**Оглавление**

[Аннотация 4](#_Toc26196310)

[Введение 5](#_Toc26196311)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ 6](#_Toc26196312)

[1.1 Основное функциональное назначение программного средства 6](#_Toc26196313)

[1.2 Полное наименование программного средства 6](#_Toc26196314)

[1.3 Условное обозначение программного средства 6](#_Toc26196315)

[1.4 Разработчики программного средства 6](#_Toc26196316)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc26196317)

[2.1 Основание для разработки 7](#_Toc26196318)

[2.2 Назначение разработки 7](#_Toc26196319)

[2.3 Требование к программному средству 7](#_Toc26196320)

[2.4 Требования к программной документации 8](#_Toc26196321)

[2.5 Требования к эргономике и технической эстетике 8](#_Toc26196322)

[2.6 Стадии и этапы разработки 9](#_Toc26196323)

[2.7 Порядок контроля и приемки 10](#_Toc26196324)

[3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ 11](#_Toc26196325)

[3.1 Декомпозиция поставленной задачи 11](#_Toc26196326)

[3.2 Общая архитектура программного средства 12](#_Toc26196327)

[3.3 Разработка алгоритма решения задачи 15](#_Toc26196328)

[3.4 Реализация функционального назначения программного средства 16](#_Toc26196329)

[3.5 Структурная организация данных 16](#_Toc26196330)

[3.6 Разработка интерфейса ПС](#_Toc26196331) 24

[3.7 Описание структуры выходной информации](#_Toc26196332) 30

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 31](#_Toc26196333)

[4.1 Назначение программного средства 31](#_Toc26196334)

[4.2 Условия выполнения программного средства 31](#_Toc26196335)

[4.3 Эксплуатация программного средства](#_Toc26196336) 31

[4.4 Сообщения пользователю 37](#_Toc26196337)

[5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА 39](#_Toc26196338)

[5.1 Обоснование необходимого количества тестов 39](#_Toc26196339)

[5.2 Описание тестовых пакетов 40](#_Toc26196340)

[6. ЗАКЛЮЧЕНИЕ](#_Toc26196341) 50

[6.1 Оценка качества программного средства с помощью метрик 50](#_Toc26196342)

[7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 53](#_Toc26196343)

[Приложение А листинг программного модуля](#_Toc26196344) 55

[Приложение B диск с исполняемым кодом программного модуля](#_Toc26196345) 128

**Аннотация**

На курсовой проект студента Адельгареева Р.Р. на тему: Разработка интегрированного программного модуля «Санаторий». Курсовой проект представлен пояснительной запиской на 65 листов машинописного текста. Состоит из введения, 5 глав, заключения и списка литературных источников.

В главе 1 содержится общие сведения о программном средстве Санаторий. В главе 2 приведено техническое задание на проектирование БД согласно ГОСТ 19.201. Глава 3 содержит пояснительную записку к программному продукту согласно ГОСТ 19.404-79. В главе 4 содержится руководство пользователя согласно ГОСТ 19.504-79. Глава 5 содержит обоснование количества необходимых тестов и тестовые пакеты для оценки работоспособности программного продукта. В главе 6 приведены результаты ручного и автоматического тестирования программного продукта и выполнена оценка качества с помощью метрик.

В работе содержится общее описание программного средства, предназначенного для объекта проектирования – интегрированного программного модуля «Санаторий», изложены требования к программному средству, описаны его архитектура, реализация функций, интерфейс, приведено руководство пользователя, обоснованы тестовые пакеты и выполнена оценка качества программного продукта с помощью метрик.

**Введение**

В курсовом проекте была поставлена задача разработки интегрированного программного модуля «Санаторий», который позволяет автоматизировать бронирование санаториев.

Для достижения поставленной цели был составлен основной алгоритм работы программы, а также разработаны алгоритмы вспомогательных процедур.

Для реализации алгоритмов был выбран язык программирования C#. Выбор обусловлен тем, что возможностей этого языка достаточно для достижения поставленной цели моей курсового проекта.

При разработке интерфейса программы, я учитывал наиболее нужные и важные функциональные возможности, которые должна выполнять данная программа, а также тип пользователей, которые с ней будут работать.

**1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ**

**1.1 Основное функциональное назначение программного средства**

Программа предназначена для автоматизации бронирования санаториев. Программа не требует владения современными информационными технологиями для эффективного использования данного программного средства, т.е. подходит для любого пользователя.

**1.2 Полное наименование программного средства**

Разработка интегрированного программного модуля «Санаторий».

**1.3 Условное обозначение программного средства**

«SanatoriyApp»

**1.4 Разработчики программного средства**

Программное средство разработала студент Зеленодольского Механического колледжа группы 205 Адельгареев Р.Р.

**2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**2.1 Основание для разработки**

Работа ведётся на основании задания на курсовой проект по специальности 09.02.03 — «Программирование в компьютерных системах».

**2.2 Назначение разработки**

Программа «SanatoriyApp» работает под управление операционной системы Windows 7, Windows 8, Windows 10. Программа предназначена для автоматизации бронирования доступных мест в санатории. Программа позволяет вводить, изменять и удалять данные о санаториях, просматривать категории.

**2.3 Требование к программному средству**

«SanatoriyApp» устанавливается на персональный компьютер, имеющий представленные минимальные системные требования:

Оперативная память: 4 Гб;

Процессор: Intel Core i / Phenom II X4 / AMD Athlon64 X2 4600 + 2.4 ГГц;

Видео карта: nVidia GeForce 8600 / ATI Radeon HD 2400;

Звуковая карта: Совместимая с DirectX;

Программный продукт «SanatoriyApp» является кроссплатформенным продуктом, что позволяет не ограничиваться в работе с определенной версией ОС.

«SanatoriyApp» написана на объектно-ориентированном языке программировании C# (си-шарп) в интегрированной среде разработки Visual Studio 2019. В качестве СУБД используется MS SQL. Программа не может подлежать сторонней оптимизации и редактированию программного кода.

**2.4 Требования к программной документации**

Состав программной документации должен включать в себя:

1. техническое задание;

2. сроки выполнения работ;

3. программу и методику испытаний;

4. эксплуатационные инструкции пользователю;

**2.5 Требования к эргономике и технической эстетике**

Система обеспечивает удобный для пользователей системный интерфейс, отвечающий следующим требованиям:

- единый стиль оформления для пользовательских интерфейсов;

- удобная, интуитивно понятная навигация в интерфейсе пользователя;

- взаимодействие пользователя с системой осуществляется на двух языках по выбору (русский и английский), для наибольшего удобства

Пользовательские интерфейсы системы спроектированы и разработаны с применением единых принципов графического представления информации и организации доступа к функциональным возможностям и сервисам. Разработан графический дизайн пользовательских интерфейсов, цветовые, шрифтовые и композиционные решения для отображения текстов, изображений, таблиц, гиперссылок, управляющих и навигационных элементов (меню, кнопок, форм и т.п.).

Экранные формы спроектированы с учетом требований унификации:

- все экранные формы пользовательского интерфейса выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

- для обозначения сходных операций использованы сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении унифицированы;

- внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) реализованы одинаково для однотипных элементов.

**2.6 Стадии и этапы разработки**

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4. определение стадий, этапов и сроков разработки программы и документации на нее;

5. выбор языков программирования;

6. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с пунктом Предварительный состав программной документации настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение порядка и методики испытаний;

2. проведение испытаний;

3. корректировка программы и программной документации по результатам испытаний.

**2.7 Порядок контроля и приемки**

Сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом. Сдача-приемка осуществляется комиссией, в состав которой входят представители ГАПОУ и работодатель. По результатам приемки подписывается акт приемочной комиссии.

В указанном подразделе, согласно таблице «Сроки выполнения работ» технического задания, будет описаны перечень участвующих организаций, место и сроки проведения работ, согласно п. 2.8 ГОСТ 34.602-89.

Порядок согласования и утверждения приемочной документации должен регламентироваться организационно-распорядительной документацией организации, принимающей участие в создании программного продукта. Согласно разделу «Приемка результатов разработки» ГОСТ 15.001-88 для согласования и утверждения приемочной документации создается приемочная комиссия (приказом).

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

**3. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ**

**3.1 Декомпозиция поставленной задачи**

Для достижения цели работы была поставлена следующая задача:

Создать модуль “CodeDatabase” отвечающий за сохранение БД. Результат работы модуля - сохранение БД.

Поставленные задачи наглядно отображаются на диаграмме деятельности.

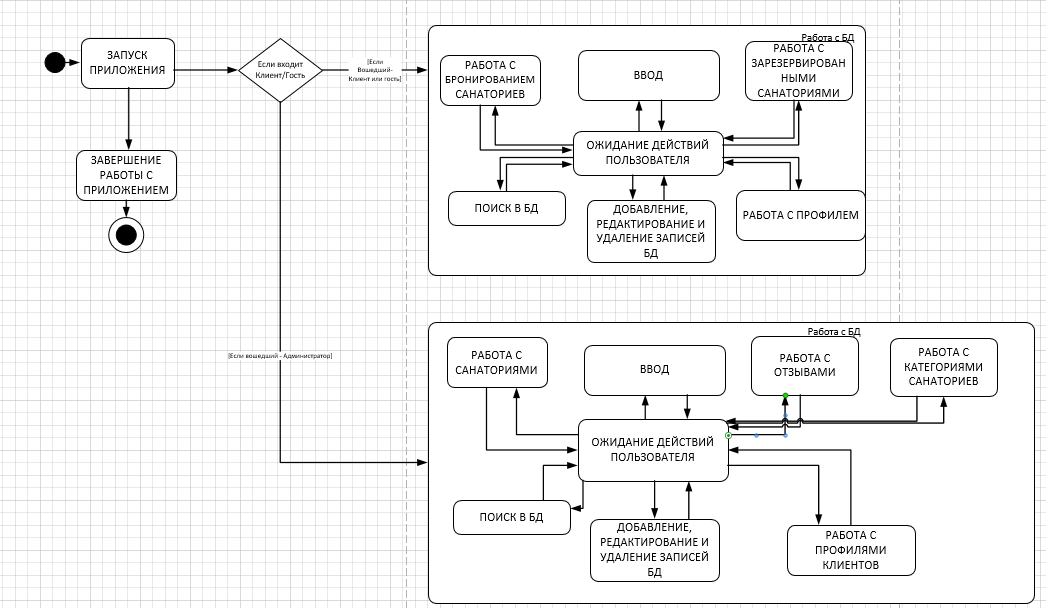


Рис.1 Декомпозиция поставленной задачи

**3.2 Общая архитектура программного средства**

Объектом моделирования является Санаторий.

Рассматриваемые процессы: Добавление и удаление информации о мероприятиях.

Объекты моделирования представлены на диаграмме классов, рис.2.

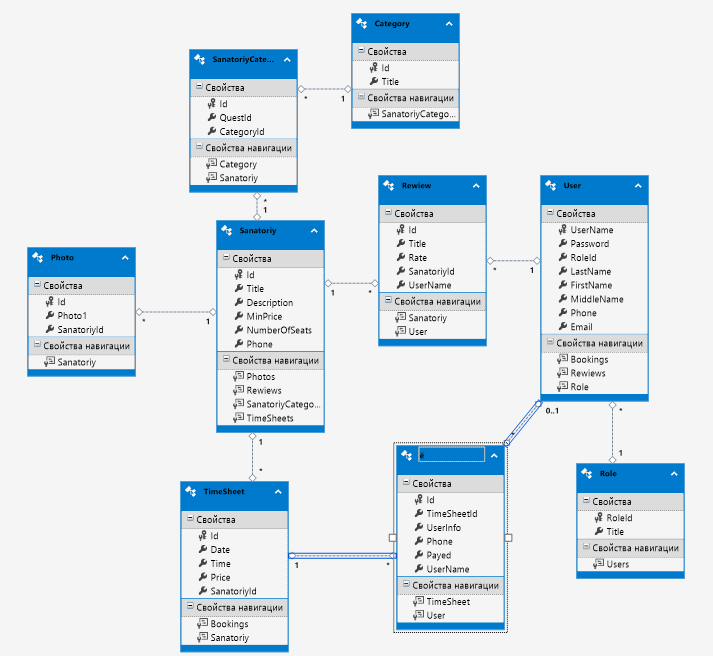


Рис.2 Диаграмма классов

По полученной декомпозиции задач была спроектирована следующая архитектура программного средства.

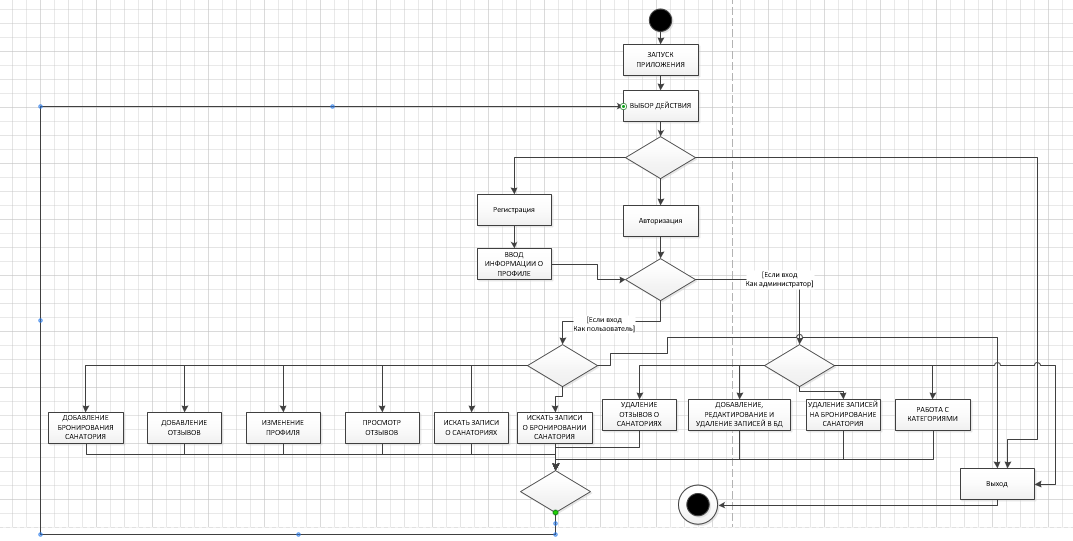


Рис.3 Схема событийно-ориентированной архитектуры

В «SanatoriyApp» используется событийно-ориентированная архитектура.(Рис.3) Роль агента (источник событий) в программе выполняют: пользователи программы, за роль стоков (потребители событий) отвечают таблицы, входящие в базу данных. Например, когда пользователь выбирает определенное действие: добавить, удалить, редактировать, вывести и т.д. система осуществит выбранные действия, и база отреагирует соответствующим образом: запись добавлена, удалена, отредактирована, либо отчет выведен на экран. Системная архитектура пользователя рассматривает это изменение состояния как событие, создаваемое, публикуемое, определяемое и потребляемое различными приложениями в составе архитектуры.

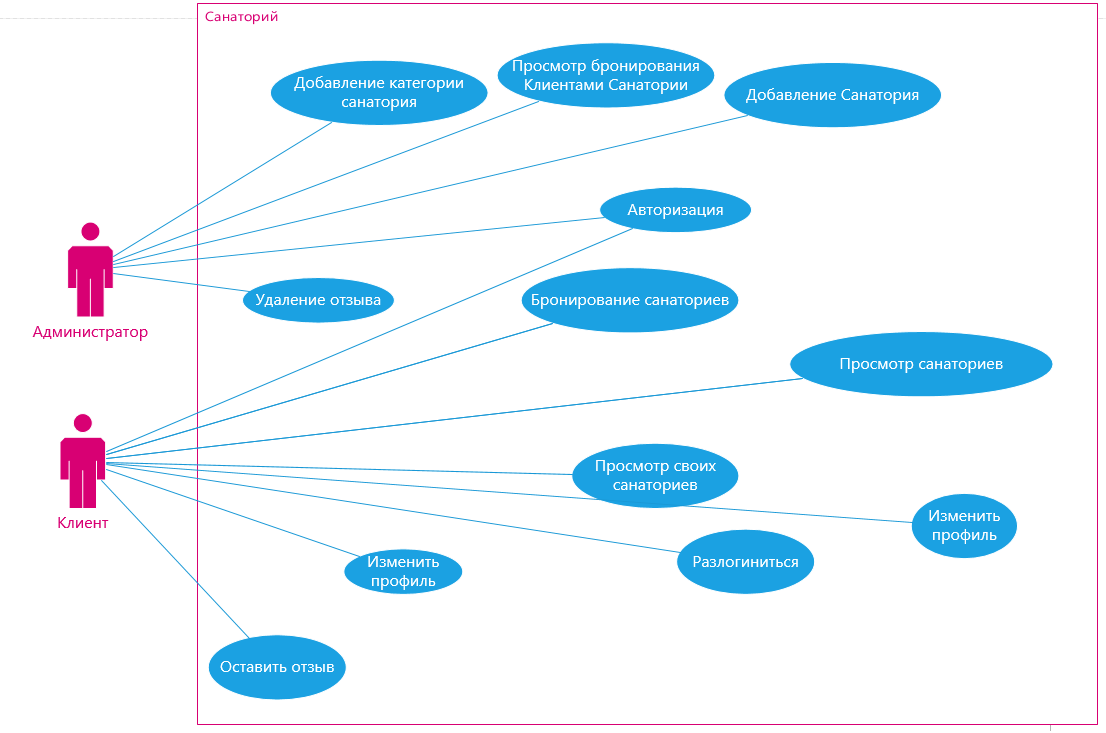
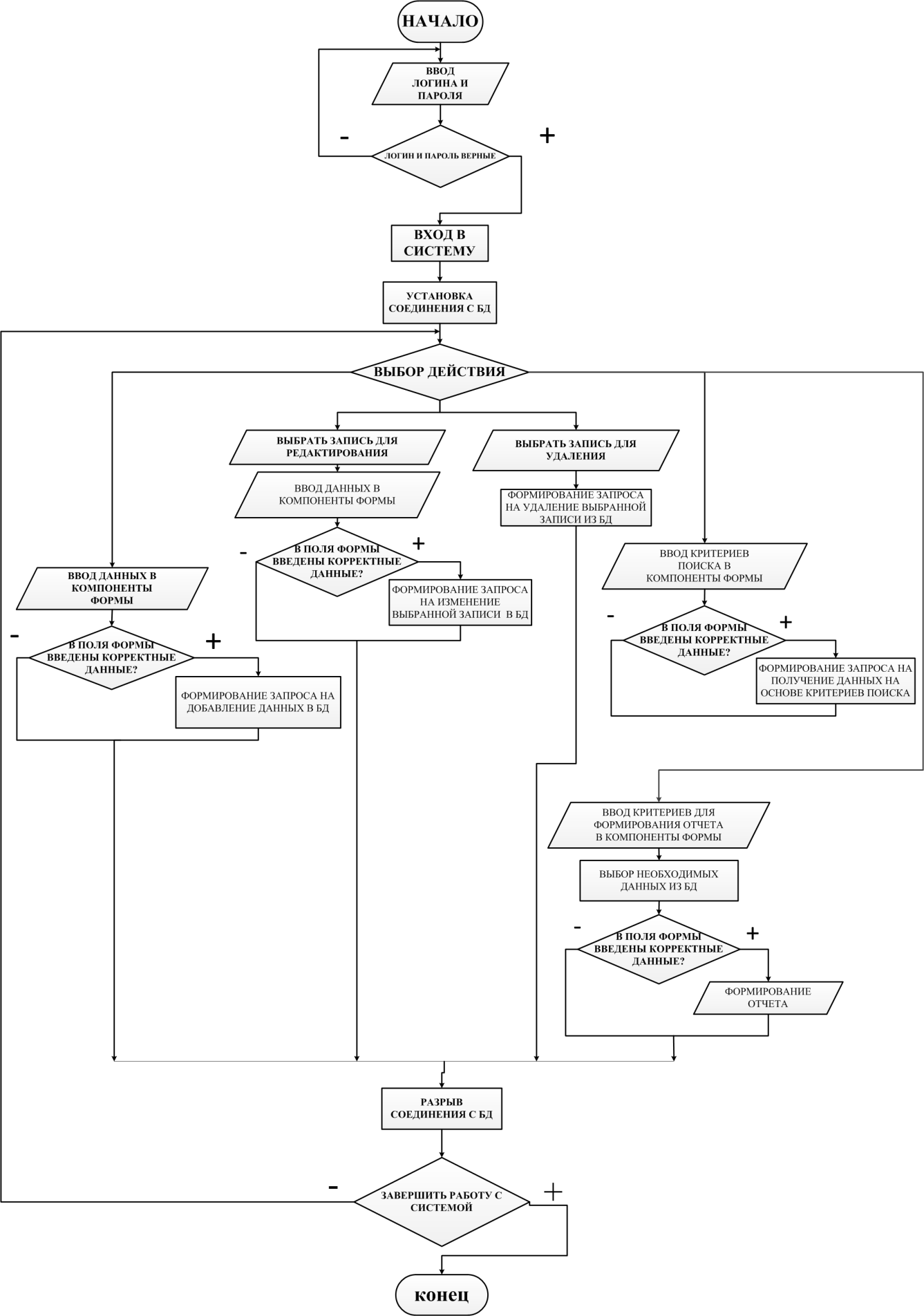


Рис.4 Диаграмма вариантов использования

**3.3 Разработка алгоритма решения задачи**

**Общий алгоритм**

****Рис.5 Алгоритм программы

**3.4 Реализация функционального назначения программного средства**

Программа «SanatoriyApp» имеет следующий набор входных данных, такие как: санатории и информация о них.

Данные вводятся администратором в соответствующие поля ввода, снабженные всплывающими подсказками.

Выходными данными являются: отображение нового списка санаториев в виде плиточного интерфейса.

Данные вводятся пользователем в соответствующие поля вывода, снабженные всплывающими подсказками.

Выходные данные редактировать вручную невозможно, т.к. они служат в качестве отчетной информации для пользователя.

**3.5 Структурная организация данных**

Для создания БД необходимо определиться с данными, которые необходимы для полноценного функционирования системы. Все эти данные указаны в реляционной модели «БД Санаторий» представленная на рисунке. Любая реляционная база данных и называется реляционной, что характеризуется отношениями (relation) между таблицами. На рисунке изображены таблицы моей базы данных. При этом одна таблица является родительской (главной), а вторая – дочерней (подчиненной). Главной таблицей является «Пользователь». Реляционная модель автоматизированной системы соответствует всем 12 правилам Кодда. (рис.6)

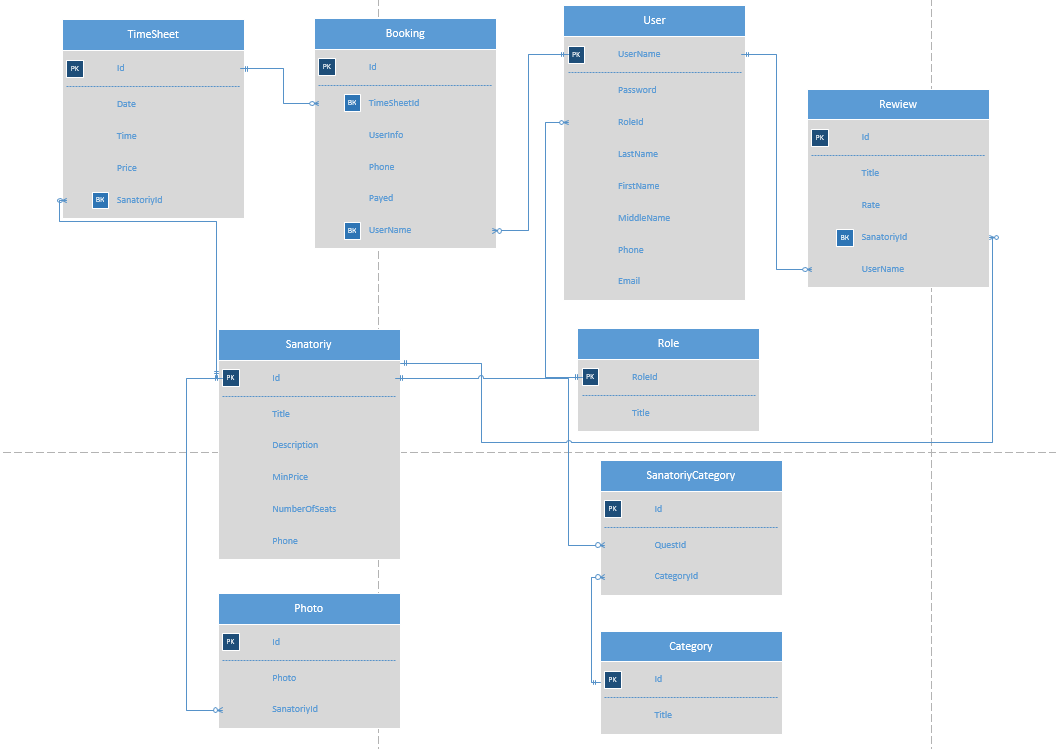


Рис.6 Реляционная модель

Первичный ключ в базе уникален, используется для организации отношений между таблицами, который не может иметь пустых и повторяющихся значений. Первичными ключами в базе являются поля: Id (таблица TimeSheet),Id (таблица Booking),Id (таблица Sanatoriy),RoleId (таблица Role), Id (таблица Sanatoriy),Id (таблица Photo),Id (таблица Rewiew),Id (таблица SanatoriyCategory),Id (таблица Category), UserName (таблица UserName). Остальные ID-ключи являются внешними ключами.

Для организации более эффективной обработки данных применяется нормализация. Таблицы моей БД находятся в 3НФ:

* БД находится в форме -1НФ потому, что
  + Таблица не имеет повторяющихся записей;
  + Каждый атрибут отношения хранит одно-единственное значение и не является списком, ни множеством значений;
  + Таблица не имеет повторяющихся групп полей.
    - Вторая нормальная форма(2НФ):
      * Устранены атрибуты, зависящие только от части уникального (первичного) идентификатора, т.е. ID.
        + Третья нормальная форма(3НФ):

Отсутствуют атрибуты, зависящие от атрибутов, не входящих в уникальный (первичный) ключ.

На основе реляционной модели базы данных мною в MS SQL была построена следующая база данных. (Рис.7) В ней первичным ключом является поле таблицы “Photo” – “Id”. Связь таблиц “Sanatoriy” и “Photo” по полю “SanatoriyId” один ко многим (1:М). Таблицы “Category” и “SanatoriyCategory” связаны по полю “CategoryId” типом связи 1:М. Таблицы “Sanatory” и “SanatoriyCategory” связаны по полю “QuestId” типом связи 1:М. Таблицы “Sanatory” и “TimeSheet” связаны по полю “SanatoriyId” типом связи 1:M. Таблицы “Role” и “User” связаны по полю “RoleId” типом связи 1:M. Таблицы “User” и “Booking” связаны по полю “UserName” типом связи 1:M. Таблицы “Booking” и “TimeSheet” связаны по полю “TimeSheetId” типом связи M:1. Таблицы “Rewiew” и “User” связаны по полю “UserName” типом связи M:1. Таблицы “Sanatory” и “Rewiew” связаны по полю “SanatoriyId” типом связи 1:М. Все связи между таблицами базы данных типа один ко многим.

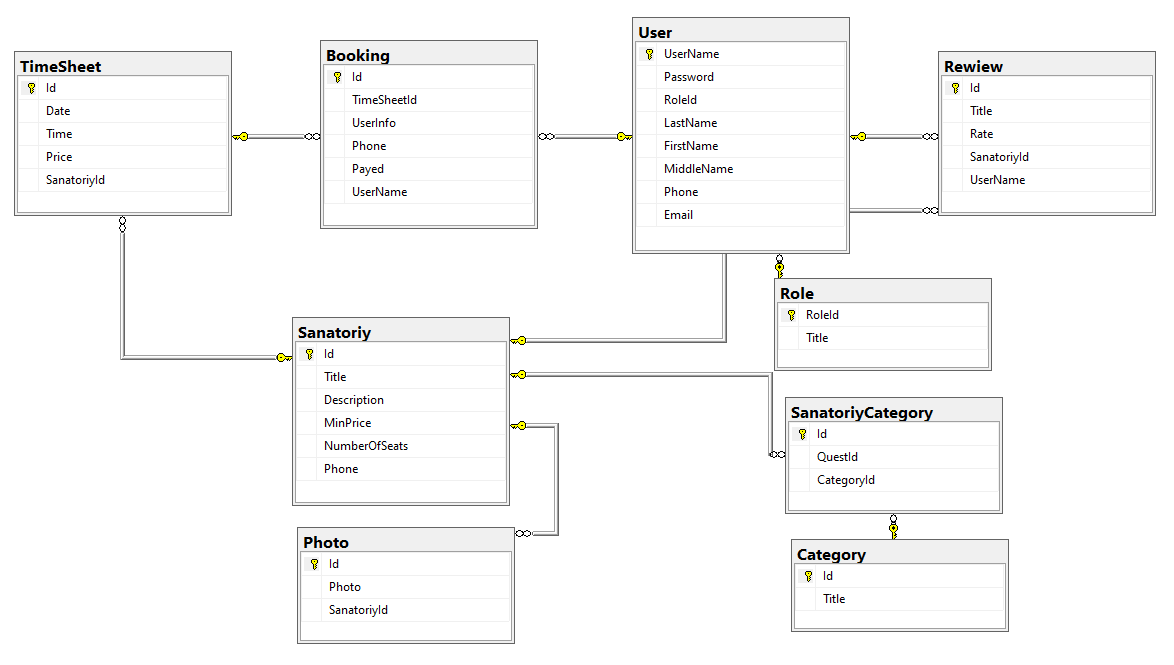


Рис.7 Схема базы данных

Таблица “User”(Пользователи)

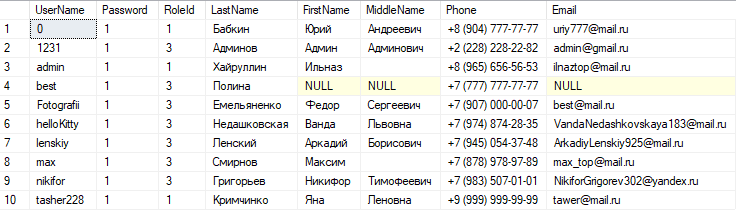


Рис.8 Таблица User

Таблица “TimeSheet”( Табель учета рабочего времени)

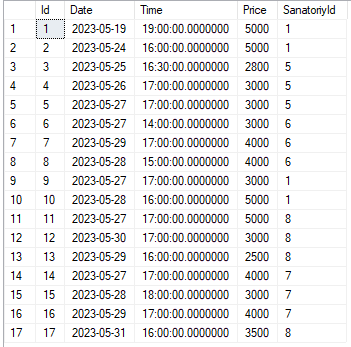


Рис.9 Таблица TimeSheet

Таблица “SanatoriyCategory”(КатегорииСанатория)

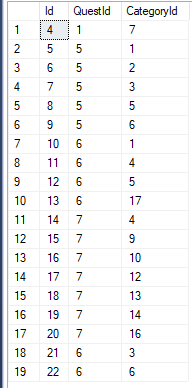


Рис.10 Таблица SanatoriyCategory

Таблица “Sanatoriy”(Санаторий)

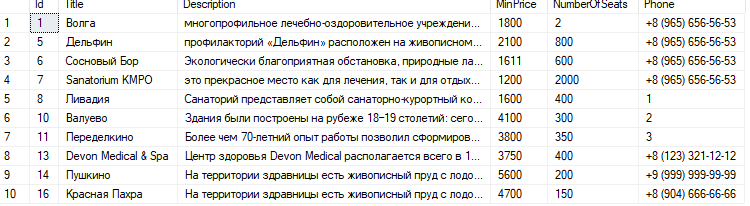


Рис.11 Таблица Sanatoriy

Таблица “Role”(Роль)

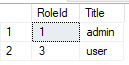


Рис.12 Таблица Role

Таблица “Rewiews”(Отзывы)

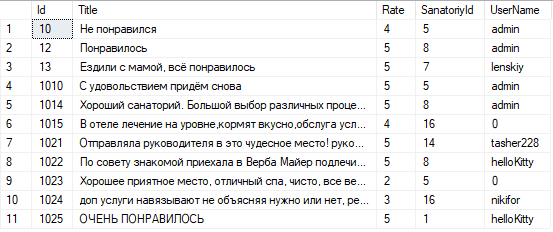


Рис.13 Таблица Rewiews

Таблица “Photo”(Фотографии)

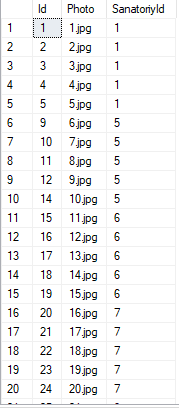


Рис.14 Таблица Photo

Таблица “Category”(Категории)

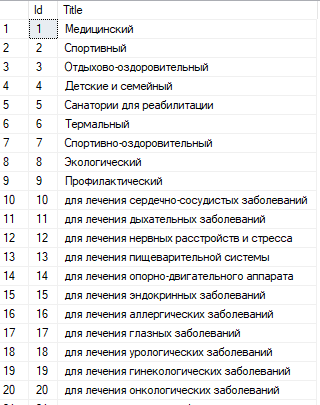
****

Рис.15 Таблица Category

Таблица “Booking”(Бронирование)

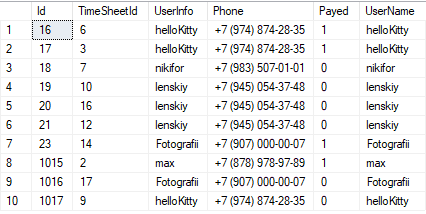
****

Рис.15 Таблица Booking

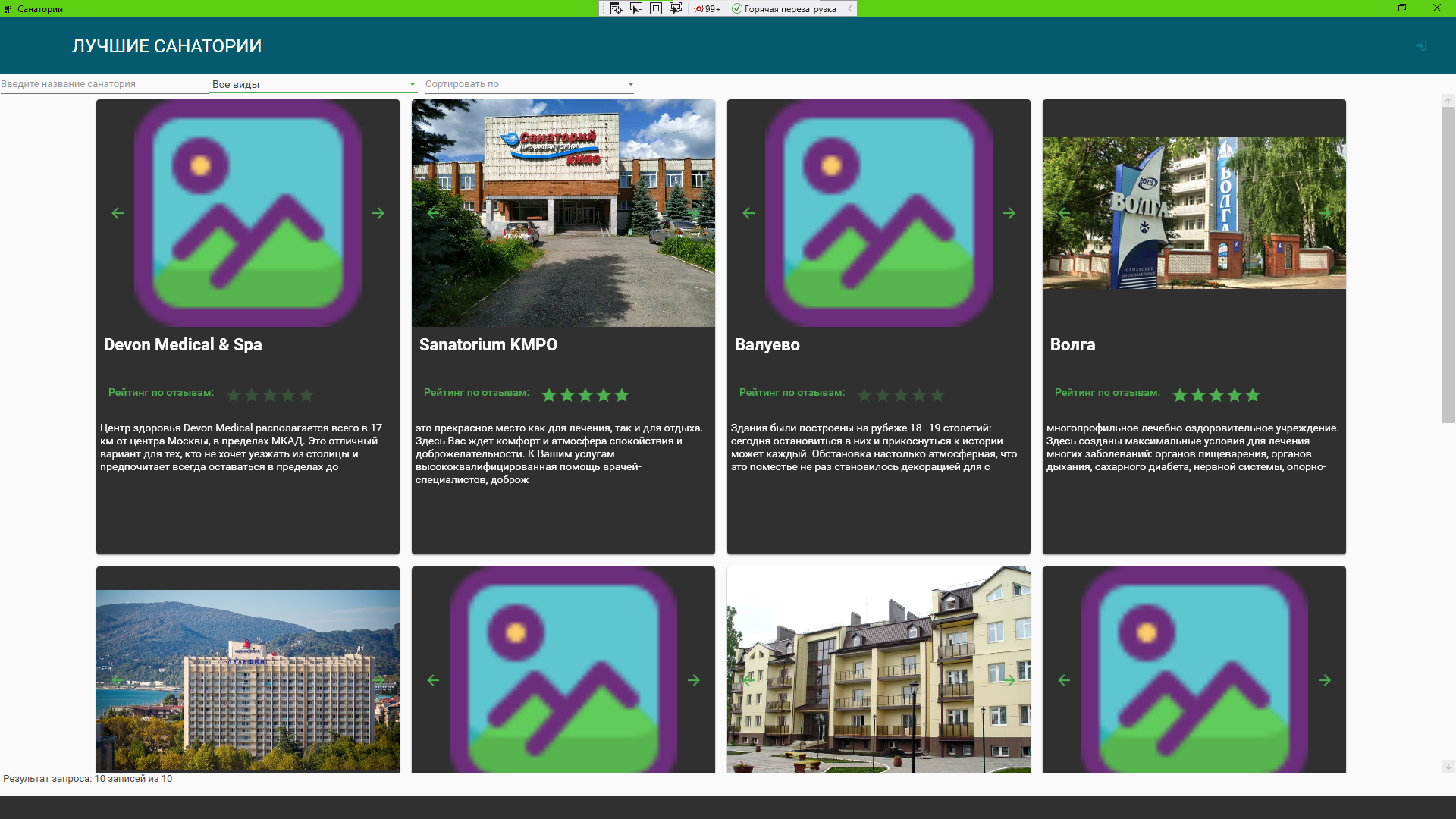
**3.6 Разработка интерфейса ПС**

В приложении предусмотрена авторизация, так как оно многопользовательское. Для запуска приложения достаточно щелкнуть дважды левой кнопкой мыши по ярлыку.

После старта приложения пользователь приступит к работе с главной формой программы, на которой и будут отображаться все события в базе данных. «SanatoriyApp» обладает удобным интерфейсом, так что даже неопытному в обращение с пользовательскими компьютерными программами пользователю не составит труда быстро привыкнуть к работе с данным программным средством.

На рабочем экране расположены несколько кнопок для комфортной работы с приложением.

**Форма Каталог**. Основная форма приложения, в которой в виде плитки отображаются санатории. Компоненты формы: ListView – для отображения мероприятий, Combobox – выпадающие списки, TextBox и Button. (Рис.16)



ListView

TextBox

Image

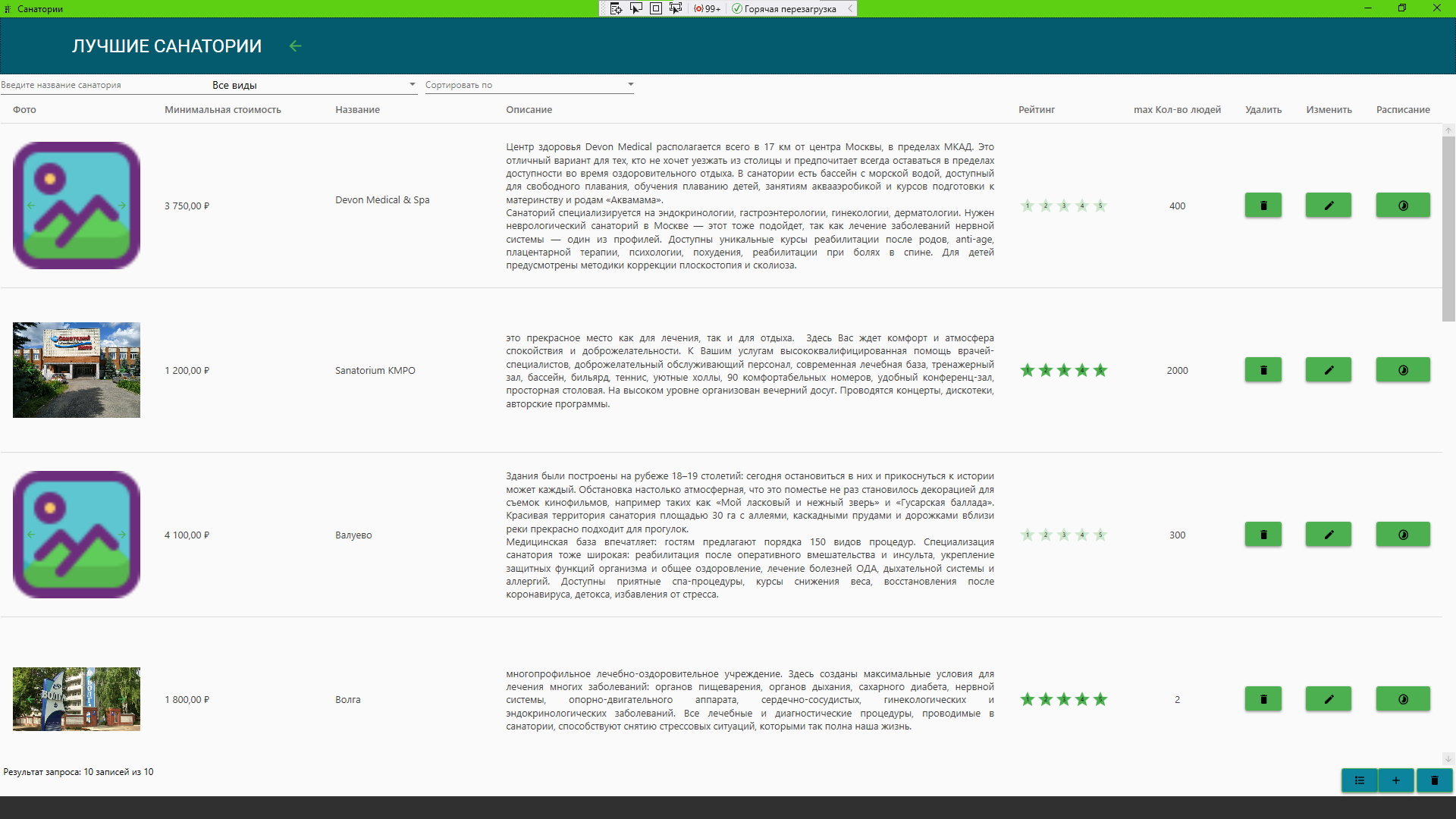
RatingBar

Button

ComboBox

Рис.16 Каталог

**Страница Санаториев**. Компоненты Страницы: DataGrid – для отображения списка мероприятий и кнопки Button для перехода на дополнительные страницы. (Рис.17)



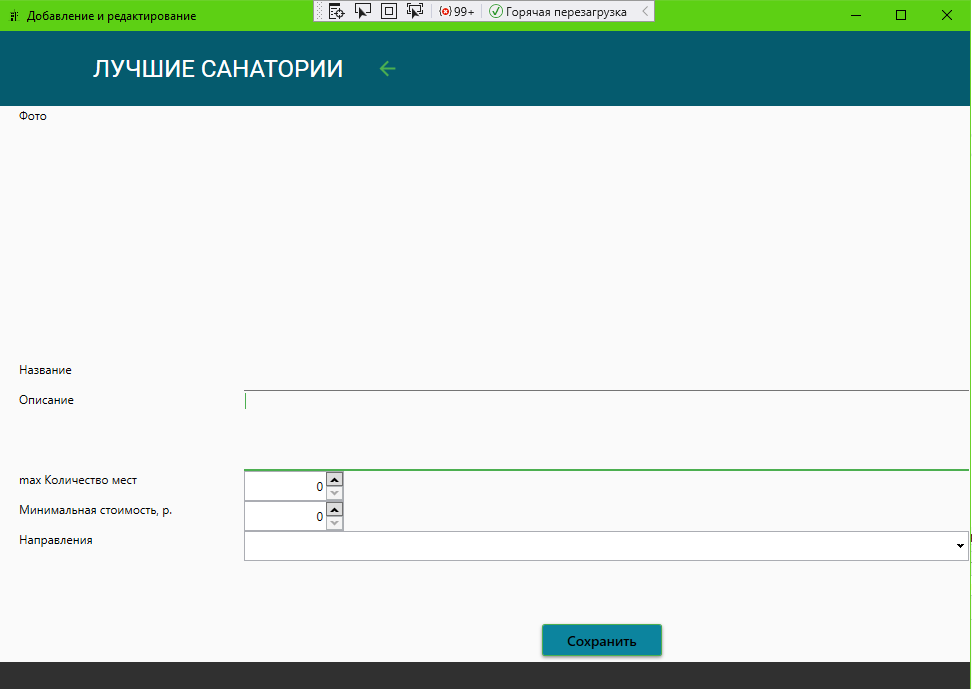
Button

Button

DataGrid

Рис.17 Санатории

**Страница Добавление и редактирование**. Интерфейс данной формы состоит из компонентов: TextBox – поля ввода, ComboBox – выпадающие списки, Image – дляотображения изображения и Button. Также есть возможность открытия дополнительных окон. (Рис.18)



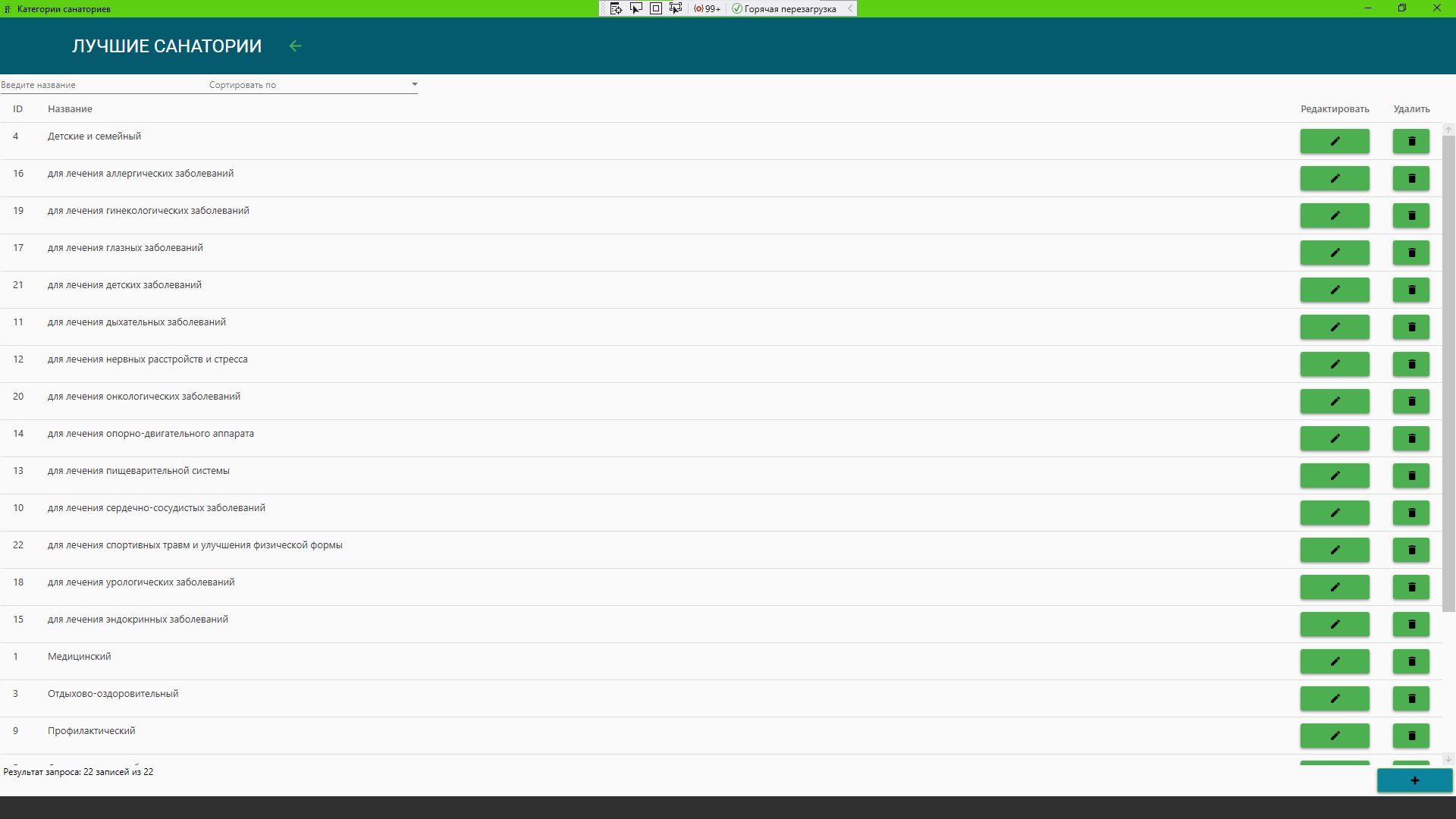
TextBox

Button

CheckComboBox

IntegerUpDown

Рис.18 Страница Добавление

**Страница Категории**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок.(Рис.19) ****

Кнопка Назад

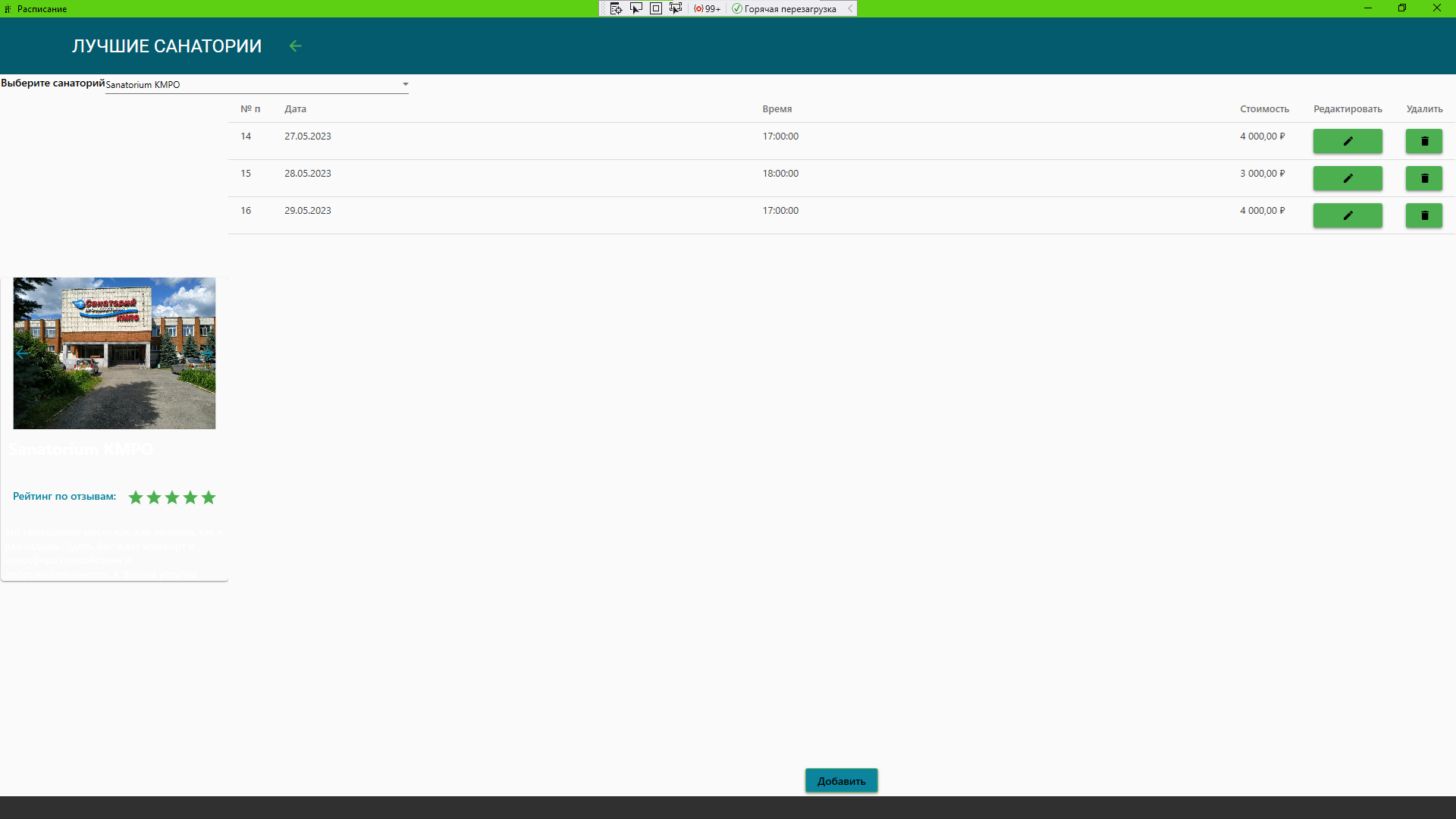
Кнопки Изменить и Удалить

DataGrid

Кнопка Добавить категорию

Рис.19 Страница Категории

**Страница Забронированных санаториев с датой и стоимостью**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок.(Рис.20)



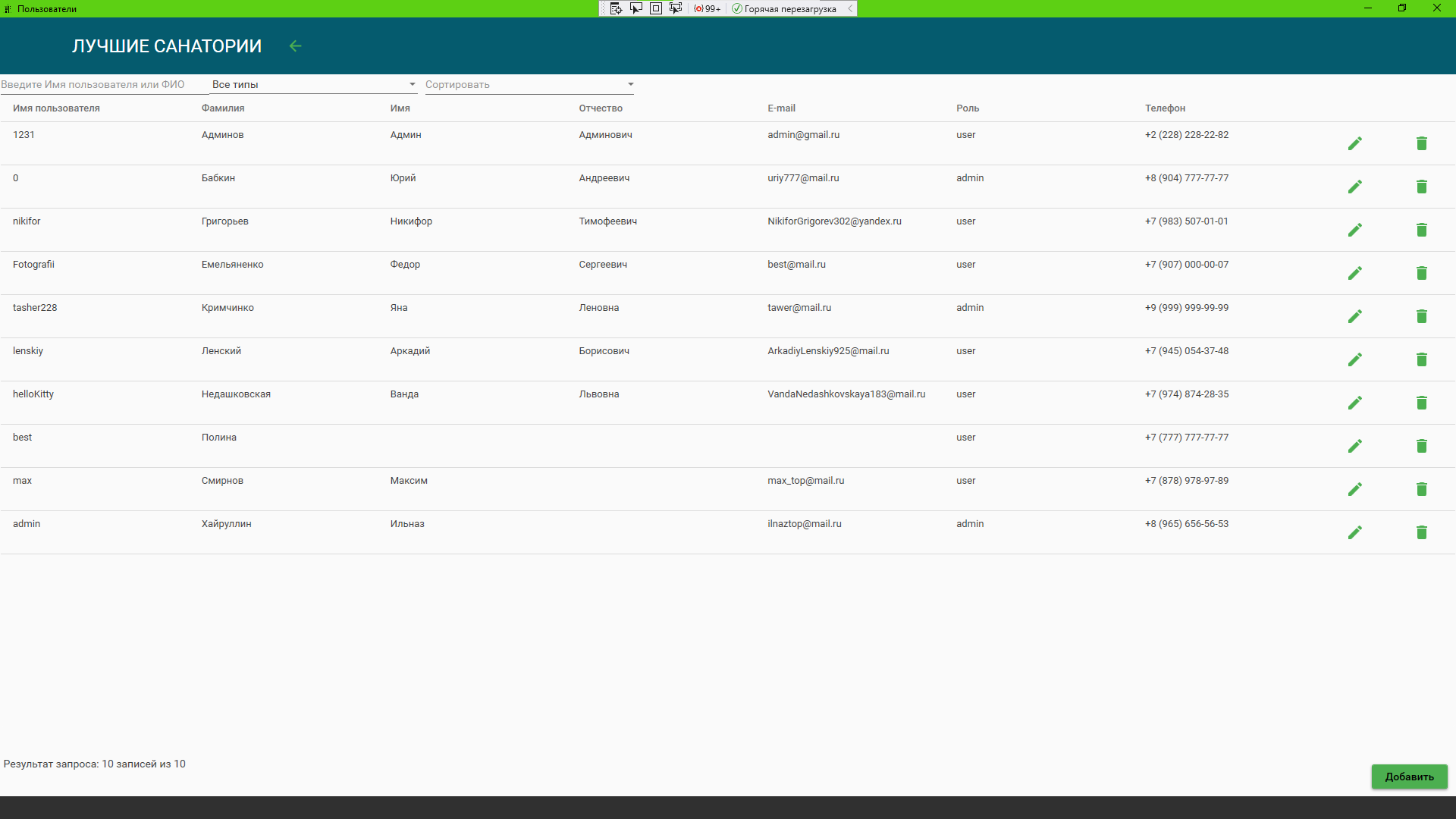
Кнопка Назад

Кнопки Добавить и Удалить и редактировать

DataGrid

Рис.20 Страница Забронированных мест в санатории

**Страница Пользователй**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок.(Рис.21)

****

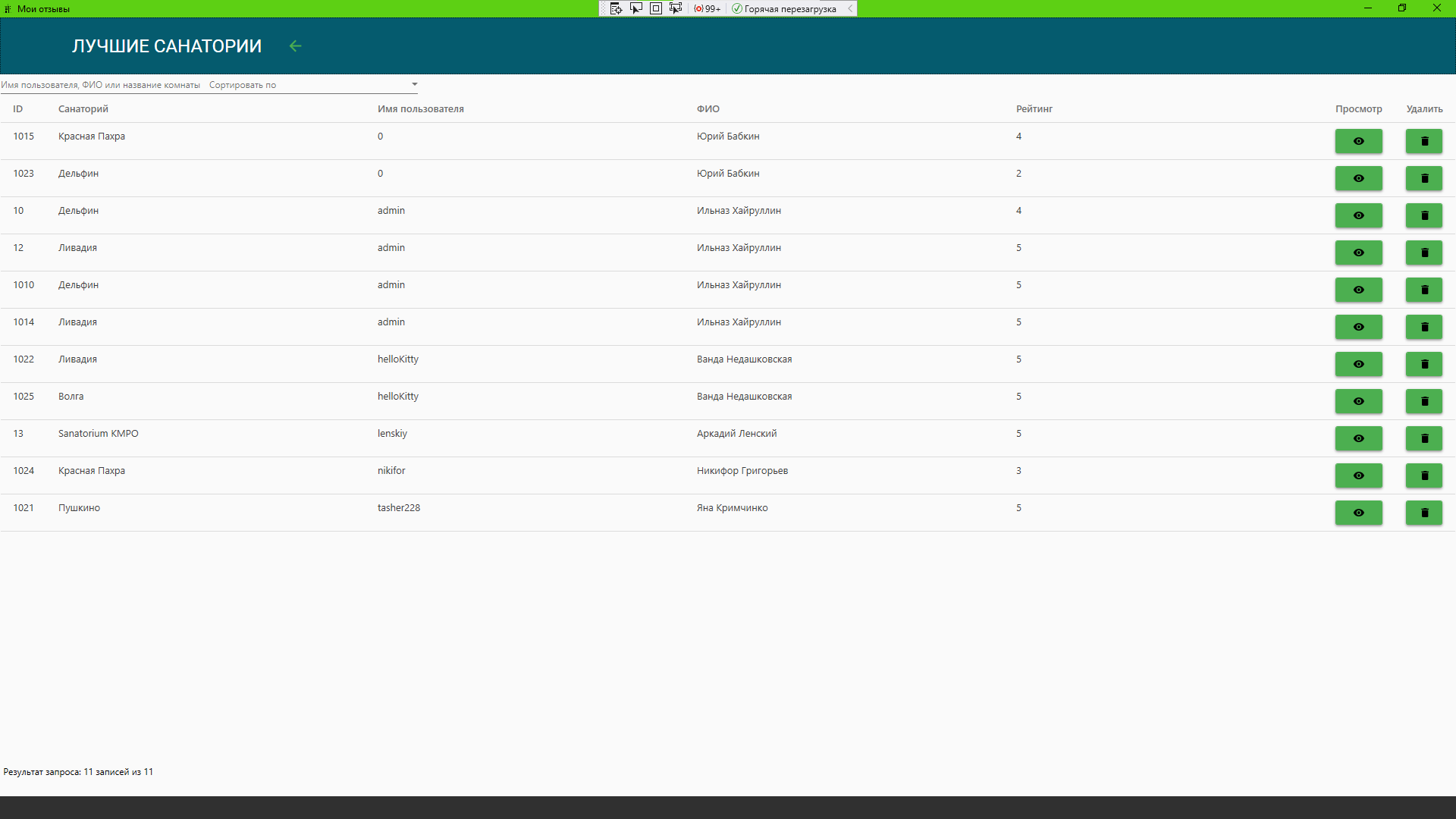
Кнопка Назад

Кнопки Изменить и Удалить

DataGrid

Кнопка Добавить Пользователя

Рис.21 Страница Пользователей

**Страница Отзывов**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок.(Рис.22) ****

Кнопка Назад

Кнопки Просмотреть и Удалить

DataGrid

Рис.22 Страница Отзывов

**3.7 Описание структуры выходной информации**

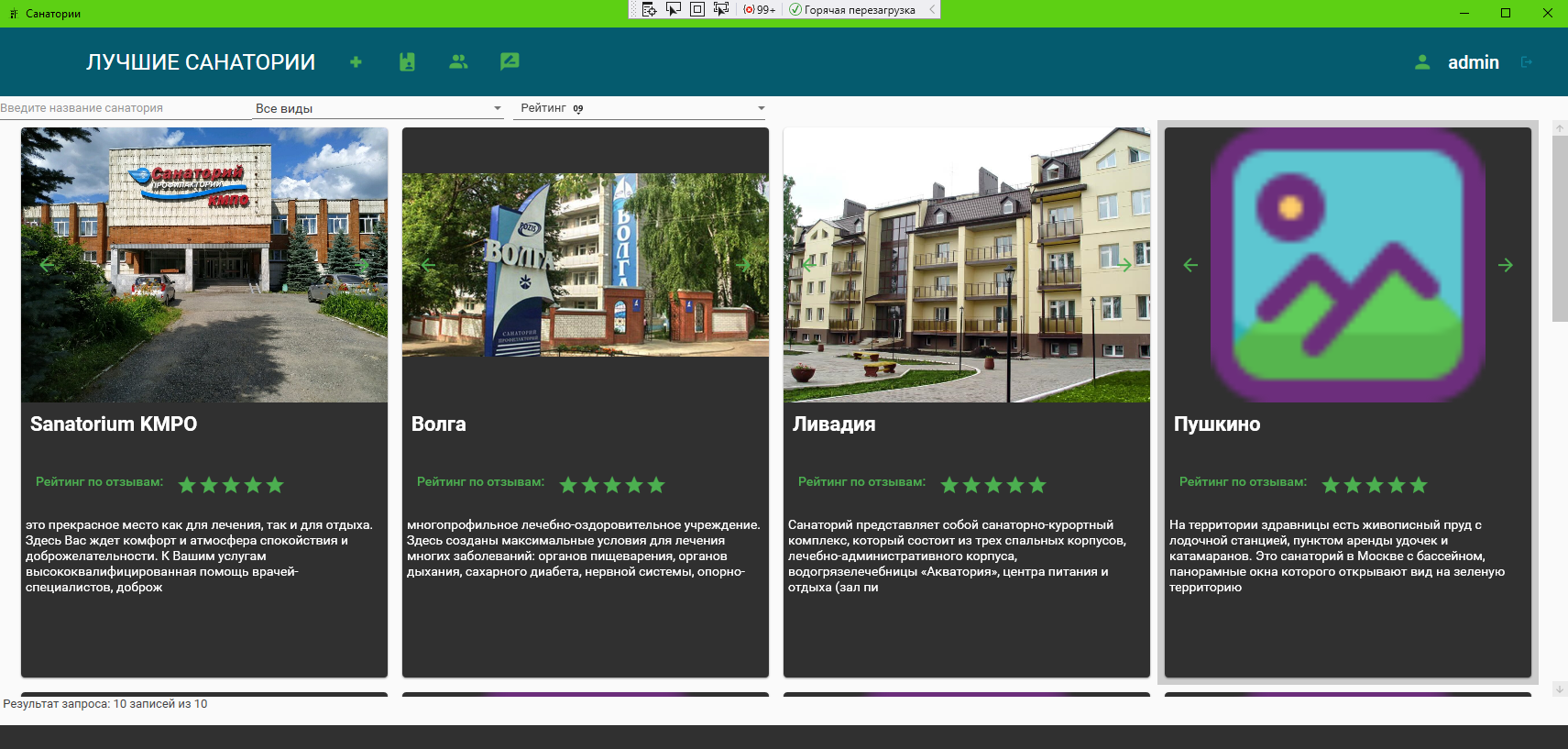
Выходными данными являются списки санаториевв отфильтрованные по разным критериям. 

Рис.23 Списки санаториев

**4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**4.1 Назначение программного средства**

Назначением программного средства является бронирование санаториев.

**4.2 Условия выполнения программного средства**

Условия, при соблюдении которых возможно выполнение работы с программным продуктом:

-Компьютер с установленной на нём «SanatoriyApp».

Подготовительные действия:

-На иконке «SanatoriyApp» рабочего стола произвести двойной щелчок левой кнопкой мыши.

**4.3 Эксплуатация программного средства**

Этот раздел содержит обзор приложений «SanatoriyApp» и ее основных параметров.

***Активация продукта***

Активировать копию программы SanatoriyApp можно сразу после её установки, когда появится окно подтверждения подлинности Вашей версии. Ключ активации расположен внутри упаковки программного продукта. Для успешного выполнения активации ключ активации необходимо вводить именно в том виде, в котором он предоставлен.

Вводные сведения об интерфейсе пользователя

После старта приложения откроется главное окно приложения – Каталог мероприятий в плиточном интерфейсе. На форме есть кнопка для перехода на страницу санаториев. Есть поля для поиска санатория по названию и выпадающие списки для фильтрации. (Рис.17)



Рис.24 Окно Санаторий

Кликнув по кнопке «Выход» программа потребует подтверждение. (Рис.25)

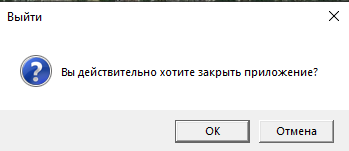


Рис.25 Сообщение программы

Ниже описаны пункты меню управления и информации.

«Страница Санатории»: эта форма предназначена для отображения списка санаториев, а также для перехода на дополнительные страницы. (Рис.26)

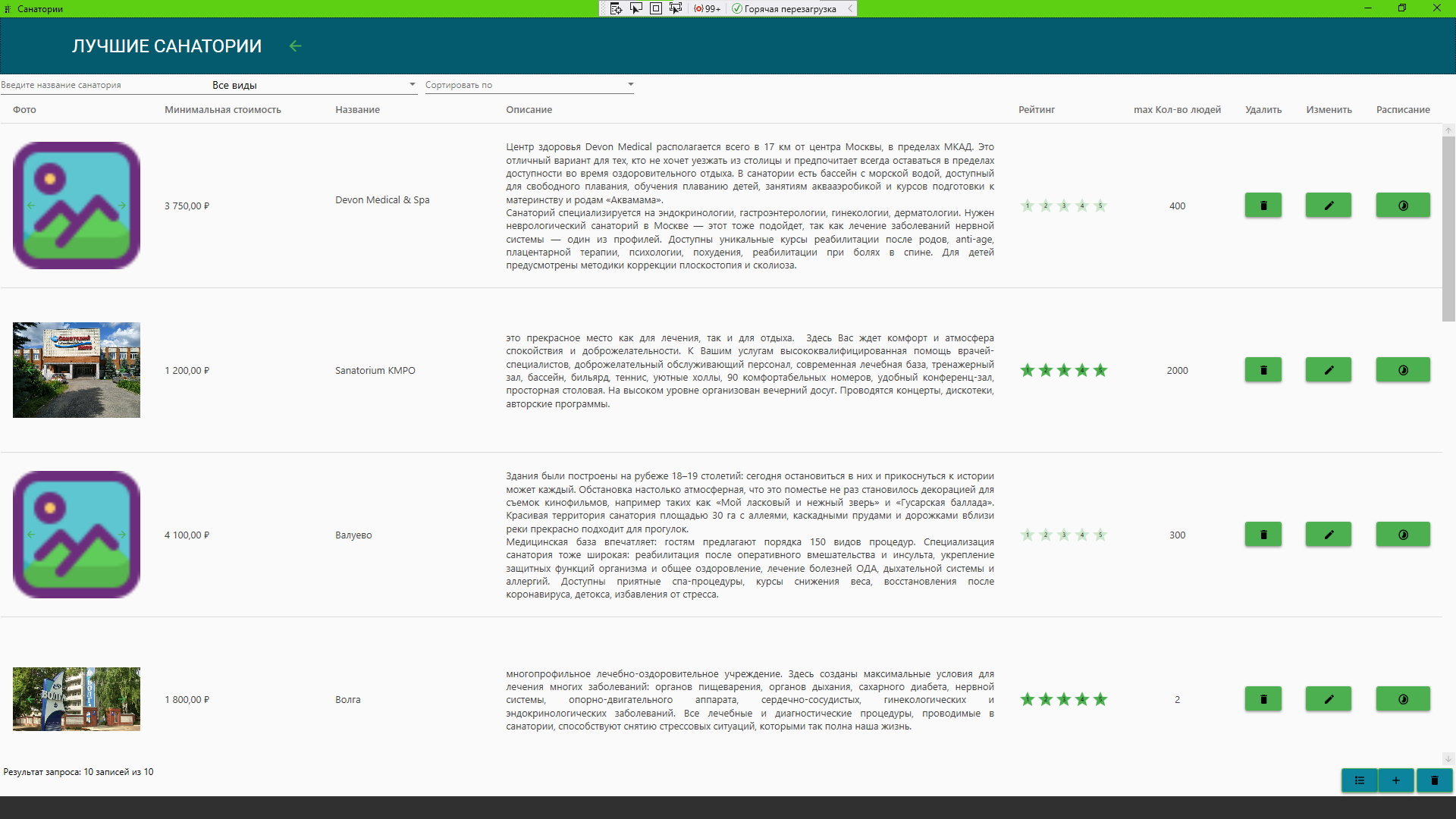


Рис.26 Страница Санатории

«Страница Добавление и и редактирования»: эта форма предназначена для редактирования и добавление данных в таблице Санатории. (Рис.26)

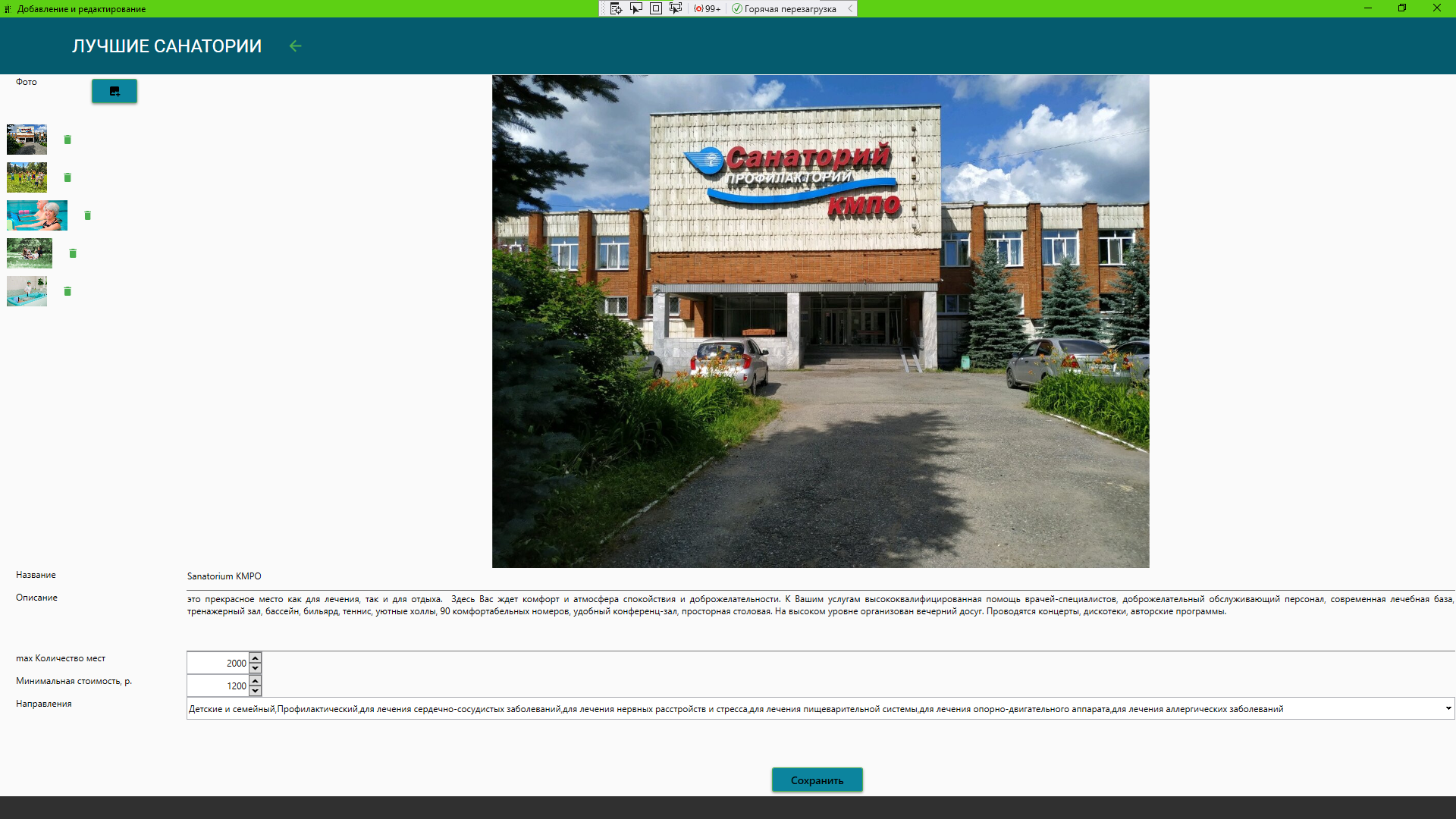


Рис.26 Страница Добавление и редактирование

«Категория»: эта страница предназначена для ввода, удаления и редактирования данных в таблице Категория. (Рис.27)

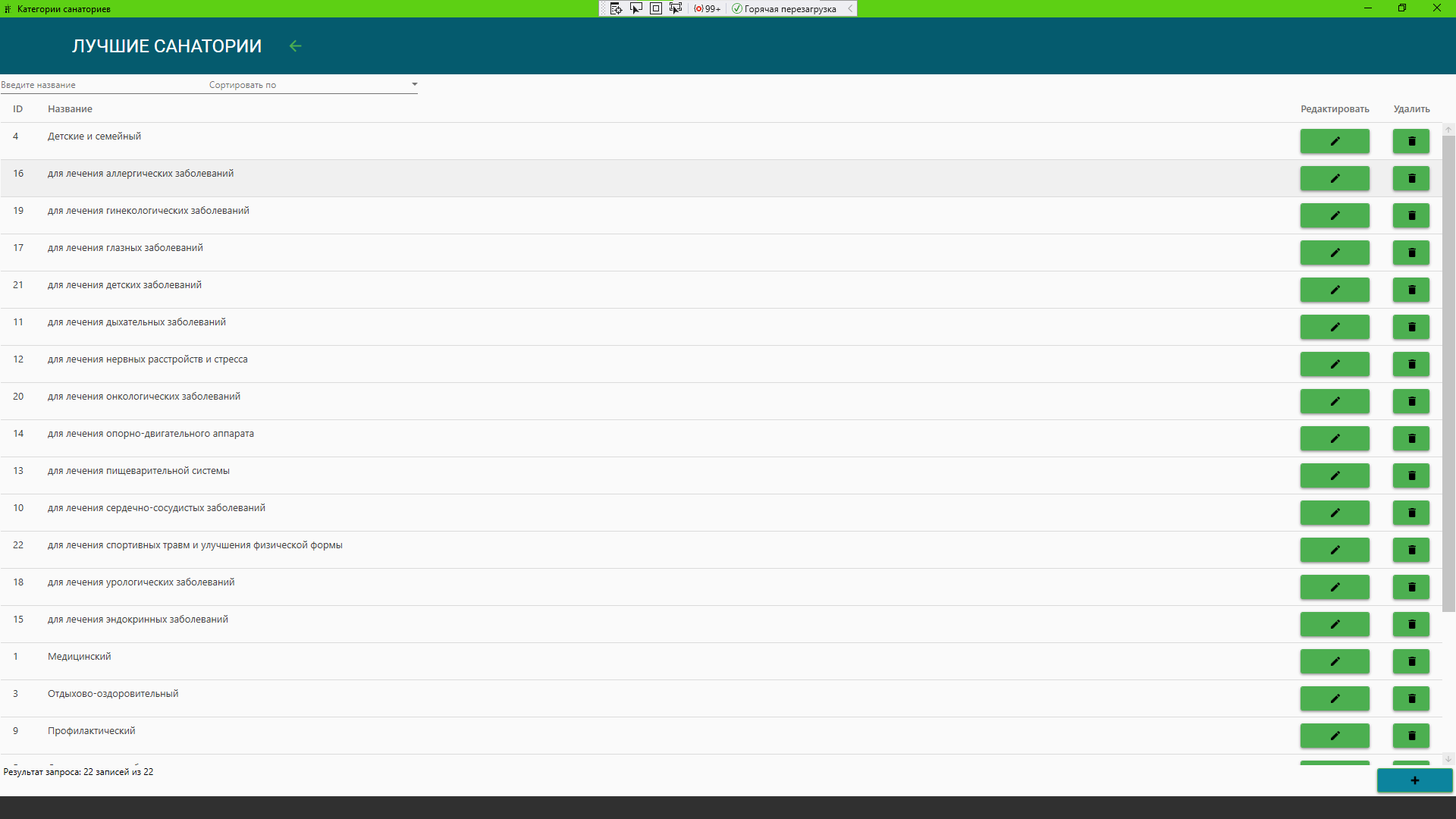


Рис.27 Страница Категория

Работа с SanatoriyApp

После запуска программы Вы перейдете на главную форму программы. В формах «Мероприятия», «Категории», «Статусы», «Добавление и удаление» заполняем все необходимые для работы пункты.

После этого кликаем по кнопке «Добавить запись».

Если Вы желаете удалить любую введенную ранее информацию, Вам необходимо кликнуть по нужной кнопке.

Для того чтобы изменить запись, введите в редактируемые поля данные и нажмите кнопку сохранить. Данные формы редактированию и управлению со стороны пользователя не подлежат.

После того, как Вы закончите сеанс работы с программой, если Вы хотите выйти из SanatoriyApp кликните на соответствующей кнопке «Выход».

Основные кнопки SanatoriyApp

Кнопки управления «Добавить запись», «Удалить запись», «Сохранить» расположены на каждой конкретной форме в базе данных.

Действия этих кнопок приведены в таблице 1:

Таблица-1 Действие кнопок

|  |  |
| --- | --- |
| Добавить запись | Прежде чем добавить запись в таблицу необходимо нажать на данную кнопку. Поля формы будут очищены, пользователю требуется ввести необходимые данные. Если пользователь нажмет кнопку «Сохранить данные», то введенная информация добавится в БД. Если нажать кнопку «Отменить», то введенная информация пользователем удалится. |
| Удалить запись/отменить  , , | Кнопка «Удалить запись» позволяет отменить действия при вводе, либо удаляет конкретную запись. |
| Сохранить данные/обновить запись  , , | Кнопка «Сохранить данные» добавляет запись введенную пользователем в БД, либо обновляет существующую. |
| Редактировать | Открывает форму в режиме редактирования информации |
|  | «Поиск» позволяет найти нужную информацию. |

**4.4 Сообщения пользователю**

Для уверенной и комфортной работы пользователю важно понимать поведение программы, а программе, в свою очередь, поведение пользователя. Для информирования и уточнения намерений пользователей в программах используются сообщения.

При попытке удалить запись, пользователю выводится сообщение с запросом на подтверждение или отмену действия. (Рис.28)

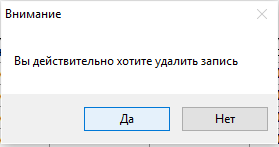


Рис.28 Сообщение Подтверждение удаления

После добавления новой записи, пользователю выводится сообщение. (Рис.29)

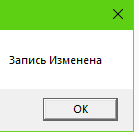


Рис.29 Сообщение Запись добавлена

После обновления записи, пользователю выводится сообщение. (Рис.30)

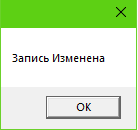


Рис.30 Сообщение Запись обновлена

При попытке удалить связанные записи, программа выводит пользователю сообщение о невозможности действия. (Рис.31)

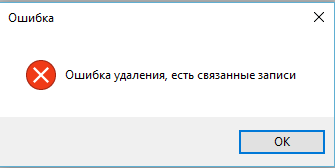


Рис.31 Сообщение Ошибка удаления

**5. РАЗРАБОТКА ТЕСТОВОГО НАБОРА**

**5.1 Обоснование необходимого количества тестов**

Для разработанного алгоритма был построен граф.

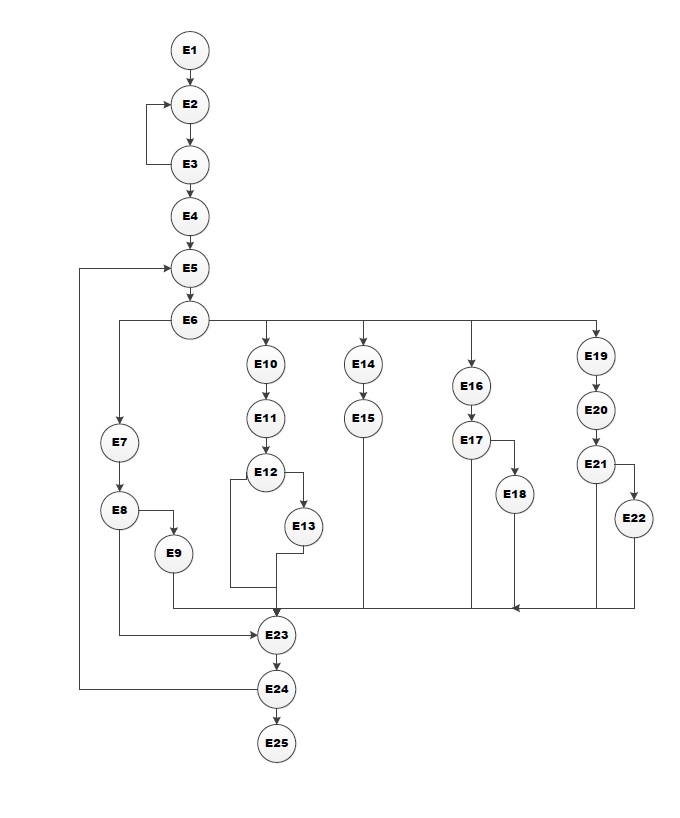
****

Рис.27 Потоковый граф

Необходимое количество тестов определяется по формуле V(G) = E-N+2, где E – это число ребер, а N–это число вершин данного графа. Согласно представленному графу E=34, N=25. Далее по формуле рассчитаем необходимое количество тестовых сценариев (тест-кейсов).

V(G)=34-25+2=11.

Для данного программного средства необходимо разработать не менее 11 тестовых сценариев.

Чтобы определить количество тестовых прогонов для программного продукта SanatoriyApp необходимо выделить требования. Из требований к приложению выделим поддержку 2-х операционных систем с одним основным языком локализации и выполнения 6-ти основных функций. Кроме того, приложение позволяет выполнять 5 функций по взаимодействию с системой (запуск приложения, выход из приложения, сохранение результатов в файл, работа с буфером и т.п.).

Таким образом, полное покрытие требований задаёт набор из 2\*1\*(6+5) =22 тестовых прогонов.

**5.2 Описание тестовых пакетов**

Таблица 2 – Тестирование корректного добавления информации в Категории

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Категория | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категории | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Категория = Лечебный | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категория открыта | | Отображается форма Категория |  |

Таблица 3 – Тестирование некорректного добавления информации в таблицу Категория

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Категория | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категория | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Категория = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод messageBox, то что поле пустое | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категория открыта | | Отображается форма Категория |  |

Таблица 4 – Тестирование корректного редактирования информации в таблице Категория

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Категория | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категория | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Категория = Оздоровительный=>Психологический | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категория  открыта | | Отображается форма Категория |  |

Таблица 5 – Тестирование некорректного редактирования информации в таблице Категория

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Категория | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категория | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Категория = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод messageBox, то что поле пустое | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категория  открыта | | Отображается форма Категория |  |

Таблица 6 – Тестирование корректного добавления информации в таблицу Санаторий

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Санаторий | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Санаторий | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = Пушкин дом  Описание = Этот дом когда-то принадлежал Пушкину  Max Количество мест = 52  Минимальная стоимость = 52.000  Направления: Медецинский, Изображение = загружено | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Санаторий открыта | | Отображается форма Санаторий |  |

Таблица 7 – Тестирование некорректного добавления информации в таблицу Санаторий

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу  Санаторий | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Мероприятие | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название =  Описание =  Max Количество мест =  Минимальная стоимость =  Направления =  Изображение = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод messageBox, то что поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Санаторий  открыта | | Отображается форма Санаторий |  |

Таблица 8 – Тестирование корректного редактирования информации в таблице

Санаторий

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Санаторий | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Санаторий | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название = Пушкин дом  Описание = Этот дом когда-то принадлежал Пушкину  Max Количество мест = 520  Минимальная стоимость = 52.000  Направления: Медецинский, Изображение = загружено | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Санаторий открыта | | Отображается форма Санаторий |  |

Таблица 9 – Тестирование некорректного редактирования информации в таблице Санаторий

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Санаторий | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Санаторий | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название =  Описание =  Max Количество мест =  Минимальная стоимость =  Направления =  Изображение = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод messageBox, то что поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Санаторий открыта | | Отображается форма Санаторий |  |

Таблица 10 – Тестирование корректного добавления информации в Пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Пользователи | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Пользователи | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Имя = Иван  Фамилия = Иванов  Отчество = Иванович  Тип пользователя = user  Номер Телефона = 8 999 999 99 99  E-mail = [Ivan@mail.ru](mailto:Ivan@mail.ru)  Пароль = 1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Пользователи открыта | | Отображается форма Пользователи |  |

Таблица 11 – Тестирование некорректного добавления информации в таблицу Пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Пользователи | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категория | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Нажмите кнопку добавить  Заполните форму  Имя =  Фамилия =  Отчество =  Тип пользователя =  Номер Телефона =  E-mail =  Пароль = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод messageBox, то что поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Пользователи открыта | | Отображается форма Пользователи |  |

Таблица 12 – Тестирование корректного редактирования информации в таблице Пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Пользователи | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Пользователи | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку изменить  Заполните форму  Имя = Иван2  Фамилия = Иванов2  Отчество = Иванович2  Тип пользователя = admin  Номер Телефона = 8 888 888 88 88  E-mail = [tost@mail.ru](mailto:tost@mail.ru)  Пароль = 0 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Пользователи  открыта | | Отображается форма Пользователи |  |

Таблица 13 – Тестирование некорректного редактирования информации в таблице Категория

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Категория | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категория | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Категория = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод messageBox, то что поле пустое | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категория  открыта | | Отображается форма Категория |  |

Таблица 14 – Тестирование корректного добавления информации в МоиОтзывы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу МоиОтзывы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму МоиОтзывы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Отзыв: Не понравилось  Количество звёзд: 1 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма МоиОтзывы открыта | | Отображается форма МоиОтзывы |  |

Таблица 15 – Тестирование некорректного добавления информации в таблицу МоиОтзывы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу МоиОтзывы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму МоиОтзывы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Отзыв:  Количество звёзд: | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод об ошибке в виде messageBox, о том, что поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма МоиОтзывы открыта | | Отображается форма МоиОтзывы |  |

Таблица 16 – Тестирование корректного редактирования информации в таблице МоиОтзывы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице МоиОтзывы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму МоиОтзывы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Отзыв: понравилось очень  Количество звёзд: 5 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма МоиОтзывы открыта | | Отображается форма МоиОтзывы |  |

Таблица 17 – Тестирование некорректного редактирования информации в таблице МоиОтзывы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице МоиОтзывы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму МоиОтзывы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Отзыв:  Количество звёзд: | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Вывод об ошибке в виде messageBox, о том, что поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма МоиОтзывы открыта | | Отображается форма МоиОтзывы |  |

**6. ЗАКЛЮЧЕНИЕ**

**6.1 Оценка качества программного средства с помощью метрик**

Оценка качества программного средства проводится исходя из стандарта оценки качества ISО 9126.

На компьютере следующей конфигурации:

Установленная операционная система – Windows 7 Ultimate 32бит.

Процессор: Intel Pentium G2030 3.0 GHz.

ОЗУ: 2 Gb

Таблица 18 – Требования к количественным характеристикам качества программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Надежность** |  | | |
| *Завершенность:*  *-* наработка на отказ при отсутствии рестарта; | Часы | 10 | 10 |
| *Устойчивость:*  - относительные ресурсы на обеспечение надежности и рестарта. | % | 0,5 | 0,5 |
| *Восстанавливаемость:*  -длительность восстановления. | Минуты | 2 | 2 |
| *Доступность-готовность:*  -относительное время работоспособного функционирования. | Вероятность | 0,998 | 0,998 |
| **Эффективность** |  | | |
| *Временная эффективность:*  *-*время отклика - получение результатов на типовое задание;  - пропускная способность число типовых заданий, исполняемых в единицу времени. | Секунды  Число в минуту | 3  60 | 3  60 |
| *Используемость ресурсов:*  *-* относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства. |  | 0,001 | 0,001 |

Таблица 19 – Требования к качественным характеристикам программного средства.

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Практичность** |  | | |
| *Простота использования:*  **-**среднее время ввода заданий;  -среднее время отклика на задание. | Секунды  Секунды | 10  3 | 10  3 |
| *Изучаемость :*  - трудоемкость изучения применения ПС;  -продолжительность изучения;  -объем эксплуатационной документации; | Чел.-часы  Часы  Страницы | 5  2  15 | 5  2  15 |
| **Сопровождаемость** |  | | |
| *Изменяемость:*  -трудоемкость подготовки изменений;  -длительность подготовки изменений. | Чел.-часы  Часы | 5  5 | 5  5 |
| *Тестируемость:*  *-трудоемкость тестирования изменений;*  *- длительность тестирования изменений.* | Чел.-часы  Часы | 3  3 | 3  3 |
| **Мобильность** |  | | |
| *Адаптируемость:*  - трудоемкость адаптации;  -длительность адаптации. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Простота установки:*  -трудоемкость инсталляции;  -длительность инсталляции. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Замещаемость:*  -трудоемкость замены компонентов;  - длительность замены компонентов. | Чел.-часы  Часы | 0,2  0,2 | 0,2  0,2 |

Таблица 20 – Сложность

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип объекта | Количество | Вес | | | Итого |
|  |  | Простой | Средний | Сложный |  |
| Экран | 9 | x1 | x2 | **x3** | =27 |
| Отчет | 2 | x2 | **x5** | x8 | =10 |

Таблица 21 – Сложность экрана

|  |  |  |  |
| --- | --- | --- | --- |
| Экраны | Количество Категорииских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | **Всего <8** | Всего >8 |
|  | (<2 срв, <3 клт) | **(2-3 срв, 3-5 клт)** | (>3 срв, >5 клт) |
| <3 | Простой | Простой | Средний |
| 3-7 | Простой | Средний | Сложный |
| **>8** | Средний | **Сложный** | Сложный |

Таблица 22 – Сложность отчета

|  |  |  |  |
| --- | --- | --- | --- |
| Отчеты | Количество Категорииских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | Всего <8 | Всего >8 |
|  | (<2 срв, <3 клт) | **(2-3 срв, 3-5 клт)** | (>3 срв, >5 клт) |
| 0 или 1 | Простой | Простой | Средний |
| **2 или 3** | Простой | **Средний** | Сложный |
| >4 | Средний | Сложный | Сложный |

Производительность = 4 тысяч строк кода/2 месяца= 2 тысяч строк кода/месяц

Качество = 2 ошибки /4 тысяч строк кода=0,5 ошибок/тысяч строк кода

**7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Г. Шилдт C#: Учебный курс. - Спб.: Питер, 2022. - 512с

2. Либерти, Д. Программирование на Си Шарп. - М.: Символ-плюс, 2022. - 684 с.

3. Ватсон К. Си Шарп/К.Ватсон.-М.:Лори, 2022.-862 с

4. Балена Ф., Димауро Д. Современная практика программирования на Visual Basic и Visual C#.-М.:Русская редакция,2022,604с.

5. Енин А., Енин Н. Локальная СУБД своими руками. Учимся на примерах.-М.:СОЛОН-ПРЕСС,2022,464с.,СД

6. Лабор В.В. Си Шарп: СоЗдание приложений для Windows/-Мн.: Харвест,2022-384с.

7. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2022-512с.

8. ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению

9. ГОСТ 19.102-77.ЕСПД. Стадии разработки

10. ГОСТ 19.505-79.ЕСПД. Руководство оператора. Требования к содержанию и оформлению

11. ГОСТ 19.002-80.ЕСПД. Схемы алгоритмов и программ. Правила выполнения

12. ГОСТ 19.404-79.ЕСПД. Пояснительная записка. Требования к содержанию и оформлению

13. Владимир Биллинг. Основы программирования на С#. 2016 г.

[Электронный ресурс http://mexalib.com/view/42671]

14. Никита Культин. Основы программирования в MicrosoftVisualC# 2017. 2017 г.

[Электронный ресурс http://www.litres.ru/nikita-kultin/osnovy-programmirovaniya-v-microsoft-visual-c-2022-2]

15. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2017.

[Электронный ресурс http://www.twirpx.com/file/27852]

16. Йен Гриффитс. Программирование на С#. 2022 г.

[Электронный ресурс http://rutracker.org/forum/viewtopic.php?t=4772853]

**Приложение А**

Разработка интегрированного программного модуля «Санаторий»

Текст кода программного средства

460.ЗМК.14-01 12 01

**Текст кода программного средства**

**Словарь ресурсов**

<Application x:Class="SanatoriyApp.App"

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

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

xmlns:local="clr-namespace:SanatoriyApp"

StartupUri="Windows/MainWindow.xaml"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<materialDesign:BundledTheme BaseTheme="Light" PrimaryColor="Green" SecondaryColor="Lime" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace SanatoriyApp.Models

{

public class Manager

{

public static Frame MainFrame { get; set; }

public static User CurrentUser { get; set; }

}

}

Класс Booking.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

namespace SanatoriyApp.Models

{

public partial class Booking

{

public Visibility GetVisibility

{

get

{

if (Payed)

return Visibility.Collapsed;

return Visibility.Visible;

}

}

}

}

Класс Photo.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SanatoriyApp.Models

{

public partial class Photo

{

public string GetPhoto

{

get

{

if (Photo1 is null)

return Directory.GetCurrentDirectory() + @"\Images\picture.png";

return Directory.GetCurrentDirectory() + @"\Images\" + Photo1.Trim();

}

}

}

}

Класс Sanatoriy.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

namespace SanatoriyApp.Models

{

public partial class Sanatoriy

{

List<Photo> photos = null;

int index = 0;

public string GetCurrentPhoto

{

get

{

if (photos == null)

photos = Photos.ToList();

if (photos.Count == 0)

return Directory.GetCurrentDirectory() + @"\Images\picture.png";

return photos[index].GetPhoto;

}

}

public int ReloadPhotos

{

set

{

if (photos != null)

{

photos = null;

photos = Photos.ToList();

}

}

}

public int SetPrevPhoto

{

set

{

if (index == 0)

return;

index--;

}

}

public int SetNextPhoto

{

set

{

if (index == photos.Count - 1)

return;

index++;

}

}

public string GetInfo

{

get

{

if (Description.Length >= 200)

return Description.Substring(0, 200);

return Description;

}

}

public double GetRate

{

get

{

int rate = 0;

foreach (Rewiew rewiew in Rewiews)

{

rate += rewiew.Rate;

}

if (rate == 0)

return 0;

return (rate / Rewiews.Count);

}

}

public Visibility GetVisibility

{

get

{

if (Manager.CurrentUser == null)

return Visibility.Collapsed;

if (Manager.CurrentUser.RoleId < 3)

return Visibility.Collapsed;

return Visibility.Visible;

}

}

}

}

Класс SanatoriyBDEntities.cs

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SanatoriyApp.Models

{

public partial class SanatoriyBDEntities : DbContext

{

private static SanatoriyBDEntities \_context;

public static SanatoriyBDEntities GetContext()

{

if (\_context == null)

{

\_context = new SanatoriyBDEntities();

}

return \_context;

}

}

}

Класс TimeSheet.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SanatoriyApp.Models

{

public partial class TimeSheet

{

public string GetFullDateTime

{

get

{

return $"{Date.Day:00} {Date.ToString("MMMM")} {Time.Hours:00}:{Time.Minutes:00}";

}

}

}

}

Класс User.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

namespace SanatoriyApp.Models

{

public partial class User

{

public string GetFIO

{

get

{

return $"{FirstName} {LastName}";

}

}

public Visibility GetVisibility

{

get

{

if (RoleId == 3)

return Visibility.Visible;

return Visibility.Hidden;

}

}

}

}

Контекст DataModel.Context.cs

//------------------------------------------------------------------------------

// <auto-generated>

// Этот код создан по шаблону.

//

// Изменения, вносимые в этот файл вручную, могут привести к непредвиденной работе приложения.

// Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном создании кода.

// </auto-generated>

//------------------------------------------------------------------------------

namespace SanatoriyApp.Models

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class SanatoriyBDEntities : DbContext

{

public SanatoriyBDEntities()

: base("name=SanatoriyBDEntities")

{

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Booking> Bookings { get; set; }

public virtual DbSet<Category> Categories { get; set; }

public virtual DbSet<Photo> Photos { get; set; }

public virtual DbSet<Rewiew> Rewiews { get; set; }

public virtual DbSet<Role> Roles { get; set; }

public virtual DbSet<Sanatoriy> Sanatoriys { get; set; }

public virtual DbSet<SanatoriyCategory> SanatoriyCategories { get; set; }

public virtual DbSet<sysdiagram> sysdiagrams { get; set; }

public virtual DbSet<TimeSheet> TimeSheets { get; set; }

public virtual DbSet<User> Users { get; set; }

}

}

Форма Главная

Код интерфейса

<Window x:Class="SanatoriyApp.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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:SanatoriyApp"

mc:Ignorable="d"

Title="{Binding ElementName=MainFrame, Path=Content.Title}"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

Height="450" Width="800"

Closing="WindowClosing" WindowStartupLocation="CenterScreen"

Closed="WindowClosed" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid/>

<materialDesign:ColorZone Padding="16" Background="#FF055B6E" ClipToBounds="False"

Foreground="White" Mode="Custom">

<DockPanel >

<StackPanel Orientation="Horizontal" DockPanel.Dock="Left" HorizontalAlignment="Left">

<Image Source="/Resources/logo.png"

HorizontalAlignment="Left" Height="55" Width="63"/>

<TextBlock Margin="16,0" VerticalAlignment="Center" Text="ЛУЧШИЕ САНАТОРИИ" Style="{StaticResource MaterialDesignHeadline5TextBlock}" />

<Button x:Name="BtnSanatories" Click="BtnSanatories\_Click"

ToolTip="Санатории" Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=Hospital, Size=24}" Visibility="Collapsed"/>

<Button x:Name="BtnBooking" Click="BtnBooking\_Click" Visibility="Collapsed"

ToolTip="Бронирование санаториев" Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=BookAccount, Size=24}"/>

<Button x:Name="BtnUsers" Click="BtnUsers\_Click" ToolTip="Пользователи"

Style="{StaticResource MaterialDesignFlatButton}" Visibility="Collapsed"

Content="{materialDesign:PackIcon Kind=People, Size=24}"

/>

<Button x:Name="BtnMyrewiew" Click="BtnMyrewiew\_Click" Visibility="Collapsed"

ToolTip="Мои отзывы"

Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=MessageDraw, Size=24}"

/>

<Button x:Name="BtnAllRewiew" Click="BtnAllRewiew\_Click" Visibility="Collapsed"

ToolTip="Отзывы"

Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=MessageDraw, Size=24}"

/>

<Button x:Name="BtnMyBooking" Click="BtnMyOrders\_Click" Visibility="Collapsed"

ToolTip="Бронирование"

Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=Book, Size=24}"

/>

<Button x:Name="BtnBack" Click="BtnBackClick" ToolTip="Назад" Visibility="Collapsed" Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowBack, Size=24}"/>

</StackPanel>

<StackPanel DockPanel.Dock="Right" Orientation="Horizontal" HorizontalAlignment="Right">

<Button x:Name="BtnMyAccount" Click="BtnMyAccount\_Click" ToolTip="Мой профиль"

Style="{StaticResource MaterialDesignFlatButton}" Visibility="Collapsed"

Content="{materialDesign:PackIcon Kind=Account, Size=24}"

/>

<TextBlock x:Name="TextBlockUser" VerticalAlignment="Center" Style="{StaticResource MaterialDesignHeadline6TextBlock}"/>

<Button x:Name="BtnEnter" Click="BtnEnter\_Click"

ToolTip="Войти" Style="{StaticResource MaterialDesignFlatButton}"

Width="60" >

<materialDesign:PackIcon x:Name="IconBtnKey" Kind="Login" Padding="0" Foreground="#FF0C849E"/>

</Button>

</StackPanel>

</DockPanel>

</materialDesign:ColorZone>

<Frame x:Name="MainFrame" Grid.Row="1"

NavigationUIVisibility="Hidden"

BorderThickness="1" ContentRendered="MainFrameContentRendered"/>

<!--<StackPanel Grid.Row="0" Orientation="Horizontal" HorizontalAlignment="Right" Margin="0 0 20 0">

<Button x:Name="BtnEditGoods" Click="BtnEditGoodsClick" ToolTip="Тренеры" Visibility="Collapsed">

<materialDesign:PackIcon Kind="AccountMultiplePlus" Padding="0"/>

</Button>

<Button x:Name="BtnOrder" Click="BtnOrder\_Click" ToolTip="Заявки" Visibility="Collapsed">

<materialDesign:PackIcon Kind="OrderBoolAscending" Padding="0"/>

</Button>

<Button x:Name="BtnBuyes" Click="BtnBuyes\_Click" ToolTip="Абонементы" Visibility="Collapsed">

<materialDesign:PackIcon Kind="Cash" Padding="0"/>

</Button>

<Button x:Name="BtnMyAccount" Click="BtnMyAccount\_Click" ToolTip="Мой профиль" Visibility="Collapsed">

<materialDesign:PackIcon Kind="Account" Padding="0"/>

</Button>

<Button x:Name="BtnMyOrders" Click="BtnMyOrders\_Click" ToolTip="Оставить заявку" Visibility="Collapsed">

<materialDesign:PackIcon Kind="StarBorder" Padding="0"/>

</Button>

<Button x:Name="BtnEnter" Click="BtnEnter\_Click" ToolTip="Войти">

<materialDesign:PackIcon x:Name="IconBtnKey" Kind="Login" Padding="0"/>

</Button>

</StackPanel>-->

<Rectangle Grid.Row="2" Fill="{StaticResource MaterialDesignDarkBackground}"/>

</Grid>

</Window>

Код программы

using AfishaApp.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity;

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;

using SanatoriyApp.Models;

using SanatoriyApp.Pages;

using SanatoriyApp.Windows;

namespace SanatoriyApp

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new SanatoriyCataloguePage());

Manager.CurrentUser = null;

Manager.MainFrame = MainFrame;

}

private void WindowClosed(object sender, EventArgs e)

{

App.Current.Shutdown();

}

//событие попытки закрытия окна,

// если пользователь выберет Cancel, то форму не закроем

private void WindowClosing(object sender, System.ComponentModel.CancelEventArgs e)

{

MessageBoxResult x = MessageBox.Show("Вы действительно хотите выйти?",

"Выйти", MessageBoxButton.OKCancel, MessageBoxImage.Question);

if (x == MessageBoxResult.Cancel)

e.Cancel = true;

}

// Кнопка назад

private void BtnBackClick(object sender, RoutedEventArgs e)

{

Manager.MainFrame.GoBack();

}

// Кнопка навигации

// Событие отрисовки страницы

// Скрываем или показываем кнопку Назад

// Скрываем или показываем кнопки Для перехода к остальным страницам

private void MainFrameContentRendered(object sender, EventArgs e)

{

if (MainFrame.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

BtnEnter.Visibility = Visibility.Collapsed;

BtnBooking.Visibility = Visibility.Collapsed;

BtnSanatories.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Collapsed;

BtnUsers.Visibility = Visibility.Collapsed;

BtnAllRewiew.Visibility = Visibility.Collapsed;

BtnMyBooking.Visibility = Visibility.Collapsed;

BtnMyrewiew.Visibility = Visibility.Collapsed;

TextBlockUser.Text = "";

//if (Manager.CurrentUser is null)

// return;

//if (Manager.CurrentUser.RoleId == 1)

//{

// BtnBooking.Visibility = Visibility.Collapsed;

// BtnQuests.Visibility = Visibility.Collapsed;

//}

//else

//{

// BtnMyAccount.Visibility = Visibility.Collapsed;

// BtnMyOrders.Visibility = Visibility.Collapsed;

//}

}

else

{

BtnBack.Visibility = Visibility.Collapsed;

BtnEnter.Visibility = Visibility.Visible;

if (Manager.CurrentUser is null)

return;

if (Manager.CurrentUser.RoleId == 1)

{

BtnBooking.Visibility = Visibility.Visible;

BtnSanatories.Visibility = Visibility.Visible;

BtnMyAccount.Visibility = Visibility.Visible;

BtnAllRewiew.Visibility = Visibility.Visible;

BtnMyrewiew.Visibility = Visibility.Collapsed;

BtnMyBooking.Visibility = Visibility.Collapsed;

BtnUsers.Visibility = Visibility.Visible;

}

else

{

BtnBooking.Visibility = Visibility.Collapsed;

BtnSanatories.Visibility = Visibility.Collapsed;

BtnUsers.Visibility = Visibility.Collapsed;

BtnAllRewiew.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Visible;

BtnMyrewiew.Visibility = Visibility.Visible;

BtnMyBooking.Visibility = Visibility.Visible;

}

IconBtnKey.Kind = MaterialDesignThemes.Wpf.PackIconKind.Logout;

BtnEnter.ToolTip = "Выйти";

TextBlockUser.Text = $"{Manager.CurrentUser.UserName}";

}

}

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

{

}

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

{

if (Manager.CurrentUser != null)

{

MessageBoxResult messageBoxResult = MessageBox.Show($"Выйти из системы??? ", "Выход", MessageBoxButton.OKCancel,

MessageBoxImage.Question);

if (messageBoxResult == MessageBoxResult.OK)

{

IconBtnKey.Kind = MaterialDesignThemes.Wpf.PackIconKind.Login;

BtnEnter.ToolTip = "Войти";

Manager.CurrentUser = null;

BtnBack.Visibility = Visibility.Collapsed;

BtnBooking.Visibility = Visibility.Collapsed;

BtnSanatories.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Collapsed;

BtnAllRewiew.Visibility = Visibility.Collapsed;

BtnMyBooking.Visibility = Visibility.Collapsed;

BtnMyrewiew.Visibility = Visibility.Collapsed;

BtnUsers.Visibility = Visibility.Collapsed;

TextBlockUser.Text = "";

MainFrame.NavigationService.Refresh();

return;

}

}

LoginWindow window = new LoginWindow();

if (window.ShowDialog() == true)

{

if (Manager.CurrentUser.RoleId == 1)

{

BtnBooking.Visibility = Visibility.Visible;

BtnSanatories.Visibility = Visibility.Visible;

BtnMyAccount.Visibility = Visibility.Visible;

BtnAllRewiew.Visibility = Visibility.Visible;

BtnMyrewiew.Visibility = Visibility.Collapsed;

BtnUsers.Visibility = Visibility.Visible;

BtnMyBooking.Visibility = Visibility.Collapsed;

}

else

{

BtnBooking.Visibility = Visibility.Collapsed;

BtnSanatories.Visibility = Visibility.Collapsed;

BtnUsers.Visibility = Visibility.Collapsed;

BtnAllRewiew.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Visible;

BtnMyrewiew.Visibility = Visibility.Visible;

BtnMyBooking.Visibility = Visibility.Visible;

}

IconBtnKey.Kind = MaterialDesignThemes.Wpf.PackIconKind.Logout;

BtnEnter.ToolTip = "Выйти";

TextBlockUser.Text = $"{Manager.CurrentUser.UserName}";

}

MainFrame.NavigationService.Refresh();

}

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

{

//MainFrame.Navigate(new StatusPage());

}

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

{

try

{

MyAccountWindow window = new MyAccountWindow();

if (window.ShowDialog() == true)

{

MessageBox.Show("Запись изменена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

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

{

// MainFrame.Navigate(new OrderPage());

}

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

{

MainFrame.Navigate(new MyBookingPage());

}

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

{

// MainFrame.Navigate(new BuyPage());

}

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

{

MainFrame.Navigate(new BookingPage());

}

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

{

MainFrame.Navigate(new MyRewiewPage());

}

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

{

MainFrame.Navigate(new UsersPage());

}

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

{

MainFrame.Navigate(new RewiewsPage());

}

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

{

MainFrame.Navigate(new SanatoriyPage());

}

}

}

Страница Добавить Санаторий

Код интерфейса

<Page x:Class="SanatoriyApp.Pages.AddSanatoriyPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:local="clr-namespace:SanatoriyApp.Pages" xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="800"

Title="Добавление и редактирование"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition Height="30"/>

<RowDefinition Height="80"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="245"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="BtnSave" Grid.Row="9" Content="Сохранить"

HorizontalAlignment="Center" VerticalAlignment="Top" Width="120"

Click="BtnSave\_Click" Grid.Column="1" Margin="213,3,222,0" Background="#FF0C849E"/>

<TextBlock Grid.Row="0" Grid.Column="0" Text="Фото" Margin="20 0"/>

<TextBlock Grid.Row="1" Grid.Column="0" Text="Название" Margin="20 0"/>

<TextBlock Grid.Row="2" Grid.Column="0" Text="Описание" Margin="20 0"/>

<TextBlock Grid.Row="3" Grid.Column="0" Text="max Количество мест" Margin="20 0"/>

<TextBlock Grid.Row="4" Grid.Column="0" Text="Минимальная стоимость, р." Margin="20 0"/>

<TextBlock Grid.Row="5" Grid.Column="0" Text="Направления" Margin="20 0"/>

<!--<Image x:Name="ImagePhoto" Source="{Binding GetPhoto}"

Grid.Row="0" Grid.Column="1" Margin="10 0 0 0"

VerticalAlignment="Stretch" HorizontalAlignment="Stretch" />-->

<Button x:Name="BtnLoad" Grid.Row="0" Grid.Column="0" Margin="120,5,0,0"

HorizontalAlignment="Left" VerticalAlignment="Top"

Click="BtnLoad\_Click" Content="{materialDesign:PackIcon Kind=ImageAdd}" Width="60" BorderBrush="#FF51AC54" Background="#FF0C849E"/>

<TextBox x:Name="TextBoxTitle" Text="{Binding Title}"

Grid.Row="1" Grid.Column="1" Padding="0,2" Height="Auto"/>

<TextBox x:Name="TextBoxDescription" Text="{Binding Description}"

Grid.Row="2" Grid.Column="1" Padding="0,2" Height="Auto" TextWrapping="Wrap"/>

<ListBox x:Name="ListBoxPhotos" Grid.Column="0" ItemsSource="{Binding Photos}"

Margin="0 60 0 0" SelectionChanged="ListBoxPhotos\_SelectionChanged" SelectionMode="Single">

<ListBox.ItemContainerStyle>

<Style TargetType="ListBoxItem">

<Setter Property="Height" Value="50" />

</Style>

</ListBox.ItemContainerStyle>

<ListBox.ItemTemplate>

<DataTemplate>

<StackPanel Orientation="Horizontal">

<Image Margin="3" Source="{Binding GetPhoto}"/>

<Button x:Name="BtnDelete" HorizontalAlignment="Right"

Click="BtnDelete\_Click" ToolTip="Удалить" Style="{StaticResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=Trash}"/>

</StackPanel>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<Image Grid.Column="1" Stretch="Uniform" x:Name="ImageCurrentPhoto"/>

<xctk:IntegerUpDown x:Name="IntegerUpDownCountOfSeats" Grid.Row="3" HorizontalAlignment="Left" DefaultValue="1"

Grid.Column="1" Minimum="1" Width="100" Value="{Binding NumberOfSeats}"/>

<xctk:IntegerUpDown x:Name="IntegerUpDownMinPrice" HorizontalAlignment="Left" Grid.Row="4" DefaultValue="1" Grid.Column="1" Minimum="1" Width="100" Value="{Binding MinPrice, StringFormat='c', ConverterCulture='ru-RU'}"/>

<xctk:CheckComboBox x:Name="ComboCategories" Grid.Row="5" Grid.Column="1"

DisplayMemberPath="Title"

SelectedMemberPath="Selected"

TabIndex="34" MaxDropDownHeight="150"/>

</Grid>

</Page>

Код программный

using Microsoft.Win32;

using SanatoriyApp.Models;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для AddSanatoriyPage.xaml

/// </summary>

public partial class AddSanatoriyPage : Page

{

private Sanatoriy \_currentItem = new Sanatoriy();

private static string \_currentDirectory = Directory.GetCurrentDirectory() + @"\Images\";

List<SanatoriyCategory> currentItems = new List<SanatoriyCategory>();

class ComboBoxTypeItem

{

public int Id { get; set; }

public int SanatoriyCategoryId { get; set; }

public string Title { get; set; }

public bool Selected { get; set; }

}

List<ComboBoxTypeItem> categoriesItems = new List<ComboBoxTypeItem>();

private string GetComboBoxItemContent(List<ComboBoxTypeItem> items)

{

List<string> s = new List<string>();

foreach (ComboBoxTypeItem item in items)

{

if (item.Selected == true)

{

s.Add(item.Title);

}

}

return String.Join(", ", s);

}

private void LoadSanatories()

{

categoriesItems.Clear();

List<SanatoriyCategory> xlist = SanatoriyBDEntities.GetContext().SanatoriyCategories.Where(i => i.QuestId == \_currentItem.Id).ToList();

List<Category> items = SanatoriyBDEntities.GetContext().Categories.ToList();

List<int> datas = new List<int>();

foreach (SanatoriyCategory c in xlist)

{

datas.Add(c.CategoryId);

currentItems.Add(c);

}

// MessageBox.Show(currentItems.Count.ToString());

foreach (Category item in items)

{

if (datas.Contains(item.Id))

{

categoriesItems.Add(new ComboBoxTypeItem

{

Id = item.Id,

Title = item.Title,

Selected = true

});

}

else

categoriesItems.Add(new ComboBoxTypeItem

{

Id = item.Id,

Title = item.Title,

Selected = false

});

}

ComboCategories.ItemsSource = null;

ComboCategories.ItemsSource = categoriesItems;

// производители товаров

}

//сохраytybt

void SaveSanatory()

{

List<SanatoriyCategory> saveItems = new List<SanatoriyCategory>();

foreach (ComboBoxTypeItem item in categoriesItems)

{

if (item.Selected == true)

{

saveItems.Add(new SanatoriyCategory

{

QuestId = \_currentItem.Id,

CategoryId = item.Id,

}); ;

}

}

// MessageBox.Show(currentItems.Count.ToString());

// MessageBox.Show(saveItems.Count.ToString());

List<SanatoriyCategory> delItems = new List<SanatoriyCategory>();

List<SanatoriyCategory> addItems = new List<SanatoriyCategory>();

foreach (SanatoriyCategory x in currentItems)

{

bool b = false;

foreach (SanatoriyCategory y in saveItems)

{

if ((y.CategoryId == x.CategoryId) && (y.QuestId == x.QuestId))

{

b = true;

break;

}

}

if (!b)

delItems.Add(x);

}

foreach (SanatoriyCategory x in saveItems)

{

bool b = false;

foreach (SanatoriyCategory y in currentItems)

{

if ((y.CategoryId == x.CategoryId) && (y.QuestId == x.QuestId))

{

b = true;

break;

}

}

if (!b)

addItems.Add(x);

}

SanatoriyBDEntities.GetContext().SanatoriyCategories.RemoveRange(delItems);

SanatoriyBDEntities.GetContext().SanatoriyCategories.AddRange(addItems);

SanatoriyBDEntities.GetContext().SaveChanges();

}

public AddSanatoriyPage(Sanatoriy selected)

{

InitializeComponent();

LoadAndInitData(selected);

}

void LoadAndInitData(Sanatoriy selected)

{

BtnLoad.Visibility = Visibility.Collapsed;

if (selected != null)

{

\_currentItem = selected;

BtnLoad.Visibility = Visibility.Visible;

if (selected.Photos.Count > 0)

{

ImageCurrentPhoto.Source = new BitmapImage(new Uri(selected.Photos.ToList()[0].GetPhoto));

}

else

{

ImageCurrentPhoto.Source = null;

}

}

DataContext = \_currentItem;

LoadSanatories();

}

/// <summary>

/// Проверка полей ввод на корректыне данные

/// </summary>

/// <returns>текст StringBuilder содержащий ошибки, если они есть</returns>

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

// проверка полей на содержимое

if (string.IsNullOrWhiteSpace(\_currentItem.Title))

s.AppendLine("Поле название пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.Description))

s.AppendLine("Поле описание пустое");

return s;

}

//подбор имени файла

string ChangePhotoName(string \_photoName)

{

string x = \_currentDirectory + \_photoName;

string photoname = \_photoName;

int i = 0;

if (File.Exists(x))

{

while (File.Exists(x))

{

i++;

x = \_currentDirectory + i.ToString() + photoname;

}

photoname = i.ToString() + photoname;

}

return photoname;

}

public AddSanatoriyPage()

{

InitializeComponent();

}

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

{

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

try

{

// проверка полей прошла успешно

// если товар новый, то его ID == 0

if (\_currentItem.Id == 0)

{

SanatoriyBDEntities.GetContext().Sanatoriys.Add(\_currentItem);

}

SanatoriyBDEntities.GetContext().SaveChanges(); // Сохраняем изменения в БД

SaveSanatory();

MessageBox.Show("Запись Изменена");

BtnLoad.Visibility = Visibility.Visible;

// Manager.MainFrame.GoBack(); // Возвращаемся на предыдущую форму

}

catch (Exception ex)

{

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

}

} // загрузка фото

}

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

{

try

{

//Диалог открытия файла

OpenFileDialog op = new OpenFileDialog();

op.Title = "Select a picture";

op.Filter = "JPEG Files (\*.jpeg)|\*.jpeg|PNG Files (\*.png)|\*.png|JPG Files (\*.jpg)|\*.jpg|GIF Files (\*.gif)|\*.gif";

// диалог вернет true, если файл был открыт

if (op.ShowDialog() == true)

{

// проверка размера файла

// по условию файл дожен быть не более 2Мб.

FileInfo fileInfo = new FileInfo(op.FileName);

string photo = ChangePhotoName(op.SafeFileName);

// путь куда нужно скопировать файл

string dest = \_currentDirectory + photo;

File.Copy(op.FileName, dest);

Photo newPhoto = new Photo();

newPhoto.SanatoriyId = \_currentItem.Id;

newPhoto.Photo1 = photo;

SanatoriyBDEntities.GetContext().Photos.Add(newPhoto);

SanatoriyBDEntities.GetContext().SaveChanges();

\_currentItem.ReloadPhotos = 1;

//PhotoStudioSeverBDEntities.GetContext().Entry(\_currentItem).Reload();

// ListBoxPhotos.ItemsSource = null;

ListBoxPhotos.Items.Refresh();

}

}

catch (Exception ex)

{

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

}

}

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

{

{

ListBoxItem selectedItem = (ListBoxItem)ListBoxPhotos.ItemContainerGenerator.

ContainerFromItem(((Button)sender).DataContext);

selectedItem.IsSelected = true;

Photo deletedPhoto = (sender as Button).DataContext as Photo;

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить изображение???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

int k = ListBoxPhotos.SelectedIndex;

SanatoriyBDEntities.GetContext().Photos.Remove(deletedPhoto);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Изображение удалено");

\_currentItem.ReloadPhotos = 1;

//PhotoStudioSeverBDEntities.GetContext().Entry(\_currentItem).Reload();

// ListBoxPhotos.ItemsSource = null;

ListBoxPhotos.Items.Refresh();

if (k > 0) k--;

ListBoxPhotos.SelectedIndex = k;

if (k == 0)

ImageCurrentPhoto.Source = null;

}

catch (Exception ex)

{

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

}

}

}

}

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

{

if (ListBoxPhotos.SelectedItems.Count == 0)

return;

ImageCurrentPhoto.Source = new BitmapImage(new Uri((ListBoxPhotos.SelectedItem as Photo).GetPhoto));

}

}

}

Страница Добавить пользователя

<Page x:Class="SanatoriyApp.Pages.AddUserPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="14"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

d:DesignHeight="600" d:DesignWidth="800"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

IsVisibleChanged="Page\_IsVisibleChanged"

Title="Пользователь" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBox x:Name="TextBoxUserName"

Text="{Binding UserName}" Grid.Column="0" VerticalAlignment="Center" Grid.Row="0" Height="30" Margin="0,7"

materialDesign:HintAssist.Hint="Имя пользователя" Padding="20 5 0 0" TextChanged="TextBoxUserName\_TextChanged" PreviewTextInput="TextBoxUserName\_PreviewTextInput"

/>

<TextBox Text="{Binding LastName}" materialDesign:HintAssist.Hint="Фамилия" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="1" Height="30" Margin="0,7" />

<TextBox Text="{Binding FirstName}" materialDesign:HintAssist.Hint="Имя" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="2" Height="30" Margin="0,7" />

<TextBox Text="{Binding MiddleName}" materialDesign:HintAssist.Hint="Отчество" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="3" Height="30" Margin="0,7" />

<ComboBox x:Name="ComboUserType" materialDesign:HintAssist.Hint="Тип пользователя" Padding="20 5 0 0"

SelectedItem="{Binding Role}"

Grid.Row="4"

DisplayMemberPath="Title" VerticalAlignment="Stretch" VerticalContentAlignment="Center"

SelectedValuePath="Id" />

<xctk:MaskedTextBox x:Name="TbPhone" Text="{Binding Phone}"

Mask="+0 (000) 000-00-00" Style="{DynamicResource MaterialDesignTextBox}"

Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="5" Height="30" Margin="0,7" />

<TextBox Text="{Binding Email}" materialDesign:HintAssist.Hint="E-mail"

Padding="20 5 0 0" Grid.Column="0"

VerticalAlignment="Center" Grid.Row="6" Height="30" Margin="0,7" />

<TextBox Text="{Binding Password}" Grid.Column="0"

Margin="20,0,0,2" x:Name="PasswordBoxPassword" VerticalAlignment="Center"

materialDesign:HintAssist.Hint="Пароль" Grid.Row="7" Height="30" />

<Button IsDefault="True" Click="BtnSaveClick" Width="120" Grid.Row="8" Margin="5" Background="#FF0C849E">Сохранить</Button>

</Grid>

</Page>

Код программный

using SanatoriyApp.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для AddUserPage.xaml

/// </summary>

public partial class AddUserPage : Page

{

public User \_currentItem { get; private set; }

List<User> users = new List<User>();

bool isNew = false;

bool badName = false;

string msg = "Изменения сохранены";

public AddUserPage(User user)

{

InitializeComponent();

LoadAndInitData(user);

}

void LoadAndInitData(User selected)

{ // если передано null, то мы добавляем новый товар

users = SanatoriyBDEntities.GetContext().Users.OrderBy(p => p.UserName).ToList();

ComboUserType.ItemsSource = SanatoriyBDEntities.GetContext().Roles.ToList();

if (selected != null)

{

\_currentItem = selected;

TextBoxUserName.IsReadOnly = true;

}

else

{

\_currentItem = new User();

isNew = true;

msg = "Запись добавлена";

}

// контекст данных текущий товар

DataContext = \_currentItem;

}

/// <summary>

/// Проверка полей ввод на корректыне данные

/// </summary>

/// <returns>текст StringBuilder содержащий ошибки, если они есть</returns>

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentItem.UserName))

s.AppendLine("Задайте имя пользователя");

if (badName)

s.AppendLine("Выберите другое имя пользователя");

if (string.IsNullOrWhiteSpace(\_currentItem.Password))

s.AppendLine("Задайте пароль");

if (ComboUserType.SelectedIndex == -1)

s.AppendLine("Укажите тип пользователя");

return s;

}

private void BtnSaveClick(object sender, RoutedEventArgs e)

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

// проверка полей прошла успешно

// если товар новый, то его ID == 0

if (isNew)

{

// добавление нового товара,

SanatoriyBDEntities.GetContext().Users.Add(\_currentItem);

}

try

{

SanatoriyBDEntities.GetContext().SaveChanges(); // Сохраняем изменения в БД

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

Manager.MainFrame.GoBack(); // Возвращаемся на предыдущую форму

}

catch (Exception ex)

{

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

}

}

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

{

if (Visibility == Visibility.Visible)

{

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

}

}

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

{

string username = TextBoxUserName.Text.ToLower();

User user = SanatoriyBDEntities.GetContext().Users.Where(p => p.UserName.ToLower() == username).FirstOrDefault();

if (user == \_currentItem)

return;

if (user != null)

{

TextBoxUserName.Foreground = new SolidColorBrush(Colors.Red);

badName = true;

}

else

{

TextBoxUserName.Foreground = new SolidColorBrush(Colors.Green);

badName = false;

}

}

private void TextBoxUserName\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

Regex regex = new Regex(@"[^a-zA-Z\s]");

e.Handled = regex.IsMatch(e.Text);

}

}

}

Страница Бронирования санаториев

<Page x:Class="SanatoriyApp.Pages.BookingPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

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

Title="Бронирование места санатория" IsVisibleChanged="PageIsVisibleChanged" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<StackPanel Orientation="Horizontal">

<TextBox Width="275" x:Name="TBoxSearch" materialDesign:HintAssist.Hint="Введите название санатория, клиента"

TextChanged="TBoxSearchTextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

</StackPanel>

<StackPanel Orientation="Horizontal">

<ComboBox Width="275" materialDesign:HintAssist.Hint="Сортировать по"

x:Name="ComboSort"

SelectionChanged="ComboSortSelectionChanged" >

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

</WrapPanel>

<DataGrid x:Name="DataGridGood" Grid.Row="1"

AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single"

LoadingRow="DataGridGoodLoadingRow">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Id}" Header="ID" />

<DataGridTextColumn Binding="{Binding TimeSheet.Sanatoriy.Title}" Header="Санаторий" Width="\*"/>

<DataGridTextColumn Binding="{Binding TimeSheet.Price, StringFormat='c', ConverterCulture='ru-RU'}" Header="Стоимость" Width="\*"/>

<DataGridTextColumn Binding="{Binding UserInfo}" Header="Клиент" Width="\*"/>

<DataGridTextColumn Binding="{Binding Phone}" Header="Телефон" Width="\*"/>

<DataGridTextColumn Binding="{Binding TimeSheet.Date, StringFormat='d', ConverterCulture='ru-RU'}" Header="Дата" Width="\*"/>

<DataGridTextColumn Binding="{Binding TimeSheet.Time, StringFormat='t', ConverterCulture='ru-RU'}" Header="Время" Width="\*"/>

<!--<DataGridCheckBoxColumn Binding="{Binding Payed}" Header="Оплачен" Width="\*"/>-->

<DataGridTemplateColumn Width="auto" Header="Оплачен">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<ToggleButton Content="NO" IsChecked="{Binding Payed}"

Style="{StaticResource MaterialDesignActionLightToggleButton}"

ToolTip="Оплачен" Click="ToggleButton\_Click" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<!--<DataGridTemplateColumn Width="auto" Header="Редактировать">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" Click="BtnEdit\_Click" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>-->

<DataGridTemplateColumn Width="auto" Header="Удалить">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Content="{materialDesign:PackIcon Kind=Delete}" Click="BtnDeleteClick" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<TextBlock Grid.Row="2" x:Name="TextBlockCount" Width="300" HorizontalAlignment="Left" />

</Grid>

</Page>

Код программный

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

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.Windows.Controls.Primitives;

using SanatoriyApp.Models;

namespace SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для BookingPage.xaml

/// </summary>

public partial class BookingPage : Page

{

int \_itemcount = 0;

public BookingPage()

{

InitializeComponent();

}

void LoadData()

{

try

{

DataGridGood.ItemsSource = null;

//загрузка обновленных данных

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

List<Booking> goods = SanatoriyBDEntities.GetContext().Bookings.OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnAddClick(object sender, RoutedEventArgs e)

{

//try

//{

// OrganizerWindow window = new OrganizerWindow(new Organizer());

// if (window.ShowDialog() == true)

// {

// PhotoStudioSeverBDEntities.GetContext().Organizers.Add(window.currentItem);

// PhotoStudioSeverBDEntities.GetContext().SaveChanges();

// LoadData();

// MessageBox.Show("Запись добавлена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

// }

//}

//catch

//{

// MessageBox.Show("Ошибка");

//}

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Booking selected = (sender as Button).DataContext as Booking;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

SanatoriyBDEntities.GetContext().Bookings.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

var currentGoods = SanatoriyBDEntities.GetContext().Bookings.OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

// выбор только тех товаров, по определенному диапазону скидки

currentGoods = currentGoods.Where(p => p.UserInfo.ToLower().Contains(TBoxSearch.Text.ToLower()) ||

p.TimeSheet.Sanatoriy.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.TimeSheet.Date).ThenByDescending(p => p.TimeSheet.Time).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

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

{

//try

//{

// // если ни одного объекта не выделено, выходим

// if (DataGridGood.SelectedItem == null) return;

// // получаем выделенный объект

// Organizer selected = DataGridGood.SelectedItem as Organizer;

// OrganizerWindow window = new OrganizerWindow(selected);

// if (window.ShowDialog() == true)

// {

// if (window.currentItem != null)

// {

// PhotoStudioSeverBDEntities.GetContext().Entry(window.currentItem).State = EntityState.Modified;

// PhotoStudioSeverBDEntities.GetContext().SaveChanges();

// LoadData();

// MessageBox.Show("Запись изменена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

// }

// }

// else

// {

// PhotoStudioSeverBDEntities.GetContext().Entry(window.currentItem).Reload();

// LoadData();

// }

//}

//catch

//{

// MessageBox.Show("Ошибка");

//}

}

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

{

Booking selected = (sender as ToggleButton).DataContext as Booking;

if ((sender as ToggleButton).IsChecked == true)

{

selected.Payed = true;

SanatoriyBDEntities.GetContext().SaveChanges();

}

else

{

selected.Payed = false;

SanatoriyBDEntities.GetContext().SaveChanges();

}

}

}

}

Страница Категория

<Page x:Class="SanatoriyApp.Pages.CategoryPage"

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:SanatoriyApp.Pages"

mc:Ignorable="d"

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

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

Title="Категории санаториев" IsVisibleChanged="PageIsVisibleChanged" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<StackPanel Orientation="Horizontal">

<TextBox Width="275" x:Name="TBoxSearch" materialDesign:HintAssist.Hint="Введите название"

TextChanged="TBoxSearchTextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

</StackPanel>

<StackPanel Orientation="Horizontal">

<ComboBox Width="275" materialDesign:HintAssist.Hint="Сортировать по"

x:Name="ComboSort"

SelectionChanged="ComboSortSelectionChanged" >

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

</WrapPanel>

<DataGrid x:Name="DataGridGood" Grid.Row="1"

AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single"

LoadingRow="DataGridGoodLoadingRow">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Id}" Header="ID" />

<DataGridTextColumn Binding="{Binding Title}" Header="Название" Width="\*"/>

<DataGridTemplateColumn Width="auto" Header="Редактировать">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" Click="BtnEdit\_Click" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto" Header="Удалить">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Content="{materialDesign:PackIcon Kind=Delete}" Click="BtnDeleteClick" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<TextBlock Grid.Row="2" x:Name="TextBlockCount" Width="300" HorizontalAlignment="Left" />

<Button x:Name="BtnAdd" Grid.Row="2" Width="100" HorizontalAlignment="Right" Margin="3"

Content="{materialDesign:PackIcon Kind=AddBold}" Click="BtnAddClick" Background="#FF0C849E"/>

</Grid>

</Page>

Код программный

using SanatoriyApp.Models;

using SanatoriyApp.Windows;

using System;

using System.Collections.Generic;

using System.Data.Entity;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для CategoryPage.xaml

/// </summary>

public partial class CategoryPage : Page

{

int \_itemcount = 0;

public CategoryPage()

{

InitializeComponent();

}

void LoadData()

{

try

{

DataGridGood.ItemsSource = null;

//загрузка обновленных данных

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

List<Category> goods = SanatoriyBDEntities.GetContext().Categories.OrderBy(p => p.Title).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnAddClick(object sender, RoutedEventArgs e)

{

try

{

CategoryWindow window = new CategoryWindow(new Category());

if (window.ShowDialog() == true)

{

SanatoriyBDEntities.GetContext().Categories.Add(window.currentItem);

SanatoriyBDEntities.GetContext().SaveChanges();

LoadData();

MessageBox.Show("Запись добавлена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Category selected = (sender as Button).DataContext as Category;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

if (selected.SanatoriyCategories.Count > 0)

throw new Exception("Есть зависимые записи о санаториях");

SanatoriyBDEntities.GetContext().Categories.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

UpdateData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

var currentGoods = SanatoriyBDEntities.GetContext().Categories.OrderBy(p => p.Title).ToList();

// выбор только тех товаров, по определенному диапазону скидки

currentGoods = currentGoods.Where(p => p.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.Title).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.Title).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

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

{

try

{

// если ни одного объекта не выделено, выходим

if (DataGridGood.SelectedItem == null) return;

// получаем выделенный объект

Category selected = DataGridGood.SelectedItem as Category;

CategoryWindow window = new CategoryWindow(selected);

if (window.ShowDialog() == true)

{

if (window.currentItem != null)

{

SanatoriyBDEntities.GetContext().Entry(window.currentItem).State = EntityState.Modified;

SanatoriyBDEntities.GetContext().SaveChanges();

LoadData();

MessageBox.Show("Запись изменена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

else

{

SanatoriyBDEntities.GetContext().Entry(window.currentItem).Reload();

LoadData();

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

}

}

Страница моих бронирований

Код интерфейса

<Page x:Class="SanatoriyApp.Pages.MyBookingPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

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

Title="Бронирование" IsVisibleChanged="PageIsVisibleChanged" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<StackPanel Orientation="Horizontal">

<TextBox Width="275" x:Name="TBoxSearch" materialDesign:HintAssist.Hint="Введите название санатория"

TextChanged="TBoxSearchTextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

</StackPanel>

<StackPanel Orientation="Horizontal">

<ComboBox Width="275" materialDesign:HintAssist.Hint="Сортировать по"

x:Name="ComboSort"

SelectionChanged="ComboSortSelectionChanged" >

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

</WrapPanel>

<DataGrid x:Name="DataGridGood" Grid.Row="1"

AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single"

LoadingRow="DataGridGoodLoadingRow">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Id}" Header="ID" />

<DataGridTextColumn Binding="{Binding TimeSheet.Sanatoriy.Title}" Header="Квест" Width="\*"/>

<DataGridTextColumn Binding="{Binding TimeSheet.Price, StringFormat='c', ConverterCulture='ru-RU'}" Header="Стоимость" Width="\*"/>

<DataGridTextColumn Binding="{Binding UserInfo}" Header="Клиент" Width="\*"/>

<DataGridTextColumn Binding="{Binding Phone}" Header="Телефон" Width="\*"/>

<DataGridTextColumn Binding="{Binding TimeSheet.Date, StringFormat='d', ConverterCulture='ru-RU'}" Header="Дата" Width="\*"/>

<DataGridTextColumn Binding="{Binding TimeSheet.Time, StringFormat='t', ConverterCulture='ru-RU'}" Header="Время" Width="\*"/>

<!--<DataGridCheckBoxColumn Binding="{Binding Payed}" Header="Оплачен" Width="\*"/>-->

<DataGridTemplateColumn Width="auto" Header="Отменить" >

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Visibility="{Binding GetVisibility}" Content="{materialDesign:PackIcon Kind=Delete}" Click="BtnDeleteClick" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<TextBlock Grid.Row="2" x:Name="TextBlockCount" Width="300" HorizontalAlignment="Left" />

</Grid>

</Page>

Код программный using SanatoriyApp.Models;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для MyBookingPage.xaml

/// </summary>

public partial class MyBookingPage : Page

{

int \_itemcount = 0;

public MyBookingPage()

{

InitializeComponent();

}

void LoadData()

{

try

{

DataGridGood.ItemsSource = null;

//загрузка обновленных данных

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

string username = Manager.CurrentUser.UserName;

List<Booking> goods = SanatoriyBDEntities.GetContext().Bookings.Where(p => p.UserName == username).OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Booking selected = (sender as Button).DataContext as Booking;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

SanatoriyBDEntities.GetContext().Bookings.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

string username = Manager.CurrentUser.UserName;

var currentGoods = SanatoriyBDEntities.GetContext().Bookings.Where(p => p.UserName == username).OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

// выбор только тех товаров, по определенному диапазону скидки

currentGoods = currentGoods.Where(p => p.UserInfo.ToLower().Contains(TBoxSearch.Text.ToLower()) ||

p.TimeSheet.Sanatoriy.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.TimeSheet.Date).ThenByDescending(p => p.TimeSheet.Time).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

}

}

Страница мои отзывы

Код интерфейса

using SanatoriyApp.Models;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для MyBookingPage.xaml

/// </summary>

public partial class MyBookingPage : Page

{

int \_itemcount = 0;

public MyBookingPage()

{

InitializeComponent();

}

void LoadData()

{

try

{

DataGridGood.ItemsSource = null;

//загрузка обновленных данных

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

string username = Manager.CurrentUser.UserName;

List<Booking> goods = SanatoriyBDEntities.GetContext().Bookings.Where(p => p.UserName == username).OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Booking selected = (sender as Button).DataContext as Booking;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

SanatoriyBDEntities.GetContext().Bookings.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

string username = Manager.CurrentUser.UserName;

var currentGoods = SanatoriyBDEntities.GetContext().Bookings.Where(p => p.UserName == username).OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

// выбор только тех товаров, по определенному диапазону скидки

currentGoods = currentGoods.Where(p => p.UserInfo.ToLower().Contains(TBoxSearch.Text.ToLower()) ||

p.TimeSheet.Sanatoriy.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.TimeSheet.Date).ThenBy(p => p.TimeSheet.Time).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.TimeSheet.Date).ThenByDescending(p => p.TimeSheet.Time).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

}

} Код программный

using SanatoriyApp.Models;

using SanatoriyApp.Windows;

using System;

using System.Collections.Generic;

using System.Data.Entity;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для MyRewiewPage.xaml

/// </summary>

public partial class MyRewiewPage : Page

{

int \_itemcount = 0;

public MyRewiewPage()

{

InitializeComponent();

}

void LoadData()

{

try

{

DataGridGood.ItemsSource = null;

//загрузка обновленных данных

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

string username = Manager.CurrentUser.UserName;

List<Rewiew> goods = SanatoriyBDEntities.GetContext().Rewiews.Where(p => p.UserName == username).OrderBy(p => p.Title).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Rewiew selected = (sender as Button).DataContext as Rewiew;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

SanatoriyBDEntities.GetContext().Rewiews.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

string username = Manager.CurrentUser.UserName;

var currentGoods = SanatoriyBDEntities.GetContext().Rewiews.Where(p => p.UserName == username).OrderBy(p => p.Title).ToList();

// выбор только тех товаров, по определенному диапазону скидки

currentGoods = currentGoods.Where(p => p.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.Sanatoriy.Title).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.Sanatoriy.Title).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

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

{

DialogHostMoreInformation.IsOpen = false;

}

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

{

try

{

// если ни одного объекта не выделено, выходим

if (DataGridGood.SelectedItem == null) return;

// получаем выделенный объект

Rewiew selected = DataGridGood.SelectedItem as Rewiew;

MakeRewiewWindow window = new MakeRewiewWindow(selected, selected.Sanatoriy);

if (window.ShowDialog() == true)

{

if (window.currentRewiew != null)

{

SanatoriyBDEntities.GetContext().Entry(window.currentRewiew).State = EntityState.Modified;

SanatoriyBDEntities.GetContext().SaveChanges();

LoadData();

MessageBox.Show("Запись изменена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

else

{

SanatoriyBDEntities.GetContext().Entry(window.currentRewiew).Reload();

LoadData();

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

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

{

try

{

// если ни одного объекта не выделено, выходим

if (DataGridGood.SelectedItem == null) return;

// получаем выделенный объект

Rewiew selected = DataGridGood.SelectedItem as Rewiew;

//Trainer trainer = YogaFeatPilatesBDEntities.GetContext().Trainers.FirstOrDefault(p => p.Id == edu.TrainerId);

//List<TimeTable> timeTables = YogaFeatPilatesBDEntities.GetContext().TimeTables.Where(p => p.CategoryTrainerId == edu.Id).ToList();

//ListBoxTimeTable.ItemsSource = timeTables;

//TbCategoryName.Text = edu.Category.Name;

DialogHostMoreInformation.DataContext = selected;

DialogHostMoreInformation.IsOpen = true;

}

catch

{

MessageBox.Show("Ошибка");

}

}

}

}

Страница отзывов

Код интерфейса

<Page x:Class="SanatoriyApp.Pages.RewiewsPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

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

Title="Мои отзывы" IsVisibleChanged="PageIsVisibleChanged" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<StackPanel Orientation="Horizontal">

<TextBox Width="275" x:Name="TBoxSearch" materialDesign:HintAssist.Hint="Имя пользователя, ФИО или название комнаты"

TextChanged="TBoxSearchTextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

</StackPanel>

<StackPanel Orientation="Horizontal">

<ComboBox Width="275" materialDesign:HintAssist.Hint="Сортировать по"

x:Name="ComboSort"

SelectionChanged="ComboSortSelectionChanged" >

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

</StackPanel>

</WrapPanel>

<DataGrid x:Name="DataGridGood" Grid.Row="1"

AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single"

LoadingRow="DataGridGoodLoadingRow">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding Id}" Header="ID" />

<DataGridTextColumn Binding="{Binding Sanatoriy.Title}" Header="Санаторий" Width="\*"/>

<DataGridTextColumn Binding="{Binding UserName}" Header="Имя пользователя" Width="\*"/>

<DataGridTextColumn Binding="{Binding User.GetFIO}" Header="ФИО" Width="\*"/>

<DataGridTextColumn Binding="{Binding Rate}" Header="Рейтинг" Width="\*"/>

<DataGridTemplateColumn Width="auto" Header="Просмотр">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Eye}" Click="BtnEdit\_Click" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto" Header="Удалить">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Content="{materialDesign:PackIcon Kind=Delete}" Click="BtnDeleteClick" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<TextBlock Grid.Row="2" x:Name="TextBlockCount" Width="300" HorizontalAlignment="Left" />

<materialDesign:DialogHost x:Name="DialogHostMoreInformation"

Background="{StaticResource MaterialDesignDarkBackground}"

Width="600" MaxWidth="600"

Height="400" MaxHeight="400"

BorderThickness="1" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.RowSpan="2" Grid.ColumnSpan="2"

>

<materialDesign:DialogHost.DialogContent >

<StackPanel Margin="20" >

<Grid materialDesign:ColorZoneAssist.Mode="Dark">

<Grid.RowDefinitions>

<RowDefinition Height="60" />

<RowDefinition Height="340" />

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Margin="5" TextAlignment="Center" VerticalAlignment="Stretch"

Text="Отзывы" TextWrapping="Wrap" HorizontalAlignment="Center"

FontWeight="Bold" FontSize="14" />

<materialDesign:Card Width="600" Grid.Row="1">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="180"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<materialDesign:RatingBar Margin="10 0 0 0" IsPreviewValueEnabled="True" Grid.Row="0" Grid.Column="0" IsReadOnly="True" Value="{Binding Rate}" Max="5" />

<TextBlock FontSize="16" Margin="20 0 0 0"

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

Style="{StaticResource MaterialDesignSubtitle2TextBlock }"

Text="{Binding Path=User.GetFIO}" TextWrapping="Wrap" Width="600"

HorizontalAlignment="Stretch" />

<TextBlock FontSize="14" Margin="10 0 0 0" Text="Отзыв" Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2"

Style="{StaticResource MaterialDesignSubtitle2TextBlock }"

HorizontalAlignment="Stretch"/>

<ScrollViewer HorizontalScrollBarVisibility="Disabled" VerticalScrollBarVisibility="Auto" Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2">

<TextBlock FontSize="14" Margin="10 0 0 0" Text="{Binding Title}" TextWrapping="Wrap"

HorizontalAlignment="Stretch"/>

</ScrollViewer>

</Grid>

</materialDesign:Card>

</Grid>

<Button x:Name="btnOk" Click="BtnOk\_Click" Width="100" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<materialDesign:PackIcon Kind="Check" Foreground="#FFF" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="OK" VerticalAlignment="Center"/>

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Page>

Код программный

using SanatoriyApp.Models;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для RewiewsPage.xaml

/// </summary>

public partial class RewiewsPage : Page

{

int \_itemcount = 0;

public RewiewsPage()

{

InitializeComponent();

}

void LoadData()

{

try

{

DataGridGood.ItemsSource = null;

//загрузка обновленных данных

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

List<Rewiew> goods = SanatoriyBDEntities.GetContext().Rewiews.OrderBy(p => p.UserName).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Rewiew selected = (sender as Button).DataContext as Rewiew;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

SanatoriyBDEntities.GetContext().Rewiews.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

var currentGoods = SanatoriyBDEntities.GetContext().Rewiews.OrderBy(p => p.UserName).ToList();

// выбор только тех товаров, по определенному диапазону скидки

currentGoods = currentGoods.Where(p => p.UserName.ToLower().Contains(TBoxSearch.Text.ToLower())

|| p.User.GetFIO.ToLower().Contains(TBoxSearch.Text.ToLower())

|| p.Sanatoriy.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.Rate).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.Rate).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

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

{

DialogHostMoreInformation.IsOpen = false;

}

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

{

try

{

// если ни одного объекта не выделено, выходим

if (DataGridGood.SelectedItem == null) return;

// получаем выделенный объект

Rewiew selected = DataGridGood.SelectedItem as Rewiew;

//Trainer trainer = YogaFeatPilatesBDEntities.GetContext().Trainers.FirstOrDefault(p => p.Id == edu.TrainerId);

//List<TimeTable> timeTables = YogaFeatPilatesBDEntities.GetContext().TimeTables.Where(p => p.CategoryTrainerId == edu.Id).ToList();

//ListBoxTimeTable.ItemsSource = timeTables;

//TbCategoryName.Text = edu.Category.Name;

DialogHostMoreInformation.DataContext = selected;

DialogHostMoreInformation.IsOpen = true;

}

catch

{

MessageBox.Show("Ошибка");

}

}

}

}

Страница Санаторий каталог

Код интерфейса

<Page x:Class="SanatoriyApp.Pages.SanatoriyCataloguePage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}"

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

Title="Санатории" IsVisibleChanged="Page\_IsVisibleChanged"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="71\*"/>

<RowDefinition Height="3\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<TextBox Width="275" x:Name="TBoxSearch" materialDesign:HintAssist.Hint="Введите название санатория" TextChanged="TBoxSearch\_TextChanged"

/>

<ComboBox Width="275" x:Name="ComboCategory" FontSize="14" Padding="3" materialDesign:HintAssist.Hint="Категория"

DisplayMemberPath="Title" SelectionChanged="ComboCategory\_SelectionChanged"/>

<ComboBox Width="275" materialDesign:HintAssist.Hint="Сортировать по"

x:Name="ComboSort" SelectionChanged="ComboSort\_SelectionChanged"

Margin="10 0 0 0" >

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Название"/>

<materialDesign:PackIcon Kind="SortAscending" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Название"/>

<materialDesign:PackIcon Kind="SortDescending" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Рейтинг"/>

<materialDesign:PackIcon Kind="SortNumericAscendingVariant" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Рейтинг"/>

<materialDesign:PackIcon Kind="SortNumericDescendingVariant" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

</ComboBox>

</WrapPanel>

<ListView x:Name="LViewGoods" Grid.Row="1"

ScrollViewer.HorizontalScrollBarVisibility="Disabled" Grid.RowSpan="2"

>

<ListView.ItemsPanel >

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<materialDesign:Card Width="400" Height="600" Background="{StaticResource MaterialDesignDarkBackground}" >

<Grid materialDesign:ColorZoneAssist.Mode="SecondaryMid">

<Grid.RowDefinitions>

<RowDefinition Height="300" />

<RowDefinition Height="60" />

<RowDefinition Height="60" />

<RowDefinition Height="100" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Image x:Name="ImagePhoto" Source="{Binding Path=GetCurrentPhoto}" Grid.Row="0" Height="300" Width="400"

Stretch="Uniform" />

<Button Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left" Style="{DynamicResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowLeft, Size=24}" Click="Button\_Click\_1" Tag="{Binding ElementName=ImagePhoto}"

/>

<Button Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right" Style="{DynamicResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowRight, Size=24}" Click="Button\_Click\_2" Tag="{Binding ElementName=ImagePhoto}"

/>

<TextBlock Grid.Row="1" Text="{Binding Title}" HorizontalAlignment="Left" Foreground="White"

FontWeight="Bold" Margin="10"

FontSize="22"/>

<Button Grid.Row="2" HorizontalAlignment="Left" Style="{DynamicResource MaterialDesignFlatButton}" Click="BtnMoreInfo\_Click">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Рейтинг по отзывам: " Style="{StaticResource MaterialDesignSubtitle2TextBlock}"/>

<materialDesign:RatingBar x:Name="RatingBarRate" Margin="10 0 0 0" IsReadOnly="True"

Value="{Binding GetRate}" MouseDown="RatingBarRate\_MouseDown" />

</StackPanel>

</Button>

<TextBlock Grid.Row="3" Text="{Binding GetInfo}" TextWrapping="Wrap" Foreground="White" Padding="5" FontSize="14"/>

<Button Content="ЗАБРОНИРОВАТЬ" Grid.Row="4" Visibility="{Binding GetVisibility}" Click="Button\_Click"/>

<Button Content="Оставить отзыв" Grid.Row="5" Visibility="{Binding GetVisibility}" Click="BtnMakeRewiew\_Click" />

</Grid>

</materialDesign:Card>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBlock Grid.Row="3" x:Name="TextBlockCount" />

<materialDesign:DialogHost x:Name="DialogHostMoreInformation"

Background="{StaticResource MaterialDesignDarkBackground}"

Width="600" MaxWidth="600"

Height="400" MaxHeight="400"

BorderThickness="1" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.RowSpan="3" Margin="100,10"

>

<materialDesign:DialogHost.DialogContent >

<StackPanel Margin="20" >

<Grid materialDesign:ColorZoneAssist.Mode="Dark">

<Grid.RowDefinitions>

<RowDefinition Height="60" />

<RowDefinition Height="340" />

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Margin="5" TextAlignment="Center" VerticalAlignment="Stretch"

Text="Отзывы" TextWrapping="Wrap" HorizontalAlignment="Center"

FontWeight="Bold" FontSize="14" />

<ScrollViewer HorizontalScrollBarVisibility="Disabled" VerticalScrollBarVisibility="Auto" Grid.Row="1">

<ListBox x:Name="ListBoxRewiews" VerticalAlignment="Stretch" >

<ListBox.ItemTemplate>

<DataTemplate>

<materialDesign:Card Width="600">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition Height="30"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="180"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<materialDesign:RatingBar Margin="10 0 0 0" IsPreviewValueEnabled="True" Grid.Row="0" Grid.Column="0" IsReadOnly="True" Value="{Binding Rate}" Max="5" />

<TextBlock FontSize="16" Margin="20 0 0 0"

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

Style="{StaticResource MaterialDesignSubtitle2TextBlock }"

Text="{Binding Path=User.GetFIO}" TextWrapping="Wrap" Width="600"

HorizontalAlignment="Stretch" />

<TextBlock FontSize="14" Margin="10 0 0 0" Text="Отзыв" Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2"

Style="{StaticResource MaterialDesignSubtitle2TextBlock }"

HorizontalAlignment="Stretch"/>

<TextBlock FontSize="14" Margin="10 0 0 0" Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" Text="{Binding Title}" TextWrapping="Wrap"

HorizontalAlignment="Stretch"/>

</Grid>

</materialDesign:Card>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

</ScrollViewer>

</Grid>

<Button x:Name="btnOk" Click="BtnOk\_Click" Width="100" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<materialDesign:PackIcon Kind="Check" Foreground="#FFF" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="OK" VerticalAlignment="Center"/>

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Page>

Код программный

using SanatoriyApp.Models;

using SanatoriyApp.Windows;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для SanatoriyCataloguePage.xaml

/// </summary>

public partial class SanatoriyCataloguePage : Page

{

int \_itemcount = 0;

public SanatoriyCataloguePage()

{

InitializeComponent();

LoadComboBoxItems();

LoadDataGrid();

}

void LoadDataGrid()

{

//загрузка обновленных данных

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

List<Sanatoriy> goods = SanatoriyBDEntities.GetContext().Sanatoriys.OrderBy(p => p.Title).ToList();

LViewGoods.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

void LoadComboBoxItems()

{

var categories = SanatoriyBDEntities.GetContext().Categories.OrderBy(p => p.Title).ToList();

categories.Insert(0, new Category

{

Title = "Все виды"

}

);

ComboCategory.ItemsSource = categories;

ComboCategory.SelectedIndex = 0;

}

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

//var currentGoods = DataBDEntities.GetContext().Abonements.OrderBy(p => p.CategoryTrainer.Trainer.LastName).ToList();

var currentData = SanatoriyBDEntities.GetContext().Sanatoriys.OrderBy(p => p.Title).ToList();

// выбор только тех товаров, которые принадлежат данному производителю

if (ComboCategory.SelectedIndex > 0)

{

int catId = Convert.ToInt32((ComboCategory.SelectedItem as Category).Id);

List<Sanatoriy> quests = new List<Sanatoriy>();

foreach (Sanatoriy quest in currentData)

{

List<SanatoriyCategory> questCategories = quest.SanatoriyCategories .ToList();

if (questCategories.Any(elem => elem.CategoryId == catId))

quests.Add(quest);

}

currentData = quests;

}

// выбор тех товаров, в названии которых есть поисковая строка

currentData = currentData.Where(p => p.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentData = currentData.OrderBy(p => p.Title).ToList();

if (ComboSort.SelectedIndex == 1)

currentData = currentData.OrderByDescending(p => p.Title).ToList();

if (ComboSort.SelectedIndex == 2)

currentData = currentData.OrderBy(p => p.GetRate).ToList();

if (ComboSort.SelectedIndex == 3)

currentData = currentData.OrderByDescending(p => p.GetRate).ToList();

// сортировка по убыванию цены

}

// В качестве источника данных присваиваем список данных

LViewGoods.ItemsSource = currentData;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentData.Count} записей из {\_itemcount}";

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

Sanatoriy quest = (sender as Button).DataContext as Sanatoriy;

if (quest.Rewiews.Count == 0)

return;

//Trainer trainer = YogaFeatPilatesBDEntities.GetContext().Trainers.FirstOrDefault(p => p.Id == edu.TrainerId);

//List<TimeTable> timeTables = YogaFeatPilatesBDEntities.GetContext().TimeTables.Where(p => p.CategoryTrainerId == edu.Id).ToList();

//ListBoxTimeTable.ItemsSource = timeTables;

//TbCategoryName.Text = edu.Category.Name;

ListBoxRewiews.ItemsSource = quest.Rewiews;

DialogHostMoreInformation.DataContext = quest;

DialogHostMoreInformation.IsOpen = true;

}

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

{

DialogHostMoreInformation.IsOpen = false;

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

BookingWindow bookingWindow = new BookingWindow(sanatoriy);

bookingWindow.ShowDialog();

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

MakeRewiewWindow makeRewiew = new MakeRewiewWindow(new Rewiew(), sanatoriy);

if (makeRewiew.ShowDialog() == true)

{

Manager.MainFrame.NavigationService.Refresh();

}

}

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

{

if (Visibility == Visibility.Visible)

{

LoadDataGrid();

}

}

private void RatingBarRate\_MouseDown(object sender, MouseButtonEventArgs e)

{

Sanatoriy quest = (sender as Button).DataContext as Sanatoriy;

//Trainer trainer = YogaFeatPilatesBDEntities.GetContext().Trainers.FirstOrDefault(p => p.Id == edu.TrainerId);

////List<TimeTable> timeTables = YogaFeatPilatesBDEntities.GetContext().TimeTables.Where(p => p.CategoryTrainerId == edu.Id).ToList();

////ListBoxTimeTable.ItemsSource = timeTables;

////TbCategoryName.Text = edu.Category.Name;

ListBoxRewiews.ItemsSource = quest.Rewiews;

DialogHostMoreInformation.DataContext = quest;

DialogHostMoreInformation.IsOpen = true;

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetPrevPhoto = 1;

Image image = (sender as Button).Tag as Image;

try

{

image.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

catch (Exception)

{

MessageBox.Show("Фотографий нет");

}

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetNextPhoto = 1;

Image image = (sender as Button).Tag as Image;

try

{

image.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

catch (Exception)

{

MessageBox.Show("Фотографий нет");

}

}

}

}

Страница санаторий  
Код интерфейса

<Page x:Class="SanatoriyApp.Pages.SanatoriyPage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

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

Title="Санатории" IsVisibleChanged="Page\_IsVisibleChanged"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<TextBox Width="275" x:Name="TBoxSearch" materialDesign:HintAssist.Hint="Введите название санатория" TextChanged="TBoxSearch\_TextChanged"

/>

<ComboBox Width="275" x:Name="ComboCategory" FontSize="14" Padding="3" materialDesign:HintAssist.Hint="Категория"

DisplayMemberPath="Title" SelectionChanged="ComboCategory\_SelectionChanged"/>

<ComboBox Width="275" materialDesign:HintAssist.Hint="Сортировать по"

x:Name="ComboSort" SelectionChanged="ComboSort\_SelectionChanged"

Margin="10 0 0 0" >

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Название"/>

<materialDesign:PackIcon Kind="SortAscending" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Название"/>

<materialDesign:PackIcon Kind="SortDescending" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Рейтинг"/>

<materialDesign:PackIcon Kind="SortNumericAscendingVariant" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

<ComboBoxItem >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Рейтинг"/>

<materialDesign:PackIcon Kind="SortNumericDescendingVariant" Margin="5 0 0 0"/>

</StackPanel>

</ComboBoxItem>

</ComboBox>

</WrapPanel>

<DataGrid x:Name="DataGridGood" Grid.Row="1"

AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single"

Style="{DynamicResource MaterialDesignDataGrid}" >

<DataGrid.Resources>

<Style TargetType="DataGridRow">

<Setter Property="Background" Value="{Binding GetColor}"/>

</Style>

<Style TargetType="TextBlock" x:Key="WrapText">

<Setter Property="TextWrapping" Value="Wrap"/>

<Setter Property="TextAlignment" Value="Justify"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

<Style TargetType="TextBlock" x:Key="First">

<Setter Property="TextWrapping" Value="Wrap"/>

<Setter Property="TextAlignment" Value="Left"/>

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

<Style TargetType="TextBlock" x:Key="Align">

<Setter Property="TextAlignment" Value="Center" />

<Setter Property="VerticalAlignment" Value="Center"/>

</Style>

</DataGrid.Resources>

<DataGrid.Columns>

<DataGridTemplateColumn Width="200" Header="Фото">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Grid>

<Image x:Name="ImagePhoto" Source="{Binding Path=GetCurrentPhoto}" Grid.Row="0" Height="200"

Stretch="Uniform" />

<Button Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left" Style="{DynamicResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowLeft}" Click="Button\_Click\_1" Tag="{Binding ElementName=ImagePhoto}"

/>

<Button Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right" Style="{DynamicResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowRight}" Click="Button\_Click\_2" Tag="{Binding ElementName=ImagePhoto}"

/>

</Grid>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Binding="{Binding MinPrice, StringFormat='c', ConverterCulture='ru-RU'}" Header="Минимальная стоимость" Width="\*" ElementStyle="{StaticResource WrapText}" />

<DataGridTextColumn Binding="{Binding Title}" Header="Название" Width="\*" ElementStyle="{StaticResource WrapText}" />

<DataGridTextColumn Binding="{Binding Description}" Header="Описание" Width="3\*" ElementStyle="{StaticResource WrapText}"/>

<!--//<DataGridTextColumn Binding="{Binding FearLevel}" Header="{materialDesign:PackIcon Kind=Halloween}" />-->

<DataGridTemplateColumn Width="auto" Header="Рейтинг" >

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<materialDesign:RatingBar x:Name="ratingBarFearLevel" IsReadOnly="True" HorizontalAlignment="Center" Max="5" Orientation="Horizontal"

Value="{Binding GetRate}" VerticalAlignment="Center">

<materialDesign:RatingBar.ValueItemTemplate>

<DataTemplate DataType="system:Int32">

<Grid>

<materialDesign:PackIcon Width="24" Height="24" Kind="Star" />

<TextBlock HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="8" Foreground="{DynamicResource PrimaryHueMidForegroundBrush}" Text="{Binding}" />

</Grid>

</DataTemplate>

</materialDesign:RatingBar.ValueItemTemplate>

</materialDesign:RatingBar>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Binding="{Binding NumberOfSeats}" Header="max Кол-во людей" ElementStyle="{StaticResource Align}"/>

<DataGridTemplateColumn Width="auto" Header="Удалить">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Margin="0" Click="BtnDelete\_Click">

<materialDesign:PackIcon Kind="Delete" />

</Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto" Header="Изменить">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Margin="0" Click="BtnEdit\_Click">

<materialDesign:PackIcon Kind="Edit" />

</Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto" Header="Расписание">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnTimeTable" Margin="0" Click="BtnTimeTable\_Click">

<materialDesign:PackIcon Kind="Timelapse" />

</Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Orientation="Horizontal" Grid.Row="2" HorizontalAlignment="Right">

<Button x:Name="BtnCategories"

Click="BtnCategories\_Click" ToolTip="Категории санаториев" Background="#FF0C849E">

<materialDesign:PackIcon Kind="FormatListBulletedType" Padding="0"/>

</Button>

<Button x:Name="BtnAdd" Grid.Row="1" Width="Auto"

Content="{materialDesign:PackIcon Kind=Add}" ToolTip="Добавить" Click="BtnAdd\_Click" Background="#FF0C849E" />

<Button x:Name="BtnDelete" Grid.Row="1" Width="Auto" Margin="3"

Content="{materialDesign:PackIcon Kind=Delete}" ToolTip="Удалить" Background="#FF0C849E" />

</StackPanel>

<TextBlock Grid.Row="2" x:Name="TextBlockCount" Width="300" HorizontalAlignment="Left" />

</Grid>

</Page>

Код программный

using SanatoriyApp.Models;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для SanatoriyPage.xaml

/// </summary>

public partial class SanatoriyPage : Page

{

int \_itemcount = 0;

public SanatoriyPage()

{

InitializeComponent();

LoadComboBoxItems();

LoadDataGrid();

}

void LoadDataGrid()

{

List<Sanatoriy> goods = SanatoriyBDEntities.GetContext().Sanatoriys.OrderBy(p => p.Title).ToList();

DataGridGood.ItemsSource = goods;

\_itemcount = goods.Count;

TextBlockCount.Text = $" Результат запроса: {goods.Count} записей из {goods.Count}";

}

void LoadComboBoxItems()

{

var categories = SanatoriyBDEntities.GetContext().Categories.OrderBy(p => p.Title).ToList();

categories.Insert(0, new Category

{

Title = "Все виды"

}

);

ComboCategory.ItemsSource = categories;

ComboCategory.SelectedIndex = 0;

}

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

{

Manager.MainFrame.Navigate(new CategoryPage());

}

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

{

// Manager.MainFrame.Navigate(new AgesPage());

}

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

{

// Manager.MainFrame.Navigate(new OrganizersPage());

}

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

{

Manager.MainFrame.Navigate(new AddSanatoriyPage(null));

}

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

{

Manager.MainFrame.Navigate(new AddSanatoriyPage((sender as Button).DataContext as Sanatoriy));

}

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

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

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

DataGridGood.ItemsSource = null;

LoadDataGrid();

}

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

//var currentGoods = DataBDEntities.GetContext().Abonements.OrderBy(p => p.CategoryTrainer.Trainer.LastName).ToList();

var currentData = SanatoriyBDEntities.GetContext().Sanatoriys.OrderBy(p => p.Title).ToList();

// выбор только тех товаров, которые принадлежат данному производителю

if (ComboCategory.SelectedIndex > 0)

{

int catId = Convert.ToInt32((ComboCategory.SelectedItem as Category).Id);

List<Sanatoriy> quests = new List<Sanatoriy>();

foreach (Sanatoriy quest in currentData)

{

List<SanatoriyCategory> questCategories = quest.SanatoriyCategories.ToList();

if (questCategories.Any(elem => elem.CategoryId == catId))

quests.Add(quest);

}

currentData = quests;

}

// выбор тех товаров, в названии которых есть поисковая строка

currentData = currentData.Where(p => p.Title.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentData = currentData.OrderBy(p => p.Title).ToList();

if (ComboSort.SelectedIndex == 1)

currentData = currentData.OrderByDescending(p => p.Title).ToList();

if (ComboSort.SelectedIndex == 2)

currentData = currentData.OrderBy(p => p.GetRate).ToList();

if (ComboSort.SelectedIndex == 3)

currentData = currentData.OrderByDescending(p => p.GetRate).ToList();

// сортировка по убыванию цены

}

// В качестве источника данных присваиваем список данных

DataGridGood.ItemsSource = currentData;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentData.Count} записей из {\_itemcount}";

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

UpdateData();

}

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

{

Manager.MainFrame.Navigate(new TimeTablePage((sender as Button).DataContext as Sanatoriy));

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetPrevPhoto = 1;

//MessageBox.Show(sanatoriy.GetCurrentPhoto);

Image image = (sender as Button).Tag as Image;

image.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetNextPhoto = 1;

Image image = (sender as Button).Tag as Image;

try

{

image.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

catch (Exception ex)

{

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

}

}

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

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

Sanatoriy selected = (sender as Button).DataContext as Sanatoriy;

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить информацию о санатории, фотографии и ???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

//// проверка, есть ли у товара в таблице о продажах связанные записи

//// если да, то выбрасывается исключение и удаление прерывается

if (selected.Rewiews.Count > 0 || selected.TimeSheets.Count > 0)

throw new Exception("Есть зависимые записи о санаториях");

SanatoriyBDEntities.GetContext().SanatoriyCategories.RemoveRange(selected.SanatoriyCategories);

SanatoriyBDEntities.GetContext().Photos.RemoveRange(selected.Photos);

SanatoriyBDEntities.GetContext().Sanatoriys.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadDataGrid();

UpdateData();

}

catch (Exception ex)

{

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

}

}

}

}

}

Страница Расписания

Код интерфейса <Page x:Class="SanatoriyApp.Pages.TimeTablePage"

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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

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

mc:Ignorable="d"

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

Title="Расписание" >

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="300"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" Grid.ColumnSpan="2">

<TextBlock Text="Выберите санаторий" Style="{StaticResource MaterialDesignSubtitle2TextBlock}" />

<ComboBox HorizontalAlignment="Stretch" x:Name="ComboSanatoriy" Width="400"

SelectionChanged="ComboGoodsSelectionChanged"

SelectedValuePath="Id"

DisplayMemberPath="Title"/>

</StackPanel>

<materialDesign:Card Width="300" Height="400" x:Name="CardSanatoriy" Grid.Row="1" Background="{StaticResource MaterialDesignLightBackground}" >

<Grid materialDesign:ColorZoneAssist.Mode="SecondaryMid">

<Grid.RowDefinitions>

<RowDefinition Height="auto" />

<RowDefinition Height="60" />

<RowDefinition Height="60" />

<RowDefinition Height="100" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

</Grid.RowDefinitions>

<Image x:Name="ImagePhoto" Source="{Binding Path=GetCurrentPhoto}" Grid.Row="0" Height="200" Width="300"

Stretch="Uniform" />

<Button Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left" Style="{DynamicResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowLeft, Size=24}" Click="Button\_Click\_1" Foreground="#FF0C849E"

/>

<Button Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right" Style="{DynamicResource MaterialDesignFlatButton}"

Content="{materialDesign:PackIcon Kind=ArrowRight, Size=24}" Click="Button\_Click\_2" Foreground="#FF0C849E"

/>

<TextBlock Grid.Row="1" Text="{Binding Title}" HorizontalAlignment="Left" Foreground="White"

FontWeight="Bold" Margin="10"

FontSize="22"/>

<Button Grid.Row="2" HorizontalAlignment="Left" Style="{DynamicResource MaterialDesignFlatButton}">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Рейтинг по отзывам: " Style="{StaticResource MaterialDesignSubtitle2TextBlock}" Foreground="#FF0C849E"/>

<materialDesign:RatingBar x:Name="RatingBarRate" Margin="10 0 0 0" IsReadOnly="True"

Value="{Binding GetRate}" Foreground="#FF0C849E"/>

</StackPanel>

</Button>

<TextBlock Grid.Row="3" Text="{Binding GetInfo}" TextWrapping="Wrap" Foreground="White" Padding="5" FontSize="14"/>

</Grid>

</materialDesign:Card>

<DataGrid x:Name="DtData" AutoGenerateColumns="False" Grid.Row="1" Grid.Column="1" IsReadOnly="True"

SelectionMode="Single " LoadingRow="DataGridGoodLoadingRow">

<DataGrid.Columns>

<DataGridTextColumn Header="№ п" Binding="{Binding Id}"/>

<DataGridTextColumn Header="Дата" Binding="{Binding Date, StringFormat='d', ConverterCulture='ru-RU'}" Width="\*"/>

<DataGridTextColumn Header="Время" Binding="{Binding Time, StringFormat='t'}" Width="\*"/>

<DataGridTextColumn Header="Стоимость" Binding="{Binding Price, StringFormat='c', ConverterCulture='ru-RU'}"/>

<DataGridTemplateColumn Width="auto" Header="Редактировать">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit}" Click="BtnEdit\_Click" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Width="auto" Header="Удалить">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnDelete" Content="{materialDesign:PackIcon Kind=Delete}" Click="BtnDeleteClick" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="2" Grid.Column="1" Orientation="Horizontal" VerticalAlignment="Center" HorizontalAlignment="Center">

<Button x:Name="btnAdd" Content="Добавить" Click="btnAdd\_Click" Background="#FF0C849E"/>

</StackPanel>

</Grid>

</Page>

Код программый

using SanatoriyApp.Models;

using SanatoriyApp.Windows;

using System;

using System.Collections.Generic;

using System.Data.Entity;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для TimeTablePage.xaml

/// </summary>

public partial class TimeTablePage : Page

{

List<TimeSheet> timeSheets = new List<TimeSheet>();

public TimeTablePage(Sanatoriy sanatoriy)

{

InitializeComponent();

LoadData(sanatoriy);

}

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

// загрузка данных в DataGrid и ComboBox

void LoadData(Sanatoriy sanatoriy)

{

timeSheets = SanatoriyBDEntities.GetContext().TimeSheets.Where(p => p.SanatoriyId == sanatoriy.Id).OrderBy(p => p.Date).ThenBy(p => p.Time).ToList();

DtData.ItemsSource = timeSheets;

ComboSanatoriy.ItemsSource = SanatoriyBDEntities.GetContext().Sanatoriys.OrderBy(p => p.Title).ToList(); ;

ComboSanatoriy.SelectedIndex = 0;

ComboSanatoriy.SelectedValue = sanatoriy.Id;

ComboSanatoriy.DataContext = sanatoriy;

}

// фильтрация продаж по товару

private void ComboGoodsSelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (ComboSanatoriy.SelectedIndex >= 0)

{

int sanatoriyId = Convert.ToInt32(ComboSanatoriy.SelectedValue);

timeSheets = SanatoriyBDEntities.GetContext().TimeSheets.Where(p => p.SanatoriyId == sanatoriyId).OrderBy(p => p.Date).ThenBy(p => p.Time).ToList();

DtData.ItemsSource = timeSheets;

CardSanatoriy.DataContext = ComboSanatoriy.SelectedItem;

Sanatoriy sanatoriy = (ComboSanatoriy.SelectedItem) as Sanatoriy;

ImagePhoto.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

}

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

{

try

{

Sanatoriy g = ComboSanatoriy.SelectedItem as Sanatoriy;

TimeTableWindow window = new TimeTableWindow(new TimeSheet(), g);

if (window.ShowDialog() == true)

{

SanatoriyBDEntities.GetContext().TimeSheets.Add(window.currentItem);

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Запись добавлена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

LoadData(g);

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

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

{

try

{

TimeSheet selected = (sender as Button).DataContext as TimeSheet;

Sanatoriy g = ComboSanatoriy.SelectedItem as Sanatoriy;

//// получаем выделенный объект

TimeTableWindow window = new TimeTableWindow(selected, g);

if (window.ShowDialog() == true)

{

selected = SanatoriyBDEntities.GetContext().TimeSheets.Find(window.currentItem.Id);

// получаем измененный объект

if (selected != null)

{

SanatoriyBDEntities.GetContext().Entry(selected).State = EntityState.Modified;

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Запись изменена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

LoadData(g);

}

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetPrevPhoto = 1;

ImagePhoto.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetNextPhoto = 1;

ImagePhoto.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

try

{

// удаление выбранного санаторияиз таблицы

//получаем все выделенные санатории

TimeSheet selected = (sender as Button).DataContext as TimeSheet;

Sanatoriy g = ComboSanatoriy.SelectedItem as Sanatoriy;

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись? ", "Удаление", MessageBoxButton.OKCancel,

MessageBoxImage.Question);

if (messageBoxResult == MessageBoxResult.OK)

{

if (selected.Bookings.Count > 0)

{

MessageBox.Show("Ошибка удаления, есть связанные записи", "Error",

MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

SanatoriyBDEntities.GetContext().TimeSheets.Remove(selected);

SanatoriyBDEntities.GetContext().SaveChanges();

LoadData(g);

MessageBox.Show("Запись удалена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

Страница Пользователь

Код интерфейса

using SanatoriyApp.Models;

using SanatoriyApp.Windows;

using System;

using System.Collections.Generic;

using System.Data.Entity;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для TimeTablePage.xaml

/// </summary>

public partial class TimeTablePage : Page

{

List<TimeSheet> timeSheets = new List<TimeSheet>();

public TimeTablePage(Sanatoriy sanatoriy)

{

InitializeComponent();

LoadData(sanatoriy);

}

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

// загрузка данных в DataGrid и ComboBox

void LoadData(Sanatoriy sanatoriy)

{

timeSheets = SanatoriyBDEntities.GetContext().TimeSheets.Where(p => p.SanatoriyId == sanatoriy.Id).OrderBy(p => p.Date).ThenBy(p => p.Time).ToList();

DtData.ItemsSource = timeSheets;

ComboSanatoriy.ItemsSource = SanatoriyBDEntities.GetContext().Sanatoriys.OrderBy(p => p.Title).ToList(); ;

ComboSanatoriy.SelectedIndex = 0;

ComboSanatoriy.SelectedValue = sanatoriy.Id;

ComboSanatoriy.DataContext = sanatoriy;

}

// фильтрация продаж по товару

private void ComboGoodsSelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (ComboSanatoriy.SelectedIndex >= 0)

{

int sanatoriyId = Convert.ToInt32(ComboSanatoriy.SelectedValue);

timeSheets = SanatoriyBDEntities.GetContext().TimeSheets.Where(p => p.SanatoriyId == sanatoriyId).OrderBy(p => p.Date).ThenBy(p => p.Time).ToList();

DtData.ItemsSource = timeSheets;

CardSanatoriy.DataContext = ComboSanatoriy.SelectedItem;

Sanatoriy sanatoriy = (ComboSanatoriy.SelectedItem) as Sanatoriy;

ImagePhoto.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

}

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

{

try

{

Sanatoriy g = ComboSanatoriy.SelectedItem as Sanatoriy;

TimeTableWindow window = new TimeTableWindow(new TimeSheet(), g);

if (window.ShowDialog() == true)

{

SanatoriyBDEntities.GetContext().TimeSheets.Add(window.currentItem);

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Запись добавлена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

LoadData(g);

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

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

{

try

{

TimeSheet selected = (sender as Button).DataContext as TimeSheet;

Sanatoriy g = ComboSanatoriy.SelectedItem as Sanatoriy;

//// получаем выделенный объект

TimeTableWindow window = new TimeTableWindow(selected, g);

if (window.ShowDialog() == true)

{

selected = SanatoriyBDEntities.GetContext().TimeSheets.Find(window.currentItem.Id);

// получаем измененный объект

if (selected != null)

{

SanatoriyBDEntities.GetContext().Entry(selected).State = EntityState.Modified;

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Запись изменена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

LoadData(g);

}

}

}

catch

{

MessageBox.Show("Ошибка");

}

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetPrevPhoto = 1;

ImagePhoto.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

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

{

Sanatoriy sanatoriy = (sender as Button).DataContext as Sanatoriy;

sanatoriy.SetNextPhoto = 1;

ImagePhoto.Source = new BitmapImage(new Uri(sanatoriy.GetCurrentPhoto));

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

try

{

// удаление выбранного санаторияиз таблицы

//получаем все выделенные санатории

TimeSheet selected = (sender as Button).DataContext as TimeSheet;

Sanatoriy g = ComboSanatoriy.SelectedItem as Sanatoriy;

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись? ", "Удаление", MessageBoxButton.OKCancel,

MessageBoxImage.Question);

if (messageBoxResult == MessageBoxResult.OK)

{

if (selected.Bookings.Count > 0)

{

MessageBox.Show("Ошибка удаления, есть связанные записи", "Error",

MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

SanatoriyBDEntities.GetContext().TimeSheets.Remove(selected);

SanatoriyBDEntities.GetContext().SaveChanges();

LoadData(g);

MessageBox.Show("Запись удалена", "Внимание", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

using SanatoriyApp.Models;

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 SanatoriyApp.Pages

{

/// <summary>

/// Логика взаимодействия для UsersPage.xaml

/// </summary>

public partial class UsersPage : Page

{

int \_itemcount = 0;

List<User> data;

public UsersPage()

{

InitializeComponent();

LoadData();

}

private void ButtonClick(object sender, RoutedEventArgs e)

{

// открытие редактирования товара

// передача выбранного товара в AddGoodPage

}

void LoadData()

{

DataGridData.ItemsSource = null;

//загрузка обновленных данных

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

data = SanatoriyBDEntities.GetContext().Users.OrderBy(p => p.LastName).ThenBy(p => p.FirstName).ToList();

DataGridData.ItemsSource = data;

TextBlockCount.Text = $" Результат запроса: {\_itemcount} записей из {\_itemcount}";

var roles = SanatoriyBDEntities.GetContext().Roles.OrderBy(p => p.Title).ToList();

roles.Insert(0, new Role

{

Title = "Все типы"

}

);

ComboUserType.ItemsSource = roles;

ComboUserType.SelectedIndex = 0;

\_itemcount = data.Count;

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

//событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{

LoadData();

}

}

private void BtnAddClick(object sender, RoutedEventArgs e)

{

// открытие AddGoodPage для добавления новой записи

Manager.MainFrame.Navigate(new AddUserPage(null));

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{

// удаление выбранного товара из таблицы

//получаем все выделенные товары

var selected = (sender as Button).DataContext as User;

int k = SanatoriyBDEntities.GetContext().Users.Count(p => p.RoleId == 1);

if (selected.RoleId == 1 && k == 1)

{

MessageBox.Show("Ошибка удаления, невозможно удалить последнего администратора", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить запись???",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{

try

{

// берем из списка удаляемых товаров один элемент

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

if (selected.Bookings.Count > 0 || selected.Rewiews.Count > 0)

throw new Exception("Ошибка удаления, есть связанные записи");

SanatoriyBDEntities.GetContext().Users.Remove(selected);

//сохраняем изменения

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

LoadData();

}

catch (Exception ex)

{

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

}

}

}

// отображение номеров строк в DataGrid

private void DataGridDataLoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{

UpdateData();

}

// Поиск товаров конкретного производителя

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{

// получаем текущие данные из бд

//var currentGoods = PhotoStudioSeverBDEntities.GetContext().Abonements.OrderBy(p => p.CategoryTrainer.Trainer.LastName).ToList();

var currentData = SanatoriyBDEntities.GetContext().Users.OrderBy(p => p.LastName).ThenBy(p => p.FirstName).ToList();

// выбор только тех товаров, которые принадлежат данному производителю

if (ComboUserType.SelectedIndex > 0)

{

currentData = currentData.Where(p => p.RoleId == ((ComboUserType.SelectedItem) as Role).RoleId).ToList();

}

// выбор тех товаров, в названии которых есть поисковая строка

currentData = currentData.Where(p => p.UserName.ToLower().Contains(TBoxSearch.Text.ToLower()) ||

(p.LastName != null && p.LastName.ToLower().Contains(TBoxSearch.Text.ToLower()))

|| (p.FirstName != null && p.FirstName.ToLower().Contains(TBoxSearch.Text.ToLower()))

|| (p.MiddleName != null && p.MiddleName.ToLower().Contains(TBoxSearch.Text.ToLower()))

).ToList();

if (ComboSort.SelectedIndex >= 0)

{

// сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentData = currentData.OrderBy(p => p.LastName).ThenBy(p => p.FirstName).ThenBy(p => p.MiddleName).ToList();

if (ComboSort.SelectedIndex == 1)

currentData = currentData.OrderByDescending(p => p.LastName).ThenByDescending(p => p.FirstName).ThenByDescending(p => p.MiddleName).ToList();

}

// В качестве источника данных присваиваем список данных

DataGridData.ItemsSource = currentData;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentData.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{

UpdateData();

}

private void DataGridData\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

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

{

Manager.MainFrame.Navigate(new AddUserPage((sender as Button).DataContext as User));

}

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

{

UpdateData();

}

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

{

Manager.MainFrame.Navigate(new MyRewiewPage());

}

}

}

Окно Промотра

Код интерфейса

<Window x:Class="SanatoriyApp.Windows.BookingWindow"

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:SanatoriyApp.Windows"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

Title="Забронировать" Background="{DynamicResource MaterialDesignPaper}"

Width="600"

Height="440" WindowStyle="ToolWindow" MinHeight="440" MinWidth="600" WindowStartupLocation="CenterScreen"

FontFamily="{DynamicResource MaterialDesignFont}" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="44" />

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock x:Name="TextBlockServiceName" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Stretch" Grid.Row="0" Height="30" Margin="0,7"

Style="{DynamicResource MaterialDesignBody1TextBlock}" />

<TextBox x:Name="TbTitle" materialDesign:HintAssist.Hint="Как к Вам обращаться?" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="1" Height="30" Margin="0,7" />

<xctk:MaskedTextBox x:Name="TbPhone"

Mask="+0 (000) 000-00-00" Style="{DynamicResource MaterialDesignTextBox}"

Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="2" Height="30" Margin="0,7" />

<ComboBox x:Name="ComboBoxTimeSheet" SelectedItem="{Binding TimeSheet}" VerticalContentAlignment="Center"

materialDesign:HintAssist.Hint="Выберите дату" Grid.Row="3" Grid.Column="1" DisplayMemberPath="GetFullDateTime" Padding="10 0 0 0"

VerticalAlignment="Stretch"

SelectedValuePath="Id" SelectionChanged="ComboBoxTimeSheet\_SelectionChanged" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="4" Grid.ColumnSpan="2" >

<Button x:Name="BtnOk" IsDefault="True" MinWidth="60" Margin="5" Click="BtnOk\_Click" Background="#FF0C849E">OK</Button>

<Button x:Name="BtnCancel" IsCancel="True" MinWidth="60" Margin="5" Background="#FF0C849E">Отмена</Button>

<Button x:Name="BtnToPDF" MinWidth="60" Margin="5" Visibility="Collapsed" >Сохранить в PDF</Button>

</StackPanel>

<TextBlock Grid.Row ="5" x:Name="TextBlockPrice" HorizontalAlignment="Center" VerticalAlignment="Center" Style="{StaticResource MaterialDesignHeadline6TextBlock}"/>

<!--<CheckBox x:Name="CheckBoxAdmin" Content="Администратор" IsChecked="{Binding IsAdmin}" Visibility="{Binding GetVisibilityItem}" Grid.Row="11" Grid.Column="0" Margin="20 0"/>-->

</Grid>

</Window>

Код программный

using SanatoriyApp.Models;

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.Shapes;

namespace SanatoriyApp.Windows

{

/// <summary>

/// Логика взаимодействия для BookingWindow.xaml

/// </summary>

public partial class BookingWindow : Window

{

Sanatoriy currentSanatoriy = new Sanatoriy();

public BookingWindow(Sanatoriy quest)

{

InitializeComponent();

currentSanatoriy = quest;

ComboBoxTimeSheet.ItemsSource = SanatoriyBDEntities.GetContext().TimeSheets.Where(p => p.SanatoriyId == quest.Id && p.Bookings.Count == 0).ToList();

if (Manager.CurrentUser != null)

{

TbPhone.Text = Manager.CurrentUser.Phone;

TbTitle.Text = Manager.CurrentUser.UserName;

}

}

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(TbTitle.Text))

s.AppendLine("Поле имя пустое");

if (!TbPhone.IsMaskFull)

s.AppendLine("Укажите телефон");

if (ComboBoxTimeSheet.SelectedIndex == -1)

s.AppendLine("Выберите дату и время");

return s;

}

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

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

try

{

Booking booking = new Booking();

booking.TimeSheetId = Convert.ToInt32(ComboBoxTimeSheet.SelectedValue);

booking.UserInfo = TbTitle.Text;

booking.Phone = TbPhone.Text;

booking.Payed = false;

if (Manager.CurrentUser != null)

booking.UserName = Manager.CurrentUser.UserName;

SanatoriyBDEntities.GetContext().Bookings.Add(booking);

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Забронировано");

DialogResult = true;

}

catch (Exception ex)

{

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

}

}

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

{

if (ComboBoxTimeSheet.SelectedIndex == -1)

return;

int id = Convert.ToInt32(ComboBoxTimeSheet.SelectedValue);

TimeSheet timeSheet = SanatoriyBDEntities.GetContext().TimeSheets.FirstOrDefault(p => p.Id == id);

double price = 0;

if (timeSheet != null)

price = timeSheet.Price;

TextBlockPrice.Text = price.ToString("c");

}

}

}

Окно Категория

Код интерфейса

using SanatoriyApp.Models;

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.Shapes;

namespace SanatoriyApp.Windows

{

/// <summary>

/// Логика взаимодействия для CategoryWindow.xaml

/// </summary>

public partial class CategoryWindow : Window

{

public Category currentItem { get; private set; }

public CategoryWindow(Category p)

{

InitializeComponent();

currentItem = p;

DataContext = currentItem;

}

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

if (TbName.Text == "")

s.AppendLine("Укажите название");

return s;

}

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

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

currentItem.Title = TbName.Text;

this.DialogResult = true;

}

}

}

Окно Логин

Код интерфейса

<Window x:Class="SanatoriyApp.Windows.LoginWindow"

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:SanatoriyApp.Windows"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular"

TextElement.FontSize="13"

TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}" Title="Вход" Height="250"

Width="400" MinHeight="250" MinWidth="400"

WindowStartupLocation="CenterScreen" ResizeMode="NoResize">

<!--NoResize - нужен, чтобы нельзя было изменить зааднные размеры окна-->

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition/>

<RowDefinition Height="20"/>

</Grid.RowDefinitions>

<Grid />

<materialDesign:PackIcon Kind="Key" Foreground="{DynamicResource {x:Static SystemColors.ActiveCaptionBrushKey}}"

Margin="10,10,0,0"

HorizontalAlignment="Left" Height="55" Width="62"/>

<TextBlock Grid.Row="0" Text="Санатории"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="16" TextAlignment="Center"

FontWeight="Bold" Height="60" Width="281" />

<TextBox x:Name="TbLogin" Grid.Row="1"

Grid.Column="1" materialDesign:HintAssist.Hint="Логин"

Width="300" Text="admin"/>

<PasswordBox x:Name="TbPass" Height="30" Width="300" materialDesign:HintAssist.Hint="Пароль" Password="1" Grid.Row="2" />

<StackPanel Orientation="Horizontal" Grid.Row="3" HorizontalAlignment="Center">

<Button Width="100" Height="30"

Content="OK"

x:Name="BtnOk"

Click="BtnOkClick" IsDefault="True" Background="#FF0C849E" BorderBrush="#FF0C849E"/>

<Button Width="100"

Height="30"

Content="Cancel"

x:Name="BtnCancel" IsCancel="True"

Click="BtnCancelClick" Background="#FF0C849E" BorderBrush="#FF0C849E"/>

</StackPanel>

<Button Grid.Row="4" Content="Регистрация" Padding="0" FontSize="10" Height="20" Click="Button\_Click" Background="#FF0C849E"/>

<materialDesign:DialogHost x:Name="DialogHost"

BorderThickness="1">

<materialDesign:DialogHost.DialogContent>

<StackPanel>

<Grid VerticalAlignment="Top" Margin="35, 60" Height="180" Width="420">

<Label Content="Регистрация " HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0 0 0 20" Foreground="Gray" FontSize="18"/>

<StackPanel VerticalAlignment="Center" Margin="15">

<TextBox x:Name="tbRegLogin" Margin="0, 10" TextAlignment="Left" materialDesign:HintAssist.Hint="ЛОГИН" FontSize="18"/>

<PasswordBox x:Name="psbPassword1" Margin="0,10" materialDesign:HintAssist.Hint="ПАРОЛЬ" materialDesign:TextFieldAssist.HasClearButton="True" FontSize="18"/>

<PasswordBox x:Name="psbPassword2" Margin="0,10" materialDesign:HintAssist.Hint="ПАРОЛЬ" materialDesign:TextFieldAssist.HasClearButton="True" FontSize="18"/>

</StackPanel>

</Grid>

<StackPanel HorizontalAlignment="Center" Margin="10 20">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<Button x:Name="btnOk" Click="btnOk\_Click">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<materialDesign:PackIcon Kind="Check" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Ok" VerticalAlignment="Center"/>

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<materialDesign:PackIcon Kind="Cancel" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Cancel" VerticalAlignment="Center"/>

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Window>

Код программный

using SanatoriyApp.Models;

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.Shapes;

namespace SanatoriyApp.Windows

{

/// <summary>

/// Логика взаимодействия для LoginWindow.xaml

/// </summary>

public partial class LoginWindow : Window

{

public LoginWindow()

{

InitializeComponent();

}

private void BtnOkClick(object sender, RoutedEventArgs e)

{

try

{ //загрузка всех пользователей из БД в список

List<User> users = SanatoriyBDEntities.GetContext().Users.ToList();

//попытка найти пользователя с указанным паролем и логином

//если такого пользователя не будет обнаружено то переменная u будет равна null

User u = users.FirstOrDefault(p => p.Password == TbPass.Password && p.UserName == TbLogin.Text);

if (u != null)

{

// логин и пароль корректные запускаем главную форму приложения

//MainWindow mainWindow = new MainWindow();

//mainWindow.Owner = this;

//this.Hide();

//mainWindow.Show();

Manager.CurrentUser = u;

this.DialogResult = true;

}

else

{

MessageBox.Show("Не верный логин или пароль");

}

}

catch (Exception ex)

{

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

}

}

//код кнопки Cancel

private void BtnCancelClick(object sender, RoutedEventArgs e)

{

this.DialogResult = false;

}

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

{

DialogHost.IsOpen = true;

}

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

{

if ((tbRegLogin.Text == "") || (psbPassword1.Password == "") || (psbPassword2.Password == ""))

{

MessageBox.Show("Поля пустые", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

List<User> users = SanatoriyBDEntities.GetContext().Users.ToList();

//попытка найти пользователя с указанным паролем и логином

//если такого пользователя не будет обнаружено то переменная u будет равна null

User u = users.FirstOrDefault(p => p.UserName == tbRegLogin.Text);

if (u != null)

{

MessageBox.Show("Данный логин занят, выберите другой логин", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (psbPassword1.Password != psbPassword2.Password)

{

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

return;

}

User user = new User();

user.UserName = tbRegLogin.Text;

user.Password = psbPassword1.Password;

user.RoleId = 3;

user.FirstName = "";

user.LastName = "";

user.MiddleName = "";

user.Phone = "";

user.Email = "";

SanatoriyBDEntities.GetContext().Users.Add(user);

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Регистраця прошла успешно", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

DialogHost.IsOpen = false;

}

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

{

DialogHost.IsOpen = false;

}

}

}

Окно Сделать отзыв

Код интерфейс

<Window x:Class="SanatoriyApp.Windows.MakeRewiewWindow"

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:SanatoriyApp.Windows"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

Title="Отзыв"

WindowStartupLocation="CenterScreen" Height="600" Width="800" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition Height="40" />

<RowDefinition Height="200" />

<RowDefinition Height="40" />

<RowDefinition Height="80"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="160" />

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<TextBox x:Name="TextBoxSanatoriyTitle" Grid.Column="0" VerticalAlignment="Center" Grid.Row="0" Height="30" Margin="0,5,-0.4,5" IsReadOnly="True" Grid.ColumnSpan="2" />

<TextBox x:Name="TbUserName" Text="{Binding UserName}" Grid.Column="0" VerticalAlignment="Center" Grid.Row="1" Height="30" Margin="0,5,-0.4,5" IsReadOnly="True" Grid.ColumnSpan="2" />

<TextBox x:Name="TbTitle" Text="{Binding Title}" materialDesign:HintAssist.Hint="Отзыв"

Padding="20 5 0 0" Grid.Column="0"

VerticalAlignment="Stretch" TextWrapping="Wrap" Grid.Row="2" Margin="0,7,-0.4,7" Grid.ColumnSpan="2" />

<TextBlock Grid.Row="3" Margin="10 0 0 0" Text="Рейтинг"/>

<materialDesign:RatingBar x:Name="RatingBarRate" Grid.Row="3" IsPreviewValueEnabled="True"

Max="5" Min="1" ValueIncrements="1"

Value="{Binding Rate}" Grid.Column="1" Foreground="#FF0C849E" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="6" Margin="160.8,0,303.6,0" Width="162" Grid.Column="1">

<Button IsDefault="True" Click="Button\_Click" MinWidth="60" Margin="5" BorderBrush="#FF0C849E" Background="#FF0C849E">OK</Button>

<Button IsCancel="True" MinWidth="60" Margin="5" BorderBrush="#FF0C849E" Background="#FF0C849E">Отмена</Button>

</StackPanel>

</Grid>

</Window>

Код программный

using SanatoriyApp.Models;

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.Shapes;

namespace SanatoriyApp.Windows

{

/// <summary>

/// Логика взаимодействия для MakeRewiewWindow.xaml

/// </summary>

public partial class MakeRewiewWindow : Window

{

public Rewiew currentRewiew { get; set; }

Sanatoriy currentSanatoriy = new Sanatoriy();

public MakeRewiewWindow(Rewiew rewiew, Sanatoriy sanatoriy)

{

InitializeComponent();

currentRewiew = rewiew;

if (sanatoriy != null)

{

currentSanatoriy = sanatoriy;

TextBoxSanatoriyTitle.Text = sanatoriy.Title;

}

TbUserName.Text = Manager.CurrentUser.UserName;

currentRewiew.UserName = Manager.CurrentUser.UserName;

DataContext = currentRewiew;

}

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(TbTitle.Text))

s.AppendLine("Отзыв пуст");

//if (!CheckDateAndTime())

// s.AppendLine("Укажите корректные дату и время");

return s;

}

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

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

currentRewiew.SanatoriyId = currentSanatoriy.Id;

currentRewiew.Title = TbTitle.Text;

currentRewiew.Rate = Convert.ToInt32(RatingBarRate.Value);

currentRewiew.UserName = Manager.CurrentUser.UserName;

try

{

if (currentRewiew.Id == 0)

{

SanatoriyBDEntities.GetContext().Rewiews.Add(currentRewiew);

}

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Отзыв сохранен");

DialogResult = true;

}

catch (Exception ex)

{

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

}

}

}

}

Окно мой аккаунт

Код интерфейса

<Window x:Class="SanatoriyApp.Windows.MyAccountWindow"

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:SanatoriyApp.Windows"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

Title="Мой профиль"

WindowStartupLocation="CenterScreen" Height="500" Width="800" >

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="40" />

<RowDefinition Height="80"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBox Text="{Binding UserName}" Grid.Column="0" VerticalAlignment="Center" Grid.Row="0" Height="30" Margin="0,7" IsReadOnly="True" />

<TextBox Text="{Binding LastName}" materialDesign:HintAssist.Hint="Фамилия" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="1" Height="30" Margin="0,7" />

<TextBox Text="{Binding FirstName}" materialDesign:HintAssist.Hint="Имя" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="2" Height="30" Margin="0,7" />

<TextBox Text="{Binding MiddleName}" materialDesign:HintAssist.Hint="Отчество" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="3" Height="30" Margin="0,7" />

<xctk:MaskedTextBox x:Name="TbPhone" Text="{Binding Phone}"

Mask="+0 (000) 000-00-00" Style="{DynamicResource MaterialDesignTextBox}"

Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="4" Height="30" Margin="0,7" />

<TextBox Text="{Binding Email}" materialDesign:HintAssist.Hint="E-mail" Padding="20 5 0 0" Grid.Column="0" VerticalAlignment="Center" Grid.Row="5" Height="30" Margin="0,7" />

<CheckBox x:Name="CheckBoxChangePassword" Content="Изменить пароль" Width="200" VerticalAlignment="Center" HorizontalAlignment="Left" Grid.Row="6" Margin="20 0 0 0" Grid.Column="0" Height="30" />

<PasswordBox Grid.Column="0" Margin="200 0 0 0" x:Name="PasswordBoxOldPassword" VerticalAlignment="Center" materialDesign:HintAssist.Hint="Введите старый пароль" Grid.Row="6" Height="30" />

<PasswordBox Grid.Column="0" Margin="200 0 0 0" VerticalAlignment="Center" x:Name="PasswordBoxNewPassword1" materialDesign:HintAssist.Hint="Введите новый пароль" Grid.Row="7" Height="30" />

<PasswordBox Grid.Column="0" Margin="200 0 0 0" VerticalAlignment="Center" x:Name="PasswordBoxNewPassword2" materialDesign:HintAssist.Hint="Введите новый пароль" Grid.Row="8" Height="30" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="9" Grid.ColumnSpan="2" Margin="127,0,103,0" Width="162">

<Button IsDefault="True" Click="Accept\_Click" MinWidth="60" Margin="5" Background="#FF0C849E" BorderBrush="#FF0C849E">OK</Button>

<Button IsCancel="True" MinWidth="60" Margin="5" Background="#FF0C849E" BorderBrush="#FF0C849E">Отмена</Button>

</StackPanel>

</Grid>

</Window>

Код программный

using SanatoriyApp.Models;

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.Shapes;

namespace SanatoriyApp.Windows

{

/// <summary>

/// Логика взаимодействия для MyAccountWindow.xaml

/// </summary>

public partial class MyAccountWindow : Window

{

public User currentItem { get; private set; }

public MyAccountWindow()

{

InitializeComponent();

currentItem = Manager.CurrentUser;

this.DataContext = Manager.CurrentUser;

}

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(currentItem.UserName))

s.AppendLine("Задайте имя пользователя");

if (string.IsNullOrWhiteSpace(currentItem.FirstName))

s.AppendLine("Поле имя пустое");

if (string.IsNullOrWhiteSpace(currentItem.LastName))

s.AppendLine("Поле фамилия пустое");

if (string.IsNullOrWhiteSpace(currentItem.Phone))

s.AppendLine("Поле телефон пустое");

if (string.IsNullOrWhiteSpace(currentItem.Email))

s.AppendLine("Поле email пустое");

if (CheckBoxChangePassword.IsChecked == true)

{

User user = SanatoriyBDEntities.GetContext().Users.Find(currentItem.UserName);

if ((PasswordBoxNewPassword1.Password != PasswordBoxNewPassword2.Password) || (PasswordBoxOldPassword.Password != user.Password))

{

s.AppendLine("Пароли не совпадают");

}

else

{

currentItem.Password = PasswordBoxNewPassword1.Password;

}

}

return s;

}

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

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

try

{

SanatoriyBDEntities.GetContext().SaveChanges();

MessageBox.Show("Запись изменена");

this.DialogResult = true;

}

catch

{

MessageBox.Show("Ошибка");

this.DialogResult = false;

}

}

}

}

Окно Выбора времени

Код интерфейса

<Window x:Class="SanatoriyApp.Windows.TimeTableWindow"

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:SanatoriyApp.Windows"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d"

Title="Расписание" Height="260" Width="800" WindowStartupLocation="CenterScreen" Loaded="Window\_Loaded\_1">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="60" />

<RowDefinition Height="40" />

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="187" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Center">

<DatePicker Width="200" x:Name="DatePickerDate"

materialDesign:HintAssist.Hint="Выберите дату"

Style="{StaticResource MaterialDesignOutlinedDatePicker}" />

<materialDesign:TimePicker x:Name="TimePickerTime" Width="200"

materialDesign:HintAssist.Hint="Выберите время"

Style="{StaticResource MaterialDesignOutlinedTimePicker}" Is24Hours="True" />

<TextBlock x:Name="TextBlockAnswer" VerticalAlignment="Center" Style="{DynamicResource MaterialDesignCaptionTextBlock}"/>

</StackPanel>

<TextBlock Grid.Row="1" Grid.Column="0" Text="Стоимость" Margin="20 0 0 0" VerticalAlignment="Center" />

<xctk:DoubleUpDown Grid.Row="1" Grid.Column="1" x:Name="DoubleUpDownPrice" FormatString="c" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="5" Grid.ColumnSpan="2" Margin="127,0,103,0" Width="162">

<Button IsDefault="True" MinWidth="60" Margin="5" Click="Button\_Click" Background="#FF0C849E" BorderBrush="#FF0C849E">OK</Button>

<Button IsCancel="True" MinWidth="60" Margin="5" Background="#FF0C849E" BorderBrush="#FF0C849E">Отмена</Button>

</StackPanel>

</Grid>

</Window>

Код программный

using SanatoriyApp.Models;

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.Shapes;

namespace SanatoriyApp.Windows

{

/// <summary>

/// Логика взаимодействия для TimeTableWindow.xaml

/// </summary>

public partial class TimeTableWindow : Window

{

public TimeSheet currentItem { get; private set; }

Sanatoriy sanatoriy;

public TimeTableWindow(TimeSheet p, Sanatoriy g)

{

InitializeComponent();

currentItem = p;

// currentItem.TrainerId = g.Id;

sanatoriy = g;

DataContext = currentItem;

}

private StringBuilder CheckFields()

{

StringBuilder s = new StringBuilder();

if (DatePickerDate.SelectedDate is null)

s.AppendLine("Не выбрана дата");

if (TimePickerTime.SelectedTime is null)

s.AppendLine("Укажите время");

return s;

}

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

{

StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{

MessageBox.Show(\_error.ToString());

return;

}

//currentItem.CategoryId = Convert.ToInt32(ComboCategory.SelectedValue);

currentItem.Date = Convert.ToDateTime(DatePickerDate.SelectedDate).Date;

currentItem.Time = Convert.ToDateTime(TimePickerTime.SelectedTime).TimeOfDay;

currentItem.Price = Convert.ToDouble(DoubleUpDownPrice.Value);

currentItem.SanatoriyId = sanatoriy.Id;

this.DialogResult = true;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

}

private void Window\_Loaded\_1(object sender, RoutedEventArgs e)

{

DoubleUpDownPrice.Value = currentItem.Price;

if (currentItem.Id == 0)

{

DatePickerDate.SelectedDate = DateTime.Today;

TimePickerTime.SelectedTime = DateTime.Today;

}

else

{

DatePickerDate.SelectedDate = Convert.ToDateTime(currentItem.Date);

TimePickerTime.SelectedTime = Convert.ToDateTime(currentItem.Time.ToString());

}

}

}

}

**Приложение B**

**Диск с исполняемым кодом программного модуля**