

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно–Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Рецензент, заместитель Министра
Здравоохранения Челябинской области

_____ / А.В. Кузнецов /

« _____ » _____ 2021 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой, д.т.н., с.н.с.

_____ / Б.М. Суховилов /

« _____ » _____ 2021 г.

Разработка программного обеспечения «Анализу динамики медицинских исследований»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.04.03.2021.403.ВКР

Руководитель, к.т.н. доцент

_____ / В.А. Конов /

« _____ » _____ 20 ____ г.

Автор

студент группы ЭУ–221

/ Е.М. Касаткина /

« _____ » _____ 20 ____ г.

Нормоконтролер,

старший преподаватель

/ Е.Н. Горных /

« _____ » _____ 20 ____ г.

Челябинск 2021

АННОТАЦИЯ

Касаткина Е.М. Разработка программного обеспечения по «Аналізу динамики медицинских исследований»– Челябинск: ЮУрГУ, ЭУ–221, 84стр., 42 ил., 3 табл., библиогр. список –10 наим.

Разработка архитектуры программного обеспечения по «Аналізу динамики медицинских исследований»

В данной выпускной квалификационной работе выполнена разработка программного обеспечения по «Аналізу динамики медицинских исследований».

В ходе работы была изучена предметная область. Объектом исследования являются программное обеспечение «Аналитика динамики медицинских исследований».

Цель работы – рассмотреть самые известные сервисы по обработке медицинских анализов, проанализировать и разработать программного продукта удобную для врачей, помогающую в работе.

В процессе работы были изучены такие сервисы, как – Барс МИС, Медик +. В результате исследования можно сказать, каждый из них достоин внимания, сервисы легки в освоении. Но динамики, более развернутой по показателям медицинских анализов, в данных программных продуктах не представлено и не предусмотрено прогнозирования дальнейших результатов медицинских показателей. На основании этого были сформулированы требования к разрабатываемому программному решению, аналитика и прогнозирование медицинских лабораторных исследований.

В следствии актуальности и для модернизации работы медицинской организации по диагностике, контролю хода лечения и восстановлению

пациентов, разработано программное обеспечение по «Аналізу динаміки медичинських досліджень».

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ПОСТАНОВКА ЗАДАЧИ	8
I. РАЗРАБОТКА АРХИТЕКТРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «АНАЛИЗУ ДИНАМИКИ МЕДИЦИНСКИХ ИССЛЕДОВАНИЙ»	9
1.1 Оценка функционала «Барс МИС» и «Медик +»	9
1.2 Общая концепция	13
1.3 Системный администратор	15
1.4 Регистратура	17
1.5 Лаборатория	19
1.6 Вывод данных	20
II. СОЗДАНИЕ БАЗЫ ДАННЫХ, СОЗДАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ДЛЯ ВВОДА ДАННЫХ	22
2.1 Описание структуры данных и проектирование базы данных	22
2.2 Создание графического интерфейса	26
V. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ В ГБУЗ «ОКСПНБ №1»	40
5.1 Подготовка к внедрению	40
5.2 План внедрения	42
5.3 Процесс внедрения	43
5.3 Документооборот в процессе внедрения	49
ЗАКЛЮЧЕНИЕ	50
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	51

ВВЕДЕНИЕ

Лабораторная диагностика (греч, diagnostikos способный распознавать) — раздел клинической диагностики, изучающий методы оценки физиологических и распознавания, и оценки патологических состояний и заболеваний, основанный на исследованиях клеточного и химического состава, физико–химических и биологических свойств тканей и жидкостей организма, а также методы обнаружения возбудителей заболеваний.

На сегодняшний день в медицине лабораторные исследования занимает одно из важнейших мест среди диагностических данных. Лабораторная диагностика раскрывает метаболические и клеточные процессы, позволяет выявить ранние отклонения от норм Всемирной организации здравоохранения, иногда задолго до появления симптомов, и видимых изменений структуры пораженных органов.

Научной основой лабораторной диагностики является медицинская научная дисциплина, называемая в СССР клинической лабораторной диагностикой, за рубежом — клинической биологией, клинической патологией, лабораторной медициной.

В клинической медицине технологии лабораторной диагностики употребляют генеральным манером ради утверждения диагноза болезни, ради характеристики тяжести, периода и срока заболевания, временами ради нахождения его прогноза, для контроля после лечения.

Лабораторная диагностика – неотъемлемая часть современной медицины. Что делает врач после вопроса: «На что жалуетесь?», дает направление на анализы.

Сегодня, когда многие страны стараются оптимизировать расходы на здравоохранение, услуги лабораторий становятся особенно востребованными, так как помогают сократить расходы на лечение.

Цели автоматизации лаборатории:

- Избавление от непроизводительной рутинной работы, уменьшение затрат времени на проведение исследований.
- Поддержка медицинских лабораторных процессов: использование референсных норм, ввод правил, ведение статистики по пациенту, поддержка стандартов.
- Минимизация количества ошибок и контроль рабочих процессов.
- Комплексный взгляд на проведенные исследования до передачи результата.
- Контроль качества лабораторных исследований в соответствии с российскими и международными стандартами.
- Мониторинг и контроль результатов, ответственность лабораторной службы за переданные данные.
- Управление рабочими потоками лаборатории, их планирование и оптимизация.
- Возможность лаборанта уверенно опираться на полученные данные.

Медицинские анализы помогают сэкономить денежные средства, т.к. уменьшают:

- количество неверно поставленных диагнозов,
- контроль результативности лечения,
- уменьшения случаев осложнений, госпитализаций и реабилитацию.

Если обратиться к мировой статистике, то увеличение затрат на 1% на услуги лаборатории, в последствии понижает расходы на лечение на 5%.

Не стоит думать, что анализы нужны только для подтверждения диагноза. Они имеют большое значение и для профилактики, так как помогают выявить заболевание еще до появления клинических симптомов.

Сегодня основными причинами смерти россиян являются онкология и сердечно–сосудистые заболевания. Их ранняя диагностика повышает шансы на успешное лечение и снижает смертность от этих болезней. Так, жидкостная цитология (входит в новый порядок диспансеризации в РФ) снижает смертность от рака шейки матки в среднем на 80%, исследование на скрытую кровь в кале (входит в новый порядок диспансеризации в РФ) – смертность от рака толстой или прямой кишки в среднем на 15%. Динамика и аналитика могут показать полную картину и ход лечения, восстановления, зарождение болезни.

Цель работы – рассмотреть самые известные сервисы по обработке медицинских анализов, проанализировать и разработать программный продукт. Для достижения цели исследования поставлены следующие задачи:

1. Анализ предметной области
2. Анализ существующих программных решений
3. Формулирование требований к программному обеспечению
4. Формирование аппаратной части:
 - a. Разработка структурной части программного обеспечения
 - b. Сформировать базу данных, ввод данных в базу
5. Анализ существующих методов обработки данных
6. Тестирование программного продукта
7. Внедрение программного обеспечения

ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать программное обеспечение по «Аналізу динаміки медичинських досліджень», дозволяюче:

- Аналізувати динаміку медичинських досліджень в певний проміжок часу
- Прогнозувати результати медичинських досліджень на 1 день вперед

Для рішення поставленої мети необхідно реалізувати поставлені завдання:

1. Аналіз предметної області, медичинських лабораторних досліджень
2. Аналіз існуючих програмних рішень на прикладі двох найбільш великих і поширених програмних продуктів «Барс МІС» і «Медик +»
3. Формулювання вимог до програмного забезпечення, після аналізу інших програмних продуктів
4. Формування технічної частини:
 - а. Розробка оптимізованої структурної частини програмного забезпечення
 - б. Сформулювати оптимізовану базу даних, адаптований ввід даних в базу даних
5. Аналіз існуючих методів обробки даних, реалізація методів обробки
6. Тестування програмного продукту на ГБУЗ «ОКСПНБ№1»

I. РАЗРАБОТКА АРХИТЕКТРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «АНАЛИЗУ ДИНАМИКИ МЕДИЦИНСКИХ ИССЛЕДОВАНИЙ»

1.1 Оценка функционала «Барс МИС» и «Медик +»

Мед. Программные решения использовались с 1960-х годов, первое программное обеспечение в медицине Lockheed. В конце 1970-х и в начале 1980-х годов информационные программные продукты стали более используемыми и эффективными, «медицинское программное обеспечение» стала помогать управлять данными и операциями в медицинской индустрии. В начале 1990-х годов медицинское программные продукты стали важными для медицинских устройств:

- как ядерная медицина
- кардиология
- медицинская робототехника

Реализован стандарт ISO 9000-3 и Европейской директивы по медицинским устройствам в 1993 году внесли формализацию существующих законов для работы с медицинскими устройствами и с программными решениями.

В 2006 г. ИЕС 62304 был закреплён принцип разработки и тестирования программных продуктов для медицинских устройств.

Российский рынок медицинских информационных систем состоит из большого количества программных решений.

Медицинские информационные системы производятся 45% – в Москве, 10% – в Санкт-Петербурге.

Мы рассмотрим основные 2 которые распространены в большем количестве и являются лидерами продаж на рынке в Челябинской области.

Рассмотрим программный продукт «Барс МИС» в разрезе медицинских исследований, и дадим оценку его работе.

Так изучив работу программного продукта «Барс МИС», можно более подробно остановиться на вкладке, представленной на рисунке 1.

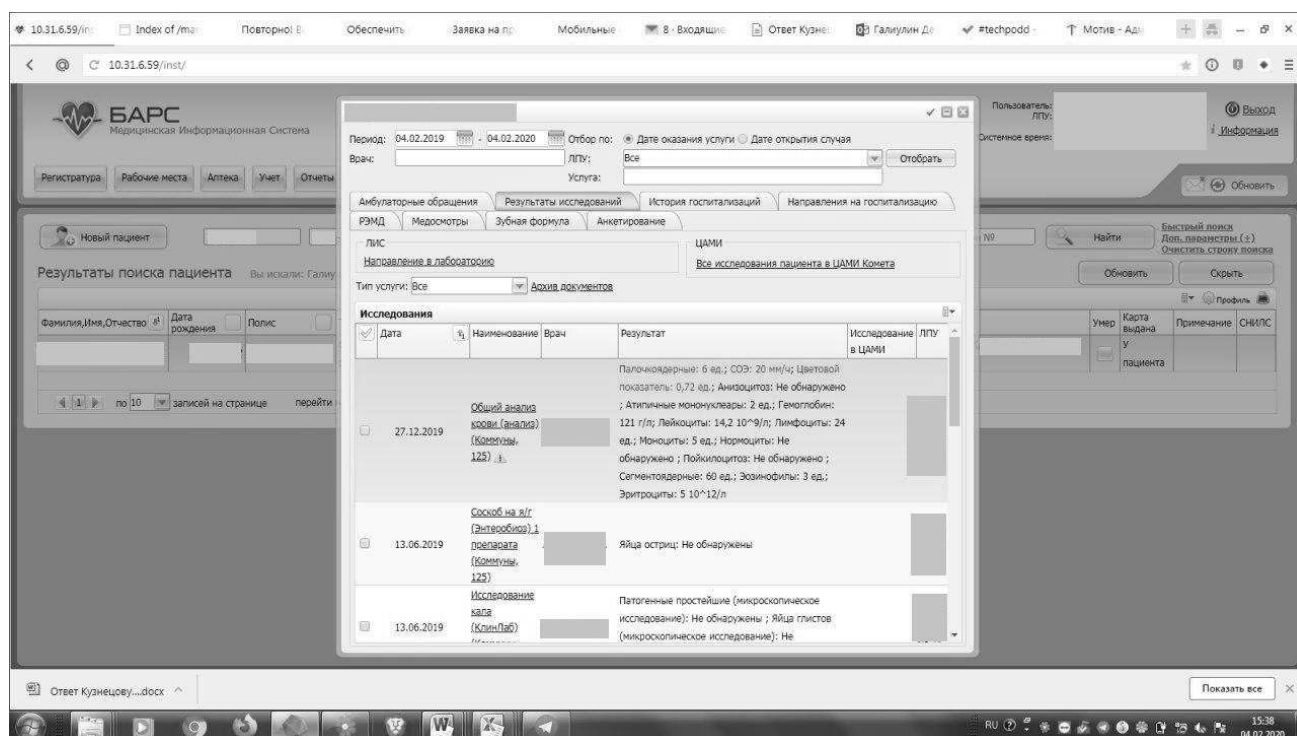


Рисунок 1–Лабораторные исследования в «Барс МИС»

В данном случае видно, что проанализировать медицинские исследования в динамике, а также увидеть улучшение или ухудшение состояния пациента в полной мере не представляется возможным, не столкнувшись с рядом затруднений.

Медицинские исследования выдаются списком, но необходимо рассматривать каждый показатель, что не совсем удобно для понимания имеющейся динамики. Например, в течении недели можно столкнуться с рядом затруднений и упустить незначительные изменения или отклонения, что очень важно в данной области.

В программном продукте «Барс МИС» очень удобный функционал для внесения результатов параметров исследований «Скрин» для заполнения представлен на рисунке 2. Единственным минусом является то, что нормыпоказателей медицинских анализов Всемирной организации

здравоохранения не вносят многие медицинские организации, в справочник для последующего использования и аналитики, что могло предотвратить ряд медицинских ошибок.

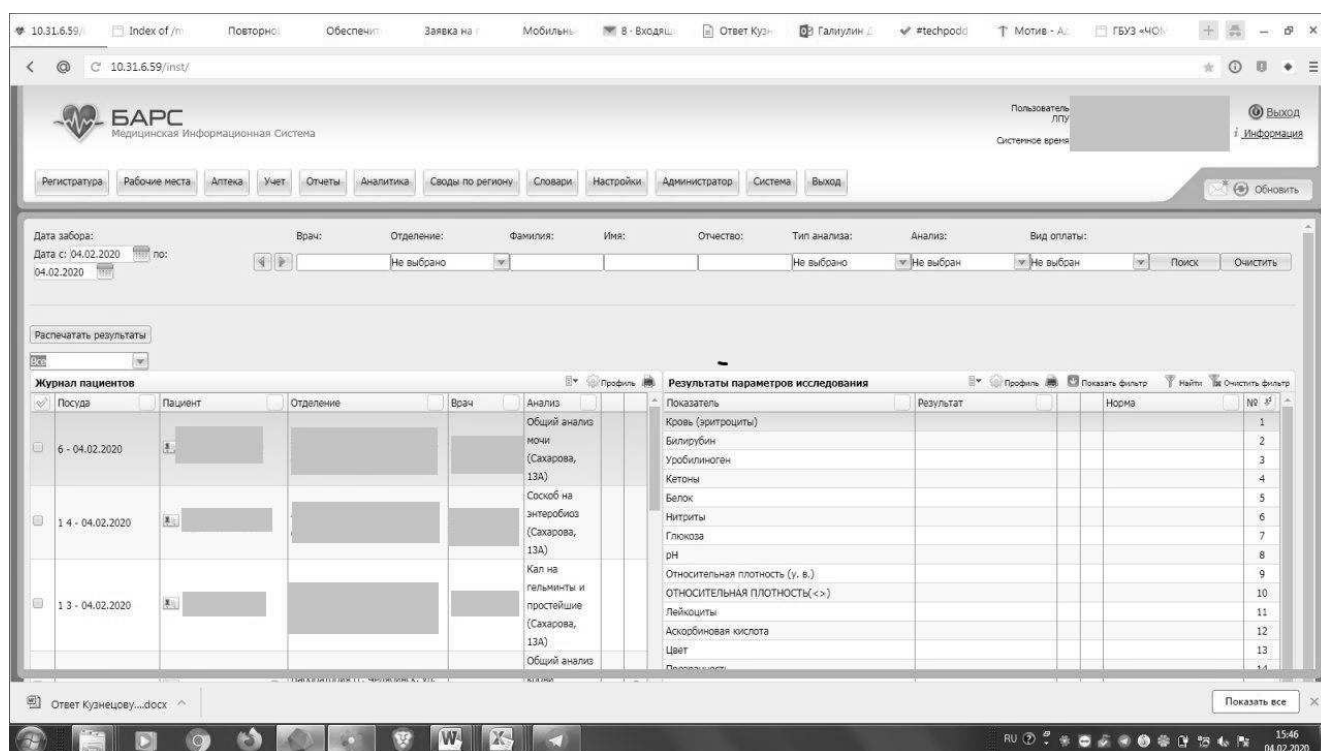


Рисунок 2

Изучив интерфейс медицинской информационной системы «Барс. МИС» мы можем прийти к выводу, что по заявленной цели в данном случае мы не можем анализировать конкретный показатель медицинского лабораторного исследования пациента в динамике, при этом опираясь на нормы Всемирной организации здравоохранения, а также не можем спрогнозировать следующий результат исследования, кроме того еще одним не мало важным минусом данной информационной системы является облачность хранения данных, соответственно без доступа к сети интернет или выхода из строя любого другого модуля система не будет доступна, как результат задержка в работе.

Далее рассмотрим программный продукт «Медик +» в разрезе исследования, и их анализ.

Рассмотрим раздел «лабораторные исследования» который представлен на рисунке 3.

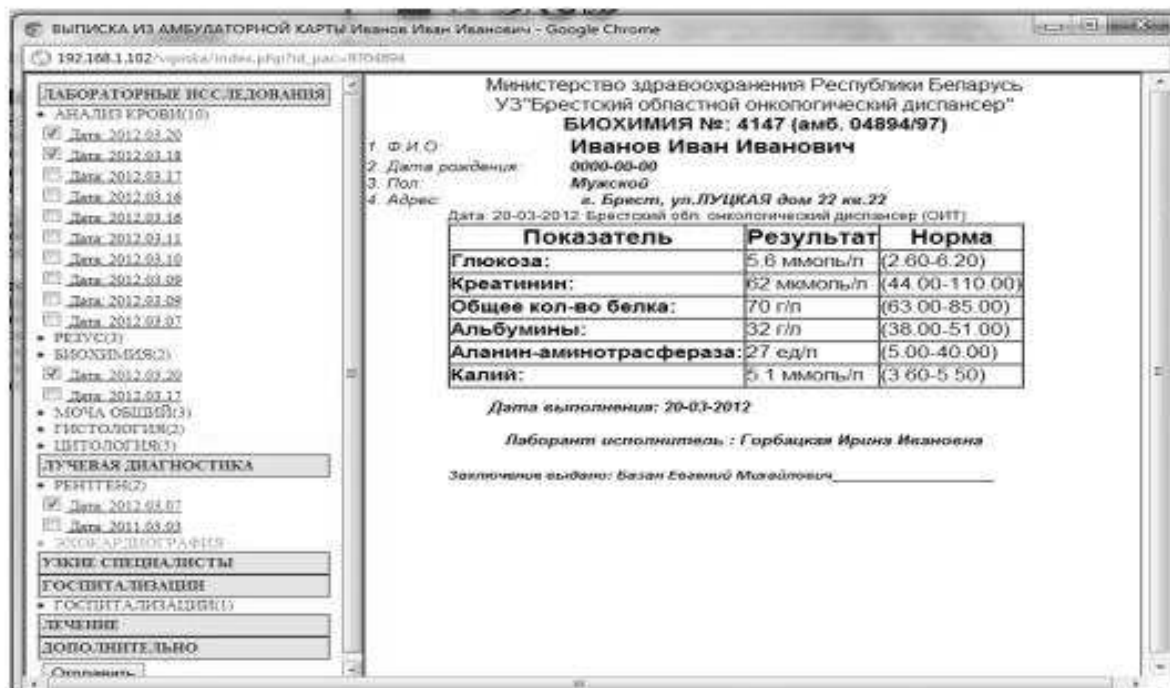


Рисунок 3

В данной системе медицинские анализы выбираются каждый отдельно, а для просмотра динамики необходимо распечатать и проанализировать ее вручную. Но большим преимуществом в данном случае является отображение норм показателей медицинских анализов Всемирной организации здравоохранения, что дает возможность специалисту сразу проверить показатели, не соответствующие норм и принять решение оперативно.

Важным условием программного продукта «Медик +» является очень удобный функционал для внесения результатов исследования для лаборатории который представлен на рисунке 4.

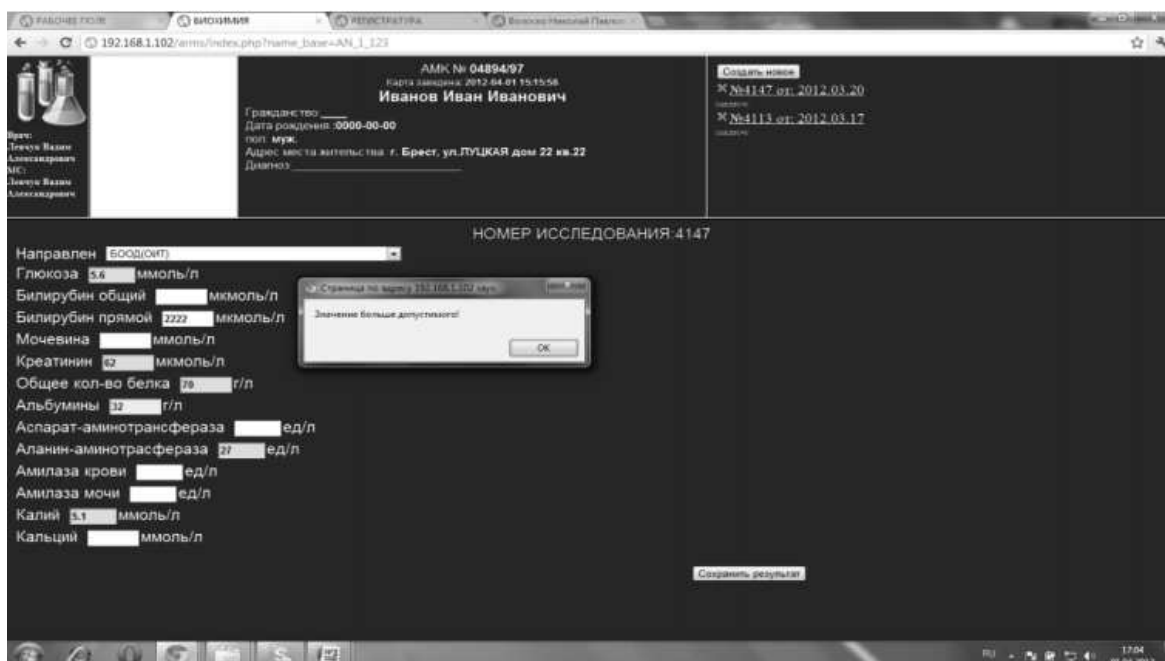


Рисунок 4

Во время ввода данных, которые отличаются от нормы показателей медицинских анализов Всемирной организации здравоохранения, у специалиста высвечивается оповещение.

1.2 Общая концепция

В соответствии с поставленной задачей и проанализированными функциональными способностями рассмотренных медицинских систем, мной была разработана архитектура программного продукта.

Представлен планируемый интерфейс выводного результата, для анализа врачей показателей медицинских исследований в динамике.

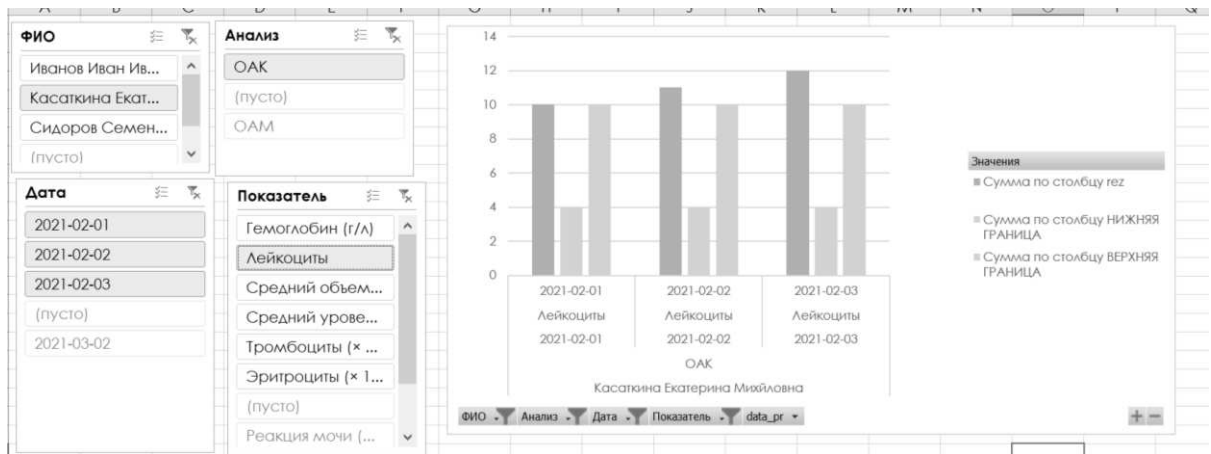


Рисунок 5– Аналитика медицинских исследований

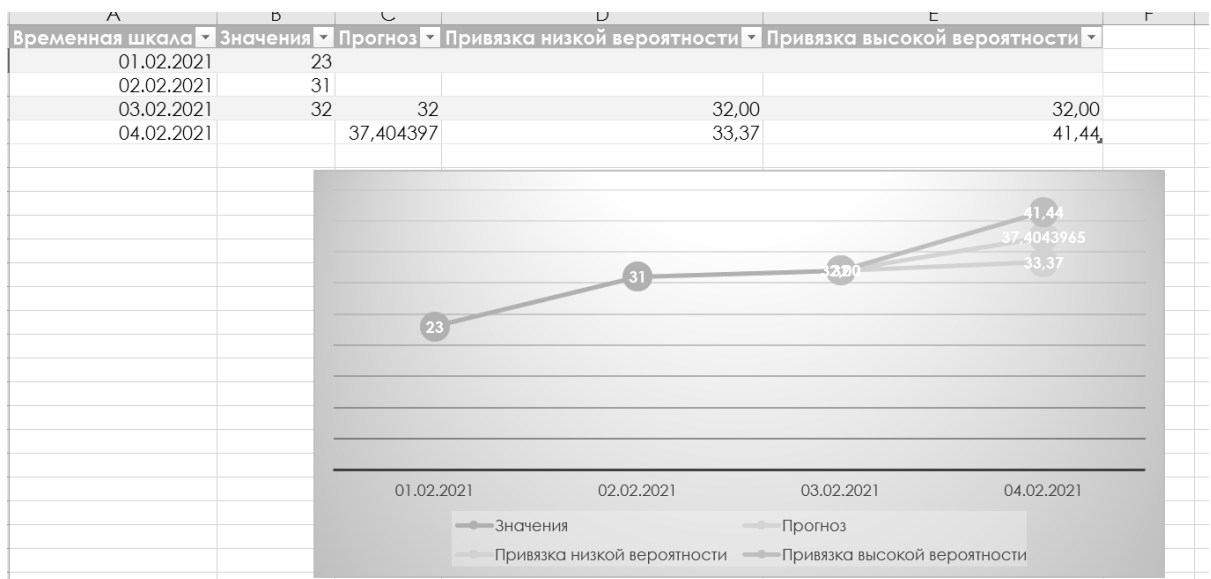


Рисунок 6– Прогнозирование результата

Данные будут браться из базы данных. Выгрузка производится будет в формат установленной таблицы с фиксированными полями, данные будут подгружаться в одну таблицу по заранее заданному стандарту, показатели анализируются при выгрузке по показателям Всемирной организации здравоохранения. Кроме того, для каждой строки (для каждого отдельного показателя) должна быть выгрузка в график. Возможность анализа данных для прогнозирования.

Вместе с тем, была проанализирована потребность в медицинских организациях участников работы в данной системе, и выявлено, что возможно создать следующие роли:

1. Системный администратор, его задачей будет формировать анализ и вносить данные Всемирной организации здравоохранения (ВОЗ) в зависимости от нормативных правовых актов и потребностей медицинской организации.
2. Лаборатория – внесение данных по медицинским показателям.
3. Регистратура, которая формирует базу данных пациентов.
4. Врач выгружает данные для анализа, назначения или коррекции лечения.

Схема ввода и вывода данных:

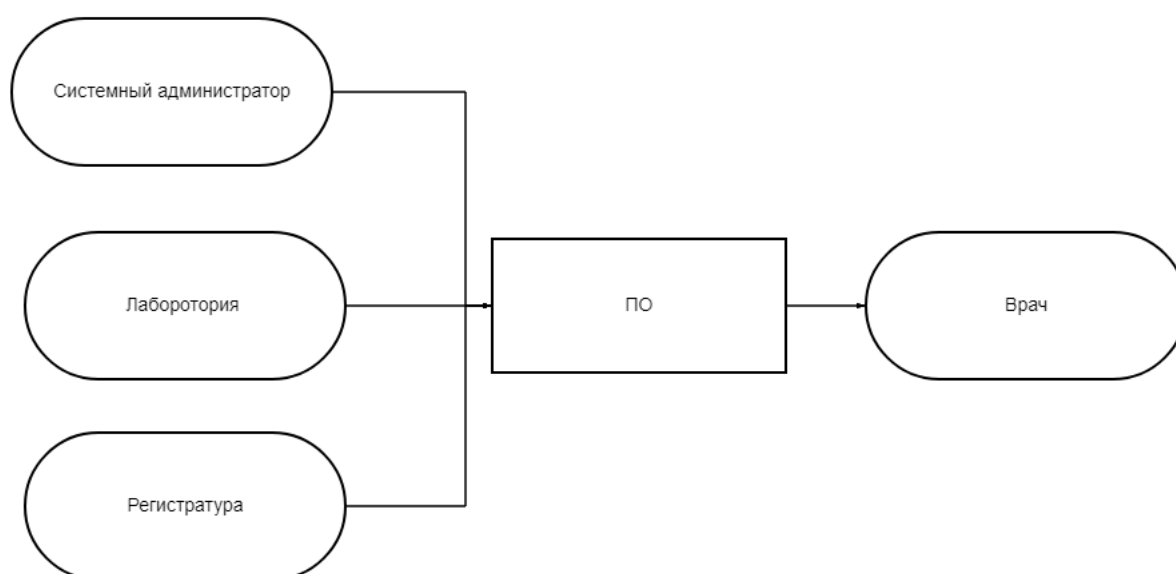


Рисунок 7 –Схема ввода и вывода данных.

1.3 Системный администратор

Системный администратор формирует анализ и вводит данные, данная роль выбрана именно для системного администратора, так как в основном в медицинских организациях настройкой программного обеспечения выступает

отдел информационных технологий, по данным предоставленным медицинским персоналом:

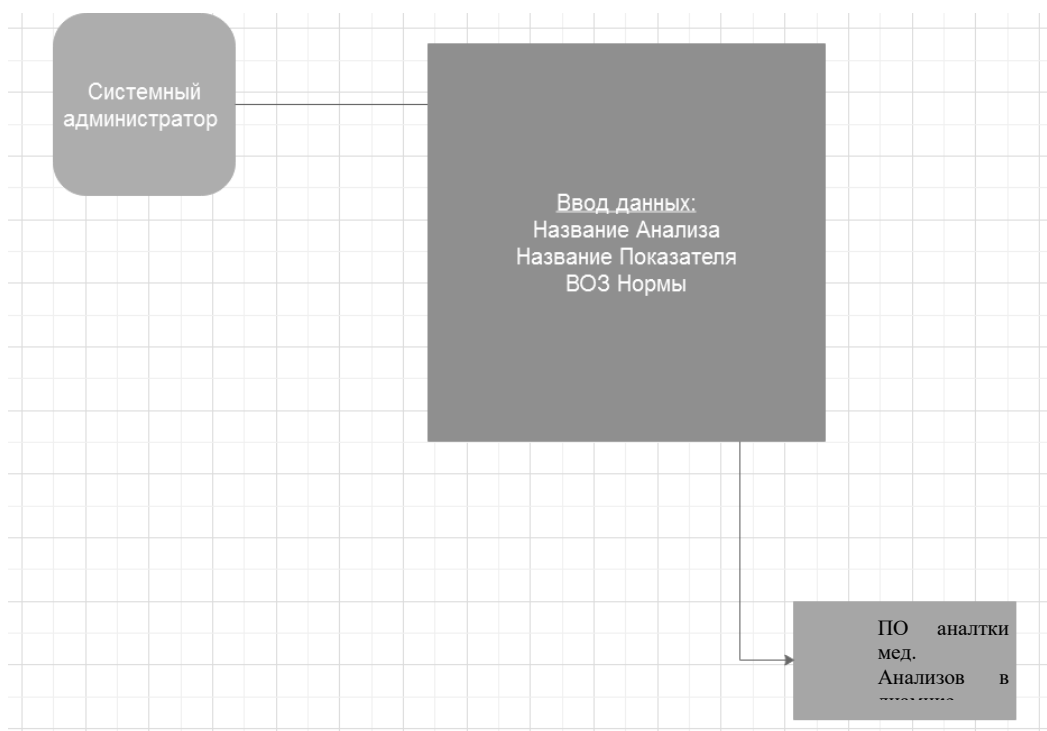


Рисунок 8– Схема вводных данных системным администратором

Системный администратор будет формировать анализ. В базу данных будет заносить названия медицинских анализов, названия показателей, данные Всемирной организации здравоохранения (в данном случае эта программа будет иметь актуальность при любых изменениях, а также будет достаточно гибкая)

id_analis	nazv_analiz
2	OAK
3	OAM
4	OAM

Рисунок 9– Таблица название медицинских анализов

id_pokaz	nazv_pokaz	id_analiz	v_voz	n_voz
1	Лейкоциты	2	10	4
2	Эритроциты (× 10 ¹² /л)	2	6	4
3	Средний объем эритроцитов (фл или мкм ³)	2	95	81
5	Гемоглобин (г/л)	2	150	128
6	Средний уровень HGB в эритроците (пг)	2	32	26
7	Тромбоциты (× 10 ⁹ /л)	2	318	178
8	Удельный вес мочи	3	1025	1010
9	Реакция мочи (pH)	3	6	5

Рисунок 10– Таблица название медицинских показателей и данных ВОЗ

На рисунке 9 представлена таблица `analiz` будет присваиваться полю `id_analiz` автоматически, с типом данных `int`, полю `nazv_analiz` будет вноситься название анализа с типом данных `nvarchar(50)`.

Таблица 1. Наименование полей и тип данных.

Наименование поля таблицы <code>pokaz</code>	Тип данных	Данные
<code>Id_pokaz</code>	<code>Int</code>	Автоматический номер
<code>Nazv_pokaz</code>	<code>Nvarchar(50)</code>	Название показателя
<code>Id_analiz</code>	<code>Int</code>	Номер названия анализа
<code>V_voz</code>	<code>Int</code>	Верхняя граница нормы Всемирной организации здравоохранения
<code>N_voz</code>	<code>Int</code>	Нижняя граница нормы Всемирной организации здравоохранения

Базы данных, название анализа и название показателя вынесены в разные отдельные таблицы для того чтобы при изменении потребностей показателей в конкретном анализе можно было формировать на уровне администратора разные параметры.

1.4 Регистратура

Регистратор ввод данные:

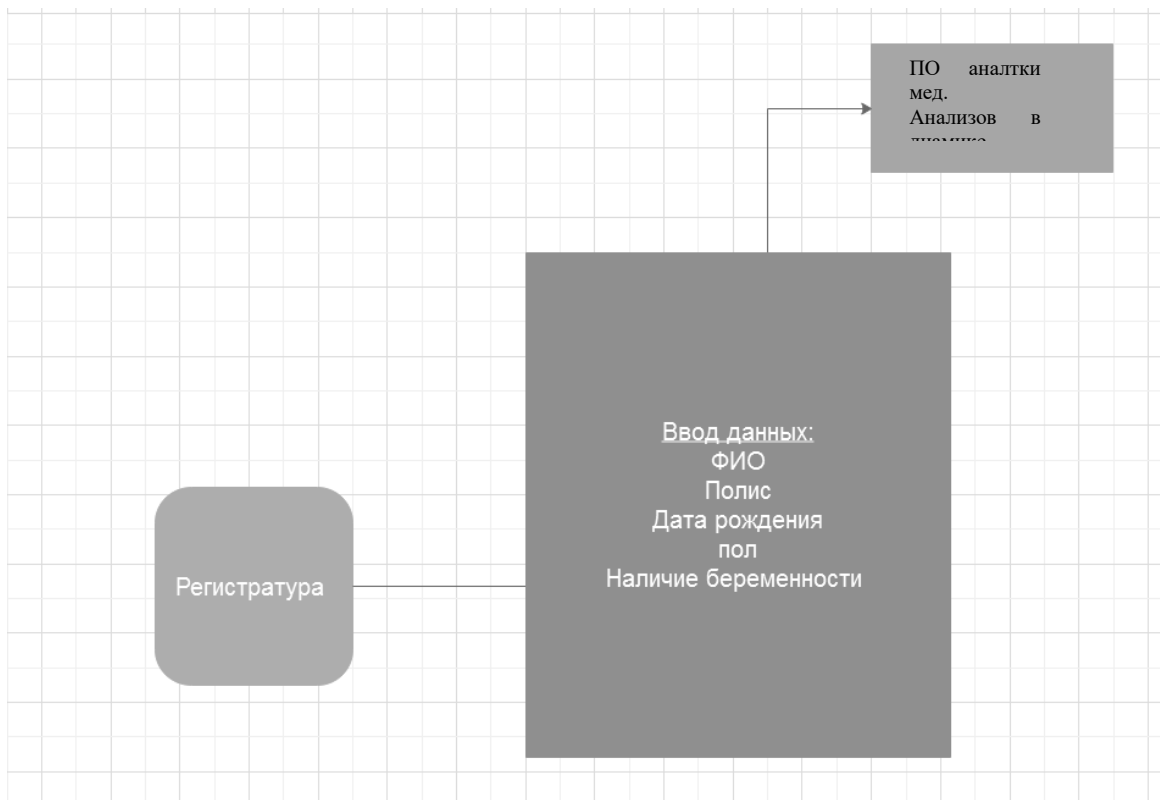


Рисунок 11 –Схема вводных данных регистратурой.

id_pac	fio	polis	dr	otd
7	Касаткина Екатерина Михайловна	12345678	1991-01-01	12
8	Иванов Иван Иванович	987654321	1990-01-01	27
9	Сидоров Семен Александрович	987456321	1992-02-01	8

Рисунок 12 –Вводные данные заполняемые регистратурой

Таблица 2. Наименование полей и тип данных.

Наименование поля таблицы rasient	Тип данных	Данные
Id_pac	Int	Автоматический номер
fio	Nvarchar(50)	Фамилия имя отчество
polis	Int	Номер медицинского полиса
dr	data	Дата рождения
otd	Int	Номер отделения

1.5 Лаборатория

Лаборант вводит данные:

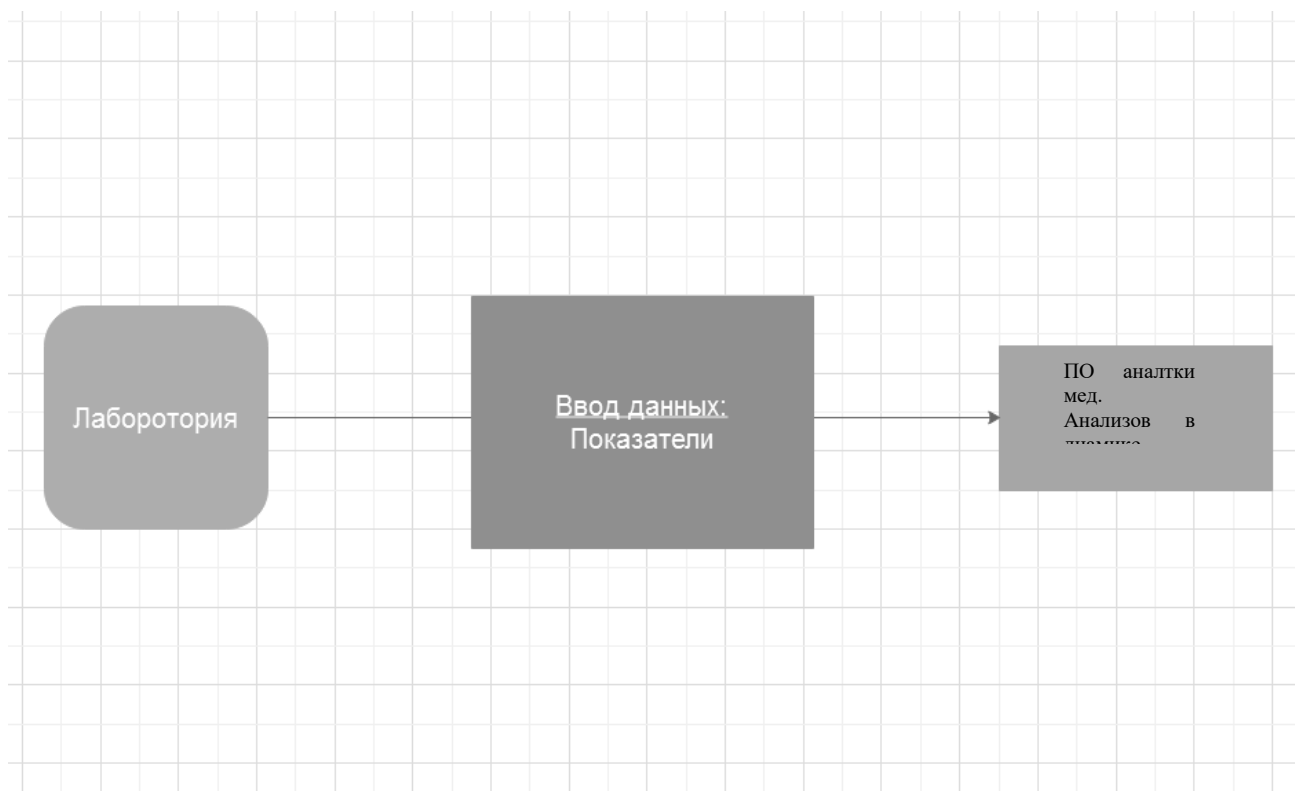


Рисунок 13 –Схема данных вводимых лабораторией.

id_itog	id_pac	id_pokaz	Дата	rez	data_pr
5	7	6	2021-02-01	23	2021-02-01
22	7	6	2021-02-02	31	2021-02-02
23	7	6	2021-02-03	32	2021-02-03

Рисунок 14– Вводимы данные лабораторией

Лаборатория будет вносить данные значения показателя и даты проведения исследования.

Таблица 3. Наименование полей и тип данных.

Наименование поля таблицы itogi	Тип данных	Данные
Id_itog	Int	Автоматический номер
Id_pac	Int	Номер пациента
Id_pokaz	Int	Номер названия показателя
Data_pr	Data	Дата проведения анализа
rez	Int	Значение результата

1.6 Вывод данных

После проведения исследования врач получает данные:

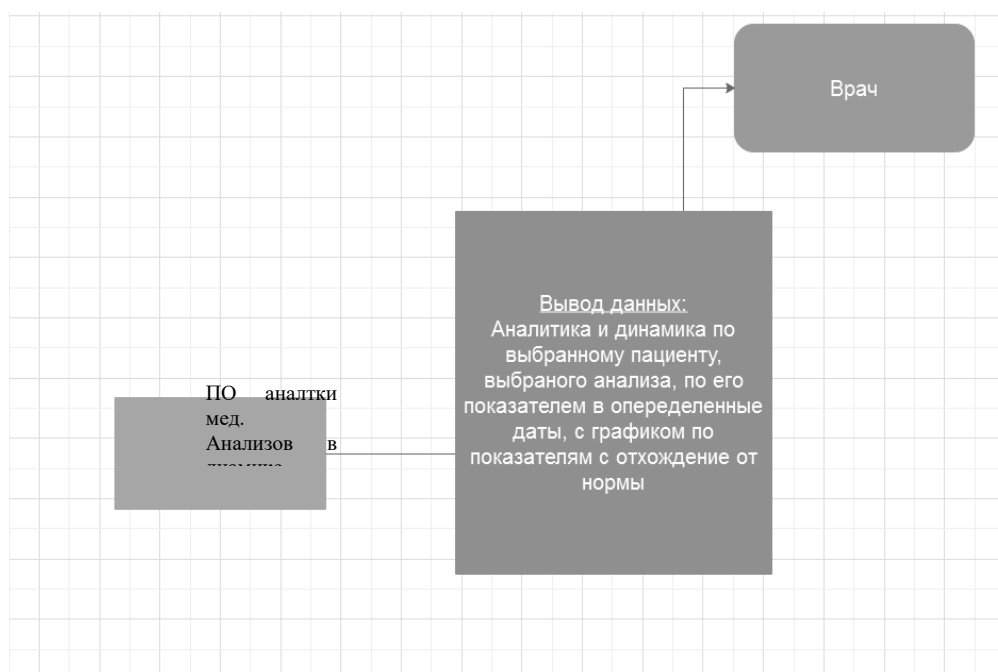


Рисунок 15– Схема вывода данных врача

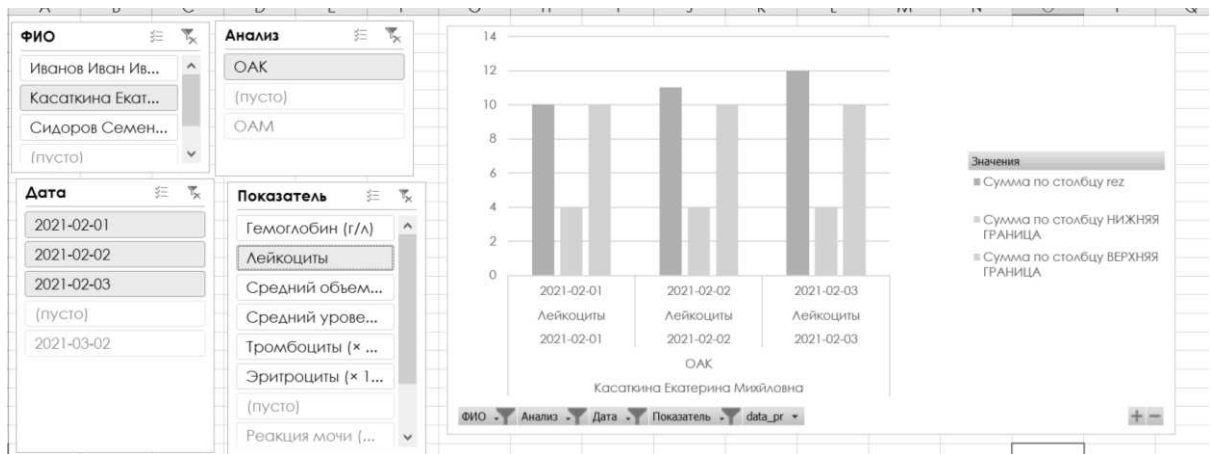


Рисунок 16 –Вывод данных в результате

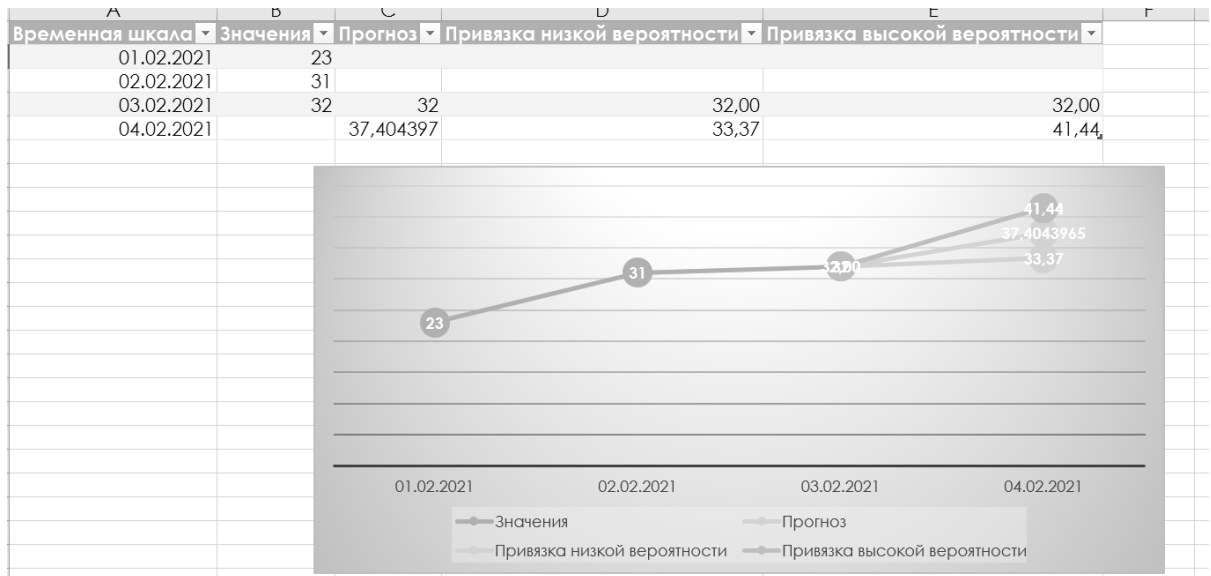


Рисунок 17 –Прогнозируемый результат

Выгрузка будет производиться в формат установленной таблицы с фиксированными полями. Данные будут подгружаться в одну таблицу по заранее заданному стандарту (Рисунок 16). Показатели выгружаются в диаграмму показателями ВОЗ (нижняя/верхняя граница). Кроме того, для каждой строки (для каждого отдельного показателя) должна быть выгрузка в график и возможность анализировать данные и рассчитывать прогноз (Рисунок 17).

II. СОЗДАНИЕ БАЗЫ ДАННЫХ, СОЗДАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ДЛЯ ВВОДА ДАННЫХ

2.1 Описание структуры данных и проектирование базы данных

Этап проектирования базы данных является одним из ключевых в работе. От успешности принятых на данном шаге решений зависит простота построения системы.

Медицинские анализы пациента представляют собой записи, которые содержат данные пациента, название анализа, название показателей, результат исследования и дату исследования.

На основании данной информации было принято решение создать следующую структуру базы данных (Рисунок 18):

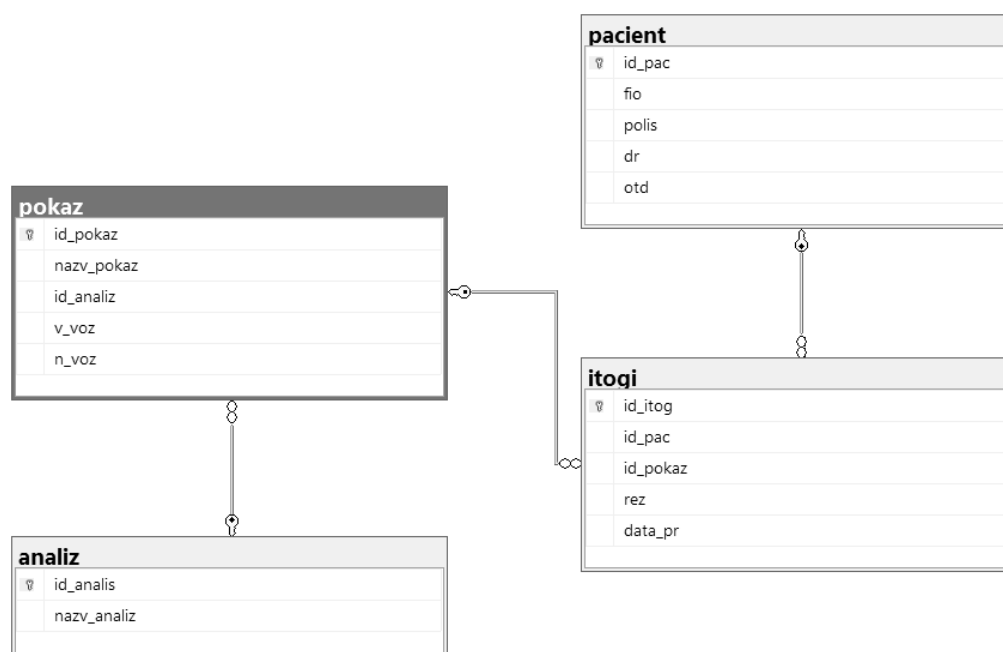


Рисунок 18 – Схема базы данных

Основной нашей Таблицей является itogi(Итоговые показатели медицинских исследований, вносимые лабораторией), в данную таблицу подгружаются 2 таблицы:

1. Pacient таблица с данными пациентов (вносимая регистратурой)

2. Показтаблица с названиями показателей и данными Всемирной организацией здравоохранения, в эту таблицу также приходят данные из таблицы analiz с названием анализа (данные 2 таблицы вносит системный администратор).

Построение данной базы данных выполняется в среде управления MS SQL Server 2019.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основным используемым языком запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Для работы с данными написаны хранимые процедуры для добавления, изменения, удаления данных в каждой таблице (Рисунок 19).

Хранимая процедура — это объект базы данных, в виде набора SQL-инструкций, который компилируется 1 раз и хранится на сервере.

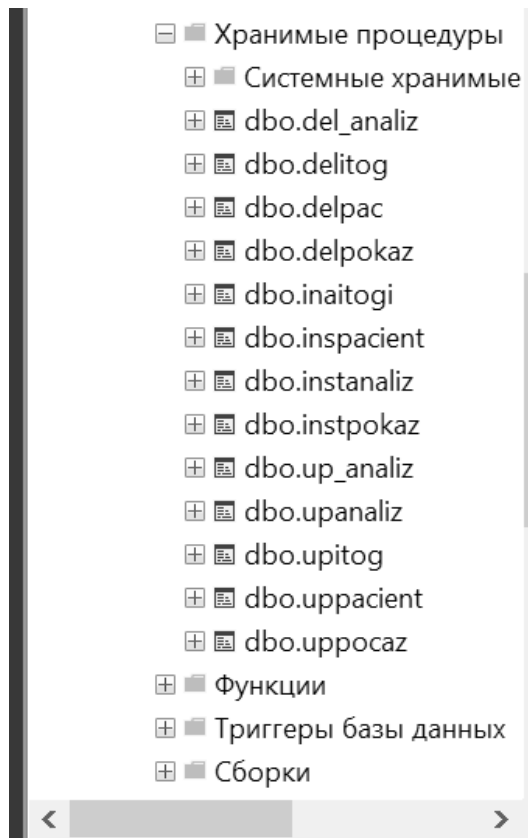


Рисунок 19 – Хранимые процедуры

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[instanaliz]    Script Da
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[instanaliz]
    @nazv_analiz nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    insert into analiz(nazv_analiz)
    values (@nazv_analiz)
END

```

Рисунок 20 Хранимая процедура по добавлению в таблицу Название анализа


```

SQLQuery1.sql - WI...VL6EVJ\User (140)  X WIN-I4C76VL6EVJ.A...итика - Diagram_1*
USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[del_analiz]    Script
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[del_analiz]
    @id1 int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    delete [dbo].[analiz] where id_analis = @id1
END

```

Рисунок 21—Хранимая процедура на удаление из таблицы Название анализа

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[up_analiz]    Script Date:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[up_analiz]
    @nazv_analiz nvarchar(50),
    @id_a int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    update analiz set nazv_analiz=@nazv_analiz where id_analis=@id_a
END

```

Рисунок 22 –Хранимая процедура по изменению таблица Название анализ

2.2 Создание графического интерфейса

Для работы программы аналитики медицинских анализов создан графический интерфейс на основе форм WindowsForms на языке С#.

С# — объектно-ориентированный язык программирования. Создан в 1998—2001 инженерами Microsoft во главе Андерса Хейлсберга и Скотта Вильтаумота, для разработки приложений Microsoft .NET Framework. Далее стандартизован ECMA-334 и ISO/IEC 23270.

Созданная программа позволяет создавать новые формы анализов, пациентов и вносить показатели.

(между тем допускается множественная реализация интерфейсов).



Рисунок 23– Стартовая страница

The image shows a Visual Studio code editor window with three tabs: 'start.cs', 'admin.cs [Конструктор]', and 'admin.cs'. The active tab is 'start.cs [Конструктор]'. The code is written in C# and is part of a project named 'Analitika'. The code defines a namespace 'Analitika' and a partial class 'start' that inherits from 'Form'. The class contains three event handlers: 'start()', 'button1_Click', and 'button2_Click'. Each handler creates and shows a new instance of a form (Form1, admin, or lab). The code is as follows:

```
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Analitika
12 {
13     public partial class start : Form
14     {
15         public start()
16         {
17             InitializeComponent();
18         }
19
20         private void button1_Click(object sender, EventArgs e)
21         {
22             admin newForm = new admin();
23             newForm.Show();
24         }
25
26         private void button2_Click(object sender, EventArgs e)
27         {
28             Form1 newForm1 = new Form1();
29             newForm1.Show();
30         }
31
32         private void button3_Click(object sender, EventArgs e)
33         {
34             lab newForm2 = new lab();
35             newForm2.Show();
36         }
37     }
38 }
```

Рисунок 24 –Кодстраница программы

На стартовой странице (Рисунок 23, Рисунок 24) мы создаем 3 роли для работы:

- Системный администратор
- Регистратура
- Лаборатория

Данные роли выбираются при помощи кнопах, при нажатии которых мы можем перейти на окна для ввода данных.

Интерфейс рабочего окна системного администратора (Рисунок 25).

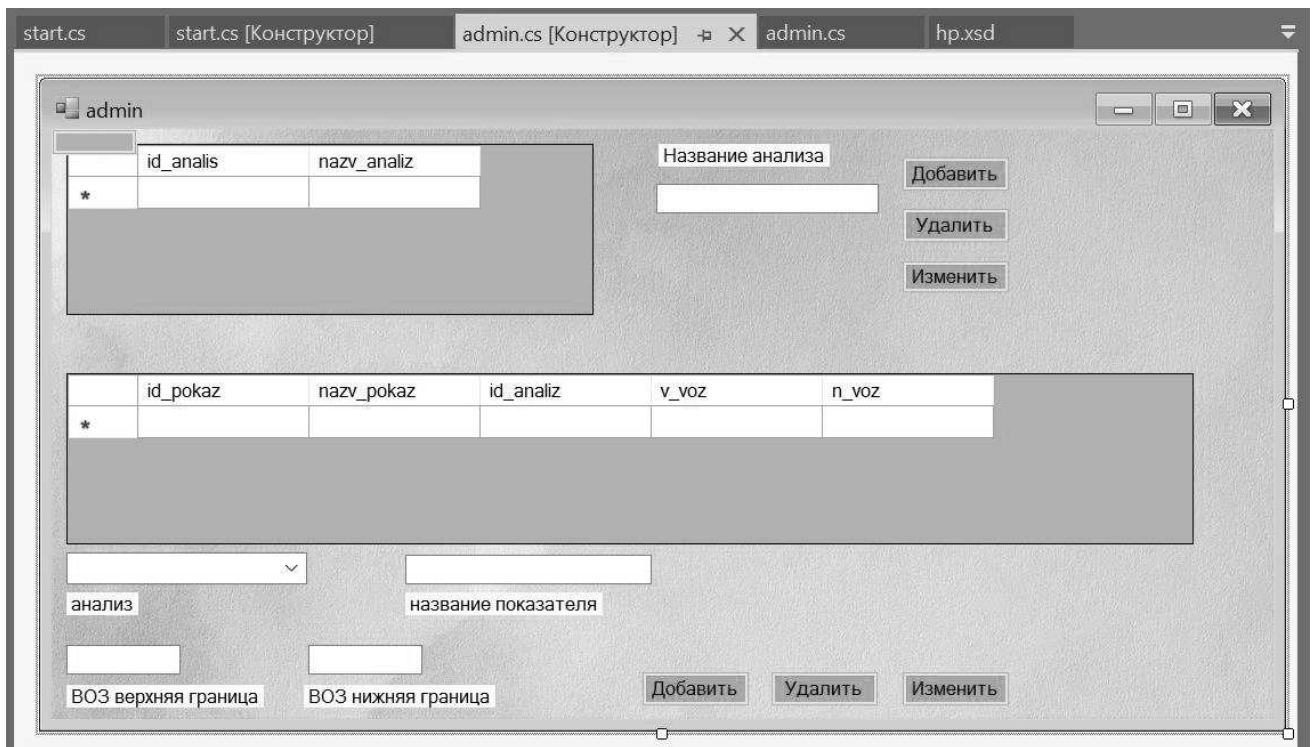


Рисунок 25– Интерфейс рабочего окна системного администратора

В рабочем процессе нам необходимо заполнять 2 таблицы, для этого мы добавили на форму 2 таблицы из базы данных:

- Analizc названиями анализа
- Pokazc названиями показателей и данными Всемирной организации здравоохранения

Для заполнения таблицы rokazv поле анализ мы добавили выпадающий список с названием показателя, для упрощения работы системного администратора.

Кнопки на форме работают при помощи вызова хранимой процедуры

- Добавить
- Удалить
- Изменить

```

18     }
19
20     ССЫЛКА:1
21     private void Form1_Load(object sender, EventArgs e)
22     {
23         // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.pokaz". При необходи
24         this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);
25         // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.analiz". При необход
26         this.analizTableAdapter.Fill(this.analitikabaza.analiz);
27     }
28
29     ССЫЛКА:1
30     private void button1_Click(object sender, EventArgs e)
31     {
32         this.analizTableAdapter.instanaliz(this.textBox4.Text);
33         this.analizTableAdapter.Fill(this.analitikabaza.analiz);
34     }
35     ССЫЛКА:1
36     private void label4_Click(object sender, EventArgs e)
37     {
38         admin form2 = new admin();
39         form2.Show();
40     }
41
42     ССЫЛКА:1
43     private void button2_Click(object sender, EventArgs e)
44     {
45         DataRowView ss;
46         int d;
47         ss = (DataRowView)this.analizBindingSource.Current;
48         d = (int)ss["id_analiz"];
49         this.analizTableAdapter.del_analiz(d);
50         this.analizTableAdapter.Fill(this.analitikabaza.analiz);
51     }

```

Рисунок 26 –Код окна системного администратора

Для разработки рабочего окна регистратуры мы добавили на форму таблицу из базы данных расient (Рисунок 27).

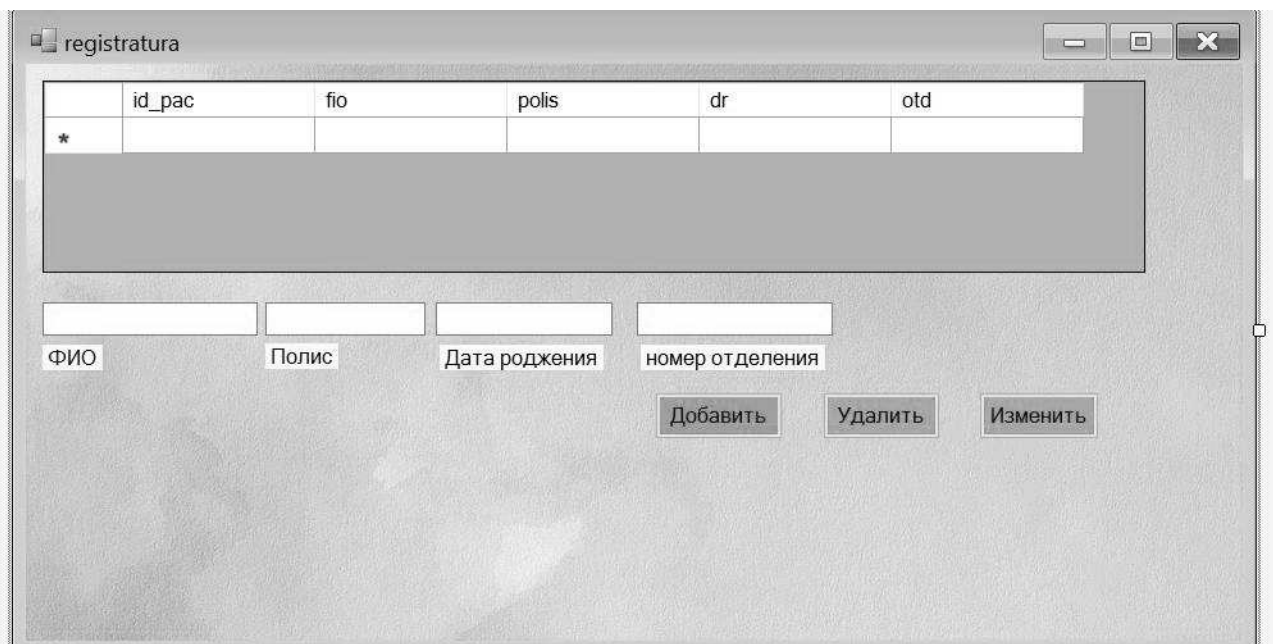


Рисунок 27– Рабочая форма регистратуры

В данной форме мы вводим пол фамилия имя отчество, полис, дата рождения, номер отделения.

Еще одна форма для заполнения базы данных – Лаборатория (Рисунок 28)

	id_itog	id_pac	id_pokaz	rez	data_pr
*					

Формы ввода:

- ФИО:
- Показатель:
- Дата:
- Результат:

Кнопки: Добавить, Удалить, Изменить

Рисунок 28 – Рабочая форма лаборатории

Лаборатория вводит данные выбирая из выпадающего списка фамилию отчество, и наименование показателя, и результат исследования и дату проведенного исследования.

Работа с базой данных происходит с помощью хранимых процедур, ниже приведен пример их использования для страницы и соответственно таблицы в Базы данных Analitika.

```
namespace Analitika
{
    public partial class admin : Form
    {
        public admin()
        {
            InitializeComponent();
        }
    }
}
```

```

    }

    privatevoid Form1_Load(object sender, EventArgs e)
    {
        // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.pokaz".
        При необходимости она может быть перемещена или удалена.
        this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);
        // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.analiz".
        При необходимости она может быть перемещена или удалена.
        this.analizTableAdapter.Fill(this.analitikabaza.analiz);

    }

    privatevoid button1_Click(object sender, EventArgs e)
    {
        this.analizTableAdapter.instanaliz(this.textBox4.Text);
        this.analizTableAdapter.Fill(this.analitikabaza.analiz);
    }

    privatevoid label4_Click(object sender, EventArgs e)
    {
        admin form2 = new admin();
        form2.Show();

    }

    privatevoid button2_Click(object sender, EventArgs e)
    {
        DataRowView ss;
        int d;
        ss = (DataRowView)this.analizBindingSource.Current;
        d = (int)ss["id_analiz"];
        this.analizTableAdapter.del_analiz(d);
        this.analizTableAdapter.Fill(this.analitikabaza.analiz);
    }

    privatevoid button3_Click(object sender, EventArgs e)
    {
        DataRowView ss;
        int d;
        ss = (DataRowView)this.analizBindingSource.Current;

```

```
        d = (int)ss["id_analiz"];
this.analizTableAdapter.up_analiz(Convert.ToString(this.textBox4.Text), d);
this.analizTableAdapter.Fill(this.analitikabaza.analiz);
}
}
```


III. ПОСТРОЕНИЕ OLAP-КУБА

Первым вариантом вывода данных был построение OLAP-куба и вывод данных.

OLAP-куб (On-Line Analytical Processing в переводе интерактивный анализ данных) многомерный массив данных, используемый в OLAP (технология обработки данных, подготавливающая суммарную информацию на основе больших структурированных, по многомерному принципу, массивов данных,).

Кубы данных OLAP позволяют эффективно извлекать и анализировать многомерные данные. В отличие от других типов баз данных, базы данных OLAP разработаны специально для аналитической обработки и быстрого извлечения из них всевозможных наборов данных. На самом деле существует несколько ключевых различий между стандартными реляционными базами данных, такими как Access или SQL Server, и базами данных OLAP.

Для успешного развертывания OLAP-куба необходимо наличие следующих компонентов сервера (Рисунок 29):

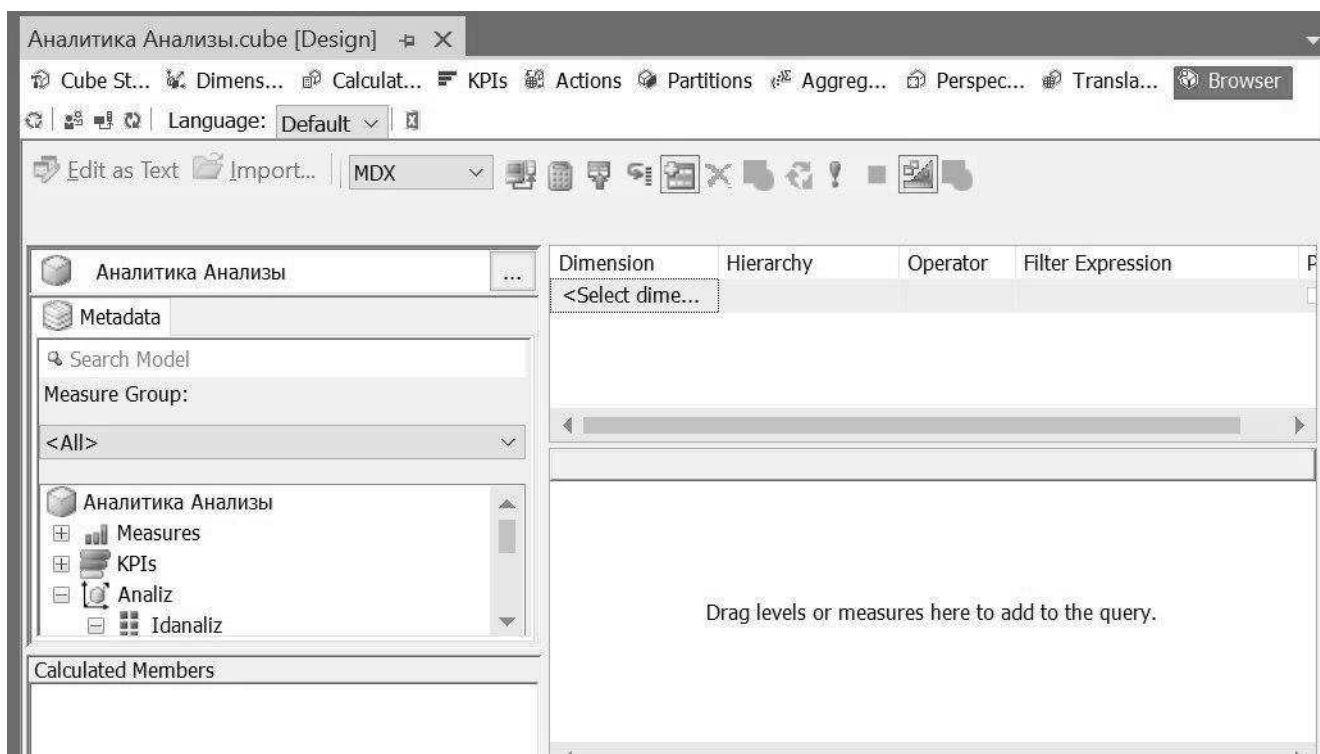


Рисунок 29 – Набор компонентов

Построение куба выполняется в среде Data Tools for MS Visual Studio 2019. Для выбранной структуры данных («снежинка») мастер автоматически находит таблицу фактов и все связанные с ней таблицы. Итоговая структура решения приведена на рисунке 30

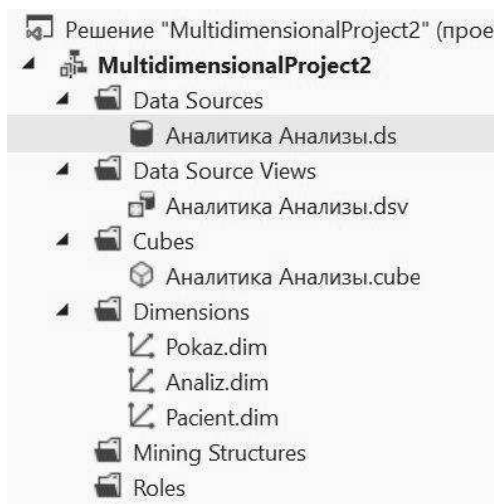


Рисунок30 – Итоговая структура решения

При сборке получаем куб и появляется возможность брать данные и выводить из куба (Рисунок 31)

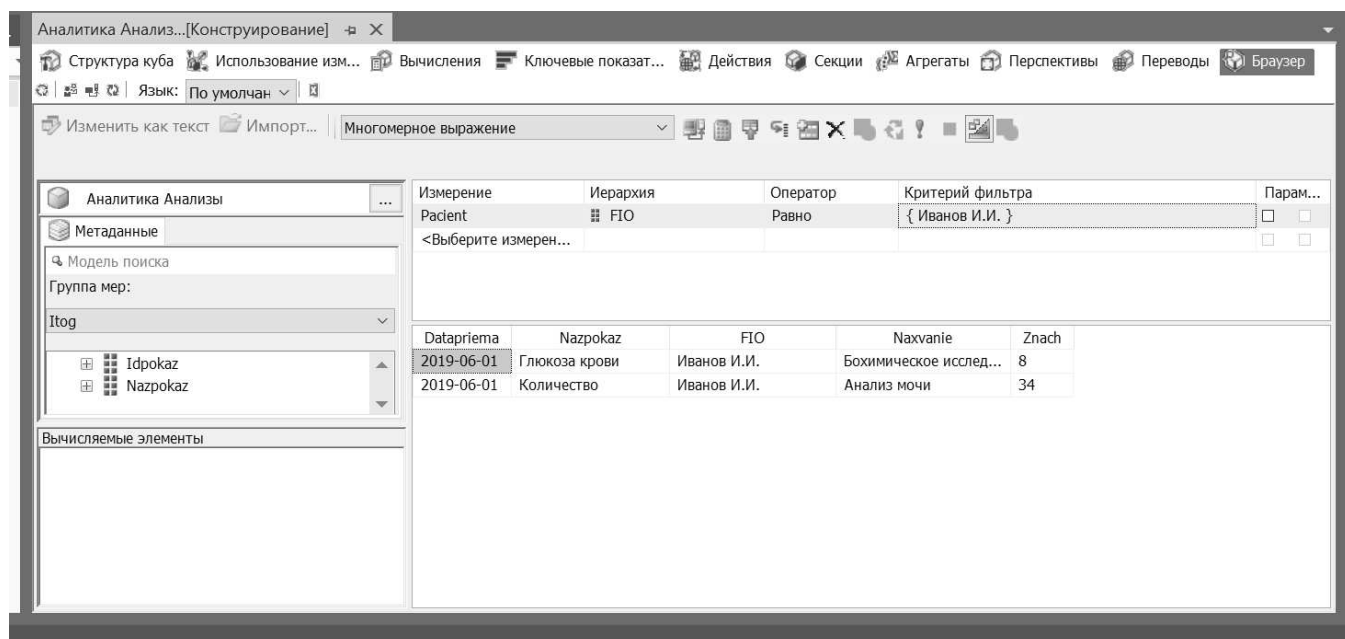


Рисунок 31–Вывод данных из Куба

После реализации первого этапа проекта, он был представлен в ГБУЗ «ОКСПНБ №1» и после тестового периода, по коллегиальным обсуждениям было принято решение реализовать вывод данных в более привычном интерфейсе для врачебного персонала, для этого была выбрана платформа MSExcel.

IV. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ВЫВОДА ДАННЫХ В MSEXCEL

Microsoft Excel (также иногда называется Microsoft Office Excel[5]) — программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone. Она предоставляет возможности экономико–статистических расчетов, графические инструменты и, за исключением Excel 2008 под Mac OS X, язык макропрограммирования VBA (Visual Basic for Application). Microsoft Excel входит в состав Microsoft Office.

Для реализации интерфейса в MSExcel, была подключена база данных из MS SQL Server 2019 (Рисунок 32).

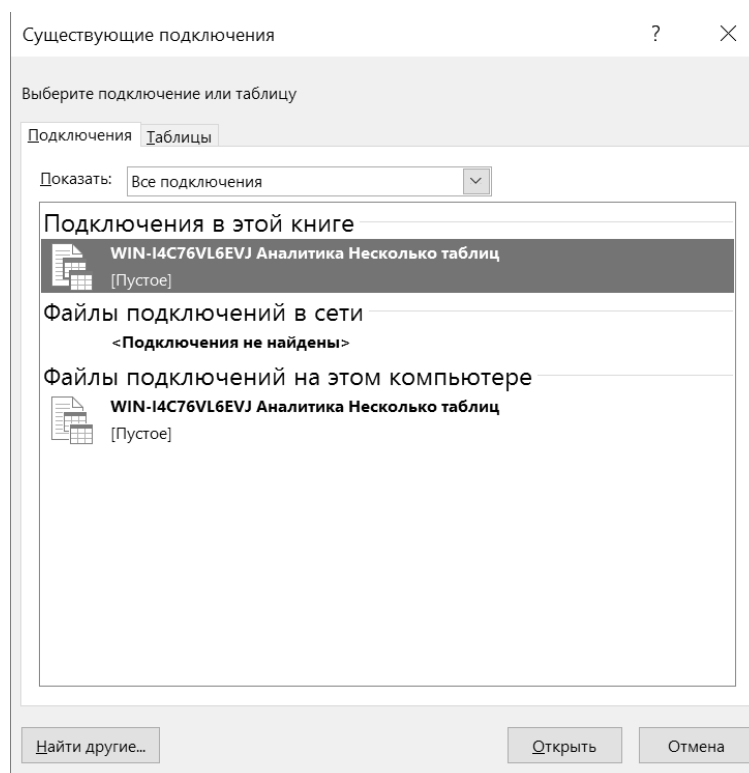


Рисунок 32–Подключенная база данных

Для более дружелюбного интерфейса мы подгружаем в итоговую таблицу (Рисунок 33) данные пациента, название анализа и названия показателей с помощью функции ВПР, которая дает нам возможность преобразовать данные из столбца id_расв фамилию имя отчество из таблицы расient.

id_itog	id_pac	id_pokaz	ФИО	Анализ	Дата	Показатель	rez	data_pr	НИЖНЯЯ ГРАНИ
5	7	6	Касаткина Екатерина Михайловна	ОАК	2021-02-01	Средний уровень HGB в эритроците (пг)	23	2021-02-01	
22	7	6	Касаткина Екатерина Михайловна	ОАК	2021-02-02	Средний уровень HGB в эритроците (пг)	31	2021-02-02	
23	7	6	Касаткина Екатерина Михайловна	ОАК	2021-02-03	Средний уровень HGB в эритроците (пг)	32	2021-02-03	

Рисунок 33–Итоговая таблица с результатами

Создаем сводную диаграмму с возможностями фильтрации для оперативной работы врача (Рисунок 34)

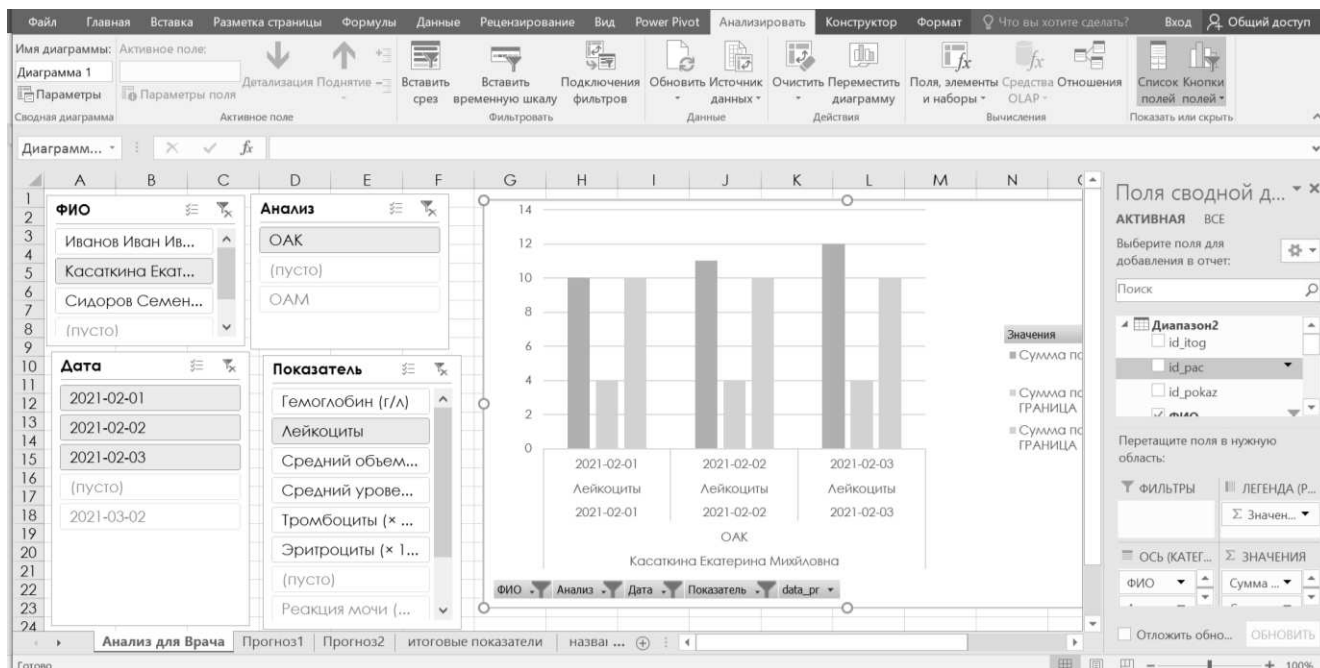


Рисунок 34 –Создание сводной диаграммы

Данная диаграмма поможет врачу проанализировать показатели в динамике, она сможет выбрать фамилию пациента название анализа показатель и диапазон дат который его интересует, автоматический подгрузиться диаграмма с показателями данных Всемирной организации здравоохранения, что позволит врачу проанализировать, насколько данные отходят от норма, и в какой динамике это происходит.

Авторитетность Всемирной организации здравоохранения находится на высоком уровне, так как в своей работе нацелена на поддержку, а также на усиление здоровья граждан как наиболее важную сферу граждан. Отслеживая действия государств в обеспечивании граней с целью сбережения также поддержания самочувствия собственных людей, специалисты проработали

концепцию оценок, позволяющую осуществлять постоянное наблюдение работы здравоохранения во всех государствах общества. Итоги данной непростой умозаключительной работы постоянно делаются преимущественно интернационального общества посредством публикуемые годовые сведения Всемирной организации здравоохранения. Так как не возможно знать наизусть все показатели и их нормы, то в этом случае нам на помощь приходят информационные системы, для автоматизации работы медицинского персонала и во избежании ошибок.

Создаем прогноз для врачей чтобы появилась возможность анализировать результаты медицинских исследований в MS Excel.

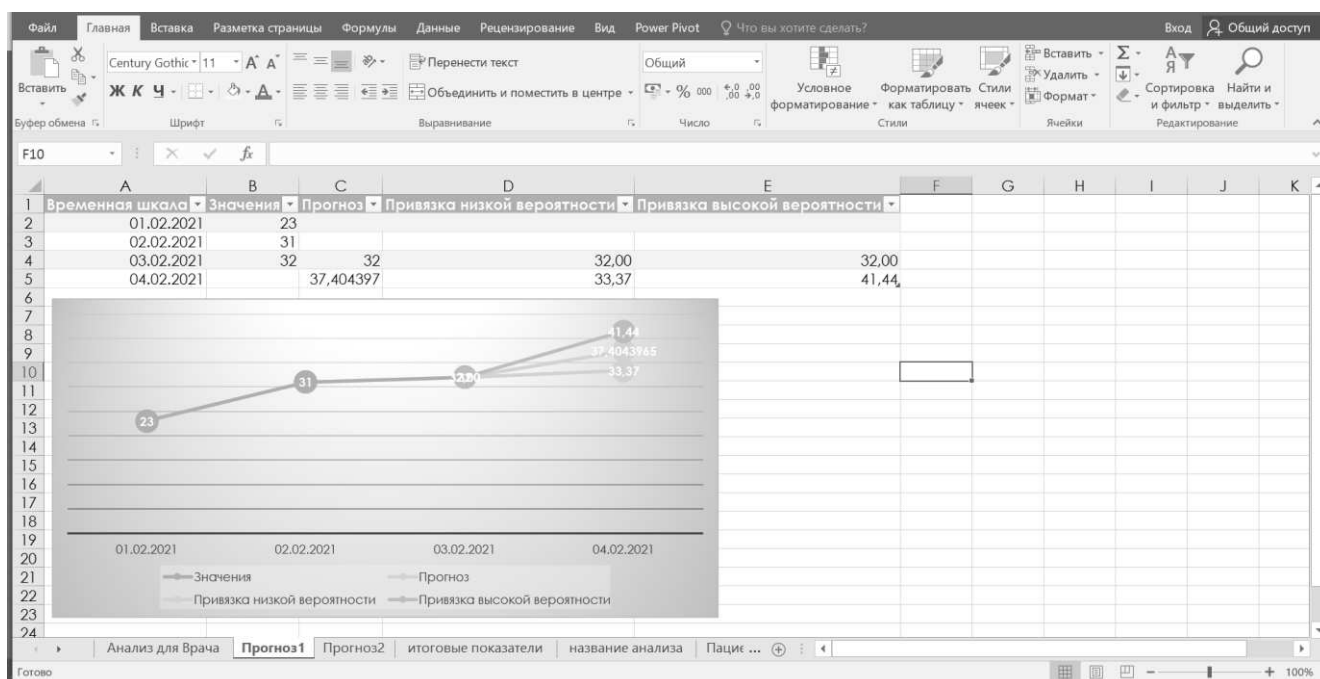


Рисунок 35—Прогноз результатов медицинских исследований на 1 день в перед

Данный прогноз поможет врачу для понимания куда ведут результаты и что будет на 1 день в перед, дольше прогноз делать не эффективно, так как, на медицинские исследования влияет много факторов.

Биологические факторы оказывающие влияние на результаты медицинских исследований

Физиологические факторы влияющие на медицинские исследования:

- Раса – разные расы по-разному реагируют на лекарственную терапию

- Пол – у каждого пола свои особенности организма

- Возраст

- Тип телосложения

- Питание пациента до поступления и после поступления, многие медицинские показатели выравниваются с нормализованием питания.

Окружающая среда не малую роль так же играет на исследования пациентов, после поступления могут резко измениться показатели (климат, время года, состав воды и почвы в зоне проживания, социально-бытовые условия проживания).

Фактор воздействия профессиональных и бытовых токсичных средств (алкогольные вещества, наркотические вещества).

Фактор метода взятия крови (венепункция, вакутейнеры);

Фактор условия хранения и транспортировки из отделения в лабораторию (температура, влияние света, времени транспортировки, встряхивания).

Для прогнозирования результата медицинского исследования функция ПРЕДСКАЗ в Excel позволяет с некоторой точностью предсказать значения на основе выполненных результатов медицинских исследований, и возвращает возможное значение результата медицинских исследований. Наши данные набираются с течением времени. Изменения медицинских исследований могут быть зафиксированы опытным путем. С помощью функции ПРЕДСКАЗ можно предположить следующий результат.

V. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ В ГБУЗ «ОКСПНБ №1»

5.1 Подготовка к внедрению

В Государственном бюджетном учреждении здравоохранения Челябинской области «Областная клиническая психоневрологическая больница №1» было предложено протестировать данное программное обеспечение для работы 23 инфекционного отделения, под контролем Министерства здравоохранения Челябинской области.

Перед внедрением информационной системы был поднят вопрос по информационной безопасности в соответствии с законами о защите информации и персональных данных:

- Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 N 149–ФЗ
 - Федеральный закон «О персональных данных» от 27.07.2006 N 152–ФЗ
- Приказ ФСБ России от 10 июля 2014 г. N 378 «Об утверждении Составы и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств криптографической защиты информации, необходимых для выполнения установленных Правительством Российской Федерации требований к защите персональных данных для каждого из уровней защищенности»
- Постановление Правительства РФ от 1 ноября 2012 г. N 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных»
- Постановление Правительства РФ от 06.07.2015 N 676. «О требованиях к порядку создания, развития, ввода в эксплуатацию, эксплуатации и вывода из эксплуатации государственных информационных систем и дальнейшего хранения содержащейся в их базах данных информации»

- Приказ ФСТЭК России от 11.02.2013 N 17 «Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах»

- Приказ ФСТЭК России от 18.02.2013 N 21 «Об утверждении Составы и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных»

Для реализации требования выполнения нормативно правовых актов по информационной безопасности и защите персональных данных, так как тестирование данной системы проходит на реальных данных пациентов, то программное обеспечение и база данных развернуты на одном автоматизированном рабочем месте и были приняты следующие меры по защите информации:

- Подготовлено отдельное автоматизированное рабочее место
- Вход в доменную систему настроен через электронную подпись, полученную в Удостоверяющем центре ГБУЗ «ЧОМИАЦ», настроена групповая политика по защите файлов на данном рабочем месте (автоматический выход при отключении электронной подписи)

- Программное обеспечение по антивирусной защите и средства криптозащите (КриптоПро, Secret Net)

- ViPNet Coordinator 4 Программный шлюз безопасности (контроль доступа пользователей из публичной сети к предоставляемым организацией ресурсам и сервисам, разграничение доступа к информации в локальной сети, защита сетей)

- Инструктаж с специалистами по защите персональных данных и работе с электронными подписями.

5.2 План внедрения

Были разработаны этапы внедрения, где заказчиком выступает ГБУЗ «ОКСПНБ №1» и Министерство здравоохранения Челябинской области.

Таблица 3. План внедрения.

п/н	Наименование работ	Срок реализации	Ответственная сторона
1	Направление на согласование полей базы данных	14.04.2020 г.	Исполнитель
2	Согласование полей базы данных	21.04.2020 г.	Заказчик
3	Проведение настройки, доработки полейсоздание базы данных, хранимых процедур	21.05.2020 г.	Исполнитель
4	Доработка системы ввода данных под новую базу данных Заказчика,	21.05.2020 г.	Заказчик
5	Направление на согласование интерфейса для ввода данных	25.05.2020	Исполнитель
6	Согласование интерфейса для ввода данных	25.06.2020	Заказчик
7	Проведение инструктажа по заполнению базы данных ответственных исполнителей	25.07.2020	Исполнитель
8	Тестовая работа в системе сотрудников Заказчика	25.08.2020	Заказчик
9	Консультации пользователей	25.08.2020	Исполнитель
10	Согласование интерфейса для вывода данных	01.10.2020	Заказчик
11	Доработка системы вывода данных под пользователей Заказчика	01.02.2021	Исполнитель
12	Проведение инструктажа по выводу данных для ответственных исполнителей	01.03.2021	Исполнитель
13	Тестовая работа в системе сотрудников Заказчика	23.04.2021	Заказчик
14	Консультации пользователей	23.04.2021	Исполнитель
15	Написание рекомендательных писем по результатам внедрения	26.04.2021 г.	Исполнитель
Итого		1 год	

5.3 Процесс внедрения

В процессе внедрения были произведены доработки системы.

Первой доработкой системы было изменение базы данных, первоначальная база данных в своей структуре 6 таблиц (Рисунок 36)

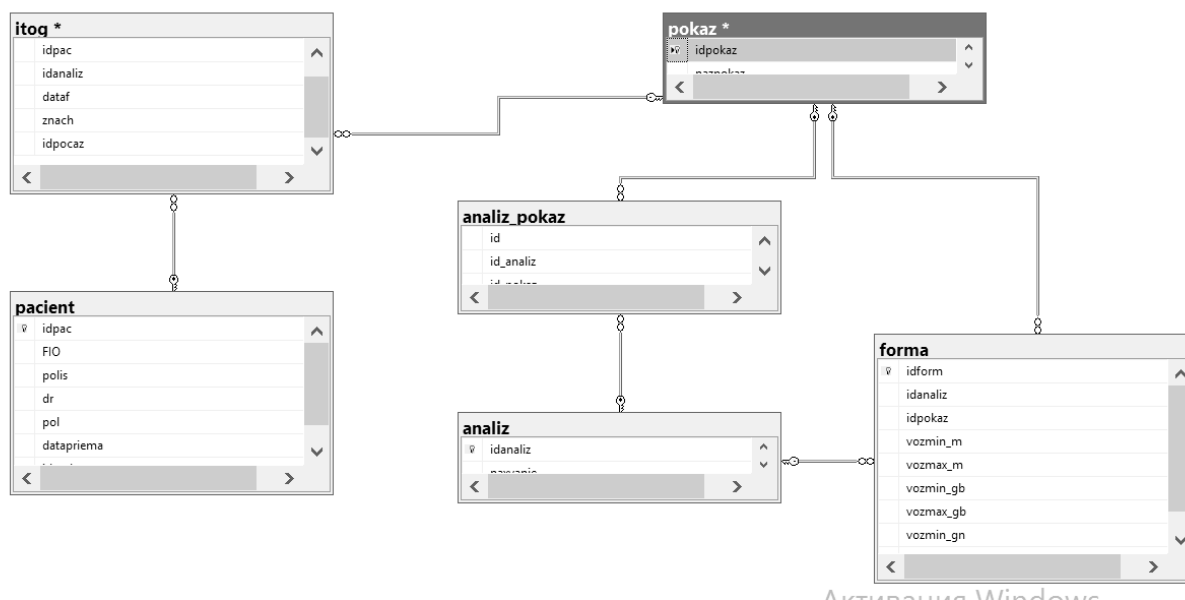


Рисунок 36– Структура первой базы данных

Нормы показателей Всемирной организации здравоохранения вводились в таблице forma и по разделению на пол и возврат, после коллективного решения было решено что данные воз не нужно подразделять на пол и возраст, так как от пола показатели анализов в данном медицинском учреждении не сильно отличаются, а пациенты в 23 инфекционном отделении лежат только старше 18 лет, по этой причине нет необходимости в подразделении на возраст. Ввести данные нормы было бы оптимальные в таблица с названием analiz и в этой же таблице имеет смысл делать ссылку на название анализа, так как название в разных медицинских показателях могут совпадать но при этом нормы будут отличаться. Для работы системного администратор более рационально заполнять 2 таблица а не 4. Итоговая структура базы данных получилась из 4 таблиц (Рисунок 37)



Рисунок 37– Итоговая база данных

После доработки системы был проведен инструктаж работы в информационной системе, для занесения базы данных медицинских исследований, для этого была проведена работа совместно с заведующим отделения 23 и заведующим клинической лабораторией для работы в системе были выбраны 3 медицинских исследования на которые на данный период времени осваивается контроль состояния пациента в отделении.

Общий анализ крови помогает определить концентрацию основных форменных элементов крови:

- Эритроцитов (составляют 40% объема крови. Основная функция — это транспорт химического элемента кислород из легких по всему организму)
- Лейкоцитов (Клетки иммунной системы, отвечающие за антимикробную защиту, а также указывают на воспалительные процессы и аллергические реакции)
- Тромбоцитов (Клетки крови, участвующие в формировании тромбов)

- Гемоглобин (железосодержащий белок. Позволяет эритроцитам связываться с кислородом в легких и переносить его по всему телу)
- Гемокрит (это отношение объема красных кровяных телец к общему объему цельной крови)
- Эритроцитарный индексов (Являются важным показателем для диагностики анемий и контроля эффективности терапии.)

Биохимический анализ крови развернутое медицинское лабораторное исследование венозной крови, для оценки работы внутренних органов пациента выявляет :

- Дефицит витаминов,
- Дефицит ферментов,
- Дефицит макро– и микроэлементов,
- Диагностики патологии обмена веществ

В Биохимический анализ крови входят такие показатели как:

- Глюкоза (Для диагностики и контроля сахарного диабета и других заболеваний, связанных с нарушением обмена углеводов)
- Белки и аминокислоты (Белковый обмен, отражающим содержание фракций белков в крови)
- Белки и аминокислоты (Диагностика состояния, сопровождающегося аномальным синтезом или потерей белка)
- Желчные пигменты и кислоты(Уровень билирубина в крови используют для выявления поражений печени, закупорки желчных путей, гемолитической анемии)
- Желчные пигменты и кислоты (Концентрация прямого билирубина в крови для диагностике заболеваний, сопровождающихся желтухой)
- Липиды (Оценку уровня холестерина в крови используют)
- Ферменты (для диагностики и контроле течения болезней печени, патологии печени и желчевыводящих путей костной ткани)

- Маркеры функции почек (для диагностики и контроле лечения состояния почек)

- Неорганические вещества/электролиты (для выявления нарушений электролитного баланса и в контроле лечения.)

Общий анализ мочи еще один распространённый вид лабораторных медицинских исследований для диагностики патологических процессов. Основной задачей анализа является выявление заболеваний мочевыделительной системы.

- Кислотность мочи рН (Диагностика почечной недостаточности, туберкулеза, обезвоживании, гипокалиемии, гиперкалиемии, инфекциях мочевыделительной системы, сахарного диабета, пищевого отравления.

- Удельный вес (Диагностика отеков; нефротический синдром, несхарный диабет, повышение показателей артериального давления, обусловленное злокачественными опухолями)

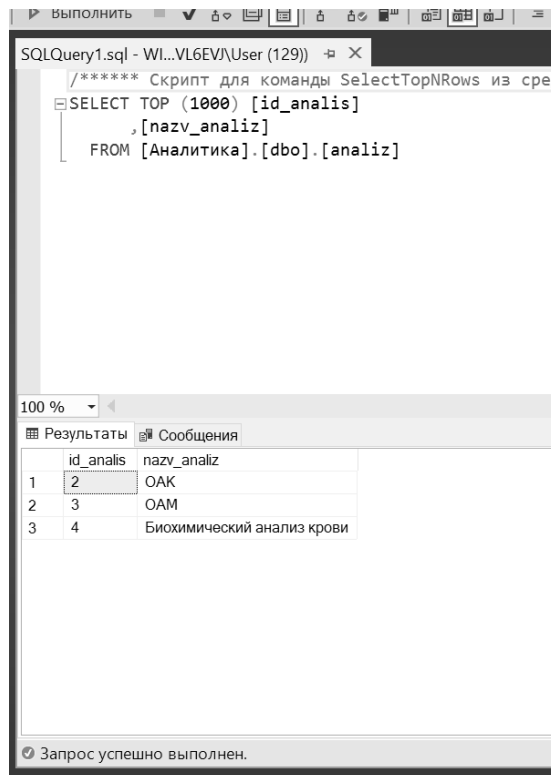


Рисунок 38– Таблица названий медицинских исследований в базе данных

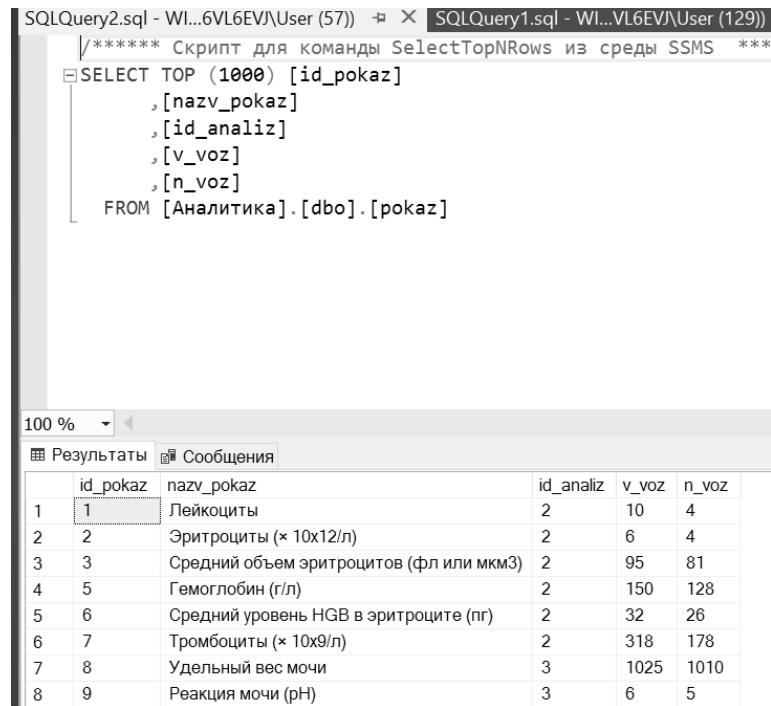


Рисунок 39– Таблица с показателями в базе данных

Данные медицинские анализы были введены в базу данных системы администратором с показателями данных Всемирной организацией здравоохранения (Рисунок 38, рисунок 39), в базу были внесены пациенты, находящиеся в 23 отделении на стационарном лечении.

Лаборатория начала вносить результаты исследования исследование. При работе в системе у пользователей вводящих данные в базу данных не возникло пререканий, проводились консультации, пользователи научились работать в системе.

Первый вариант вывода данных был представлен в виде OLAP–куба (Рисунок 40)

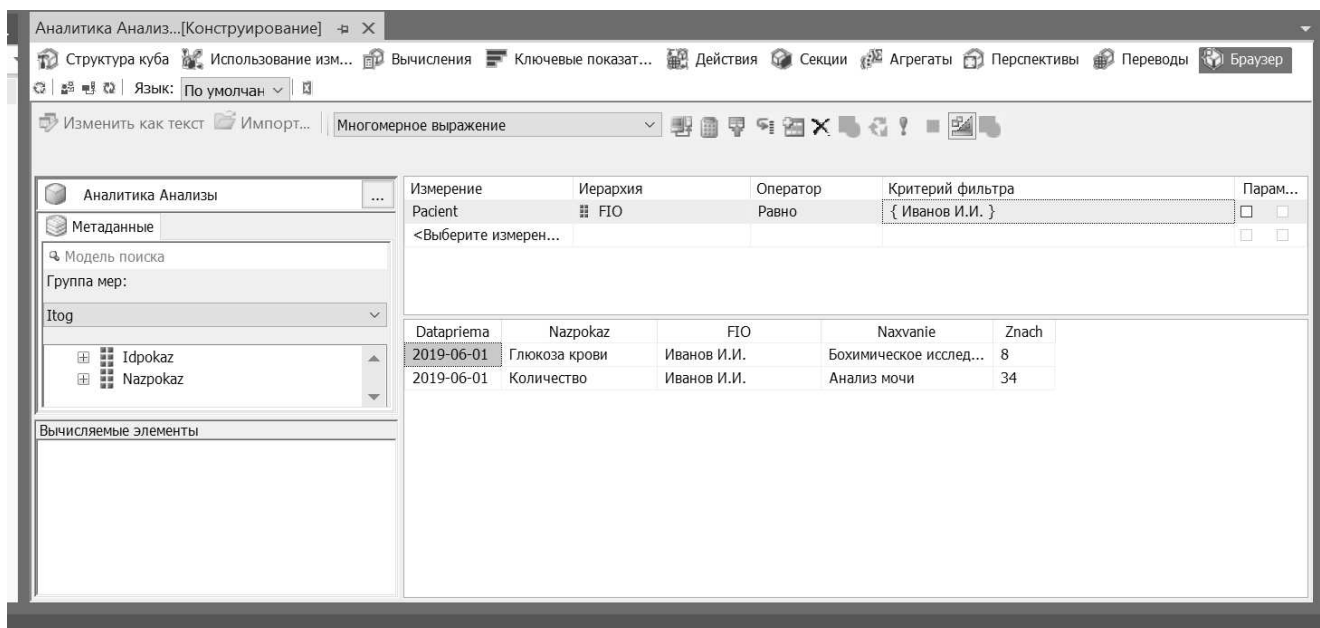


Рисунок 40– вывод данных в виде OLAP–куба

Данный интерфейс для конечных пользователей показался не приемлемый, в ходе изучения знаний пользователей в работе различного программного обеспечения было принято решение разработать вывод в MSExcel так как данный интерфейс знаком хорошо, работа с данным программным обеспечением проходит на регулярной основе. Было реализовано новое программное решение для вывода данных (Рисунок 41, Рисунок 42).

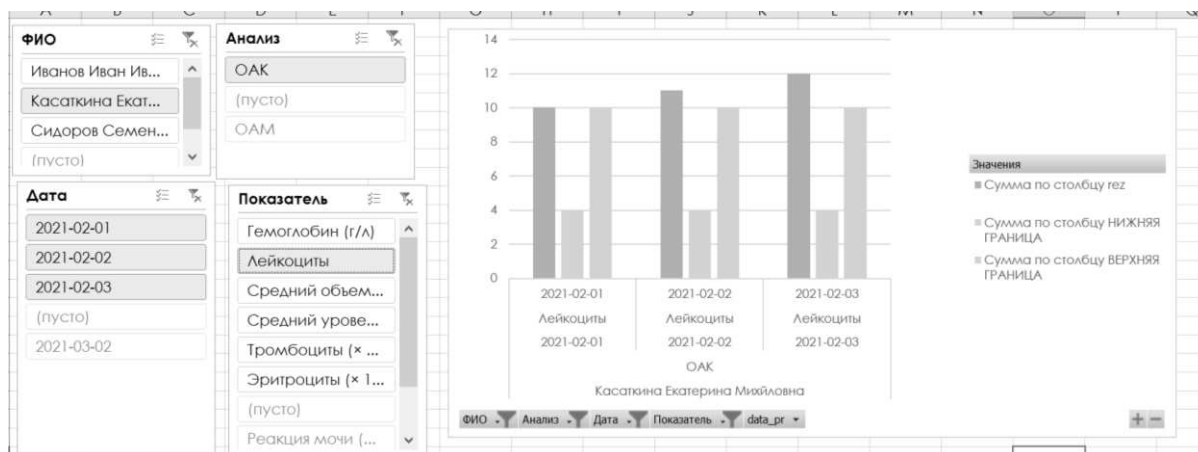


Рисунок 41– Аналитика медицинских исследований

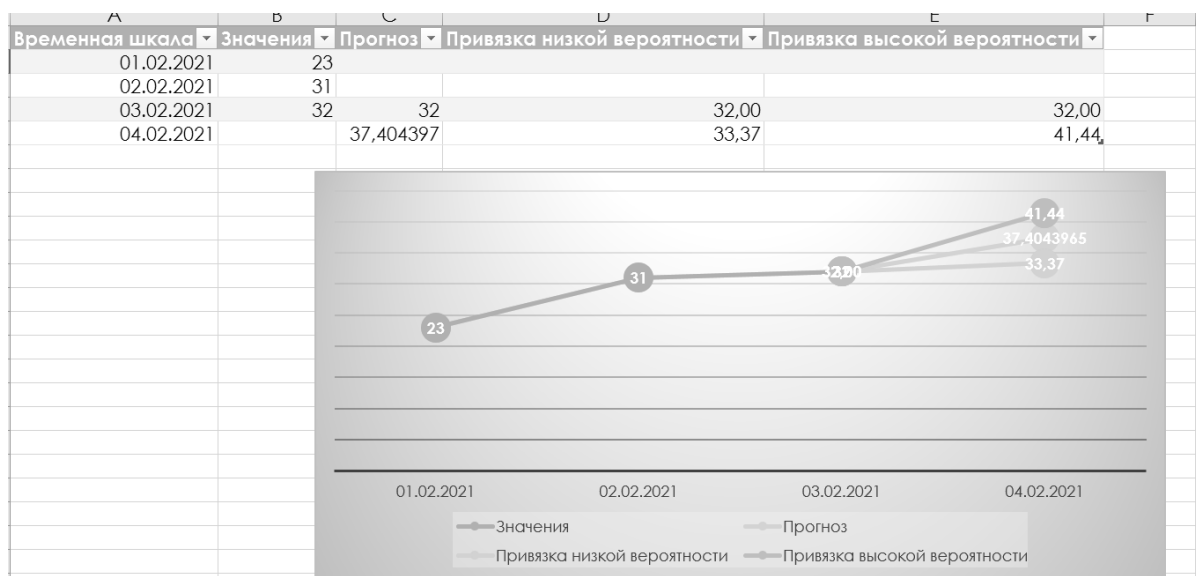


Рисунок 42– Прогнозирование результата

Пользователям стало на много удобнее и понятнее работать с данным программным решение, и анализировать показатели медицинских исследований.

5.3 Документооборот в процессе внедрения

В процессе внедрения было подготовлено ряд документов:

1. Календарный план внедрения программного обеспечения по «Аналізу динамики медицинских анализов». (Приложение А)
2. Лист ознакомления с инструкциями (Приложение Б)
3. Журнал фиксации консультации пользователей во время тестового периода (Приложение В)

По результатам внедрения было проведено совещание с представителями Государственного бюджетного учреждения здравоохранения «Областной клинической психоневрологической больницей №1», участвующих во внедрении и Министерством здравоохранения Челябинской области и принято решение по положительному эффекту работы данного продукта по результатам чего, были написаны рекомендательные письма.

ЗАКЛЮЧЕНИЕ

Информационной основой исследования были выбраны медицинские программные продукты, а именно «Барс МИС», «Медик+».

Актуальность разработки архитектуры программного продукта обусловлены тем, что рассмотренные сервисы легки в освоении, каждый из них достоин внимания, но динамики более развернутой по показателям медицинских исследований в данных программных продуктах не представлено.

Исходя из необходимости модернизации работы медицинской организации по диагностике, контролю хода лечения и восстановлению здоровья пациентов, разработано программное обеспечение по «Аналізу динамики медицинских анализов».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. <https://www.kp.ru/guide/analiz-krovi.html>
2. <https://www.7771000.ru/faq/edinyy-meditsinskiy-tsentr/chto-takoe-klinicheskie-ili-obshchie-analizi-dlya-chego-oni-nuzhny-skolko-eto-stoit-101/>
3. <https://belinnovation.by/projects/avtomatizirovannaya-informatsionnaya-sistema-medik/>
4. <https://bars.group/>
5. Эльянов М. М. Медицинские информационные технологии. Каталог. Вып. 5. М.: Третья медицина, 2005. – 320 с.
6. Назаренко Г. И., Гулиев Я. И., Ермаков Д. Е. Медицинские информационные системы: теория и практика. М.: Физматлит, 2005. – 320 с.
7. Гусев А. В., Романов Ф. А., Дуданов И. П., Воронин А. В. Медицинские информационные системы. Петрозаводск: Изд-во ПетрГУ, 2005. – 404 с.
8. Рот Г. З., Фихман М. И., Шульман Е. И. Медицинские информационные системы. Учебное пособие. Новосибирск: Изд-во НГТУ, 2005. – 70 с.
9. Красильников И. А., Усеинов Э. Р. Ресурсы информационных технологий в системе здравоохранения Санкт-Петербурга // Информационные технологии в здравоохранении: Доклады VI Санкт-Петербургской международной конференции "Региональная информатика 98". СПб., 1998. С. 70–72 (доступ в Интернете: <http://www.ctmed.ru/InfoServ/MedSci/health/index.html>)
10. Эльянов М. М. Медицинские информационные технологии: цивилизованный рынок или "зоопарк" // Информационные технологии в медицине–2002: Сборник тезисов. М.: ВК ВВЦ "Наука и образование", 2002. С. 54–58.

ПРИЛОЖЕНИЕ А

Календарный план внедрения программного обеспечения по «Аналізу динамики медицинских анализов»

п/н	Наименование работ	Срок реализации	Ответственная сторона
1	Направление на согласование полей базы данных	14.05.2020 г.	Исполнитель
2	Согласование полей базы данных	21.05.2020 г.	Заказчик
3	Проведение настройки, доработки полей создание базы данных, хранимых процедур	21.06.2020 г.	Исполнитель
4	Доработка системы ввода данных под новую базу данных Заказчика,	21.06.2020 г.	Заказчик
5	Направление на согласование интерфейса для ввода данных	25.06.2020	Исполнитель
6	Согласование интерфейса для ввода данных	25.07.2020	Заказчик
7	Проведение инструктажа по заполнению базы данных ответственных исполнителей	25.08.2020	Исполнитель
8	Тестовая работа в системе сотрудников Заказчика	25.09.2020	Заказчик
9	Консультации пользователей	25.09.2020	Исполнитель
10	Согласование интерфейса для вывода данных	01.11.2020	Заказчик

Окончание приложения номер А

11	Доработка системы вывода данных под пользователей Заказчика	01.03.2021	Исполнитель
12	Проведение инструктажа по выводу данных для ответственных исполнителей	01.04.2021	Исполнитель
13	Тестовая работа в системе сотрудников Заказчика	23.05.2021	Заказчик
14	Консультации пользователей	23.05.2021	Исполнитель
Итого		1 год	

ПРИЛОЖЕНИЕ Б

Лист ознакомления с инструкциями по работе в программном обеспечении по
«Аналізу динаміки
медичинських аналізів» в ГБУЗ «ОКСПНБ №1»

п/н	ФІО	Должность	Дата	Наименование инструкции	Подпись
1		Системный администратор		Инструктаж системного администратора	
2		Медицинский регистратор		Инструктаж медицинского регистратора	
3		Лаборант		Инструктаж сотрудника лаборатории	
4		Заведующий лабораторией		Инструктаж сотрудника лаборатории	
5		Заведующий отделением		Инструктаж врача	
6		Врач		Инструктаж врача	
7		Врач		Инструктаж врача	
8		Врач		Инструктаж врача	

ПРИЛОЖЕНИЕ В

Журнал фиксации консультацией пользователей во время тестового периода использования программного обеспечения по «Анализу динамики медицинских анализов» в ГБУЗ «ОКСПНБ №1»

п/н	Вопрос обращения	Тип обращения	Дата
1		Консультация	
2		Запрос на изменение	
3			
4			

ПРИЛОЖЕНИЕ Г

Хранимые процедуры

Хранимая процедура по удалению анализа

```
USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[del_analiz]
*****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[del_analiz]
    @id1 int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SETNOCOUNT ON;

delete [dbo].[analiz] where id_analis = @id1
END
```

Хранимая процедура по удалению результата медицинского исследования

```
USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[delitog]
*****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[delitog]
    @id2 int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SETNOCOUNT ON;

delete [dbo].[itogi] where id_itog = @id2
END
```

Хранимая процедура по удалению пациента

```
USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[delpac]
*****/
SETANSI_NULLSON
GO
```



```

SETQUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[delpac]
    @id int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    delete [dbo].[pacient] where id_pac = @id
END

```

Хранимая процедура по удалению показателя

```

USE [Аналитика]
GO
/***** Object:  StoredProcedure [dbo].[delpokaz]
SETANSI_NULLS ON
GO
SETQUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[delpokaz]
    @id3 int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    delete [dbo].[pokaz] where id_pokaz = @id3
END

```

Хранимая процедура по добавлению результата

```

USE [Аналитика]
GO
/***** Object:  StoredProcedure [dbo].[inaitogi]
SETANSI_NULLS ON
GO
SETQUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[inaitogi]
    @id_pac int,
    @id_pokaz int,
    @rez int,

```

```
@data_prdate
```

```
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SETNOCOUNTON;
    insertinto itogi(id_pac, id_pokaz, rez, data_pr)
values (@id_pac, @id_pokaz, @rez, @data_pr)
END
```

Хранимая процедура по добавлению пациента

```
USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[inspacient] *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:     <Description,,>
-- =====
ALTERPROCEDURE [dbo].[inspacient]

    @fio nvarchar(50),
    @polis int,
    @dr date,
    @otd int
```

```
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SETNOCOUNTON;
    insertinto pacient(fio, polis, dr, otd)
values (@fio, @polis, @dr, @otd)
END
```

Хранимая процедура по добавлению анализа

```
USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[instanaliz] *****/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:     <Description,,>
-- =====
ALTERPROCEDURE [dbo].[instanaliz]

    @nazv_analiz nvarchar(50)
```

```

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    insert into analiz(nazv_analiz)
values (@nazv_analiz)
END

```

Хранимая процедура по добавлению показателя

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[instpokaz]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER PROCEDURE [dbo].[instpokaz]
    @nazv_pokaz nvarchar(50),
    @id_analiz int,
    @v_voz int,
    @n_voz int

```

```

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    insert into pokaz(nazv_pokaz, id_analiz, v_voz, n_voz)
values (@nazv_pokaz, @id_analiz,@v_voz,@n_voz)
END

```

Хранимая процедура по изменению анализа

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[up_analiz]    SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER PROCEDURE [dbo].[up_analiz]
    @nazv_analiz nvarchar(50),
    @id_a int

```

```

AS
BEGIN

```

```

-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SETNOCOUNTON;
update analiz set nazv_analiz=@nazv_analiz where id_analiz=@id_a
END

```

Хранимая процедура по изменению результата

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[upitog]
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[upitog]
    @id_pac int,
    @id_pokaz int,
    @rez int,
    @id_itog int,
    @data_pr date
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SETNOCOUNTON;
update itogi set id_pac=@id_pac, id_pokaz=@id_pokaz, rez=@rez, data_pr=@data_pr where
id_itog=@id_itog
END

```

Хранимая процедура по изменению пациента

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[uppacient]
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[uppacient]
    @fio varchar(50),
    @polis int,
    @dr date,
    @otd int,
    @id_pac int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.

```

```

SETNOCOUNTON;
update pacient set fio=@fio, polis=@polis, dr=@dr, otd=@otd where id_pac=@id_pac
END

```

Хранимая процедура по изменению показателей

```

USE [Аналитика]
GO
/***** Object: StoredProcedure [dbo].[uppocaz]
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTERPROCEDURE [dbo].[uppocaz]
    @nazv_pokaz nvarchar(50),
    @id_analiz int,
    @v_voz int,
    @in_voz int,
    @id_pokaz int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SETNOCOUNTON;
update pokaz set nazv_pokaz=@nazv_pokaz, id_analiz=@id_analiz, v_voz=@v_voz, n_voz=@in_voz
where id_pokaz=@id_pokaz
END

```

ПРИЛОЖЕНИЕ Д

Код стартовой формы

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Analitika
{
    public partial class start : Form
    {
        public start()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            admin newForm = new admin();
            newForm.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form1 newForm1 = new Form1();
            newForm1.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            lab newForm2 = new lab();
            newForm2.Show();
        }
    }
}

namespace Analitika
{
    partial class start
    {
        ///<summary>
        /// Required designer variable.
        ///</summary>
        private System.ComponentModel.IContainer components = null;

        ///<summary>
        /// Clean up any resources being used.
        ///</summary>
        ///<param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
    }
}
```

```

protectedoverridevoid Dispose(bool disposing)
    {
    if (disposing && (components != null))
        {
            components.Dispose();
        }
    base.Dispose(disposing);
    }

#region Windows Form Designer generated code

///<summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
///</summary>
privatevoid InitializeComponent()
    {
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(start));
this.button1 = new System.Windows.Forms.Button();
this.button2 = new System.Windows.Forms.Button();
this.button3 = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// button1
//
this.button1.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("button1.BackgroundImage")));
this.button1.Location = new System.Drawing.Point(12, 12);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(341, 299);
this.button1.TabIndex = 3;
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// button2
//
this.button2.Image = ((System.Drawing.Image)(resources.GetObject("button2.Image")));
this.button2.Location = new System.Drawing.Point(369, 12);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(303, 299);
this.button2.TabIndex = 4;
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new System.EventHandler(this.button2_Click);
//
// button3
//
this.button3.Image = ((System.Drawing.Image)(resources.GetObject("button3.Image")));
this.button3.Location = new System.Drawing.Point(692, 12);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(323, 299);
this.button3.TabIndex = 5;
this.button3.UseVisualStyleBackColor = true;
this.button3.Click += new System.EventHandler(this.button3_Click);
//
// start

//

```

```

this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 20F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(192)))),
((int)(((byte)(255)))));
this.BackgroundImage = global::Analitika.Properties.Resources.texture_1668079_1280;
this.ClientSize = new System.Drawing.Size(1027, 450);
this.Controls.Add(this.button3);
this.Controls.Add(this.button2);
this.Controls.Add(this.button1);
this.Name = "start";
this.SizeGripStyle = System.Windows.Forms.SizeGripStyle.Show;
this.Text = "Start";
this.TransparencyKey = System.Drawing.Color.FromArgb(((int)(((byte)(224)))),
((int)(((byte)(224)))), ((int)(((byte)(224)))));
this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
}
}

```


ПРИЛОЖЕНИЕ Е

Кода формы системного администратора

\\Analitika\admin.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Analitika
{
    public partial class admin : Form
    {
        public admin()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.pokaz". При
            // необходимости она может быть перемещена или удалена.
            this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);
            // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.analiz". При
            // необходимости она может быть перемещена или удалена.
            this.analizTableAdapter.Fill(this.analitikabaza.analiz);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.analizTableAdapter.instanaliz(this.textBox4.Text);
            this.analizTableAdapter.Fill(this.analitikabaza.analiz);
        }

        private void label4_Click(object sender, EventArgs e)
        {
            admin form2 = new admin();
            form2.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            DataRowView ss;
            int d;
            ss = (DataRowView)this.analizBindingSource.Current;
            d = (int)ss["id_analis"];
            this.analizTableAdapter.del_analiz(d);
            this.analizTableAdapter.Fill(this.analitikabaza.analiz);
        }

        private void button3_Click(object sender, EventArgs e)
        {
            DataRowView ss;
            int d;

```

```

ss = (DataRowView)this.analizBindingSource.Current;
    d = (int)ss["id_analiz"];
this.analizTableAdapter.up_analiz(Convert.ToString(this.textBox4.Text), d);
this.analizTableAdapter.Fill(this.analitikabaza.analiz);
    }

private void button6_Click(object sender, EventArgs e)
    {
        DataRowView ss = (DataRowView)this.comboBox1.SelectedItem;
int d = (int)ss["id_analiz"];
this.pokazTableAdapter.instpokaz(this.textBox1.Text, d, Convert.ToInt32(this.textBox2.Text),
Convert.ToInt32(this.textBox3.Text));
this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);
    }

private void textBox4_TextChanged(object sender, EventArgs e)
    {

    }

private void button5_Click(object sender, EventArgs e)
    {
        DataRowView ss;
int d1;
        ss = (DataRowView)this.pokazBindingSource.Current;
        d1 = (int)ss["id_pokaz"];
this.pokazTableAdapter.delpokaz(d1);
this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);

    }

private void button4_Click(object sender, EventArgs e)
    {
        DataRowView ss;
int d;
        ss = (DataRowView)this.pokazBindingSource.Current;
        d = (int)ss["id_analiz"];
int d1;
        d1 = (int)ss["id_pokaz"];
this.pokazTableAdapter.uppokaz(this.textBox1.Text, d, Convert.ToInt32(this.textBox2.Text),
Convert.ToInt32(this.textBox3.Text), d1);
this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);
    }

}
}
\Analitika\admin.Designer.cs
namespace Analitika
{
    partial class admin
    {
        ///<summary>
        /// Обязательная переменная конструктора.
        ///</summary>
private System.ComponentModel.IContainer components = null;

        ///<summary>

```

```

/// Освободить все используемые ресурсы.
///</summary>
///<param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе
ложно.</param>
protectedoverridevoid Dispose(bool disposing)
    {
    if (disposing && (components != null))
        {
            components.Dispose();
        }
    base.Dispose(disposing);
    }

#region Код, автоматически созданный конструктором форм Windows

///<summary>
/// Требуемый метод для поддержки конструктора – не изменяйте
/// содержимое этого метода с помощью редактора кода.
///</summary>
privatevoid InitializeComponent()
    {
    this.components = new System.ComponentModel.Container();
        System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(admin));
    this.dataGridView1 = new System.Windows.Forms.DataGridView();
    this.idanalisDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.nazvanalizDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.analizBindingSource = new System.Windows.Forms.BindingSource(this.components);
    this.analitikabaza = new Analitika.Analitikabaza();
    this.analizTableAdapter = new Analitika.AnalitikabazaTableAdapters.analizTableAdapter();
    this.dataGridView2 = new System.Windows.Forms.DataGridView();
    this.idpokazDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.nazvpokazDataGridViewTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.idanalizDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.vvozDataGridViewTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.pokazBindingSource = new System.Windows.Forms.BindingSource(this.components);
    this.pokazTableAdapter = new Analitika.AnalitikabazaTableAdapters.pokazTableAdapter();
    this.button1 = new System.Windows.Forms.Button();
    this.button2 = new System.Windows.Forms.Button();
    this.button3 = new System.Windows.Forms.Button();
    this.button4 = new System.Windows.Forms.Button();
    this.button5 = new System.Windows.Forms.Button();
    this.button6 = new System.Windows.Forms.Button();
    this.comboBox1 = new System.Windows.Forms.ComboBox();
    this.pokazBindingSource1 = new System.Windows.Forms.BindingSource(this.components);
    this.analizBindingSource1 = new System.Windows.Forms.BindingSource(this.components);

    this.textBox1 = new System.Windows.Forms.TextBox();
    this.textBox2 = new System.Windows.Forms.TextBox();
    this.textBox3 = new System.Windows.Forms.TextBox();
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.label3 = new System.Windows.Forms.Label();
    this.label4 = new System.Windows.Forms.Label();
    this.label5 = new System.Windows.Forms.Label();
    this.textBox4 = new System.Windows.Forms.TextBox();

```

```

this.button7 = new System.Windows.Forms.Button();
this.pokazBindingSource2 = new System.Windows.Forms.BindingSource(this.components);
this.fKpokazanalizBindingSource = new System.Windows.Forms.BindingSource(this.components);
    ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.analizBindingSource)).BeginInit();
    ((System.ComponentModel.ISupportInitialize)(this.analitikabaza)).BeginInit();
    ((System.ComponentModel.ISupportInitialize)(this.dataGridView2)).BeginInit();
    ((System.ComponentModel.ISupportInitialize)(this.pokazBindingSource)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.pokazBindingSource1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.analizBindingSource1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.pokazBindingSource2)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.fKpokazanalizBindingSource)).BeginInit();
this.SuspendLayout();
//
// dataGridView1
//
this.dataGridView1.AutoGenerateColumns = false;
this.dataGridView1.BackgroundColor = System.Drawing.SystemColors.ActiveCaption;
this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridColumn[] {
this.idanalizDataGridViewTextBoxColumn,
this.nazvanalizDataGridViewTextBoxColumn});
this.dataGridView1.DataSource = this.analizBindingSource;
this.dataGridView1.Location = new System.Drawing.Point(13, 13);
this.dataGridView1.Name = "dataGridView1";
this.dataGridView1.RowHeadersWidth = 62;
this.dataGridView1.RowTemplate.Height = 28;
this.dataGridView1.Size = new System.Drawing.Size(462, 150);
this.dataGridView1.TabIndex = 0;
//
// idanalizDataGridViewTextBoxColumn
//
this.idanalizDataGridViewTextBoxColumn.DataPropertyName = "id_analiz";
this.idanalizDataGridViewTextBoxColumn.HeaderText = "id_analiz";
this.idanalizDataGridViewTextBoxColumn.MinimumWidth = 8;
this.idanalizDataGridViewTextBoxColumn.Name = "idanalizDataGridViewTextBoxColumn";
this.idanalizDataGridViewTextBoxColumn.ReadOnly = true;
this.idanalizDataGridViewTextBoxColumn.Visible = false;
this.idanalizDataGridViewTextBoxColumn.Width = 150;
//
// nazvanalizDataGridViewTextBoxColumn
//
this.nazvanalizDataGridViewTextBoxColumn.DataPropertyName = "nazv_analiz";
this.nazvanalizDataGridViewTextBoxColumn.HeaderText = "Названиеанализа";
this.nazvanalizDataGridViewTextBoxColumn.MinimumWidth = 8;
this.nazvanalizDataGridViewTextBoxColumn.Name = "nazvanalizDataGridViewTextBoxColumn";
this.nazvanalizDataGridViewTextBoxColumn.Width = 150;
//
// analizBindingSource
//
this.analizBindingSource.DataMember = "analiz";
this.analizBindingSource.DataSource = this.analitikabaza;
//

```

ПродолжениеприложенияномерЕ

```

// analitikabaza
//
this.analitikabaza.DataSetName = "Analitikabaza";
this.analitikabaza.SchemaSerializationMode =
System.Data.SchemaSerializationMode.IncludeSchema;
//
// analizTableAdapter
//this.analizTableAdapter.ClearBeforeFill = true;
//
// dataGridView2
//
this.dataGridView2.AutoGenerateColumns = false;
this.dataGridView2.BackgroundColor = System.Drawing.SystemColors.ActiveCaption;
this.dataGridView2.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView2.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.idpokazDataGridViewTextBoxColumn,
this.nazvpokazDataGridViewTextBoxColumn,
this.idanalizDataGridViewTextBoxColumn,
this.vvozDataGridViewTextBoxColumn,
this.nvozDataGridViewTextBoxColumn});
this.dataGridView2.DataSource = this.pokazBindingSource;
this.dataGridView2.Location = new System.Drawing.Point(13, 214);
this.dataGridView2.Name = "dataGridView2";
this.dataGridView2.RowHeadersWidth = 62;
this.dataGridView2.RowTemplate.Height = 28;
this.dataGridView2.Size = new System.Drawing.Size(988, 150);
this.dataGridView2.TabIndex = 1;
//
// idpokazDataGridViewTextBoxColumn
//
this.idpokazDataGridViewTextBoxColumn.DataPropertyName = "id_pokaz";
this.idpokazDataGridViewTextBoxColumn.HeaderText = "id_pokaz";
this.idpokazDataGridViewTextBoxColumn.MinimumWidth = 8;
this.idpokazDataGridViewTextBoxColumn.Name = "idpokazDataGridViewTextBoxColumn";
this.idpokazDataGridViewTextBoxColumn.ReadOnly = true;
this.idpokazDataGridViewTextBoxColumn.Visible = false;
this.idpokazDataGridViewTextBoxColumn.Width = 150;
//
// nazvpokazDataGridViewTextBoxColumn
//
this.nazvpokazDataGridViewTextBoxColumn.DataPropertyName = "nazv_pokaz";
this.nazvpokazDataGridViewTextBoxColumn.HeaderText = "Названиепоказателя";
this.nazvpokazDataGridViewTextBoxColumn.MinimumWidth = 8;

this.nazvpokazDataGridViewTextBoxColumn.Name = "nazvpokazDataGridViewTextBoxColumn";
this.nazvpokazDataGridViewTextBoxColumn.Width = 150;
//
// idanalizDataGridViewTextBoxColumn
//
this.idanalizDataGridViewTextBoxColumn.DataPropertyName = "id_analiz";
this.idanalizDataGridViewTextBoxColumn.HeaderText = "id_analiz";
this.idanalizDataGridViewTextBoxColumn.MinimumWidth = 8;
this.idanalizDataGridViewTextBoxColumn.Name = "idanalizDataGridViewTextBoxColumn";
this.idanalizDataGridViewTextBoxColumn.Width = 150;
//
// vvozDataGridViewTextBoxColumn
//
this.vvozDataGridViewTextBoxColumn.DataPropertyName = "v_voz";

```

Продолжение приложения номер E

```

this.vvozDataGridViewTextBoxColumn.HeaderText = "ВерхняяграницаВ03";
this.vvozDataGridViewTextBoxColumn.MinimumWidth = 8;
this.vvozDataGridViewTextBoxColumn.Name = "vvozDataGridViewTextBoxColumn";
this.vvozDataGridViewTextBoxColumn.Width = 150;
//
// nvozDataGridViewTextBoxColumn
//
this.nvozDataGridViewTextBoxColumn.DataPropertyName = "n_voz";
this.nvozDataGridViewTextBoxColumn.HeaderText = "НижняяграницаВ03";
this.nvozDataGridViewTextBoxColumn.MinimumWidth = 8;
this.nvozDataGridViewTextBoxColumn.Name = "nvozDataGridViewTextBoxColumn";
this.nvozDataGridViewTextBoxColumn.Width = 150;
//
// pokazBindingSource
//
this.pokazBindingSource.DataMember = "pokaz";
this.pokazBindingSource.DataSource = this.analitikabaza;
//
// pokazTableAdapter
//
this.pokazTableAdapter.ClearBeforeFill = true;
//
// button1
//
this.button1.Image = ((System.Drawing.Image)(resources.GetObject("button1.Image")));
this.button1.Location = new System.Drawing.Point(745, 24);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(95, 31);
this.button1.TabIndex = 2;
this.button1.Text = "Добавить";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// button2
//
this.button2.Image = ((System.Drawing.Image)(resources.GetObject("button2.Image")));
this.button2.Location = new System.Drawing.Point(745, 69);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(95, 31);
this.button2.TabIndex = 3;
this.button2.Text = "Удалить";
this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2_Click);
//
// button3
//
this.button3.Image = ((System.Drawing.Image)(resources.GetObject("button3.Image")));
this.button3.Location = new System.Drawing.Point(745, 115);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(95, 29);
this.button3.TabIndex = 4;
this.button3.Text = "Изменить";
this.button3.UseVisualStyleBackColor = true;
this.button3.Click += new System.EventHandler(this.button3_Click);
//
// button4
//
this.button4.Image = ((System.Drawing.Image)(resources.GetObject("button4.Image")));
this.button4.Location = new System.Drawing.Point(745, 475);

```

ПродолжениеприложенияномерЕ

```

this.button4.Name = "button4";
this.button4.Size = new System.Drawing.Size(95, 31);
this.button4.TabIndex = 7;
this.button4.Text = "ИЗМЕНИТЬ";
this.button4.UseVisualStyleBackColor = true;
this.button4.Click += new System.EventHandler(this.button4_Click);
//
// button5
//
this.button5.Image = ((System.Drawing.Image)(resources.GetObject("button5.Image")));
this.button5.Location = new System.Drawing.Point(630, 475);
this.button5.Name = "button5";
this.button5.Size = new System.Drawing.Size(95, 31);
this.button5.TabIndex = 6;
this.button5.Text = "УДАЛИТЬ";
this.button5.UseVisualStyleBackColor = true;
this.button5.Click += new System.EventHandler(this.button5_Click);
//
// button6
//
this.button6.Image = ((System.Drawing.Image)(resources.GetObject("button6.Image")));
this.button6.Location = new System.Drawing.Point(517, 475);
this.button6.Name = "button6";
this.button6.Size = new System.Drawing.Size(95, 31);
this.button6.TabIndex = 5;
this.button6.Text = "ДОБАВИТЬ";
this.button6.UseVisualStyleBackColor = true;
this.button6.Click += new System.EventHandler(this.button6_Click);
//
// comboBox1
//
this.comboBox1.DataBindings.Add(new System.Windows.Forms.Binding("SelectedValue",
this.analizBindingSource, "id_analiz", true));
this.comboBox1.DataSource = this.analizBindingSource1;
this.comboBox1.DisplayMember = "nazv_analiz";
this.comboBox1.FormattingEnabled = true;
this.comboBox1.Location = new System.Drawing.Point(13, 371);
this.comboBox1.Name = "comboBox1";
this.comboBox1.Size = new System.Drawing.Size(211, 28);

this.comboBox1.TabIndex = 8;
this.comboBox1.ValueMember = "id_analiz";
//
// pokazBindingSource1
//
this.pokazBindingSource1.DataMember = "pokaz";
this.pokazBindingSource1.DataSource = this.analitikabaza;
//
// analizBindingSource1
//
this.analizBindingSource1.DataMember = "analiz";
this.analizBindingSource1.DataSource = this.analitikabaza;
//
// textBox1
//
this.textBox1.Location = new System.Drawing.Point(310, 373);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(216, 26);
this.textBox1.TabIndex = 9;
//

```

Продолжение приложения номер E

```

// textBox2
//
this.textBox2.Location = new System.Drawing.Point(13, 452);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(100, 26);
this.textBox2.TabIndex = 10;
//
// textBox3
//
this.textBox3.Location = new System.Drawing.Point(225, 452);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(100, 26);
this.textBox3.TabIndex = 11;
//
// label1
//
this.label1.AutoSize = true;
this.label1.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
((int)(((byte)(255)))), ((int)(((byte)(192))))));
this.label1.Location = new System.Drawing.Point(13, 406);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(63, 20);
this.label1.TabIndex = 12;
this.label1.Text = "анализ"; //
// label2
//
this.label2.AutoSize = true;
this.label2.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
((int)(((byte)(255)))), ((int)(((byte)(192))))));
this.label2.Location = new System.Drawing.Point(310, 406);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(173, 20);
this.label2.TabIndex = 13;
this.label2.Text = "названиепоказателя";
//
// label3
//
this.label3.AutoSize = true;

```

Продолжение приложения номер E

```

this.label3.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
((int)(((byte)(255)))), ((int)(((byte)(192))))));
this.label3.Location = new System.Drawing.Point(13, 487);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(173, 20);
this.label3.TabIndex = 14;
this.label3.Text = "В03верхняяграница";
//
// label4
//
this.label4.AutoSize = true;
this.label4.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
((int)(((byte)(255)))), ((int)(((byte)(192))))));
this.label4.Location = new System.Drawing.Point(221, 487);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(168, 20);
this.label4.TabIndex = 15;
this.label4.Text = "В03нижняяграница";
this.label4.Click += new System.EventHandler(this.label4_Click);
//

```



```

// label5
//
this.label5.AutoSize = true;
this.label5.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(255)))),
((int)(((byte)(255)))), ((int)(((byte)(192))))));
this.label5.Location = new System.Drawing.Point(532, 13);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(150, 20);
this.label5.TabIndex = 16;
this.label5.Text = "Названиеанализа";
//
// textBox4
//
this.textBox4.Location = new System.Drawing.Point(530, 48);
this.textBox4.Name = "textBox4";
this.textBox4.Size = new System.Drawing.Size(195, 26);
this.textBox4.TabIndex = 17;
this.textBox4.TextChanged += new System.EventHandler(this.textBox4_TextChanged);
//
// button7
//
this.button7.Location = new System.Drawing.Point(0, 0);
this.button7.Name = "button7";
this.button7.Size = new System.Drawing.Size(75, 23);
this.button7.TabIndex = 0;
//
// pokazBindingSource2
//
this.pokazBindingSource2.DataMember = "pokaz";
this.pokazBindingSource2.DataSource = this.analitikabaza;
//
// fKpokazanalizBindingSource
//
this.fKpokazanalizBindingSource.DataMember = "FK_pokaz_analiz";
this.fKpokazanalizBindingSource.DataSource = this.analizBindingSource;
//
// admin
//
this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 20F);

```

Продолжение приложения номер E

```

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
this.ClientSize = new System.Drawing.Size(1071, 517);
this.Controls.Add(this.button7);
this.Controls.Add(this.textBox4);
this.Controls.Add(this.label5);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.textBox3);
this.Controls.Add(this.textBox2);
this.Controls.Add(this.textBox1);
this.Controls.Add(this.comboBox1);
this.Controls.Add(this.button4);
this.Controls.Add(this.button5);
this.Controls.Add(this.button6);
this.Controls.Add(this.button3);

```

```

this.Controls.Add(this.button2);
this.Controls.Add(this.button1);
this.Controls.Add(this.dataGridView2);
this.Controls.Add(this.dataGridView1);
this.Name = "admin";
this.Text = "admin";
this.Load += new System.EventHandler(this.Form1_Load);
    ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.analizBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.analitikabaza)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.dataGridView2)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.pokazBindingSource)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.pokazBindingSource1)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.analizBindingSource1)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.pokazBindingSource2)).EndInit();}

((System.ComponentModel.ISupportInitialize)(this.fKpokazanalizBindingSource)).EndInit();
this.ResumeLayout(false);
this.PerformLayout();

```

```

private System.Windows.Forms.DataGridView dataGridView1;
private Analitikabaza analitikabaza;
private System.Windows.Forms.BindingSource analizBindingSource;
private AnalitikabazaTableAdapters.analizTableAdapter analizTableAdapter;
private System.Windows.Forms.DataGridView dataGridView2;
private System.Windows.Forms.BindingSource pokazBindingSource;
private AnalitikabazaTableAdapters.pokazTableAdapter pokazTableAdapter;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Button button4;
private System.Windows.Forms.Button button5;
private System.Windows.Forms.Button button6;
private System.Windows.Forms.ComboBox comboBox1;
private System.Windows.Forms.BindingSource pokazBindingSource1;
private System.Windows.Forms.BindingSource analizBindingSource1;
private System.Windows.Forms.TextBox textBox1;

```

Окончание приложения номер E

```

private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.TextBox textBox3;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.TextBox textBox4;
private System.Windows.Forms.Button button7;
private System.Windows.Forms.DataGridViewTextBoxColumn idanalizDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn nazvanalizDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn idpokazDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn nazvpokazDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn idanalizDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn vvozDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn nvozDataGridViewTextBoxColumn;
private System.Windows.Forms.BindingSource pokazBindingSource2;
private System.Windows.Forms.BindingSource fKpokazanalizBindingSource;
}
}

```


ПРИЛОЖЕНИЕЖ Кодформырегистратуры

\\Analitika\registratura.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Analitika
{
    publicpartialclassForm1 : Form
    {
        publicForm1()
        {
            InitializeComponent();
        }

        privatevoid Form1_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.pacient". При
            // необходимости она может быть перемещена или удалена.
            this.pacientTableAdapter.Fill(this.analitikabaza.pacient);
        }

        privatevoid button1_Click(object sender, EventArgs e)
        {
            this.pacientTableAdapter.inspacient(this.textBox1.Text, Convert.ToInt32(this.textBox2.Text),
            Convert.ToDateTime(this.textBox3.Text), Convert.ToInt32(this.textBox4.Text));
            this.pacientTableAdapter.Fill(this.analitikabaza.pacient);
        }

        privatevoid label2_Click(object sender, EventArgs e)
        {
        }

        privatevoid button2_Click(object sender, EventArgs e)
        {
            DataRowView ss;
            int d;
            ss = (DataRowView)this.pacientBindingSource.Current;
            d = (int)ss["id_pac"];
            this.pacientTableAdapter.delpac(d);
            this.pacientTableAdapter.Fill(this.analitikabaza.pacient);
        }

        privatevoid button3_Click(object sender, EventArgs e)
        {
            DataRowView ss;
            int d;
            ss = (DataRowView)this.pacientBindingSource.Current;
            d = (int)ss["id_pac"];
        }
    }
}
```

ОкончаниеприложенияномерЖ

```

this.pacientTableAdapter.uppacient(this.textBox1.Text, Convert.ToInt32(this.textBox2.Text),
Convert.ToDateTime(this.textBox3.Text), Convert.ToInt32(this.textBox4.Text), d);
this.pacientTableAdapter.Fill(this.analitikabaza.pacient);
}
}
}

```

\\Analitika\\Designer.cs

```

namespace Analitika
{
    partialclassForm1
    {
        ///<summary>
        /// Required designer variable.
        ///</summary>
        private System.ComponentModel.IContainer components = null;

        ///<summary>
        /// Clean up any resources being used.
        ///</summary>
        ///<param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protectedoverridevoid Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
            private System.Windows.Forms.DataGridView dataGridView1;
            private Analitikabaza analitikabaza;
            private System.Windows.Forms.BindingSource pacientBindingSource;
            private AnalitikabazaTableAdapters.pacientTableAdapter pacientTableAdapter;
            private System.Windows.Forms.Button button1;
            private System.Windows.Forms.Button button2;
            private System.Windows.Forms.Button button3;
            private System.Windows.Forms.TextBox textBox1;
            private System.Windows.Forms.TextBox textBox2;
            private System.Windows.Forms.TextBox textBox3;
            private System.Windows.Forms.TextBox textBox4;
            private System.Windows.Forms.Label label1;
            private System.Windows.Forms.Label label2;
            private System.Windows.Forms.Label label3;
            private System.Windows.Forms.Label label4;
            private System.Windows.Forms.DataGridViewTextBoxColumn idpacDataGridViewTextBoxColumn;
            private System.Windows.Forms.DataGridViewTextBoxColumn fioDataGridViewTextBoxColumn;
            private System.Windows.Forms.DataGridViewTextBoxColumn polisDataGridViewTextBoxColumn;
            private System.Windows.Forms.DataGridViewTextBoxColumn drDataGridViewTextBoxColumn;
            private System.Windows.Forms.DataGridViewTextBoxColumn otdDataGridViewTextBoxColumn;
        }
    }
}

```

ПРИЛОЖЕНИЕ 3

Код формы лаборатории

\\Analitika\\lab.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Analitika
{
    public partial class lab : Form
    {
        public lab()
        {
            InitializeComponent();
        }

        private void lab_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу "dataSet2.View_1". При
            // необходимости она может быть перемещена или удалена.
            this.view_1TableAdapter1.Fill(this.dataSet2.View_1);
            // TODO: данная строка кода позволяет загрузить данные в таблицу "dataSet1.View_1". При
            // необходимости она может быть перемещена или удалена.
            this.view_1TableAdapter.Fill(this.dataSet1.View_1);
            // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.pokaz". При
            // необходимости она может быть перемещена или удалена.
            this.pokazTableAdapter.Fill(this.analitikabaza.pokaz);
            // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.pacient". При
            // необходимости она может быть перемещена или удалена.
            this.pacientTableAdapter.Fill(this.analitikabaza.pacient);
            // TODO: данная строка кода позволяет загрузить данные в таблицу "analitikabaza.itogi". При
            // необходимости она может быть перемещена или удалена.
            this.itogiTableAdapter.Fill(this.analitikabaza.itogi);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DataRowView ss;
            int d = (int)this.comboBox1.SelectedValue;
            int d1 = (int)this.comboBox2.SelectedValue;
            //ss = (DataRowView)this.itogiBindingSource.Current;
            /*d = (int)ss["id_pac"];
            int d1;

            d1 = (int)ss["id_pokaz"];*/
            this.itogiTableAdapter.inaitogi(d, d1, Convert.ToInt32(this.textBox2.Text),
            Convert.ToDateTime(this.textBox1.Text));
            this.view_1TableAdapter1.Fill(this.dataSet2.View_1);
        }

        private void button2_Click(object sender, EventArgs e)
```

```

        {
DataRowViewss;
int d1;
            ss = (DataRowView)this.view1BindingSource3.Current;
            d1 = (int)ss["id_itog"];
this.itogiTableAdapter.delitog(d1);
this.view_1TableAdapter1.Fill(this.dataSet2.View_1);
        }

privatevoid button3_Click(object sender, EventArgs e)
        {
            DataRowView ss;
int d = (int)this.comboBox1.SelectedValue;
int d1 = (int)this.comboBox2.SelectedValue;
int d2;
            ss = (DataRowView)this.view1BindingSource3.Current;
            d2 = (int)ss["id_itog"];
//ss = (DataRowView)this.itogiBindingSource.Current;
/*d = (int)ss["id_pac"];
            int d1;

            d1 = (int)ss["id_pokaz"];*/
this.itogiTableAdapter.upitog(d, d1, Convert.ToInt32(this.textBox2.Text), d2,
Convert.ToDateTime(this.textBox1.Text));
this.view_1TableAdapter1.Fill(this.dataSet2.View_1);
        }
    }
}

```

\\Analitika\\lab.Designer.cs

```

namespace Analitika
{
partialclasslab
    {
///<summary>
/// Required designer variable.
///</summary>
private System.ComponentModel.IContainer components = null;

///<summary>
/// Clean up any resources being used.
///</summary>
///<param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
protectedoverridevoid Dispose(bool disposing)
    {
if (disposing && (components != null))
    {
        components.Dispose();
    }
base.Dispose(disposing);
    } private System.Windows.Forms.DataGridView dataGridView1;
private Analitikabaza analitikabaza;
private System.Windows.Forms.BindingSource itogiBindingSource;
private AnalitikabazaTableAdapters.itogiTableAdapter itogiTableAdapter;
private System.Windows.Forms.ComboBox comboBox1;
private System.Windows.Forms.ComboBox comboBox2;

```

```
private System.Windows.Forms.Button button1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.BindingSource pacientBindingSource;
private AnalytikabazaTableAdapters.pacientTableAdapter pacientTableAdapter;
private System.Windows.Forms.BindingSource itogiBindingSource1;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.BindingSource pokazBindingSource;
private AnalytikabazaTableAdapters.pokazTableAdapter pokazTableAdapter;
private System.Windows.Forms.BindingSource itogiBindingSource2;
private DataSet1 dataSet1;
private System.Windows.Forms.BindingSource view1BindingSource;
private DataSet1TableAdapters.View_1TableAdapter view_1TableAdapter;
private System.Windows.Forms.BindingSource view1BindingSource1;
private System.Windows.Forms.BindingSource view1BindingSource2;
private DataSet2 dataSet2;
private System.Windows.Forms.BindingSource view1BindingSource3;
private DataSet2TableAdapters.View_1TableAdapter view_1TableAdapter1;
private System.Windows.Forms.DataGridViewTextBoxColumn iditogDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn fio;
private System.Windows.Forms.DataGridViewTextBoxColumn nazv_analiz;
private System.Windows.Forms.DataGridViewTextBoxColumn nazv_pokaz;
private System.Windows.Forms.DataGridViewTextBoxColumn rezDataGridViewTextBoxColumn;
private System.Windows.Forms.DataGridViewTextBoxColumn data_pr;
}
}
```