**Содержание**

[**Введение** 4](#_Toc168414097)

[**Глава 1. Анализ предметной области и формулировка требований** 5](#_Toc168414098)

[**Глава 2. Анализ аналогов и прототипов** 11](#_Toc168414099)

[**Глава 3. Проектирование и программирование** 17](#_Toc168414100)

[Раздел 3.1. Описание интерфейса и архитектуры разрабатываемого продукта ……………………………………………………………………………..17](#_Toc168414101)

[Раздел 3.2. Разработка внутренней структуры и программирование …...22](#_Toc168414102)

[Раздел 3.3. Обеспечение безопасности информации ………...………......32](#_Toc168414103)

[Раздел 3.4. Тестирование и проверка работоспособности ……………….33](#_Toc168414104)

[**Глава 4. Аппаратные и программные средства для функционирования** 36](#_Toc168414105)

[**Глава 5. Инструкция пользователя и администратора системы** 39](#_Toc168414106)

[Заключение 70](#_Toc168414107)

[Список литературы 73](#_Toc168414108)

[Примечание 76](#_Toc168414109)

**Введение**

С каждым годом использование автоматизированных информационных в медицинской сфере становится все более популярным. Такие приложения помогают сотрудникам медицинских организаций получать быстрый доступ к различным базам данных, записывать пациентов на приемы, проводить медицинские обследования и многое другое.

Предпосылками автоматизации являются: замедление рабочих процессов организации, использование больших объемов трудовых и материальных ресурсов на ведение и контроль документов, поддержание данных о клиентах в достоверном состоянии.

Основными требованиями к системе являются:

* Уменьшение затрат на ведение информации предприятия;
* Оптимизация рабочих процессов организации;
* Автоматизация процессов добавления, изменения и удаления информации предприятия.

Целью дипломного проекта является разработка автоматизированной информационной системы, выполняющей описанные требования, а также облегчающей и упрощающей работу медицинского центра.

При разработке программного средства будет использоваться реляционная база данных.

Разрабатываемое приложение должно поддерживать современные версии Microsoft Windows 10 и Windows 11.

При работе над дипломным проектом будут использованы такие программные средства как:

* Microsoft Visual Studio 2019 (язык программирования C#, формат WPF xaml);
* Microsoft SQL Server Management Studio 2019.

**Глава 1. Анализ предметной области и формулировка требований**

**Назначение, цель и задачи создания приложения**

Предметной областью выступает медицинский центр “Гиппократ”, оказывающий частные медицинские услуги клиентам, такие как проведение анализов, диагностических исследований, комплексные обследования, вызов врача на дом, проведение предрейсовых осмотров.

На данный момент предприятие не имеет специализированной информационной системы, используются только бумажные документы и различные таблицы в Microsoft Office, исходя из этого информационная система, разработанная рамках дипломного проекта, будет предназначаться для медицинских работников и администраторов предприятия.

**Цель:** разработать автоматизированную информационную систему медицинского центра.

**Задачи:**

1. Формирование конкурентного имиджа медицинского центра.
2. Повышение объемов и качества обработки информации, расширение возможностей на современном уровне рабочего процесса.
3. Повышение административной эффективности медицинского центра за счет автоматизации процессов записи на прием и обработки медицинской документации.
4. Снижение потерь времени и денежных средств на создание и оформление бумажной документации.
5. Обеспечение возможности просмотра медицинских результатов и рецептов онлайн.
6. Разработка каталога специалистов и услуг медицинского центра.
7. Создание возможности регистрации сотрудников медицинского центра в приложении.
8. Создание возможности изменения данных о сотрудниках медицинского центра.
9. Создание возможности изменения данных о клиентах медицинского центра.
10. Добавление информации о расписании работы сотрудников.

**Функциональные возможности разрабатываемой системы:**

1. Авторизация персонала медицинского центра в приложении и разграничение их функциональных возможностей по ролям.
2. Хранение, добавление, изменение, удаление и отображение информации о сотрудниках.
3. Хранение, добавление, изменение, удаление и отображение информации о клиентах.
4. Хранение, добавление, изменение, удаление и отображении информации об услугах.
5. Хранение, добавление, изменение, удаление и отображении информации об оказанных услугах.
6. Хранение и отображение информации об оказанных услугах.
7. Хранение, добавление, изменение, удаление и отображении информации о приемах.
8. Хранение, добавление, изменение, удаление и отображении информации о медицинском центре.
9. Запись пациентов на прием.
10. Проведение профессиональных осмотров пациентов (профосмотров).
11. Отправка отчетной таблицы на печать.
12. Функция вывода данных в таблицу формата Excel.
13. Функция поиска в базах данных по заданным значениям.
14. Функция сортировки отображаемых данных из баз данных.

Разрабатываемая информационная система состоит из двух частей: база данных и приложение.

**База данных будет хранить следующую информацию:**

* Информация о сотрудниках (код сотрудника, ФИО, день рождения, должность, специализация, квалификация, расписание, электронная почта, номер телефона, адрес, логин, пароль, код центра, код статуса, кол пола);
* Информация о статусе сотрудника (код статуса, статус);
* Информация о поле (код пола, пол);
* Информация о клиентах (код клиента, ФИО, дата оформления, электронная почта, номер телефона, расчетный счет, название банка, код сотрудника, код центра, код пола);
* Информация об услугах (код услуги, наименование, стоимость, категория услуг, статус, срок выполнения, расписание, код центра);
* Информация об оказанных услугах (код оказанной услуги, наименование, стоимость, дата оказания, код приема);
* Информация о приемах (код приема, наименование, стоимость, время приема, дата оказания, симптомы, диагноз, примечание, код сотрудника, код клиента);
* Информация о медицинском центре (код центра, наименование, ИНН, адрес, электронная почта, телефон).

**Разрабатываемая информационная система будет иметь следующий функционал, разделенный по ролям:**

1. **Роль: директор.**
2. Просмотр списка услуг (с возможностью поиска и сортировки).
3. Добавление новых услуг.
4. Изменение данных уже имеющейся услуги.
5. Удаление данных об услугах.
6. Вывод базы данных «Услуги» в формате Excel.
7. Просмотр списка клиентов (с возможностью поиска и сортировки).
8. Регистрация новых клиентов.
9. Редактирование данных клиентов.
10. Удаление данных клиентов.
11. Вывод базы данных «Клиент» в формате Excel.
12. Просмотр списка сотрудников (с возможностью поиска и сортировки).
13. Регистрация новых сотрудников.
14. Редактирование данных сотрудников.
15. Удаление данных сотрудников.
16. Изменение статуса сотрудников.
17. Вывод базы данных «Сотрудники» в формате Excel.
18. Просмотр информации о поле клиентов и сотрудников.
19. Просмотр отчетной таблицы с уже оказанными услугами (с возможностью поиска и сортировки).
20. Расчет общей стоимости оказанных услуг.
21. Расчет общего количества оказанных услуг.
22. Вывод базы данных «Оказанные услуги» в формате Excel.
23. Отправка отчетной таблицы на печать.
24. Просмотр списка приемов (с возможностью поиска и сортировки).
25. Вывод базы данных «Приемы» в формате Excel.
26. Просмотр информации о медицинском центре (с возможностью поиска и сортировки).
27. Добавление новых данных о медицинском центре.
28. Изменение и удаление данных о медицинском центре.
29. Вывод базы данных «Медицинский центр» в формате Excel.
30. **Роль: администратор.**
31. Просмотр списка услуг (с возможностью поиска и сортировки).
32. Вывод базы данных «Услуги» в формате Excel.
33. Просмотр списка приемов (с возможностью поиска и сортировки).
34. Запись клиентов на прием.
35. Вывод базы данных «Приемы» в формате Excel.
36. Просмотр списка клиентов (с возможностью поиска и сортировки).
37. Регистрация новых клиентов.
38. Редактирование данных клиентов.
39. Вывод базы данных «Клиенты» в формате Excel.
40. **Роль: врач.**
41. Просмотр списка услуг (с возможностью поиска и сортировки).
42. Вывод базы данных «Услуги» в формате Excel.
43. Просмотр списка приемов (с возможностью поиска и сортировки).
44. Добавление нового приема.
45. Изменение данных о приемах.
46. Удаление данных о приемах.
47. Вывод базы данных «Приемы» в формате Excel.
48. Просмотр списка клиентов (с возможностью поиска и сортировки).
49. Вывод базы данных «Клиенты» в формате Excel.
50. Просмотр отчетной таблицы с уже оказанными услугами (с возможностью поиска и сортировки).
51. Добавление новых данных об оказанных услугах.
52. Изменение и удаление данных об оказанных услугах.
53. Вывод базы данных «Оказанные услуги» в формате Excel.
54. Отправка отчетной таблицы на печать.

**Описание бизнес-процессов**

Ниже представлена блок-схема бизнес-процессов приложения (рисунок 1):

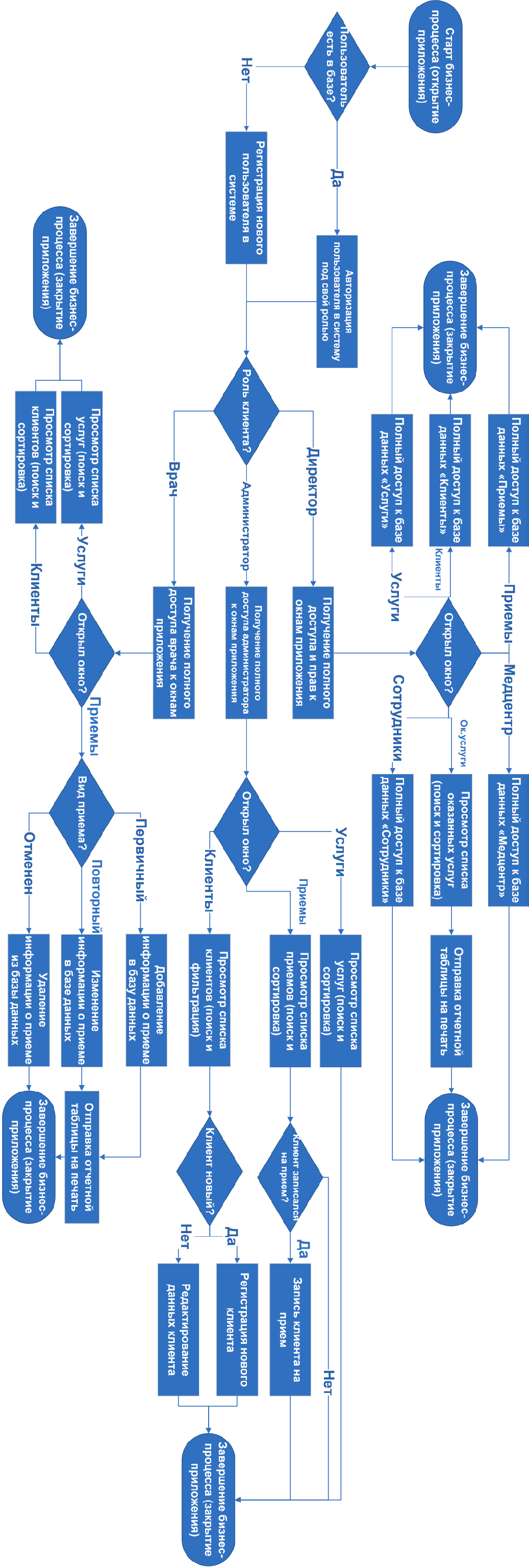


Рисунок 1 - Блок-схема бизнес-процессов

**Глава 2. Анализ аналогов и прототипов**

1. **Медицинская система «Archimed+»** (<https://archimed.pro/>):

Данная информационная система предназначена для работников медицинских центров — медрегистраторов, врачей, бухгалтеров, лаборантов и других. Для каждого из них в системе реализован свой набор функций. Программа позволяет быстро найти карту пациента и любую информацию в ней, позволяет вести базу пациентов, вводить и хранить персональные данные, в систему можно загрузить любую форму для печати, в которую можно подставлять данные пациентов. Также система позволяет автоматизировать график работы врачей и вести журнал записей пациентов на прием.

Ниже представлен интерфейс рассматриваемого аналога (рисунок 2):

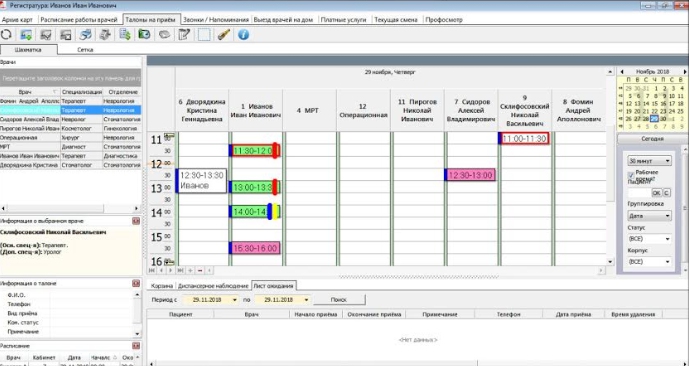


Рисунок 2 - Медицинская система «Archimed+»

**Плюсы:**

* Простота использования;
* Удобный интерфейс;
* Взаимодействие с другими полезными сервисами, такими как: лаборатории, запись в режиме онлайн, система МДЛП и остальные;
* Выгрузка данных в ЕГИСЗ;
* Доступна мобильная версия для медицинских работников;
* Модуль CRM-системы;
* Возможность телемедицины для коммуникации с пациентами;
* Круглосуточная техническая поддержка
* Для обработки звонков предусмотрен робот-оператор;
* Работа с онлайн-кассами и IP-телефонией;
* Экспорт прайс-листа на веб-сайт;
* Доступен персональный кабинет для посетителей;
* Поддерживает интеграцию с маркетинговой платформой RoiStat;
* Система лояльности;
* Комплект базовых и расширенных модулей, в том числе: «Запись на прием», «Медицинские консультации», «Финансовый учет», «Стационарное лечение», «Отчетность» и дополнительные.

**Минусы:**

* Большое количество различных кнопок и функций в одном окне;
* Высокая стоимость;
* Низкая скорость работы.

1. **Медицинская система «Sycret Med»** (<https://sycret.ru/>):

Данная информационная система предназначена для управления работой с пациентами частных клиник и отслеживания истории коммуникации с ними: от канала привлечения до полученной выручки. Подготавливает различные виды отчетов из полученной информации. Программа имеет базу данных пациентов и журналы записи на прием, электронные медицинские карты и документы, удаленный мониторинг работы, учет лекарственных препаратов. Позволяет изменять записи в медкартах — переносить или отменять прием, записывать другого больного на процедуру и многое другое.

Ведет самостоятельный контроль назначенных процедур и формирует план лечения, а также учитывает акционные предложения и скидки, исключая при этом возможные ошибки.

Ниже представлен интерфейс рассматриваемого аналога (рисунок 3):



Рисунок 3 - Медицинская система «Sycret Med»

**Плюсы:**

* Ведение журналов регистраций и ведение базы пациентов;
* Инструменты CRM-системы для автоматизации рабочих процессов;
* Поддержка обмена сведениями с «ЕГИСЗ» и «Честный знак»;
* Индивидуальный ассистент в начальной стадии и постоянная техническая помощь;
* Интеграция с онлайн-кассой и IP-телефонией;
* Мобильное приложение для сотрудников и пациентов;
* Расширенная система лояльности — бонусы, дисконтные карты и гибкая система скидок;
* Личный кабинет руководителя с возможность удаленного доступа с любого устройства;
* Контроль работы склада и учет лекарственных препаратов;
* Работа с электронными медицинскими картами по форме «025У»;
* Создание плана лечения, готовые шаблоны и поиск диагнозов по «МКБ10»;
* Интеграция с сайтов агрегатором для онлайн записи «ПроДокторов»;
* Аналитика и различные виды отчетности;
* Уведомления по электронной почте, SMS, а также «WhatsApp» напоминания рассылка;
* Поддерживает работу в «облаке».

**Минусы:**

* Требуется проходить обучение для работы с программой;
* Сложный интерфейс.

1. **Медицинская система «YCLIENTS»** (<https://www.yclients.com/>):

Данная информационная система предназначена для автоматизации рабочих процессов организаций, таких как создание онлайн-записей, гибкая сегментация базы клиентов, ведение электронного журнала, ведение аналитики, складской учет, финансовый учет, отправка уведомлений, автоматический расчет зарплат сотрудников и выплата по заданным правилам, программы лояльности, поддержка кассовых сервисов, работа с платежами в едином интерфейсе, наличие мобильного приложения.

Ниже представлен интерфейс рассматриваемого аналога (рисунок 4):

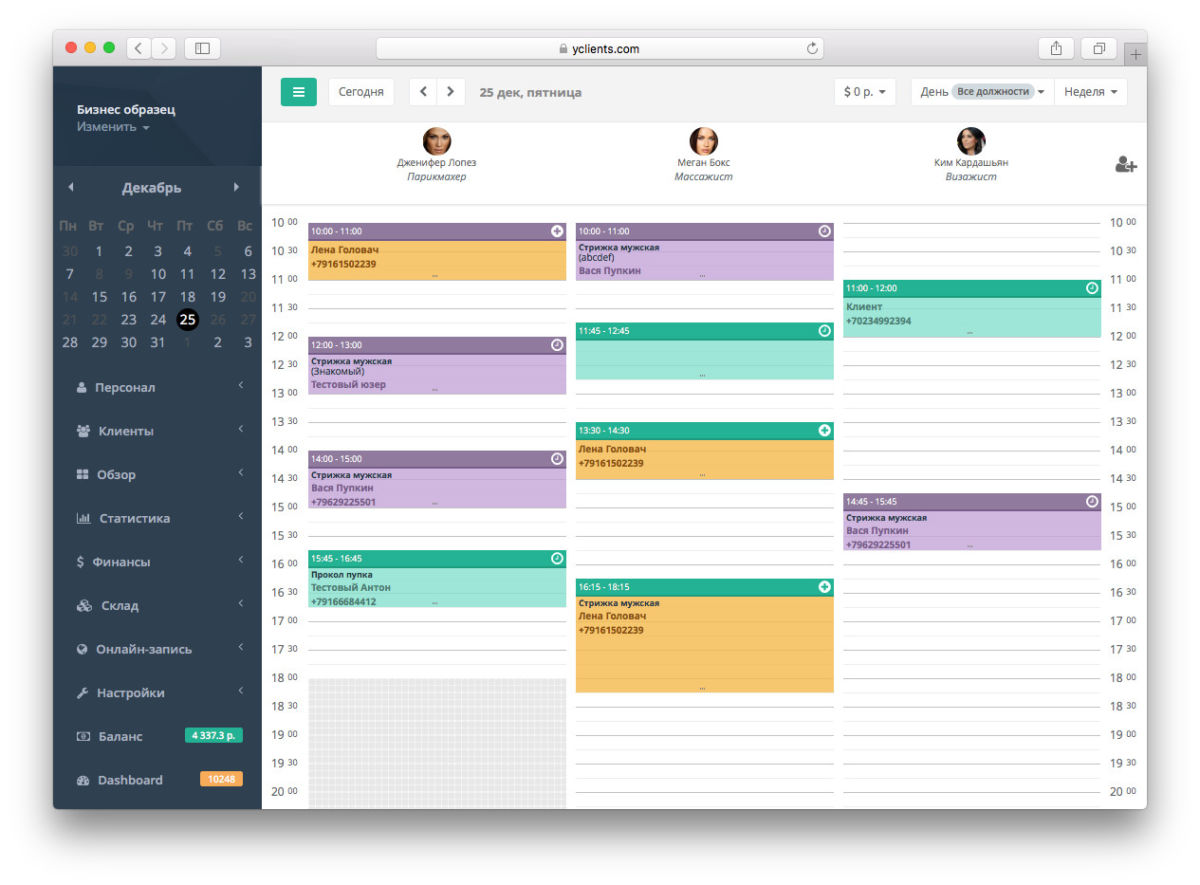


Рисунок 4 - Медицинская система «YCLIENTS»

**Плюсы:**

* Наличие виджета для онлайн-записи;
* Встроенная CRM-система;
* Модули финансового и складского учета;
* Программа лояльности для клиентов;
* SMS-рассылка и уведомление о записи;
* Круглосуточная служба поддержки;
* Выгодные тарифные планы;
* Удобный и понятный функционал сервиса;
* Интеграция с кассовым оборудованием.

**Минусы:**

* Высокая стоимость;
* Нет возможности проработки «холодных» клиентов.

**Выводы:**

Медицинский центр имеет много сильных конкурентов, предоставляющих широкий спектр медицинских услуги и высокое качество обслуживания клиентов. Они обладают высокой популярностью и имеют хорошую репутацию на рынке.

Конкуренты медицинского центра активно используют различные информационные системы для управления клиентской базой, записи на приемы, учета услуг и финансовых операций.

Большинство медицинских центров имеют недостаточную интеграцию различных информационных систем, в результате этого информация о пациентах может быть разбросана по разным базам данных, что затрудняет доступ к ней и может привести к неполным или ошибочным данным.

Медицинским центрам часто не хватает современных систем аналитики данных, которые помогли бы им обрабатывать большие объемы информации и делать точные прогнозы.

Современные медицинские информационные системы чаще всего имеют ограниченные возможности обмена данными, что может затруднять обмен данными между различными учреждениями здравоохранения.

Таким образом, имеющиеся информационные системы медицинских центров часто не удовлетворяют современные требования к эффективной и безопасной работе медицинских учреждений, вследствие чего требуется разработка информационной системы медицинского центра, с использованием всех современных технологий и методов разработки программного обеспечения для оказания медицинских услуг на 2024 год.

# **Глава 3. Проектирование и программирование**

## **Раздел 3.1. Описание интерфейса и архитектуры разрабатываемого продукта**

Основной шрифт в разрабатываемом приложении – Segoe UI. Исходя из цветов, входящих в состав логотипа предприятия, для которого производится разработка, были выбраны цвета:

* Красный – для акцентирования внимания на функциональных элементах, таких как кнопки;
* Белый – фон приложения;
* Черный – цвет шрифта приложения.

Ниже приведены макеты основной и типовой форм приложения:

1. Макет основной формы приложения.

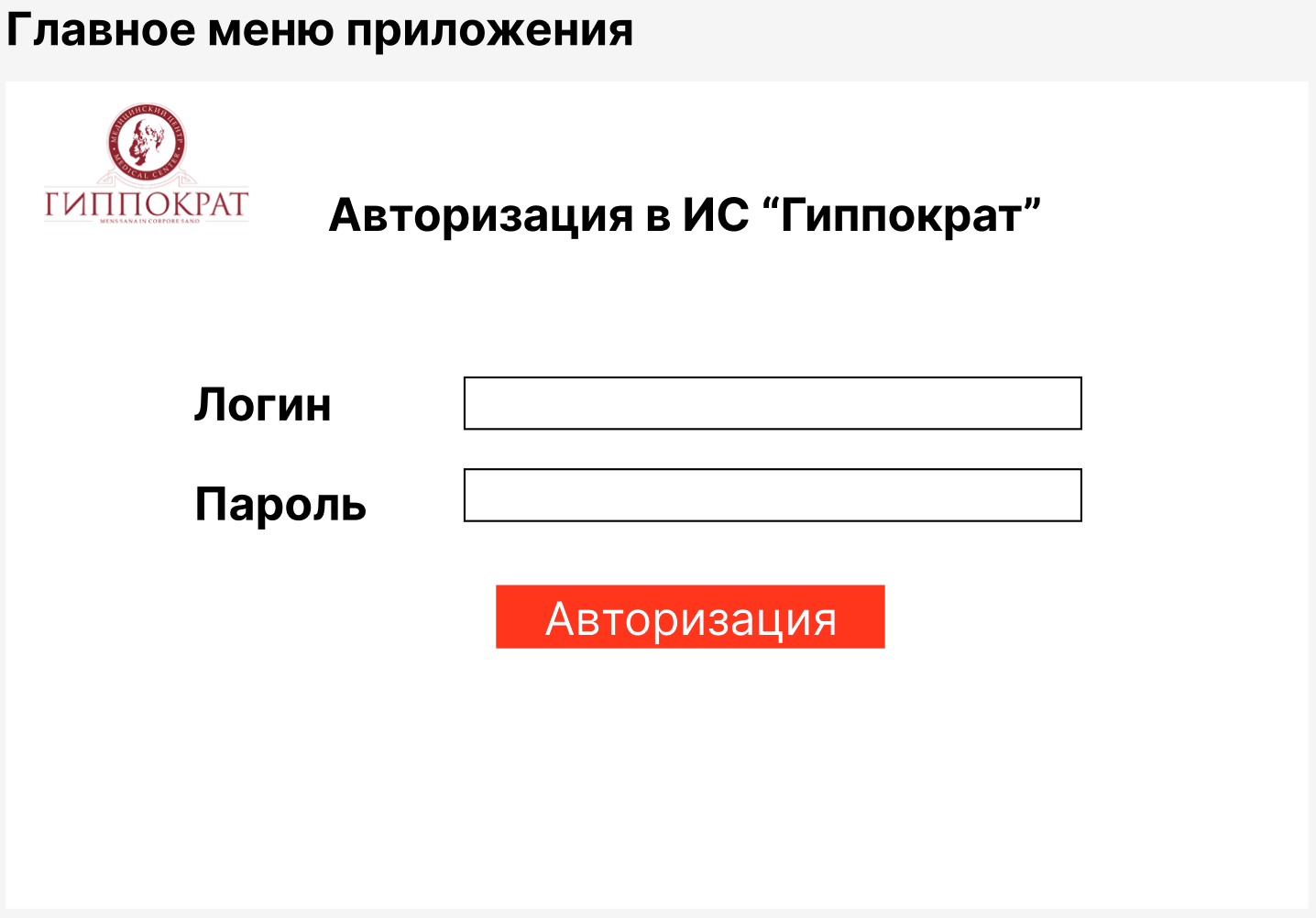


Рисунок 5 - Макет основной формы приложения

1. Первый макет типовой формы приложения.

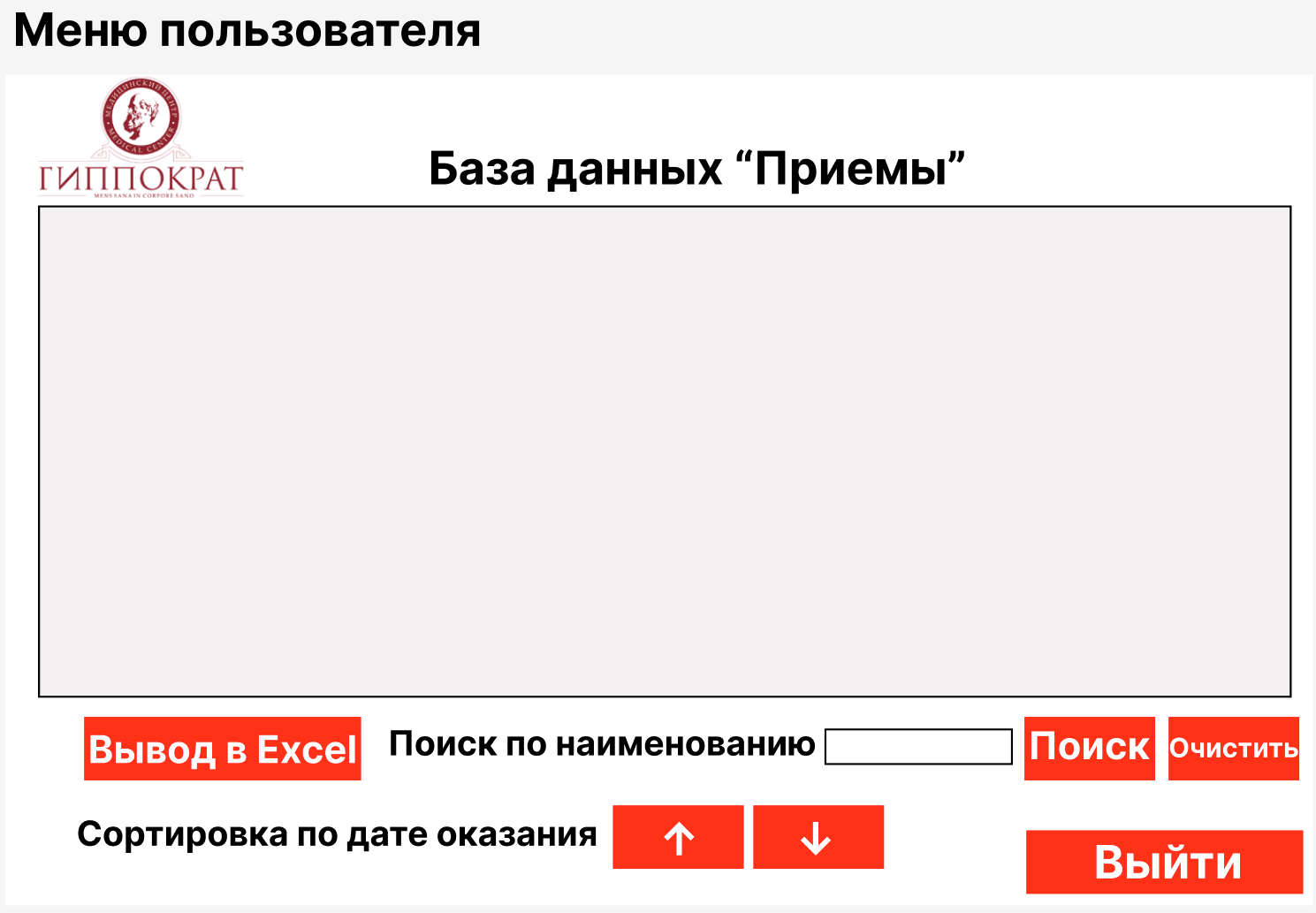


Рисунок 6 - Первый макет типовой формы приложения

1. Второй макет типовой формы приложения.

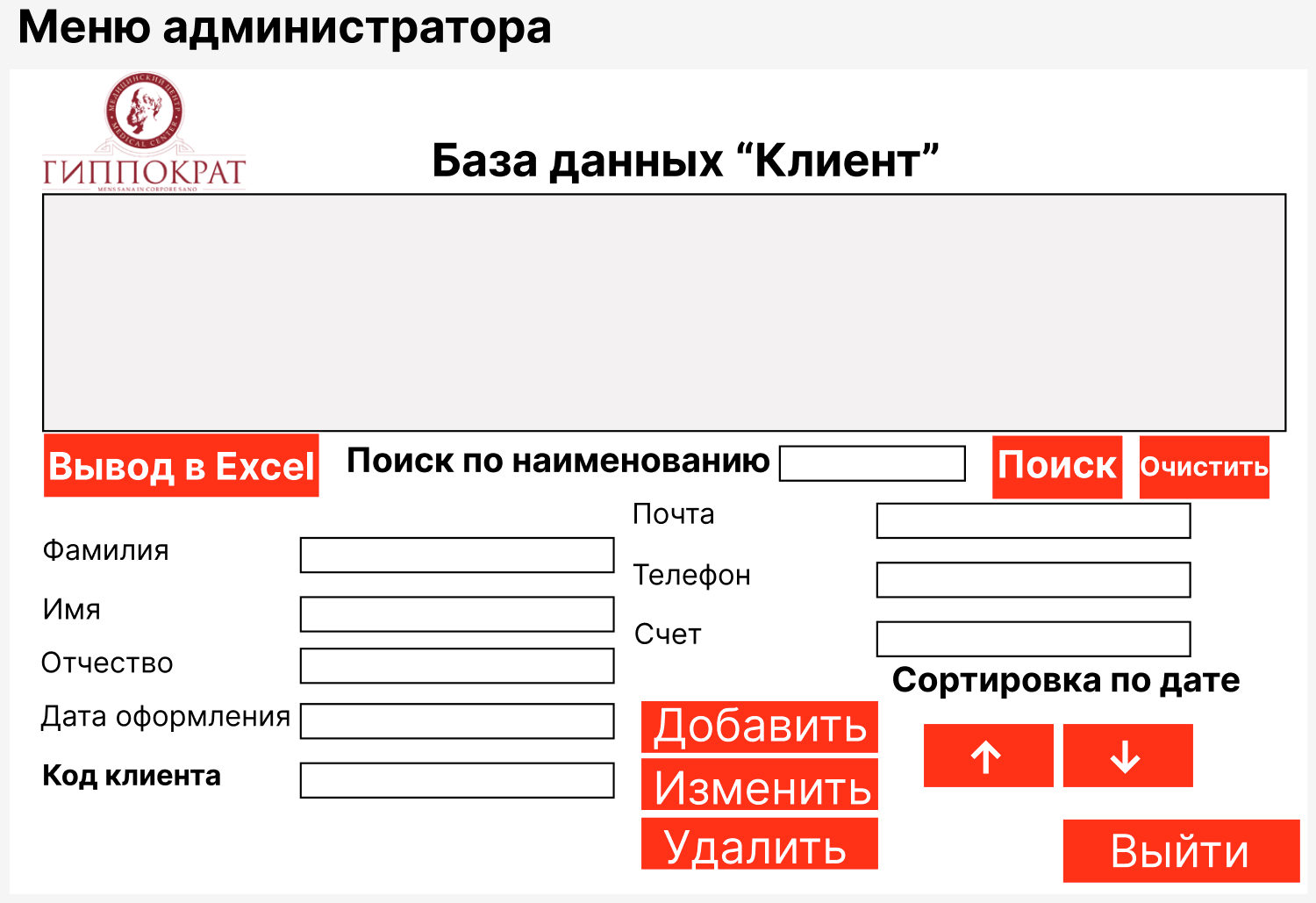


Рисунок 7 - Второй макет типовой формы приложения

**Требования к внешнему оформлению разрабатываемой системы:**

1. Все формы должны быть выполнены в едином цветовом оформлении и использовании одного шрифта.
2. Хорошая читаемость шрифтов и размеров текста, адаптированных под компьютерные мониторы и различное разрешение экранов.
3. Наличие интуитивной навигации и логической структуры приложения для быстрого доступа к необходимой информации и функции.
4. Использование качественных изображений и иконок для визуального улучшения интерфейса приложения.
5. Логотип медицинского центра должен присутствовать на всех формах информационной системы.
6. Функциональные элементы приложения должны иметь одинаковый размер.
7. Цветовое оформление форм приложения должно соответствовать цветам логотипа медицинского центра.
8. Русские и английские буквы, а также цифры должны иметь схожее отображение на формах.
9. Регулярное обновление дизайна приложения с учетом новых трендов и потребностей пользователей для поддержания актуальности и конкурентоспособности.

**Основные характеристики программного продукта:**

1. Корректность и целостность отображаемой информации.
2. Наличие пользовательского интерфейса, который позволяет взаимодействовать с программой.
3. Функциональность, которая определяет, какие задачи может выполнять программа.
4. Надежность и стабильность работы программы в заданных условиях.
5. Производительность, которая определяет скорость выполнения задач.
6. Совместимость с другими программами и популярными операционными системами.
7. Обеспечение безопасности и защиты информации от несанкционированного доступа.
8. Единый способ перехода между формами.
9. Поддержка и обновление программы.
10. Разграничение прав доступа.
11. Стоимость и условия лицензирования программного продукта.

**В качестве входной информации будет выступать:**

* Данные о пациентах;
* Данные о посещениях пациентов и их болезнях;
* Данные о предоставляемых услугах;
* Данные о сотрудниках;
* Данные о приемах;
* Данные о медицинском центре.

**Выходной информацией будет выступать:**

* Информация об оказанных услугах и их стоимости.
* Информация о проведенных приемах и их стоимости.

**Функциональная структура разрабатываемого приложения**

Ниже представлена функциональная схема приложения (рисунок 8):

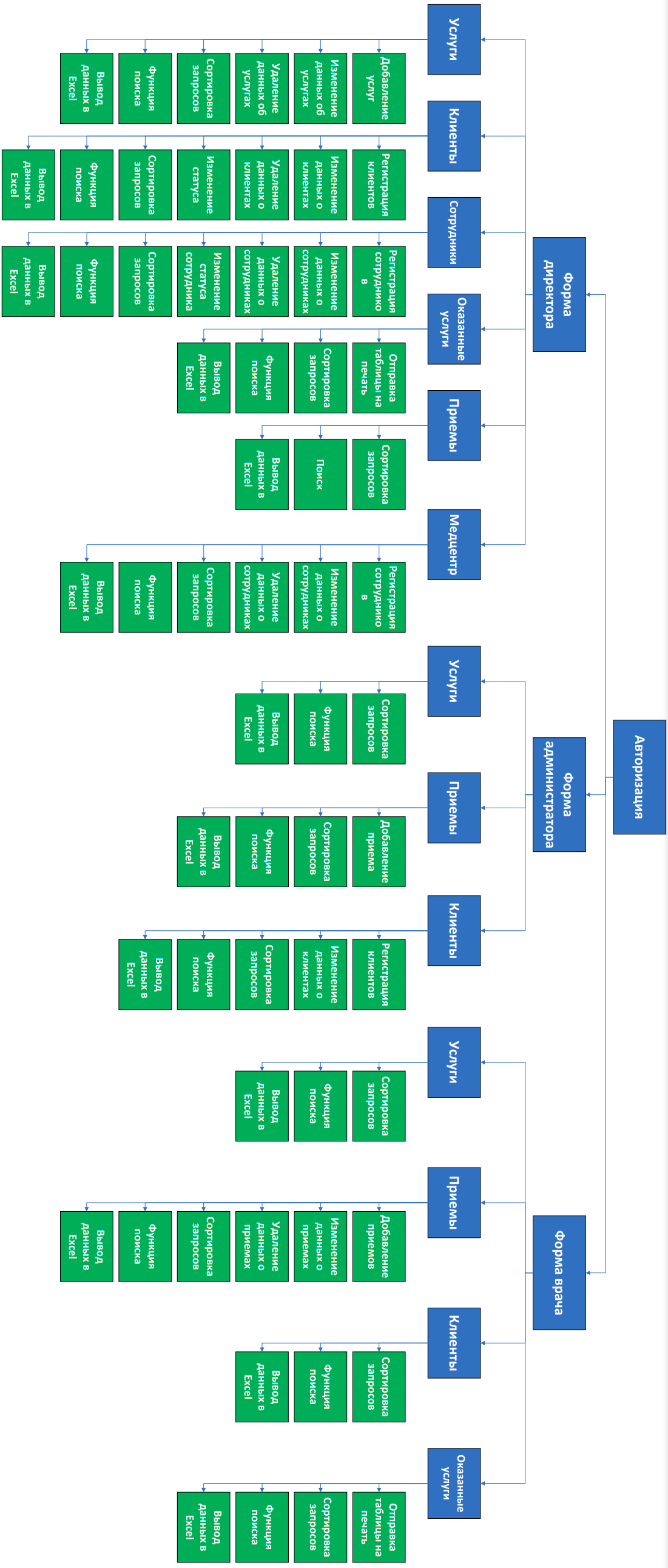


Рисунок 8 - Функциональная схема приложения

Ниже представлена логическая схема базы данных приложения (рисунок 9):

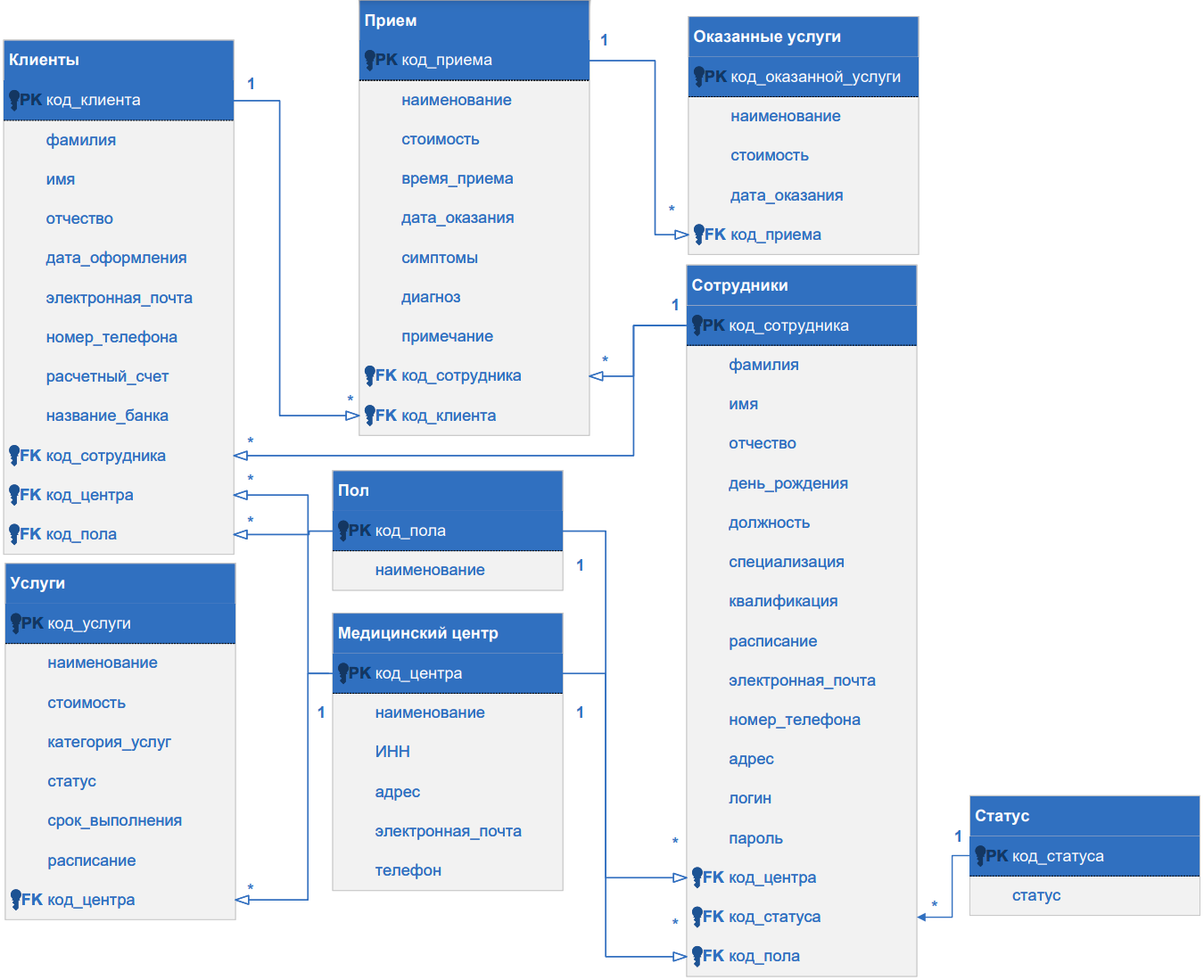


Рисунок 9 **-** Логическая схема базы данных

**Раздел 3.2. Разработка внутренней структуры и программирование**

Ниже будет приведена физическая структура разрабатываемого приложения медицинского центра.

* В папке **«Diplom\_Ohotnikov»** хранятся программные файлы приложения;
* В папке **«Images»** хранятся изображения, размещаемые на формах приложения, такие как логотип и так далее;
* В папке **«Library»** хранятся юнит-тесты, применяемые для тестирования работоспособности приложения;
* Файл **«Diplom\_Ohotnikov.sln»**, является исполняемым файлом запуска приложения.

Ниже представлена физическая структура приложения (рисунок 10):

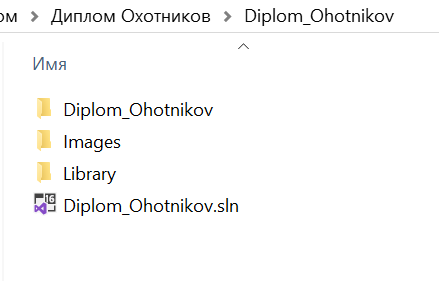


Рисунок 10 - Физическая структура приложения

В процессе разработки были использованы такие программные обеспечения как:

* Microsoft SQL Server Management Studio 2019 (версия 19.1);
* Microsoft Visual Studio Community 2019 (версия 16.11.25).

Для разработки на стороне сервера базы данных был использован язык запросов SQL, с помощью которого были реализованы такие процессы как:

* Вывод данных;
* Обновление данных;
* Удаление данных;
* Занесение данных.

Разработка на стороне клиента была осуществлена на языке C#, в формате приложений WPF (.NET Framework). С помощью данного языка, были реализованы такие аспекты как:

* Внешний вид приложения;
* Визуализация информации;
* Активация вышеперечисленных запросов и функций языка SQL;
* Переходы между формами приложения;
* Фильтрация и сортировка;
* Поиск;
* Прочие функции приложения.

**Детальный алгоритм обработки данных работает следующим образом:**

1. Пользователь вводит данные в текстовые блоки, расположенные на форме приложения, доступной данному пользователю.
2. После нажатия определенной кнопки, происходит автоматическая активация запроса SQL на добавление, обновление и удаление данных (активация происходит только при корректном заполнении данных и автоматической проверки).
3. При корректном вводе данных, выводится всплывающее окно, уведомляющее пользователя об успехе произведенной операции (при некорректном вводе всплывает окно с подсказкой о неправильном вводе или отсутствии текстовых данных в полях).

Ниже приведены примеры программной реализации функций приложения:

1. Пример реализации авторизации в информационной системе.

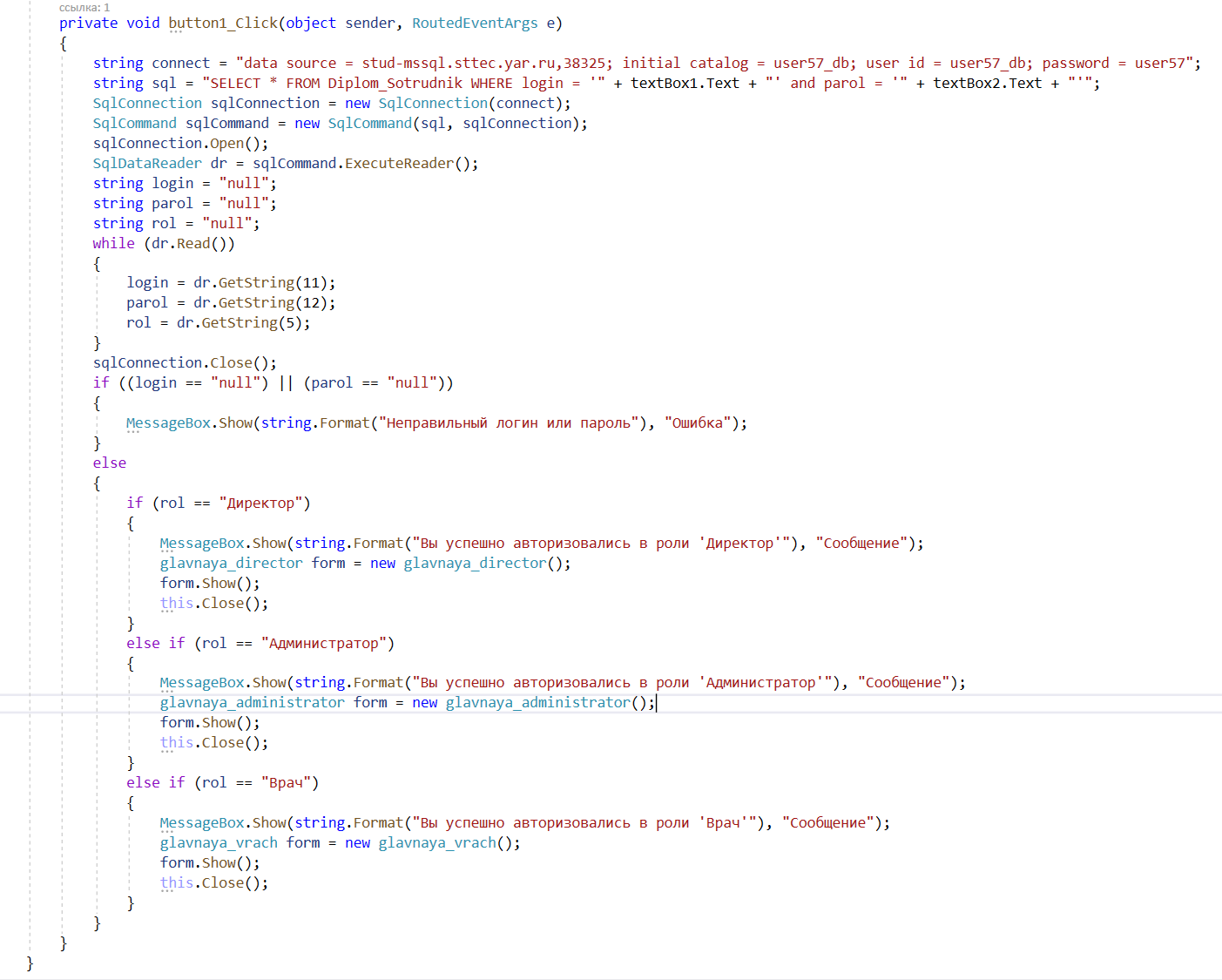


Рисунок 11 - Пример реализации авторизации в приложении

1. Пример реализации добавления данных в информационной системе.

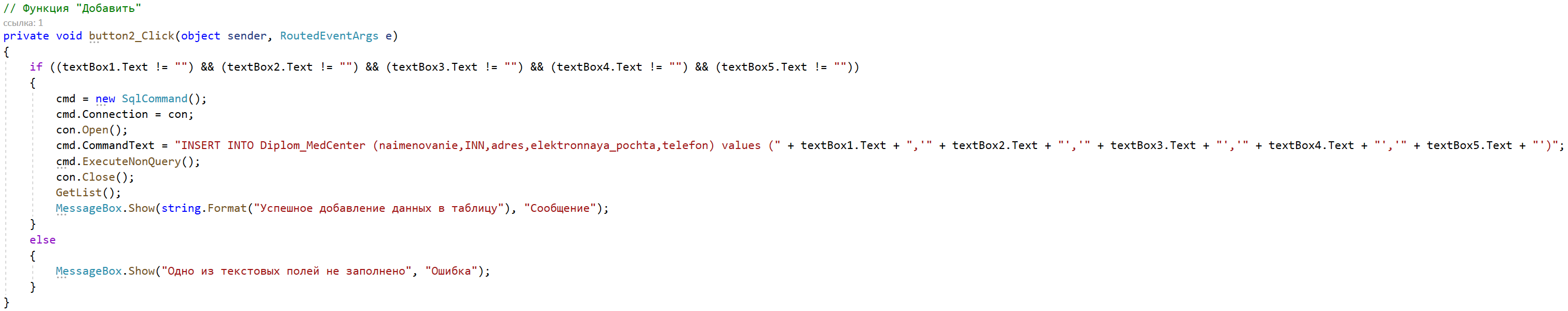


Рисунок 12 - Пример реализации добавления данных через приложение

1. Пример реализации изменения данных в информационной системе



Рисунок 13 - Пример реализации изменения данных через приложение

1. Пример реализации удаления данных в информационной системе.



Рисунок 14 - Пример реализации удаления данных через приложение

**Описание функционала формы «Сотрудники»**

Ниже представлен функционал формы «Сотрудники» (рисунок 15):



Рисунок 15 - Функционал формы «Сотрудники»

1. Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу «Сотрудники».
2. Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице «Сотрудники», в зависимости от значения в поле «Код сотрудника»
3. Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице «Сотрудники», в зависимости от значения в поле «Код сотрудника», также очищает текстовые поля формы.
4. Кнопка **«Поиск»** - производит автоматический поиск по введенному значению фамилии сотрудника и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
5. Кнопка **«Очистить»** - очищает текстовое поле «Поиск по фамилии» и обновляет форму.
6. Кнопка **«Вывод в Excel»** - сохраняет базу данных «Сотрудники» в формате файла Excel, что позволяет произвести дальнейшею печать.
7. Кнопка «**Перезагрузка»** - перезагружает форму.
8. Кнопка **«БД ‘Статус’»** - открывает форму базы данных «Статус».
9. Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дню рождения в порядке возрастания.
10. Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дню рождения в порядке убывания.
11. Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

**Схема базы данных «Медицинский центр»**

Ниже представлена схема базы данных информационной системы (рисунок 16):

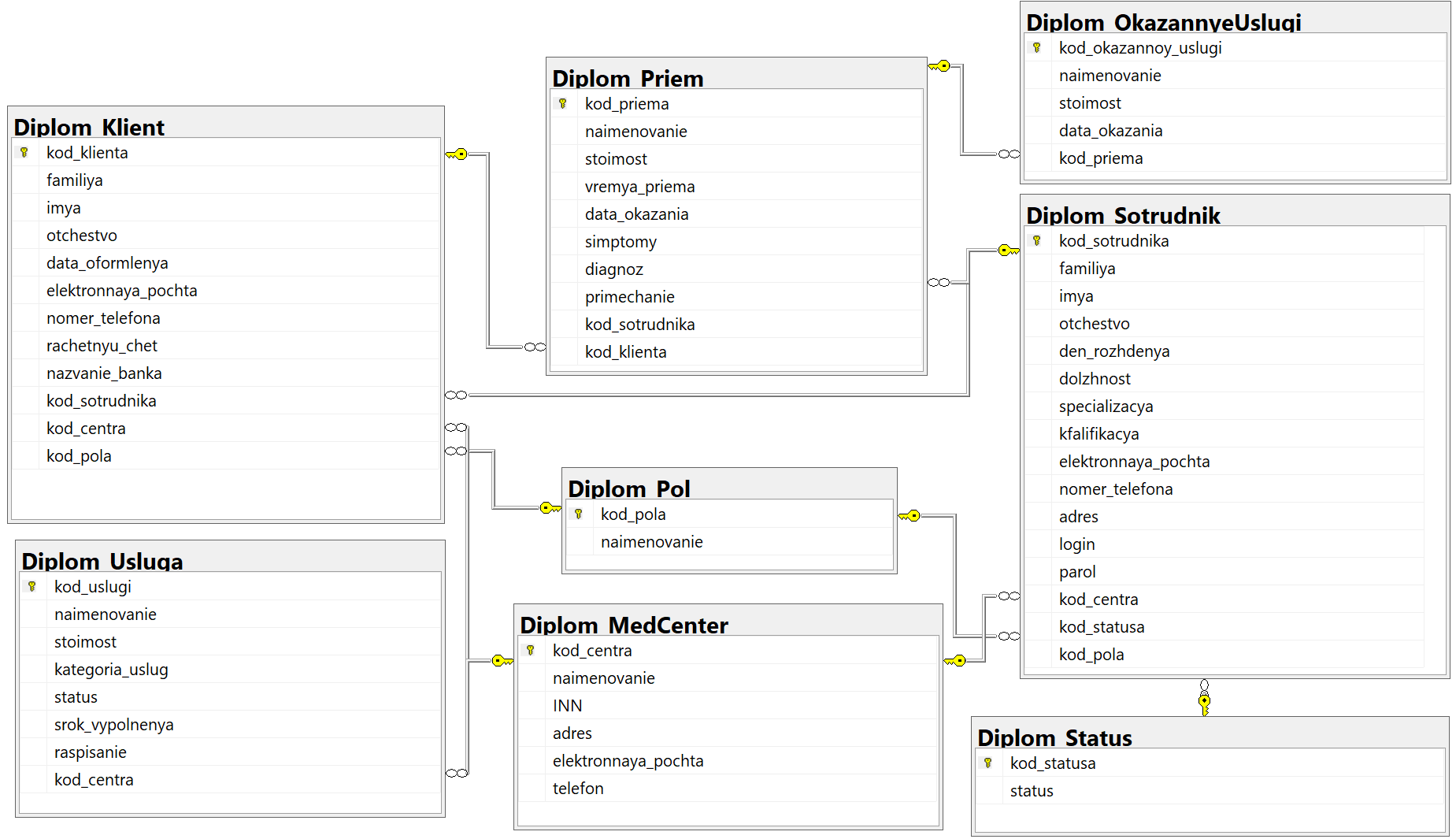


Рисунок 16 - Схема базы данных

**Детальное описание таблиц базы данных и хранимой в них информации:**

1. **Клиенты (Diplom\_Klient).**
2. kod\_klienta (PK, int, NOT NULL): уникальный идентификатор клиента.
3. familiya (varchar(50), NOT NULL): фамилия клиента.
4. imya (varchar(50), NOT NULL): имя клиента.
5. otchestvo (varchar(50), NOT NULL): отчество клиента.
6. data\_oformlenya (varchar(50), NOT NULL): дата оформления клиента.
7. elektronnaya\_pochta (varchar(50), NOT NULL): электронная почта клиента.
8. nomer\_telefona (varchar(50), NOT NULL): номер телефона клиента.
9. rachetnyu\_chet (varchar(50), NOT NULL): расчетный счет клиента.
10. nazvanie\_banka (varchar(50), NOT NULL): название банка клиента.
11. kod\_sotrudnika (FK, int, NOT NULL): уникальный идентификатор сотрудника.
12. kod\_centra (FK, int, NOT NULL): уникальный идентификатор медицинского центра.
13. kod\_pola (FK, int, NOT NULL): уникальный идентификатор пола.
14. **Медицинский центр (Diplom\_MedCenter).**
15. kod\_centra (PK, int, NOT NULL): уникальный идентификатор медицинского центра.
16. naimenovanie (varchar(50), NOT NULL): наименование организации.
17. INN (varchar(50), NOT NULL): идентификационный номер налогоплательщика.
18. adres (varchar(50), NOT NULL): физический адрес организации.
19. elektronnaya\_pochta (varchar(50), NOT NULL): электронная почта организации.
20. telefon (varchar(50), NOT NULL): номер телефона организации.
21. **Оказанные услуги (Diplom\_OkazannyeUslugi).**
22. kod\_okazannoy\_uslugi (PK, int, NOT NULL): уникальный идентификатор оказанной услуги.
23. naimenovanie (varchar(50), NOT NULL): наименование оказанной услуги.
24. stoimost (varchar(50), NOT NULL): стоимость оказанной услуги.
25. data\_okazania (varchar(50), NOT NULL): дата оказания услуги.
26. kod\_priema (FK, int, NOT NULL): уникальный идентификатор приема.
27. **Прием (Diplom\_Priem).**
28. kod\_priema (PK, int, NOT NULL): уникальный идентификатор приема.
29. naimenovanie (varchar(50), NOT NULL): наименование приема.
30. stoimost (varchar(50), NOT NULL): стоимость оказанной услуги.
31. vremya\_priema (varchar(50), NOT NULL): время приема.
32. data\_okazania (varchar(50), NOT NULL): дата оказания приема.
33. simptomy (varchar(50), NOT NULL): симптомы пациента.
34. diagnoz (varchar(50), NOT NULL): диагноз пациента.
35. primechanie (varchar(50), NOT NULL): примечание.
36. kod\_sotrudnika (FK, int, NOT NULL): уникальный идентификатор сотрудника.
37. kod\_klienta (FK, int, NOT NULL): уникальный идентификатор клиента.
38. **Сотрудник (Diplom\_Sotrudnik).**
39. kod\_sotrudnika (PK, int, NOT NULL): уникальный идентификатор сотрудника.
40. familiya (varchar(50), NOT NULL): фамилия сотрудника.
41. imya (varchar(50), NOT NULL): имя сотрудника.
42. otchestvo (varchar(50), NOT NULL): отчество сотрудника.
43. den\_rozhdenya (varchar(50), NOT NULL): день рождения сотрудника.
44. dolzhnost (varchar(50), NOT NULL): должность сотрудника.
45. specializacya (varchar(50), NOT NULL): специализация.
46. kfalifikacya (varchar(50), NOT NULL): квалификация.
47. elektronnaya\_pochta (varchar(50), NOT NULL): электронная почта сотрудника.
48. nomer\_telefona (varchar(50), NOT NULL): номер телефона сотрудника.
49. adres (varchar(50), NOT NULL): адрес сотрудника.
50. login (varchar(50), NOT NULL): логин сотрудника.
51. parol (varchar(50), NOT NULL): пароль сотрудника.
52. kod\_centra (FK, int, NOT NULL): уникальный идентификатор медицинского центра.
53. kod\_statusa (FK, int, NOT NULL): уникальный идентификатор статуса сотрудника.
54. kod\_pola (FK, int, NOT NULL): уникальный идентификатор пола.
55. **Статус (Diplom\_Status).**
56. kod\_statusa (PK, int, NOT NULL): уникальный идентификатор статуса сотрудника.
57. status (varchar(50), NOT NULL): статус сотрудника.
58. **Пол (Diplom\_Pol).**
59. kod\_pola (PK, int, NOT NULL): уникальный идентификатор пола.
60. naimenovanie (varchar(50), NOT NULL): наименование.
61. **Услуга (Diplom\_Usluga).**
62. kod\_uslugi (PK, int, NOT NULL): уникальный идентификатор услуги.
63. naimenovanie (varchar(50), NOT NULL): наименование.
64. stoimost (varchar(50), NOT NULL): стоимость услуги.
65. kategoria\_uslug (varchar(50), NOT NULL): категория.
66. status (varchar(50), NOT NULL): статус услуги.
67. srok\_vypolnenya (varchar(50), NOT NULL): срок выполнения услуги.
68. raspisanie (varchar(50), NOT NULL): расписание услуги.
69. kod\_centra (FK, int, NOT NULL): уникальный идентификатор медицинского центра.

**Раздел 3.3. Обеспечение безопасности информации**

Основная угроза для конфиденциальности и целостности информации информационной системы – несанкционированный доступ. Несанкционированный доступ может происходить как извне (хакеры, киберпреступники), так и изнутри (недобросовестные сотрудники или посторонние лица, которые получили доступ к информации). В результате такого доступа конфиденциальная информация может быть украдена, изменена или уничтожена, что приведет к утечкам данных, нарушению приватности клиентов или сотрудников, или даже к финансовым потерям для компании. Для предотвращения угроз, необходимо создать ограничения доступа и разграничить доступный функционал для каждого сотрудника после пройденной авторизации и входа в приложение.

Защита информации от несанкционированного доступа реализована посредством проверки вводимого пользователем логина и пароля в текстовые поля на форме «авторизации». Во время авторизации проводится автоматическое считывание роли сотрудника и потом, в зависимости от введенных пользователем данных, происходит переход на определенные формы, которые соответствуют доступному пользователю функционалу и правам. На некоторых формах для определенных ролей различные функции заблокированы.

Существует другая ошибка, которая может привести к неполадкам работы приложения или нарушению свойств информации – это ошибки ввода. Для предотвращения возникновения данной угрозы, каждое текстовое поле в приложении имеет ограничение на ввод только определенных символов.

Также для обеспечения защиты информационной системы, проводятся регулярные аудиты безопасности и обновление систем защиты, которые помогают предотвратить несанкционированный доступ к конфиденциальной информации организации и иные различные проблемы безопасности.

**Раздел 3.4. Тестирование и проверка работоспособности**

Для проведения тестирования и проверки работоспособности информационной системы, был выбран метод юнит-тестирования, так как он позволяет быстро и удобно провести тестирование отдельных модулей приложения, без связки с другими частями.

Для проверки основной функции приложения – авторизации, был написан ряд unit-тестов, проверяющих корректность вводимого пользователем при входе пароля.

Разработанные тесты проверяют вводимую текстовую строку на следующие критерии:

* Пароль должен иметь длину не менее 8 символов;
* Пароль должен содержать в себе цифры;
* Пароль должен содержать в себе хотя бы одну заглавную букву.

**Примеры unit-тестов:**

1. Unit-тест на длину пароля не менее 8 символов.



Рисунок 17 - Код юнит-теста 1

1. Unit-тест на содержание цифр в пароле.



Рисунок 18 - Код юнит-теста 2

1. Unit-тест на содержание заглавной буквы в пароле.



Рисунок 19 - Код юнит-теста 3

Ниже представлен результат прохождения unit-тестов (рисунок 20):

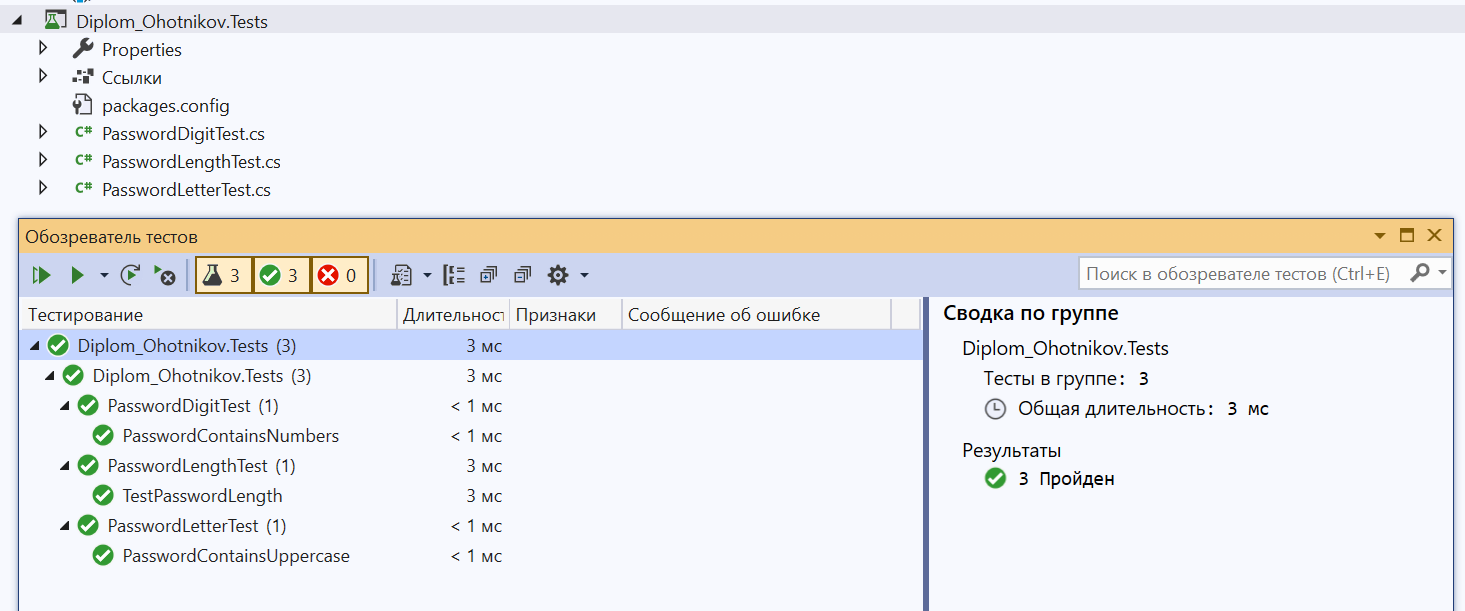


Рисунок 20 - Результат прохождения unit-тестов

**Глава 4. Аппаратные и программные средства для функционирования**

Для корректного функционирования информационной системы, персональный компьютер пользователей должен отвечать следующим минимальным требованиям:

* **Поддерживаемые операционные системы:**

1. Windows 11 версии 21H2 и выше.
2. Windows 10 версии 2004 и выше.
3. Windows Server 2019 (с обновлением 10.0.17763).
4. Windows Server 2016 (с обновлением 10.0.14393.6981).
5. Windows 8.1 (с обновлением 6.3.9600.17031).
6. Windows Server 2012 R2 версии 6.3.
7. Windows 7 с пакетом обновления 2 (SP2) и последними обновлениями Windows.

* **Оборудование:**

1. Процессор с тактовой частотой не ниже 2.0 ГГц. Рекомендуется использовать двухъядерные или четырехъядерные процессоры.
2. 4 ГБ оперативной памяти (ОЗУ). Рекомендуется не менее 8 ГБ ОЗУ (минимум 2,5 ГБ при выполнении приложения на виртуальной машине).
3. Место на жестком диске: до 300 ГБ (минимум 900 МБ) свободного места в зависимости от установленных системных компонентов, обычно для установки требуется от 30 до 70 ГБ свободного места.
4. Видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей), для стабильной и оптимальной работы программы рекомендуется разрешение 1080p (1920 на 1080 пикселей) или более высокое; минимальная видеопамять – 2 ГБ.

* **Поддерживаемые языки:**

1. Русский.
2. Английский.

Для реализации высокой надежности, стабильности и эффективности использования разрабатываемой информационной системы, необходимо проводить эксплуатацию при соблюдении всех норм использования соответствующего компьютерного оборудования.

Для реализации общей понятности приложения пользователем и обеспечения защиты персональных данных, необходимо выдавать своим работникам персональные логины и пароли для авторизации в системе, а также проводить начальный инструктаж.

**Требования надежности информационной системы:**

1. Высокая доступность: информационная система должна быть доступна постоянно, без существенных периодов недоступности.
2. Надежная защита данных: данные, хранящиеся в системе, должны быть защищены от утраты, повреждения и несанкционированного доступа.
3. Защита от вирусов и других вредоносных программ: система должна быть защищена от вирусов, хакеров и других угроз информационной безопасности.

**Требования эффективности информационной системы:**

1. Быстрая и надежная обработка данных: информационная система должна быть способна быстро обрабатывать большие объемы информации без задержек и ошибок.
2. Удобный и простой интерфейс: пользователи должны легко осваивать и использовать систему без необходимости дополнительного обучения.
3. Доступ к информации: информационная система должна обеспечивать доступ к нужной информации в любое время и в любом месте.

**Требования модифицируемости информационной системы:**

1. Гибкость и расширяемость: информационная система должна быть способна легко адаптироваться к изменениям в бизнес-процессах и требованиях пользователей. Новые функции и модули должны добавляться без значительных изменений в основной структуре системы.
2. Открытые стандарты: система должна быть построена с использованием открытых стандартов и технологий, что обеспечит возможность легкого взаимодействия с другими приложениями и системами.
3. Модульность: информационная система должна быть разделена на независимые модули, каждый из которых может быть изменен или заменен без влияния на другие части системы.

**Требования к защите данных информационной системы:**

1. Аутентификация пользователей: система должна обеспечивать проверку личности пользователей перед предоставлением им доступа к информации.
2. Авторизация доступа: установление прав доступа к информации в зависимости от роли пользователя в системе.
3. Физическая безопасность: обеспечение защиты серверов и сетевого оборудования от несанкционированного доступа.
4. Ведение журналов: система должна регистрировать все действия пользователей для обеспечения возможности выявления несанкционированных операций.
5. Программная защита: регулярное обновление программного обеспечения и установка антивирусных программ.

**Глава 5. Инструкция пользователя и администратора системы**

1. **Инструкция администратора (роль «Директор»).**
2. Для запуска приложения перейдите в корневую папку приложения и запустите файл запуска \*.sln (в данном случае «Diplom\_Ohotnikov.sln»).

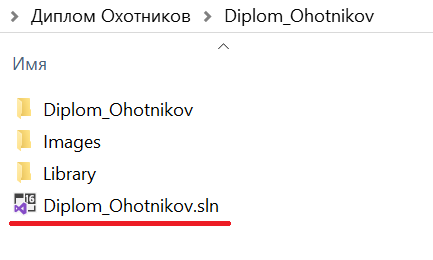


Рисунок 21 - Файл запуска

1. После запуска приложения Visual Studio, нажмите кнопку **«Пуск»** в верхней части экрана.

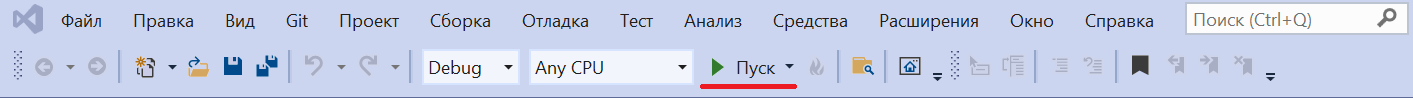


Рисунок 22 **-** Кнопка «Пуск»

1. Далее на форме **«Авторизация в информационную систему»** необходимо корректно ввести свой логин и пароль в соответствующие текстовые поля и нажать кнопку **«Авторизация»**. При успешной авторизации должно появится сообщение о входе в профиль **«Директор»**.

Ниже представлено окно авторизации в профиль «Директор» (рисунок 23):

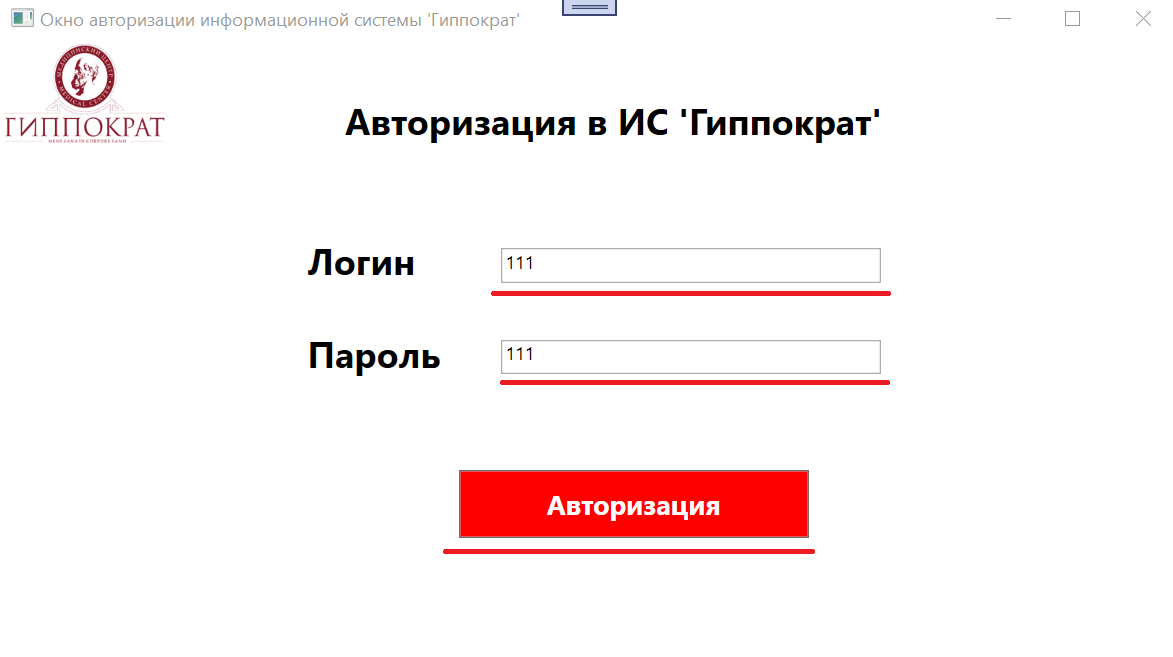


Рисунок 23 - Окно авторизации в профиль «Директор»



Рисунок 24 - Сообщение об успешной авторизации

1. При вводе неправильных данных в текстовые поля на форме авторизации, появится сообщение об ошибке входа в профиль **«Директор»**. Исправьте неправильные данные и нажмите **«Авторизация»**.

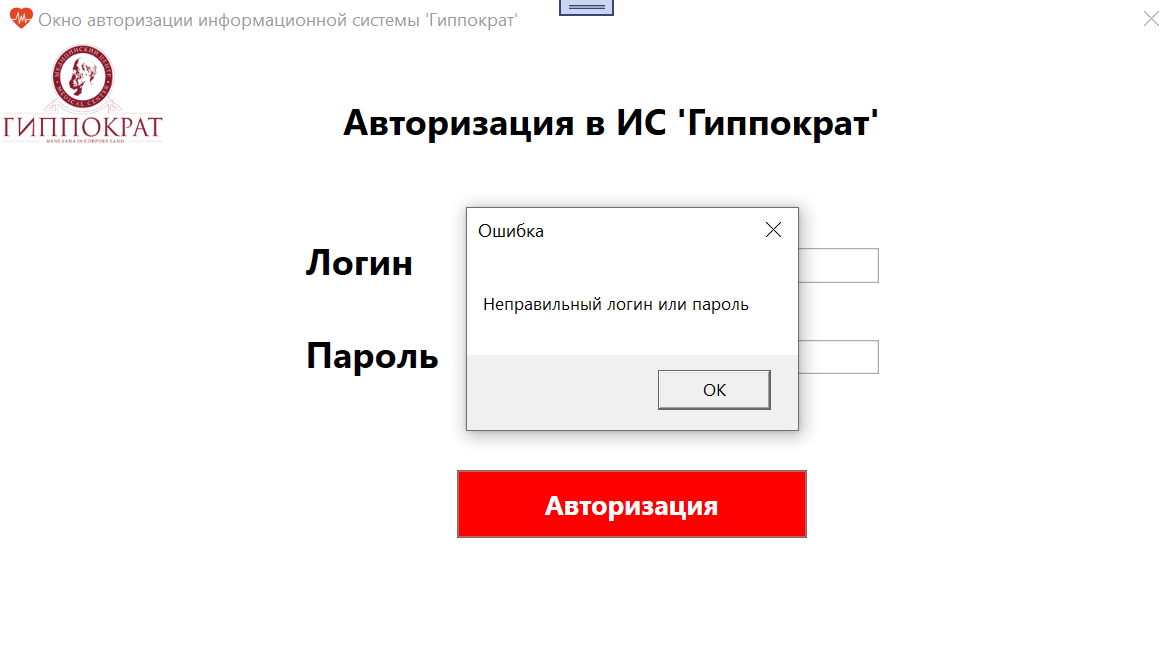


Рисунок 25 - Сообщение об ошибке процесса авторизации

1. После успешной авторизации Вы попадете в главное меню профиля **«Директор»**, где необходимо нажать определенную кнопку для перехода на соответствующую кнопку. Доступны формы баз данных **«Услуги»**, **«Клиенты»**, **«Сотрудники»**, **«Оказанные услуги»**, **«Приемы»** и **«Медцентр»**. Кнопка **«Выйти»** переведет пользователя обратно на форму авторизации и произведет выход из профиля.



Рисунок 26 - Профиль администратора «Директор»

1. При нажатии кнопки **«БД Услуги»**, осуществится переход на соответствующую форму **«Услуги»**.



Рисунок 27 - База данных «Услуги»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Услуги»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.



Рисунок 28 - Сообщение об ошибке добавления

Ниже представлено окно с успешным добавлением данных в таблицу (рисунок 29):

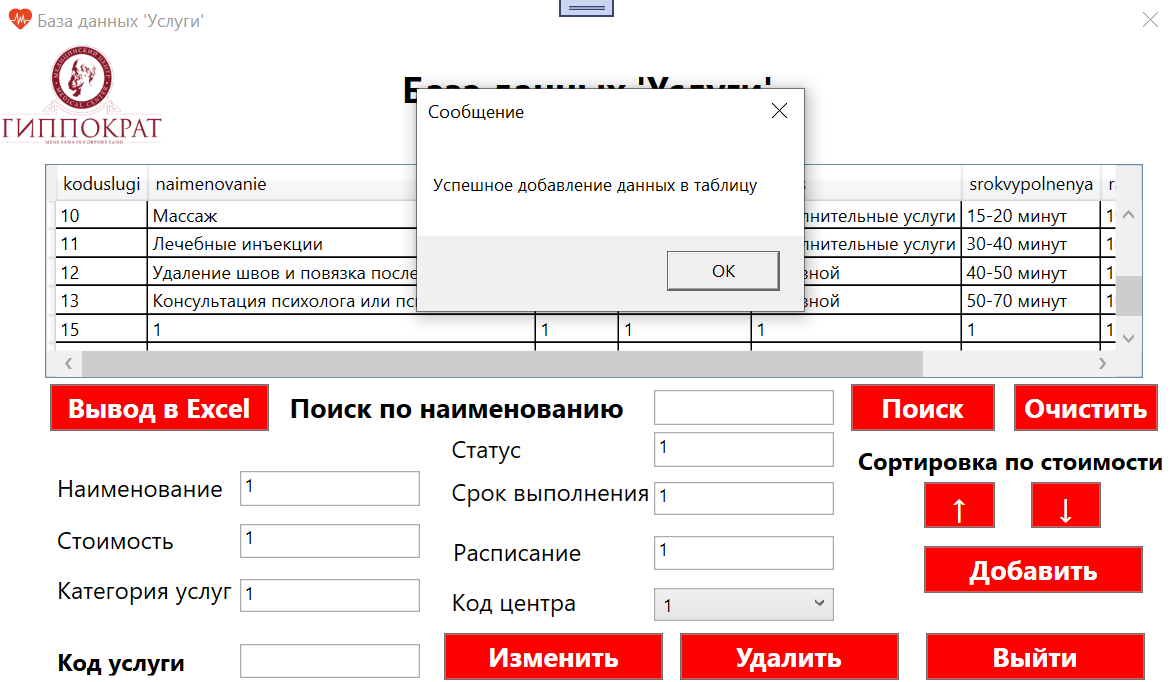


Рисунок 29 - Успешное добавление данных в таблицу

* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Услуги»**, в зависимости от значения в поле «**Код услуги»**. Если оставить поле **«Код услуги»** пустым, то выйдет сообщение о том, что поле не было заполнено.

Ниже представлена работа функции «Изменить» (рисунок 30, 31):

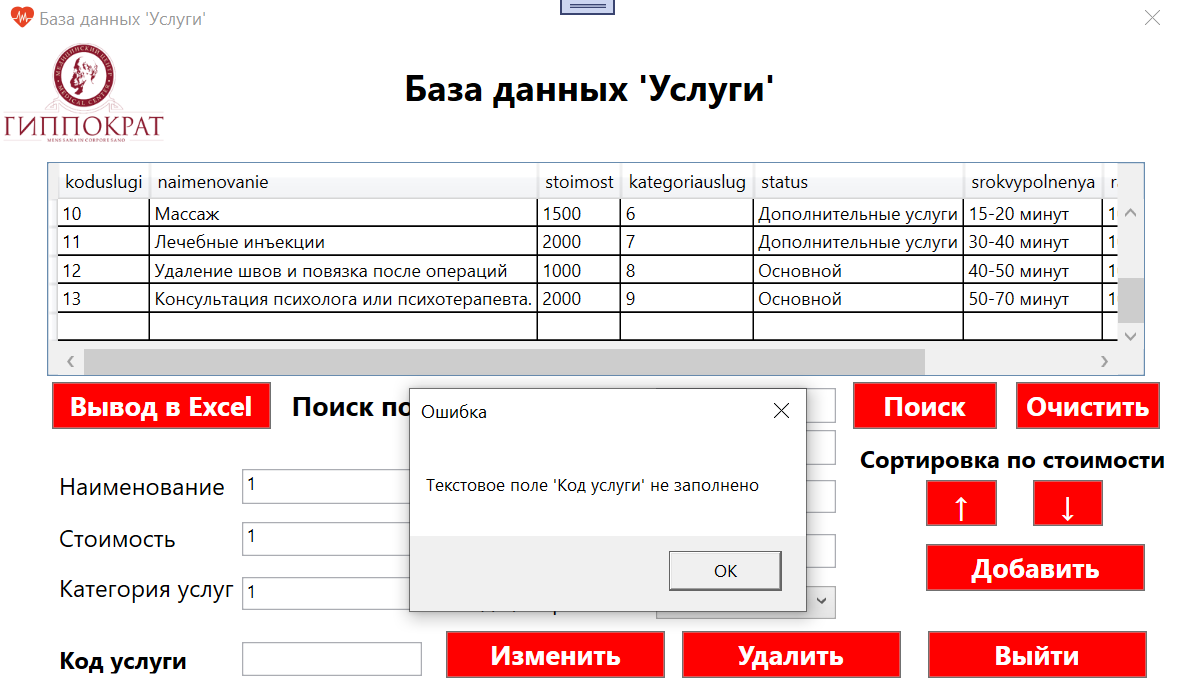


Рисунок 30 - Сообщение об ошибке изменения

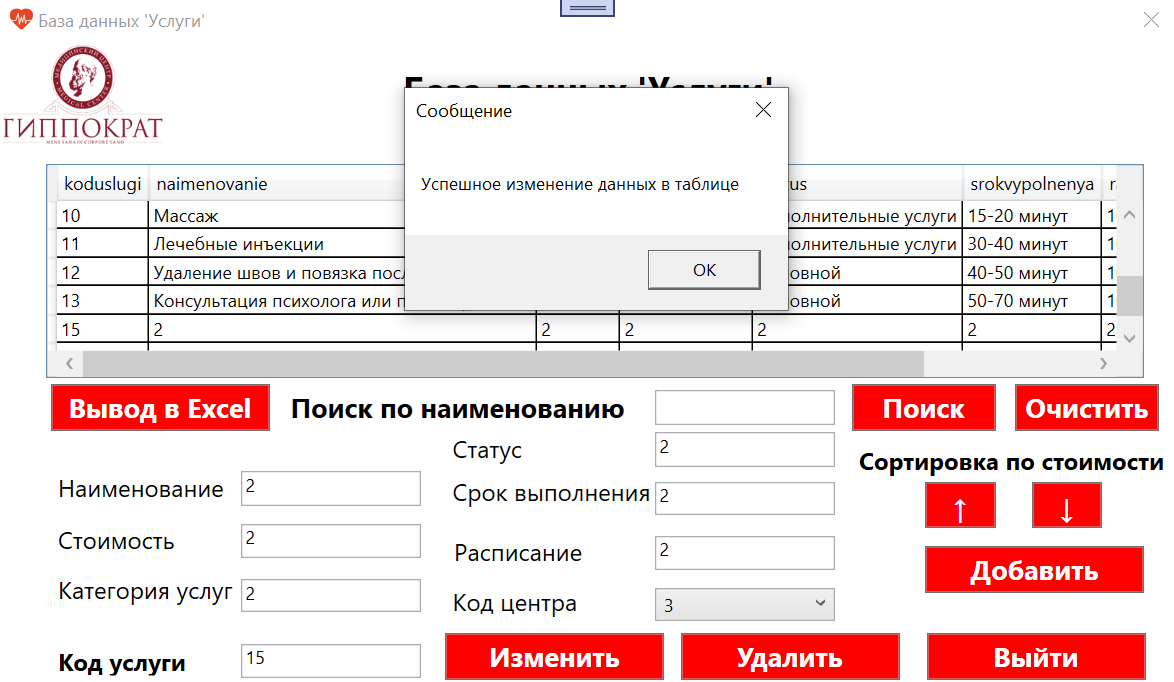


Рисунок 31 - Сообщение об успешном изменении данных в таблице

* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Услуги»**, в зависимости от значения в поле **«Код услуги»**, также очищает текстовые поля формы. Если оставить поле **«Код услуги»** пустым, то выйдет сообщение о том, что поле не было заполнено.

Ниже представлено окно с ошибкой удаления (рисунок 32):

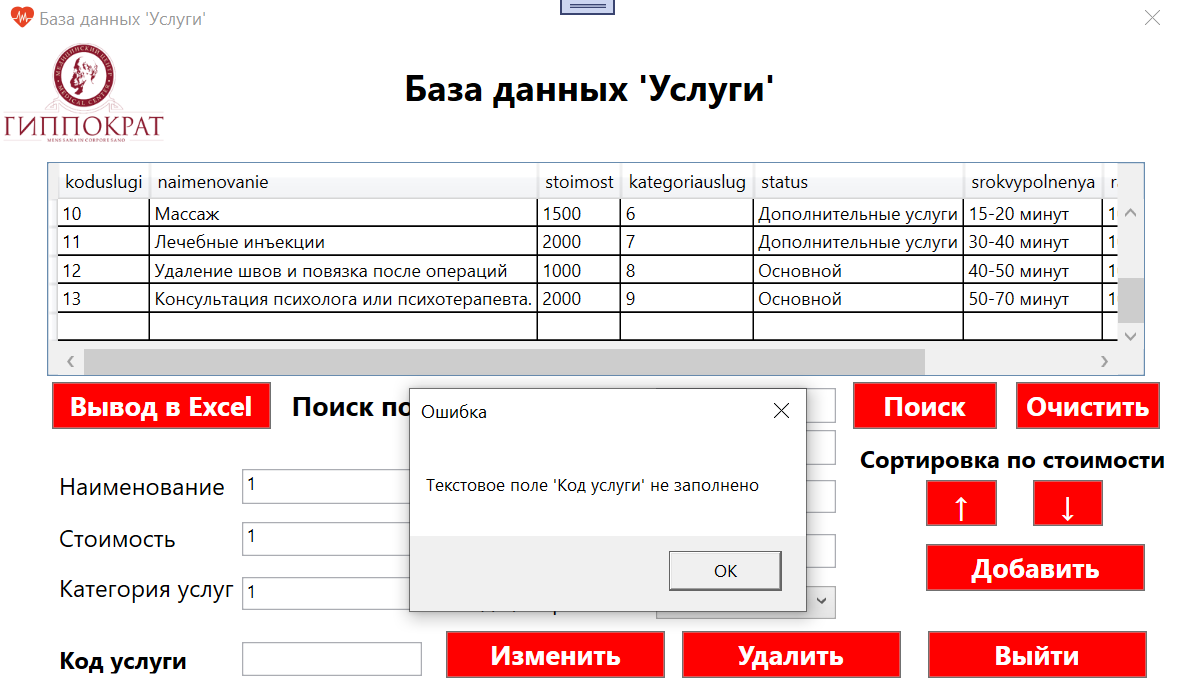


Рисунок 32 - Сообщение об ошибке удаления

Ниже представлено окно с успешным удалением данных в таблице (рисунок 33):

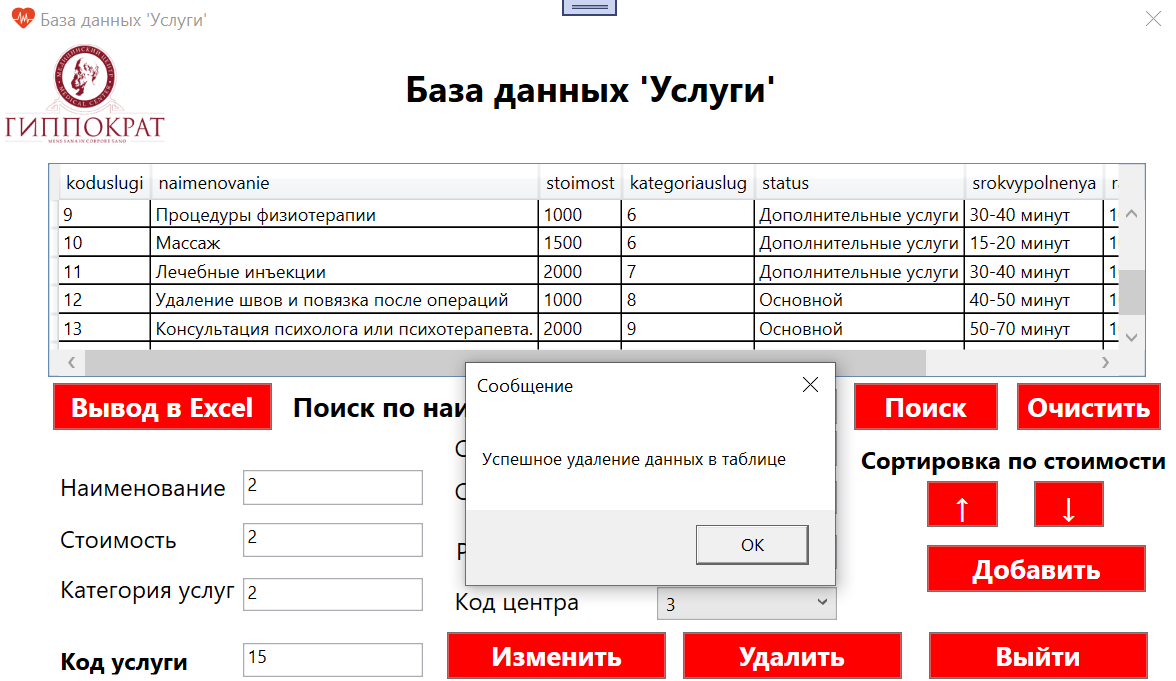


Рисунок 33 - Сообщение об успешном удалении данных в таблице

* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименованию услуги и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.

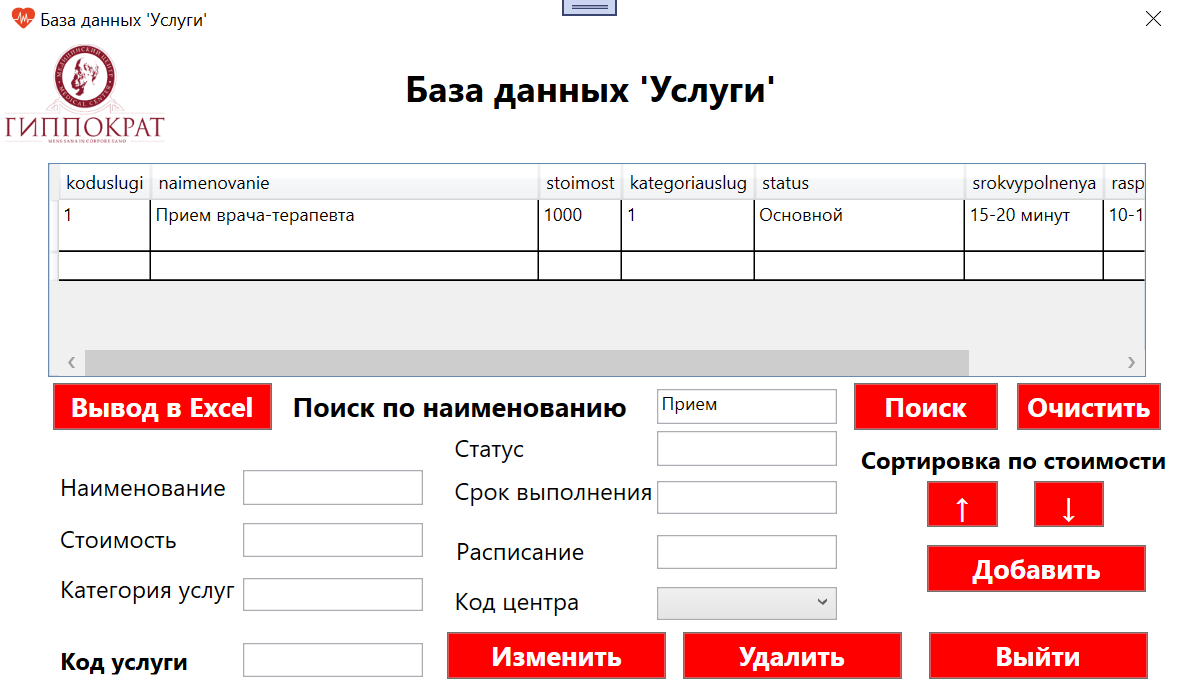


Рисунок 34 - Функция поиска

* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по наименованию» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Услуги» в формате файла Excel, что позволяет произвести дальнейшею печать.

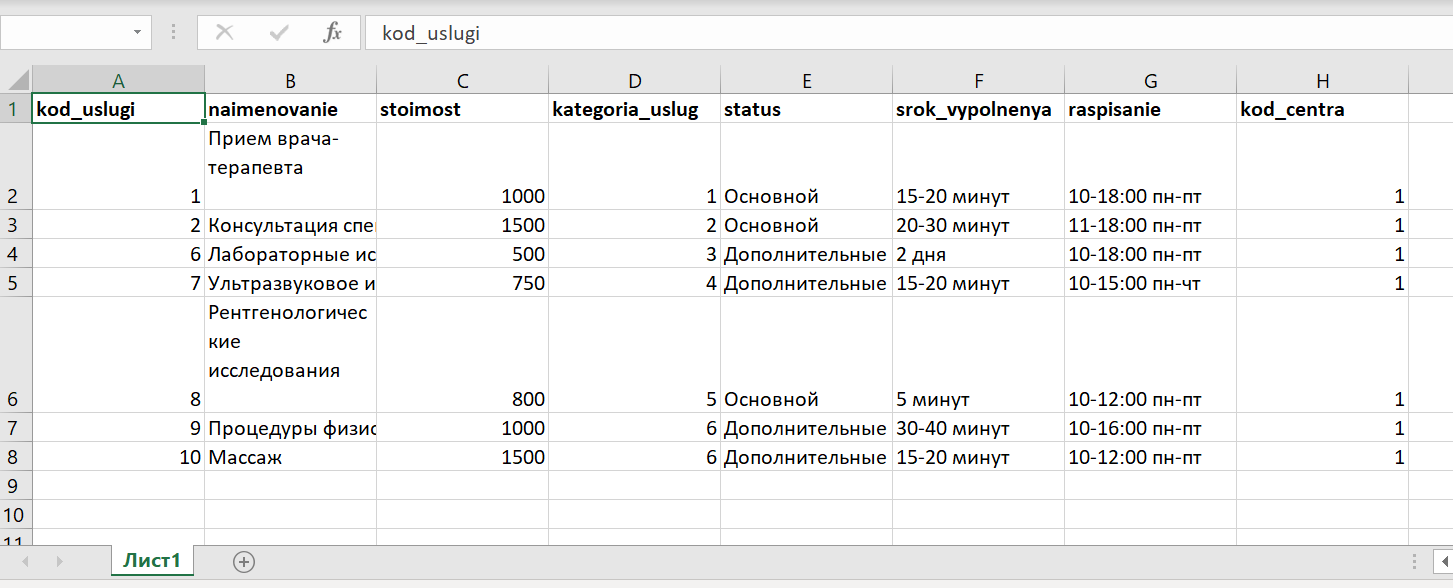


Рисунок 35 - Функция вывода в формате Excel

* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по стоимости услуги в порядке возрастания.

Ниже представлена работа функции сортировки в порядке возрастания (рисунок 36):

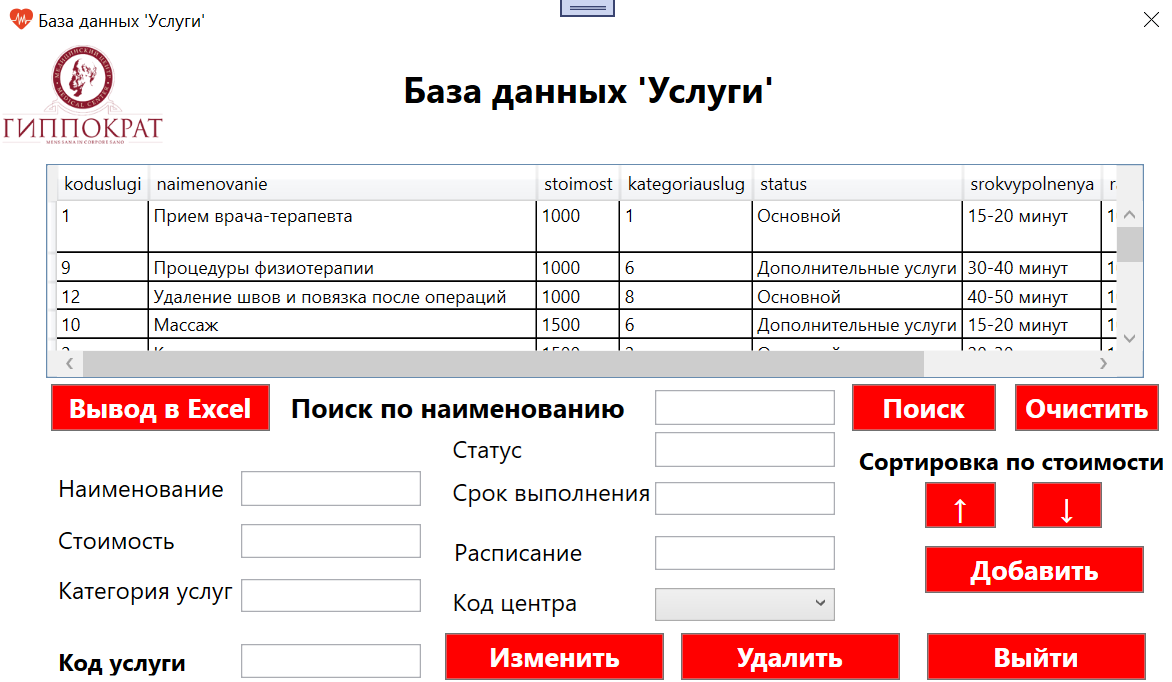


Рисунок 36 - Функция сортировки в порядке возрастания

* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по стоимости услуги в порядке убывания.

Ниже представлена работа функции сортировки в порядке убывания (рисунок 37):

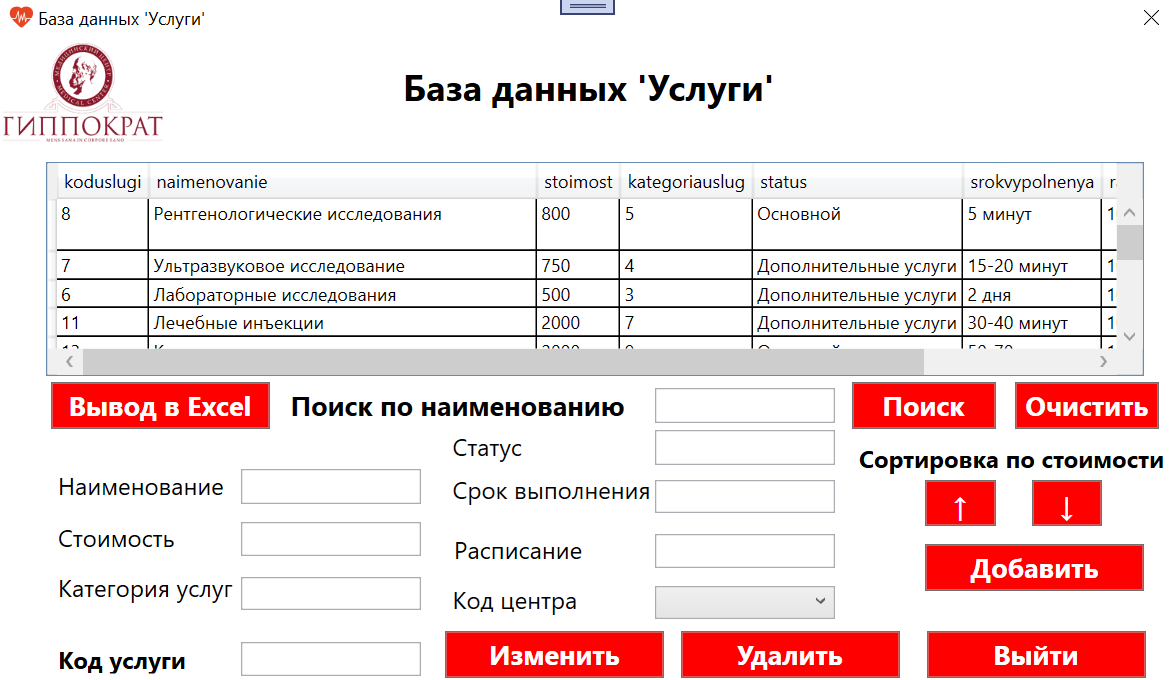


Рисунок 37 - Функция сортировки в порядке убывания

* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

1. При нажатии кнопки **«БД Клиенты»**, осуществится переход на соответствующую форму **«Клиенты»**.



Рисунок 38 - База данных «Клиент»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Клиент»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Клиент»**, в зависимости от значения в поле **«Код клиента»**. Если оставить поле **«Код клиента»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Клиент»**, в зависимости от значения в поле **«Код клиента»**, также очищает текстовые поля формы. Если оставить поле **«Код клиента»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименованию услуги и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле **«Поиск по фамилии»** и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных **«Клиент»** в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оформления клиента в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оформления клиента в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

1. При нажатии кнопки **«БД Сотрудники»**, осуществится переход на соответствующую форму **«Сотрудники»**.

Ниже представлено окно с базой данных «Сотрудники» (рисунок 39):



Рисунок 39 - База данных «Сотрудники»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Сотрудники»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Сотрудники»**, в зависимости от значения в поле «Код сотрудника». Если оставить поле **«Код сотрудника»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Сотрудники»**, в зависимости от значения в поле **«Код сотрудника»**, также очищает текстовые поля формы. Если оставить поле **«Код сотрудника»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению фамилии сотрудника и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле **«Поиск по фамилии»** и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных **«Сотрудники»** в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка «**Перезагрузка»** - перезагружает форму.
* Кнопка **«БД Статус»** - открывает форму базы данных **«Статус»**.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дню рождения в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дню рождения в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

1. При нажатии кнопки **«БД Статус»** в форме **«БД Сотрудники»**, откроется окно формы **«Статус»**.

Ниже представлено окно базы данных «Статус» (рисунок 40):

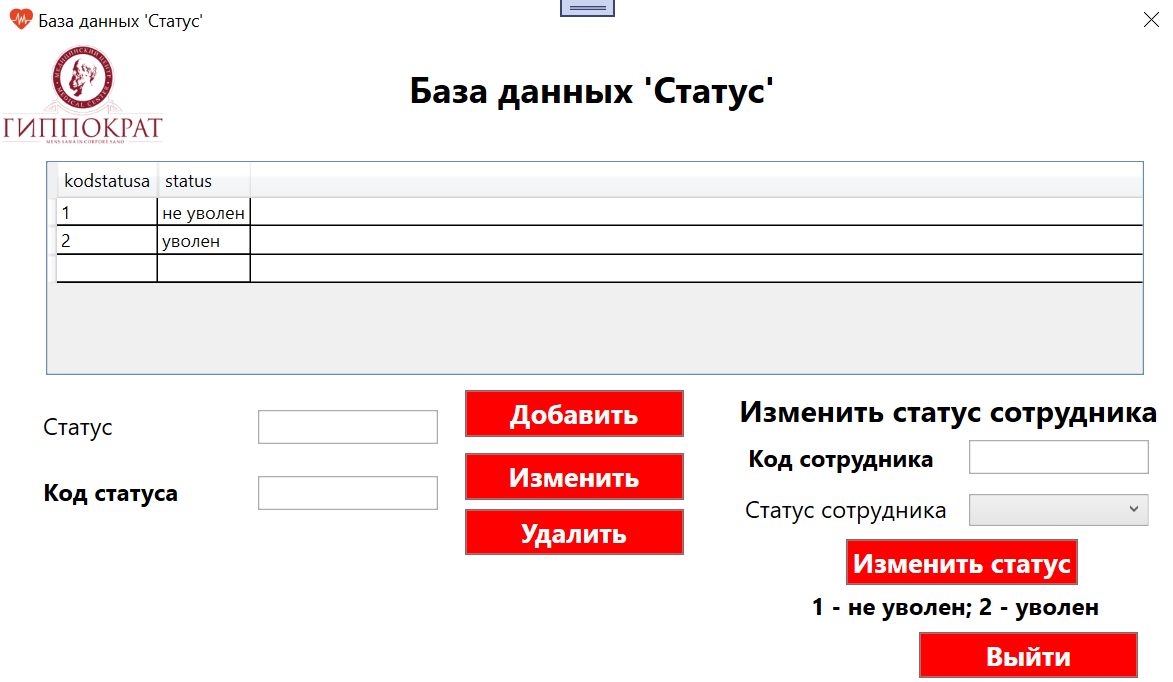


Рисунок 40 - База данных «Статус»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Статус»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Статус»**, в зависимости от значения в поле **«Код статуса»**. Если оставить поле **«Код статуса»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Статус»**, в зависимости от значения в поле **«Код статуса»**, также очищает текстовые поля формы. Если оставить поле **«Код статуса»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Изменить статус»** - изменяет статус сотрудника в таблице **«Сотрудники»**, в зависимости от введенного кода сотрудника и выбранного статуса сотрудника. Если оставить поле **«Код сотрудника»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Выйти»** - закрывает текущею форму.

1. При нажатии кнопки **«БД Оказанные услуги»**, осуществится переход на соответствующую форму **«Оказанные услуги»**.



Рисунок 41 - База данных «Оказанные услуги»

* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименования оказанной услуги и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле **«Поиск по наименованию»** и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных **«Оказанные услуги»** в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Рассчитать 1»** - производит автоматический расчет общей стоимости оказанных услуг.

Ниже представлена работа функции расчета общей стоимости (рисунок 42):



Рисунок 42 - Функция расчета общей стоимости

* Кнопка **«Рассчитать 2»** - производит автоматический расчет общего количества оказанных услуг.

Ниже представлена работа функции расчета общего количества (рисунок 43):



Рисунок 43 - Функция расчета общего количества

* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оказания услуги в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оказания услуги в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

1. При нажатии кнопки **«БД Приемы»**, осуществится переход на соответствующую форму **«Приемы»**.

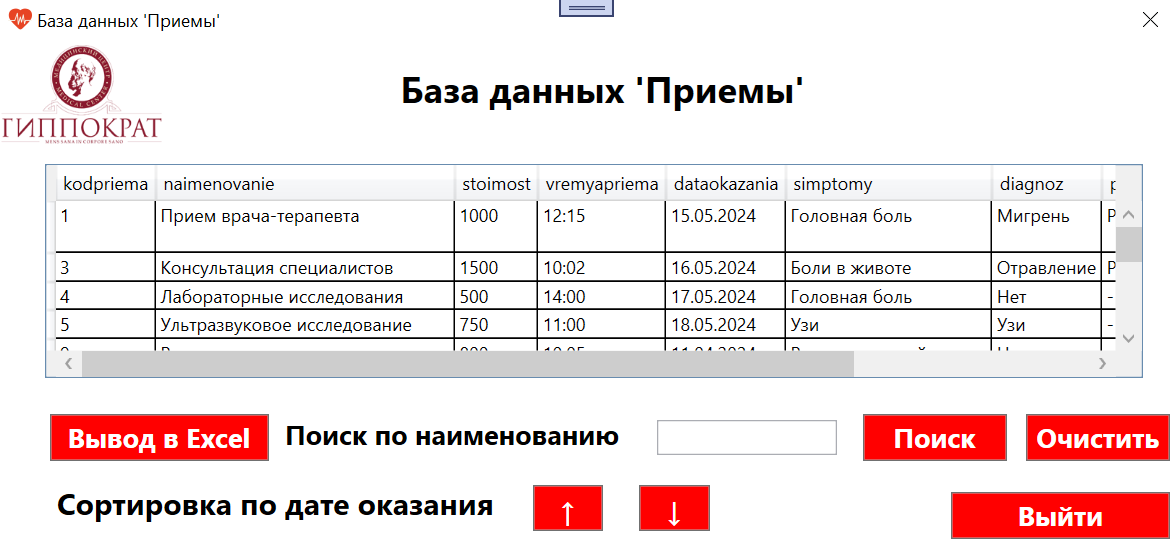


Рисунок 44 - База данных «Приемы»

* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименования приема и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле **«Поиск по наименованию»** и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных **«Приемы»** в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оказания приема в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оказания приема в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

1. При нажатии кнопки **«БД Медцентр»**, осуществится переход на соответствующую форму **«Медицинский центр»**.

Ниже представлено окно базы данных «Медицинский центр» (рисунок 45):

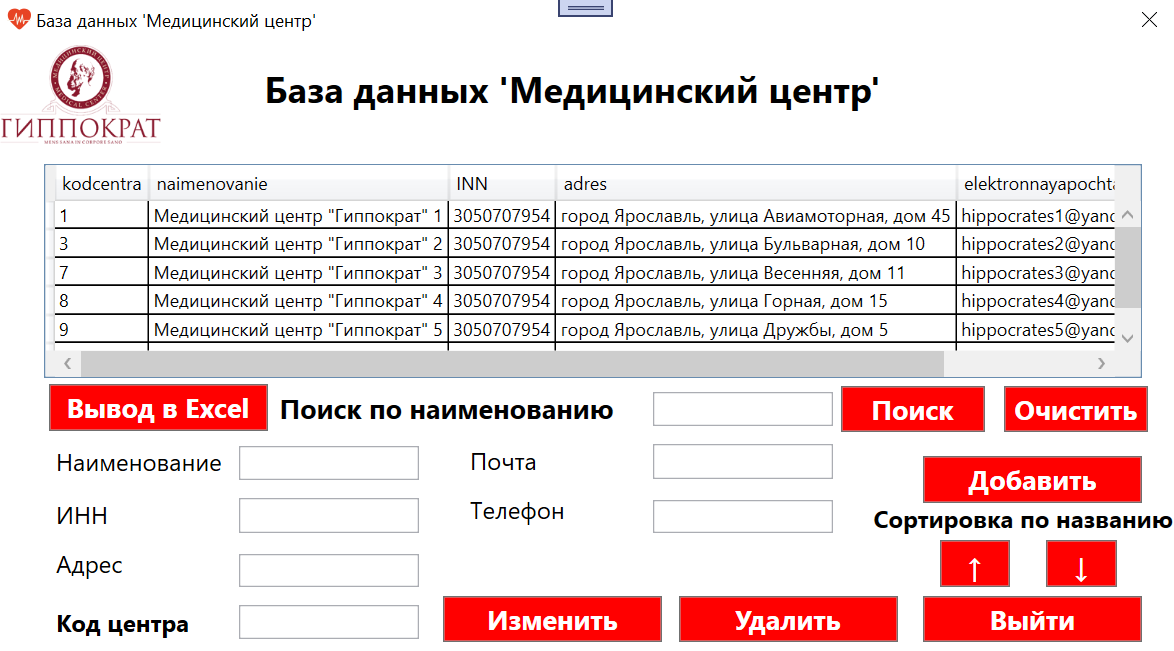


Рисунок 45 - База данных «Медицинский центр»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Медицинский центр»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Медицинский центр»**, в зависимости от значения в поле **«Код центра»**. Если оставить поле **«Код центра»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Медицинский центр»**, в зависимости от значения в поле **«Код центра»**, также очищает текстовые поля формы. Если оставить поле **«Код центра»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименования медицинского центра и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле **«Поиск по наименованию»** и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Медицинский центр» в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по названию медицинского центра в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по названию медицинского центра в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно директора.

1. **Инструкция пользователя (роль «Администратор»).**
2. Для запуска приложения перейдите в корневую папку приложения и запустите файл запуска \*.sln (в данном случае «Diplom\_Ohotnikov.sln»).

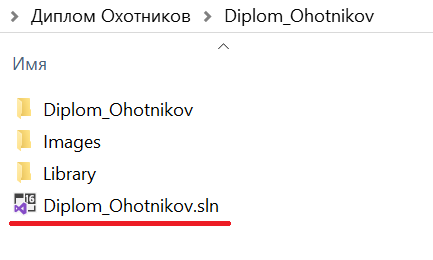


Рисунок 46 -Файл запуска

1. После запуска приложения Visual Studio, нажмите кнопку **«Пуск»** в верхней части экрана.

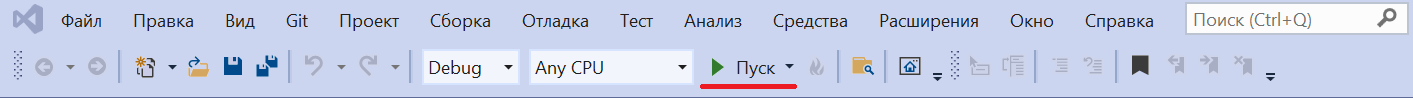


Рисунок 47 - Кнопка «Пуск»

1. Далее на форме **«Авторизация в информационную систему»** необходимо корректно ввести свой логин и пароль в соответствующие текстовые поля и нажать кнопку **«Авторизация»**. При успешной авторизации должно появится сообщение о входе в профиль **«Администратор»**.

Ниже представлено окно авторизации в профиль «Администратор» (рисунок 48):

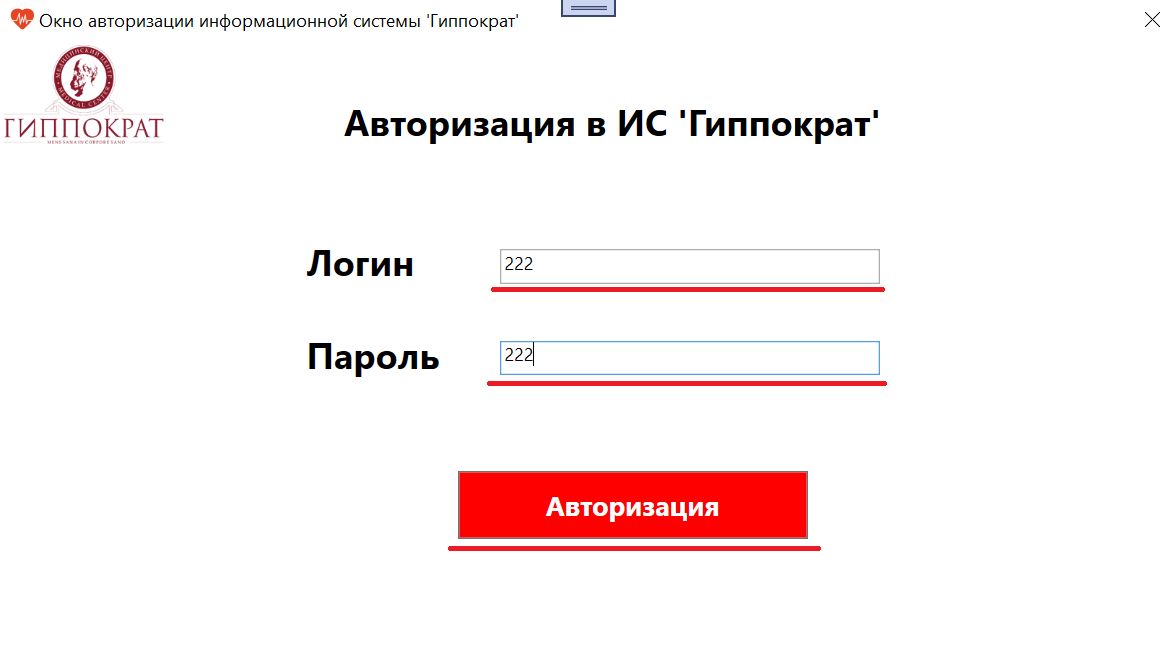


Рисунок 48 - Окно авторизации в профиль «Администратор»

Ниже представлено окно с сообщением об успешной авторизации (рисунок 49):

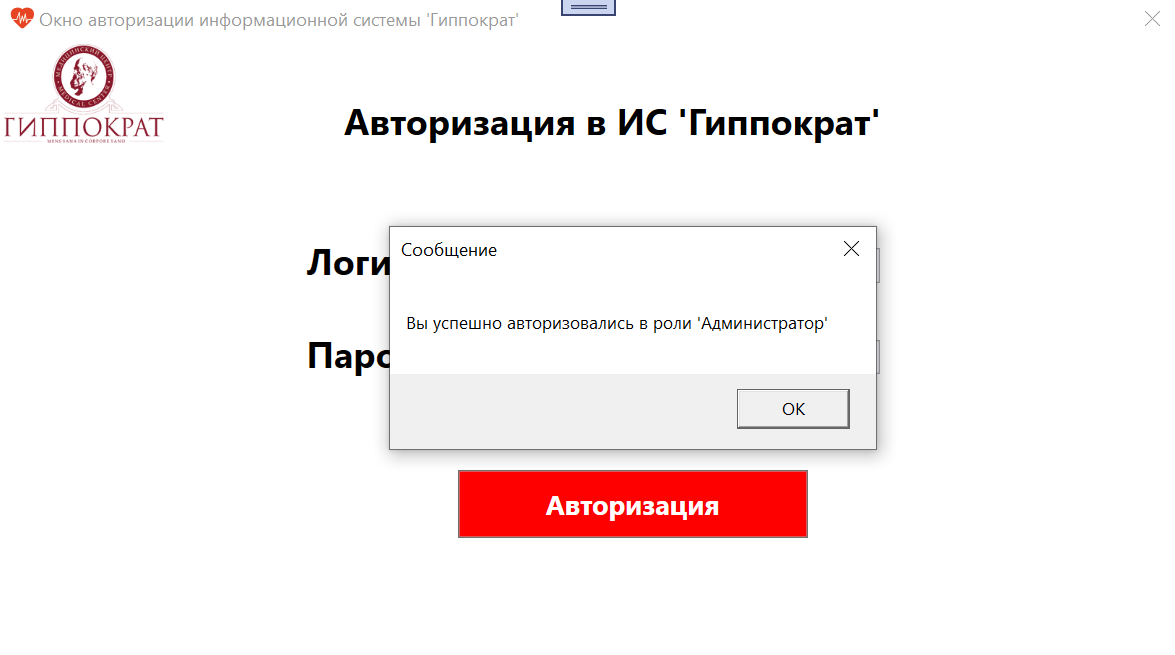


Рисунок 49 - Окно об успешной авторизации

1. Если вы ввели неправильные данные в текстовые поля на форме авторизации, появится сообщение об ошибке входа в профиль **«Администратор»**. Исправьте неправильные данные и нажмите **«Авторизация»**.

Ниже представлено окно с ошибкой процесса авторизации (рисунок 50):

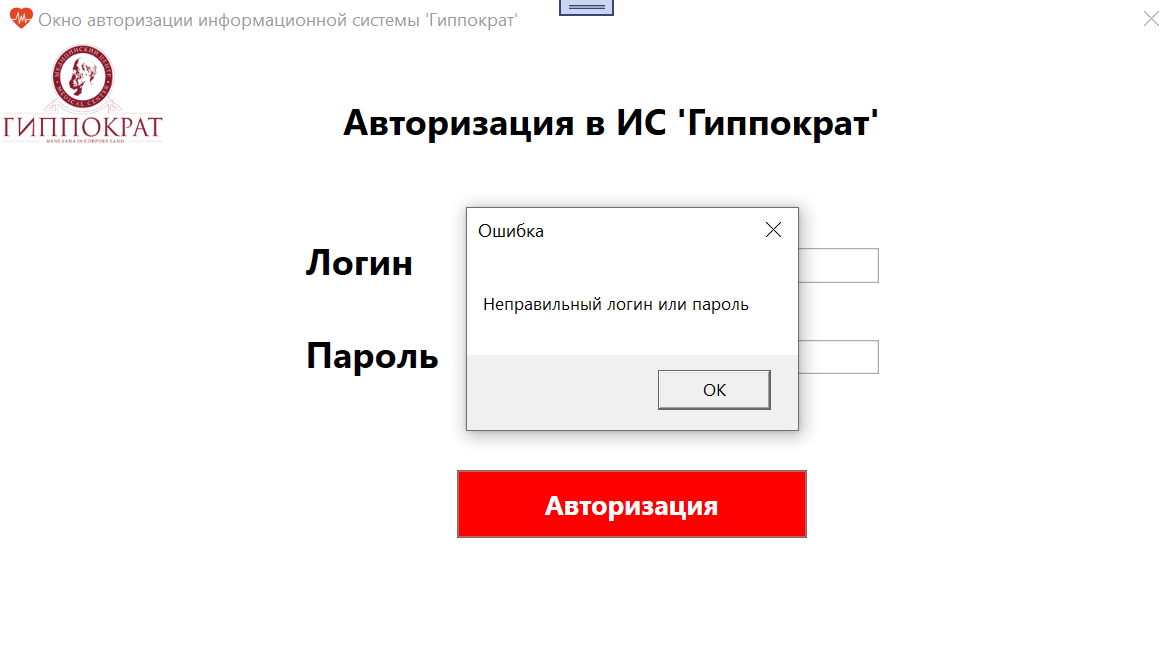


Рисунок 50 - Сообщение об ошибке процесса авторизации

1. После успешной авторизации Вы попадете в главное меню профиля **«Администратор»**, где необходимо нажать определенную кнопку для перехода на соответствующую кнопку. Доступны формы баз данных **«Услуги»**, **«Клиенты»**. Кнопка **«Выйти»** переведет пользователя обратно на форму авторизации и произведет выход из профиля.

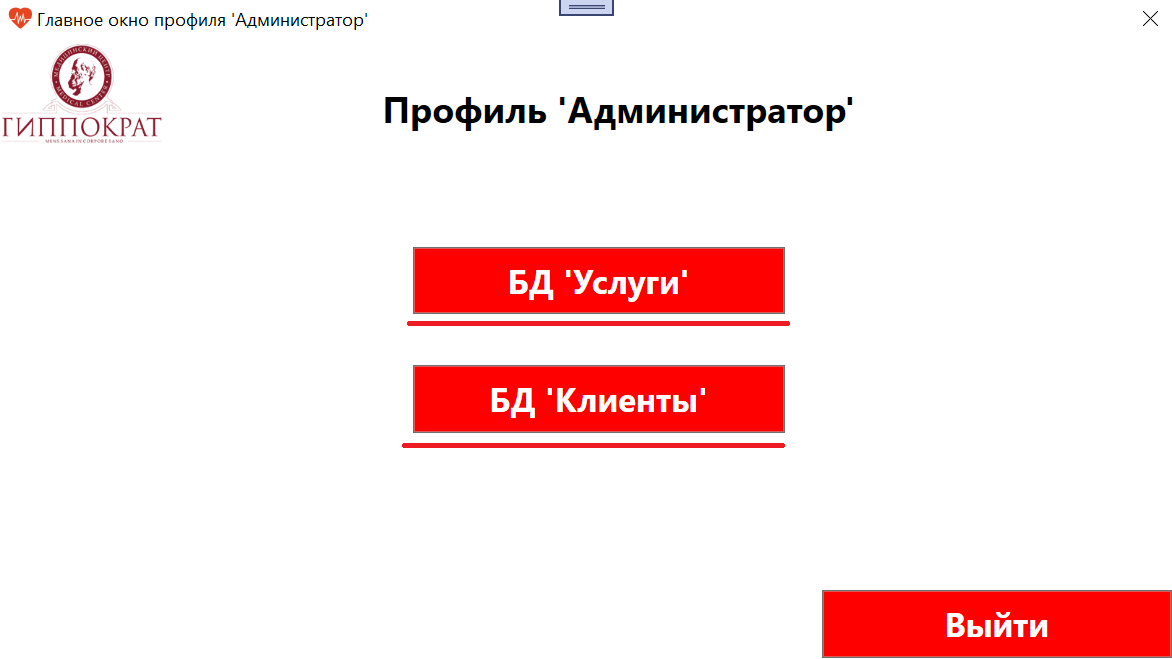


Рисунок 51 - Профиль пользователя «Администратор»

1. При нажатии кнопки **«БД Услуги»**, осуществится переход на соответствующую форму **«Услуги»**.



Рисунок 52 - База данных «Услуги»

* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименования услуги и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по наименованию» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Услуги» в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по стоимости услуги в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по стоимости услуги в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно администратора.

1. При нажатии кнопки **«БД Клиенты»**, осуществится переход на соответствующую форму **«Клиенты»**.

Ниже представлено окно базы данных «Клиент» (рисунок 53):



Рисунок 53 - База данных «Клиент»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Клиент»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Клиент»**, в зависимости от значения в поле **«Код клиента»**. Если оставить поле **«Код клиента»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Клиент»**, в зависимости от значения в поле «Код клиента», также очищает текстовые поля формы. Если оставить поле **«Код клиента»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению фамилии клиента и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по фамилии» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Клиенты» в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка «**Перезагрузка»** - перезагружает форму.
* Кнопка **«БД Приемы»** - открывает форму базы данных «Приемы».
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оформления клиента в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оформления клиента в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно администратора.

1. При нажатии кнопки **«БД Приемы»** в форме **«БД Клиенты»**, откроется окно формы **«Приемы»**.

Ниже представлено окно базы данных «Приемы» (рисунок 54):



Рисунок 54 - База данных «Приемы»

* Кнопка **«Записать»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Приемы»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименования приема и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по наименованию» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Приемы» в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«БД Приемы»** - открывает форму базы данных «Приемы».
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оказания приема в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оказания приема в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно администратора.

1. **Инструкция пользователя (роль «Врач»).**
2. Для запуска приложения перейдите в корневую папку приложения и запустите файл запуска \*.sln (в данном случае «Diplom\_Ohotnikov.sln»).

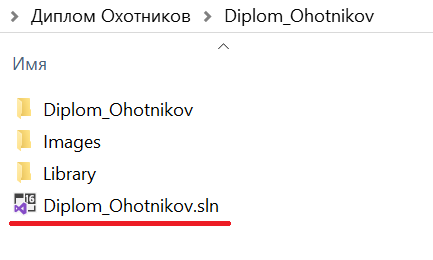


Рисунок 55 - Файл запуска

1. После запуска приложения Visual Studio, нажмите кнопку **«Пуск»** в верхней части экрана.

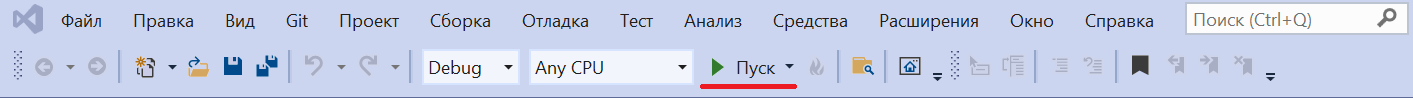


Рисунок 56 - Кнопка «Пуск»

1. Далее на форме **«Авторизация в информационную систему»** необходимо корректно ввести свой логин и пароль в соответствующие текстовые поля и нажать кнопку **«Авторизация»**. При успешной авторизации должно появится сообщение о входе в профиль **«Врач»**.

Ниже представлено окно авторизации в профиль «Врач» (рисунок 57):

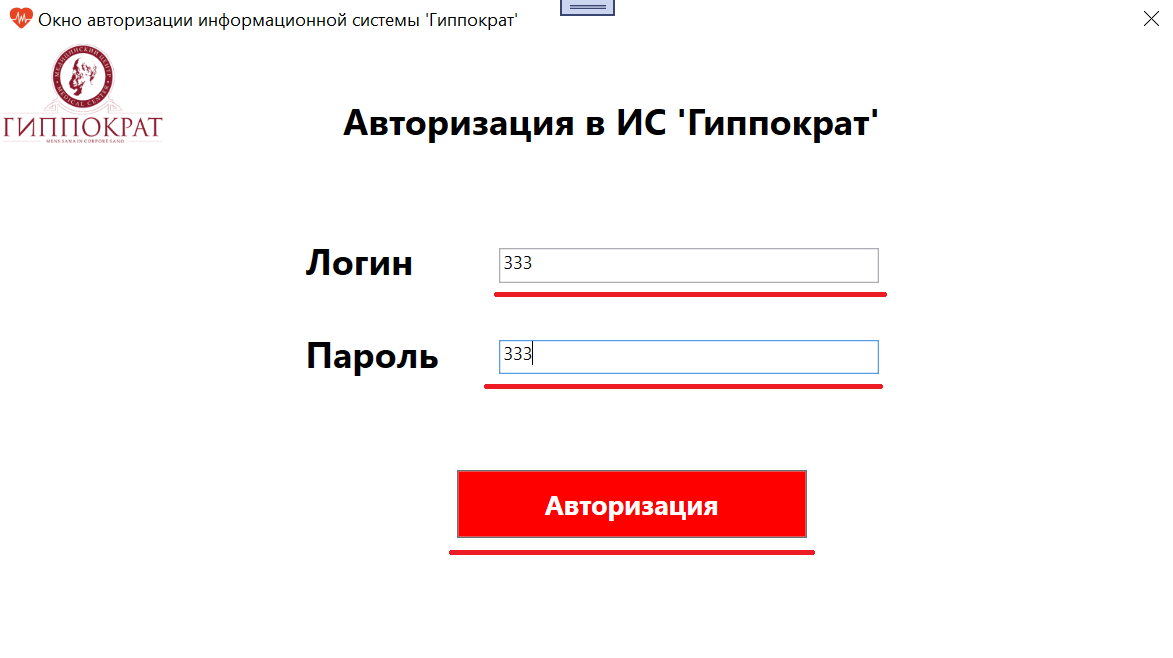


Рисунок 57 - Окно авторизации в профиль «Врач»

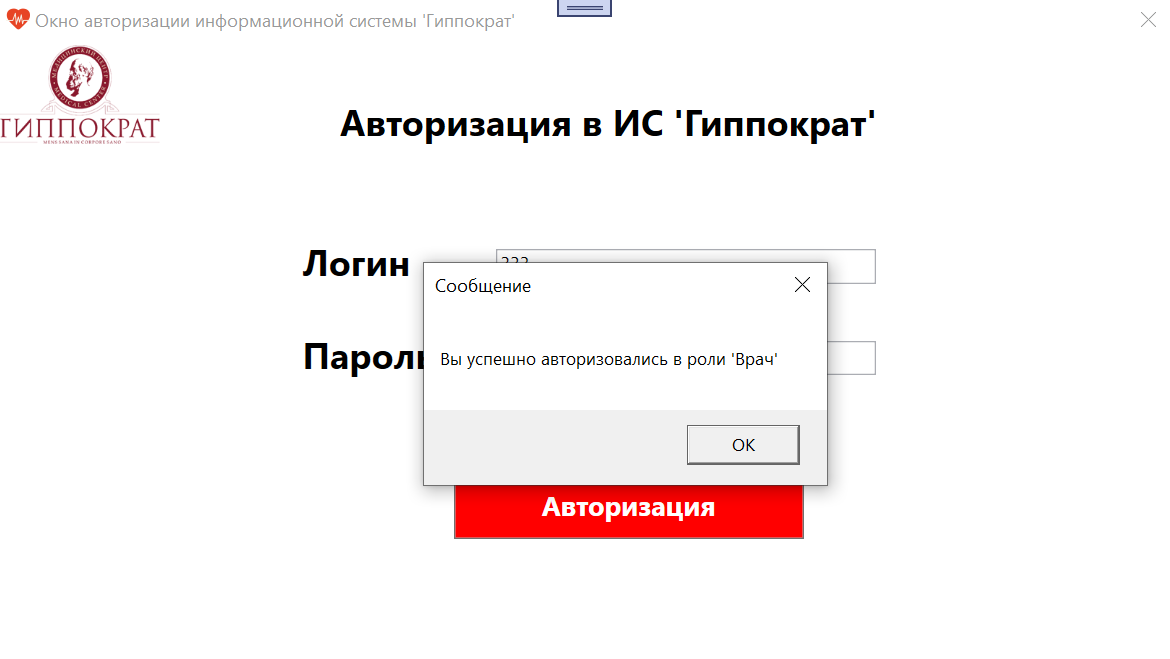


Рисунок 58 - Сообщение об успешной авторизации

1. Если вы ввели неправильные данные в текстовые поля на форме авторизации, появится сообщение об ошибке входа в профиль **«Врач»**. Исправьте неправильные данные и нажмите **«Авторизация»**.

Ниже представлено окно с ошибкой авторизации (рисунок 59):

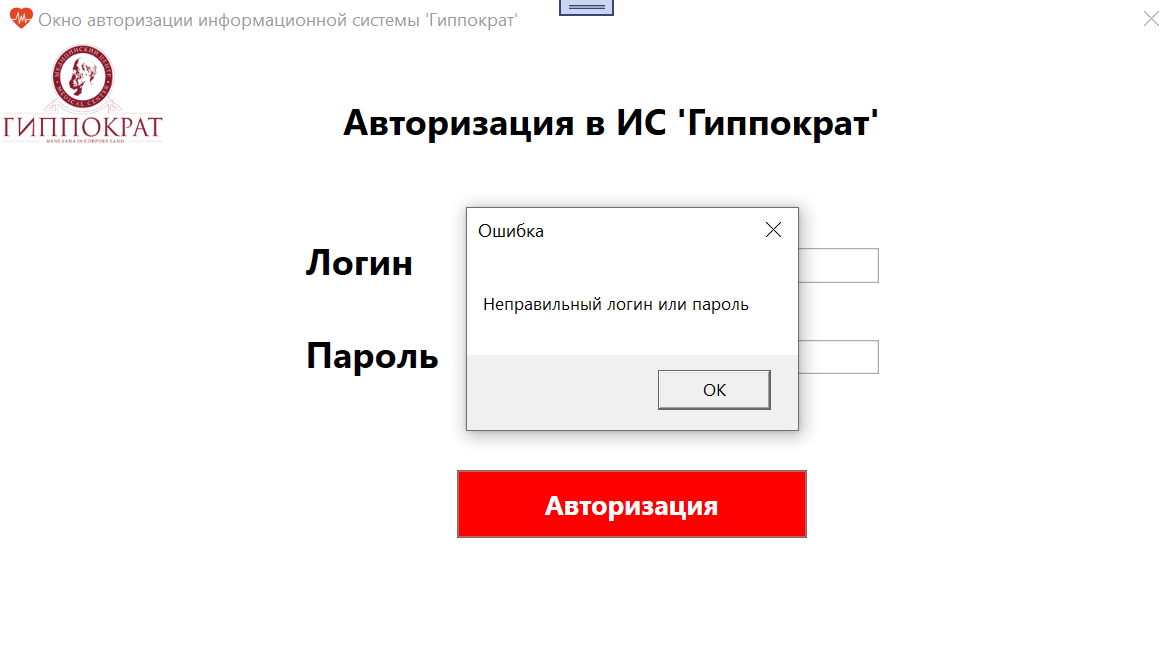


Рисунок 59 - Сообщение об ошибке процесса авторизации

1. После успешной авторизации Вы попадете в главное меню профиля **«Врач»**, где необходимо нажать определенную кнопку для перехода на соответствующую кнопку. Доступны формы баз данных **«Услуги»**, **«Приемы», «Оказанные услуги»**. Кнопка **«Выйти»** переведет пользователя обратно на форму авторизации и произведет выход из профиля.



Рисунок 60 - Профиль пользователя «Врач»

1. При нажатии кнопки **«БД Услуги»**, осуществится переход на соответствующую форму **«Услуги»**.

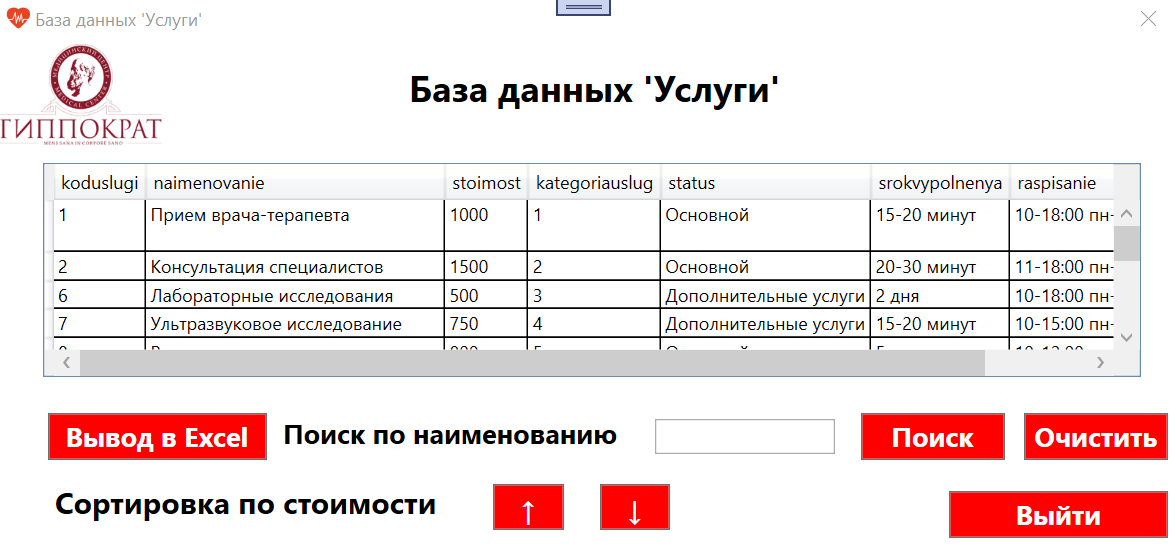


Рисунок 61 - База данных «Услуги»

* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименования услуги и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по наименованию» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Услуги» в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по стоимости услуги в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по стоимости услуги в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно врача.

1. При нажатии кнопки **«БД Приемы»**, осуществится переход на соответствующую форму **«Приемы»**.



Рисунок 62 - База данных «Приемы»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Приемы»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Приемы»**, в зависимости от значения в поле **«Код приема»**. Если оставить поле **«Код приема»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Приемы»**, в зависимости от значения в поле **«Код приема»**, также очищает текстовые поля формы. Если оставить поле **«Код приема»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному наименованию приема и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по наименованию» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных «Приемы» в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка «**Перезагрузка»** - перезагружает форму.
* Кнопка **«БД Клиент»** - открывает форму базы данных «Клиенты».
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оказания приема в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оказания приема в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно врача.

1. При нажатии кнопки **«БД Клиент»** в форме **«БД Приемы»**, откроется окно формы **«Клиенты»**.

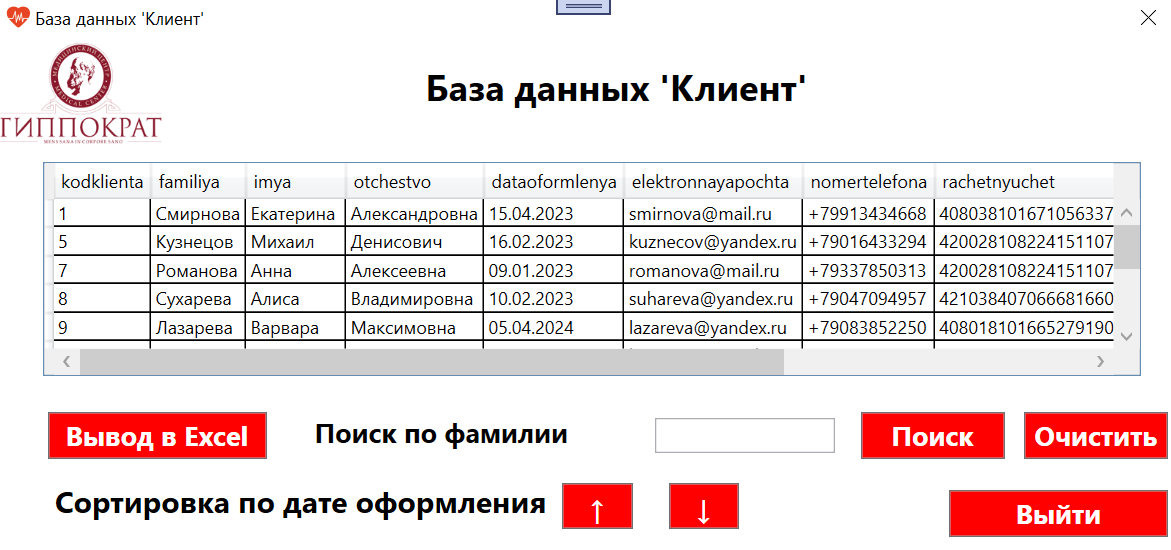


Рисунок 63 - База данных «Клиент»

* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению фамилии клиента и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле **«Поиск по фамилии»** и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных **«Клиенты»** в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оформления клиента в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оформления клиента в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно врача.

1. При нажатии кнопки **«БД Оказанные услуги»**, осуществится переход на соответствующую форму **«Оказанные услуги»**.



Рисунок 64 - База данных «Оказанные услуги»

* Кнопка **«Добавить»** - обрабатывает введенный текст в текстовых полях, создает подключение и добавляет данные в таблицу **«Оказанные услуги»**. Если одно из полей останется пустым, то выйдет сообщение о том, что поле одно из текстовых полей не было заполнено.
* Кнопка **«Изменить»** - обрабатывает введенный текст в текстовых полях, создает подключение и изменяет данные в таблице **«Оказанные услуги»**, в зависимости от значения в поле «**Код оказанной услуги»**. Если оставить поле **«Код оказанной услуги»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Удалить»** - создает подключение и удаляет данные в таблице **«Оказанные услуги**», в зависимости от значения в поле **«Код оказанной услуги»**, также очищает текстовые поля формы. Если оставить поле **«Код оказанной услуги»** пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Поиск»** - производит автоматический поиск по введенному значению наименованию оказанной услуги и обновляет таблицу. Если оставить текстовое поле поиска пустым, то выйдет сообщение о том, что поле не было заполнено.
* Кнопка **«Очистить»** - очищает текстовое поле «Поиск по наименованию» и обновляет форму.
* Кнопка **«Вывод в Excel»** - сохраняет базу данных **«Оказанные услуги»** в формате файла Excel, что позволяет произвести дальнейшею печать.
* Кнопка **«Вверх ↑»** - производит автоматическую сортировку по дате оказанной услуги в порядке возрастания.
* Кнопка **«Вниз ↓»** - производит автоматическую сортировку по дате оказанной услуги в порядке убывания.
* Кнопка **«Выйти»** - закрывает текущею форму и открывает главное окно врача.

**Заключение**

Вне зависимости от того, каким типом медицинской организации управляете – больница, поликлиника, медицинский центр, в любом случае, по мере расширения предприятия и кадров, вы столкнетесь с рядом стандартных для этого бизнеса проблем, которые включают в себя:

* Систематизация и контроль работы сотрудников;
* Учет осуществленных медицинских работ, а также оказываемых услуг, в электронном виде;
* Ведение и учет базы клиентов и сотрудников.

Целью, в рамках разработки дипломной работы, была разработка информационной системы для медицинского центра «Гиппократ».

**Для реализации и осуществления данной цели, способствовало решение следующих задач:**

* Проведен анализ предметной области и сформулированы требования к информационной системе.
* Проведен анализ аналогов и прототипов информационной системы.
* Проведено описание интерфейса и архитектуры разрабатываемого продукта.
* Было разработано приложение в соответствии с требованиями разработки и заявленным функционалом.
* Была разработана база данных информационной системы.
* Были разработаны требования к обеспечению безопасности информации системы.
* Проведено тестирование и проверка работоспособности внутренних компонентов программного обеспечения.
* Были разработаны требования к аппаратным и программным средствам для функционирования системы.
* Были разработаны инструкции пользователя и администратора.

**Во время разработки дипломного проекта, были применены навыки:**

* определять функциональные задачи;
* выявлять проблему;
* разрабатывать решения для ее устранения;
* проектировать информационную систему приложения;
* создавать интерфейс приложения;
* оформлять проектную документацию;
* оценивать результаты.

**Оценка готовности анализируемого предприятия:**

* Инфраструктура: предприятие имеет современное оборудования, удобное расположение и оно доступно для пациентов, оно соответствует требованиям размещения, безопасности и санитарии медицинских центров.
* Персонал: сотрудники обладают высокой квалификацией и имеют большой опыт работы, была выявлена их готовность к внедрению новых методик и технологий.
* Процессы: предприятие готово к более эффективной и продуктивной работе медицинского центра, внедрены системы контроля и управления качеством, установлена готовность к внедрению новых стандартов и регламентов.

**Перечень мероприятий, создающих условия для успешного внедрения изменений, может включать в себя следующие шаги:**

* Обучение персонала: проведение тренингов и семинаров по новой информационной системе, повышение квалификации специалистов.
* Обновление оборудования: закупка современного оборудования и техники для улучшения качества медицинских услуг и увеличении скорости работы информационной системы.
* Создание системы контроля и управления качеством: разработка и внедрение стандартов и процедур контроля качества медицинских услуг, обратная связь с сотрудниками.
* Развитие маркетинга и PR: проведение рекламных кампаний, участие в выставках и конференциях, партнерство с другими медицинскими центрами и организациями.

В целом, успешное внедрение изменений требует комплексного подхода и согласования действий с различными отделами и специалистами медицинского центра.

Перспективы разработки приложения для медицинских центров кажутся очень обещающими. С учетом быстрого развития технологий и все большего числа компаний, которые предпочитают пользоваться автоматизированными информационными системами для решения производственных задач, приложение будет обладать большим спросом на рынке программного обеспечения.

В целом, рынок разработки приложений для медицинских центров пока еще не насыщен, и есть хорошие возможности для успешного внедрения такого проекта. Важно правильно определить целевую аудиторию и потребности пользователей, а также сделать упор на высокую функциональность, удобство использования и безопасность приложения.

Поставленная цель и задачи при создании дипломного проекта были полностью выполнены. Программный продукт выполнен в полном объеме и соответствует всем заявленным требованиям и функционалом.

**Список литературы**

1. Гриффитс И. Программируем на C# 8.0. Разработка приложений [Текст] / Гриффитс И. — 1-е изд. — Санкт-Петербург: Питер, 2021 — 944 c.
2. Прайс, М. Д. C# 10 и .NET 6. Современная кросс-платформенная разработка [Текст] / М. Д. Прайс — 6-е изд. — Санкт-Петербург: Питер, 2023 — 1019 c.
3. Стиллмен Э., Грин Д. Изучаем C# [Текст] / Стиллмен Э., Грин Д. — 4-е изд. — Санкт-Петербург: Питер, 2022 — 769 c.
4. Вильчинский Н., Симан М., ван Дерсен С. Внедрение зависимостей на платформе .NET [Текст] / Вильчинский Н., Симан М., ван Дерсен С. — 2-е изд. — Санкт-Петербург: Питер, 2021 — 608 c.
5. Фленов М. Библия C# [Текст] / Фленов М. — 5-е изд. — Санкт-Петербург: БХВ-Петербург, 2022 — 463 c.
6. Арораа Г., Чилберто Д. Паттерны проектирования для C# и платформы .NET Core [Текст] / Арораа Г., Чилберто Д. — 1-е изд. — Санкт-Петербург: Питер, 2021 — 352 c.
7. Фленов М. C# глазами хакера [Текст] / Фленов М. — 1-е изд. — Санкт-Петербург: БХВ-Петербург, 2023 — 221 c.
8. Тидвелл Д., Брюэр Ч., Валенсия Э. Разработка интерфейсов. Паттерны проектирования [Текст] / Тидвелл Д., Брюэр Ч., Валенсия Э. — 3-е изд. — Санкт-Петербург: Питер, 2022 — 560 c.
9. Фримен А. ASP.NET Core 3 с примерами на C# для профессионалов [Текст] / Фримен А. — 8-е изд. — Москва: Вильямс, 2021 — 1184 c.
10. Щербак, А. В. Информационная безопасность. Учебник для СПО [Текст] / А. В. Щербак — 1-е изд. — Москва: ЮРАЙТ, 2022 — 260 c.

**Электронные ресурсы**

1. Иванов П.В. Wpf И C# | Полное Руководство - Metanit.com / Иванов П.В. [Электронный ресурс] // Metanit : [сайт]. — URL: https://metanit.com/sharp/wpf/ (дата обращения: 01.06.2024).
2. Иванов П.В. ADO.NET и MS SQL Server / Иванов П.В. [Электронный ресурс] // Metanit : [сайт]. — URL: https://metanit.com/sharp/adonetcore/2.4.php (дата обращения: 01.06.2024).
3. Дударь Г. Изучение C# WPF / Дударь Г. [Электронный ресурс] // itProger : [сайт]. — URL: https://itproger.com/course/csharp-wpf (дата обращения: 01.06.2024).
4. Баева Е. C#: всё о языке программирования, особенности и применение / Баева Е. [Электронный ресурс] // Skypro : [сайт]. — URL: https://sky.pro/media/c-vsyo-o-yazyke-programmirovaniya-osobennosti-i-primenenie (дата обращения: 02.06.2024).
5. Джордж Э. Введение в WPF / Джордж Э. [Электронный ресурс] // Microsoft Learn : [сайт]. — URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/introduction-to-wpf?view=netframeworkdesktop-4.8 (дата обращения: 02.06.2024).
6. Джордж Э. Разработка приложений - WPF .NET Framework / Джоржд Э. [Электронный ресурс] // Microsoft Learn : [сайт]. — URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/app-development/?view=netframeworkdesktop-4.8 (дата обращения: 03.06.2024).
7. Каримов А. WPF Подключение к базе данных MS SQL Server / Каримов А. [Электронный ресурс] // Твои руки : [сайт]. — URL: https://you-hands.ru/2018/08/31/wpf-podklyuchenie-k-baze-dannyx-ms-sql-server/ (дата обращения: 03.06.2024).
8. Сафронов М. База данных MS SQL Server / Сафронов М. [Электронный ресурс] // Skillbox Media : [сайт]. — URL: https://skillbox.ru/media/code/baza-dannykh-ms-sql-server-chto-eto-zachem-nuzhna-kak-poyavilas-i-chem-khorosha/ (дата обращения: 03.06.2024).
9. Петров Г. Быстрый старт с WPF. Часть 1. / Петров Г. [Электронный ресурс] // Habr : [сайт]. — URL: https://habr.com/ru/articles/427325/ (дата обращения: 04.06.2024).
10. Джордж Э. Безопасность - WPF .NET Framework / Джордж Э. [Электронный ресурс] // Microsoft Learn : [сайт]. — URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/security-wpf?view=netframeworkdesktop-4.8 (дата обращения: 04.06.2024).

**Примечание**

**Код формы авторизации:**

<Window x:Class="Diplom\_Ohotnikov.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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="Окно авторизации информационной системы 'Гиппократ' " Height="450" Width="800">

<Grid>

<Label Content="Авторизация в ИС 'Гиппократ'" HorizontalAlignment="Left" Margin="225,32,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<Image Margin="3,2,675.333,344.667" Source="logo.png" Stretch="Fill"/>

<Label Content="Логин" HorizontalAlignment="Left" Margin="200,125,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<Label Content="Пароль" HorizontalAlignment="Left" Margin="200,187,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<Button x:Name="button1" Content="Авторизация" HorizontalAlignment="Left" Margin="306,286,0,0" VerticalAlignment="Top" Width="233" Height="45" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="334,138,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="253"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="334,199,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="253"/>

</Grid>

</Window>

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;

using System.Data.SqlClient;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

string connect = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

string sql = "SELECT \* FROM Diplom\_Sotrudnik WHERE login = '" + textBox1.Text + "' and parol = '" + textBox2.Text + "'";

SqlConnection sqlConnection = new SqlConnection(connect);

SqlCommand sqlCommand = new SqlCommand(sql, sqlConnection);

sqlConnection.Open();

SqlDataReader dr = sqlCommand.ExecuteReader();

string login = "null";

string parol = "null";

string rol = "null";

while (dr.Read())

{

login = dr.GetString(11);

parol = dr.GetString(12);

rol = dr.GetString(5);

}

sqlConnection.Close();

if ((login == "null") || (parol == "null"))

{

MessageBox.Show(string.Format("Неправильный логин или пароль"), "Ошибка");

}

else

{

if (rol == "Директор")

{

MessageBox.Show(string.Format("Вы успешно авторизовались в роли 'Директор'"), "Сообщение");

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

else if (rol == "Администратор")

{

MessageBox.Show(string.Format("Вы успешно авторизовались в роли 'Администратор'"), "Сообщение");

glavnaya\_administrator form = new glavnaya\_administrator();

form.Show();

this.Close();

}

else if (rol == "Врач")

{

MessageBox.Show(string.Format("Вы успешно авторизовались в роли 'Врач'"), "Сообщение");

glavnaya\_vrach form = new glavnaya\_vrach();

form.Show();

this.Close();

}

}

}

}

}

**Код главной формы директора:**

<Window x:Class="Diplom\_Ohotnikov.glavnaya\_director"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="Главное окно профиля 'Директор'" Height="450" Width="800">

<Grid>

<Image Margin="3,2,675.333,344.667" Source="logo.png" Stretch="Fill"/>

<Label Content="Профиль 'Директор'" HorizontalAlignment="Left" Margin="275,24,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<Button x:Name="button1" Content="БД 'Медцентр'" HorizontalAlignment="Left" Margin="473,255,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Button x:Name="button2" Content="БД 'Услуги'" HorizontalAlignment="Left" Margin="102,110,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button3" Content="БД 'Клиенты'" HorizontalAlignment="Left" Margin="102,181,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button4" Content="БД 'Сотрудники'" HorizontalAlignment="Left" Margin="102,255,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button5" Content="БД 'Оказанные услуги'" HorizontalAlignment="Left" Margin="473,110,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Button x:Name="button6" Content="БД 'Приёмы'" HorizontalAlignment="Left" Margin="473,181,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button7" Content="Выйти" HorizontalAlignment="Left" Margin="550,366,0,0" VerticalAlignment="Top" Width="233" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

</Grid>

</Window>

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 Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для glavnaya\_director.xaml

/// </summary>

public partial class glavnaya\_director : Window

{

public glavnaya\_director()

{

InitializeComponent();

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

uslugi form = new uslugi();

form.Show();

this.Close();

}

private void button7\_Click(object sender, RoutedEventArgs e)

{

MainWindow form = new MainWindow();

form.Show();

this.Close();

}

private void button6\_Click(object sender, RoutedEventArgs e)

{

medcenter form = new medcenter();

form.Show();

this.Close();

}

private void button2\_Click(object sender, RoutedEventArgs e)

{

klient form = new klient();

form.Show();

this.Close();

}

private void button3\_Click(object sender, RoutedEventArgs e)

{

sotrudniki form = new sotrudniki();

form.Show();

this.Close();

}

private void button4\_Click(object sender, RoutedEventArgs e)

{

okazannye\_uslugi form = new okazannye\_uslugi();

form.Show();

this.Close();

}

private void button5\_Click(object sender, RoutedEventArgs e)

{

priem form = new priem();

form.Show();

this.Close();

}

}

}

**Код главной формы администратора:**

<Window x:Class="Diplom\_Ohotnikov.glavnaya\_administrator"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="Главное окно профиля 'Администратор'" Height="450" Width="800">

<Grid>

<Image Margin="3,2,675.333,344.667" Source="logo.png" Stretch="Fill"/>

<Label Content="Профиль 'Администратор'" HorizontalAlignment="Left" Margin="252,24,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<Button x:Name="button3" Content="Выйти" HorizontalAlignment="Left" Margin="550,366,0,0" VerticalAlignment="Top" Width="233" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button1" Content="БД 'Услуги'" HorizontalAlignment="Left" Margin="277,137,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="БД 'Клиенты'" HorizontalAlignment="Left" Margin="277,216,0,0" VerticalAlignment="Top" Width="248" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

</Grid>

</Window>

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 Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для glavnaya\_administrator.xaml

/// </summary>

public partial class glavnaya\_administrator : Window

{

public glavnaya\_administrator()

{

InitializeComponent();

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

adm\_uslugi form = new adm\_uslugi();

form.Show();

this.Close();

}

private void button2\_Click(object sender, RoutedEventArgs e)

{

adm\_klient form = new adm\_klient();

form.Show();

this.Close();

}

// Функция "Выйти"

private void button3\_Click(object sender, RoutedEventArgs e)

{

MainWindow form = new MainWindow();

form.Show();

this.Close();

}

}

}

**Код главной формы врача:**

<Window x:Class="Diplom\_Ohotnikov.glavnaya\_vrach"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="Главное окно профиля 'Врач'" Height="450" Width="800">

<Grid>

<Image Margin="3,2,675.333,344.667" Source="logo.png" Stretch="Fill"/>

<Label Content="Профиль 'Врач'" HorizontalAlignment="Left" Margin="304,24,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<Button x:Name="button3" Content="Выйти" HorizontalAlignment="Left" Margin="550,366,0,0" VerticalAlignment="Top" Width="233" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button1" Content="БД 'Услуги'" HorizontalAlignment="Left" Margin="276,111,0,0" VerticalAlignment="Top" Width="253" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="БД 'Приемы'" HorizontalAlignment="Left" Margin="276,190,0,0" VerticalAlignment="Top" Width="253" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button4" Content="БД 'Оказанные услуги'" HorizontalAlignment="Left" Margin="276,270,0,0" VerticalAlignment="Top" Width="253" Height="45" FontSize="22" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

</Grid>

</Window>

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 Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для glavnaya\_vrach.xaml

/// </summary>

public partial class glavnaya\_vrach : Window

{

public glavnaya\_vrach()

{

InitializeComponent();

}

private void button3\_Click(object sender, RoutedEventArgs e)

{

MainWindow form = new MainWindow();

form.Show();

this.Close();

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

vr\_uslugi form = new vr\_uslugi();

form.Show();

this.Close();

}

private void button2\_Click(object sender, RoutedEventArgs e)

{

vr\_priem form = new vr\_priem();

form.Show();

this.Close();

}

private void button4\_Click(object sender, RoutedEventArgs e)

{

vr\_okazannye\_uslugi form = new vr\_okazannye\_uslugi();

form.Show();

this.Close();

}

}

}

**Код формы директора «БД Услуги»:**

<Window x:Class="Diplom\_Ohotnikov.uslugi"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Услуги'" Height="464.891" Width="800">

<Grid>

<Label Content="База данных 'Услуги'" HorizontalAlignment="Left" Margin="265,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="162,286,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="162,321,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="162,358,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Наименование" HorizontalAlignment="Left" Margin="35,281,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Стоимость" HorizontalAlignment="Left" Margin="35,316,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Категория услуг" HorizontalAlignment="Left" Margin="35,349,0,0" VerticalAlignment="Top" Width="154" FontSize="16"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="438,260,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox5" HorizontalAlignment="Left" Height="22" Margin="438,293,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Статус" HorizontalAlignment="Left" Margin="298,255,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Срок выполнения" HorizontalAlignment="Left" Margin="298,284,0,0" VerticalAlignment="Top" Width="154" FontSize="16"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="618,336,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="162,401,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код услуги" HorizontalAlignment="Left" Margin="35,397,0,0" VerticalAlignment="Top" Width="114" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="619,394,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="298,394,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="Удалить" HorizontalAlignment="Left" Margin="455,394,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="23" Margin="438,329,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Расписание" HorizontalAlignment="Left" Margin="299,324,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Код центра" HorizontalAlignment="Left" Margin="298,357,0,0" VerticalAlignment="Top" Width="154" FontSize="16"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="569,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="190,226,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="438,232,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="678,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Image Margin="3,2,675.333,359.333" Source="logo.png" Stretch="Fill"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="618,293,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по стоимости" HorizontalAlignment="Left" Margin="569,263,-24.333,0" VerticalAlignment="Top" Width="250" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="689,293,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<ComboBox x:Name="comboBox1" HorizontalAlignment="Left" Margin="438,364,0,0" VerticalAlignment="Top" Width="120">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="3" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="7" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="8" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="9" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для uslugi.xaml

/// </summary>

public partial class uslugi

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public uslugi()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Usluga", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Usluga");

DataView dview = new DataView(ds.Tables["Diplom\_Usluga"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Usluga";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (textBox5.Text != "") && (textBox7.Text != "") && (comboBox1.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Usluga (naimenovanie,stoimost,kategoria\_uslug,status,srok\_vypolnenya,raspisanie,kod\_centra) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox7.Text + "','" + comboBox1.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox6.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Usluga SET naimenovanie = '" + textBox1.Text + "',stoimost='" + textBox2.Text + "',kategoria\_uslug='" + textBox3.Text + "',status='"

+ textBox4.Text + "',srok\_vypolnenya='" + textBox5.Text + "',raspisanie='" + textBox7.Text + "',kod\_centra='" + comboBox1.Text + "' where kod\_uslugi = '" + textBox6.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код услуги' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

if (textBox6.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_Usluga WHERE kod\_uslugi = '" + textBox6.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox15.Clear();

comboBox1.Items.Clear();

}

else

{

MessageBox.Show("Текстовое поле 'Код услуги' не заполнено", "Ошибка");

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Поиск по наименованию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

uslugi form = new uslugi();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Usluga ORDER BY stoimost ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Usluga ORDER BY stoimost DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы директора «БД Клиенты»:**

<Window x:Class="Diplom\_Ohotnikov.klient"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Клиент'" Height="523.666" Width="809.334">

<Grid>

<Label Content="База данных 'Клиент'" HorizontalAlignment="Left" Margin="265,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="757"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="162,271,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="162,306,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="162,343,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Фамилия" HorizontalAlignment="Left" Margin="35,266,0,0" VerticalAlignment="Top" Width="114" FontSize="16"/>

<Label Content="Имя" HorizontalAlignment="Left" Margin="35,301,0,0" VerticalAlignment="Top" Width="114" FontSize="16"/>

<Label Content="Отчество" HorizontalAlignment="Left" Margin="35,334,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="162,377,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox5" HorizontalAlignment="Left" Height="22" Margin="428,271,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Дата оформления" HorizontalAlignment="Left" Margin="22,370,0,0" VerticalAlignment="Top" Width="149" FontSize="16"/>

<Label Content="Почта" HorizontalAlignment="Left" Margin="288,265,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="624,415,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<TextBox x:Name="textBox11" HorizontalAlignment="Left" Height="23" Margin="162,418,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код клиента" HorizontalAlignment="Left" Margin="35,412,0,0" VerticalAlignment="Top" Width="114" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="624,450,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="298,413,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="Удалить" HorizontalAlignment="Left" Margin="455,413,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="428,340,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="22" Margin="428,308,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Номер телефона" HorizontalAlignment="Left" Margin="289,302,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Расчетный счет" HorizontalAlignment="Left" Margin="288,335,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<TextBox x:Name="textBox8" HorizontalAlignment="Left" Height="23" Margin="428,373,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox9" HorizontalAlignment="Left" Height="22" Margin="675,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="115"/>

<Label Content="Название банка" HorizontalAlignment="Left" Margin="288,366,0,0" VerticalAlignment="Top" Width="135" FontSize="16"/>

<Label Content="Код сотрудника" HorizontalAlignment="Left" Margin="551,261,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<TextBox x:Name="textBox10" HorizontalAlignment="Left" Height="23" Margin="675,295,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="115"/>

<Label Content="Код центра" HorizontalAlignment="Left" Margin="552,290,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="569,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по фамилии" HorizontalAlignment="Left" Margin="206,226,0,0" VerticalAlignment="Top" Width="226" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="426,233,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="678,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Label Content="Код пола" HorizontalAlignment="Left" Margin="554,317,0,0" VerticalAlignment="Top" Width="84" FontSize="16"/>

<ComboBox x:Name="comboBox2" HorizontalAlignment="Left" Margin="675,321,0,0" VerticalAlignment="Top" Width="114">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="2" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

<Label Content="1 - мужской; 2 - женский" HorizontalAlignment="Left" Margin="609,340,0,0" VerticalAlignment="Top" Width="162" FontSize="12" FontWeight="Bold"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="621,381,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по дате" HorizontalAlignment="Left" Margin="591,351,-15.667,0" VerticalAlignment="Top" Width="218" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="692,381,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<Image Margin="3,2,683.333,416.667" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для klient.xaml

/// </summary>

public partial class klient

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public klient()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Klient", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Klient");

DataView dview = new DataView(ds.Tables["Diplom\_Klient"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Klient";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (textBox5.Text != "") && (textBox6.Text != "") && (textBox7.Text != "") && (textBox8.Text != "") && (textBox9.Text != "") && (textBox10.Text != "") && (comboBox2.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Klient (familiya,imya,otchestvo,data\_oformlenya,elektronnaya\_pochta,nomer\_telefona,rachetnyu\_chet,nazvanie\_banka,kod\_sotrudnika,kod\_centra,kod\_pola) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox7.Text + "','" + textBox8.Text + "','" + textBox9.Text + "','" + textBox10.Text + "','" + comboBox2.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox11.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Klient SET familiya = '" + textBox1.Text + "',imya='" + textBox2.Text + "',otchestvo='" + textBox3.Text + "',data\_oformlenya='"

+ textBox4.Text + "',elektronnaya\_pochta='" + textBox5.Text + "',nomer\_telefona='" + textBox6.Text + "',rachetnyu\_chet='" + textBox7.Text + "',nazvanie\_banka='" + textBox8.Text + "',kod\_sotrudnika='" + textBox9.Text + "',kod\_centra='" + textBox10.Text + "',kod\_pola='" + comboBox2.Text + "' where kod\_klienta = '" + textBox11.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код клиента' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

if (textBox11.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_Klient WHERE kod\_klienta = '" + textBox11.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код клиента' не заполнено", "Ошибка");

}

}

// Вывод в Excel

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

// Функция "Поиск по фамилии"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"familiya LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по фамилии' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

klient form = new klient();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Klient ORDER BY data\_oformlenya ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Klient ORDER BY data\_oformlenya DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы директора «БД Сотрудники»:**

<Window x:Class="Diplom\_Ohotnikov.sotrudniki"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Сотрудники'" Height="511.819" Width="980.479">

<Grid Margin="0,0,4.333,-0.333" Height="479" VerticalAlignment="Top">

<Label Content="База данных 'Сотрудники'" HorizontalAlignment="Left" Margin="328,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="917"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="162,273,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="162,308,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="162,345,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Фамилия" HorizontalAlignment="Left" Margin="35,268,0,0" VerticalAlignment="Top" Width="113" FontSize="16"/>

<Label Content="Имя" HorizontalAlignment="Left" Margin="35,303,0,0" VerticalAlignment="Top" Width="113" FontSize="16"/>

<Label Content="Отчество" HorizontalAlignment="Left" Margin="35,336,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="426,268,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="День рождения" HorizontalAlignment="Left" Margin="286,263,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<Label Content="Должность" HorizontalAlignment="Left" Margin="286,296,0,0" VerticalAlignment="Top" Width="125" FontSize="16"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="618,371,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<TextBox x:Name="textBox14" HorizontalAlignment="Left" Height="23" Margin="162,413,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код сотрудника" HorizontalAlignment="Left" Margin="28,408,0,0" VerticalAlignment="Top" Width="133" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button6" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="12,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="618,441,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="298,407,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="Удалить" HorizontalAlignment="Left" Margin="455,407,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="426,341,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="22" Margin="426,378,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Специализация" HorizontalAlignment="Left" Margin="287,336,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Квалификация" HorizontalAlignment="Left" Margin="286,369,0,0" VerticalAlignment="Top" Width="135" FontSize="16"/>

<TextBox x:Name="textBox8" HorizontalAlignment="Left" Height="23" Margin="655,268,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="115"/>

<TextBox x:Name="textBox9" HorizontalAlignment="Left" Height="22" Margin="655,305,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="115"/>

<Label Content="Почта" HorizontalAlignment="Left" Margin="544,263,0,0" VerticalAlignment="Top" Width="104" FontSize="16"/>

<Label Content="Телефон" HorizontalAlignment="Left" Margin="544,296,0,0" VerticalAlignment="Top" Width="104" FontSize="16"/>

<TextBox x:Name="textBox10" HorizontalAlignment="Left" Height="23" Margin="655,341,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="115"/>

<Label Content="Адрес" HorizontalAlignment="Left" Margin="545,336,0,0" VerticalAlignment="Top" Width="103" FontSize="16"/>

<TextBox x:Name="textBox11" HorizontalAlignment="Left" Height="23" Margin="860,235,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100"/>

<TextBox x:Name="textBox12" HorizontalAlignment="Left" Height="22" Margin="860,272,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="100"/>

<Label Content="Логин" HorizontalAlignment="Left" Margin="769,230,0,0" VerticalAlignment="Top" Width="86" FontSize="16"/>

<Label Content="Пароль" HorizontalAlignment="Left" Margin="769,263,0,0" VerticalAlignment="Top" Width="86" FontSize="16"/>

<Label Content="Код центра" HorizontalAlignment="Left" Margin="770,297,0,0" VerticalAlignment="Top" Width="102" FontSize="16"/>

<Button x:Name="button5" Content="БД 'Статус'" HorizontalAlignment="Left" Margin="618,408,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<ComboBox x:Name="comboBox1" HorizontalAlignment="Left" Margin="861,305,0,0" VerticalAlignment="Top" Width="98">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="3" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="7" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="8" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="9" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

<Button x:Name="button7" Content="Перезагрузка" HorizontalAlignment="Left" Margin="455,440,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Button x:Name="button8" Content="Поиск" HorizontalAlignment="Left" Margin="523,230,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Поиск по фамилии" HorizontalAlignment="Left" Margin="180,227,0,0" VerticalAlignment="Top" Width="187" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="372,235,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button9" Content="Очистить" HorizontalAlignment="Left" Margin="632,230,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<Label Content="Код пола" HorizontalAlignment="Left" Margin="771,329,0,0" VerticalAlignment="Top" Width="84" FontSize="16"/>

<ComboBox x:Name="comboBox2" HorizontalAlignment="Left" Margin="861,333,0,0" VerticalAlignment="Top" Width="99">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="2" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

<ComboBox x:Name="comboBox3" HorizontalAlignment="Left" Margin="426,303,0,0" VerticalAlignment="Top" Width="120">

<ComboBoxItem Content="Директор" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="Администратор" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="Врач" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

<Label Content="1 - мужской; 2 - женский" HorizontalAlignment="Left" Margin="792,348,-34,0" VerticalAlignment="Top" Width="212" FontSize="12" FontWeight="Bold"/>

<Button x:Name="button10" Content="↑" HorizontalAlignment="Left" Margin="816,412,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button10\_Click"/>

<Label Content="Сортировка по дню рождения" HorizontalAlignment="Left" Margin="761,386,0,0" VerticalAlignment="Top" Width="208" FontSize="13" FontWeight="Bold"/>

<Button x:Name="button11" Content="↓" HorizontalAlignment="Left" Margin="887,412,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button11\_Click"/>

<Image Margin="3,2,851.999,401.666" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для sotrudniki.xaml

/// </summary>

public partial class sotrudniki

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public sotrudniki()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Sotrudnik", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Sotrudnik");

DataView dview = new DataView(ds.Tables["Diplom\_Sotrudnik"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Sotrudnik";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (comboBox3.Text != "") && (textBox6.Text != "") && (textBox7.Text != "") && (textBox8.Text != "") && (textBox9.Text != "") && (textBox10.Text != "") && (textBox11.Text != "") && (textBox12.Text != "") && (comboBox1.Text != "") && (comboBox2.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Sotrudnik (familiya,imya,otchestvo,den\_rozhdenya,dolzhnost,specializacya,kfalifikacya,elektronnaya\_pochta,nomer\_telefona,adres,login,parol,kod\_centra, kod\_pola) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + comboBox3.Text + "','" + textBox6.Text + "','" + textBox7.Text + "','" + textBox8.Text + "','" + textBox9.Text + "','" + textBox10.Text + "','" + textBox11.Text + "','" + textBox12.Text + "','" + comboBox1.Text + "','" + comboBox2.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox14.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Sotrudnik SET familiya = '" + textBox1.Text + "',imya='" + textBox2.Text + "',otchestvo='" + textBox3.Text + "',den\_rozhdenya='"

+ textBox4.Text + "',dolzhnost='" + comboBox3.Text + "',specializacya='" + textBox6.Text + "',kfalifikacya='" + textBox7.Text + "',elektronnaya\_pochta='" + textBox8.Text + "',nomer\_telefona='" + textBox9.Text + "',adres='" + textBox10.Text + "',login='" + textBox11.Text + "',parol='" + textBox12.Text + "',kod\_centra='" + comboBox1.Text + "',kod\_pola='" + comboBox2.Text + "' where kod\_sotrudnika = '" + textBox14.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код сотрудника' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

if (textBox14.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_Sotrudnik WHERE kod\_sotrudnika = '" + textBox14.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox9.Clear();

textBox10.Clear();

textBox11.Clear();

textBox12.Clear();

textBox14.Clear();

comboBox1.Items.Clear();

comboBox2.Items.Clear();

comboBox3.Items.Clear();

}

else

{

MessageBox.Show("Текстовое поле 'Код сотрудника' не заполнено", "Ошибка");

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

// Функция "Переход на форму 'БД Статус'"

private void button5\_Click(object sender, RoutedEventArgs e)

{

status form = new status();

form.Show();

}

// Вывод в Excel

private void button6\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Перезагрузка формы"

private void button7\_Click(object sender, RoutedEventArgs e)

{

sotrudniki form = new sotrudniki();

form.Show();

this.Close();

}

// Функция "Поиск по фамилии"

private void button8\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"familiya LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по фамилии' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button9\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

sotrudniki form = new sotrudniki();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button10\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Sotrudnik ORDER BY den\_rozhdenya ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button11\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Sotrudnik ORDER BY den\_rozhdenya DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы директора «БД Статус»:**

<Window x:Class="Diplom\_Ohotnikov.status"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Статус'" Height="465.174" Width="800">

<Grid>

<Label Content="База данных 'Статус'" HorizontalAlignment="Left" Margin="269,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,79,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="173,289,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="173,245,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код статуса" HorizontalAlignment="Left" Margin="25,284,0,0" VerticalAlignment="Top" Width="133" FontSize="16" FontWeight="Bold"/>

<Label Content="Статус" HorizontalAlignment="Left" Margin="25,240,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Статус сотрудника" HorizontalAlignment="Left" Margin="493,295,0,0" VerticalAlignment="Top" Width="154" FontSize="16"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="311,232,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="614,393,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="311,274,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click" RenderTransformOrigin="1.51,1.052"/>

<Button x:Name="button4" Content="Удалить" HorizontalAlignment="Left" Margin="311,311,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Label Content="Изменить статус сотрудника" HorizontalAlignment="Left" Margin="489,227,0,0" VerticalAlignment="Top" FontSize="20" FontWeight="Bold"/>

<ComboBox x:Name="comboBox1" HorizontalAlignment="Left" Margin="647,301,0,0" VerticalAlignment="Top" Width="120">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="117.333333333333"/>

<ComboBoxItem Content="2" HorizontalAlignment="Left" Width="117.333333333333"/>

</ComboBox>

<Label Content="1 - не уволен; 2 - уволен" HorizontalAlignment="Left" Margin="537,360,0,0" VerticalAlignment="Top" Width="209" FontSize="16" FontWeight="Bold" />

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="23" Margin="647,265,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код сотрудника" HorizontalAlignment="Left" Margin="495,261,0,0" VerticalAlignment="Top" Width="137" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button5" Content="Изменить статус" HorizontalAlignment="Left" Margin="565,331,0,0" VerticalAlignment="Top" Width="155" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click" RenderTransformOrigin="1.51,1.052"/>

<Image Margin="3,2,675.333,360.333" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для status.xaml

/// </summary>

public partial class status : Window

{

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public status()

{

InitializeComponent();

GetList();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Status", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Status");

DataView dview = new DataView(ds.Tables["Diplom\_Status"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

void GetList2()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Sotrudnik", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Sotrudnik");

con.Close();

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if (textBox2.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Status (status) values (" + textBox2.Text + ")";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox1.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Status SET status = '" + textBox2.Text + "' where kod\_statusa = '" + textBox1.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код статуса' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

if (textBox1.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_Status WHERE kod\_statusa = '" + textBox1.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

textBox1.Clear();

textBox2.Clear();

}

else

{

MessageBox.Show("Текстовое поле 'Код статуса' не заполнено", "Ошибка");

}

}

// Функция "Изменить статус сотрудника"

private void button5\_Click(object sender, RoutedEventArgs e)

{

if (textBox3.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Sotrudnik SET kod\_statusa = '" + comboBox1.Text + "' where kod\_sotrudnika = '" + textBox3.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList2();

MessageBox.Show(string.Format("Успешное изменение данных в таблице 'Сотрудник'"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код сотрудника' не заполнено", "Ошибка");

}

}

}

}

**Код формы директора «БД Оказанные услуги»:**

<Window x:Class="Diplom\_Ohotnikov.okazannye\_uslugi"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Оказанные услуги'" Height="442.014" Width="800">

<Grid Margin="1,0,-1.333,0">

<Label Content="База данных 'Оказанные услуги'" HorizontalAlignment="Left" Margin="197,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="621,354,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,231,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Label Content="Общая стоимость оказанных услуг:" HorizontalAlignment="Left" Margin="35,272,0,0" VerticalAlignment="Top" FontSize="20" FontWeight="Bold"/>

<Label Content="Общее количество оказанных услуг:" HorizontalAlignment="Left" Margin="35,314,0,0" VerticalAlignment="Top" FontSize="20" FontWeight="Bold"/>

<Label x:Name="label1" Content="" HorizontalAlignment="Left" Margin="415,272,0,0" VerticalAlignment="Top" Width="176" FontSize="20" FontWeight="Bold"/>

<Label x:Name="label2" Content="" HorizontalAlignment="Left" Margin="415,312,0,0" VerticalAlignment="Top" Width="176" FontSize="20" FontWeight="Bold"/>

<Button x:Name="button3" Content="Рассчитать 1" HorizontalAlignment="Left" Margin="621,271,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="Рассчитать 2" HorizontalAlignment="Left" Margin="621,310,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="563,229,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="184,227,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="432,233,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="672,229,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="358,354,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по дате оказания" HorizontalAlignment="Left" Margin="36,348,0,0" VerticalAlignment="Top" Width="302" FontSize="20" FontWeight="Bold"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="429,354,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<Image Margin="3,2,675.666,336.666" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для okazannye\_uslugi.xaml

/// </summary>

public partial class okazannye\_uslugi

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public okazannye\_uslugi()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_OkazannyeUslugi", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_OkazannyeUslugi");

DataView dview = new DataView(ds.Tables["Diplom\_OkazannyeUslugi"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_OkazannyeUslugi";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

// Функция "Вывод в Excel"

private void button2\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Рассчитать стоимость оказанных услуг"

private void button3\_Click(object sender, RoutedEventArgs e)

{

string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

string query = "SELECT SUM(stoimost) FROM Diplom\_OkazannyeUslugi";

GetList();

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand(query, connection))

{

connection.Open();

int total = (int)command.ExecuteScalar();

label1.Content = $"{total} рублей";

}

}

}

// Функция "Рассчитать общее количество оказанных услуг"

private void button4\_Click(object sender, RoutedEventArgs e)

{

string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

string query = "SELECT COUNT(kod\_okazannoy\_uslugi) FROM Diplom\_OkazannyeUslugi";

GetList();

using (SqlConnection connection = new SqlConnection(connectionString))

{

using (SqlCommand command = new SqlCommand(query, connection))

{

connection.Open();

int total = (int)command.ExecuteScalar();

label2.Content = $"{total}";

}

}

}

// Функция "Поиск по наименованию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

okazannye\_uslugi form = new okazannye\_uslugi();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_OkazannyeUslugi ORDER BY data\_okazania ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_OkazannyeUslugi ORDER BY data\_okazania DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы директора «БД Приемы»:**

<Window x:Class="Diplom\_Ohotnikov.priem"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Приемы'" Height="376.634" Width="800">

<Grid>

<Label Content="База данных 'Приемы'" HorizontalAlignment="Left" Margin="264,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="636,300,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,248,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button3" Content="Поиск" HorizontalAlignment="Left" Margin="577,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="187,244,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="440,252,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button4" Content="Очистить" HorizontalAlignment="Left" Margin="686,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Image Margin="3,2,675.334,271.667" Source="logo.png" Stretch="Fill"/>

<Button x:Name="button5" Content="↑" HorizontalAlignment="Left" Margin="357,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Label Content="Сортировка по дате оказания" HorizontalAlignment="Left" Margin="35,289,0,0" VerticalAlignment="Top" Width="302" FontSize="20" FontWeight="Bold"/>

<Button x:Name="button6" Content="↓" HorizontalAlignment="Left" Margin="428,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для priem.xaml

/// </summary>

public partial class priem

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public priem()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Priem", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Priem");

DataView dview = new DataView(ds.Tables["Diplom\_Priem"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Priem";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Поиск по наименованию"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

priem form = new priem();

form.Show();

this.Close();

}

// Функция "Вывод в Excel"

private void button2\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button5\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Priem ORDER BY data\_okazania ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Priem ORDER BY data\_okazania DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы директора «БД Медцентр»:**

<Window x:Class="Diplom\_Ohotnikov.medcenter"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Медицинский центр'" Height="444.189" Width="800">

<Grid>

<Label Content="База данных 'Медицинский центр'" HorizontalAlignment="Left" Margin="174,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="162,269,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="162,304,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="162,341,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Наименование" HorizontalAlignment="Left" Margin="35,264,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<Label Content="ИНН" HorizontalAlignment="Left" Margin="35,299,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<Label Content="Адрес" HorizontalAlignment="Left" Margin="35,332,0,0" VerticalAlignment="Top" Width="114" FontSize="16"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="438,268,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox5" HorizontalAlignment="Left" Height="22" Margin="438,305,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Почта" HorizontalAlignment="Left" Margin="311,263,0,0" VerticalAlignment="Top" Width="118" FontSize="16"/>

<Label Content="Телефон" HorizontalAlignment="Left" Margin="311,296,0,0" VerticalAlignment="Top" Width="118" FontSize="16"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="618,369,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="618,276,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="298,369,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="Удалить" HorizontalAlignment="Left" Margin="455,369,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="162,375,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код центра" HorizontalAlignment="Left" Margin="35,371,0,0" VerticalAlignment="Top" Width="114" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="563,229,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="184,227,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="438,233,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="672,229,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="629,332,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по названию" HorizontalAlignment="Left" Margin="580,302,-35.333,0" VerticalAlignment="Top" Width="250" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="700,332,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<Image Margin="3,2,675.333,338.666" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для medcenter.xaml

/// </summary>

public partial class medcenter

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public medcenter()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_MedCenter", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_MedCenter");

DataView dview = new DataView(ds.Tables["Diplom\_MedCenter"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_MedCenter";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (textBox5.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_MedCenter (naimenovanie,INN,adres,elektronnaya\_pochta,telefon) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox6.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_MedCenter SET naimenovanie = '" + textBox1.Text + "',INN='" + textBox2.Text + "',adres='" + textBox3.Text + "',elektronnaya\_pochta='"

+ textBox4.Text + "',telefon='" + textBox5.Text + "' where kod\_centra = '" + textBox6.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код центра' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

if (textBox6.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_MedCenter WHERE kod\_centra = '" + textBox6.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код центра' не заполнено", "Ошибка");

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_director form = new glavnaya\_director();

form.Show();

this.Close();

}

// Функция "Вывод в Excel"

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

// Функция "Поиск по наименованию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

medcenter form = new medcenter();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_MedCenter ORDER BY naimenovanie ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_MedCenter ORDER BY naimenovanie DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы администратора «БД Услуги»:**

<Window x:Class="Diplom\_Ohotnikov.adm\_uslugi"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Услуги'" Height="372.202" Width="800">

<Grid>

<Label Content="База данных 'Услуги'" HorizontalAlignment="Left" Margin="264,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="636,300,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,248,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button3" Content="Поиск" HorizontalAlignment="Left" Margin="577,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="206,244,0,0" VerticalAlignment="Top" Width="234" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="440,252,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button4" Content="Очистить" HorizontalAlignment="Left" Margin="686,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Button x:Name="button5" Content="↑" HorizontalAlignment="Left" Margin="332,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Label Content="Сортировка по стоимости" HorizontalAlignment="Left" Margin="35,289,0,0" VerticalAlignment="Top" Width="284" FontSize="20" FontWeight="Bold"/>

<Button x:Name="button6" Content="↓" HorizontalAlignment="Left" Margin="403,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Image Margin="3,2,674.332,265.333" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для adm\_uslugi.xaml

/// </summary>

public partial class adm\_uslugi

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public adm\_uslugi()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Usluga", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Usluga");

DataView dview = new DataView(ds.Tables["Diplom\_Usluga"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Usluga";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Поиск по наименованию"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

adm\_uslugi form = new adm\_uslugi();

form.Show();

this.Close();

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_administrator form = new glavnaya\_administrator();

form.Show();

this.Close();

}

// Функция "Вывод в Excel"

private void button2\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Сортировка по возрастанию"

private void button5\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Usluga ORDER BY stoimost ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Usluga ORDER BY stoimost DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

}

}

**Код формы администратора «БД Клиенты»:**

<Window x:Class="Diplom\_Ohotnikov.adm\_klient"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Клиент'" Height="520.275" Width="808.667">

<Grid>

<Label Content="База данных 'Клиент'" HorizontalAlignment="Left" Margin="265,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="162,271,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="162,306,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="162,343,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Фамилия" HorizontalAlignment="Left" Margin="35,266,0,0" VerticalAlignment="Top" Width="114" FontSize="16"/>

<Label Content="Имя" HorizontalAlignment="Left" Margin="35,301,0,0" VerticalAlignment="Top" Width="114" FontSize="16"/>

<Label Content="Отчество" HorizontalAlignment="Left" Margin="35,334,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="162,377,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox5" HorizontalAlignment="Left" Height="22" Margin="428,271,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Дата оформления" HorizontalAlignment="Left" Margin="22,370,0,0" VerticalAlignment="Top" Width="149" FontSize="16"/>

<Label Content="Почта" HorizontalAlignment="Left" Margin="288,265,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="457,413,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<TextBox x:Name="textBox11" HorizontalAlignment="Left" Height="23" Margin="162,418,0,-20.333" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код клиента" HorizontalAlignment="Left" Margin="35,412,0,-22.667" VerticalAlignment="Top" Width="114" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="624,450,0,-60.333" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="298,413,0,-23.333" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="БД 'Приемы'" HorizontalAlignment="Left" Margin="624,413,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="428,340,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="22" Margin="428,308,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Номер телефона" HorizontalAlignment="Left" Margin="289,302,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Label Content="Расчетный счет" HorizontalAlignment="Left" Margin="288,335,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<TextBox x:Name="textBox8" HorizontalAlignment="Left" Height="23" Margin="428,373,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox9" HorizontalAlignment="Left" Height="22" Margin="675,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="115"/>

<Label Content="Название банка" HorizontalAlignment="Left" Margin="288,366,0,0" VerticalAlignment="Top" Width="135" FontSize="16"/>

<Label Content="Код сотрудника" HorizontalAlignment="Left" Margin="551,261,0,0" VerticalAlignment="Top" Width="134" FontSize="16"/>

<Label Content="Код центра" HorizontalAlignment="Left" Margin="552,290,0,0" VerticalAlignment="Top" Width="133" FontSize="16"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="569,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по фамилии" HorizontalAlignment="Left" Margin="206,226,0,0" VerticalAlignment="Top" Width="226" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="426,233,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="678,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Label Content="Код пола" HorizontalAlignment="Left" Margin="554,317,0,0" VerticalAlignment="Top" Width="84" FontSize="16"/>

<ComboBox x:Name="comboBox2" HorizontalAlignment="Left" Margin="675,321,0,0" VerticalAlignment="Top" Width="115">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="2" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

<ComboBox x:Name="comboBox1" HorizontalAlignment="Left" Margin="675,296,0,0" VerticalAlignment="Top" Width="115">

<ComboBoxItem Content="1" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="3" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="7" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="8" HorizontalAlignment="Left" Width="95.3333333333333"/>

<ComboBoxItem Content="9" HorizontalAlignment="Left" Width="95.3333333333333"/>

</ComboBox>

<Label Content="1 - мужской; 2 - женский" HorizontalAlignment="Left" Margin="609,340,0,0" VerticalAlignment="Top" Width="162" FontSize="12" FontWeight="Bold"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="621,381,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по дате" HorizontalAlignment="Left" Margin="591,351,-15.667,0" VerticalAlignment="Top" Width="218" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="692,381,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<Button x:Name="button10" Content="Перезагрузка" HorizontalAlignment="Left" Margin="457,450,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button10\_Click"/>

<Image Margin="3,2,683.999,414.666" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для adm\_klient.xaml

/// </summary>

public partial class adm\_klient

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public adm\_klient()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Klient", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Klient");

DataView dview = new DataView(ds.Tables["Diplom\_Klient"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Klient";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox11.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Klient SET familiya = '" + textBox1.Text + "',imya='" + textBox2.Text + "',otchestvo='" + textBox3.Text + "',data\_oformlenya='"

+ textBox4.Text + "',elektronnaya\_pochta='" + textBox5.Text + "',nomer\_telefona='" + textBox6.Text + "',rachetnyu\_chet='" + textBox7.Text + "',nazvanie\_banka='" + textBox8.Text + "',kod\_sotrudnika='" + textBox9.Text + "',kod\_centra='" + comboBox1.Text + "',kod\_pola='" + comboBox2.Text + "' where kod\_klienta = '" + textBox11.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код клиента' не заполнено", "Ошибка");

}

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (textBox5.Text != "") && (textBox6.Text != "") && (textBox7.Text != "") && (textBox8.Text != "") && (textBox9.Text != "") && (comboBox1.Text != "") && (comboBox2.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Klient (familiya,imya,otchestvo,data\_oformlenya,elektronnaya\_pochta,nomer\_telefona,rachetnyu\_chet,nazvanie\_banka,kod\_sotrudnika,kod\_centra,kod\_pola) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox7.Text + "','" + textBox8.Text + "','" + textBox9.Text + "','" + comboBox1.Text + "','" + comboBox2.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Вывод в Excel"

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Поиск по фамилии"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"familiya LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по фамилии' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

adm\_klient form = new adm\_klient();

form.Show();

this.Close();

}

// Функция "Переход на форму 'БД Приемы'"

private void button4\_Click(object sender, RoutedEventArgs e)

{

adm\_priem form = new adm\_priem();

form.Show();

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_administrator form = new glavnaya\_administrator();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Klient ORDER BY data\_oformlenya ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Klient ORDER BY data\_oformlenya DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Перезагрузка"

private void button10\_Click(object sender, RoutedEventArgs e)

{

adm\_klient form = new adm\_klient();

form.Show();

this.Close();

}

}

}

**Код формы администратора «БД Приемы»:**

<Window x:Class="Diplom\_Ohotnikov.adm\_priem"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Приемы'" Height="474.523" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition Height="0\*"/>

<RowDefinition Height="0\*"/>

</Grid.RowDefinitions>

<Label Content="База данных 'Приемы'" HorizontalAlignment="Left" Margin="265,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold" Height="42" Width="279"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="175,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="175,305,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="175,342,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Наименование" HorizontalAlignment="Left" Margin="48,265,0,0" VerticalAlignment="Top" Width="122" FontSize="16" Height="31"/>

<Label Content="Стоимость" HorizontalAlignment="Left" Margin="48,300,0,0" VerticalAlignment="Top" Width="122" FontSize="16" Height="31"/>

<Label Content="Время приема" HorizontalAlignment="Left" Margin="48,333,0,0" VerticalAlignment="Top" Width="122" FontSize="16" Height="31"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="438,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox5" HorizontalAlignment="Left" Height="22" Margin="438,303,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Дата оказания" HorizontalAlignment="Left" Margin="298,265,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Label Content="Симптомы" HorizontalAlignment="Left" Margin="298,294,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Button x:Name="button2" Content="Записать" HorizontalAlignment="Left" Margin="486,399,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="637,399,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="438,335,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Диагноз" HorizontalAlignment="Left" Margin="299,330,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Label Content="Примечание" HorizontalAlignment="Left" Margin="298,363,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="569,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="190,226,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold" Height="34"/>

<TextBox x:Name="textBox10" HorizontalAlignment="Left" Height="23" Margin="438,232,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="678,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="618,351,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по дате" HorizontalAlignment="Left" Margin="594,321,0,0" VerticalAlignment="Top" Width="189" FontSize="16" FontWeight="Bold" Height="31"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="689,351,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="23" Margin="438,364,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox8" HorizontalAlignment="Left" Height="23" Margin="689,269,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="94"/>

<Label Content="Код сотрудника" HorizontalAlignment="Left" Margin="560,264,0,0" VerticalAlignment="Top" Width="137" FontSize="16" Height="31" FontWeight="Bold"/>

<Label Content="Код клиента" HorizontalAlignment="Left" Margin="560,295,0,0" VerticalAlignment="Top" Width="113" FontSize="16" Height="31" FontWeight="Bold"/>

<TextBox x:Name="textBox9" HorizontalAlignment="Left" Height="23" Margin="689,298,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="94" RenderTransformOrigin="1.935,-2.738"/>

<Image Margin="3,2,673.999,367.999" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для adm\_priem.xaml

/// </summary>

public partial class adm\_priem

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlCommand cmd;

SqlDataAdapter da;

DataSet ds;

public adm\_priem()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Priem", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Priem");

DataView dview = new DataView(ds.Tables["Diplom\_Priem"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Priem";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Записать на прием"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (textBox5.Text != "") && (textBox6.Text != "") && (textBox7.Text != "") && (textBox8.Text != "") && (textBox9.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Priem (naimenovanie,stoimost,vremya\_priema,data\_okazania,simptomy,diagnoz,primechanie,kod\_sotrudnika,kod\_klienta) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox7.Text + "','" + textBox8.Text + "','" + textBox9.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Вывод в Excel"

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Поиск по наименованию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox10.Text != "")

{

string filter = textBox10.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox10.Clear();

adm\_priem form = new adm\_priem();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Priem ORDER BY data\_okazania ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Priem ORDER BY data\_okazania DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

**Код формы врача «БД Услуги»:**

<Window x:Class="Diplom\_Ohotnikov.vr\_uslugi"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Услуги'" Height="382.446" Width="800">

<Grid>

<Label Content="База данных 'Услуги'" HorizontalAlignment="Left" Margin="271,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="636,300,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,248,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button3" Content="Поиск" HorizontalAlignment="Left" Margin="577,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="187,244,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="440,252,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button4" Content="Очистить" HorizontalAlignment="Left" Margin="686,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Image Margin="3,2,673.999,275.999" Source="logo.png" Stretch="Fill"/>

<Button x:Name="button5" Content="↑" HorizontalAlignment="Left" Margin="332,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Label Content="Сортировка по стоимости" HorizontalAlignment="Left" Margin="35,289,0,0" VerticalAlignment="Top" Width="284" FontSize="20" FontWeight="Bold"/>

<Button x:Name="button6" Content="↓" HorizontalAlignment="Left" Margin="403,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для vr\_uslugi.xaml

/// </summary>

public partial class vr\_uslugi

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public vr\_uslugi()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Usluga", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Usluga");

DataView dview = new DataView(ds.Tables["Diplom\_Usluga"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Usluga";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Вывод в Excel"

private void button2\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Поиск по наименованию"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

vr\_uslugi form = new vr\_uslugi();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button5\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Usluga ORDER BY stoimost ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Usluga ORDER BY stoimost DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_vrach form = new glavnaya\_vrach();

form.Show();

this.Close();

}

}

}

**Код формы врача «БД Приемы»:**

<Window x:Class="Diplom\_Ohotnikov.vr\_priem"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Прием'" Height="505.932" Width="804.358">

<Grid>

<Label Content="База данных 'Приемы'" HorizontalAlignment="Left" Margin="265,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold" Height="42" Width="279"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="175,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="175,305,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="175,342,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Наименование" HorizontalAlignment="Left" Margin="48,265,0,0" VerticalAlignment="Top" Width="122" FontSize="16" Height="31"/>

<Label Content="Стоимость" HorizontalAlignment="Left" Margin="48,300,0,0" VerticalAlignment="Top" Width="122" FontSize="16" Height="31"/>

<Label Content="Время приема" HorizontalAlignment="Left" Margin="48,333,0,0" VerticalAlignment="Top" Width="122" FontSize="16" Height="31"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="438,270,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox5" HorizontalAlignment="Left" Height="22" Margin="438,303,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Дата оказания" HorizontalAlignment="Left" Margin="298,265,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Label Content="Симптомы" HorizontalAlignment="Left" Margin="298,294,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="474,399,0,-9.333" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="639,435,0,0" VerticalAlignment="Top" Width="144" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="438,335,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Диагноз" HorizontalAlignment="Left" Margin="299,330,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Label Content="Примечание" HorizontalAlignment="Left" Margin="298,363,0,0" VerticalAlignment="Top" Width="133" FontSize="16" Height="31"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="569,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="190,226,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold" Height="34"/>

<TextBox x:Name="textBox10" HorizontalAlignment="Left" Height="23" Margin="438,232,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="678,228,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="618,351,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по дате" HorizontalAlignment="Left" Margin="594,321,0,0" VerticalAlignment="Top" Width="189" FontSize="16" FontWeight="Bold" Height="31"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="689,351,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<TextBox x:Name="textBox7" HorizontalAlignment="Left" Height="23" Margin="438,364,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox8" HorizontalAlignment="Left" Height="23" Margin="689,269,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="94"/>

<Label Content="Код сотрудника" HorizontalAlignment="Left" Margin="560,264,0,0" VerticalAlignment="Top" Width="137" FontSize="16" Height="31" FontWeight="Bold"/>

<Label Content="Код клиента" HorizontalAlignment="Left" Margin="560,295,0,0" VerticalAlignment="Top" Width="113" FontSize="16" Height="31" FontWeight="Bold"/>

<TextBox x:Name="textBox9" HorizontalAlignment="Left" Height="23" Margin="689,298,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="94" RenderTransformOrigin="1.935,-2.738"/>

<TextBox x:Name="textBox14" HorizontalAlignment="Left" Height="23" Margin="175,400,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код приема" HorizontalAlignment="Left" Margin="48,394,0,0" VerticalAlignment="Top" Width="122" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="313,399,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button10" Content="Удалить" HorizontalAlignment="Left" Margin="313,435,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button10\_Click"/>

<Button x:Name="button11" Content="Перезагрузка" HorizontalAlignment="Left" Margin="475,435,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button11\_Click"/>

<Button x:Name="button12" Content="БД 'Клиент'" HorizontalAlignment="Left" Margin="637,400,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button12\_Click"/>

<Image Margin="3,2,678.665,399.333" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для vr\_priem.xaml

/// </summary>

public partial class vr\_priem

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlCommand cmd;

SqlDataAdapter da;

DataSet ds;

public vr\_priem()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Priem", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Priem");

DataView dview = new DataView(ds.Tables["Diplom\_Priem"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Priem";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != "") && (textBox5.Text != "") && (textBox6.Text != "") && (textBox7.Text != "") && (textBox8.Text != "") && (textBox9.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_Priem (naimenovanie,stoimost,vremya\_priema,data\_okazania,simptomy,diagnoz,primechanie,kod\_sotrudnika,kod\_klienta) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + textBox7.Text + "','" + textBox8.Text + "','" + textBox9.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox14.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_Priem SET naimenovanie = '" + textBox1.Text + "',stoimost='" + textBox2.Text + "',vremya\_priema='" + textBox3.Text + "',data\_okazania='"

+ textBox4.Text + "',simptomy='" + textBox5.Text + "',diagnoz='" + textBox6.Text + "',primechanie='" + textBox7.Text + "',kod\_sotrudnika='" + textBox8.Text + "',kod\_klienta='" + textBox9.Text + "' where kod\_priema = '" + textBox14.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код приема' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button10\_Click(object sender, RoutedEventArgs e)

{

if (textBox14.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_Priem WHERE kod\_priema = '" + textBox14.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox9.Clear();

textBox14.Clear();

}

else

{

MessageBox.Show("Текстовое поле 'Код приема' не заполнено", "Ошибка");

}

}

// Функция "Перезагрузка формы"

private void button11\_Click(object sender, RoutedEventArgs e)

{

vr\_priem form = new vr\_priem();

form.Show();

this.Close();

}

// Функция "Вывод в Excel"

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Поиск по наименованию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox10.Text != "")

{

string filter = textBox10.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox10.Clear();

vr\_priem form = new vr\_priem();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Priem ORDER BY data\_okazania ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Priem ORDER BY data\_okazania DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_vrach form = new glavnaya\_vrach();

form.Show();

this.Close();

}

private void button12\_Click(object sender, RoutedEventArgs e)

{

vr\_klient form = new vr\_klient();

form.Show();

}

}

}

**Код формы врача «БД Клиенты»:**

<Window x:Class="Diplom\_Ohotnikov.vr\_klient"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Клиент'" Height="378.814" Width="800">

<Grid>

<Label Content="База данных 'Клиент'" HorizontalAlignment="Left" Margin="282,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="636,300,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,248,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button3" Content="Поиск" HorizontalAlignment="Left" Margin="577,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Label Content="Поиск по фамилии" HorizontalAlignment="Left" Margin="208,244,0,0" VerticalAlignment="Top" Width="217" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="440,252,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button4" Content="Очистить" HorizontalAlignment="Left" Margin="686,248,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<Button x:Name="button5" Content="↑" HorizontalAlignment="Left" Margin="378,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Label Content="Сортировка по дате оформления" HorizontalAlignment="Left" Margin="35,289,0,0" VerticalAlignment="Top" Width="340" FontSize="20" FontWeight="Bold"/>

<Button x:Name="button6" Content="↓" HorizontalAlignment="Left" Margin="449,295,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Image Margin="3,2,674,272.333" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для vr\_klient.xaml

/// </summary>

public partial class vr\_klient

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public vr\_klient()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_Klient", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_Klient");

DataView dview = new DataView(ds.Tables["Diplom\_Klient"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_Klient";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Вывод в Excel"

private void button2\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

int rowCount = dataGridView1.Items.Count;

int columnCount = dataGridView1.Columns.Count;

for (int j = 0; j < columnCount; j++)

{

Range headerRange = (Range)sheet1.Cells[1, j + 1];

headerRange.Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

headerRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < columnCount; i++)

{

for (int j = 0; j < rowCount; j++)

{

TextBlock textBlock = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

if (textBlock != null)

{

Range cellRange = (Range)sheet1.Cells[j + 2, i + 1];

cellRange.Value2 = textBlock.Text;

}

}

}

}

// Функция "Поиск по фамилии"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"familiya LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по фамилии' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

vr\_klient form = new vr\_klient();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button5\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Klient ORDER BY familiya ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_Klient ORDER BY familiya DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

**Код формы врача «БД Оказанные услуги»:**

<Window x:Class="Diplom\_Ohotnikov.vr\_okazannye\_uslugi"

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:Diplom\_Ohotnikov"

mc:Ignorable="d"

ResizeMode="NoResize"

Icon = "logo2.ico"

Title="База данных 'Оказанные услуги'" Height="438.742" Width="800">

<Grid>

<Label Content="База данных 'Оказанные услуги'" HorizontalAlignment="Left" Margin="210,10,0,0" VerticalAlignment="Top" FontSize="24" FontWeight="Bold"/>

<DataGrid x:Name="dataGridView1" HorizontalAlignment="Left" Height="143" Margin="32,81,0,0" VerticalAlignment="Top" Width="732"/>

<TextBox x:Name="textBox1" HorizontalAlignment="Left" Height="23" Margin="215,269,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox2" HorizontalAlignment="Left" Height="23" Margin="215,304,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<TextBox x:Name="textBox3" HorizontalAlignment="Left" Height="22" Margin="215,341,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Наименование" HorizontalAlignment="Left" Margin="35,264,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<Label Content="Стоимость" HorizontalAlignment="Left" Margin="35,299,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<Label Content="Дата оказания" HorizontalAlignment="Left" Margin="35,332,0,0" VerticalAlignment="Top" Width="122" FontSize="16"/>

<TextBox x:Name="textBox4" HorizontalAlignment="Left" Height="23" Margin="438,269,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" />

<Label Content="Код приема" HorizontalAlignment="Left" Margin="335,263,0,0" VerticalAlignment="Top" Width="103" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button1" Content="Выйти" HorizontalAlignment="Left" Margin="617,369,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button1\_Click"/>

<Button x:Name="button2" Content="Добавить" HorizontalAlignment="Left" Margin="363,299,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button2\_Click"/>

<Button x:Name="button3" Content="Изменить" HorizontalAlignment="Left" Margin="363,333,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button3\_Click"/>

<Button x:Name="button4" Content="Удалить" HorizontalAlignment="Left" Margin="363,369,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button4\_Click"/>

<TextBox x:Name="textBox6" HorizontalAlignment="Left" Height="23" Margin="215,375,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Label Content="Код оказанной услуги" HorizontalAlignment="Left" Margin="35,371,0,0" VerticalAlignment="Top" Width="190" FontSize="16" FontWeight="Bold"/>

<Button x:Name="button5" Content="Вывод в Excel" HorizontalAlignment="Left" Margin="35,228,0,0" VerticalAlignment="Top" Width="146" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button5\_Click"/>

<Button x:Name="button6" Content="Поиск" HorizontalAlignment="Left" Margin="563,229,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button6\_Click"/>

<Label Content="Поиск по наименованию" HorizontalAlignment="Left" Margin="184,227,0,0" VerticalAlignment="Top" Width="245" FontSize="18" FontWeight="Bold"/>

<TextBox x:Name="textBox15" HorizontalAlignment="Left" Height="23" Margin="438,233,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120"/>

<Button x:Name="button7" Content="Очистить" HorizontalAlignment="Left" Margin="667,229,0,0" VerticalAlignment="Top" Width="96" Height="31" FontSize="18" Background="Red" Foreground="White" FontWeight="Bold" Click="button7\_Click"/>

<Button x:Name="button8" Content="↑" HorizontalAlignment="Left" Margin="608,292,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button8\_Click"/>

<Label Content="Сортировка по дате" HorizontalAlignment="Left" Margin="571,259,0,0" VerticalAlignment="Top" Width="192" FontSize="18" FontWeight="Bold"/>

<Button x:Name="button9" Content="↓" HorizontalAlignment="Left" Margin="679,292,0,0" VerticalAlignment="Top" Width="47" Height="31" FontSize="24" Background="Red" Foreground="White" FontWeight="Bold" Click="button9\_Click"/>

<Image Margin="3,2,673.999,332" Source="logo.png" Stretch="Fill"/>

</Grid>

</Window>

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 System.Data;

using System.Data.SqlClient;

using Excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Window = Microsoft.Office.Interop.Excel.Window;

namespace Diplom\_Ohotnikov

{

/// <summary>

/// Логика взаимодействия для vr\_okazannye\_uslugi.xaml

/// </summary>

public partial class vr\_okazannye\_uslugi

{

private string connectionString = "data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57";

SqlCommand cmd;

SqlConnection con;

SqlDataAdapter da;

DataSet ds;

public vr\_okazannye\_uslugi()

{

InitializeComponent();

GetList();

LoadData();

}

void GetList()

{

con = new SqlConnection(@"data source = stud-mssql.sttec.yar.ru,38325; initial catalog = user57\_db; user id = user57\_db; password = user57");

con.Open();

da = new SqlDataAdapter("SELECT \* FROM Diplom\_OkazannyeUslugi", con);

ds = new DataSet();

da.Fill(ds, "Diplom\_OkazannyeUslugi");

DataView dview = new DataView(ds.Tables["Diplom\_OkazannyeUslugi"]);

dataGridView1.ItemsSource = dview;

con.Close();

}

private void LoadData()

{

string query = "SELECT \* FROM Diplom\_OkazannyeUslugi";

SqlConnection connection = new SqlConnection(connectionString);

da = new SqlDataAdapter(query, connection);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(da);

System.Data.DataTable table = new System.Data.DataTable();

da.Fill(table);

dataGridView1.ItemsSource = table.DefaultView;

}

// Функция "Добавить"

private void button2\_Click(object sender, RoutedEventArgs e)

{

if ((textBox1.Text != "") && (textBox2.Text != "") && (textBox3.Text != "") && (textBox4.Text != ""))

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "INSERT INTO Diplom\_OkazannyeUslugi (naimenovanie,stoimost,data\_okazania,kod\_priema) values (" + textBox1.Text + ",'" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "')";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное добавление данных в таблицу"), "Сообщение");

}

else

{

MessageBox.Show("Одно из текстовых полей не заполнено", "Ошибка");

}

}

// Функция "Изменить"

private void button3\_Click(object sender, RoutedEventArgs e)

{

if (textBox6.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "UPDATE Diplom\_OkazannyeUslugi SET naimenovanie = '" + textBox1.Text + "',stoimost='" + textBox2.Text + "',data\_okazania='" + textBox3.Text + "',kod\_priema='"

+ textBox4.Text + "' where kod\_okazannoy\_uslugi = '" + textBox6.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное изменение данных в таблице"), "Сообщение");

}

else

{

MessageBox.Show("Текстовое поле 'Код оказанной услуги' не заполнено", "Ошибка");

}

}

// Функция "Удалить"

private void button4\_Click(object sender, RoutedEventArgs e)

{

if (textBox6.Text != "")

{

cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "DELETE FROM Diplom\_OkazannyeUslugi WHERE kod\_okazannoy\_uslugi = '" + textBox6.Text + "'";

cmd.ExecuteNonQuery();

con.Close();

GetList();

MessageBox.Show(string.Format("Успешное удаление данных в таблице"), "Сообщение");

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox6.Clear();

}

else

{

MessageBox.Show("Текстовое поле 'Код оказанной услуги' не заполнено", "Ошибка");

}

}

// Функция "Вывод в Excel"

private void button5\_Click(object sender, RoutedEventArgs e)

{

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

excel.Visible = true;

Workbook workbook = excel.Workbooks.Add(System.Reflection.Missing.Value);

Worksheet sheet1 = (Worksheet)workbook.Sheets[1];

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

Range myRange = (Range)sheet1.Cells[1, j + 1];

sheet1.Cells[1, j + 1].Font.Bold = true;

sheet1.Columns[j + 1].ColumnWidth = 15;

myRange.Value2 = dataGridView1.Columns[j].Header;

}

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

for (int j = 0; j < dataGridView1.Items.Count; j++)

{

TextBlock b = dataGridView1.Columns[i].GetCellContent(dataGridView1.Items[j]) as TextBlock;

Microsoft.Office.Interop.Excel.Range myRange = (Microsoft.Office.Interop.Excel.Range)sheet1.Cells[j + 2, i + 1];

myRange.Value2 = b.Text;

}

}

}

// Функция "Поиск по наименованию"

private void button6\_Click(object sender, RoutedEventArgs e)

{

if (textBox15.Text != "")

{

string filter = textBox15.Text;

DataView view = dataGridView1.ItemsSource as DataView;

if (view != null)

{

view.RowFilter = $"naimenovanie LIKE '%{filter}%'";

}

}

else

{

MessageBox.Show("Текстовое поле 'Поиск по наименованию' не заполнено", "Ошибка");

}

}

// Функция "Очистить"

private void button7\_Click(object sender, RoutedEventArgs e)

{

textBox15.Clear();

vr\_okazannye\_uslugi form = new vr\_okazannye\_uslugi();

form.Show();

this.Close();

}

// Функция "Сортировка по возрастанию"

private void button8\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_OkazannyeUslugi ORDER BY naimenovanie ASC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Сортировка по убыванию"

private void button9\_Click(object sender, RoutedEventArgs e)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand("SELECT \* FROM Diplom\_OkazannyeUslugi ORDER BY naimenovanie DESC", connection);

SqlDataReader reader = command.ExecuteReader();

System.Data.DataTable table = new System.Data.DataTable();

table.Load(reader);

dataGridView1.ItemsSource = table.DefaultView;

reader.Close();

}

}

// Функция "Выйти"

private void button1\_Click(object sender, RoutedEventArgs e)

{

glavnaya\_vrach form = new glavnaya\_vrach();

form.Show();

this.Close();

}

}

}