**Оглавление**

[Аннотация 4](#_Toc8739487)

[Введение 5](#_Toc8739488)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ 6](#_Toc8739489)

[1.1 Основное функциональное назначение программного средства 6](#_Toc8739490)

[1.2 Полное наименование программного средства 6](#_Toc8739491)

[1.3 Условное обозначение программного средства 6](#_Toc8739492)

[1.4 Разработчики программного средства 6](#_Toc8739493)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc8739494)

[2.1 Основание для разработки 7](#_Toc8739495)

[2.2 Назначение разработки 7](#_Toc8739496)

[2.3 Требование к программному средству 7](#_Toc8739497)

[2.4 Требования к программной документации 8](#_Toc8739498)

[2.5 Требования к эргономике и технической эстетике 8](#_Toc8739499)

[2.6 Стадии и этапы разработки 9](#_Toc8739500)

[2.7 Порядок контроля и приемки 10](#_Toc8739501)

[3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ 11](#_Toc8739502)

[3.1 Декомпозиция поставленной задачи 11](#_Toc8739503)

[3.2 Общая архитектура программного средства 12](#_Toc8739504)

[3.3 Разработка алгоритма решения задачи 15](#_Toc8739505)

[3.4 Реализация функционального назначения программного средства 16](#_Toc8739506)

[3.5 Структурная организация данных 16](#_Toc8739507)

[3.6 Разработка интерфейса ПС 22](#_Toc8739508)

[3.7 Описание структуры выходной информации 32](#_Toc8739509)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 33](#_Toc8739510)

[4.1 Назначение программного средства 33](#_Toc8739511)

[4.2 Условия выполнения программного средства 33](#_Toc8739512)

[4.3 Эксплуатация программного средства 33](#_Toc8739513)

[4.4 Сообщения пользователю 42](#_Toc8739514)

[5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА 43](#_Toc8739515)

[5.1 Определение верхней границы количества тестов 43](#_Toc8739516)

[5.2 Разработка тестовых вариантов 44](#_Toc8739517)

[5.3 Составление отчета о тестировании 51](#_Toc8739518)

[ЗАКЛЮЧЕНИЕ 52](#_Toc8739519)

[Оценка качества программного средства с помощью метрик 52](#_Toc8739520)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 55](#_Toc8739521)

[Приложение А Текст кода программного средства 57](#_Toc8739522)

[Приложение Б Спецификация 92](#_Toc8739523)

[Приложение B Диск с программным средством 94](#_Toc8739524)

**Аннотация**

На дипломную работу студента Ильина К.И. на тему: Фирма по продаже и обслуживанию кондиционеров. Дипломная работа представлена пояснительной запиской на 94 листа машинописного текста. Состоит из введения, 5 глав, заключения и списка литературных источников.

В главе 1 содержится общие сведения о программном средстве Фирма по продаже и обслуживанию кондиционеров. 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 – Фирма по продаже и обслуживанию кондиционеров, 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 дипломной работе была поставлена задача разработки модуля доступа к данным базы данных фирмы по продаже и обслуживанию кондиционеров, который позволяет автоматизировать деятельность фирмы.

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 дипломной работы.

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, в данном случае менеджер фирмы.

**1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ**

**1.1 Основное функциональное назначение программного средства**

Программа предназначена для автоматизации деятельности фирмы. 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 обработки данных Фирма по продаже и обслуживанию кондиционеров.

**1.3 Условное обозначение программного средства**

«AirSystem 1.0»

**1.4 Разработчики программного средства**

Программное средство разработал студент Зеленодольского Механического колледжа группы 207 Ильин К.И.

**2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**2.1 Основание для разработки**

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

**2.2 Назначение разработки**

Программа «AirSystem 1.0» работает под управление операционной системы Windows 7, Windows 8, Windows 10. Программа предназначена для ведения учёта заказов по обслуживанию и покупке кондиционеров. Цель создания программы – упростить сбор статистики. Программа позволяет вводить, изменять и удалять данные о каталоге товаров и услуг, создавать заявки.

**2.3 Требование к программному средству**

«AirSystem 1.0» 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 «AirSystem 1.0» является кроссплатформенным продуктом, что позволяет не ограничиваться в работе с определенной версией ОС.

«AirSystem 1.0» 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.

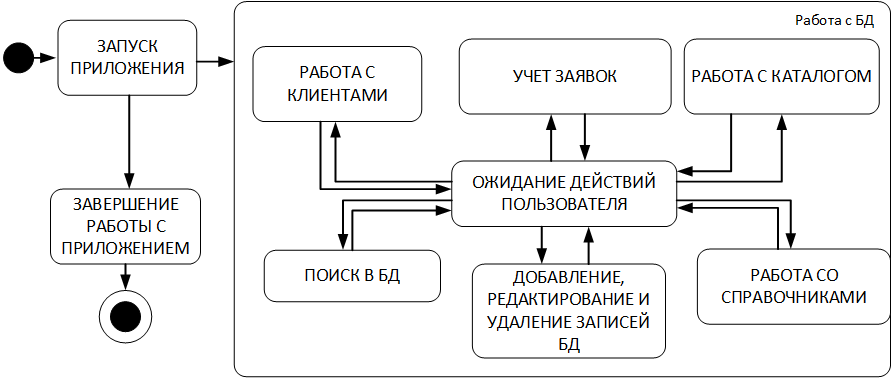


Рис.1 Декомпозиция поставленной задачи

**3.2 Общая архитектура программного средства**

Объектом моделирования является деятельность фирмы.

Рассматриваемые процессы: учёт заявок.

Объекты моделирования представлены на диаграмме классов, рис.2.

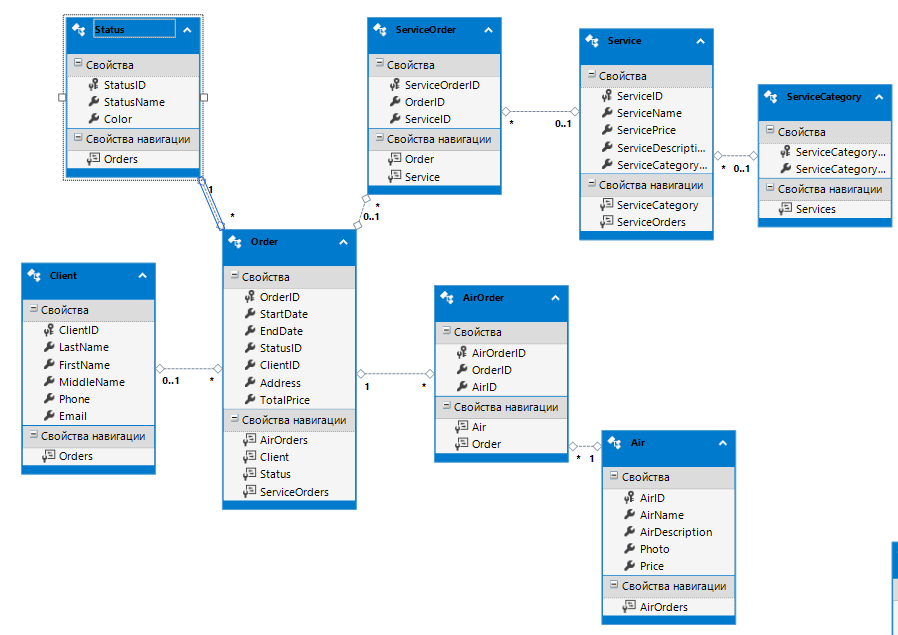


Рис.2 Диаграмма классов

По полученной декомпозиции задач была спроектирована следующая архитектура программного средства.

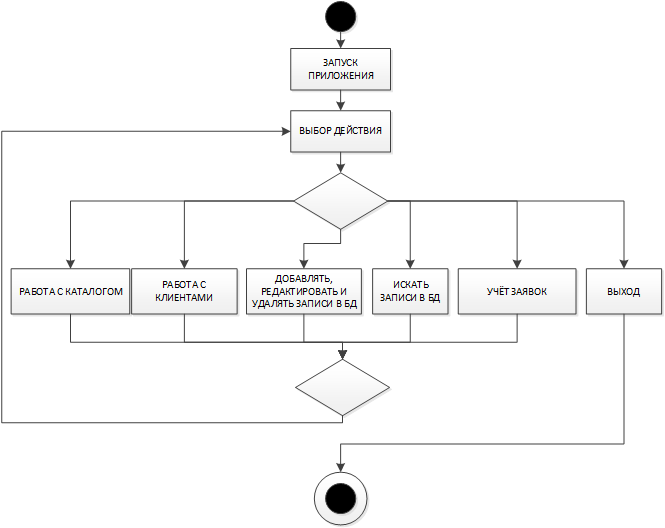


Рис.3 Схема событийно-ориентированной архитектуры

В «AirSystem 1.0» 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 в Excel и т.д. система осуществит выбранные действия, и база отреагирует соответствующим образом: запись добавлена, удалена, отредактирована, либо отчет выведен в Excel. Системная архитектура пользователя рассматривает это изменение состояния как событие, создаваемое, публикуемое, определяемое и потребляемое различными приложениями в составе архитектуры.

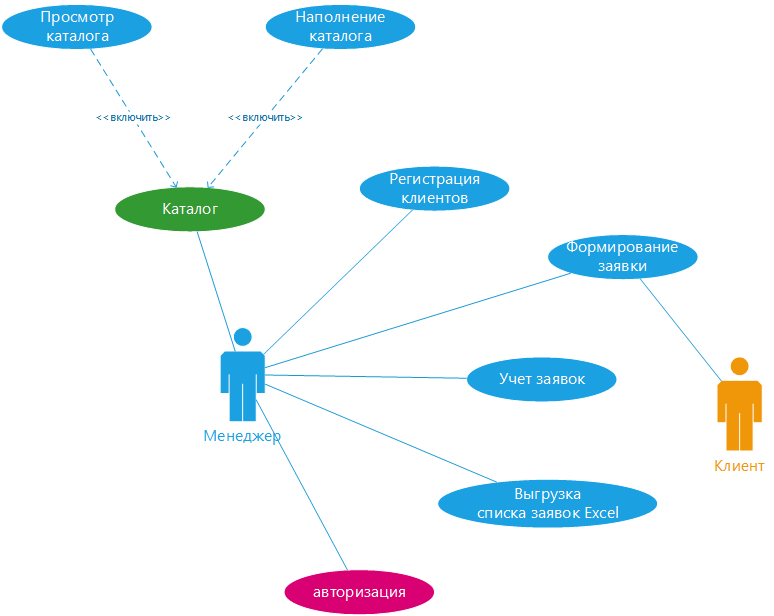


Рис.4 Диаграмма вариантов использования

**3.3 Разработка алгоритма решения задачи**

**Общий алгоритм**

****Рис.5 Алгоритм программы

**3.4 Реализация функционального назначения программного средства**

Программа «AirSystem 1.0» имеет следующий набор входных данных, такие как: кондиционеры, услуги, клиенты и заявки.

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

Выходными данными являются: учёт заявок.

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 Фирма по продаже и обслуживанию кондиционеров 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 «Мероприятия». Реляционная модель автоматизированной системы соответствует всем 12 правилам Кодда. (рис.6)

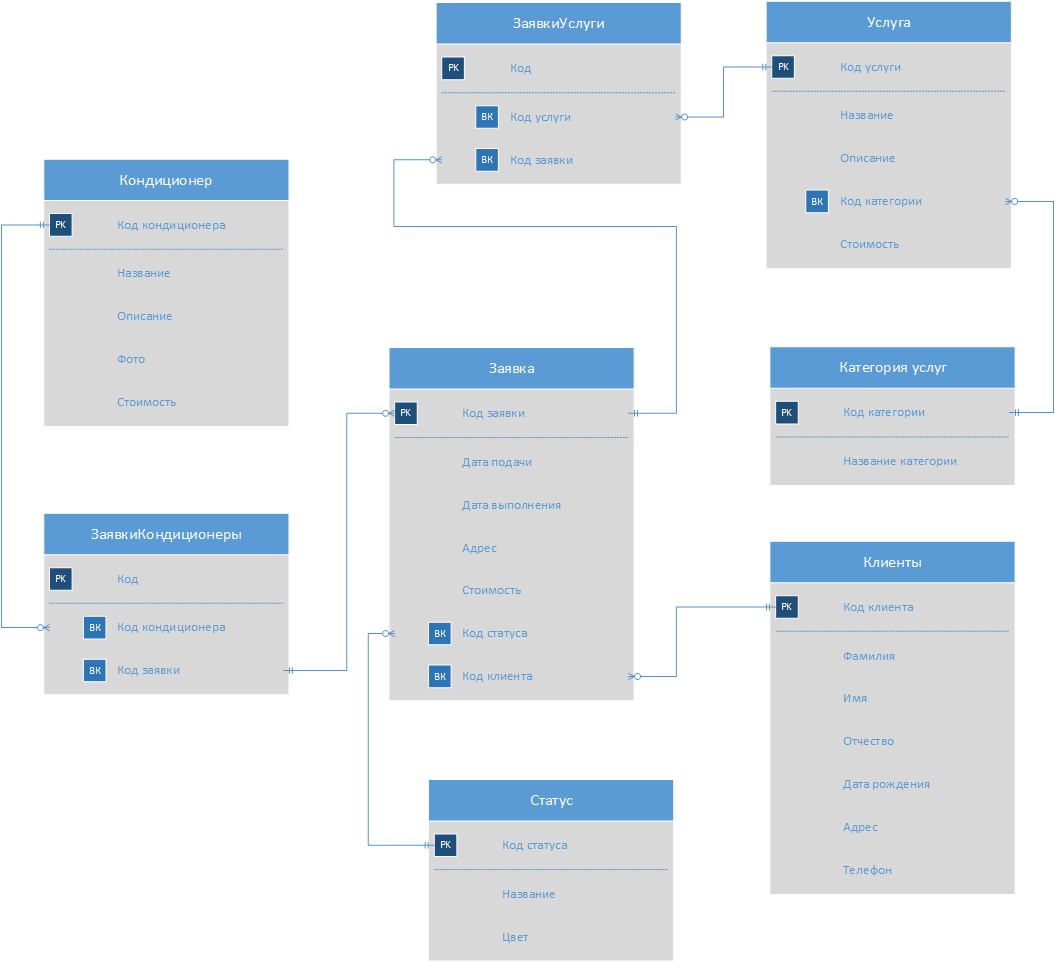


Рис.6 Реляционная модель

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: Код\_кондиционера (таблица Кондиционер), Код\_статуса (Таблица Статус), Код\_клиента (Таблица Клиент), Код (таблица ЗаявкиКондиционеры), Код (таблица ЗаявкиУслуги), Код\_заявки (таблица Заявки), Код\_услуги(таблица Услуга), Код\_категории (таблица Категория услуг). 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 “Order” – “OrderID”. Связь таблиц “ServiceCategory” и “Service” по полю “ServiceCategoryID” один ко многим (1:М). Связь таблиц “Service” и “ServiceOrder” по полю “ServiceID” один ко многим (1:М). Связь таблиц “Air” и “AirOrder” по полю “AirID” один ко многим (1:М). Связь таблиц “Client” и “Order” по полю “ClientID” один ко многим (1:М). Связь таблиц “Status” и “Order” по полю “StatusID” один ко многим (1:М). Связь таблиц “AirOrder” и “Order” по полю “OrderID” один ко многим (1:М). Связь таблиц “ServiceOrder” и “Order” по полю “OrderID” один ко многим (1:М).Все связи между таблицами базы данных типа один ко многим.

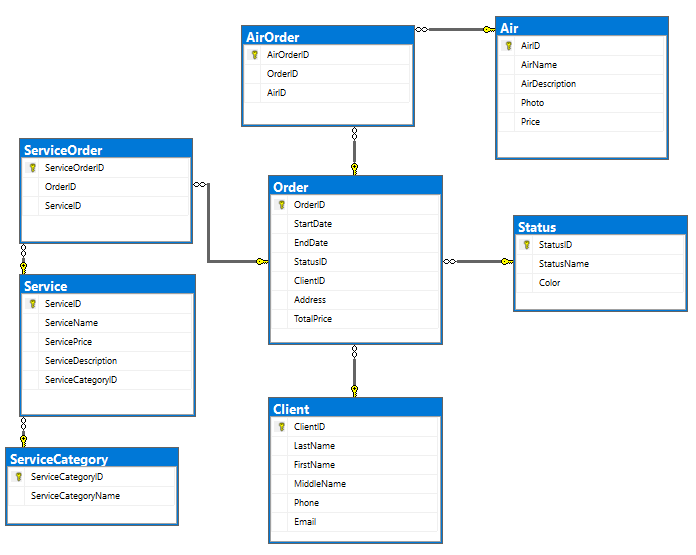


Рис.7 Схема базы данных

Таблица “Status”(Статус)

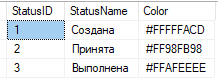


Рис.8 Таблица Status

Таблица “Client”(Клиент)

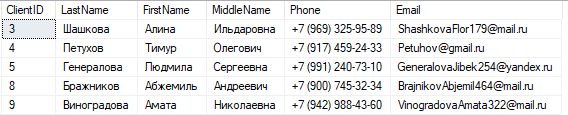


Рис.9 Таблица Client

Таблица “ServiceCategory”(Виды сервиса)

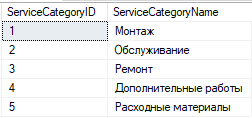


Рис.10 Таблица ServiceCategory

Таблица “Service”(Сервис)

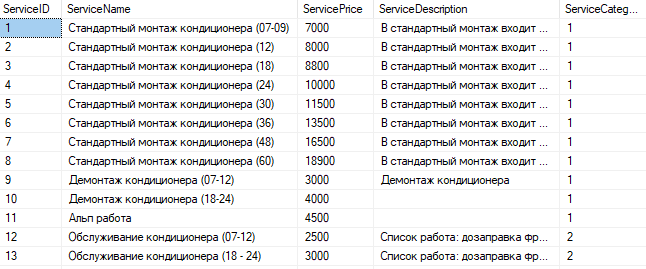


Рис.11 Таблица Service

Таблица “Air”(Кондиционер)

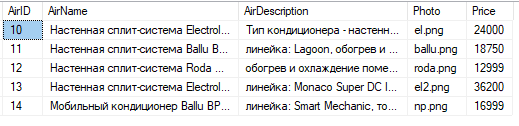


Рис.12 Таблица Air

Таблица “Order”(Заявки)

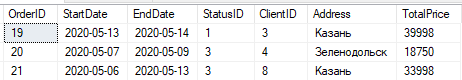


Рис.13 Таблица Order

Таблица “AirOrder”(ЗаявкиКондиционеры)

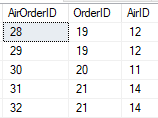


Рис.14 Таблица AirOrder

Таблица “ServiceOrder”(ЗаявкиСервисы)

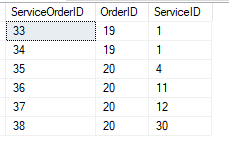


Рис.15 Таблица ServiceOrder

**3.6 Разработка интерфейса ПС**

Для запуска приложения достаточно щелкнуть дважды левой кнопкой мыши по ярлыку приложения.

В начале работы с программой пользователю будет необходимо ввести свои идентификационные данные(рис. 16).

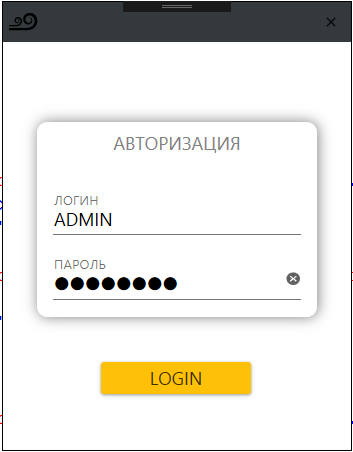


Рис.16 Форма авторизации

Если пользователь введет неправильное имя пользователя или пароль, то программа сообщит об ошибке(рис. 17).

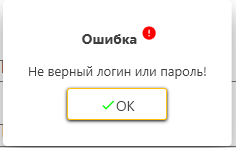


Рис.17 Ошибка

В таком случае пользователю нужно лишь заново ввести свои данные правильно и начать работу с самой программой.

После успешной авторизации пользователь приступит к работе с главной формой программы, на которой и будут отображаться все событий в базе данных.

После запуска программы пользователь приступит к работе с главной формой программы, на которой и будут отображаться все события в базе данных. «AirSystem 1.0» 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.

**Главная форма приложения**. На главной странице приложения отображены 6 кнопок для открытия основных страниц приложения (Рис.18)

Плиточный интерфейс, каждая плитка – кнопка(Button) для перехода на нужную страницу

Компонент Button открыть Меню

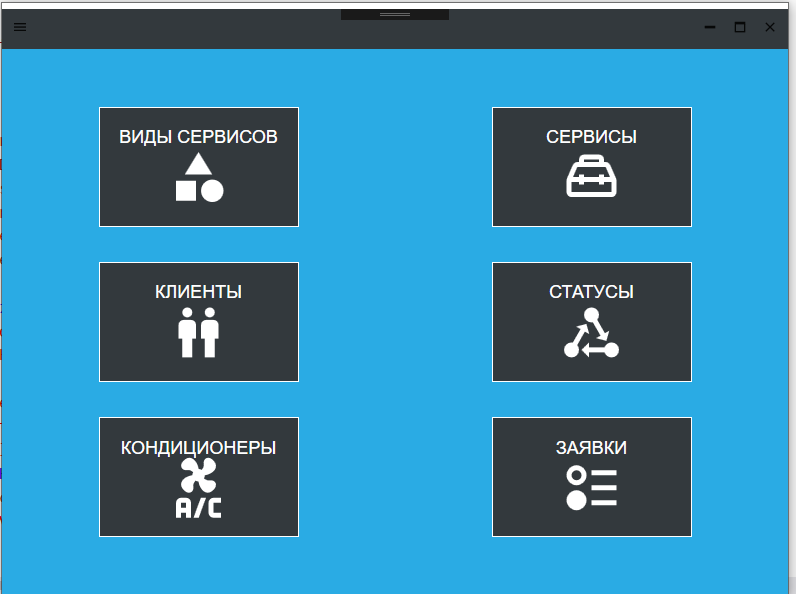
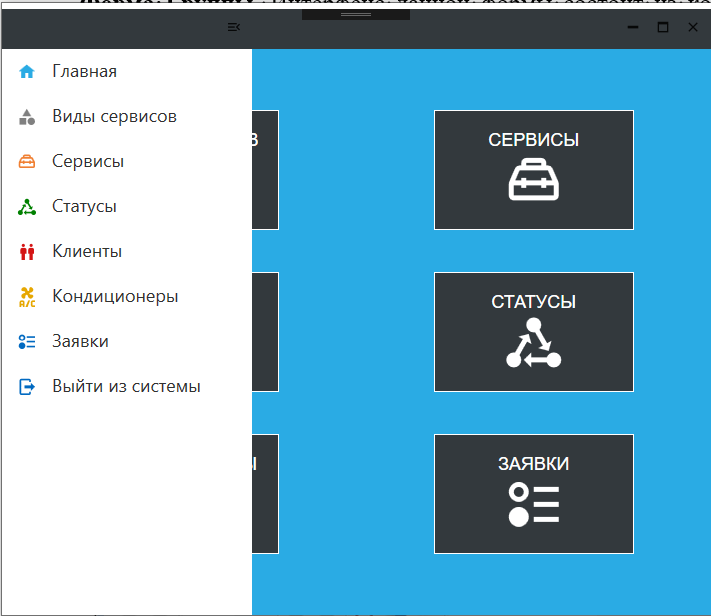


Рис.18 Главная форма приложения

**Меню**. На главной странице приложения по нажатию на кнопку  открывается меню вкладок. (Рис.19)

Компонент Button Закрыть Меню

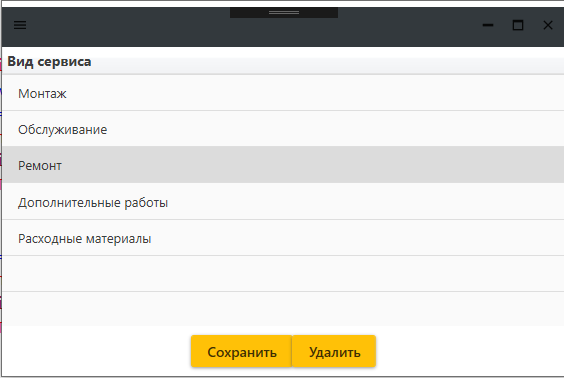


ListView

Вкладки для перехода на страницы приложения

Рис.19 Меню вкладок

Страница **Виды сервиса**. Интерфейс данной формы состоит из компонентов: DataGrid и двух кнопок: Сохранить и Удалить. (Рис.20)



Компонент DataGrid - таблица

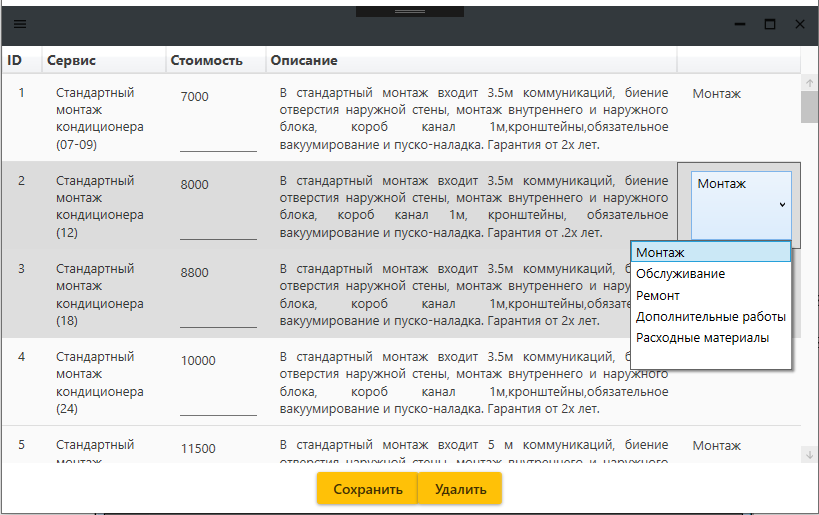
Button – кнопки Сохранить и Удалить

Компонент Button

открыть Меню

Рис.20 Страница Виды сервиса

**Страница Сервисы**. Интерфейс данной формы состоит из компонентов: DataGrid, двух кнопок: Сохранить и Удалить. (Рис.21)



Button – кнопки Сохранить и Удалить

Компонент Button

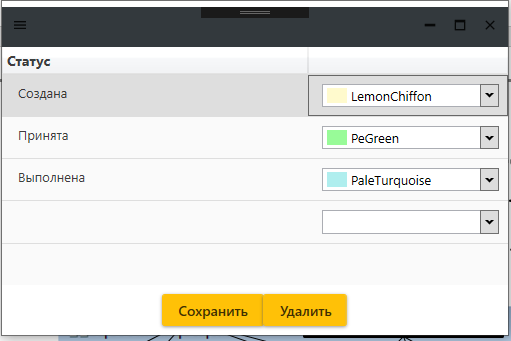
открыть Меню

Столбец DataGridComboBoxColumn

Компонент DataGrid - таблица

Рис.21 Страница Сервисы

**Страница Статус**. Интерфейс данной формы состоит из компонентов: DataGrid, двух кнопок: Сохранить и Удалить. (Рис.22)



Столбец Template c компонентом ColorPicker из палитры ExtendetWpfToolKit

Компонент DataGrid - таблица

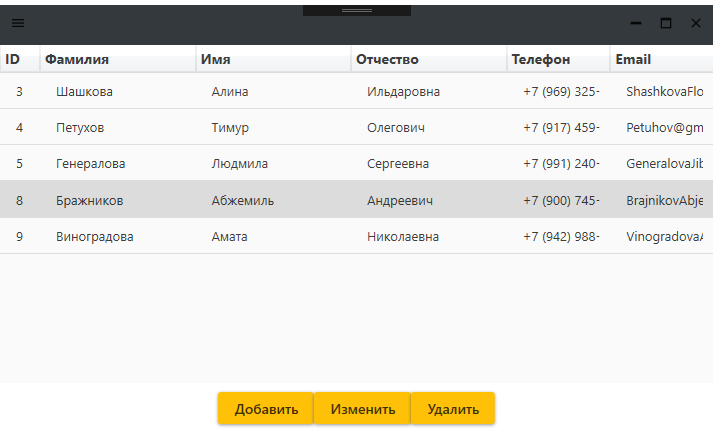
Button – кнопки Сохранить и Удалить

Компонент Button

открыть Меню

Рис.22 Страница Статус

**Страница Клиенты**. Интерфейс данной формы состоит из компонентов: DataGrid, трех кнопок: Сохранить, Изменить и Удалить. (Рис.23)



Button – кнопки Сохранить, Изменить и Удалить

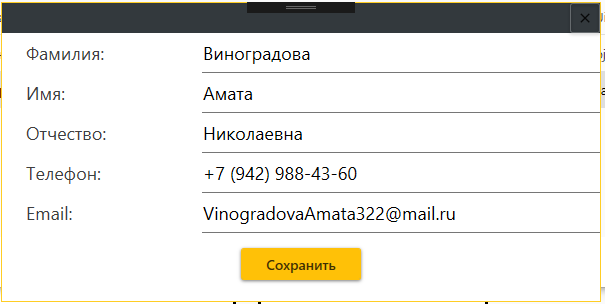
Компонент Button

открыть Меню

Компонент DataGrid - таблица

Рис.23 Страница Клиенты

**Форма Клиент**. Интерфейс данной формы состоит из компонентов: TextBox, Label и кнопки Сохранить. (Рис.24)



Кнопка Сохранить

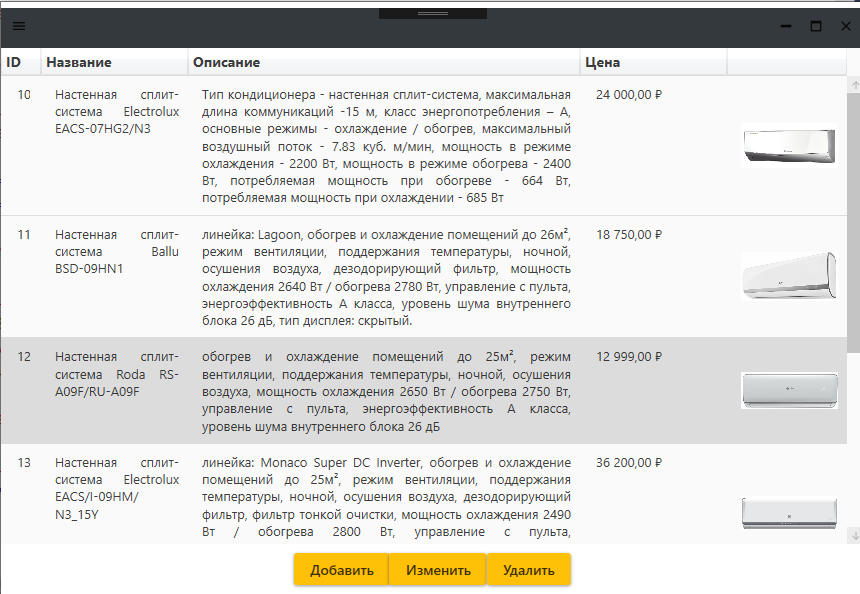
Рис.24 Форма Клиент

**Страница Кондиционеры**. Интерфейс данной формы состоит из компонентов: DataGrid, трех кнопок: Сохранить, Изменить и Удалить. (Рис.25)

Button – кнопки Сохранить, Изменить и Удалить

Компонент Button

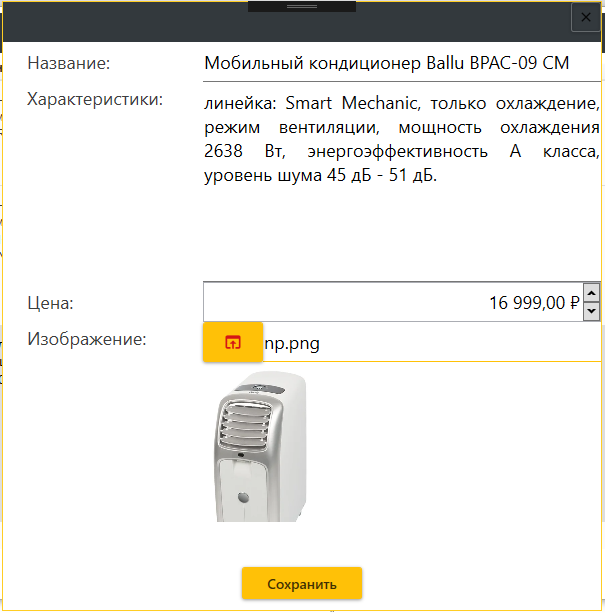
открыть Меню



Компонент DataGrid - таблица

Рис.25 Страница Кондиционеры

**Форма Кондиционер**. Интерфейс данной формы состоит из компонентов: TextBox, Label, кнопок Загрузить и Сохранить. (Рис.26)

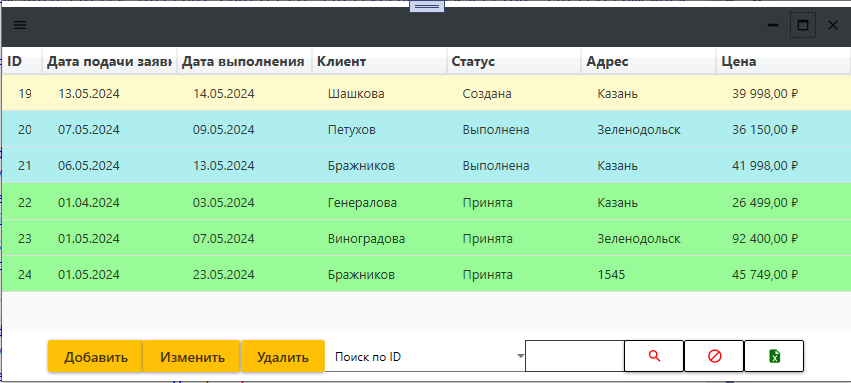


Кнопка Сохранить

Кнопка Загрузить изображение

Рис.26 Форма Кондиционер

**Страница Кондиционеры**. Интерфейс данной формы состоит из компонентов: DataGrid, шести кнопок: Сохранить, Изменить, Удалить, Найти, Отмена поиска и вывод в Excel, combobox с выбором способа поиска, TextBox – поле для ключевых символов для поиска. (Рис.27)



Компонент DataGrid - таблица

TextBox – ключ для поиска

Combobox – выбор способа поиска

Button – кнопки Найти, Отмена поиска и Вывод в Excel

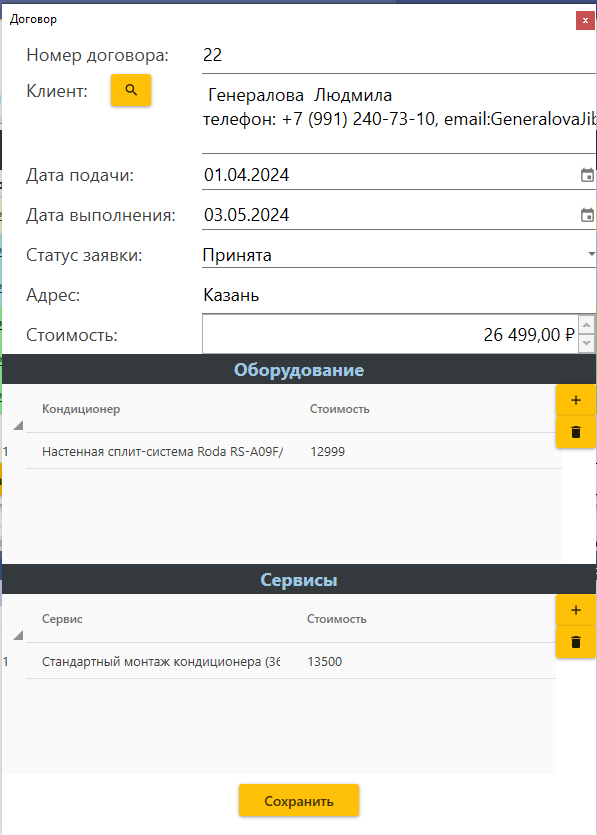
Button – кнопки Сохранить, Изменить и Удалить

Компонент Button

открыть Меню

Рис.27 Страница Заявки

**Форма Заявка**. Интерфейс данной формы состоит из компонентов: DataGrid, кнопок: Выбор клиента, Добавить, Удалить. (Рис.28)



Кнопка выбора клиента

Кнопки

Добавить и Удалить сервис

Кнопка Сохранить

Кнопки

Добавить и Удалить оборудование

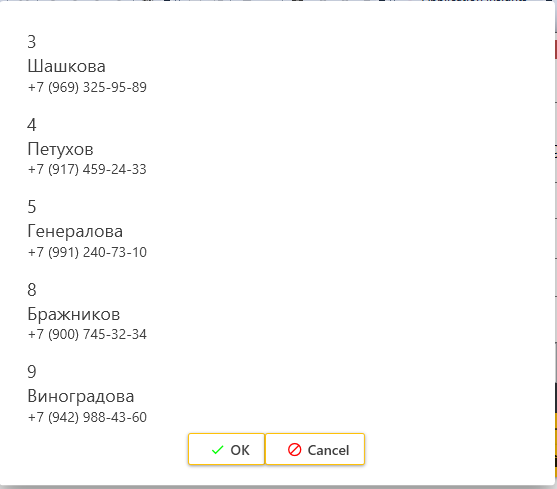
DatePicker – выбор даты

ComboBox – статус договора

DoubleUpDown из из палитры ExtendetWpfToolKit

Рис.28 Форма Заявка

**Форма Выбора клиента**. Интерфейс данной формы состоит из компонентов: ListView, кнопок:OK и Cancel. (Рис.29)



ListView – список клиентов

Button – кнопки ОК и Cancel

Рис.29 Форма Выбора клиента

**Форма Выбора кондиционера**. Интерфейс данной формы состоит из компонентов: ListView, кнопок:OK и Cancel. (Рис.30)

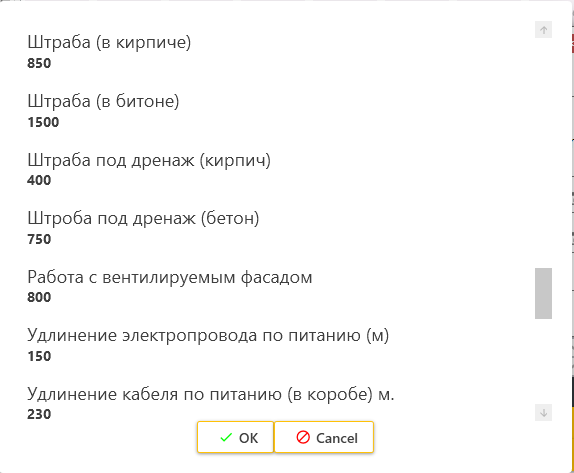


Button – кнопки ОК и Cancel

ListView – список кондиционеров

Рис.30 Форма Выбора кондиционера

**Форма Выбора сервисов**. Интерфейс данной формы состоит из компонентов: ListView, кнопок:OK и Cancel. (Рис.31)



Button – кнопки ОК и Cancel

ListView – список кондиционеров

Рис.31 Форма Выбора сервисов

**3.7 Описание структуры выходной информации**

Выходными данными являются списки отфильтрованных данных.

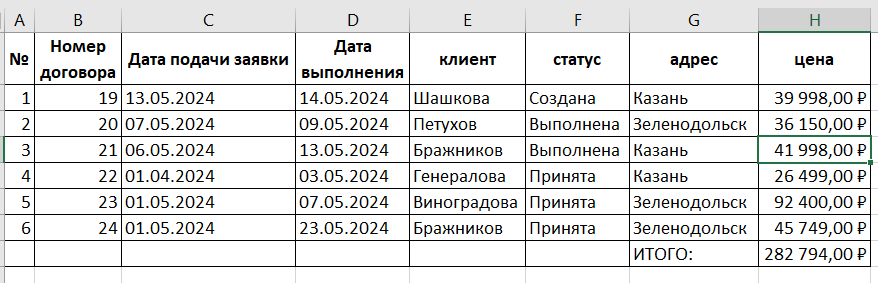


Рис.32 Список всех заявок

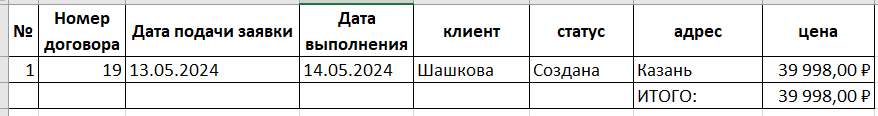


Рис.33 Заявка с указанным номером договора

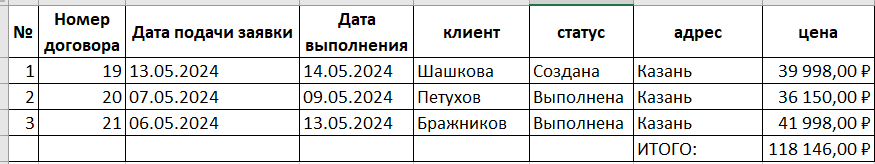


Рис.34 Заявки в Казани

**4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**4.1 Назначение программного средства**

Назначением программного средства является автоматизация деятельности фирмы.

**4.2 Условия выполнения программного средства**

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

-Компьютер с установленной на нём «AirSystem 1.0».

Подготовительные действия:

-На иконке «AirSystem 1.0» рабочего стола произвести двойной щелчок левой кнопкой мыши.

**4.3 Эксплуатация программного средства**

Этот раздел содержит обзор приложений «AirSystem 1.0» и ее основных параметров.

***Активация продукта***

Активировать копию программы AirSystem 1.0 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

Первая форма приложения – форма авторизации.

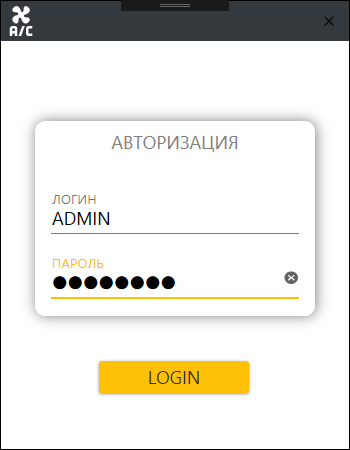


Рис.35 Форма авторизации

У пользователя есть возможность завершить работу с приложением нажав на кнопку закрыть. Приложение выведет сообщение для подтверждения или отмены закрытия приложения.

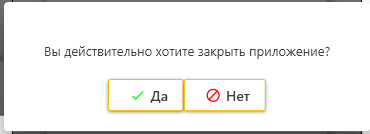


Рис.36 Сообщение пользователю

После успешной авторизации пользователю становится доступной главная форма приложения, на которой открыта стартовая страница с плиточным интерфейсом. Дополнительные страницы открываются после нажатия на кнопки на главной странице, либо выбрав нужный пункт слева в меню. Пункты меню также дублируют функционал кнопок. (Рис.37)

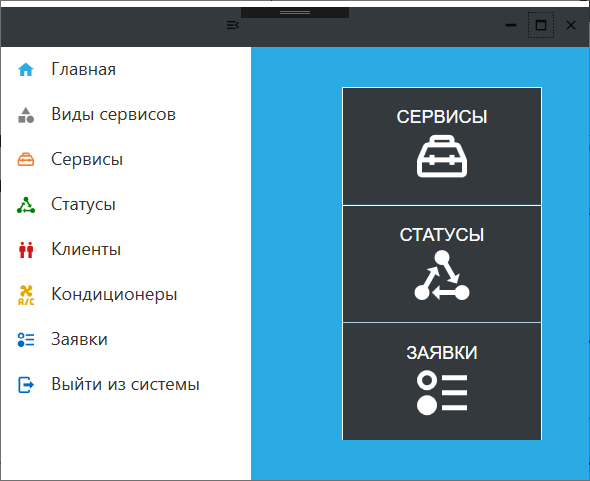


Рис.37 Главное окно приложения

Кликнув по кнопке «Закрыть» программа потребует подтверждение. (Рис.38)

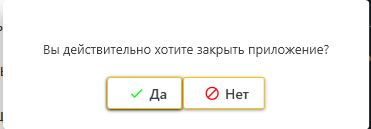


Рис.38 Сообщение программы

Ниже описаны пункты меню управления и информации.

«**Главная**»: эта страница предназначена для перехода на другие страниы приложения. (Рис.39)

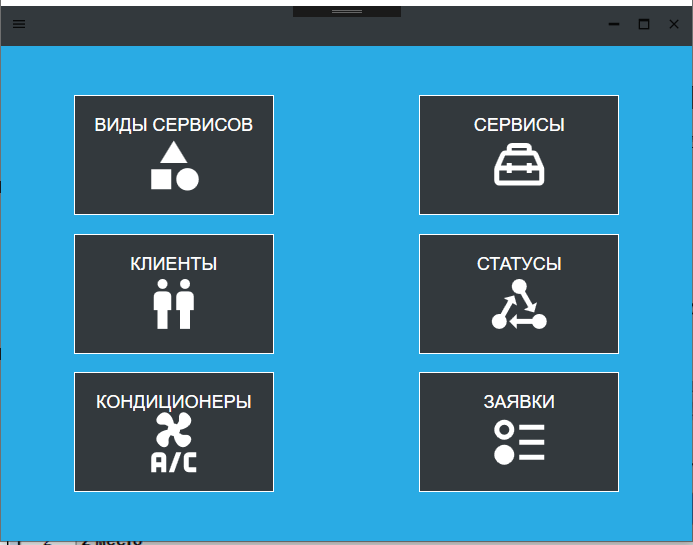


Рис.39 Страница Главная

«Виды сервисов»: эта страница предназначена для ввода, удаления и редактирования в таблице Виды сервиса. (Рис.40)

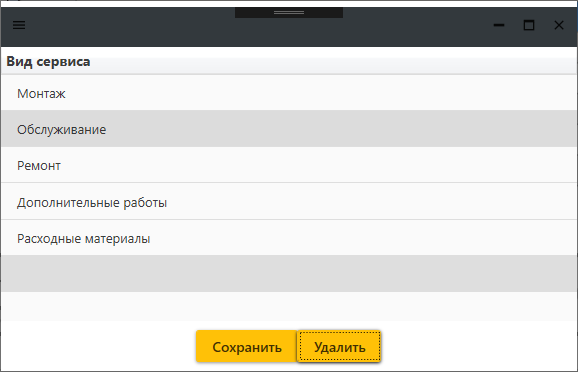


Рис.40 Страница Виды Сервисов

«Сервисы»: эта страница предназначена для ввода, удаления и редактирования в таблице Сервисы. (Рис.41)

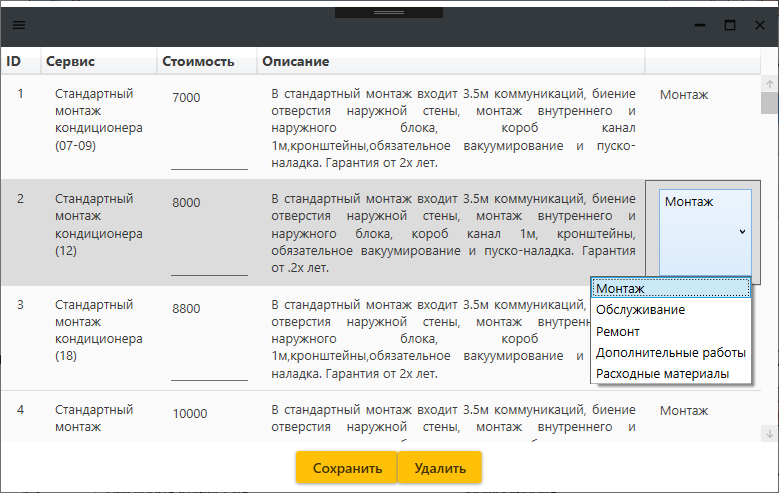


Рис.41 Страница Сервисы

«Статусы»: эта страница предназначена для ввода, удаления и редактирования в таблице Статусы. (Рис.42)

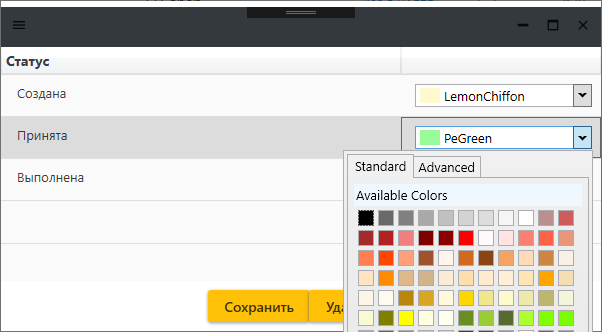


Рис.42 Страница Статусы

«Клиенты»: эта страница предназначена для отображения списка клиентов фирмы. Кнопка Удалить позволяет удалить выбранного клиента, если у клиента нет заявок в системе. Кнопки Добавить и Изменить открывают дополнительную форму. (Рис.43)

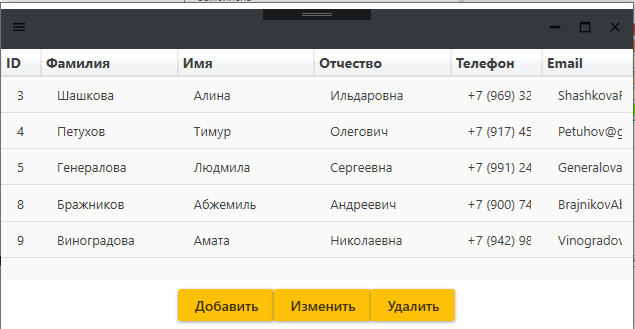


Рис.43 Страница Клиенты

«Клиент»: эта форма предназначена для редактирования или добавления новой записи. (Рис.44)

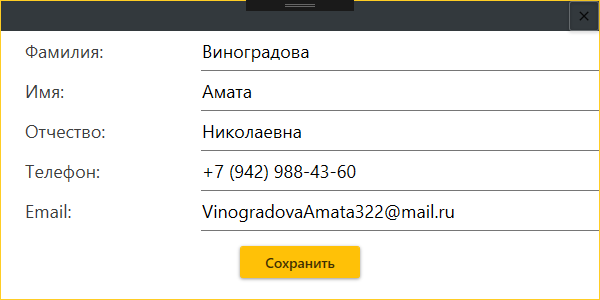


Рис.44 Форма Клиент

«Кондиционеры»: эта страница предназначена для отображения списка кондиционеров. Кнопка Удалить позволяет удалить выбранный кондиционер. Кнопки Добавить и Изменить открывают дополнительную форму. (Рис.45)

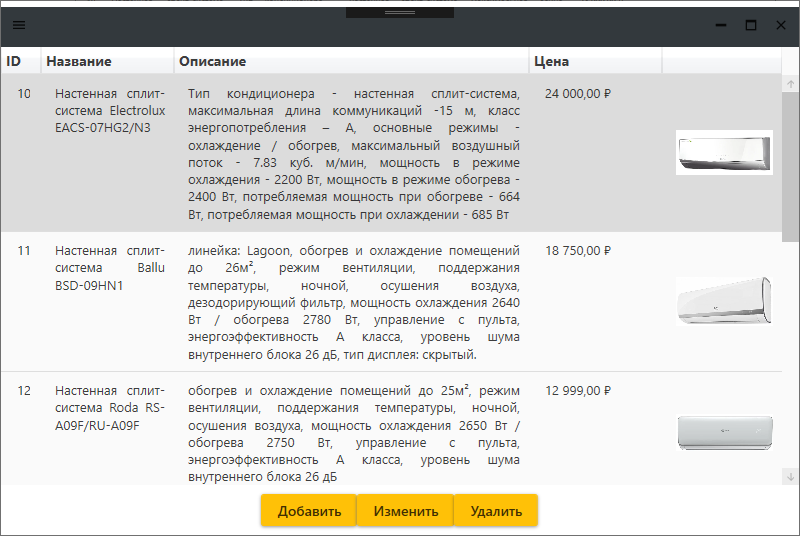


Рис.45 Страница Кондиционеры

«Кондиционер»: эта форма предназначена для редактирования или добавления новой записи. (Рис.46)

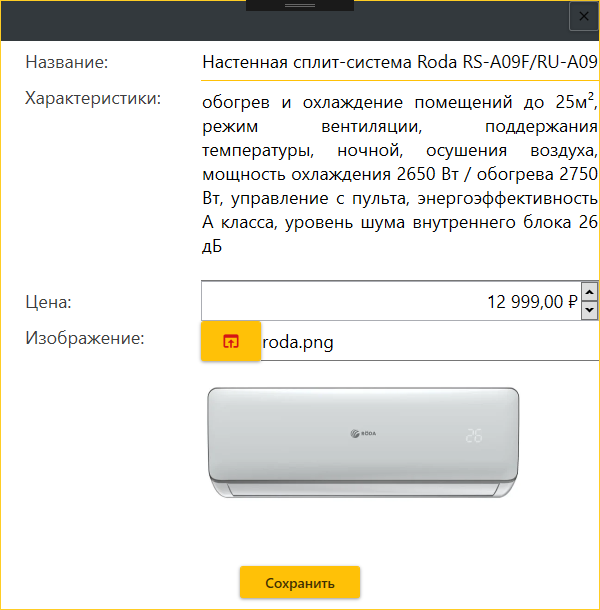


Рис.46 Форма Кондиционер

«Заявки»: основная форма приложения. Эта форма позволяет вывести список отфильтрованных данных в Excel. Кроме этого также реализована возможность удаления записи. Для того чтобы добавить новую запись нужно нажать кнопку Добавить. Для того чтобы отредактировать запись нужно либо нажать на кнопку Изменить, предварительно выбрав нужную запись.(Рис.47)

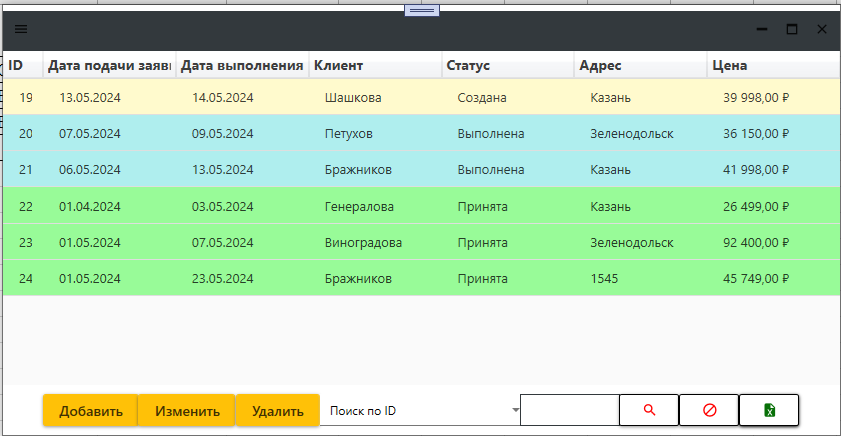


Рис.47 Страница Заявки

«Договор»: форма открывается после нажатия на кнопку Добавить.(Рис.48)

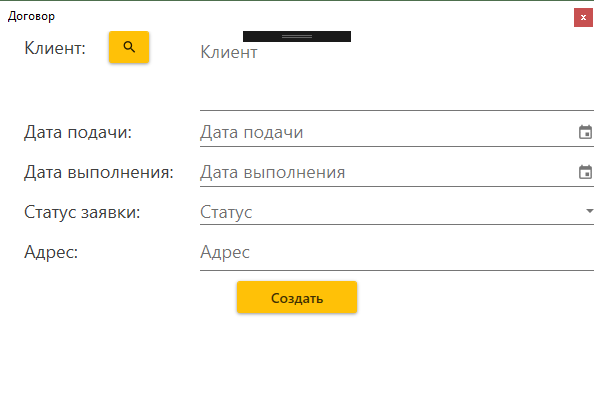


Рис.48 Окно Договор

«Договор»: форма открывается после нажатия на кнопку Изменить или после нажатия кнопки Создать в предыдущей форме .(Рис.49)

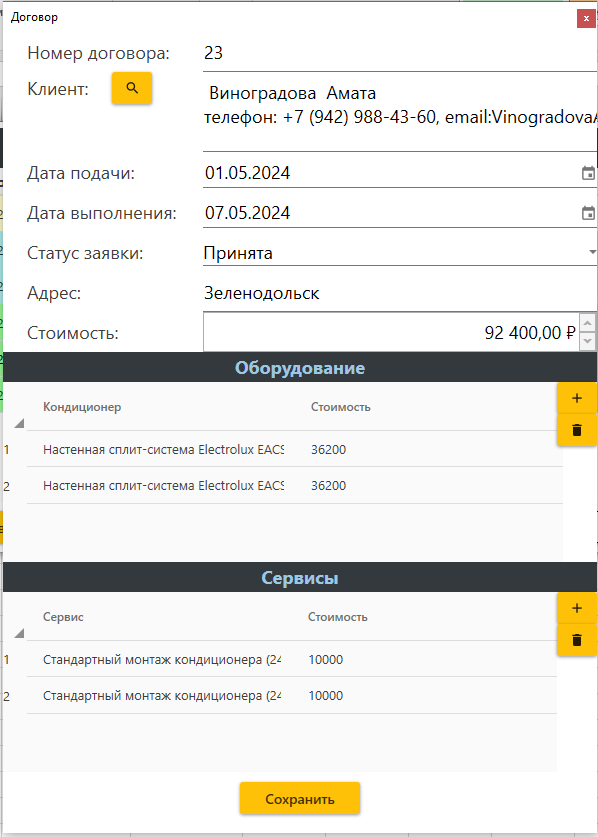


Рис.49 Окно Договор

Работа с AirSystem 1.0

После авторизации в приложении Вы перейдете на главную форму программы. На страницах Виды сервисов, Сервисы, Статусы, Клиенты, Кондиционеры и Заявки 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 AirSystem 1.0 кликните на соответствующей кнопке «Выход».

Основные кнопки AirSystem 1.0

Кнопки управления «Добавить», «Удалить запись», «Сохранить» расположены на каждой конкретной форме приложения.

Действия этих кнопок приведены в таблице 1:

Таблица-1 Действие кнопок

|  |  |
| --- | --- |
| Добавить запись  , | После нажатия на эту кнопку выйдет вспомогательная форма для добавления записей |
| Удалить запись | Кнопка «Удалить запись» позволяет удалить выделенную запись. |
| Сохранить данные | Кнопка «Сохранить данные» добавляет запись введенную пользователем в БД, либо обновляет существующую. |
| Изменить | Открывает форму в режиме редактирования информации |
| Поиск  , | «Поиск» позволяет найти нужную информацию. |
|  | Выгрузка в Excel |

**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.50)

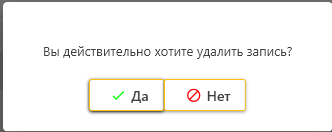


Рис.50 Сообщение Подтверждение удаления

После добавления или изменения записи, пользователю выводится сообщение. (Рис.51)

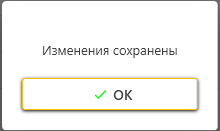


Рис.51 Сообщение Изменения сохранены

После Добавления записи, пользователю выводится сообщение. (Рис.52)

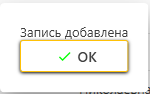


Рис.52 Сообщение Запись добавлена

При попытке удалить связанные записи, программа выводит пользователю сообщение о невозможности действия. (Рис.53)

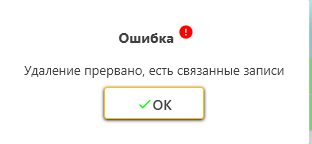


Рис.53 Сообщение Ошибка удаления

**5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА**

**5.1 Определение верхней границы количества тестов**

Для разработанного алгоритма был построен граф.

****

Рис.54 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 AirSystem 1.0 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 Разработка тестовых вариантов**

Таблица 2 – Тестирование корректного добавления информации в Виды сервисов

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

Таблица 3 – Тестирование некорректного добавления информации в таблицу Виды сервисов

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Виды сервисов | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Виды сервисов | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Вид сервиса = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, так как поле пустое | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Виды сервисов открыта | | Отображается форма Виды сервисов |  |

Таблица 4 – Тестирование корректного редактирования информации в таблице Виды сервисов

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

Таблица 5 – Тестирование некорректного редактирования информации в таблице Виды сервисов

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Виды сервисов | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Виды сервисов | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Вид сервиса = | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Виды сервисов  открыта | | Отображается форма Виды сервисов |  |

Таблица 6 – Тестирование корректного добавления информации в таблицу Студент

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Клиент | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Клиент | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Фамилия = Андронова  Имя = Мария  Отчество = Олеговна  телефон = 8964574575,  Email = masha@mail.ru | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Клиент  открыта | | Отображается форма Клиент |  |

Таблица 7 – Тестирование некорректного добавления информации в таблицу Клиент

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Клиент | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Клиент | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Фамилия =  Имя =  Отчество = Олеговна  телефон = 8964574575,  Email = masha@mail.ru | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Клиент  открыта | | Отображается форма Клиент |  |

Таблица 8 – Тестирование корректного редактирования информации в таблице Клиент

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Клиент | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Клиент | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Фамилия = Андронова  Имя = Мария  Отчество = Олеговна  телефон = 8964574575,  Email = masha@mail.ru | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Клиент  открыта | | Отображается форма Клиент |  |

Таблица 9 – Тестирование некорректного редактирования информации в таблице Клиент

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Клиент | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Клиент | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Фамилия =  Имя =  Отчество = Олеговна  телефон = 8964574575,  Email = masha@mail.ru | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Клиент  открыта | | Отображается форма Клиент |  |

Таблица 10 – Тестирование корректного добавления информации в таблицу Преподаватель

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Кондиционер | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Кондиционер | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название = Electrolux EACS-07HG2/N3  Описание = настенная сплит-система  Фото = Загружено  Цена = 13660 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Кондиционер  открыта | | Отображается форма Кондиционер |  |

Таблица 11 – Тестирование некорректного добавления информации в таблицу Кондиционер

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Кондиционер | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Кондиционер | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Название =  Описание = настенная сплит-система, Фото = Загружено  Цена = 13660 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Кондиционер  открыта | | Отображается форма Кондиционер |  |

Таблица 12 – Тестирование корректного редактирования информации в таблице Кондиционер

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Кондиционер | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Кондиционер | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название = Electrolux EACS-07HG2/N3, Описание = настенная сплит-система  Фото = Загружено,Цена = 13660 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Кондиционер  открыта | | Отображается форма Кондиционер |  |

Таблица 13 – Тестирование некорректного редактирования информации в таблице Кондиционер

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Кондиционер | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Кондиционер | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Название =  Описание = настенная сплит-система  Фото = Загружено  Цена = 13660 | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Кондиционер  открыта | | Отображается форма Кондиционер |  |

**5.3 Составление отчета о тестировании**

Для форм проекта были созданы и проведены тесты, результаты сведены в таблицу:

Таблица 14 – Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Прохождение | | |
| Форма | Число ТП | пройдено | | не пройдено |
| Сервисы | 4 | 4 | | 0 |
| Виды сервисов | 4 | 4 | | 0 |
| Кондиционер | 4 | 4 | | 0 |
| Клиенты | 4 | 4 | | 0 |
| Заявки | 4 | 4 | | 0 |
| Статусы | 4 | 4 | | 0 |
| ИТОГО | 24 | 24 | 0 | |

По результатам тестирования были выявлены и устранены следующие проблемы:

При выполнении теста таблица №8. Тестирование формы «Сервисы», Некорректный запрос

**Ошибка**



Рис.54 Ошибка теста №8

**Причина**

Поле цена имеет тип вещественное число, было введено строковое значение.

**Принятые меры**

Ошибка устранена. В поле цена можно ввести только цифры и запятую

**ЗАКЛЮЧЕНИЕ**

**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 10 Professional.

Процессор: Intel Pentium G2030 3.0 GHz.

ОЗУ: 4 Gb

Таблица 15 – 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 16 – 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 17 – Сложность

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

Таблица 18 – Сложность экрана

|  |  |  |  |
| --- | --- | --- | --- |
| Экраны | Количество Кондиционерских (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 19 – Сложность отчета

|  |  |  |  |
| --- | --- | --- | --- |
| Отчеты | Количество Кондиционерских (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 = 6 тысяч строк кода/1 месяца= 6 тысяч строк кода/месяц

Качество = 6 ошибок /6 тысяч строк кода=1 ошибок/тысяч строк кода

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

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

2. Либерти, Д. Программирование на Си Шарп. - М.: Символ-плюс, 2021. - 684 с.

3. Ватсон К. Си Шарп/К.Ватсон.-М.:Лори, 2021.-862 с

4. Балена Ф., Димауро Д. Современная практика программирования на Visual Basic и Visual C#.-М.:Русская редакция, 2021,604с.

5. Енин А., Енин Н. Локальная СУБД своими руками. Учимся на примерах.-М.:СОЛОН-ПРЕСС, 2021,464с.,СД

6. Лабор В.В. Си Шарп: СоВозврат приложений для Windows/-Мн.: Харвест, 2021-384с.

7. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2021-512с.

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#. 2018 г.

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

14. Никита Культин. Основы программирования в MicrosoftVisualC# 2019. 2019 г.

[Электронный ресурс http://www.litres.ru/nikita-kultin/osnovy-programmirovaniya-v-microsoft-visual-c-2010-2]

15. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2021.

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

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

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

**Приложение А**

Фирма по продаже и обслуживанию кондиционеров

Текст кода программного средства

460.ЗМК.00011-01 12 01

**Текст кода программного средства**

**App.config**

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<configSections>

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />

</configSections>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />

</startup>

<connectionStrings>

<add name="AirDBEntities" connectionString="metadata=res://\*/Models.AirModel.csdl|res://\*/Models.AirModel.ssdl|res://\*/Models.AirModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=.\SQLEXPREss;initial catalog=AirDB;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

<entityFramework>

<defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />

</providers>

</entityFramework>

</configuration>

**App.xaml**

<Application x:Class="AirSystemApp.App"

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

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

xmlns:local="clr-namespace:AirSystemApp"

StartupUri="LoginWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

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

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

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

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

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.Amber.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.Lime.xaml" />

<ResourceDictionary Source="Resourses/Dictionary.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

LoginWindow.xaml

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

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

xmlns:local="clr-namespace:AirSystemApp"

mc:Ignorable="d"

Title="Login" Height="450" Width="350"

ResizeMode="NoResize"

BorderThickness="1"

WindowStartupLocation="CenterScreen"

WindowStyle="None"

FontFamily="Segoe UI Emoji">

<Grid >

<Rectangle Height="40" VerticalAlignment="Top" Fill="{DynamicResource dark}" >

<!--<Rectangle.Fill>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#FF2281D1"/>

<GradientStop Color="#FF34268A" Offset="1"/>

<GradientStop Color="#FF33288B" Offset="0.546"/>

</LinearGradientBrush>

</Rectangle.Fill>-->

</Rectangle>

<Rectangle Width="280" Height="195" VerticalAlignment="Top" Margin="34,120,34,0" RadiusY="10" RadiusX="10" Fill="White">

<Rectangle.Effect>

<DropShadowEffect BlurRadius="20" Direction="0" RenderingBias="Quality" ShadowDepth="1" Color="#FFBBBBBB"/>

</Rectangle.Effect>

</Rectangle>

<Grid VerticalAlignment="Top" Margin="35,120" Height="240">

<Label Content="АВТОРИЗАЦИЯ" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="5" Foreground="Gray" FontSize="18"/>

<StackPanel VerticalAlignment="Center" Margin="15">

<TextBox x:Name="tbLogin" Margin="0,10" materialDesign:HintAssist.Hint="ЛОГИН" Style="{StaticResource MaterialDesignFloatingHintTextBox}" FontSize="18"/>

<PasswordBox x:Name="psbPassword" Margin="0,10" materialDesign:HintAssist.Hint="ПАРОЛЬ" Style="{StaticResource MaterialDesignFloatingHintPasswordBox}" materialDesign:TextFieldAssist.HasClearButton="True" FontSize="18"/>

</StackPanel>

</Grid>

<materialDesign:PackIcon Kind="AirConditioner" Foreground="#FFFFFF" Width="30" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="5"/>

<!--<Image Source="Resourses/wind.png" Width="30" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="5"/>-->

<Button x:Name="btnExit" Width="30" Height="30" BorderThickness="0" Padding="0" Margin="5" HorizontalAlignment="Right" VerticalAlignment="Top" FontSize="16" Background="{x:Null}" Click="BtnExit\_Click" >

<materialDesign:PackIcon Kind="Close"/>

</Button>

<Button Width="150" HorizontalAlignment="Right" VerticalAlignment="Top" Margin="0,360,100,0" FontSize="18" Content="LOGIN" Click="Button\_Click"/>

<materialDesign:DialogHost x:Name="DialogHost" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите закрыть приложение?" Margin="20" />

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

<Button x:Name="btnExitApp" Click="BtnExitM\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="4" Grid.ColumnSpan="2" x:Name="ErrorHost" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<StackPanel Orientation="Horizontal" Margin="5" HorizontalAlignment="Center">

<Label Content="Ошибка" FontSize="14" FontWeight="Bold"/>

<materialDesign:PackIcon Kind="Error" Foreground="#FF0000" HorizontalAlignment="Center"/>

</StackPanel>

<TextBlock x:Name="tbError" Text="" VerticalAlignment="Top"

HorizontalAlignment="Center" TextWrapping="Wrap" Margin="5"/>

<Button x:Name="btnOkError" Click="BtnOk\_Click" Background="White" Width="100" Margin="5">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Window>

**LoginWindow.cs**

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 AirSystemApp

{

/// <summary>

/// Логика взаимодействия для LoginWindow.xaml

/// </summary>

public partial class LoginWindow : Window

{

public LoginWindow()

{

InitializeComponent();

}

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

{

DialogHost.IsOpen = true;

}

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

{

Close();

}

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

{

DialogHost.IsOpen = false;

}

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

{

if ((tbLogin.Text == "ADMIN") && (psbPassword.Password == "1"))

{

MainWindow mainWindow = new MainWindow();

mainWindow.Owner = this;

this.Hide();

mainWindow.Show();

}

else

{

tbError.Text = "Не верный логин или пароль!";

ErrorHost.IsOpen = true;

}

}

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

{

ErrorHost.IsOpen = false;

}

}

}

**MainWindow.xaml**

<Window x:Class="AirSystemApp.MainWindow"

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

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

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

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

xmlns:local="clr-namespace:AirSystemApp"

mc:Ignorable="d"

Style="{DynamicResource base\_window}"

WindowStyle="None"

Title="Фирма Кондиционер" Icon="Resourses/wind.ico" WindowStartupLocation="CenterScreen" Closed="Window\_Closed"

>

<Grid>

<Frame x:Name="MainFrame" NavigationUIVisibility="Hidden" />

</Grid>

</Window>

**MainWindow.cs**

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 AirSystemApp.Pages;

namespace AirSystemApp

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public Frame xFrame

{

get;

set;

}

public ListView listView

{

get;

set;

}

public MainWindow()

{

InitializeComponent();

UIPage1 uIPage1 = new UIPage1();

MainFrame.NavigationService.Navigate(uIPage1);

xFrame = uIPage1.ContentFrame;

listView = uIPage1.lstView;

}

private void Window\_Closed(object sender, EventArgs e)

{

// Owner.Show();

}

}

}

UIPage1.xaml

<Page x:Class="AirSystemApp.Pages.UIPage1"

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:AirSystemApp.Pages"

mc:Ignorable="d"

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

Title="UIPage1" Loaded="Page\_Loaded">

<Page.Resources>

<Storyboard x:Key="CloseMenu">

<DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.X)" Storyboard.TargetName="GridMenu">

<EasingDoubleKeyFrame KeyTime="0" Value="250"/>

<EasingDoubleKeyFrame KeyTime="0:0:0.5" Value="0"/>

</DoubleAnimationUsingKeyFrames>

<DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.Opacity)" Storyboard.TargetName="GridBackground">

<EasingDoubleKeyFrame KeyTime="0" Value="1"/>

<EasingDoubleKeyFrame KeyTime="0:0:0.5" Value="0"/>

</DoubleAnimationUsingKeyFrames>

</Storyboard>

<Storyboard x:Key="OpenMenu">

<DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.X)" Storyboard.TargetName="GridMenu">

<EasingDoubleKeyFrame KeyTime="0" Value="0"/>

<EasingDoubleKeyFrame KeyTime="0:0:0.5" Value="250"/>

</DoubleAnimationUsingKeyFrames>

<DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.Opacity)" Storyboard.TargetName="GridBackground">

<EasingDoubleKeyFrame KeyTime="0" Value="0"/>

<EasingDoubleKeyFrame KeyTime="0:0:0.5" Value="1"/>

</DoubleAnimationUsingKeyFrames>

</Storyboard>

</Page.Resources>

<Page.Triggers>

<!--<EventTrigger RoutedEvent="Grid.MouseLeave" SourceName="GridMenu">

<BeginStoryboard x:Name="CloseMenu\_BeginStoryboard" Storyboard="{StaticResource CloseMenu}"/>

</EventTrigger>-->

<EventTrigger RoutedEvent="ButtonBase.Click" SourceName="ButtonClose">

<BeginStoryboard Storyboard="{StaticResource CloseMenu}"/>

</EventTrigger>

<EventTrigger RoutedEvent="ListView.SelectionChanged" SourceName="lstView">

<BeginStoryboard Storyboard="{StaticResource CloseMenu}"/>

</EventTrigger>

<EventTrigger RoutedEvent="ButtonBase.Click" SourceName="ButtonOpen">

<BeginStoryboard Storyboard="{StaticResource OpenMenu}"/>

</EventTrigger>

</Page.Triggers>

<Grid>

<Grid x:Name="GridBackground" Background="#55313131" Opacity="0"/>

<Frame x:Name="ContentFrame" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Margin="0,40,0,0"/>

<Rectangle Height="40" VerticalAlignment="Top" Fill="{DynamicResource dark}"/>

<!--<Button x:Name="btnLogout" Margin="3, 3,93,3" HorizontalAlignment="Right" ToolTip="Выйти"

VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{x:Null}" Width="30" Height="30" Padding="0" Click="BtnLogout\_Click">

<materialDesign:PackIcon Kind="Logout" />

</Button>-->

<Button x:Name="btnExit" ToolTip="Закрыть приложение" Margin="3" HorizontalAlignment="Right" VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{x:Null}" Width="30" Height="30" Padding="0" Click="BtnExit\_Click">

<materialDesign:PackIcon Kind="Close" />

</Button>

<materialDesign:DialogHost x:Name="DialogHost" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите закрыть приложение?" Margin="20" />

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

<Button x:Name="btnExitApp" Click="BtnExitM\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost x:Name="DialogHostLogout" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите выйти из системы?" Margin="20" />

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

<Button x:Name="btnLogoutYes" Click="BtnLogout\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancelLogout" Click="BtnCancelLogout\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<Button x:Name="btnMaximize" Margin="3, 3,33,3" HorizontalAlignment="Right" VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{x:Null}" Width="30" Height="30" Padding="0" Click="BtnMaximize\_Click">

<materialDesign:PackIcon Kind="WindowMaximize" />

</Button>

<Button x:Name="btnMinimize" Margin="3, 3,63,3" HorizontalAlignment="Right" VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{x:Null}" Width="30" Height="30" Padding="0" Click="BtnMinimize\_Click">

<materialDesign:PackIcon Kind="WindowMinimize" />

</Button>

<Button x:Name="ButtonOpen" Margin="3" HorizontalAlignment="Left" VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{x:Null}" Width="30" Height="30" Padding="0">

<materialDesign:PackIcon Kind="Menu" />

</Button>

<Grid x:Name="GridMenu" Width="250" HorizontalAlignment="Left" Margin="-250 0 0 0" Background="White" RenderTransformOrigin="0.5,0.5">

<Grid.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform/>

</TransformGroup>

</Grid.RenderTransform>

<StackPanel>

<Rectangle Height="40" Fill="{DynamicResource dark}"/>

<ListView x:Name="lstView" Foreground="#FF313131" FontFamily="Champagne &amp; Limousines" FontSize="18">

<ListViewItem x:Name="itemHome" Height="45" Padding="0" Selected="ItemHome\_Selected" >

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="Home" Width="20" Height="20" Foreground="{DynamicResource blue}" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Главная" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem x:Name="itemServiceCategory" Height="45" Padding="0" Selected="ItemServiceCategory\_Selected" >

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="Category" Width="20" Height="20" Foreground="Gray" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Виды сервисов" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem x:Name="itemServices" Height="45" Padding="0" Selected="ItemServices\_Selected">

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="ServiceToolbox" Width="20" Height="20" Foreground="#FFF08033" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Сервисы" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem Height="45" Padding="0" x:Name="itemStatus" Selected="ItemStatus\_Selected">

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="StateMachine" Width="20" Height="20" Foreground="Green" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Статусы" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem Height="45" Padding="0" x:Name="itemClients" Selected="ItemClients\_Selected">

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="ManMan" Width="20" Height="20" Foreground="#FFD41515" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Клиенты" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem Height="45" Padding="0" x:Name="itemAirs" Selected="ItemAirs\_Selected">

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="AirConditioner" Width="20" Height="20" Foreground="#FFE6A701" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Кондиционеры" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem Height="45" Padding="0" x:Name="itemOrders" Selected="ItemOrders\_Selected">

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="OrderBoolAscending" Width="20" Height="20" Foreground="#FF0069C1" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Заявки" Margin="10"/>

</StackPanel>

</ListViewItem>

<ListViewItem x:Name="itemLogout" Height="45" Padding="0" PreviewMouseLeftButtonDown="ItemLogout\_MouseDown">

<StackPanel Orientation="Horizontal" Margin="10 0">

<materialDesign:PackIcon Kind="Logout" Width="20" Height="20" Foreground="#FF0069C1" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="Выйти из системы" Margin="10"/>

</StackPanel>

</ListViewItem>

</ListView>

</StackPanel>

<Button x:Name="ButtonClose" HorizontalAlignment="Right" Margin="3" VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{x:Null}" Width="30" Height="30" Padding="0">

<materialDesign:PackIcon Kind="MenuOpen"/>

</Button>

</Grid>

<!--<materialDesign:PopupBox Margin="3" HorizontalAlignment="Right" VerticalAlignment="Top" PlacementMode="BottomAndAlignCentres" Background="{x:Null}" BorderBrush="{x:Null}"

ToolTipService.Placement="Right" Height="30" Width="30"

Style="{DynamicResource MaterialDesignMultiFloatingActionPopupBox}">

<StackPanel>

<Button x:Name="btnExitM" ToolTip="Закрыть приложение" Click="BtnExitM\_Click" >

<materialDesign:PackIcon Kind="ExitToApp"/>

</Button>

<Button x:Name="btnLogout"

ToolTip="Выйти" Click="BtnLogout\_Click">

<materialDesign:PackIcon Kind="Logout"/>

</Button>

</StackPanel>

</materialDesign:PopupBox>-->

</Grid>

</Page>

**UIPage1.cs**

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 MaterialDesignThemes.Wpf;

namespace AirSystemApp.Pages

{

/// <summary>

/// Логика взаимодействия для UIPage1.xaml

/// </summary>

public partial class UIPage1 : Page

{

public UIPage1()

{

InitializeComponent();

}

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

{

DialogHost.IsOpen = true;

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

if (mainWindow.WindowState == WindowState.Normal)

{

mainWindow.WindowState = WindowState.Maximized;

btnMaximize.Content = new PackIcon { Kind = PackIconKind.WindowRestore };

}

else if (mainWindow.WindowState == WindowState.Maximized)

{

mainWindow.WindowState = WindowState.Normal;

btnMaximize.Content = new PackIcon { Kind = PackIconKind.WindowMaximize };

}

// btnMaximize.SetResourceReference(Control.ForegroundProperty, "gray");

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.WindowState = WindowState.Minimized;

}

private void ItemServiceCategory\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new ServiceCategoryPage());

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new StartPage());

}

private void ItemHome\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new StartPage());

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.Owner.Close();

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.Close();

mainWindow.Owner.Show();

}

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

{

DialogHost.IsOpen = false;

}

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

{

DialogHostLogout.IsOpen = false;

}

private void ItemLogout\_Selected(object sender, RoutedEventArgs e)

{

DialogHostLogout.IsOpen = true;

}

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

{

DialogHostLogout.IsOpen = true;

}

private void ItemServices\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new ServicePage());

}

private void ItemStatus\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new StatusPage());

}

private void ItemClients\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new ClientPage());

}

private void ItemAirs\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new AirPage());

}

private void ItemOrders\_Selected(object sender, RoutedEventArgs e)

{

ContentFrame.Navigate(new OrdersPage());

}

}

}

**StartPage.xaml**

<Page

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:AirSystemApp.Pages"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes" x:Class="AirSystemApp.Pages.StartPage"

mc:Ignorable="d"

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

Title="StartPage">

<Grid Background="{DynamicResource blue}">

<!--<Image HorizontalAlignment="Left" Height="100" Margin="331,32,0,0" VerticalAlignment="Top" Width="100" Source="/AirSystemApp;component/Resourses/logo.png"/>-->

<Grid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Button x:Name="btnServiceType" Style="{DynamicResource button\_start\_page}" Grid.Row="1" Grid.Column="0" Click="BtnServiceType\_Click">

<StackPanel>

<TextBlock Text="ВИДЫ СЕРВИСОВ" HorizontalAlignment="Center"/>

<materialDesign:PackIcon Kind="Category" Foreground="#FFFFFF" HorizontalAlignment="Center" VerticalAlignment="Center" Height="60" Width="60" />

</StackPanel>

</Button>

<Button x:Name="btnServices" Style="{DynamicResource button\_start\_page}" Grid.Row="1" Grid.Column="1" Click="BtnServices\_Click">

<StackPanel>

<TextBlock Text="СЕРВИСЫ" HorizontalAlignment="Center"/>

<materialDesign:PackIcon HorizontalAlignment="Center" Kind="ServiceToolbox" Foreground="#FFFFFF" VerticalAlignment="Center" Height="60" Width="60"/>

</StackPanel>

</Button>

<Button x:Name="btnClients" Style="{DynamicResource button\_start\_page}" Grid.Row="2" Grid.Column="0" Click="BtnClients\_Click">

<StackPanel>

<TextBlock Text="КЛИЕНТЫ" HorizontalAlignment="Center" />

<materialDesign:PackIcon Kind="ManMan" HorizontalAlignment="Center" Foreground="#FFFFFF" VerticalAlignment="Center" Height="60" Width="60"/>

</StackPanel>

</Button>

<Button x:Name="btnStatus" Style="{DynamicResource button\_start\_page}" Grid.Row="2" Grid.Column="1" Click="BtnStatus\_Click">

<StackPanel>

<TextBlock Text="СТАТУСЫ" HorizontalAlignment="Center"/>

<materialDesign:PackIcon Kind="StateMachine" HorizontalAlignment="Center" Foreground="#FFFFFF" VerticalAlignment="Center" Height="60" Width="60"/>

</StackPanel>

</Button>

<Button x:Name="btnAirCond" Style="{DynamicResource button\_start\_page}" Grid.Row="3" Grid.Column="0" Click="BtnAirCond\_Click">

<StackPanel>

<TextBlock Text="КОНДИЦИОНЕРЫ" HorizontalAlignment="Center"/>

<materialDesign:PackIcon Kind="AirConditioner" Foreground="#FFFFFF" HorizontalAlignment="Center" VerticalAlignment="Center" Height="60" Width="60"/>

</StackPanel>

</Button>

<Button x:Name="btnOrders" Style="{DynamicResource button\_start\_page}" Grid.Row="3" Grid.Column="1" Click="BtnOrders\_Click">

<StackPanel>

<TextBlock Text="ЗАЯВКИ" HorizontalAlignment="Center"/>

<materialDesign:PackIcon Kind="OrderBoolAscending" HorizontalAlignment="Center" Foreground="#FFFFFF" VerticalAlignment="Center" Height="60" Width="60"/>

</StackPanel>

</Button>

</Grid>

</Page>

**StartPage.cs**

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 AirSystemApp.Pages

{

/// <summary>

/// Логика взаимодействия для StartPage.xaml

/// </summary>

public partial class StartPage : Page

{

public StartPage()

{

InitializeComponent();

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.listView.SelectedIndex = -1;

mainWindow.xFrame.Navigate(new ServiceCategoryPage());

//NavigationService.Navigate(new ServiceCategoryPage());

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.listView.SelectedIndex = -1;

mainWindow.xFrame.Navigate(new ServicePage());

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.listView.SelectedIndex = -1;

mainWindow.xFrame.Navigate(new ClientPage());

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.listView.SelectedIndex = -1;

mainWindow.xFrame.Navigate(new StatusPage());

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.listView.SelectedIndex = -1;

mainWindow.xFrame.Navigate(new AirPage());

}

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

{

MainWindow mainWindow = Application.Current.Windows.OfType<MainWindow>().SingleOrDefault(x => x.IsActive);

mainWindow.listView.SelectedIndex = -1;

mainWindow.xFrame.Navigate(new OrdersPage());

}

}

}

**ServiceCategoryPage.xaml**

<Page

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:AirSystemApp.Pages"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes" x:Class="AirSystemApp.Pages.ServiceCategoryPage"

mc:Ignorable="d"

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

Title="ServiceCategoryPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid Grid.Row="0" x:Name="dtServiceCategory" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" ColumnHeaderStyle="{DynamicResource header\_column}"

AutoGenerateColumns="False" SelectionMode="Single" AlternationCount="1" RowEditEnding="DtServiceCategory\_RowEditEnding">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding ServiceCategoryID}" Header = "ID" Visibility="Hidden" />

<DataGridTextColumn Binding="{Binding ServiceCategoryName}" Header = "Вид сервиса" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

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

<Button x:Name="btnSave" Content="Сохранить" Click="BtnSave\_Click" />

<Button x:Name="btnDelete" Content="Удалить" Click="BtnDelete\_Click"/>

</StackPanel>

<materialDesign:DialogHost x:Name="dialogHostAdd" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Изменения сохранены" Margin="20" />

<Button x:Name="btnOk" Click="BtnOk\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost x:Name="DialogHost" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите удалить запись?" Margin="20" />

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

<Button x:Name="btnYes" Click="BtnYes\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Page>

**ServiceCategoryPage.cs**

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 System.Data.Entity;

using AirSystemApp.Models;

namespace AirSystemApp.Pages

{

/// <summary>

/// Логика взаимодействия для ServiceCategoryPage.xaml

/// </summary>

public partial class ServiceCategoryPage : Page

{

AirDBEntities db;

public ServiceCategoryPage()

{

InitializeComponent();

LoadData();

}

void LoadData()

{

db = new AirDBEntities();

db.ServiceCategories.Load();

dtServiceCategory.ItemsSource = db.ServiceCategories.Local.ToBindingList();

}

void SaveData()

{

try

{

db.SaveChanges();

dialogHostAdd.IsOpen = true;

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

}

finally

{

db.Dispose();

LoadData();

}

}

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

{

SaveData();

}

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

{

DialogHost.IsOpen = true;

}

private void DtServiceCategory\_RowEditEnding(object sender, DataGridRowEditEndingEventArgs e)

{

//SaveData();

}

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

{

try

{

if (dtServiceCategory.SelectedItems.Count > 0)

{

for (int i = 0; i < dtServiceCategory.SelectedItems.Count; i++)

{

ServiceCategory serviceCategory = dtServiceCategory.SelectedItems[i] as ServiceCategory;

if (serviceCategory != null)

{

db.ServiceCategories.Remove(serviceCategory);

}

}

}

db.SaveChanges();

DialogHost.IsOpen = false;

//MaterialDesignThemes.Wpf.DialogHost.Show("Запись вфыафыва");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

db.Dispose();

LoadData();

}

}

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

{

DialogHost.IsOpen = false;

}

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

{

dialogHostAdd.IsOpen = false;

}

}

}

**ServicePage.xaml**

<Page x:Class="AirSystemApp.Pages.ServicePage"

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:AirSystemApp.Pages"

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

mc:Ignorable="d"

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

Title="ServiceCategoryPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid Grid.Row="0" x:Name="dtServices" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" ColumnHeaderStyle="{DynamicResource header\_column}"

AutoGenerateColumns="False" SelectionMode="Single" AlternationCount="1" RowEditEnding="DtServiceCategory\_RowEditEnding">

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding ServiceID}" Header = "ID" Width="40" />

<DataGridTextColumn Binding="{Binding ServiceName}" Header = "Cервис" Width="3\*">

<DataGridTextColumn.ElementStyle>

<Style>

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

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

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<!--<DataGridTextColumn Binding="{Binding ServicePrice}" Header = "Описание" Width="10\*"/>-->

<DataGridTemplateColumn Header="Стоимость" Width="100">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBox x:Name="tbPrice" Text="{Binding ServicePrice}" PreviewTextInput="TbPrice\_PreviewTextInput" TextChanged="TbPrice\_TextChanged" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Binding="{Binding ServiceDescription}" Header = "Описание" Width="10\*" >

<DataGridTextColumn.ElementStyle>

<Style>

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

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

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<!--<DataGridTextColumn Binding="{Binding ServiceCategory.ServiceCategoryName}" Header = "Вид сервиса" Width="120"/>-->

<DataGridComboBoxColumn x:Name="colServiceCategories"

SelectedValueBinding="{Binding ServiceCategoryID}"

DisplayMemberPath="ServiceCategoryName"

SelectedValuePath="ServiceCategoryID" Width="3\*"/>

</DataGrid.Columns>

</DataGrid>

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

<Button x:Name="btnSave" Content="Сохранить" Click="BtnSave\_Click" />

<Button x:Name="btnDelete" Content="Удалить" Click="BtnDelete\_Click"/>

</StackPanel>

<materialDesign:DialogHost x:Name="dialogHostAdd" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Изменения сохранены" Margin="20" />

<Button x:Name="btnOk" Click="BtnOk\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost x:Name="DialogHost" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите удалить запись?" Margin="20" />

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

<Button x:Name="btnYes" Click="BtnYes\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Page>

**ServicePage.cs**

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 System.Data.Entity;

using AirSystemApp.Models;

using System.Text.RegularExpressions;

namespace AirSystemApp.Pages

{

/// <summary>

/// Логика взаимодействия для ServicePage.xaml

/// </summary>

public partial class ServicePage : Page

{

AirDBEntities db;

public ServicePage()

{

InitializeComponent();

LoadData();

}

void LoadData()

{

db = new AirDBEntities();

db.Services.Load(); db.ServiceCategories.Load();

dtServices.ItemsSource = db.Services.Local.ToBindingList();

colServiceCategories.ItemsSource = db.ServiceCategories.ToList();

}

void SaveData()

{

try

{

db.SaveChanges();

dialogHostAdd.IsOpen = true;

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

}

finally

{

//db.Dispose();

//LoadData();

}

}

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

{

SaveData();

}

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

{

DialogHost.IsOpen = true;

}

private void DtServiceCategory\_RowEditEnding(object sender, DataGridRowEditEndingEventArgs e)

{

//SaveData();

}

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

{

try

{

if (dtServices.SelectedItems.Count > 0)

{

for (int i = 0; i < dtServices.SelectedItems.Count; i++)

{

Service service = dtServices.SelectedItems[i] as Service;

if (service != null)

{

db.Services.Remove(service);

}

}

}

db.SaveChanges();

DialogHost.IsOpen = false;

//MaterialDesignThemes.Wpf.DialogHost.Show("Запись вфыафыва");

}

catch (Exception ex)

{

MessageBox.Show("Ошибка");

db.Dispose();

LoadData();

}

}

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

{

DialogHost.IsOpen = false;

}

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

{

dialogHostAdd.IsOpen = false;

}

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

{

//if (e.Text.Contains("."))

// e.Handled = true;

//else

e.Handled = ! new Regex(@"^[0-9]\*(?:\.[0-9]\*)?$").IsMatch(e.Text);

} }}

OrderPage.xaml

<Page x:Class="AirSystemApp.Pages.OrdersPage"

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:AirSystemApp.Pages"

mc:Ignorable="d"

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

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

Title="OrdersPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<DataGrid Grid.Row="0" x:Name="dtOrders" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" ColumnHeaderStyle="{DynamicResource header\_column}"

AutoGenerateColumns="False" AlternationCount="1" SelectionMode="Single" CanUserAddRows="False" IsReadOnly="True" >

<DataGrid.RowStyle>

<Style TargetType="DataGridRow">

<Setter Property="Background" Value="{Binding Status.Color}"/>

</Style>

</DataGrid.RowStyle>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding OrderID}" Header = "ID" Width="40" />

<DataGridTextColumn Binding="{Binding StartDate, StringFormat='d', ConverterCulture=ru-RU}" Header = "Дата подачи заявки" Width="3\*"/>

<DataGridTextColumn Binding="{Binding EndDate, StringFormat='d', ConverterCulture=ru-RU}" Header = "Дата выполнения" Width="3\*"/>

<DataGridTextColumn Binding="{Binding Client.LastName}" Header = "Клиент" Width="3\*"/>

<DataGridTextColumn Binding="{Binding Status.StatusName}" Header = "Статус" Width="3\*"/>

<DataGridTextColumn Binding="{Binding Address}" Header = "Адрес" Width="3\*"/>

<DataGridTextColumn Binding="{Binding TotalPrice, StringFormat='c', ConverterCulture=ru-RU}" Header = "Цена" Width="3\*"/>

<!--<DataGridTextColumn Binding="{Binding Photo}" Header = "Изображение" Width="2\*"/>-->

</DataGrid.Columns>

</DataGrid>

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

<Button x:Name="btnAdd" Content="Добавить" Click="BtnAdd\_Click" />

<Button x:Name="btnSave" Content="Изменить" Click="BtnSave\_Click" />

<Button x:Name="btnDelete" Content="Удалить" Click="BtnDelete\_Click"/>

<StackPanel Orientation="Horizontal">

<ComboBox x:Name="cmbSearchType" Background="White" VerticalAlignment="Stretch" VerticalContentAlignment="Center" Width="200" SelectedIndex="0" Padding="10,0,0,0">

<ComboBoxItem Content="Поиск по ID"/>

<ComboBoxItem Content="Поиск по клиенту"/>

<ComboBoxItem Content="Поиск по дате подачи заявки"/>

<ComboBoxItem Content="Поиск адресу"/>

</ComboBox>

<TextBox x:Name="tbSearchID" Width=" 100" BorderBrush="{DynamicResource dark}" BorderThickness="1" Background="White"/>

<Button x:Name="btnSearch" Width="60" Click="BtnSearch\_Click" Background="White" BorderBrush="Black" >

<materialDesign:PackIcon Kind="Search" Foreground="#FF0000" VerticalAlignment="Center"/>

</Button>

<Button x:Name="btnClearSearch" Width="60" Click="BtnClearSearch\_Click" Background="White" BorderBrush="Black" >

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" VerticalAlignment="Center"/>

</Button>

<Button x:Name="btnExcel" Width="60" Background="White" BorderBrush="Black" Click="BtnExcel\_Click" >

<materialDesign:PackIcon Kind="FileExcel" Foreground="#FF087008" VerticalAlignment="Center"/>

</Button>

</StackPanel>

</StackPanel>

<materialDesign:DialogHost x:Name="dialogHostAdd" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Изменения сохранены" Margin="20" />

<Button x:Name="btnOk" Click="BtnOk\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="0" Grid.ColumnSpan="2" x:Name="resultDialog" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<TextBlock x:Name="tbMessage" Text="" VerticalAlignment="Top"

HorizontalAlignment="Center" TextWrapping="Wrap"/>

<Button x:Name="btnOkResult" Click="BtnOkResult\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="0" Grid.ColumnSpan="2" x:Name="errorDialog" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<StackPanel Orientation="Horizontal" Margin="5" HorizontalAlignment="Center">

<Label Content="Ошибка" FontSize="14" FontWeight="Bold"/>

<materialDesign:PackIcon Kind="Error" Foreground="#FF0000" HorizontalAlignment="Center"/>

</StackPanel>

<TextBlock x:Name="tbError" Text="" VerticalAlignment="Top"

HorizontalAlignment="Center" TextWrapping="Wrap" Margin="5"/>

<Button x:Name="btnOkError" Click="BtnOkError\_Click" Background="White" Width="100" Margin="5">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost x:Name="DialogHost" Background="{DynamicResource gray}" BorderBrush="{DynamicResource blue}" BorderThickness="1" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите удалить запись?" Margin="20" />

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

<Button x:Name="btnYes" Click="BtnYes\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Page>

**OrderPage.cs**

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 System.Data.Entity;

using AirSystemApp.Models;

using System.ComponentModel;

using Excel = Microsoft.Office.Interop.Excel;

using System.Collections;

namespace AirSystemApp.Pages

{

/// <summary>

/// Логика взаимодействия для AirPage.xaml

/// </summary>

public partial class OrdersPage : Page

{

//AirDBEntities db;

ICollectionView collectionView;

string imagePath = AppDomain.CurrentDomain.BaseDirectory + "Images/";

public OrdersPage()

{

InitializeComponent();

LoadData();

}

void LoadData()

{

using (AirDBEntities db = new AirDBEntities())

{

db.Orders.Load();

var orders = db.Orders.Include(order => order.Client).Include(order => order.Status).ToList();

collectionView = CollectionViewSource.GetDefaultView(orders);

dtOrders.ItemsSource = collectionView;

}

}

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

{

try

{

if (dtOrders.SelectedItems.Count > 0)

{

for (int i = 0; i < dtOrders.SelectedItems.Count; i++)

{

Order order = dtOrders.SelectedItems[i] as Order;

if (order != null)

{

int id = order.OrderID;

using (AirDBEntities db = new AirDBEntities())

{

Order a = db.Orders.FirstOrDefault(x => x.OrderID == id);

db.Orders.Remove(a);

db.SaveChanges();

}

}

}

}

DialogHost.IsOpen = false;

//MaterialDesignThemes.Wpf.DialogHost.Show("Запись вфыафыва");

}

catch (Exception ex)

{

DialogHost.IsOpen = false;

// MessageBox.Show("Удаление прервано, есть связанные записи", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

tbError.Text = "Удаление прервано, есть связанные записи";

errorDialog.IsOpen = true;

}

finally

{

LoadData();

}

}

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

{

DialogHost.IsOpen = false;

}

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

{

dialogHostAdd.IsOpen = false;

}

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

{

AddOrderWindow addOrderWindow = new AddOrderWindow();

addOrderWindow.ShowDialog();

LoadData();

}

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

{

//try

//{

if (dtOrders.SelectedItems.Count > 0)

{

for (int i = 0; i < dtOrders.SelectedItems.Count; i++)

{

Order order = dtOrders.SelectedItems[i] as Order;

if (order != null)

{

AddOrderWindow addOrderWindow = new AddOrderWindow(order.OrderID);

addOrderWindow.ShowDialog();

LoadData();

}

}

}

//}

//catch (Exception ex)

//{

// MessageBox.Show("Ошибка");

//}

}

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

{

if (dtOrders.SelectedItems.Count > 0)

DialogHost.IsOpen = true;

}

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

{

resultDialog.IsOpen = false;

}

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

{

int selind = cmbSearchType.SelectedIndex;

if (tbSearchID.Text == "")

{

collectionView.Filter = null;

return;

}

switch (selind)

{

case 0:

FilterByID(tbSearchID.Text);

break;

case 1:

FilterByClient(tbSearchID.Text);

break;

case 2:

FilterByDate(tbSearchID.Text);

break;

case 3:

FilterByAddress(tbSearchID.Text);

break;

default: collectionView.Filter = null; break;

}

}

void FilterByID(string s)

{

int id = -1;

bool b = int.TryParse(s, out id);

if (!b)

collectionView.Filter = null;

collectionView.Filter = item =>

{

Order x = item as Order;

return x.OrderID == id;

};

collectionView.Refresh();

}

void FilterByClient(string s)

{

collectionView.Filter = item =>

{

Order x = item as Order;

//return x.OrderID == id;

return x.Client.LastName.ToLower().Contains(s.ToLower());

};

collectionView.Refresh();

}

void FilterByAddress(string s)

{

collectionView.Filter = item =>

{

Order x = item as Order;

//return x.OrderID == id;

return x.Address.ToLower().Contains(s.ToLower());

};

collectionView.Refresh();

}

void FilterByDate(string s)

{

DateTime y = DateTime.Now;

bool b = DateTime.TryParse(s, out y);

if (b == false)

{

return;

}

collectionView.Filter = item =>

{

Order x = item as Order;

//return x.OrderID == id;

return x.StartDate >= y;

};

collectionView.Refresh();

}

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

{

collectionView.Filter = null;

collectionView.Refresh();

}

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

{

PrintExcel();

}

private void PrintExcel()

{

string fileName = AppDomain.CurrentDomain.BaseDirectory + "\\" + "Orders" + ".xltx";

Excel.Application xlApp = new Excel.Application();

Excel.Worksheet xlSheet = new Excel.Worksheet();

try

{

//добавляем книгу

xlApp.Workbooks.Open(fileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

//делаем временно неактивным документ

xlApp.Interactive = false;

xlApp.EnableEvents = false;

Excel.Range xlSheetRange;

//выбираем лист на котором будем работать (Лист 1)

xlSheet = (Excel.Worksheet)xlApp.Sheets[1];

//Название листа

xlSheet.Name = "Список заявок";

int row = 2;

int i = 0;

if (dtOrders.Items.Count > 0)

{

for (i = 0; i < dtOrders.Items.Count; i++)

{

Order order = dtOrders.Items[i] as Order;

xlSheet.Cells[row, 1] = (i + 1).ToString();

// DateTime y = Convert.ToDateTime(dtOrders.Rows[i].Cells[1].Value);

xlSheet.Cells[row, 2] = order.OrderID.ToString();

xlSheet.Cells[row, 3] = order.StartDate.ToShortDateString();

xlSheet.Cells[row, 4] = order.EndDate.ToShortDateString();

xlSheet.Cells[row, 5] = order.Client.LastName.ToString();

xlSheet.Cells[row, 6] = order.Status.StatusName.ToString();

xlSheet.Cells[row, 7] = order.Address.ToString();

xlSheet.Cells[row, 8] = order.TotalPrice.ToString();

row++;

Excel.Range r = xlSheet.get\_Range("A" + row.ToString(), "H" + row.ToString());

r.Insert(Excel.XlInsertShiftDirection.xlShiftDown);

}

}

row--;

xlSheetRange = xlSheet.get\_Range("A2:H" + (row+1).ToString(), Type.Missing);

xlSheetRange.Borders.LineStyle = true;

xlSheet.Cells[row+1, 8] = "=SUM(H2:h" + row.ToString() + ")";

xlSheet.Cells[row + 1, 7] = "ИТОГО:";

row++;

//выбираем всю область данных\*/

xlSheetRange = xlSheet.UsedRange;

//выравниваем строки и колонки по их содержимому

//xlSheetRange.Columns.AutoFit();

//xlSheetRange.Rows.AutoFit();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

//Показываем ексель

xlApp.Visible = true;

xlApp.Interactive = true;

xlApp.ScreenUpdating = true;

xlApp.UserControl = true;

}

}

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

{

errorDialog.IsOpen = false; } }}

AddOrderWindow.xaml

<Window x:Class="AirSystemApp.Pages.AddOrderWindow"

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:AirSystemApp.Pages"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes" xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

mc:Ignorable="d" ResizeMode="NoResize"

WindowStyle="ToolWindow"

BorderThickness="0"

BorderBrush="{DynamicResource lemon}"

Title="Договор" Height="840" Width="610" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0"/>

<RowDefinition Height="40" x:Name="rwOrderNum"/>

<RowDefinition Height="80"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40"/>

<RowDefinition Height="40" x:Name="rwPrice"/>

<RowDefinition Height="30" x:Name="rwAirsName"/>

<RowDefinition Height="180" x:Name="rwAirs"/>

<RowDefinition Height="30" x:Name="rwServicesName"/>

<RowDefinition Height="180" x:Name="rwServices"/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Rectangle Grid.Row="0" Grid.ColumnSpan="2" Fill="{DynamicResource dark}"/>

<Button Grid.Row="0" Grid.ColumnSpan="2" x:Name="btnExit" ToolTip="Закрыть приложение" HorizontalAlignment="Right" VerticalAlignment="Top" Background="{x:Null}" BorderBrush="{DynamicResource gray}" Width="30" Height="30" Padding="0" Click="BtnExit\_Click">

<materialDesign:PackIcon Kind="Close" />

</Button>

<Button x:Name="btnSave" Grid.Row="12" Content="Сохранить" HorizontalAlignment="Center" VerticalAlignment="Top" Width="120" Click="BtnSave\_Click" Grid.ColumnSpan="2" Margin="10"/>

<Label Grid.Row="1" Grid.Column="0" Content="Номер договора:" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<StackPanel Grid.Row="2" Grid.Column="0" Orientation="Horizontal">

<Label Content="Клиент:" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<Button x:Name="btnLoadClient" VerticalAlignment="Top" HorizontalAlignment="Right" Padding="0" Width="40" Click="btnLoadClient\_Click" >

<materialDesign:PackIcon Kind="Search"/>

</Button>

</StackPanel>

<Label Grid.Row="3" Grid.Column="0" Content="Дата подачи:" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<Label Grid.Row="4" Grid.Column="0" Content="Дата выполнения:" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<Label Grid.Row="5" Grid.Column="0" Content="Статус заявки:" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<Label Grid.Row="6" Grid.Column="0" Content="Адрес:" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<Label Grid.Row="7" Grid.Column="0" Content="Стоимость:" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="20,0" FontSize="18"/>

<TextBox x:Name="tbOrderId" Grid.Row="1" Grid.Column="1" materialDesign:HintAssist.Hint="Номер договора" FontSize="18" Padding="0,2"/>

<TextBox x:Name="tbClient" Grid.Row="2" Grid.Column="1" materialDesign:HintAssist.Hint="Клиент" FontSize="18" Padding="0,2"/>

<DatePicker x:Name="tbStartDate" Grid.Row="3" Grid.Column="1" materialDesign:HintAssist.Hint="Дата подачи" FontSize="18" Padding="0,2"/>

<DatePicker x:Name="tbEndDate" Grid.Row="4" Grid.Column="1" materialDesign:HintAssist.Hint="Дата выполнения" FontSize="18" Padding="0,2"/>

<ComboBox x:Name="tbStatus" Grid.Row="5" Grid.Column="1" materialDesign:HintAssist.Hint="Статус" FontSize="18" Padding="0,2" DisplayMemberPath="StatusName" SelectedValuePath="StatusID"/>

<TextBox x:Name="tbAddress" Grid.Row="6" Grid.Column="1" materialDesign:HintAssist.Hint="Адрес" FontSize="18" Padding="0,2"/>

<xctk:DoubleUpDown x:Name="tbTotalPrice" Grid.Row="7" Grid.Column="1" materialDesign:HintAssist.Hint="Стоимость" FontSize="18" Padding="0,2" FormatString="C" DefaultValue="0" DisplayDefaultValueOnEmptyText="True" IsReadOnly="True" Minimum="0"/>

<TextBlock Grid.Row="8" Grid.ColumnSpan="2" Background="{DynamicResource dark}" Foreground="{DynamicResource light\_blue}" FontSize="18" Text="Оборудование" TextAlignment="Center" Padding="0,3,0,0" FontWeight="Bold"/>

<DataGrid x:Name="dtAirs" Grid.Row="9" Grid.ColumnSpan="2" AutoGenerateColumns="False" Width="560" HorizontalAlignment="Left" CanUserAddRows="False" LoadingRow="dtAirs\_LoadingRow" GridLinesVisibility="Horizontal" HeadersVisibility="All" >

<DataGrid.Columns>

<materialDesign:DataGridTextColumn Binding="{Binding AirOrderID}" Header="ID" Width="\*" Visibility="Hidden"/>

<materialDesign:DataGridTextColumn Binding="{Binding AirName}" Header="Кондиционер" Width="\*"/>

<materialDesign:DataGridTextColumn Binding="{Binding Price}" Header="Стоимость" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="9" Grid.ColumnSpan="2" HorizontalAlignment="Right" Orientation="Vertical" Width="40">

<Button x:Name="btnAddAir" Padding="0" Click="BtnAddAir\_Click" >

<materialDesign:PackIcon Kind="Add"/>

</Button>

<Button x:Name="btnDeleteAir" Padding="0" Click="BtnDeleteAir\_Click" >

<materialDesign:PackIcon Kind="Delete"/>

</Button>

</StackPanel>

<TextBlock Grid.Row="10" Grid.ColumnSpan="2" Background="{DynamicResource dark}" Foreground="{DynamicResource light\_blue}" FontSize="18" Text="Сервисы" TextAlignment="Center" Padding="0,3,0,0" FontWeight="Bold"/>

<DataGrid x:Name="dtServices" Grid.Row="11" Grid.ColumnSpan="2" Margin="0,0,40,0" AutoGenerateColumns="False" LoadingRow="dtServices\_LoadingRow" HeadersVisibility="All" CanUserAddRows="False">

<DataGrid.Columns>

<materialDesign:DataGridTextColumn Binding="{Binding ServiceOrderID}" Header="ID" Width="\*" Visibility="Hidden"/>

<materialDesign:DataGridTextColumn Binding="{Binding ServiceName}" Header="Сервис" Width="\*"/>

<materialDesign:DataGridTextColumn Binding="{Binding ServicePrice}" Header="Стоимость" Width="\*"/>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="11" Grid.ColumnSpan="2" HorizontalAlignment="Right" Orientation="Vertical" Width="40">

<Button x:Name="btnAddService" Padding="0" Click="BtnAddService\_Click" >

<materialDesign:PackIcon Kind="Add"/>

</Button>

<Button x:Name="btnDeleteService" Padding="0" Click="BtnDeleteService\_Click">

<materialDesign:PackIcon Kind="Delete"/>

</Button>

</StackPanel>

<materialDesign:DialogHost Grid.Row="4" Grid.ColumnSpan="2" x:Name="DialogHost" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<TextBlock x:Name="tbError" Text="" VerticalAlignment="Top"

HorizontalAlignment="Center" TextWrapping="Wrap"/>

<Button x:Name="btnOk" Click="BtnOk\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="4" Grid.ColumnSpan="2" x:Name="AddAirHost" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<ListBox x:Name="lbAirs" Grid.Row="2" ScrollViewer.HorizontalScrollBarVisibility="Auto" SelectedValuePath="AirID">

<ListBox.ItemTemplate>

<DataTemplate>

<StackPanel Margin="5" Orientation="Horizontal">

<Image Width="100" Height="60" Source="{Binding Path=Photo}" />

<StackPanel>

<TextBlock FontSize="18" Text="{Binding Path=AirName}" HorizontalAlignment="Left" />

<TextBlock FontWeight="Bold" FontSize="14" Text="{Binding Path=Price}" HorizontalAlignment="Left" TextWrapping="WrapWithOverflow" Width="500" />

</StackPanel>

</StackPanel>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

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

<Button x:Name="btnAddOk" Click="BtnAddOk\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancel" Click="BtnCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="4" Grid.ColumnSpan="2" x:Name="AddServiceHost" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<ListBox x:Name="lbService" Grid.Row="2" ScrollViewer.HorizontalScrollBarVisibility="Auto" SelectedValuePath="AirID" Height="400" IsTextSearchCaseSensitive="True" SelectionMode="Multiple">

<ListBox.ItemTemplate>

<DataTemplate>

<StackPanel>

<TextBlock FontSize="18" Text="{Binding Path=ServiceName}" HorizontalAlignment="Left" />

<TextBlock FontWeight="Bold" FontSize="14" Text="{Binding Path=ServicePrice}" HorizontalAlignment="Left" TextWrapping="WrapWithOverflow" Width="500" />

</StackPanel>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

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

<Button x:Name="btnServiceOK" Click="BtnServiceOK\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnServiceCancel" Click="BtnServiceCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="4" Grid.ColumnSpan="2" x:Name="hostLoadClient" Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<ListBox x:Name="lbClient" Grid.Row="2" ScrollViewer.HorizontalScrollBarVisibility="Auto" SelectedValuePath="AirID">

<ListBox.ItemTemplate>

<DataTemplate>

<StackPanel>

<TextBlock FontSize="18" Text="{Binding Path=ClientID}" HorizontalAlignment="Left" />

<TextBlock FontSize="18" Text="{Binding Path=LastName}" HorizontalAlignment="Left" />

<TextBlock FontSize="14" Text="{Binding Path=Phone}" HorizontalAlignment="Left" TextWrapping="WrapWithOverflow" Width="500" />

</StackPanel>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

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

<Button x:Name="btnClientOK" Click="btnClientOK\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnClientCancel" Click="btnClientCancel\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost x:Name="dialogDeleteAirOrder"

Grid.Row="4" Grid.ColumnSpan="2"

Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

CloseOnClickAway="True"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите удалить запись?" Margin="20" />

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

<Button x:Name="btnYesDeleteAirOrder" Click="BtnYesDeleteAirOrder\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancelDeleteAirOrder" Click="BtnCancelDeleteAirOrder\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost x:Name="dialogDeleteServiceOrder"

Grid.Row="4" Grid.ColumnSpan="2"

Background="{DynamicResource gray}"

BorderBrush="{DynamicResource blue}"

CloseOnClickAway="True"

BorderThickness="1"

HorizontalAlignment="Center"

VerticalAlignment="Center" >

<materialDesign:DialogHost.DialogContent>

<StackPanel HorizontalAlignment="Center" Margin="20">

<TextBlock Text="Вы действительно хотите удалить запись?" Margin="20" />

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

<Button x:Name="btnYesDeleteServiceOrder" Click="BtnYesDeleteServiceOrder\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Check" Foreground="#00FF00" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

<Button x:Name="btnCancelDeleteServiceOrder" Click="BtnCancelDeleteServiceOrder\_Click" Background="White">

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

<materialDesign:PackIcon Kind="Cancel" Foreground="#FF0000" Margin="5" VerticalAlignment="Center"/>

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

</StackPanel>

</Button>

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Window>

**AddOrderWindow.cs**

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;

using AirSystemApp.Models;

using System.Data.Entity;

using System.Text.RegularExpressions;

namespace AirSystemApp.Pages

{

/// <summary>

/// Логика взаимодействия для AddOrderWindow.xaml

/// </summary>

public partial class AddOrderWindow : Window

{

private class AirOrderIn

{

public int AirOrderId { set; get; }

public string AirName { set; get; }

public double Price { set; get; }

}

private class ServiceOrderIn

{

public int ServiceOrderId { set; get; }

public string ServiceName { set; get; }

public double ServicePrice { set; get; }

}

private class ClientInfo

{

public int ClientID { set; get; }

public string LastName { set; get; }

public double Phone { set; get; }

}

Order order=null;

int ClientID = -1;

public AddOrderWindow()

{

InitializeComponent();

LoadStatus();

LoadAir();

LoadClients();

LoadServices();

DisableComponent();

} void DisableComponent()

{

rwOrderNum.Height = new GridLength(0);

rwPrice.Height = new GridLength(0);

rwAirs.Height = new GridLength(0);

rwServices.Height = new GridLength(0);

rwAirsName.Height = new GridLength(0);

rwServicesName.Height = new GridLength(0);

this.Height = 420;

btnSave.Content = "Создать";

}

void EnableComponent()

{

rwAirs.Height = new GridLength(180);

rwServices.Height = new GridLength(180);

rwAirsName.Height = new GridLength(30);

rwServicesName.Height = new GridLength(30);

rwOrderNum.Height = new GridLength(40);

rwPrice.Height = new GridLength(40);

this.Height = 840;

btnSave.Content = "Сохранить"; }

public AddOrderWindow(int orderId)

{

InitializeComponent();

LoadStatus();

LoadOrder(orderId);

LoadAir();

LoadAirOrder();

LoadClients();

LoadServices();

LoadServiceOrder(); }

void LoadClients()

{

using (AirDBEntities db = new AirDBEntities())

{

lbClient.ItemsSource = db.Clients.ToList();

} }

void LoadStatus()

{

using (AirDBEntities db = new AirDBEntities())

{

tbStatus.ItemsSource = db.Status.ToList();

} }

void LoadServices()

{

using (AirDBEntities db = new AirDBEntities())

{

db.Services.Load();

lbService.ItemsSource = db.Services.ToList();

} }

void LoadAir()

{

string imagePath = AppDomain.CurrentDomain.BaseDirectory + "Images/";

using (AirDBEntities db = new AirDBEntities())

{

db.Airs.Load();

List<Air> airs = db.Airs.ToList();

foreach (Air a in airs)

{

a.Photo = imagePath + a.Photo;

} lbAirs.ItemsSource = airs;

} }

void LoadAirOrder()

{

using (AirDBEntities db = new AirDBEntities())

{

var airOrdersList = (from air in db.Airs

join airorder in db.AirOrders on air.AirID equals airorder.AirID

where airorder.OrderID == order.OrderID

select new AirOrderIn

{

AirOrderId = airorder.AirOrderID,

AirName = air.AirName,

Price = air.Price

}).ToList();

dtAirs.ItemsSource = airOrdersList;

} }

void LoadServiceOrder()

{

using (AirDBEntities db = new AirDBEntities())

{

var serviceOrdersList = (from service in db.Services

join serviceorder in db.ServiceOrders on service.ServiceID equals serviceorder.ServiceID

where serviceorder.OrderID == order.OrderID

select new ServiceOrderIn

{

ServiceOrderId = serviceorder.ServiceOrderID,

ServiceName = service.ServiceName,

ServicePrice = service.ServicePrice

}).ToList();

dtServices.ItemsSource = serviceOrdersList;

} } void LoadOrder(int orderId)

{

using (AirDBEntities db = new AirDBEntities())

{

order = db.Orders.Include(order => order.Client).Include(order => order.AirOrders).Include(order => order.ServiceOrders).FirstOrDefault(d => d.OrderID == orderId);

tbStatus.ItemsSource = db.Status.ToList();

}

ClientID = Convert.ToInt32(order.ClientID);

tbOrderId.Text = order.OrderID.ToString();

tbClient.Text =$" {order.Client.LastName} {order.Client.FirstName}\nтелефон: {order.Client.Phone}, email:{order.Client.Email}";

//tbClient.Text =

//order.Client.LastName;

tbStartDate.SelectedDate = order.StartDate;

tbEndDate.SelectedDate = order.EndDate;

tbStatus.SelectedValue = order.StatusID;

tbAddress.Text = order.Address;

tbTotalPrice.Value = order.TotalPrice; }

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

{

this.Close();

}

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

{

DialogHost.IsOpen = false;

}

private void dtAirs\_LoadingRow(object sender, DataGridRowEventArgs e)

{

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

}

private void dtServices\_LoadingRow(object sender, DataGridRowEventArgs e)

{

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

}

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

{

AddAirHost.IsOpen = true;

}

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

{

AddAirHost.IsOpen = false;

}

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

{

try

{

if (lbAirs.SelectedItems.Count > 0)

{

for (int i = 0; i < lbAirs.SelectedItems.Count; i++)

{

Air xair = lbAirs.SelectedItems[i] as Air;

if (xair != null)

{

int id = xair.AirID;

using (AirDBEntities db = new AirDBEntities())

{

AirOrder airOrder = new AirOrder

{

OrderID = order.OrderID,

AirID = id

};

db.AirOrders.Add(airOrder);

tbTotalPrice.Value += xair.Price;

db.SaveChanges();

db.Orders.Attach(order);

order.TotalPrice = tbTotalPrice.Value;

db.SaveChanges();

} } } }

AddAirHost.IsOpen = false;

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{ LoadAirOrder();

} }

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

{

dialogDeleteAirOrder.IsOpen = true;

}

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

{

try

{

if (lbService.SelectedItems.Count > 0)

{

for (int i = 0; i < lbService.SelectedItems.Count; i++)

{

Service xair = lbService.SelectedItems[i] as Service;

if (xair != null)

{

int id = xair.ServiceID;

using (AirDBEntities db = new AirDBEntities())

{

ServiceOrder serviceOrder = new ServiceOrder

{

OrderID = order.OrderID,

ServiceID = id

};

tbTotalPrice.Value += xair.ServicePrice;

db.ServiceOrders.Add(serviceOrder);

db.SaveChanges();

db.Orders.Attach(order);

order.TotalPrice = tbTotalPrice.Value;

db.SaveChanges();

} } } }

AddServiceHost.IsOpen = false;

//MaterialDesignThemes.Wpf.DialogHost.Show("Запись вфыафыва");

} catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

LoadServiceOrder();

} }

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

{

AddServiceHost.IsOpen = false;

}

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

{

AddServiceHost.IsOpen = true;

}

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

{

try

{

if (dtAirs.SelectedItems.Count > 0)

{

for (int i = 0; i < dtAirs.SelectedItems.Count; i++)

{

AirOrderIn xair = dtAirs.SelectedItems[i] as AirOrderIn;

if (xair != null)

{

int id = xair.AirOrderId;

using (AirDBEntities db = new AirDBEntities())

{

AirOrder x = db.AirOrders.FirstOrDefault(a => a.AirOrderID == id);

db.AirOrders.Attach(x);

db.AirOrders.Remove(x);

db.SaveChanges();

tbTotalPrice.Value -= xair.Price;

db.Orders.Attach(order);

order.TotalPrice = tbTotalPrice.Value;

db.SaveChanges();

} } } }

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

LoadAirOrder();

dialogDeleteAirOrder.IsOpen = false;

} }

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

{

dialogDeleteAirOrder.IsOpen = false;

}

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

{

dialogDeleteServiceOrder.IsOpen = true;

}

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

{

dialogDeleteServiceOrder.IsOpen = false;

}

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

{

try

{

if (dtServices.SelectedItems.Count > 0)

{

for (int i = 0; i < dtServices.SelectedItems.Count; i++)

{

ServiceOrderIn xair = dtServices.SelectedItems[i] as ServiceOrderIn;

if (xair != null)

{

int id = xair.ServiceOrderId;

using (AirDBEntities db = new AirDBEntities())

{

ServiceOrder x = db.ServiceOrders.FirstOrDefault(a => a.ServiceOrderID == id);

db.ServiceOrders.Attach(x);

db.ServiceOrders.Remove(x);

db.SaveChanges();

tbTotalPrice.Value -= xair.ServicePrice;

db.Orders.Attach(order);

order.TotalPrice = tbTotalPrice.Value;

db.SaveChanges();

} } } }

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

LoadServiceOrder();

dialogDeleteServiceOrder.IsOpen = false;

} }

private string CheckFields()

{

//tbOrderId.Text = order.OrderID.ToString();

////tbClient.Text =$" {order.Client.LastName} {order.Client.FirstName}\nтелефон: {order.Client.Phone}, email:{order.Client.Email}";

//tbClient.Text =

//order.Client.LastName;

//tbStartDate.SelectedDate = order.StartDate;

//tbEndDate.SelectedDate = order.EndDate;

//tbStatus.SelectedValue = order.StatusID;

//tbAddress.Text = order.Address;

//tbTotalPrice.Value = order.TotalPrice;

string s = "";

if (ClientID == -1)

s += "Клиент не выбран\n";

if (tbStartDate.Text == "")

s += "Дата не выбрана\n";

if (tbEndDate.Text == "")

s += "Дата не выбрана\n";

if (tbStatus.SelectedIndex == -1)

s += "Статус пустое\n";

if (tbAddress.Text == "")

s += "Адрес не указан\n";

return s;

}

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

{

string fields = CheckFields();

if (fields != "")

{

tbError.Text = CheckFields();

DialogHost.IsOpen = true;

return; }

if (order == null)

{

AddOrder();

}

else { UpdateOrder(); } }

void AddOrder()

{

using (AirDBEntities db = new AirDBEntities())

{

Order new\_order = new Order

{

ClientID = ClientID,

StartDate = Convert.ToDateTime(tbStartDate.SelectedDate),

EndDate = Convert.ToDateTime(tbEndDate.SelectedDate),

StatusID = Convert.ToInt32(tbStatus.SelectedValue),

Address = tbAddress.Text,

TotalPrice = tbTotalPrice.Value

};

db.Orders.Add(new\_order);

db.SaveChanges();

tbError.Text = "Запись добавлена";

order = new\_order;

LoadOrder(order.OrderID);

EnableComponent();

DialogHost.IsOpen = true;

} }

void UpdateOrder()

{

using (AirDBEntities db = new AirDBEntities())

{

db.Orders.Attach(order);

order.ClientID = ClientID;

order.StartDate = Convert.ToDateTime(tbStartDate.SelectedDate);

order.EndDate = Convert.ToDateTime(tbEndDate.SelectedDate);

order.StatusID = Convert.ToInt32(tbStatus.SelectedValue);

order.Address = tbAddress.Text;

order.TotalPrice = tbTotalPrice.Value;

db.SaveChanges();

tbError.Text = "Запись обновлена";

DialogHost.IsOpen = true; } }

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

{

hostLoadClient.IsOpen = true; }

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

{ try

{

if (lbClient.SelectedItems.Count > 0)

{

for (int i = 0; i < lbClient.SelectedItems.Count; i++)

{

Client xair = lbClient.SelectedItems[i] as Client;

if (xair != null)

{

ClientID = xair.ClientID;

tbClient.Text = $" {xair.LastName} {xair.FirstName}\nтелефон: {xair.Phone}, email:{xair.Email}";

} } }

} catch (Exception ex)

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

} hostLoadClient.IsOpen = false;

} private void btnClientCancel\_Click(object sender, RoutedEventArgs e)

{ hostLoadClient.IsOpen = false;

} }

}

**Приложение Б**

Фирма по продаже и обслуживанию кондиционеров

Спецификация

460.ЗМК.00011-01 01

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Обозначение** | | |  | | --- | | **Наименование** | | |  | | --- | | **Примечание** | |
|  | |  | | --- | | Документация | |  |
| 460.ЗМК.00011-01 12 01 | Фирма по продаже и обслуживанию кондиционеров |  |
|  | Текст программы |  |
| 460.ЗМК.00011-01 13 01 | Фирма по продаже и обслуживанию кондиционеров |  |
|  | Описание программы |  |
| 460.ЗМК.00011-01 34 01 | Фирма по продаже и обслуживанию кондиционеров |  |
|  | Руководство пользователя |  |
| 460.ЗМК.00011-01 81 01 | Фирма по продаже и обслуживанию кондиционеров |  |
|  | Пояснительная записка |  |

**Приложение B**

**Диск с программным средством**