{StaticResource StackPanelStyle}"/>

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

Style="{StaticResource StackPanelStyle}">

<TextBlock FontSize="20"

Text="Логин: "

Width="100"

Height="30"

Margin="-200,90,0,0"

Style="{StaticResource TextBlockStyle}"/>

<TextBox x:Name="TxtBoxLogAuth"

FontSize="20"

Width="150"

Height="30"

Margin="25,-30,0,0"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock IsHitTestVisible="False" Text="Введите ваш логин"

VerticalAlignment="Center" HorizontalAlignment="Left" Foreground="DarkGray"

Margin="435,-26,0,0"

Width="125"

Height="20"

Name="TxtBlockHintLogin">

<TextBlock.Style>

<Style>

<Setter Property="ItemsControl.Visibility" Value="Collapsed"/>

<Style.Triggers>

<DataTrigger Binding="{Binding Text, ElementName=TxtBoxLogAuth}" Value="">

<Setter Property="ItemsControl.Visibility" Value="Visible"/>

</DataTrigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<TextBlock FontSize="20"

Text="Пароль: "

Width="100"

Height="30"

Margin="-200,10,0,0"

Style="{StaticResource TextBlockStyle}"/>

<PasswordBox x:Name="PassBoxPassAuth"

FontSize="20"

Width="150"

Height="30"

Margin="25,-30,0,0"

Style="{StaticResource PassBoxStyle}"/>

<Button x:Name="BtnAuth"

Click="BtnAuth\_Click"

Content="Войти в систему"

Width="200"

Height="30"

FontSize="20"

Margin="0,100,0,0"

Style="{StaticResource BtnStyle}"/>

<Image Margin="-670,-255,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\Hospital\_Snake.png</Image.Source>

</Image>

<Image Margin="550,-255,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\DoctorImage\_Auth.jpg</Image.Source>

</Image>

</StackPanel>

<StackPanel Grid.Row="2" Style="{StaticResource StackPanelStyle}"/>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class AuthorizationPage : Page

{

public AuthorizationPage()

{

InitializeComponent();

TxtBoxLogAuth.MaxLength = 50;

PassBoxPassAuth.MaxLength = 25;

TxtBoxLogAuth.ToolTip = "Введите ваш логин";

PassBoxPassAuth.ToolTip = "Введите ваш пароль, длина которого не менее 6 символов";

}

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

{

String Login = TxtBoxLogAuth.Text;

String Password = PassBoxPassAuth.Password;

if (Login != "" && Password != "")

{

var Level = DBHospitalEntities.GetContext().Doctors\_Table.FirstOrDefault(x => x.User\_login == TxtBoxLogAuth.Text && x.User\_pass == PassBoxPassAuth.Password);

if (Level != null)

{

if (Level.Access\_Level\_Table.id\_access\_level == 1)

{

MessageBox.Show($"Администратор - {Level.Name\_doctor} авторизован!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.Navigate(new MainViewPage(Level));

TxtBoxLogAuth.Text = null;

PassBoxPassAuth.Password = null;

FrameClass.UserName = Level.Name\_doctor;

FrameClass.UserSurname = Level.Surname\_doctor;

}

else if (Level.Access\_Level\_Table.id\_access\_level == 2)

{

MessageBox.Show($"Главный врач - {Level.Name\_doctor}, авторизован!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.Navigate(new MainViewPage(Level));

TxtBoxLogAuth.Text = null;

PassBoxPassAuth.Password = null;

FrameClass.UserName = Level.Name\_doctor;

FrameClass.UserSurname = Level.Surname\_doctor;

}

else if (Level.Access\_Level\_Table.id\_access\_level == 3)

{

MessageBox.Show($"Доктор - {Level.Name\_doctor}, авторизован!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.Navigate(new MainViewPage(Level));

TxtBoxLogAuth.Text = null;

PassBoxPassAuth.Password = null;

Classes.FrameClass.UserID = Level.id\_doctor;

Classes.FrameClass.LvlAccID = Level.id\_access\_level;

FrameClass.UserName = Level.Name\_doctor;

FrameClass.UserSurname = Level.Surname\_doctor;

}

}

else MessageBox.Show("Данный пользователь не существует", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("Поля не должны быть пустыми", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

}

Страница MainViewPage – используется для просмотра приёмов (всех если вы администратор или главный врач, только ваших, если вы доктор) (см. Приложение № 1 «Текст приложения»).

Страница StatsPage – страница статистики, где отображается количество пациентов – общее, так и по диагнозам (общее – для администратора, главного врача, ваших если вы доктор) (см. Приложение № 1 «Текст приложения»).

Страница PatientViewPage – страница просмотра личных данных пациента (только для врачей) (см. Приложение № 1 «Текст приложения»).

Страница AddReceptionPage – страница добавления (создания) записи о приёме (см. Приложение № 1 «Текст приложения»).

Страница EditReceptionPage – страница редактирования записи о приёме (см. Приложение № 1 «Текст приложения»).

Страница ViewEmployeePage – страница просмотра сотрудников поликлиники (Доктор не может перейти на эту страницу) (см. Приложение № 1 «Текст приложения»).

Страница AddEmployeePage – страница добавления сотрудника в систему (доктор не может перейти на эту страницу) (см. Приложение № 1 «Текст приложения»).

Страница EditEmployeePage – страница редактирования личных данных сотрудника (доктор не может перейти на эту страницу) (см. Приложение № 1 «Текст приложения»).

1. **Руководство пользователя.**

**4.1** **Выполнение программного модуля.**

Для запуска программы следует запустить Hospital\_WPF.exe

Существуют 3 роли:

* Администратор – главный пользователь системы, у которого большего всего возможностей взаимодействия с программой;
* Главный врач – пользователь системы, который имеет возможностей немного меньше, чем администратор;
* Доктор – пользователь системы, у которого наименьший функционал, но он может добавлять записи о приёмах и выводить данные приёма в Excel.

Страница авторизации при запуске:

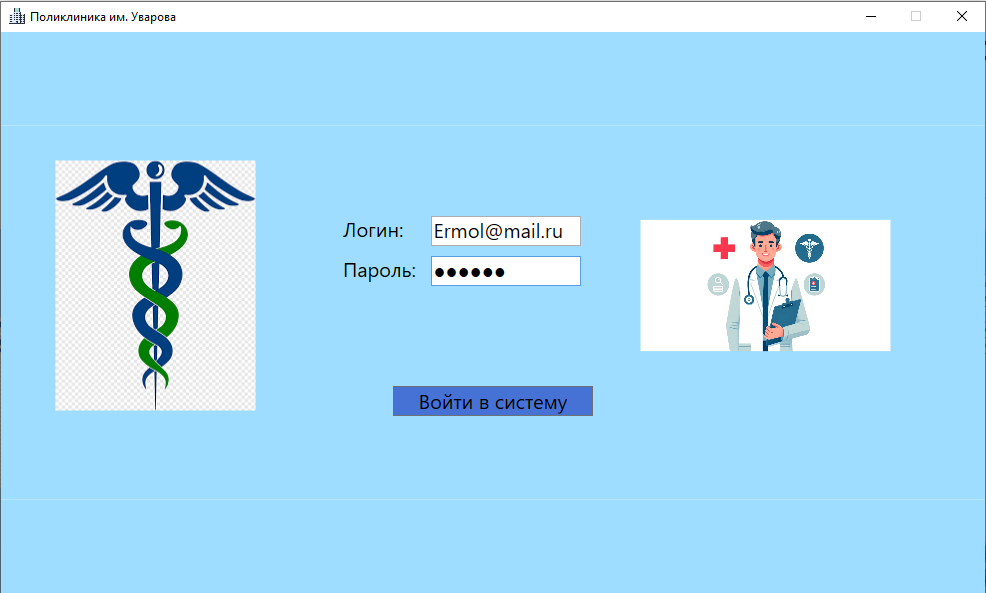


Рисунок 40. «Страница авторизации»

Основная страница от лица доктора, после входа от лица доктора:



Рисунок 41. «Основная страница от лица доктора»

Страница добавления записи (только у доктора), после нажатия кнопки «Добавить запись»:

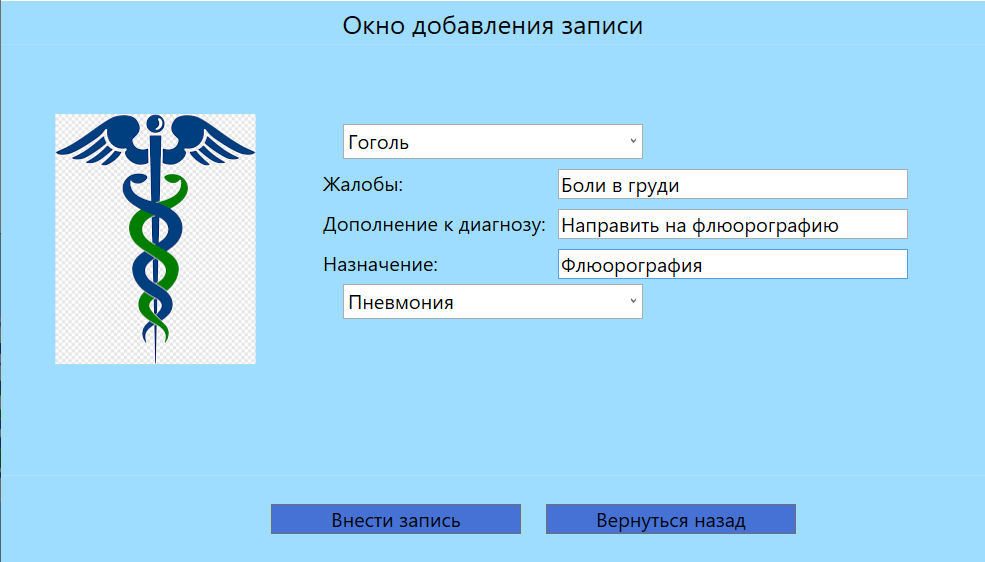


Рисунок 42. «Страница добавления записи (только у доктора)»

Страница просмотра истории болезней (только у доктора), после нажатия кнопки «Посмотреть историю болезни»:

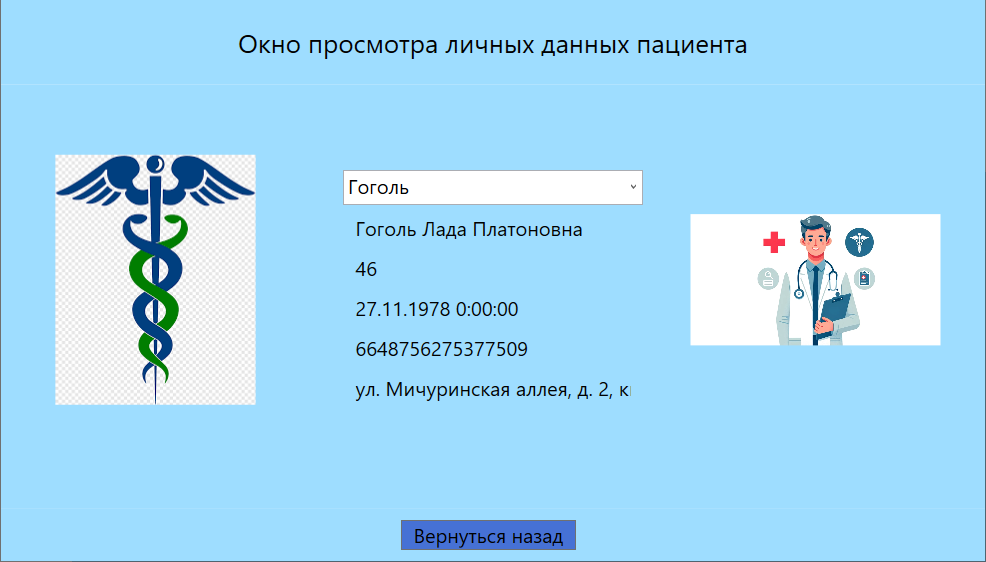
****

Рисунок 43. «Страница просмотра истории болезней (только у доктора)»

Страница редактирования записи о приёме, после нажатия кнопки «Редактировать»:



Рисунок 44. «Страница редактирования записи о приёме»

Страница просмотра статистики (от лица доктора), после нажатия кнопки «Статистика»:



Рисунок 45. «Страница просмотра статистики (от лица доктора)»

Основная страница (от лица администратора и главного врача), после авторизации от лица администратора или главного врача:



Рисунок 46. «Основная страница (от лица администратора и главного врача)»

Страница редактирования записи о приёме, после нажатия кнопки «Редактировать»:

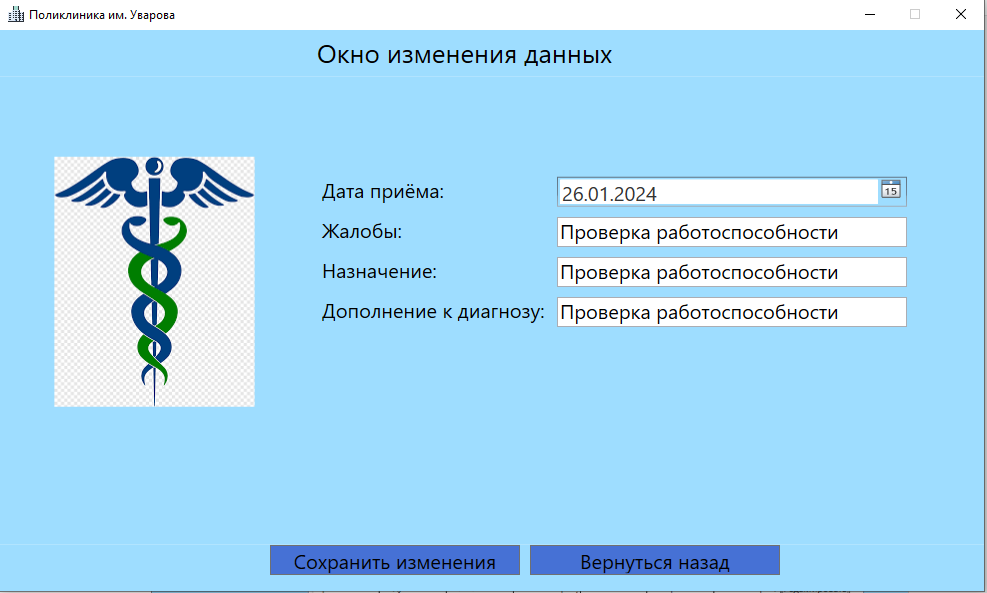


Рисунок 47. «Страница редактирования записи о приёме»

Страница просмотра всех сотрудников (доктор не может зайти на неё), после нажатия кнопки «Список всех сотрудников»:

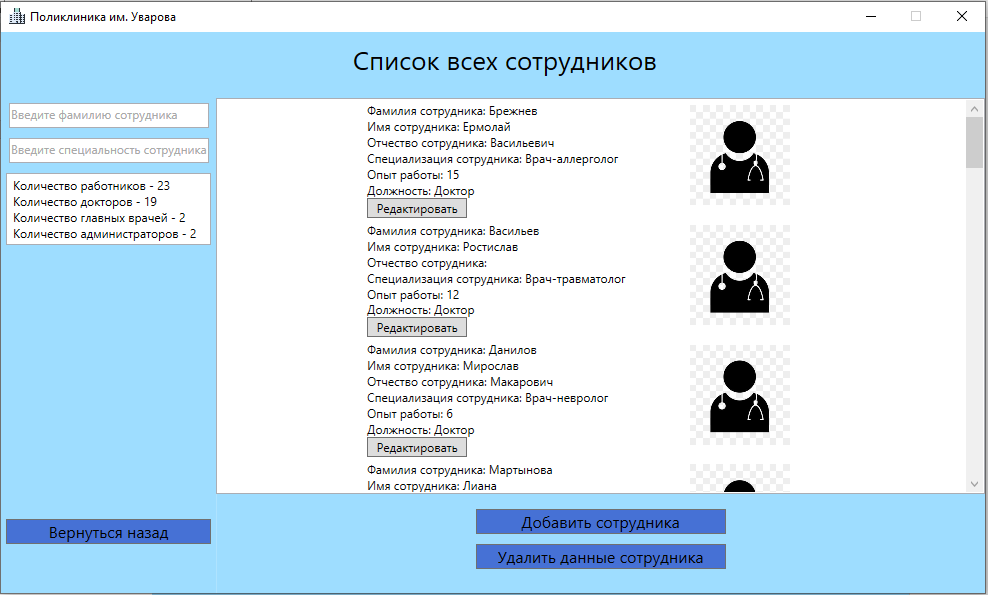


Рисунок 48. «Страница просмотра всех сотрудников (доктор не может зайти на неё)»

Страница добавления сотрудника (от лица администратора и главного врача), после нажатия кнопки «Добавить сотрудника»:

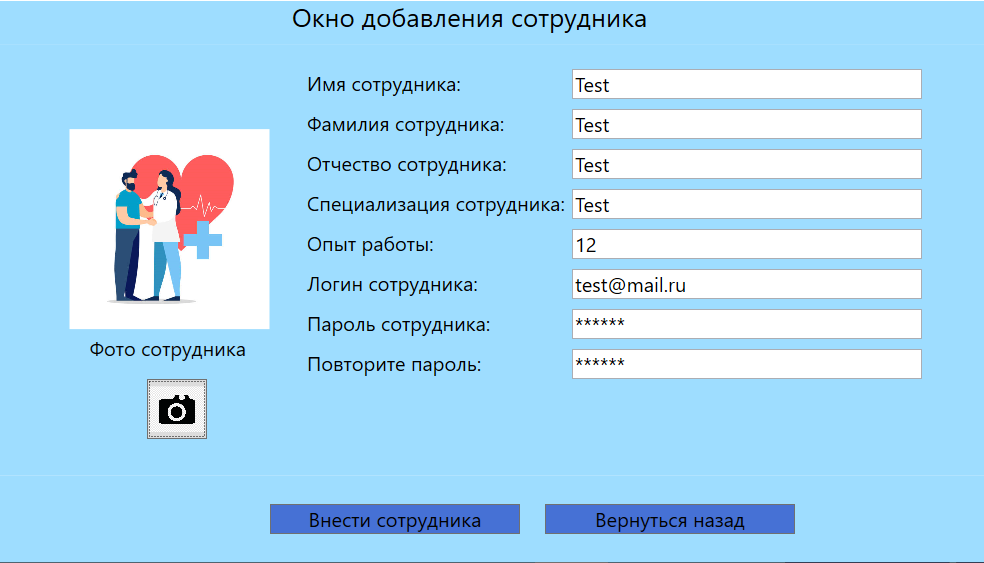


Рисунок 49. «Страница добавления сотрудника (от лица администратора и главного врача)»

Страница редактирования личных данных сотрудника, после нажатия кнопки «Редактировать»:

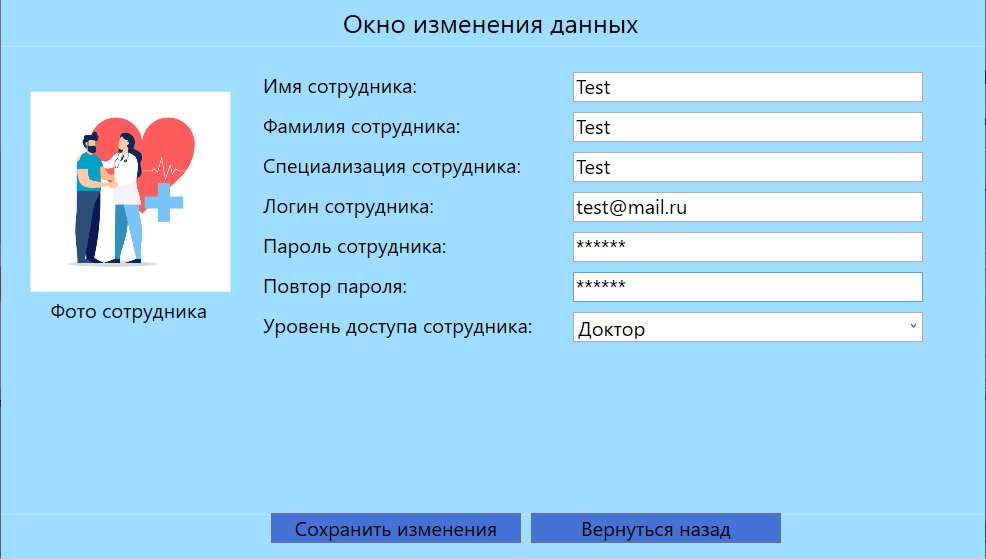


Рисунок 50. «Страница редактирования личных данных сотрудника»

* 1. **Сообщения пользователю.**
* Сообщение о пустых текстовых полях на странице авторизации.

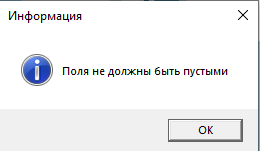


Рисунок 51. «Сообщение о пустых полях на странице авторизации»

* Сообщение о неправильном логине/пароле.

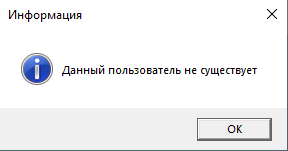


Рисунок 52. «Сообщение о неправильном логине/пароле»

* Сообщение об успешной авторизации пользователя

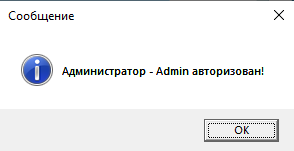


Рисунок 53. «Сообщение об успешной авторизации»

* Сообщение о пустых текстовых полях на странице добавления записи о приёме.

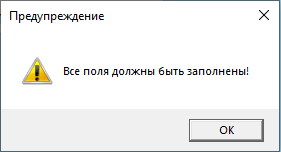


Рисунок 54. «Сообщение о пустых текстовых полях на странице добавления записи о приёме»

* Сообщение об успешно добавленной записи о приёме.

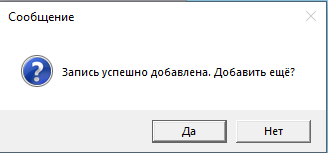


Рисунок 55. «Сообщение об успешно добавленной записи о приёме»

* Сообщение об удалении записей о приёме.

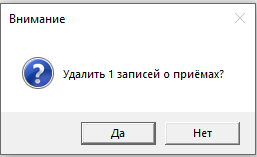


Рисунок 56. «Сообщение об удалении записей о приёме»

* Сообщение об успешном удалении записи о приёме.

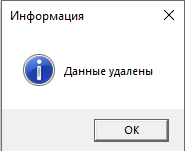


Рисунок 57. «Сообщение об успешном удалении записи о приёме»

* Сообщение о пустых текстовых полях на странице редактирования записи о приёме.

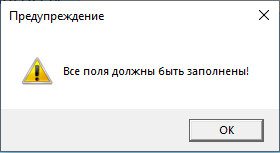


Рисунок 58. «Сообщение о пустых текстовых полях на странице редактирования записи о приёме»

* Сообщение об успешном редактировании записи о приёме.

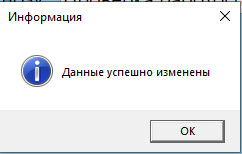


Рисунок 59. «Сообщение об успешном редактировании записи о приёме»

* Сообщение о пустых полях на странице добавления сотрудника.

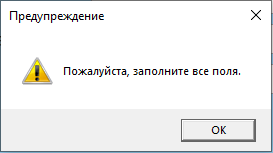


Рисунок 60. «Сообщение о пустых полях на странице добавления сотрудника»

* Сообщение о превышении размера изображения для загрузки для сотрудника.

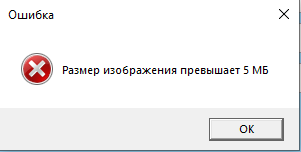


Рисунок 61. «Сообщение о превышении размера изображения для загрузки для сотрудника»

* Сообщение о минимальной и максимальной длине пароля.

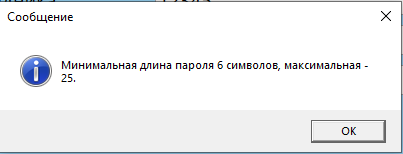


Рисунок 62. «Сообщение о минимальной и максимальной длине пароля»

* Сообщение о неправильно введённом логине.

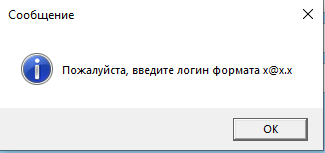


Рисунок 63. «Сообщение о неправильно введённом логине»

* Сообщение об успешном добавлении сотрудника.

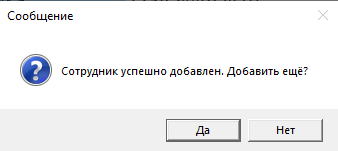


Рисунок 64. «Сообщение об успешном добавлении сотрудника»

* Сообщение о пустых полях на странице редактирования личных данных сотрудника.

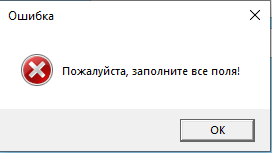


Рисунок 65. «Сообщение о пустых полях на странице редактирования личных данных сотрудника»

* Сообщение о минимальной и максимальной длине пароля.

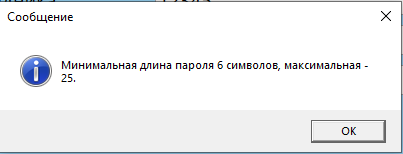


Рисунок 66. «Сообщение о минимальной и максимальной длине пароля»

* Сообщение о неправильно введённом логине.

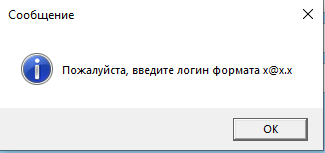


Рисунок 67. «Сообщение о неправильно введённом логине»

* Сообщение о несовпадении паролей.

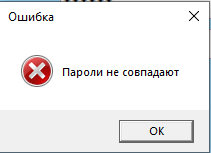


Рисунок 68. «Сообщение о несовпадении паролей»

* Сообщение об успешном редактировании записи личных данных сотрудника.

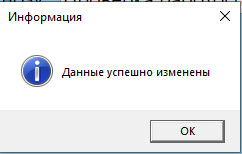


Рисунок 69. «Сообщение об успешном редактировании записи личных данных сотрудника»

* Сообщение об удалении записей личных данных сотрудника.

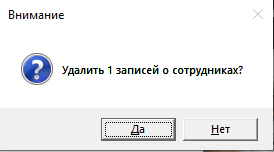


Рисунок 70. «Сообщение об удалении записей личных данных сотрудника»

* Сообщение об успешном удалении записи личных данных сотрудника.

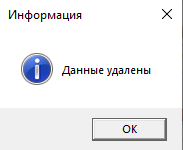


Рисунок 71. «Сообщение об успешном удалении записи личных данных сотрудника»

* Сообщение о несовпадении паролей на странице добавления сотрудника.

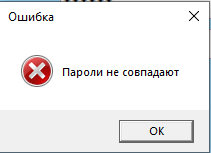
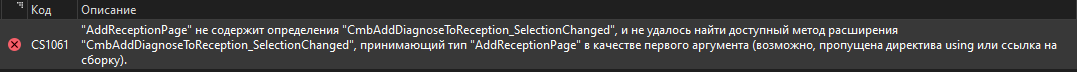


Рисунок 72. «Сообщение о несовпадении паролей на странице добавления сотрудника»

1. **Тестирование и отладка программного модуля.**

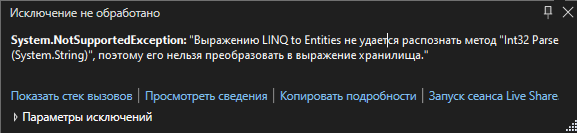
Тестирование программного обеспечения - процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом (ISO/IEC TR 19759:2005).

Ошибки:



Как решил:

Пересобрал решение.



Как решил:

Была создана глобальная переменная:

public static int PatientID;

При выборе элемента в ComboBox происходило присваивание ей значения:

FrameClass.PatientID = int.Parse(CmbSelectPatientMedRec.SelectedValue.ToString());

Дальше замена кода:

var patientrec = DBHospitalEntities.GetContext().Patients\_Medical\_Records\_Table.Where(x => x.id\_medical\_records == int.Parse(CmbSelectPatientMedRec.SelectedValue.ToString())).FirstOrDefault();

На:

var patientrec = DBHospitalEntities.GetContext().Patients\_Medical\_Records\_Table.Where(x => x.id\_medical\_records == FrameClass.PatientID).FirstOrDefault();

**Заключение.**

Для информационной системы «Поликлиника» был разработан программный модуль.

В результате выполнения исследования в рамках курсового проектирования был разработан программный модуль для информационной системы «Поликлиника». В качестве базы данных была использована СУБД SQL Server Management Studio, в качестве языка программирования – C#. Приложение всем поставленным целям соответствует, были реализованы следующие функции: просмотр, добавления, удаление, редактирование, поиск, сортировка, вывод данных в Excel. Программный модуль может быть установлен на ОС Windows 10,11.

В дальнейшем в программе может быть доработана функция редактирования личных данных сотрудника (редактирование изображения).

**Список литературы:**

1. Прайс Марк, C# 10 и .NET 6. Современная кросс-платформенная разработка / Прайс Марк. 6-е изд., Издательский дом «Питер», 17 нояб. 2022 г.
2. Гриффитс Иэн, Программируем на C# 8.0. Разработка приложений / Гриффитс Иэн. Издательский дом «Питер», 1 июн., 2021 г.
3. Арораа Гаурав, Чилберто Джеффри, Паттерны проектирования для C# и платформы .NET Core / Арораа Гаурав, Чилберто Джеффри. Издательский дом «Питер», 19 янв., 2021 г.
4. Клири Стивен, Конкурентность в C#. Асинхронное, параллельное и многопоточное программирование / Клири Стивен. 2-е межд. изд., Издательский дом «Питер», 4 мар. 2020 г.
5. Матинян Н.В., Горбунова Т.В., Епифанова Н.Ю. и др, Инфекционные осложнения в детской онкологии и гематологии / Матинян Н.В., Горбунова Т.В., Епифанова Н.Ю. и др. Издательство – ГЭОТАР – Медиа, 2024 год.
6. Радзинский В.Е, Женская консультация. Руководство для врачей / Радзинский В.Е. Издательство – ГЭОТАР – Медиа, 2024 год.
7. Лейтман М.У.; Пер. с англ. под ред. И.А. Лоскутова, Диагностическая офтальмология. Практическое руководство / Лейтман М.У.; Пер. с англ. под ред. И.А. Лоскутова. Издательство – ГЭОТАР – Медиа, 2023 год.
8. Под ред. С.Дж.К. Хси, Э.Э. Моррис; Пер. с англ.; Под ред. Д.А. Рябчикова, Рак молочной железы. От диагностики до лечения / Под ред. С.Дж.К. Хси, Э.Э. Моррис; Пер. с англ.; Под ред. Д.А. Рябчикова. Издательство – ГЭОТАР – Медиа, 2024 год.

**Приложение № 1 «Текст программы».**

Код страницы «MainWindow» (см. рисунок 30)

Xaml код:

<Window x:Class="Hospital\_WPF.MainWindow"

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

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

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

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

xmlns:local="clr-namespace:Hospital\_WPF"

mc:Ignorable="d"

Title="Поликлиника им. Уварова" Height="600" Width="1000"

ResizeMode="CanMinimize">

<Grid>

<Frame x:Name="MainFrame"

NavigationUIVisibility="Hidden"/>

</Grid>

</Window>

Cs код:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

Classes.FrameClass.frmobj = MainFrame;

Classes.FrameClass.frmobj.Navigate(new Pages.AuthorizationPage());

}

}

}

Код страницы «AuthorizationPage» (см. рисунок 31)

<Page x:Class="Hospital\_WPF.Pages.AuthorizationPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Поликлиника">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Style="{StaticResource StackPanelStyle}"/>

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

Style="{StaticResource StackPanelStyle}">

<TextBlock FontSize="20"

Text="Логин: "

Width="100"

Height="30"

Margin="-200,90,0,0"

Style="{StaticResource TextBlockStyle}"/>

<TextBox x:Name="TxtBoxLogAuth"

FontSize="20"

Width="150"

Height="30"

Margin="25,-30,0,0"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock IsHitTestVisible="False" Text="Введите ваш логин"

VerticalAlignment="Center" HorizontalAlignment="Left" Foreground="DarkGray"

Margin="435,-26,0,0"

Width="125"

Height="20"

Name="TxtBlockHintLogin">

<TextBlock.Style>

<Style>

<Setter Property="ItemsControl.Visibility" Value="Collapsed"/>

<Style.Triggers>

<DataTrigger Binding="{Binding Text, ElementName=TxtBoxLogAuth}" Value="">

<Setter Property="ItemsControl.Visibility" Value="Visible"/>

</DataTrigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<TextBlock FontSize="20"

Text="Пароль: "

Width="100"

Height="30"

Margin="-200,10,0,0"

Style="{StaticResource TextBlockStyle}"/>

<PasswordBox x:Name="PassBoxPassAuth"

FontSize="20"

Width="150"

Height="30"

Margin="25,-30,0,0"

Style="{StaticResource PassBoxStyle}"/>

<Button x:Name="BtnAuth"

Click="BtnAuth\_Click"

Content="Войти в систему"

Width="200"

Height="30"

FontSize="20"

Margin="0,100,0,0"

Style="{StaticResource BtnStyle}"/>

<Image Margin="-670,-255,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\Hospital\_Snake.png</Image.Source>

</Image>

<Image Margin="550,-255,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\DoctorImage\_Auth.jpg</Image.Source>

</Image>

</StackPanel>

<StackPanel Grid.Row="2" Style="{StaticResource StackPanelStyle}"/>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class AuthorizationPage : Page

{

public AuthorizationPage()

{

InitializeComponent();

TxtBoxLogAuth.MaxLength = 50;

PassBoxPassAuth.MaxLength = 25;

TxtBoxLogAuth.ToolTip = "Введите ваш логин";

PassBoxPassAuth.ToolTip = "Введите ваш пароль, длина которого не менее 6 символов";

}

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

{

String Login = TxtBoxLogAuth.Text;

String Password = PassBoxPassAuth.Password;

if (Login != "" && Password != "")

{

var Level = DBHospitalEntities.GetContext().Doctors\_Table.FirstOrDefault(x => x.User\_login == TxtBoxLogAuth.Text && x.User\_pass == PassBoxPassAuth.Password);

if (Level != null)

{

if (Level.Access\_Level\_Table.id\_access\_level == 1)

{

MessageBox.Show($"Администратор - {Level.Name\_doctor} авторизован!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.Navigate(new MainViewPage(Level));

TxtBoxLogAuth.Text = null;

PassBoxPassAuth.Password = null;

FrameClass.UserName = Level.Name\_doctor;

FrameClass.UserSurname = Level.Surname\_doctor;

}

else if (Level.Access\_Level\_Table.id\_access\_level == 2)

{

MessageBox.Show($"Главный врач - {Level.Name\_doctor}, авторизован!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.Navigate(new MainViewPage(Level));

TxtBoxLogAuth.Text = null;

PassBoxPassAuth.Password = null;

FrameClass.UserName = Level.Name\_doctor;

FrameClass.UserSurname = Level.Surname\_doctor;

}

else if (Level.Access\_Level\_Table.id\_access\_level == 3)

{

MessageBox.Show($"Доктор - {Level.Name\_doctor}, авторизован!", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.Navigate(new MainViewPage(Level));

TxtBoxLogAuth.Text = null;

PassBoxPassAuth.Password = null;

Classes.FrameClass.UserID = Level.id\_doctor;

Classes.FrameClass.LvlAccID = Level.id\_access\_level;

FrameClass.UserName = Level.Name\_doctor;

FrameClass.UserSurname = Level.Surname\_doctor;

}

}

else MessageBox.Show("Данный пользователь не существует", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("Поля не должны быть пустыми", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

}

Код страницы «MainViewPage» (см. рисунок 32)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.MainViewPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Поликлиника"

IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1.4\*"/>

<ColumnDefinition Width="5\*"/>

<ColumnDefinition Width="0.5\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="1.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Column="0" Grid.Row="1"

Orientation="Vertical"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Выберите фильтр: "

Margin="5,5,5,5"

Width="auto"

Style="{StaticResource TextBlockStyle}"/>

<ComboBox Name="CmbDiagnos"

Text="Выберите диагноз: " IsEditable="True"

IsReadOnly="True"

SelectionChanged="CmbDiagnos\_SelectionChanged"

Margin="5,5,5,5"

Width="auto"

Height="25"

Style="{StaticResource CmbStyle}"/>

<TextBox Name="SrcSurnamePatientTxtBox"

TextChanged="SrcSurnamePatientTxtBox\_TextChanged"

Style="{StaticResource TextBoxStyle}"

Width="190"

Margin="5,5,5,5"/>

<TextBlock IsHitTestVisible="False" Text="Введите фамилию пациента"

VerticalAlignment="Center" HorizontalAlignment="Left"

Foreground="DarkGray"

Margin="10,-25,5,5"

Width="175"

Height="18">

<TextBlock.Style>

<Style>

<Setter Property="ItemsControl.Visibility" Value="Collapsed"/>

<Style.Triggers>

<DataTrigger Binding="{Binding Text, ElementName=SrcSurnamePatientTxtBox}" Value="">

<Setter Property="ItemsControl.Visibility" Value="Visible"/>

</DataTrigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<TextBlock Text="Сортировка по дате: " Margin="5,5,5,5"

Style="{StaticResource TextBlockStyle}"

Width="auto"/>

<RadioButton Name="RadBtnAsc" Content="По возрастанию"

Click="RadBtnAsc\_Click" Margin="5,5,5,5"/>

<RadioButton Name="RadBtnDesc" Content="По убыванию"

Click="RadBtnDesc\_Click" Margin="5,5,5,5"/>

<Button Name="ResetFiltrBtn" Margin="5"

Click="ResetFiltrBtn\_Click" Content="Сброс фильтров"

Style="{StaticResource BtnStyle}"

FontSize="16"/>

<Button Name="ExcelOutPutInTemplate" Margin="5"

Click="ExcelOutPutInTemplate\_Click"

Content="Вывод данных в Excel"

FontSize="16"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Список ваших приёмов"

Margin="5,10,5,5"

FontSize="26"

Width="300"

Height="30"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="0"

Style="{StaticResource StackPanelStyle}">

<TextBlock Name="TxtBlockPrivetstvie"

Margin="5,5,5,5"

FontSize="18"

Width="auto"

Height="auto"

TextAlignment="Left"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="2"

Style="{StaticResource StackPanelStyle}">

<Button Name="LogOutBtn" Click="LogOutBtn\_Click"

Width="60"

FontSize="16"

Height="30"

Margin="5,12,5,5"

Content="Выход"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="1"

Style="{StaticResource StackPanelStyle}">

<Button Name="NavToAddReceptionPageBtn" Click="NavToAddReceptionPageBtn\_Click"

Content="Добавить запись" Margin="-215,5,5,5"

Width="250"

FontSize="16"

Height="25"

Style="{StaticResource BtnStyle}"/>

<Button Name="DelReceptionBtn" Content="Удалить запись"

Click="DelReceptionBtn\_Click" Margin="-215,5,5,5"

Width="250"

FontSize="16"

Height="25"

Style="{StaticResource BtnStyle}"/>

<Button Name="NavToViewMedRecPageBtn" Content="Посмотреть историю болезни"

Click="NavToViewMedRecPageBtn\_Click"

FontSize="16"

Width="250"

Height="25"

Margin="300,-100,5,5"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0"

Style="{StaticResource StackPanelStyle}"

Orientation="Vertical">

<Button Name="NavToViewEmployeePageBtn" Margin="5,5,5,5"

Height="25"

FontSize="16"

Content="Список всех сотрудников"

Click="NavToViewEmployeePageBtn\_Click"

Style="{StaticResource BtnStyle}"/>

<Button x:Name="StatsBtn" Margin="5,5,5,5"

Height="25"

FontSize="16"

Content="Статистика"

Click="StatsBtn\_Click"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Grid.Column="2" Grid.RowSpan="2"

Style="{StaticResource StackPanelStyle}"/>

<DataGrid Grid.Column="1" Grid.Row="1"

Name="DtgReceptionsView" AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single">

<DataGrid.Columns>

<DataGridTextColumn Header="Фамилия доктора" Width="\*"

Binding="{Binding Doctors\_Table.Surname\_doctor}"/>

<DataGridTextColumn Header="Фамилия пациента" Width="\*"

Binding="{Binding Patients\_Medical\_Records\_Table.Surname\_patient}"/>

<DataGridTextColumn Header="Дата приёма" Width="\*"

Binding="{Binding Date\_of\_reception, StringFormat=dd/MM/yyyy}"/>

<DataGridTextColumn Header="Жалобы" Width="\*"

Binding="{Binding Complaints}"/>

<DataGridTextColumn Header="Назначение" Width="\*"

Binding="{Binding Appointment}"/>

<DataGridTextColumn Header="Диагноз" Width="\*"

Binding="{Binding Diagnoses\_Table.Name\_diagnose\_rus}"/>

<DataGridTextColumn Header="Дополнение к диагнозу" Width="\*"

Binding="{Binding addition\_to\_diagnose}"/>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Name="NavToEditReceptionsPageBtn" Click="NavToEditReceptionsPageBtn\_Click"

Content="Редактировать"

Width="85"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

Код CS:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Ink;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Animation;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Excel = Microsoft.Office.Interop.Excel;

namespace Hospital\_WPF.Pages

{

public partial class MainViewPage : Page

{

public MainViewPage(Doctors\_Table Level)

{

InitializeComponent();

var Lvlaccuser = Level.id\_access\_level;

var allDiagnosis = DBHospitalEntities.GetContext().Diagnoses\_Table.Distinct().ToList();

allDiagnosis.Insert(0, new Diagnoses\_Table

{

Name\_diagnose\_rus = "Выберите диагноз: "

});

CmbDiagnos.ItemsSource = allDiagnosis;

CmbDiagnos.SelectedValuePath = "id\_diagnose";

CmbDiagnos.DisplayMemberPath = "Name\_diagnose\_rus";

UpdateDtg();

if (Lvlaccuser == 3)

{

NavToViewEmployeePageBtn.Visibility = Visibility.Collapsed;

}

if (Lvlaccuser == 1)

{

NavToAddReceptionPageBtn.Visibility = Visibility.Collapsed;

ExcelOutPutInTemplate.Visibility = Visibility.Collapsed;

}

else if (Lvlaccuser == 2)

{

NavToAddReceptionPageBtn.Visibility = Visibility.Collapsed;

ExcelOutPutInTemplate.Visibility = Visibility.Collapsed;

}

TxtBlockPrivetstvie.TextWrapping = TextWrapping.Wrap;

TxtBlockPrivetstvie.Text = $"Здравствуйте, {Level.Surname\_doctor} {Level.Name\_doctor}";

}

private void UpdateDtg()

{

if (Classes.FrameClass.LvlAccID == 3)

{

var currentreception = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).ToList();

if (CmbDiagnos.SelectedIndex > 0)

currentreception = currentreception.Where(x => x.id\_doctor\_reception == Classes.FrameClass.UserID).Where(x => x.id\_diagnose\_reception == int.Parse(CmbDiagnos.SelectedValue.ToString())).ToList();

currentreception = currentreception.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Patients\_Medical\_Records\_Table.Surname\_patient.ToLower().Contains(SrcSurnamePatientTxtBox.Text.ToLower())).ToList();

; if (RadBtnAsc.IsChecked.Value)

currentreception = currentreception.Where(x => x.id\_doctor\_reception == FrameClass.UserID).OrderBy(x => x.Date\_of\_reception).ToList();

if (RadBtnDesc.IsChecked.Value)

currentreception = currentreception.Where(x => x.id\_doctor\_reception == FrameClass.UserID).OrderByDescending(x => x.Date\_of\_reception).ToList();

DtgReceptionsView.ItemsSource = currentreception;

}

else

{

var currentreception = DBHospitalEntities.GetContext().Reception\_Table.ToList();

if (CmbDiagnos.SelectedIndex > 0)

currentreception = currentreception.Where(x => x.id\_diagnose\_reception == int.Parse(CmbDiagnos.SelectedValue.ToString())).ToList();

currentreception = currentreception.Where(x => x.Patients\_Medical\_Records\_Table.Surname\_patient.ToLower().Contains(SrcSurnamePatientTxtBox.Text.ToLower())).ToList();

if (RadBtnAsc.IsChecked.Value)

currentreception = currentreception.OrderBy(x => x.Date\_of\_reception).ToList();

if (RadBtnDesc.IsChecked.Value)

currentreception = currentreception.OrderByDescending(x => x.Date\_of\_reception).ToList();

DtgReceptionsView.ItemsSource = currentreception;

}

}

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

{

UpdateDtg();

}

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

{

UpdateDtg();

}

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

{

UpdateDtg();

}

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

{

UpdateDtg();

}

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

{

CmbDiagnos.SelectedIndex = 0;

SrcSurnamePatientTxtBox.Text = "";

RadBtnAsc.IsChecked = false;

RadBtnDesc.IsChecked = false;

if (Classes.FrameClass.LvlAccID == 3)

{

DtgReceptionsView.ItemsSource = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).ToList();

}

else

{

DtgReceptionsView.ItemsSource = DBHospitalEntities.GetContext().Reception\_Table.ToList();

}

}

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

{

if (DtgReceptionsView.SelectedCells.Count <= 0)

{

MessageBox.Show("Выберите историю болезни для экспорта данных в Excel", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

}

else if(DtgReceptionsView.SelectedCells.Count >= 1 )

{

var PatientReception = DtgReceptionsView.SelectedItems.Cast<Reception\_Table>().ToList();

try

{

var app = new Excel.Application();

Excel.Workbook wb = app.Workbooks.Add();

app.SheetsInNewWorkbook = 1;

Excel.Worksheet worksheet =

app.Worksheets.Item[1];

worksheet.Name = "Сведения о приёме";

int IndexRows = 1;

worksheet.Cells[1][IndexRows] = "Осмотр доктора";

worksheet.Cells[1][IndexRows+1] = "Дата осмотра: ";

worksheet.Cells[1][IndexRows+2] = "ФИО пациента: ";

worksheet.Cells[1][IndexRows+3] = "Дата рождения: ";

worksheet.Cells[1][IndexRows+4] = "ФИО доктора: ";

worksheet.Cells[1][IndexRows+5] = "Жалобы: ";

worksheet.Cells[1][IndexRows+6] = "Диагноз: ";

worksheet.Cells[1][IndexRows+7] = "Дополнение к диагнозу: ";

worksheet.Cells[1][IndexRows+9] = "Назначение: ";

worksheet.Cells[1][IndexRows+11] = "Подпись доктора: ";

foreach (var \_patientreception in PatientReception)

{

worksheet.Cells[2][IndexRows+1] = \_patientreception.Date\_of\_reception;

worksheet.Cells[2][IndexRows+2] = \_patientreception.Patients\_Medical\_Records\_Table.Surname\_patient + " " +

\_patientreception.Patients\_Medical\_Records\_Table.Name\_patient + " " + \_patientreception.Patients\_Medical\_Records\_Table.Patronymic\_patient;

worksheet.Cells[2][IndexRows+3] = \_patientreception.Patients\_Medical\_Records\_Table.Birth\_date\_patient;

worksheet.Cells[2][IndexRows+4] = \_patientreception.Doctors\_Table.Surname\_doctor + " " +

\_patientreception.Doctors\_Table.Name\_doctor + " " + \_patientreception.Doctors\_Table.Patronymic\_doctor;

worksheet.Cells[2][IndexRows+5] = \_patientreception.Complaints;

worksheet.Cells[2][IndexRows+6] = \_patientreception.Diagnoses\_Table.Name\_diagnose\_rus;

worksheet.Cells[2][IndexRows+7] = \_patientreception.addition\_to\_diagnose;

worksheet.Cells[2][IndexRows+9] = \_patientreception.Appointment;

Excel.Range rangeBorders = worksheet.Range[worksheet.Cells[1][IndexRows], worksheet.Cells[3][IndexRows+9]];

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle =

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle =

Excel.XlLineStyle.xlLineStyleNone;

Excel.Range rangeBoldBorders = worksheet.Range[worksheet.Cells[2][IndexRows + 11], worksheet.Cells[2][IndexRows + 11]];

rangeBoldBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Excel.XlLineStyle.xlContinuous;

Excel.Range rangeLeft = worksheet.Range[worksheet.Cells[1][IndexRows], worksheet.Cells[1][IndexRows+9]];

rangeLeft.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

Excel.Range rangeCenter = worksheet.Range[worksheet.Cells[2][IndexRows], worksheet.Cells[2][IndexRows+9]];

rangeCenter.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

Excel.Range rangeBold = worksheet.Range[worksheet.Cells[1][IndexRows], worksheet.Cells[1][IndexRows+11]];

rangeBold.Font.Bold = true;

Excel.Range rangeFontSizeHeader = worksheet.Range[worksheet.Cells[1][IndexRows], worksheet.Cells[1][IndexRows+11]];

rangeFontSizeHeader.Font.Size = 16;

Excel.Range rangeFontSize = worksheet.Range[worksheet.Cells[2][IndexRows], worksheet.Cells[2][IndexRows+9]];

rangeFontSize.Font.Size = 14;

Excel.Range rangeFontFamily = worksheet.Range[worksheet.Cells[1][IndexRows], worksheet.Cells[2][IndexRows+11]];

rangeFontFamily.Font.Name = "Times New Roman";

worksheet.Columns.AutoFit();

worksheet.Rows.AutoFit();

}

app.Visible = true;

}

catch (Exception ex)

{

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

}

}

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

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

if (Classes.FrameClass.LvlAccID == 3)

{

DtgReceptionsView.ItemsSource = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).ToList();

}

else

{

DtgReceptionsView.ItemsSource = DBHospitalEntities.GetContext().Reception\_Table.ToList();

}

}

}

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

{

FrameClass.frmobj.Navigate(new AddReceptionPage());

}

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

{

var ReceptionsForRemoving =

DtgReceptionsView.SelectedItems.Cast<Reception\_Table>().ToList();

if (MessageBox.Show

($"Удалить {ReceptionsForRemoving.Count()}" + " записей о приёмах?", "Внимание",

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

try

{

DBHospitalEntities.GetContext().Reception\_Table.

RemoveRange(ReceptionsForRemoving);

DBHospitalEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

if (Classes.FrameClass.LvlAccID == 3)

{

DtgReceptionsView.ItemsSource = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).ToList();

}

else

{

DtgReceptionsView.ItemsSource = DBHospitalEntities.GetContext().Reception\_Table.ToList();

}

}

catch (Exception ex)

{

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

}

}

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

{

FrameClass.frmobj.Navigate(new PatientViewPage());

}

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

{

FrameClass.frmobj.Navigate(new ViewEmployeePage());

}

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

{

FrameClass.frmobj.Navigate(new EditReceptionPage((sender as Button).DataContext as Reception\_Table));

}

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

{

FrameClass.frmobj.Navigate(new AuthorizationPage());

Classes.FrameClass.UserID = 0;

Classes.FrameClass.LvlAccID = 0;

Classes.FrameClass.UserName = string.Empty;

Classes.FrameClass.UserSurname = string.Empty;

}

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

{

Classes.FrameClass.frmobj.Navigate(new StatsPage());

}

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

{

UpdateDtg();

}

}

}

Код страницы «StatsPage» (см. рисунок 35)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.StatsPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="StatsPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="0.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Окно статистики"

FontSize="26"

Width="300"

Height="30"

Margin="5,5,5,5"

TextAlignment="Center"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Style="{StaticResource StackPanelStyle}">

<ListBox x:Name="LstBoxStats"

Margin="5,5,5,5"

Width="auto"

Height="auto"

FontSize="16">

</ListBox>

</StackPanel>

<StackPanel Grid.Row="2"

Style="{StaticResource StackPanelStyle}">

<Button x:Name="GoBackBtn" Content="Вернуться назад"

Margin="5,10,5,5"

Click="GoBackBtn\_Click"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class StatsPage : Page

{

public StatsPage()

{

InitializeComponent();

if (FrameClass.UserID == 3)

{

var CountPatient = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Count();

var CountAllergy = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 1).Count();

var CountLuxation = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 2).Count();

var CountPneumonia = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 3).Count();

var CountTachycardia = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 4).Count();

var CountCapitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 5).Count();

var CountAvitominosis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 6).Count();

var CountAmenrheo = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 7).Count();

var CountConjunctivitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 8).Count();

var CountSeborrhoea = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 9).Count();

var CountMastitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 10).Count();

var CountVaricocele = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 11).Count();

var CountTracheitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 12).Count();

var CountPhimao = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 13).Count();

var CountEncopresis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 14).Count();

var CountDelirium = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.id\_doctor\_reception == FrameClass.UserID).Where(x => x.Diagnoses\_Table.id\_diagnose == 15).Count();

LstBoxStats.Items.Add($"Общее количество ваших пациентов - {CountPatient}");

LstBoxStats.Items.Add($"Количество ваших пациентов с аллергией - {CountAllergy}");

LstBoxStats.Items.Add($"Количество ваших пациентов с вывихом - {CountLuxation}");

LstBoxStats.Items.Add($"Количество ваших пациентов с пневмонией - {CountPneumonia}");

LstBoxStats.Items.Add($"Количество ваших пациентов с тахикардией - {CountTachycardia}");

LstBoxStats.Items.Add($"Количество ваших пациентов с головной болью - {CountCapitis}");

LstBoxStats.Items.Add($"Количество ваших пациентов с авитаминозом - {CountAvitominosis}");

LstBoxStats.Items.Add($"Количество ваших пациентов с аменореей - {CountAmenrheo}");

LstBoxStats.Items.Add($"Количество ваших пациентов с конъюктивитом - {CountConjunctivitis}");

LstBoxStats.Items.Add($"Количество ваших пациентов с себореей - {CountSeborrhoea}");

LstBoxStats.Items.Add($"Количество ваших пациентов с маститом - {CountMastitis}");

LstBoxStats.Items.Add($"Количество ваших пациентов с варикоцеле - {CountVaricocele}");

LstBoxStats.Items.Add($"Количество ваших пациентов с трахеитом - {CountTracheitis}");

LstBoxStats.Items.Add($"Количество ваших пациентов с фимозом - {CountPhimao}");

LstBoxStats.Items.Add($"Количество ваших пациентов с энкопрезом - {CountEncopresis}");

LstBoxStats.Items.Add($"Количество ваших пациентов с делирием - {CountDelirium}");

}

else

{

var CountPatient = DBHospitalEntities.GetContext().Reception\_Table.Count();

var CountAllergy = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 1).Count();

var CountLuxation = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 2).Count();

var CountPneumonia = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 3).Count();

var CountTachycardia = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 4).Count();

var CountCapitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 5).Count();

var CountAvitominosis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 6).Count();

var CountAmenrheo = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 7).Count();

var CountConjunctivitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 8).Count();

var CountSeborrhoea = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 9).Count();

var CountMastitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 10).Count();

var CountVaricocele = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 11).Count();

var CountTracheitis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 12).Count();

var CountPhimao = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 13).Count();

var CountEncopresis = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 14).Count();

var CountDelirium = DBHospitalEntities.GetContext().Reception\_Table.Where(x => x.Diagnoses\_Table.id\_diagnose == 15).Count();

LstBoxStats.Items.Add($"Общее количество пациентов - {CountPatient}");

LstBoxStats.Items.Add($"Общее количество пациентов с аллергией - {CountAllergy}");

LstBoxStats.Items.Add($"Общее количество пациентов с вывихом - {CountLuxation}");

LstBoxStats.Items.Add($"Общее количество пациентов с пневмонией - {CountPneumonia}");

LstBoxStats.Items.Add($"Общее количество пациентов с тахикардией - {CountTachycardia}");

LstBoxStats.Items.Add($"Общее количество пациентов с головной болью - {CountCapitis}");

LstBoxStats.Items.Add($"Общее количество пациентов с авитаминозом - {CountAvitominosis}");

LstBoxStats.Items.Add($"Общее количество пациентов с аменореей - {CountAmenrheo}");

LstBoxStats.Items.Add($"Общее количество пациентов с конъюктивитом - {CountConjunctivitis}");

LstBoxStats.Items.Add($"Общее количество пациентов с себореей - {CountSeborrhoea}");

LstBoxStats.Items.Add($"Общее количество пациентов с маститом - {CountMastitis}");

LstBoxStats.Items.Add($"Общее количество пациентов с варикоцеле - {CountVaricocele}");

LstBoxStats.Items.Add($"Общее количество пациентов с трахеитом - {CountTracheitis}");

LstBoxStats.Items.Add($"Общее количество пациентов с фимозом - {CountPhimao}");

LstBoxStats.Items.Add($"Общее количество пациентов с энкопрезом - {CountEncopresis}");

LstBoxStats.Items.Add($"Общее количество пациентов с делирием - {CountDelirium}");

}

}

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

{

Classes.FrameClass.frmobj.GoBack();

}

}

}

Код страницы «PatientViewPage» (см. рисунок 34)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.PatientViewPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="PatientViewPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.8\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="0.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Окно просмотра личных данных пациента"

FontSize="26"

Margin="5,25,5,5"

TextAlignment="Center"

Width="550"

Height="40"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Style="{StaticResource StackPanelStyle}">

<ComboBox Name="CmbSelectPatientMedRec"

IsEditable="True" IsReadOnly="True"

Width="300" Margin="5,85,5,5"

Height="35"

FontSize="20"

Text="Выберите пациента: "

Style="{StaticResource CmbStyle}"

SelectionChanged="CmbSelectPatientMedRec\_SelectionChanged"/>

<TextBlock Text="ФИО пациента" Name="FIOPatientTxtBlock"

Margin="5,5,5,5"

Width="275"

Height="30"

TextAlignment="Left"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<TextBlock Text="Возраст пациента" Name="AgePatientTxtBlock"

Margin="5,5,5,5"

Width="275"

Height="30"

TextAlignment="Left"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<TextBlock Text="Дата рождения пациента" Name="BirthDatePatientTxtBlock"

Margin="5,5,5,5"

Width="275" Height="30"

TextAlignment="Left"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<TextBlock Text="Номер полиса ОМС пациента" Name="InsuranceNumberPatientTxtBlock"

Margin="5,5,5,5"

Width="275"

Height="30"

TextAlignment="Left"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<TextBlock Text="Адрес пациента" Name="AddressPatientTxtBlock"

Margin="5,5,5,5"

Width="275" Height="30"

TextAlignment="Left"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<Image Margin="-670,-255,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\Hospital\_Snake.png</Image.Source>

</Image>

<Image Margin="650,-255,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\DoctorImage\_Auth.jpg</Image.Source>

</Image>

</StackPanel>

<StackPanel Grid.Row="2"

Style="{StaticResource StackPanelStyle}"

Orientation="Horizontal">

<Button Content="Вернуться назад" Name="GoBackBtn"

Click="GoBackBtn\_Click"

Width="175"

Height="30"

FontSize="20"

Margin="400,5,5,5"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class PatientViewPage : Page

{

private Patients\_Medical\_Records\_Table \_records = new Patients\_Medical\_Records\_Table();

public PatientViewPage()

{

InitializeComponent();

var allpatients = DBHospitalEntities.GetContext().Patients\_Medical\_Records\_Table.Distinct().ToList();

allpatients.Insert(0, new Patients\_Medical\_Records\_Table

{

Surname\_patient = "Выберите пациента: "

});

CmbSelectPatientMedRec.ItemsSource = allpatients;

CmbSelectPatientMedRec.SelectedValuePath = "id\_medical\_records";

CmbSelectPatientMedRec.DisplayMemberPath = "Surname\_patient";

}

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

{

if (CmbSelectPatientMedRec.SelectedIndex > 0)

{

FrameClass.PatientID = int.Parse(CmbSelectPatientMedRec.SelectedValue.ToString());

var patientrec = DBHospitalEntities.GetContext().Patients\_Medical\_Records\_Table.Where(x => x.id\_medical\_records == FrameClass.PatientID).FirstOrDefault();

if (patientrec.Patronymic\_patient != null)

{

FIOPatientTxtBlock.Text = $"{patientrec.Surname\_patient} {patientrec.Name\_patient} {patientrec.Patronymic\_patient}";

}

else

{

FIOPatientTxtBlock.Text = $"{patientrec.Name\_patient} {patientrec.Surname\_patient}";

}

AgePatientTxtBlock.Text = patientrec.Age\_patient.ToString();

BirthDatePatientTxtBlock.Text = patientrec.Birth\_date\_patient.Date.ToString();

InsuranceNumberPatientTxtBlock.Text = patientrec.Insurance\_policy\_number\_patient.ToString();

AddressPatientTxtBlock.Text = patientrec.Addres\_patient.ToString();

}

else

{

FIOPatientTxtBlock.Text = "ФИО пациента";

AgePatientTxtBlock.Text = "Возраст пациента";

BirthDatePatientTxtBlock.Text = "Дата рождения пациента";

InsuranceNumberPatientTxtBlock.Text = "Номер полиса ОМС пациента";

AddressPatientTxtBlock.Text = "Адрес пациента";

}

}

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

{

FrameClass.frmobj.GoBack();

}

}

}

Код страницы «AddReceptionPage» (см. рисунок 33)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.AddReceptionPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Поликлиника">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Style="{StaticResource StackPanelStyle}">

<TextBlock Grid.Row="0" Text="Окно добавления записи"

Margin="5,5,5,5"

FontSize="26"

Width="500" Height="40"

TextAlignment="Center"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical"

Style="{StaticResource StackPanelStyle}">

<ComboBox Name="CmbAddMedRecToReception"

Text="Выберите пациента: "

IsEditable="True" IsReadOnly="True"

Margin="5,80,5,5"

Width="300"

Height="35"

FontSize="20"

Style="{StaticResource CmbStyle}"/>

<TextBlock Text="Жалобы: "

FontSize="20"

Width="230"

Height="30"

TextAlignment="Left"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddComplaintsToReception"

FontSize="20"

Width="350"

Height="30"

Margin="485,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Дополнение к диагнозу: "

FontSize="20"

Width="230"

Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddAdditionsToDiagnoseToReception"

Width="350"

Height="30"

Margin="485,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Назначение: "

FontSize="20"

Width="230"

Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddAppointmentToReception"

Width="350"

Height="30"

Margin="485,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<ComboBox Name="CmbAddDiagnoseToReception"

Text="Выберите диагноз: "

IsEditable="True" IsReadOnly="True"

Width="300"

Height="35"

FontSize="20"

Style="{StaticResource CmbStyle}"/>

<Image Margin="-670,-205,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\Hospital\_Snake.png</Image.Source>

</Image>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Horizontal"

Style="{StaticResource StackPanelStyle}">

<Button x:Name="BtnSaveAddPatient" Content="Внести запись"

Click="BtnSaveAddPatient\_Click"

Margin="270,5,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

<Button x:Name="BtnPreviousPage" Content="Вернуться назад" Click="BtnPreviousPage\_Click"

Margin="20,5,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class AddReceptionPage : Page

{

public AddReceptionPage()

{

InitializeComponent();

var alldiagnosis = DBHospitalEntities.GetContext().Diagnoses\_Table.Distinct().ToList();

alldiagnosis.Insert(0, new Diagnoses\_Table

{

Name\_diagnose\_rus = "Выберите диагноз: "

});

CmbAddDiagnoseToReception.ItemsSource = alldiagnosis;

CmbAddDiagnoseToReception.SelectedValuePath = "id\_diagnose";

CmbAddDiagnoseToReception.DisplayMemberPath = "Name\_diagnose\_rus";

var allmedrec = DBHospitalEntities.GetContext().Patients\_Medical\_Records\_Table.Distinct().ToList();

allmedrec.Insert(0, new Patients\_Medical\_Records\_Table

{

Surname\_patient = "Выберите пациента: "

});

CmbAddMedRecToReception.ItemsSource = allmedrec;

CmbAddMedRecToReception.SelectedValuePath = "id\_medical\_records";

CmbAddMedRecToReception.DisplayMemberPath = "Surname\_patient";

TxtBoxAddAdditionsToDiagnoseToReception.MaxLength = 50;

TxtBoxAddAdditionsToDiagnoseToReception.ToolTip = "Максимальная длина 50 символов";

TxtBoxAddAppointmentToReception.MaxLength = 50;

TxtBoxAddAppointmentToReception.ToolTip = "Максимальная длина 50 символов";

TxtBoxAddComplaintsToReception.MaxLength = 50;

TxtBoxAddComplaintsToReception.ToolTip = "Максимальная длина 50 символов";

}

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

{

FrameClass.frmobj.GoBack();

}

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

{

if (TxtBoxAddAdditionsToDiagnoseToReception.Text != "" || TxtBoxAddAppointmentToReception.Text != "" ||

TxtBoxAddComplaintsToReception.Text != "" || CmbAddDiagnoseToReception.SelectedIndex >= 0 ||

CmbAddMedRecToReception.SelectedIndex >= 0)

{

Reception\_Table reception\_Table = new Reception\_Table

{

id\_doctor\_reception = FrameClass.UserID,

id\_medical\_records\_reception = int.Parse(CmbAddMedRecToReception.SelectedValue.ToString()),

Date\_of\_reception = DateTime.Now,

Complaints = TxtBoxAddComplaintsToReception.Text,

Appointment = TxtBoxAddAppointmentToReception.Text,

id\_diagnose\_reception = int.Parse(CmbAddDiagnoseToReception.SelectedValue.ToString()),

addition\_to\_diagnose = TxtBoxAddAdditionsToDiagnoseToReception.Text,

};

if (reception\_Table.id\_reception == 0)

{

DBHospitalEntities.GetContext().Reception\_Table.Add(reception\_Table);

}

try

{

DBHospitalEntities.GetContext().SaveChanges();

MessageBoxResult BoxResult = MessageBox.Show("Запись успешно добавлена. Добавить ещё?", "Сообщение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (BoxResult == MessageBoxResult.Yes)

{

TxtBoxAddAdditionsToDiagnoseToReception.Clear();

TxtBoxAddAppointmentToReception.Clear();

TxtBoxAddComplaintsToReception.Clear();

CmbAddDiagnoseToReception.SelectedIndex = 0;

CmbAddMedRecToReception.SelectedIndex = 0;

}

else

{

FrameClass.frmobj.GoBack();

}

}

catch (Exception ex)

{

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

}

}

else

{

MessageBox.Show("Все поля должны быть заполнены!","Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

}

}

Код страницы «ViewEmployeePage» (см. рисунок 36)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.ViewEmployeePage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="ViewEmployeePage"

IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1.4\*"/>

<ColumnDefinition Width="5\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="6\*"/>

<RowDefinition Height="1.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Список всех сотрудников"

Margin="75,10,5,5"

FontSize="26"

Width="350"

Height="35"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Grid.Column="0"

Orientation="Vertical"

Style="{StaticResource StackPanelStyle}">

<TextBox Name="SrcSurnameDoctorTxtBox"

TextChanged="SrcSurnameDoctorTxtBox\_TextChanged"

Style="{StaticResource TextBoxStyle}"

Width="200"

Height="25"

FontSize="16"

Margin="5,5,5,5"/>

<TextBlock IsHitTestVisible="False" Text="Введите фамилию сотрудника"

VerticalAlignment="Center" HorizontalAlignment="Left"

Foreground="DarkGray"

Margin="10,-30,5,5"

Width="200"

Height="18">

<TextBlock.Style>

<Style>

<Setter Property="ItemsControl.Visibility" Value="Collapsed"/>

<Style.Triggers>

<DataTrigger Binding="{Binding Text, ElementName=SrcSurnameDoctorTxtBox}" Value="">

<Setter Property="ItemsControl.Visibility" Value="Visible"/>

</DataTrigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<TextBox Name="SrcSpecializationTxtBox"

TextChanged="SrcSpecializationTxtBox\_TextChanged"

Style="{StaticResource TextBoxStyle}"

Width="200"

Height="25"

FontSize="16"

Margin="5,5,5,5"/>

<TextBlock IsHitTestVisible="False" Text="Введите специальность сотрудника"

VerticalAlignment="Center" HorizontalAlignment="Left"

Foreground="DarkGray"

Margin="10,-30,5,5"

Width="200"

Height="18">

<TextBlock.Style>

<Style>

<Setter Property="ItemsControl.Visibility" Value="Collapsed"/>

<Style.Triggers>

<DataTrigger Binding="{Binding Text, ElementName=SrcSpecializationTxtBox}" Value="">

<Setter Property="ItemsControl.Visibility" Value="Visible"/>

</DataTrigger>

</Style.Triggers>

</Style>

</TextBlock.Style>

</TextBlock>

<ListBox Name="LstBoxAnalyticsData" Width="auto"

Height="auto"

Margin="5,5,5,5">

<ListBoxItem/>

</ListBox>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="1"

Style="{StaticResource StackPanelStyle}">

<Button Name="NavToAddEmployeePageBtn" Click="NavToAddEmployeePageBtn\_Click"

Content="Добавить сотрудника" Margin="5,15,5,5"

Width="250"

FontSize="16"

Height="25"

Style="{StaticResource BtnStyle}"/>

<Button Name="DelEmployeeBtn" Content="Удалить данные сотрудника"

Click="DelEmployeeBtn\_Click" Margin="5,5,5,5"

Width="250" Height="25"

FontSize="16"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

<StackPanel Grid.Row="2" Grid.Column="0"

Style="{StaticResource StackPanelStyle}">

<Button Name="NavToMainViewPage" Margin="5,25,5,5"

FontSize="16"

Content="Вернуться назад"

Click="NavToMainViewPage\_Click"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

<ListView Grid.Row="1" Grid.Column="1"

Name="LstViewEmployee"

ScrollViewer.HorizontalScrollBarVisibility="Disabled"

SelectionMode="Single">

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Vertical"

HorizontalAlignment="Center"

Width="460"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<StackPanel Width="450"

Orientation="Vertical"

HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Фамилия сотрудника: "/>

<TextBlock Text="{Binding Surname\_doctor}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Имя сотрудника: "/>

<TextBlock Text="{Binding Name\_doctor}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Отчество сотрудника: "/>

<TextBlock Text="{Binding Patronymic\_doctor}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Специализация сотрудника: "/>

<TextBlock Text="{Binding Specialization}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Опыт работы: "/>

<TextBlock Text="{Binding Experience}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Должность: "/>

<TextBlock Text="{Binding Access\_Level\_Table.Name\_of\_access\_level}"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Button Name="NavToEditEmployeePagebtn" Click="NavToEditEmployeePagebtn\_Click"

Content="Редактировать"

FontSize="12"

Width="100"/>

</StackPanel>

<Image Width="150" Height="100"

Stretch="Uniform" HorizontalAlignment="Center"

Margin="300,-120,5,5">

<Image.Source>

<Binding Path="EmployeeImage">

<Binding.TargetNullValue>

<ImageSource>\Resources\DoctorIcon.jpg</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</StackPanel>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class ViewEmployeePage : Page

{

public ViewEmployeePage()

{

InitializeComponent();

UpdateLstView();

}

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

{

FrameClass.frmobj.Navigate(new AddEmployeePage());

}

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

{

var EmployeeForRemoving =

LstViewEmployee.SelectedItems.Cast<Doctors\_Table>().ToList();

if(MessageBox.Show

($"Удалить {EmployeeForRemoving.Count} записей о врачах?", "Внимание",

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

try

{

DBHospitalEntities.GetContext().Doctors\_Table.RemoveRange(EmployeeForRemoving);

DBHospitalEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

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

}

LstViewEmployee.ItemsSource = DBHospitalEntities.GetContext().Doctors\_Table.ToList();

}

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

{

FrameClass.frmobj.GoBack();

}

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

{

FrameClass.frmobj.Navigate(new EditEmployeePage((sender as Button).DataContext as Doctors\_Table));

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

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

LstViewEmployee.ItemsSource = DBHospitalEntities.GetContext().Doctors\_Table.ToList();

}

}

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

{

UpdateLstView();

}

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

{

UpdateLstView();

}

private void UpdateLstView()

{

var LstContext = DBHospitalEntities.GetContext().Doctors\_Table.ToList();

LstContext = LstContext.Where(x => x.Surname\_doctor.ToLower().Contains(SrcSurnameDoctorTxtBox.Text.ToLower())).ToList();

LstContext = LstContext.Where(x => x.Specialization.ToLower().Contains(SrcSpecializationTxtBox.Text.ToLower())).ToList();

LstViewEmployee.ItemsSource = LstContext;

var CountEmployee = LstContext.Count();

var CountDoctors = LstContext.Where(x => x.id\_access\_level == 3).Count();

var CountChiefPhysician = LstContext.Where(x => x.id\_access\_level == 2).Count();

var CountAdministration = LstContext.Where(x => x.id\_access\_level == 1).Count();

LstBoxAnalyticsData.Items[0] = $"Количество работников - {CountEmployee}\n" +

$"Количество докторов - {CountDoctors}\n" +

$"Количество главных врачей - {CountChiefPhysician}\n" +

$"Количество администраторов - {CountAdministration}";

}

}

}

Код страницы «AddEmployeePage» (см. рисунок 37)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.AddEmployeePage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="AddEmployeePage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Style="{StaticResource StackPanelStyle}">

<TextBlock Grid.Row="0" Text="Окно добавления сотрудника"

Margin="5,-2,5,5"

Width="400"

Height="35"

FontSize="26"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Имя сотрудника: "

FontSize="20"

Width="260" Height="30"

Margin="-105,25,5,5"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddNameEmployee"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Фамилия сотрудника: "

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddSurnameEmployee"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Отчество сотрудника: "

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddPatronymicEmployee"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Специализация сотрудника: "

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddSpecializationEmployee"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Опыт работы: "

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddExperienceEmployee"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Логин сотрудника: "

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxAddLoginEmployee"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"/>

<TextBlock Text="Пароль сотрудника: "

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<PasswordBox Name="PassBoxFirstCheck"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource PassBoxStyle}"/>

<TextBlock Text="Повторите пароль:"

FontSize="20"

Width="260" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<PasswordBox Name="PassBoxSecondCheck"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource PassBoxStyle}"/>

<Image HorizontalAlignment="Center" VerticalAlignment="Center"

Stretch="Uniform"

Width="200"

Height="200"

Margin="-640,-305,5,5"

Name="ImageEmployeeElement">

<Image.Source>

<Binding Path="">

<Binding.TargetNullValue>

<ImageSource>\Resources\DoctorIcon.jpg</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock Style="{StaticResource TextBlockStyle}"

Text="Фото сотрудника"

Margin="-620,-65,5,5"

FontSize="20"

Width="180" Height="30"/>

<Button Margin="-625,-5,5,5" Width="60"

Height="60" Name="UploadEmpImgBtn"

Click="UploadEmpImgBtn\_Click"

Content="{StaticResource SecondResImg}"/>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Horizontal"

Style="{StaticResource StackPanelStyle}">

<Button x:Name="BtnSaveAddEmployee" Content="Внести сотрудника"

Click="BtnSaveAddEmployee\_Click"

Margin="270,5,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

<Button x:Name="BtnPreviousPage" Content="Вернуться назад" Click="BtnPreviousPage\_Click"

Margin="20,5,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using Microsoft.Office.Core;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Xml.Linq;

namespace Hospital\_WPF.Pages

{

public partial class AddEmployeePage : Page

{

public byte[] image\_bytes;

public int MaxValue = 5242880;

public int MinValue = 0;

public AddEmployeePage()

{

InitializeComponent();

TxtBoxAddNameEmployee.MaxLength = 50;

TxtBoxAddSurnameEmployee.MaxLength = 50;

TxtBoxAddPatronymicEmployee.MaxLength = 50;

TxtBoxAddSpecializationEmployee.MaxLength = 50;

TxtBoxAddExperienceEmployee.MaxLength = 50;

PassBoxFirstCheck.MaxLength = 25;

PassBoxSecondCheck.MaxLength = 25;

TxtBoxAddLoginEmployee.MaxLength = 50;

PassBoxFirstCheck.PasswordChar = '\*';

PassBoxSecondCheck.PasswordChar = '\*';

TxtBoxAddNameEmployee.ToolTip = "Можно вводить только буквы";

TxtBoxAddSurnameEmployee.ToolTip = "Можно вводить только буквы";

TxtBoxAddPatronymicEmployee.ToolTip = "Можно вводить только буквы";

TxtBoxAddExperienceEmployee.ToolTip = "Можно вводить только цифры";

TxtBoxAddSpecializationEmployee.ToolTip = "Можно вводить только буквы";

TxtBoxAddLoginEmployee.ToolTip = "Введите вад адрес электронной почты";

PassBoxFirstCheck.ToolTip = "Введите пароль длиной не менее 6 символов";

PassBoxSecondCheck.ToolTip = "Введите пароль длиной не менее 6 символов";

}

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

{

if (TxtBoxAddNameEmployee.Text == "" || TxtBoxAddSurnameEmployee.Text == "" ||

TxtBoxAddSpecializationEmployee.Text == "" || TxtBoxAddExperienceEmployee.Text == "" ||

PassBoxFirstCheck.Password == "" || PassBoxSecondCheck.Password == "" ||

TxtBoxAddLoginEmployee.Text == "")

{

MessageBox.Show("Пожалуйста, заполните все поля.", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else if (PassBoxFirstCheck.Password.Length < 6 || PassBoxSecondCheck.Password.Length < 6)

{

MessageBox.Show("Минимальная длина пароля 6 символов, максимальная - 25.", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

string[] datalogin = TxtBoxAddLoginEmployee.Text.Split('@');

if (datalogin.Length == 2)

{

string[] data2login = datalogin[1].Split('.');

if (data2login.Length == 2)

{

if (PassBoxFirstCheck.Password != PassBoxSecondCheck.Password)

{

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

}

else if(image\_bytes.Length > MinValue)

{

Doctors\_Table doctors\_Table = new Doctors\_Table()

{

Name\_doctor = TxtBoxAddNameEmployee.Text,

Surname\_doctor = TxtBoxAddSurnameEmployee.Text,

Patronymic\_doctor = TxtBoxAddPatronymicEmployee.Text,

Specialization = TxtBoxAddSpecializationEmployee.Text,

Experience = int.Parse(TxtBoxAddExperienceEmployee.Text),

id\_access\_level = 3,

User\_login = TxtBoxAddLoginEmployee.Text,

User\_pass = PassBoxSecondCheck.Password,

EmployeeImage = image\_bytes,

};

if (doctors\_Table.id\_doctor == 0)

{

DBHospitalEntities.GetContext().Doctors\_Table.Add(doctors\_Table);

}

try

{

DBHospitalEntities.GetContext().SaveChanges();

MessageBoxResult boxResult = MessageBox.Show("Сотрудник успешно добавлен. Добавить ещё?", "Сообщение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (boxResult == MessageBoxResult.Yes)

{

TxtBoxAddNameEmployee.Clear();

TxtBoxAddSurnameEmployee.Clear();

TxtBoxAddPatronymicEmployee.Clear();

TxtBoxAddSpecializationEmployee.Clear();

TxtBoxAddExperienceEmployee.Clear();

TxtBoxAddLoginEmployee.Clear();

PassBoxFirstCheck.Clear();

PassBoxSecondCheck.Clear();

image\_bytes.SetValue(image\_bytes, MinValue);

}

else

{

FrameClass.frmobj.GoBack();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString(), "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

else

{

Doctors\_Table doctors\_Table = new Doctors\_Table()

{

Name\_doctor = TxtBoxAddNameEmployee.Text,

Surname\_doctor = TxtBoxAddSurnameEmployee.Text,

Patronymic\_doctor = TxtBoxAddPatronymicEmployee.Text,

Specialization = TxtBoxAddSpecializationEmployee.Text,

Experience = int.Parse(TxtBoxAddExperienceEmployee.Text),

id\_access\_level = 3,

User\_login = TxtBoxAddLoginEmployee.Text,

User\_pass = PassBoxSecondCheck.Password,

EmployeeImage = null,

};

if (doctors\_Table.id\_doctor == 0)

{

DBHospitalEntities.GetContext().Doctors\_Table.Add(doctors\_Table);

}

try

{

DBHospitalEntities.GetContext().SaveChanges();

MessageBoxResult boxResult = MessageBox.Show("Сотрудник успешно добавлен. Добавить ещё?", "Сообщение",

MessageBoxButton.YesNo, MessageBoxImage.Question);

if (boxResult == MessageBoxResult.Yes)

{

TxtBoxAddNameEmployee.Clear();

TxtBoxAddSurnameEmployee.Clear();

TxtBoxAddPatronymicEmployee.Clear();

TxtBoxAddSpecializationEmployee.Clear();

TxtBoxAddExperienceEmployee.Clear();

TxtBoxAddLoginEmployee.Clear();

PassBoxFirstCheck.Clear();

PassBoxSecondCheck.Clear();

image\_bytes.SetValue(image\_bytes, MinValue);

}

else

{

FrameClass.frmobj.GoBack();

}

}

catch (Exception ex)

{

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

}

}

}

else

{

MessageBox.Show("Пожалуйста, введите логин формата x@x.x", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

else

{

MessageBox.Show("Пожалуйста, введите логин формата x@x.x", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

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

{

FrameClass.frmobj.GoBack();

}

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

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Title = "Выберите изображение: ";

ofd.Filter = "Все рисунки|\*.jpg;\*.jpeg;\*.png|" +

"Формат JPEG (\*.jpg;\*.jpeg)|\*.jpg;\*.jpeg|" +

"Формат PNG (\*.png)|\*.png";

if (ofd.ShowDialog() == true)

{

image\_bytes = File.ReadAllBytes(ofd.FileName);

MessageBox.Show(ofd.FileName.ToString());

if (image\_bytes.Length > MaxValue)

{

MessageBox.Show("Размер изображения превышает 5 МБ", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

var bi = new BitmapImage();

bi.BeginInit();

bi.CreateOptions = BitmapCreateOptions.IgnoreImageCache;

bi.CacheOption = BitmapCacheOption.OnLoad;

bi.UriSource = new Uri(ofd.FileName);

bi.EndInit();

ImageEmployeeElement.Source = bi;

}

}

else

{

var bi = new BitmapImage();

string path = System.IO.Path.GetFullPath(@"..\..\Resources\DoctorIcon.jpg");

bi.BeginInit();

bi.CreateOptions = BitmapCreateOptions.IgnoreImageCache;

bi.CacheOption = BitmapCacheOption.OnLoad;

bi.UriSource = new Uri(path, UriKind.RelativeOrAbsolute);

bi.EndInit();

ImageEmployeeElement.Source = bi;

}

}

}

}

Код страницы «EditEmployeePage» (см. рисунок 38)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.EditEmployeePage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="EditEmployeePage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="0.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Окно изменения данных"

Margin="5,5,5,5"

FontSize="26"

Width="300"

Height="30"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Имя сотрудника: "

Margin="-105,25,5,5"

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedNameEmp"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding Name\_doctor}"/>

<TextBlock Text="Фамилия сотрудника: "

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedSurnameEmp"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding Surname\_doctor}"/>

<TextBlock Text="Специализация сотрудника: "

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedSpecializationEmp"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding Specialization}"/>

<TextBlock Text="Логин сотрудника: "

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedLoginEmp"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding User\_login}"/>

<TextBlock Text="Пароль сотрудника: "

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<PasswordBox Name="PassBoxChangedPasswordEmpFirstCheck"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource PassBoxStyle}"

PasswordChar="\*"

Tag="{Binding User\_pass}"/>

<TextBlock Text="Повтор пароля: "

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<PasswordBox Name="PassBoxChangedPasswordEmpSecondCheck"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

Style="{StaticResource PassBoxStyle}"

PasswordChar="\*"

Tag="{Binding User\_pass}"/>

<TextBlock Text="Уровень доступа сотрудника: "

FontSize="20"

Width="350" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<ComboBox Name="CmbAccessLevelEmp" Text="Уровень доступа: "

IsEditable="True" IsReadOnly="True"

FontSize="20"

Width="350"

Height="30"

Margin="515,-35,5,5"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Style="{StaticResource CmbStyle}"

SelectedValue="{Binding id\_access\_level}"/>

<Image HorizontalAlignment="Center" VerticalAlignment="Center"

Stretch="Uniform"

Width="200"

Height="200"

Margin="-720,-305,5,5"

Name="ImageEmployeeElement">

<Image.Source>

<Binding Path="EmployeeImage">

<Binding.TargetNullValue>

<ImageSource>\Resources\DoctorIcon.jpg</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock Style="{StaticResource TextBlockStyle}"

Text="Фото сотрудника"

Margin="-700,-65,5,5"

FontSize="20"

Width="180" Height="30"/>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Horizontal"

Style="{StaticResource StackPanelStyle}">

<Button Name="SaveChangesInDoctorsTableBtn"

Click="SaveChangesInDoctorsTableBtn\_Click"

Content="Сохранить изменения"

Margin="270,-15,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

<Button Name="GoBackBtn" Click="GoBackBtn\_Click"

Content="Вернуться назад"

Margin="5,-15,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

public partial class EditEmployeePage : Page

{

private Doctors\_Table \_doctortable = new Doctors\_Table();

public byte[] image\_bytes;

public int MaxValue = 5242880;

public int MinValue = 0;

public EditEmployeePage(Doctors\_Table SelectedDoctor)

{

InitializeComponent();

var allAccessLevel = DBHospitalEntities.GetContext().Access\_Level\_Table.Distinct().ToList();

allAccessLevel.Insert(0, new Access\_Level\_Table

{

Name\_of\_access\_level = "Уровень доступа:"

});

CmbAccessLevelEmp.ItemsSource = allAccessLevel;

CmbAccessLevelEmp.SelectedValuePath = "id\_access\_level";

CmbAccessLevelEmp.DisplayMemberPath = "Name\_of\_access\_level";

if (SelectedDoctor != null)

{

\_doctortable = SelectedDoctor;

}

DataContext = \_doctortable;

TxtBoxChangedNameEmp.MaxLength = 50;

TxtBoxChangedSurnameEmp.MaxLength = 50;

TxtBoxChangedLoginEmp.MaxLength = 50;

TxtBoxChangedSpecializationEmp.MaxLength = 50;

PassBoxChangedPasswordEmpFirstCheck.MaxLength = 25;

PassBoxChangedPasswordEmpSecondCheck.MaxLength = 25;

TxtBoxChangedNameEmp.ToolTip = "Вводить только буквы. Максимальная длина строки 50 символов";

TxtBoxChangedSurnameEmp.ToolTip = "Вводить только буквы. Максимальная длина строки 50 символов";

TxtBoxChangedSpecializationEmp.ToolTip = "Вводить только буквы. Максимальная длина строки 50 символов";

TxtBoxChangedLoginEmp.ToolTip = "Введите свой адрес электронной почты";

PassBoxChangedPasswordEmpFirstCheck.PasswordChar = '\*';

PassBoxChangedPasswordEmpSecondCheck.PasswordChar = '\*';

}

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

{

if (TxtBoxChangedNameEmp.Text == "" || TxtBoxChangedSurnameEmp.Text == "" ||

TxtBoxChangedLoginEmp.Text == "" || TxtBoxChangedSpecializationEmp.Text == "" ||

PassBoxChangedPasswordEmpFirstCheck.Password == "" || PassBoxChangedPasswordEmpSecondCheck.Password == "" ||

CmbAccessLevelEmp.SelectedIndex <= 0)

{

MessageBox.Show("Пожалуйста, заполните все поля!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

else if (PassBoxChangedPasswordEmpFirstCheck.Password.Length < 6 || PassBoxChangedPasswordEmpSecondCheck.Password.Length < 6)

{

MessageBox.Show("Минимальная длина пароля 6 символов, максимальная - 25", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

string[] datalogin = TxtBoxChangedLoginEmp.Text.Split('@');

if (datalogin.Length == 2)

{

string[] data2login = datalogin[1].Split('.');

if (data2login.Length == 2)

{

if(PassBoxChangedPasswordEmpFirstCheck.Password != PassBoxChangedPasswordEmpSecondCheck.Password)

{

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

}

else if(image\_bytes == null || image\_bytes.Length > MinValue)

{

Doctors\_Table doctors\_Table = new Doctors\_Table()

{

Name\_doctor = TxtBoxChangedNameEmp.Text,

Surname\_doctor = TxtBoxChangedSurnameEmp.Text,

Patronymic\_doctor = \_doctortable.Patronymic\_doctor,

Specialization = TxtBoxChangedSpecializationEmp.Text,

Experience = \_doctortable.Experience,

id\_access\_level = int.Parse(CmbAccessLevelEmp.SelectedValue.ToString()),

User\_login = TxtBoxChangedLoginEmp.Text,

User\_pass = PassBoxChangedPasswordEmpSecondCheck.Password,

};

try

{

DBHospitalEntities.GetContext().SaveChanges();

MessageBox.Show("Данные успешно изменены", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.GoBack();

}

catch (Exception ex)

{

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

}

}

else

{

Doctors\_Table doctors\_Table = new Doctors\_Table()

{

Name\_doctor = TxtBoxChangedNameEmp.Text,

Surname\_doctor = TxtBoxChangedSurnameEmp.Text,

Patronymic\_doctor = \_doctortable.Patronymic\_doctor,

Specialization = TxtBoxChangedSpecializationEmp.Text,

Experience = \_doctortable.Experience,

id\_access\_level = int.Parse(CmbAccessLevelEmp.SelectedValue.ToString()),

User\_login = TxtBoxChangedLoginEmp.Text,

User\_pass = PassBoxChangedPasswordEmpSecondCheck.Password,

};

try

{

DBHospitalEntities.GetContext().SaveChanges();

MessageBox.Show("Данные успешно изменены", "Сообщение", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.GoBack();

}

catch (Exception ex)

{

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

}

}

}

else

{

MessageBox.Show("Пожалуйста введите логин формата x@x.x", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

else

{

MessageBox.Show("Пожалуйста введите логин формата x@x.x", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

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

{

FrameClass.frmobj.GoBack();

}

}

}

Код страницы «EditReceptionPage» (см. рисунок 39)

Xaml код:

<Page x:Class="Hospital\_WPF.Pages.EditReceptionPage"

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:Hospital\_WPF.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="1000"

Title="Поликлиника">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="0.5\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Окно изменения данных"

FontSize="26"

Width="350"

Height="35"

Margin="5,5,5,5"

Style="{StaticResource TextBlockStyle}"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical"

Style="{StaticResource StackPanelStyle}">

<TextBlock Text="Дата приёма: "

Margin="-105,100,5,5"

FontSize="20"

Width="230" Height="30"

Style="{StaticResource TextBlockStyle}"/>

<DatePicker Name="DtPickChangedDateOfReception"

Margin="485,-35,5,5"

Width="350"

Height="30"

FontSize="20"

DisplayDate="{Binding Date\_of\_reception}"

SelectedDate="{Binding Date\_of\_reception}"/>

<TextBlock Text="Жалобы: "

Width="230" Height="30"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedComplaints"

FontSize="20"

Width="350"

Height="30"

Margin="485,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding Complaints}"/>

<TextBlock Text="Назначение: "

Width="230" Height="30"

FontSize="20"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedAppointment"

FontSize="20"

Width="350"

Height="30"

Margin="485,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding Appointment}"/>

<TextBlock Text="Дополнение к диагнозу: "

FontSize="20"

Width="230"

Height="30"

Style="{StaticResource TextBlockStyle}"/>

<TextBox Name="TxtBoxChangedAdditionToDiagnose"

Width="350" Height="30"

FontSize="20"

Margin="485,-35,5,5"

Style="{StaticResource TextBoxStyle}"

Text="{Binding addition\_to\_diagnose}"/>

<Image Margin="-670,-175,5,5"

Width="250"

Height="250"

Stretch="Uniform">

<Image.Source>\Resources\Hospital\_Snake.png</Image.Source>

</Image>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Horizontal"

Style="{StaticResource StackPanelStyle}">

<Button Name="SaveChangesInReceptionBtn"

Click="SaveChangesInReceptionBtn\_Click"

Content="Сохранить изменения"

Margin="270,-10,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

<Button Name="GoBackToMainViewPageBtn" Click="GoBackToMainViewPageBtn\_Click"

Content="Вернуться назад"

Margin="5,-10,5,5"

Width="250"

Height="30"

FontSize="20"

Style="{StaticResource BtnStyle}"/>

</StackPanel>

</Grid>

</Page>

Cs код:

using Hospital\_WPF.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.CompilerServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Hospital\_WPF.Pages

{

/// <summary>

/// Логика взаимодействия для EditReceptionPage.xaml

/// </summary>

public partial class EditReceptionPage : Page

{

private Reception\_Table \_receptiontable = new Reception\_Table();

public EditReceptionPage(Reception\_Table SelectedReception)

{

InitializeComponent();

if (SelectedReception != null)

\_receptiontable = SelectedReception;

DataContext = \_receptiontable;

TxtBoxChangedAdditionToDiagnose.MaxLength = 150;

TxtBoxChangedAdditionToDiagnose.ToolTip = "Максимальная длина строки 150 символов";

TxtBoxChangedAppointment.MaxLength = 150;

TxtBoxChangedAppointment.ToolTip = "Максимальная длина строки 150 символов";

TxtBoxChangedComplaints.MaxLength = 150;

TxtBoxChangedComplaints.ToolTip = "Максимальная длина строки 150 символов";

}

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

{

if (TxtBoxChangedComplaints.Text != "" || TxtBoxChangedAdditionToDiagnose.Text != "" || TxtBoxChangedAppointment.Text != "")

{

Reception\_Table receptionTable = new Reception\_Table()

{

id\_reception = \_receptiontable.id\_reception,

id\_doctor\_reception = \_receptiontable.id\_doctor\_reception,

id\_medical\_records\_reception = \_receptiontable.id\_medical\_records\_reception,

Date\_of\_reception = DateTime.Parse(DtPickChangedDateOfReception.SelectedDate.Value.Date.ToString()),

Complaints = TxtBoxChangedComplaints.Text,

Appointment = TxtBoxChangedAppointment.Text,

id\_diagnose\_reception = \_receptiontable.id\_diagnose\_reception,

addition\_to\_diagnose = TxtBoxChangedAdditionToDiagnose.Text,

};

try

{

DBHospitalEntities.GetContext().SaveChanges();

MessageBox.Show("Данные успешно изменены", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

FrameClass.frmobj.GoBack();

}

catch (Exception ex)

{

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

}

}

else

{

MessageBox.Show("Все поля должны быть заполнены!", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

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

{

FrameClass.frmobj.GoBack();

}

}

}