

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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»

ПРОЕКТ ПРОВЕРЕН

Рецензент, главный врач ГБУЗ
«Городская больница №1» г. Копейск
_____ (А.В. Алешкевич)
« ____ » _____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой ИТЭ, д.т.н., с.н.с.
_____ (Б.М. Суховилов)
« ____ » _____ 2018 г.

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ
РЕНТГЕНОЛОГИЧЕСКОГО ОТДЕЛЕНИЯ: МОДУЛЬ
«СОЗДАНИЕ ФОРМАЛИЗИРОВАННОГО ПРОТОКОЛА
ФЛЮОРОГРАФИЧЕСКОГО ИССЛЕДОВАНИЯ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.03.2018.992 ПЗ ВКР

Консультант, доцент, к.т.н
_____ (О.И. Галичин)
« ____ » _____ 2018 г.

Руководитель, доцент
_____ (В.А. Конов)
« ____ » _____ 2018 г.

Автор проекта,
студент группы ЭиУ– 539
_____ (В.С. Добычин)
« ____ » _____ 2018 г.

Нормоконтролер, ст. преподаватель
_____ (Е.Н. Горных)
« ____ » _____ 2018 г.

Челябинск 2018

АННОТАЦИЯ

Добычин В.С. «Разработка автоматизированной системы для рентгенологического отделения: Модуль «создание формализованного протокола флюорографического исследования». – Челябинск: ЮУрГУ, ЗЭиУ – 539; 2018; 80 с., 42 рис., 10 лист., 12 табл., 3 прил., библиог. список – 8 наим.

Дипломный проект посвящен автоматизации процесса обработки результата флюорографического исследования для рентгенологического отделения с использованием технологии WPF. Объектом исследования проекта является рентгенологическое отделение флюорографического исследования.

В работе приведены материалы исследования организационно-функциональной структуры отделения и построена ее AS-IS модель. Сделаны предложения по устранению недостатков и их дальнейшему реинжинирингу, что послужило основой для разработки модели предполагаемых изменений TO-BE.

Изучен новый материал по обширному API – интерфейсу, позволяющему создавать графические программы с насыщенным дизайном и интерактивностью.

На основе созданной модели данных, была сделана база данных и программное обеспечение, предназначенное для создания формализованного протокола флюорографического исследования врачом-рентгенологом.

Разработан набор пиктограмм для описания скиалогической картины, с помощью которых врач-рентгенолог составляет графическую схему патологий.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1 Аналитическая часть	7
1.1 Организационная и экономическая характеристика предметной области	7
1.1.1 Организационно-функциональная модель	7
1.2.1 Обзор рынка готовых информационных систем	10
1.3 Функциональная модель AS-IS	13
1.3.1 Диаграмма верхнего уровня.....	13
1.3.2 Диаграммы декомпозиции	16
1.4 Функциональная модель TO-BE.....	18
1.5 Средства разработки программного обеспечения	21
1.5.1 Выбор системы управления базы данных	21
1.5.2 Используемые Case-средства для проектирования системы	22
1.5.3 Выбор среды разработки системы	23
1.5.4 Microsoft Expression Design.....	23
Выводы по разделу один	25
2 ПРОЕКТНАЯ ЧАСТЬ	27
2.1 Информационное обеспечение предметной области	27
2.1.1 Модель данных.....	28
2.1.2 Информативно-справочная информация.....	30
2.2 Характеристика результатов работы программного проекта.....	31
2.3 Требования к системе	32
2.3.1 Требования к функционированию и структуре системы.....	32
2.3.2 Требования к безопасности и надёжности ПО	32
2.4 Серверная часть программного обеспечения.....	33
2.5 Описание работы программы	35
2.5.1 Главная форма	35
2.5.2 Администрирование программного проекта	43
2.5.3 Редактор флюорографического протокола	45

2.5.3 Web-версия информационной системы.....	49
2.6 Структурная схема программы (дерево диалога).....	53
2.7 Безопасность информационной системы	54
2.7.1 Обеспечение системы безопасности.....	54
2.7.2 Требования к прикладному обеспечению системы.....	55
2.7.1 Компоненты программного обеспечения.....	56
2.8 Выбор технических средств для системы	57
Выводы по разделу два.....	58
3 Экономическое обоснование	59
Выводы по разделу три.....	63
ЗАКЛЮЧЕНИЕ	64
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	65
ПРИЛОЖЕНИЕ А Протокол флюорографического обследования.....	66
ПРИЛОЖЕНИЕ Б Часть кода главной формы на языке С#	67
ПРИЛОЖЕНИЕ В Часть кода формы создания протокола на языке С#	72

ВВЕДЕНИЕ

В связи с явно ухудшающейся экологией, человек всё больше внимания начинает уделять своему здоровью. По данным статистики на сегодняшний день патологии бронхов, трахеи, а также и легких занимают лидирующее место в списках заболеваемости злокачественными новообразованиями населения России. Количество болезней, связанных с органами дыхания, не идут на снижение. Проблема лечения заболеваний легких во многом зависит от ранней диагностики.

Сейчас многие больницы пытаются проводить раннюю диагностику, чтобы выявить болезнь в её раннем состоянии, когда нет ещё явных симптомов и поводов для беспокойства. Главная роль в этой диагностике играет именно флюорография. Например, туберкулёз на ранних стадиях протекает вяло и бессимптомно, и только флюорографическое обследование лёгких может обнаружить источник инфекции.

В настоящее время флюорография легких в основном выполняется на современных цифровых аппаратах. Основным преимуществом флюорографов нового поколения является малая лучевая нагрузка на пациента. Благодаря развитию технологии низко дозовой цифровой флюорографии и все более высоким возможностям постобработки лучевых изображений мы имеем возможность качественной визуализации теневых структур. К сожалению, не все медицинские учреждения способны приобрести цифровое оборудование с необходимым программным обеспечением из-за высокой ее стоимости. Сейчас на рынке много недорогих цифровых флюорографических аппаратов с простейшим ПО, которое предназначено только для просмотра и печати цифровых флюорографий, вот для таких аппаратов и создано мое программное обеспечение, с помощью которого можно будет составить графическую схему патологий, после чего автоматически сгенерируется полноценный текстовый протокол, а главное все снимки и созданный протокол хранятся в базе данных и врачу-рентгенологу не придется искать нужное исследование в архиве, когда это будет необходимо. Также созданная программа пригодится медицинских учреждений у которых есть дигитайзер – это аппарат,

предназначенный для оцифровки рентгеновских снимков (сканер). Т.е. если у пациента на руках имеется распечатанный снимок и ему необходимо флюорографическое заключение врача-рентгенолога, то этот снимок сканируется на специальном аппарате и сохраняется на жестком диске компьютера, после чего в созданном программном обеспечении открывается флюорографический снимок, и врач имеет возможность без дополнительных манипуляций описать данное изображение.

Целью данной работы является повышение эффективности и качества работы рентгенологического отделения.

Задачи работы:

1. Разработка модели AS-IS (как есть) рентгенологического отделения, где показаны бизнес-процессы с подробной детализацией, и которые позволяют узнать принципы и механизмы функционирования отделения как единого целого. Для описания модели хорошо подходит CASE-система Ramus Educational, в которой моделирование происходит в соответствии с методологиями IDEF0 [4].

2. Разработка модели «как должно быть» TO-BE, здесь предложены функциональные изменения. При изучении работы рентгенологического отделения внимание акцентировалось на недостатках, которые были подвержены изменению и представлены в модели «как должно быть».

3. Разработка программного обеспечения, которое позволит уменьшить недостатки, найденные в модели «как должно быть».

Объект дипломной работы – процесс «обработка результата флюорографического исследования» рентгенологического исследования.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Организационная и экономическая характеристика предметной области

1.1.1 Организационно-функциональная модель

Структура рентгенологического отделения является иерархической, которая представлена на рисунке 1.

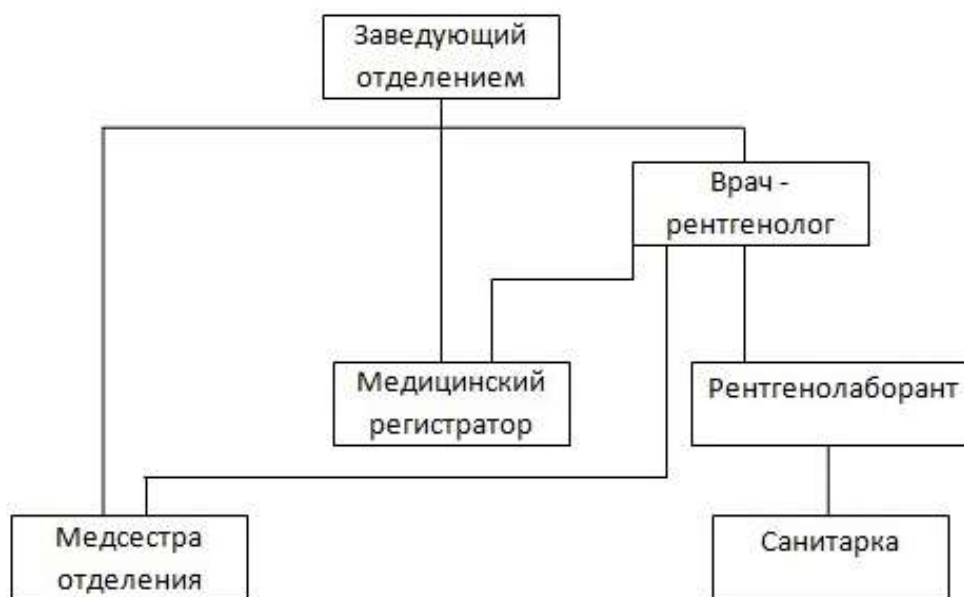


Рисунок 1 – Организационная структура рентгенологического отделения

Чтобы отобразить реально существующую деятельность отделения нужно построить модель бизнес-процесса. Для создания этой модели необходимо знать как устроена трудовая деятельность на каждом рабочем месте. Функционал сотрудников рентгенологического отделения представлен в таблице 1.

Таблица 1 – Основной функционал сотрудников

Сотрудник	Подчиняется	Основные функции и обязанности
Заведующий рентгеновским отделением	Главврачу больницы и его заместителю	• осуществляет руководство работой персонала отделения и контролирует качество работы врачей рентгенологов;

Сотрудник	Подчиняется	Основные функции и обязанности
Заведующий рентгеновским отделением	Главврачу больницы и его заместителю	<ul style="list-style-type: none"> • рационально организует труд сотрудников отделения; • проводит рентгеновские исследования в установленном объеме; • консультирует врачей-рентгенологов в наиболее сложных случаях диагностики; • внедряет современные методы исследований; • разбирает сложные случаи и ошибки в диагностике; • проводит различные мероприятия по повышению квалификации рентгенологов и рентгенолаборантов; • анализирует работу отделения за квартал, полугодие, год, представляет отчет о работе отделения в установленном порядке; • контролирует противопожарное состояние отделения, соблюдение персоналом правил охраны труда и техники безопасности; • своевременно представляет администрации больницы заявки на приобретение новой аппаратуры и запасных частей к этой аппаратуре, рентген плёнки, контрастных веществ, химических реактивов и т.д.; • систематически повышает профессиональную квалификацию.

Сотрудник	Подчиняется	Основные функции и обязанности
Врач-рентгенолог	Заведующему отделением, а при его отсутствии руководителю ЛПУ или его заместителю	<ul style="list-style-type: none"> • следит за правильностью проведения рентгенологических процедур, следит за эксплуатацией аппаратуры и оборудования, рационального использования рентген пленки; • ставит предварительный диагноз; • самостоятельно проводит или организует необходимые рентген исследования.
Медицинский регистратор	Заведующему отделением и врачу-рентгенологу	<ul style="list-style-type: none"> • производит учет и регистрацию; • распределяет поступающие снимки на хранение; • производит выдачу рентгеновских снимков по требованию врачей-рентгенологов.
Рентгенолаборант	Подчиняется врачу-рентгенологу	<ul style="list-style-type: none"> • готовит к приему пациентов, а также свое рабочее место; • делает томограммы, рентгенограммы, участвует в рентгеноскопии, проводит фотообработку; • следит за чистотой и порядком в рентген кабинете, следит за дозой рентгеновского излучения, а также за исправностью рентгеновского аппарата.
Медицинская сестра	Заведующему отделением и врачу-рентгенологу	<ul style="list-style-type: none"> • оказывает помощь заведующему отделением в административно-хозяйственных вопросах; • оказывает врачу-рентгенологу и рентгенолаборанту помощь при проведении сложных исследований;

Сотрудник	Подчиняется	Основные функции и обязанности
Медицинская сестра	Заведующему отделением и врачу-рентгенологу	<ul style="list-style-type: none"> • помогает рентгенолаборанту при обработке рентгеновской пленки; • в необходимых случаях, сопровождает больных после исследования в отделения.
Санитарка	Рентгенолаборанту	<ul style="list-style-type: none"> • содержит помещение, инвентарь и оборудование кабинета в должном порядке, проводит ежедневную влажную уборку помещения; • готовит кабинет к приему больных; • в необходимых случаях выполняет функции курьера; • сдает в установленном порядке халаты, белье в стирку и получает их.

1.2.1 Обзор рынка готовых информационных систем

1. Программное обеспечение «Carestream Image Suite»

CARESTREAM Image Suite – это система управления изображениями, которая позволяет получать, обрабатывать, просматривать, отображать, печатать и архивировать изображения и данные, полученные из различных систем визуализации. Image Suite – это автономная, не требующая аппаратного обеспечения система рентгенографии, которая является хорошим решением для управления медицинскими снимками и отчетами, а также администрирования данных пациентов/обследований в небольших клиниках [1].

Функциональные возможности:

- возможность получения изображений из беспроводных и привязанных систем DRX-1, а также систем Vita и Classic CR;

- точная настройка обработки изображений одним нажатием кнопки с использованием простых слайдов для настройки яркости, контрастности или детализации на экране сбора данных;

- компоновка изображения для печати на пленке;

- надежная портативная система рассчитана на эксплуатацию в экстремальных условиях и может использоваться для выполнения разных диагностических задач;

- выполнять резервное копирование на компакт-диски и съемные USB-устройства для автономного хранения.

2. Программный комплекс «Pulmoscreen»

Программный продукт Pulmoscreen предназначен для графического описания рентгенограмм врачом-рентгенологом [2].

Итоговый протокол содержит информацию об ЛПУ, данные пациента, исходное изображение рентгенограммы, графическую схему и сгенерированное текстовое описание. Возможно добавление собственного текста и заключения.

Основные преимущества продукта:

- информативное формализованное описание;

- однозначность и наглядность;

- повышение скорости работы;

- автоматическое создание текстового протокола;

- сохранение протоколов в базе данных.

3. Компьютерная программа «Ассистент диагноста»

Программа для рентгенолога, АРМ врача диагноста, АРМ УЗИ. Создание карт пациентов, формирование заключений врача с использованием ранее составленных шаблонов, возможность их редактирования, просмотр списка пациентов с возможностью выхода в режим просмотра и редактирования карт [3].

Программа предназначена для автоматического процесса написания протоколов исследования.

4. Автоматизированное рабочее место врача рентгенолога для компьютерной обработки пленочных рентгенограмм «Рентгенолог версия 2.10»

Функциональные возможности:

- быстрый доступ к информации;
- сохранение снимков в цифровом формате;
- снимки удобно пересылать по e-mail, просматривать и хранить;
- простое и удобное хранение: на жестком диске, Flash-карте или CD можно хранить много рентгенограмм;
- возможность экспорта информации в Excel для последующей обработки и обобщения.

Достоинства и недостатки ПО представлены в таблице 2:

Таблица 2 – Сравнительный анализ

ПО Различия	ImageSuiteV4	Pulmoscreen	Ассистент диагноста	Рентгенолог
Понятный интерфейс	–	–	+	+
Хранение снимков	+	+	–	+
Хранение протоколов	–	+	+	–
Автоматическое создание протокола	–	+	+	–
Средство для описания исследования	+	+	–	–
Полные данные пациентов	+	–	+	–
Данные врачей	+	+	+	–
Редактор снимков	+	+	–	–
Схематическое изображение очагов	+	+	–	–
Статистика (от-	+	–	+	–

четы)				
-------	--	--	--	--

1.3 Функциональная модель AS-IS

Чтобы систематизировать протекающие в данный момент процессы, необходимо создать модель деятельности рентгенологического отделения для определения наиболее слабых мест и изучения возможностей улучшения работы отделения.

Функциональная модель AS-IS делится на две части:

- контекстная диаграмма;
- диаграммы декомпозиции.

1.3.1 Диаграмма верхнего уровня

В самом начале проводится описание системы в целом и ее взаимодействие с окружающим миром. Для этого строится диаграмма верхнего уровня, которая называется контекстной диаграммой. На рисунке 2 изображена контекстная диаграмма процесса рентгенологического отделения.

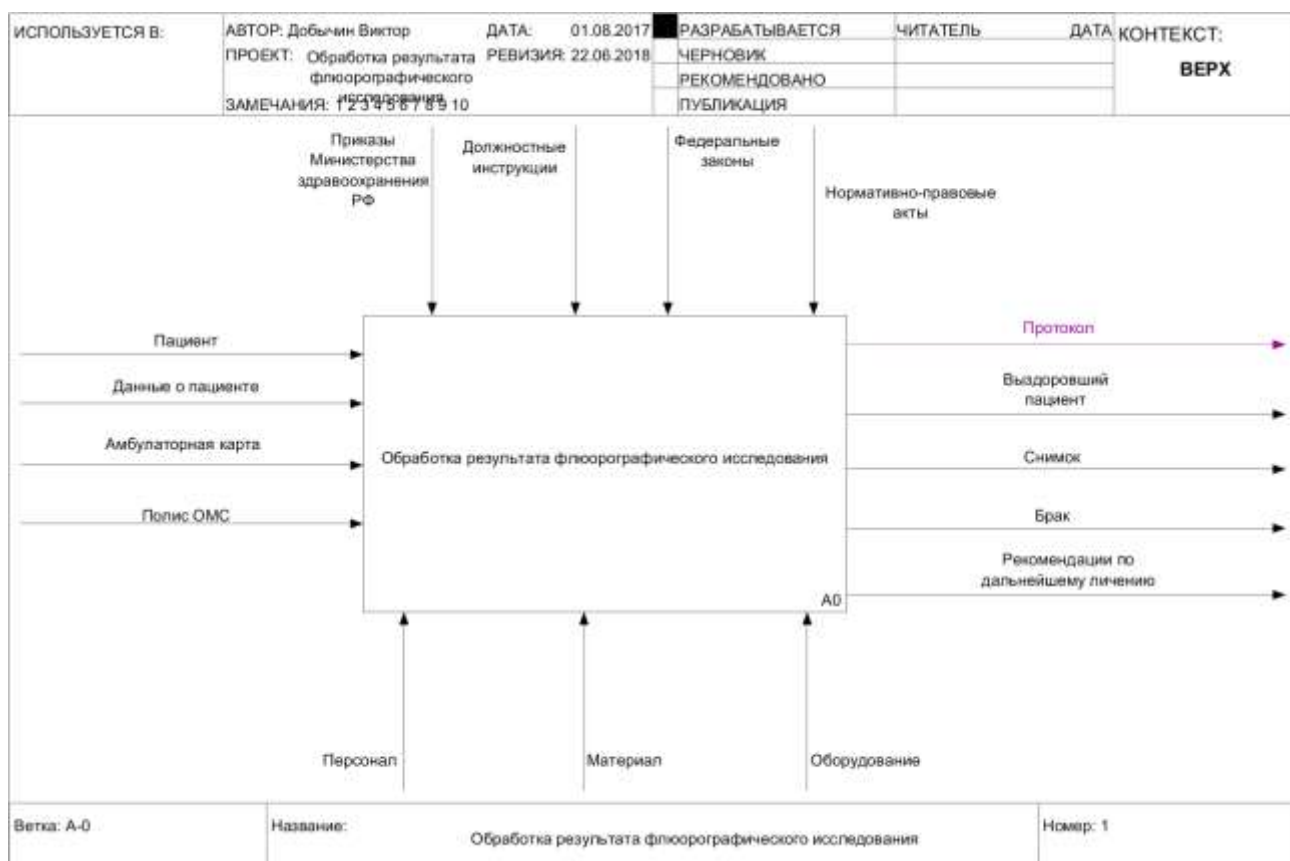


Рисунок 2 – Диаграмма верхнего уровня (контекстная диаграмма)

В модели бизнес-процессов взаимодействие осуществляется с четырех сторон. Каждая сторона блока имеет определенное название: левая сторона нужна для входов, верхняя предназначена для управления, правая предназначена для выходов и нижняя – для механизмов. Взаимодействие отделения с окружающей средой описаны в таблицах 3, 4, 5, 6.

Таблица 3 – Взаимодействие по входу

Вход	Описание
Пациент	Физическое лицо, которому оказывается медицинская помощь или имеющее проблемы с дыхательной системой.
Данные пациента	Информация, относящаяся к определенному физическому лицу (фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное положение, образование и т.д.).
Амбулаторная карта	Медицинский документ, в котором ведётся запись истории болезни пациента и назначаемого ему лечения.
Полис ОМС	Документ, гарантирующий получение бесплатной медицинской помощи в системе обязательного медицинского страхования на всей территории РФ.

Таблица 4 – Взаимодействие по выходу

Выход	Описание
Протокол	Диагностический вывод врача-рентгенолога.
Амбулаторное лечение	Лечение, проводимое на дому или при посещении самими больными лечебного учреждения.
Выздоровевший пациент	Физическое лицо, избавившееся от недуга.

Снимок	Фотографическое изображение грудной клетки.
--------	---

Окончание таблицы 4

Брак	Некачественные рентгенологические и флюорографические снимки.
Рекомендации по дальнейшему лечению	Совет, пожелание пациенту, данное с целью лечения недуга.

Таблица 5 – Взаимодействие по управлению

Управление	Описание
Приказы Министерства здравоохранения РФ	Федеральное министерство, осуществляющее функции по выработке государственной политики и нормативно-правовому регулированию в сфере здравоохранения.
Основы законодательства РФ	Законодательный акт, устанавливавший основные принципы правового регулирования в определенной сфере общественных отношений.
Должностные инструкции	Документ, в котором определяются обязанности, полномочия и ответственность сотрудника при осуществлении им деятельности в определенной должности.
Нормативно-правовые документы, регламентирующие деятельность учреждений здравоохранения	Письменные официальные документы, принятые в определенной форме правотворческим органом в пределах его компетенции и направленные на установление, изменение или отмену правовых норм.

Таблица 6 – Взаимодействие по механизму

Механизм	Описание
Материал	Медицинские расходные материалы рентгенологического отделения, предназначенные для обработки, защиты пациента и врача при рентгенологических исследованиях.
Оборудование	Технические средства отделения, предназначенные для непосредственного лечения пациентов.
Персонал	Постоянный состав работников отделения, разделенные по функциональному признаку.

1.3.2 Диаграммы декомпозиции

Диаграммы декомпозиции предназначены для детализации функций и получается при разбиении контекстной диаграммы. Детализация бизнес-процессов формирования флюорографического протокола после разбиения представлены на рисунках 3(а, б, в, г).

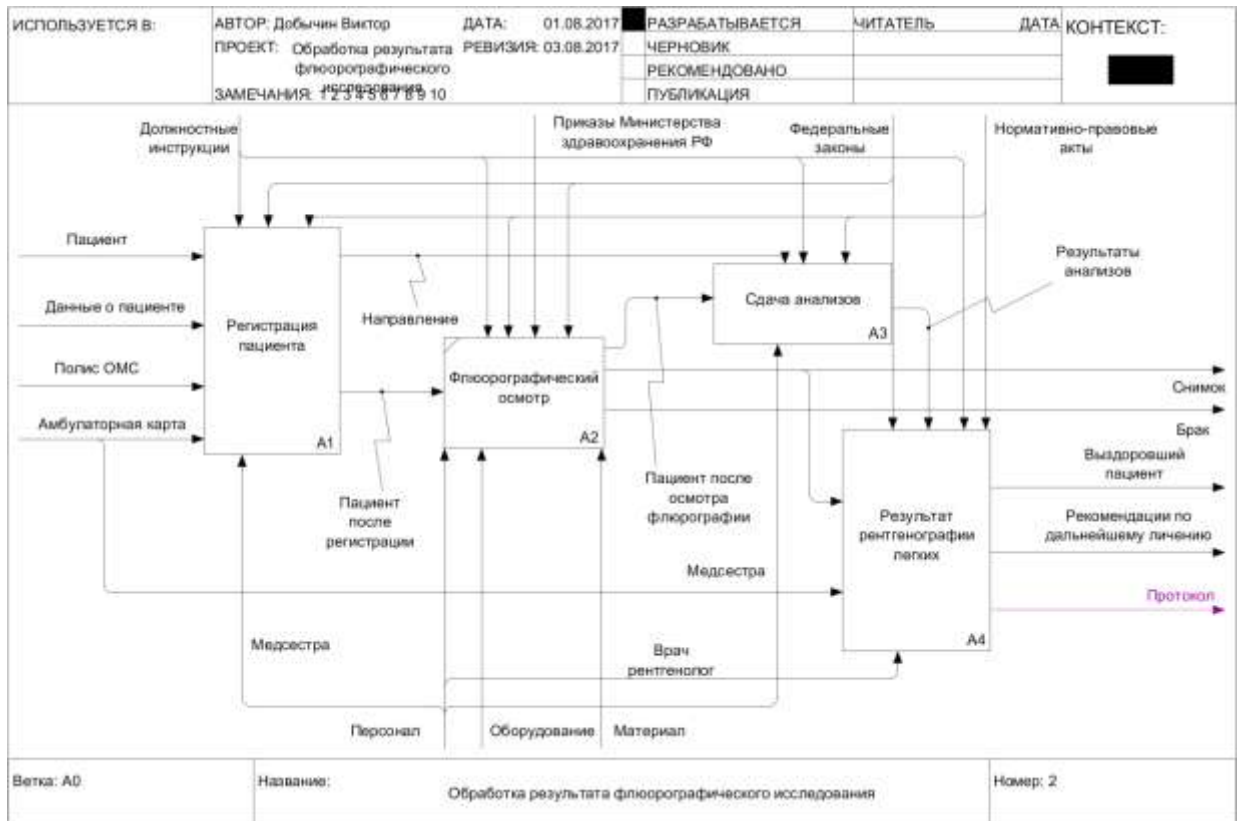


Рисунок 3а – Декомпозиция первого уровня

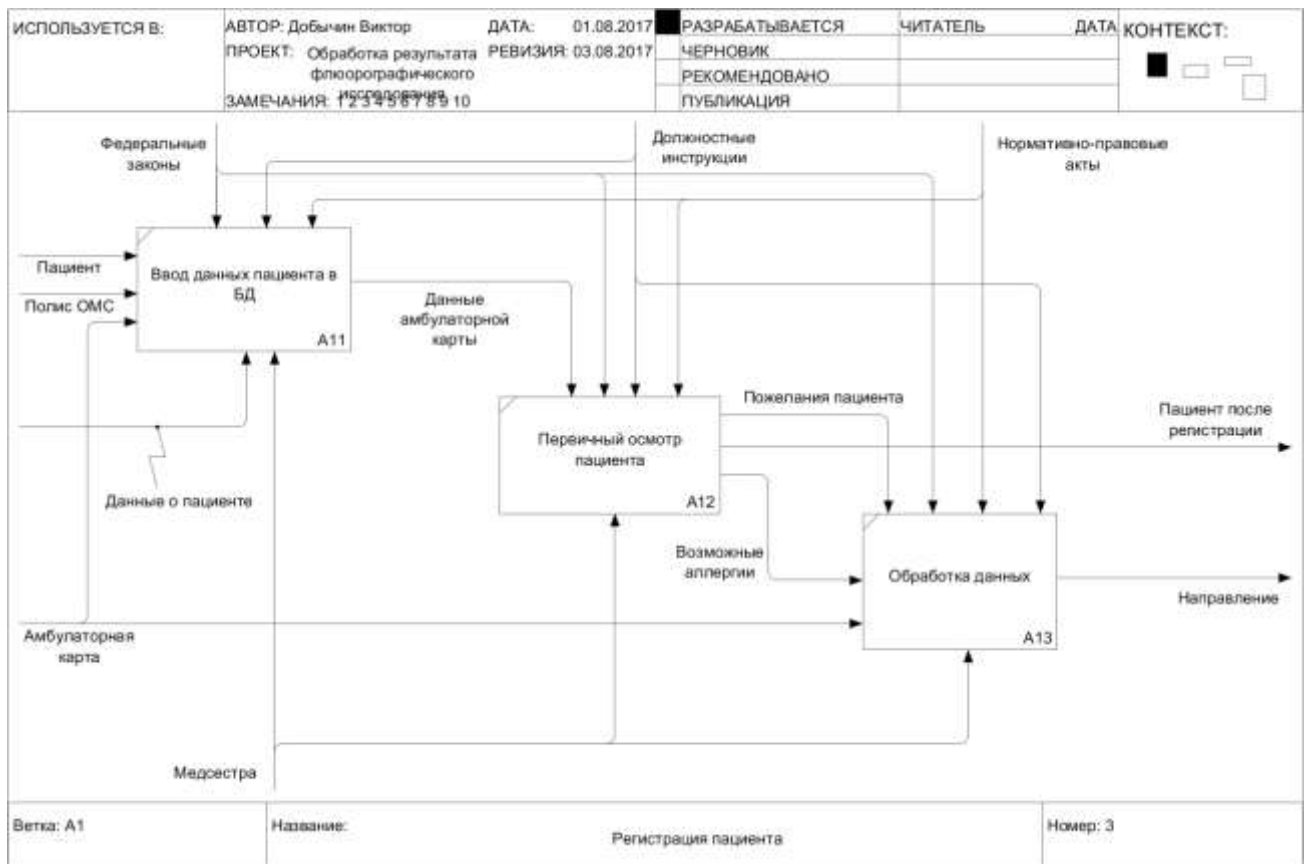


Рисунок 3б – Декомпозиция регистрации пациента

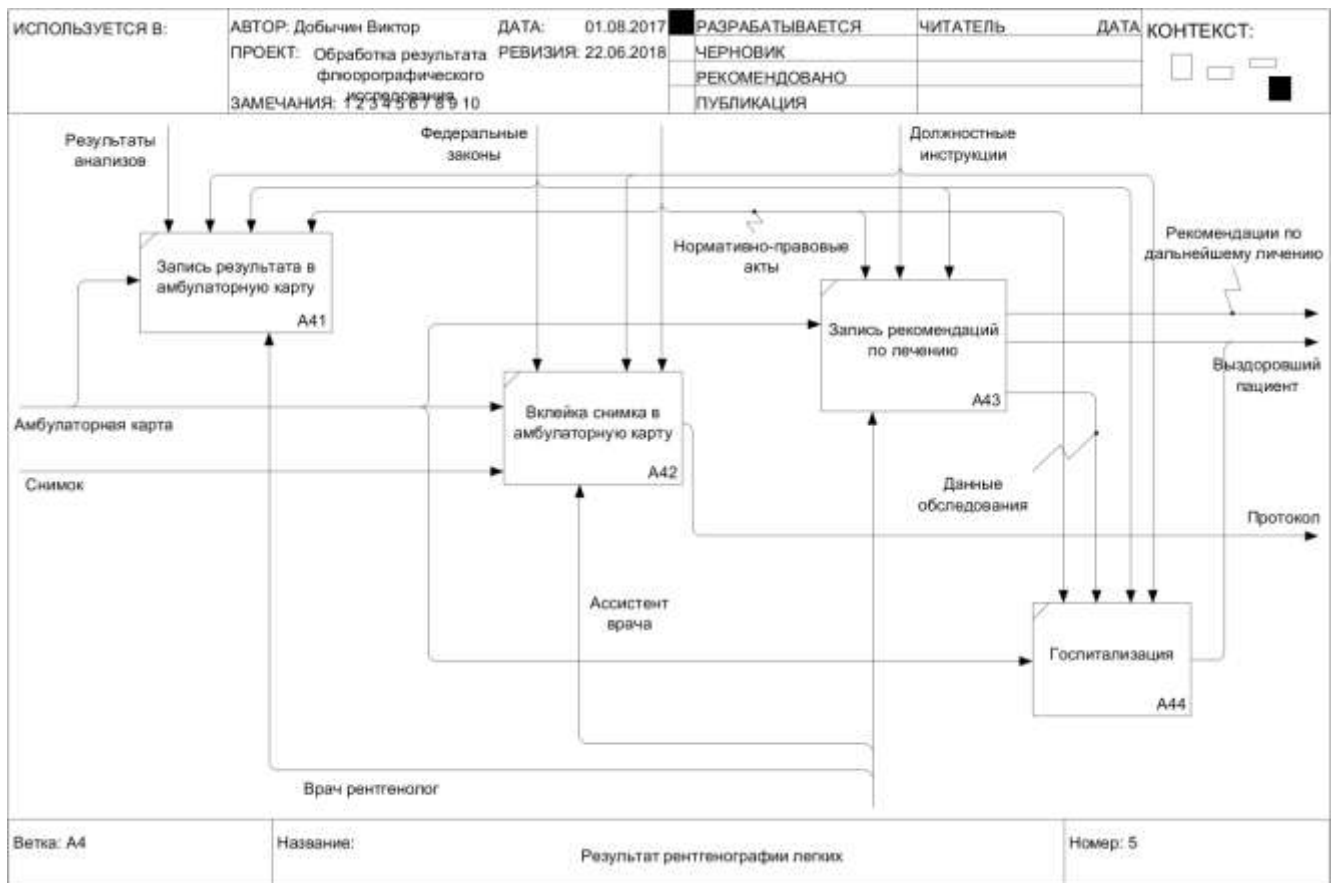


Рисунок 3г – Декомпозиция результата флюорографии

Из модели бизнес-процессов видно, что врач-рентгенолог записывает свое заключение в амбулаторную карту, либо печатает весь протокол вручную, для этого он должен сначала изучить снимок. На данный процесс уходит не мало времени. А если необходимо отметить область патологии на снимке? Опять время. Созданная мной программа автоматизирует этот процесс, делая его намного удобней и быстрее, плюс все это будет храниться в базе данных.

1.4 Функциональная модель ТО-ВЕ

В предыдущей статье были рассмотрены бизнес-процессы модели AS-IS и выявлены те из них, которые нуждаются в реинжиниринге. Переход от модели AS-IS к модели ТО-ВЕ происходит путем изменений бизнес-процессов в лучшую сторону на основе оценки их эффективности.

На основании анализа модели «как есть» были выявлены следующие недостатки:

- большая трата времени на оформление протокола вручную;
- снимки хранятся в архивах (отделения лучевой диагностики), что затрудняет быстрый доступ и поиск;
- срок хранения снимка ограничен.

Глядя на недостатки трудовой деятельности из модели AS-IS, была создана модель TO-BE («как должно быть»), устраняющая эти недостатки, а ручное оформление флюорографического протокола было автоматизировано, что облегчит расшифровку и описание рентгенограммы легких врачом и уменьшит время на оформление протокола, что, в конечном счете, должно сказаться на финансовом состоянии ЛПУ.

На рисунке 4 представлена диаграмма автоматизации оформления результата флюорографического обследования.

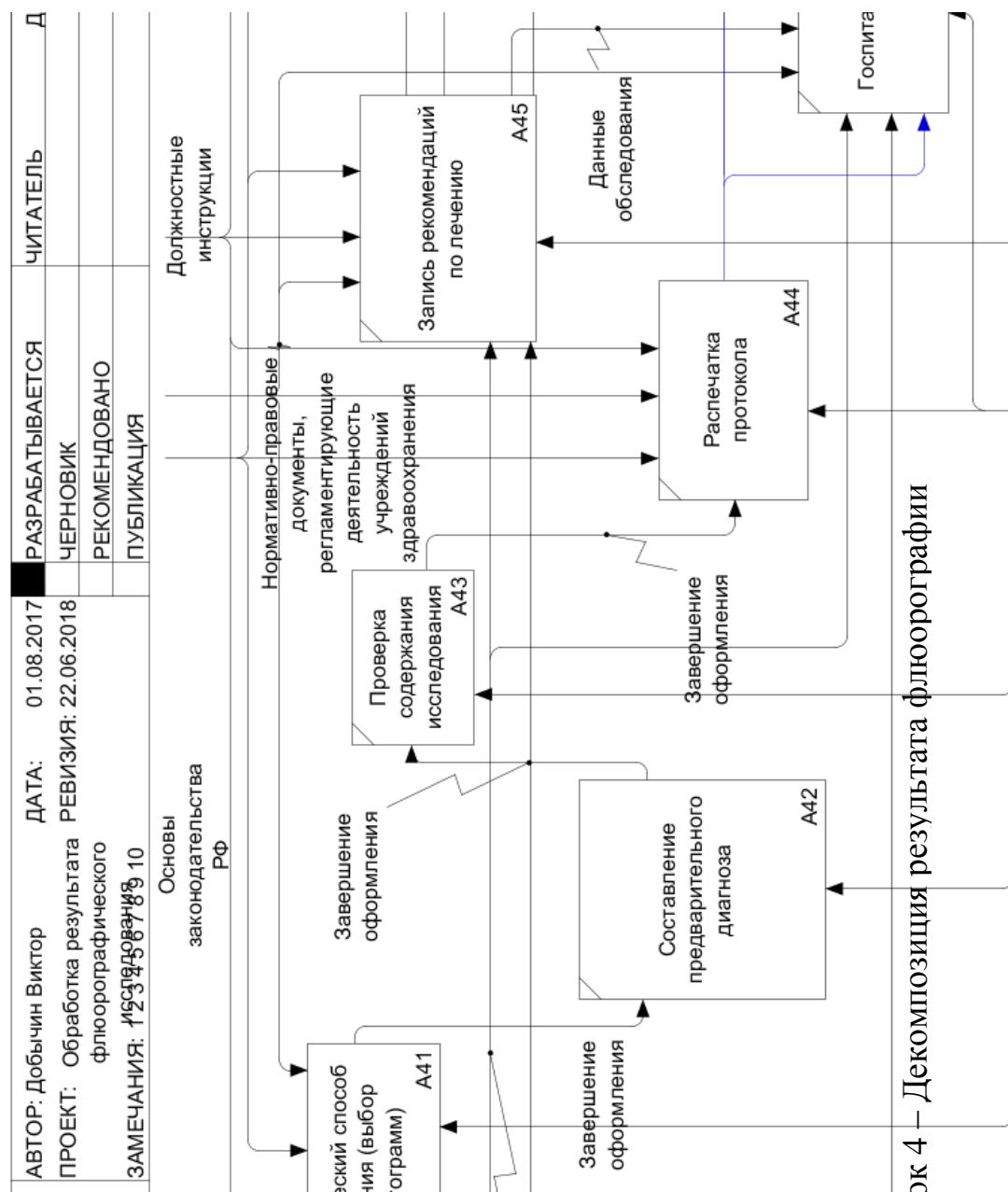


Рисунок 4 – Декомпозиция результата флюорографии

В модели TO-VE было подвергнуто легкому реинжинирингу процесс «Результат рентгенографии легких». Этот процесс подвергнут автоматизации с помощью программы, разработанной далее.

Процессы «Результата рентгенографии легких»:

- графический способ описания исследования;
- распечатка протокола на принтере;
- запись рекомендаций по лечению;
- госпитализация (по требованию врача).

Благодаря этому, процесс описания скиалогической картины для создания формализованного протокола флюорографического исследования, будет сокращен в разы, т.к. врач-рентгенолог рентгенологического отделения будет составлять графическую схему патологий на компьютере, после чего по составленной графической схеме автоматически генерируется полноценный текстовый протокол, полностью готовый к печати. Составление графического протокола с помощью мыши занимает гораздо меньше времени, чем набор текста на клавиатуре или написания с помощью ручки в амбулаторной карте, где ошибку так просто не исправить.

Так же благодаря автоматизации есть возможность пересылать или транспортировать снимки и текст протокола на любом носителе.

1.5 Средства разработки программного обеспечения

1.5.1 Выбор системы управления базы данных

Существует несколько баз данных с сетевым доступом имея различную архитектуру, а именно:

1. Встраиваемая база данных. Данная СУБД очень тесно связана с программой и расположена на этом же компьютере. Минус данной СУБД заключается в том, что она подходит только для локального использования. Примеры: Firebird Embedded, SQLite, VistaDB, TurboDB

2. Файл-серверные базы данных. Здесь файлы, в соответствии с запросом пользователей передаются на рабочие станции. На этих рабочих станциях и происходит обработка запроса. Данная технология считается устаревшей. Пример: Microsoft Office Access – СУБД реляционная, корпорации Microsoft. В данной СУБД отсутствует ряд функций, которые необходимы моему приложению.

3. Клиент-серверные.

Мною было выбрано решение использовать именно данную СУБД, так как обработка данных происходит на сервере, что увеличивает производительность созданного программного обеспечения и приводит к снижению загрузки локаль-

ной сети. К такой базе данных относится Microsoft SQL Server – система управления реляционными базами данных. Основной используемый язык запросов – Transact-SQL.

Microsoft SQL Server – это новейшая и мощнейшая система управления базами данных. Помимо стандартных функций СУБД, Microsoft SQL содержит большой набор внутренних служб для анализа данных. Доступ к этим данным, которые расположены на сервере, могут получить совершенно любые приложения, разработанные на .Net и Visual Studio, а также приложения пакета Microsoft Office. SQL Server обеспечивает высочайшую в своём классе масштабируемость, производительность и безопасность.

1.5.2 Используемые Case-средства для проектирования системы

При выборе case-средства, было выделено несколько обязательных критериев: доступность и функциональность. Во-первых, приложение должно быть бесплатным, чтобы был доступ ко всей функциональности, а во-вторых, большие возможности для проектирования, выбор различных диаграмм и т.п.

В качестве среды для моделирования бизнес-процессов мною была выбрана система Ramus Educational по следующим причинам:

Основная функциональность:

- разработка графических моделей бизнес-процессов (поддерживаются нотации IDEF0 и DFD);
- разработка систем классификации и кодирования (с привязкой к моделям процессов);
- создание отчетов по моделям и системе классификации (в виде регламентов бизнес-процессов, должностных инструкций и т.п.) [4].

Преимущества перед аналогами:

- поддержка неограниченного количества атрибутов различных типов;
- гибкий графический интерфейс пользователя.

1.5.3 Выбор среды разработки системы

Для создания своего приложения был выбран язык программирования – C#. Это объектно-ориентированный язык программирования. Данный язык был выбран из-за технологии WPF (Windows Presentation Foundation), который позволяет создавать приложения с красивым интерфейсом, используя язык XAML, а для разработки автоматизированной системы, была выбрана интегрированная среда разработки Visual Studio Ultimate2013.

1.5.4 Microsoft Expression Design

Expression Design – инструментальное средство, которое позволяет художникам-оформителям создавать изысканные образцы векторной графики.

Данное приложение позволяет сохранять результат в самых разных стандартных форматах файлов, включая PNG, JPEG, GIF, TIFF, но самое интересное, что Expression Design позволяет также сохранять графические данные в формате XAML для приложений WPF. Расширяемый язык разметки приложений (XAML) основывается на синтаксисе языка XML для описания состояния графического или другого объекта на платформе .NET.

Например, на рисунке 5, приведено векторное изображение, созданное в программе Expression Design, а в листинге 1 приведен его фрагмент кода разметки на языке XAML, который описывает внешний вид векторного изображения кнопки «Справка». Данный пример показывает, как с помощью кода XAML можно создать привлекательные эффекты закругления границ кнопки, вставлять любую картинку и создавать эффекты радиального градиента, заполняющего участок внутри кнопки [5]:



Рисунок 5 – Векторное изображение кнопки

Листинг 1 – Код векторного изображение кнопки на языке XAML

```
489 <Viewbox x:Name="viewboxhelp" Margin="0,1,403.8,0" Width="69" Height="69" VerticalAlignment="Top
490 HorizontalAlignment="Right" Cursor="Hand" MouseEnter="viewboxhelp_MouseEnter"
491 MouseLeave="viewboxhelp_MouseLeave" MouseLeftButtonUp="viewboxhelp_MouseLeftButtonUp">
492 <Canvas Width="210" Height="210">
493 <Path Width="200" Height="200" Canvas.Left="5.00001" Canvas.Top="5" Stretch="Fill"
494 Data="F1 M 55.0069,5L 154.993,5C 182.611,5 205,27.3889 205,55.007L 205,
495 154.993C 205,182.611 182.611,205 154.993,205L 55.0069,205C 27.3889,205 5.00001,
496 182.611 5.00001,154.993L 5.00001,55.007C 5.00001,27.3889 27.3889,5 55.0069,5 Z "/>
497 <Path.Fill>
498 <LinearGradientBrush StartPoint="-0.0833325,0.999999" EndPoint="1.01,0.0166663">
499 <GradientStop Color="#FF493C3C" Offset="0"/>
500 <GradientStop Color="#FFFFFF" Offset="1"/>
501 </LinearGradientBrush>
502 </Path.Fill>
503 </Path>
504 <Canvas Width="210" Height="210" Clip="F1 M 55.0069,5L 154.993,5C 182.611,5 205,27.3889 205,
505 55.007L 205,154.993C 205,182.611 182.611,205 154.993,205L 55.0069,205C 27.3889,
506 205 5.00001,182.611 5.00001,154.993L 5.00001,55.007C 5.00001,27.3889 27.3889,5 55.0069,5 Z
507 <Image Source="Image/Help.png" Width="256" Height="256" Canvas.Left="0" Canvas.Top="0">
508 <Image.RenderTransform>
509 <TransformGroup>
510 <MatrixTransform Matrix="0.703125,0,0,0.703125,15.0001,15"/>
511 </TransformGroup>
512 </Image.RenderTransform>
513 </Image>
514 </Canvas>
515 <Path x:Name="Pathhelp" Width="210" Height="210" Canvas.Left="7.62939e-006" Canvas.Top="0"
516 Stretch="Fill" StrokeThickness="10" StrokeLineJoin="Round" Stroke="#FF00008B"
517 Data="F1 M 55.0069,5L 154.993,5C 182.611,5 205,27.3889 205,55.007L 205,154.993C 205,
518 182.611 182.611,205 154.993,205L 55.0069,205C 27.3889,205 5.00001,182.611 5.00001,
519 154.993L 5.00001,55.007C 5.00001,27.3889 27.3889,5 55.0069,5 Z "/>
520 </Canvas>
521 </Viewbox>
```

Чтобы как-то взаимодействовать с пользователем, получать от пользователя ввод с клавиатуры или мыши и использовать введенные данные в программе, необходимы элементы управления. WPF предлагает богатый стандартный набор элементов управления, например: ViewBox, Border, Canvas и т.д. На рисунке 6 изображен элемент для ввода пароля, а в листинге 2 приведен его код из формы «Авторизация» [5].



Рисунок 6 – Элемент ввода пароля

Листинг 2 – Код элемента для ввода пароля

```
18 <Border BorderBrush="#FF002C88" BorderThickness="1" HorizontalAlignment="Left" Height="36"
19     Margin="239,93,0,0" VerticalAlignment="Top" Width="259"
20     CornerRadius="10" Background="White">
21     <ComboBox x:Name="ComboBox1" HorizontalAlignment="Left" VerticalAlignment="Top" Width="2
22         BorderBrush="#00ACACAC" BorderThickness="0"
23         Background="#00000000" Margin="9,0,0,-1" Height="34" FontSize="16"/>
24 </Border>
25 <Border BorderBrush="#FF002C88" BorderThickness="1" HorizontalAlignment="Left" Height="36"
26     Margin="238,158,0,0" VerticalAlignment="Top" Width="259"
27     CornerRadius="10" Background="White">
28     <PasswordBox x:Name="PasswordBox2" HorizontalAlignment="Left" Margin="-1"
29         VerticalAlignment="Top"
30         Width="259" Height="36" Background="#00000000"
31         BorderBrush="#00ABADB3" FontSize="20" PasswordChar="•" CaretBrush="Black"
32         SelectionBrush="#FF0F2DB6" BorderThickness="0"
33         Padding="10,4,0,0" KeyDown="PasswordBox2_KeyDown" />
34 </Border>
93 <TextBox x:Name="TextBox1" HorizontalAlignment="Left" Height="34"
94     Margin="238,158,0,0" TextWrapping="Wrap" Text="Введите пароль"
95     VerticalAlignment="Top" Width="259" BorderBrush="{x:Null}"
96     Background="Transparent" Padding="10,7,0,0" FontSize="14"
97     SelectionBrush="{x:Null}"
98     PreviewMouseLeftButtonUp="TextBox1_PreviewMouseLeftButtonUp"
99     Foreground="#FF838383" BorderThickness="0" />
```

Выводы по разделу один

Анализируя предметную область проекта, сделал выводы, что для экономии времени на составление протокола необходимо этот процесс автоматизировать. Поэтому необходимо разработать единую базу данных, для хранения снимка и флюорографического протокола, что сэкономит не мало времени на поиск нужного результата обследования. Для решения этой проблемы было выбрано использовать следующие средства разработки:

- ramus Educational – case-система;
- SQL Server Management Studio – интегрированная среда для управления любой инфраструктурой SQL;
- C# WPF – язык программирования с технологией Windows Presentation Foundation;

- visual Studio 2013 Ultimate – среда разработки программного обеспечения.

Для шаблона формализованного протокола решено было использовать Microsoft Word так как он тесно взаимосвязан с языком С# имея для этого специальные библиотеки.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Информационное обеспечение предметной области

Протоколирование результатов рентгенологического исследования легких предполагает заключение, расшифровку результатов флюорографии врачом рентгенологом.

Протокол исследования должен быть написан словами понятными всем специалистам медицины, и он не должен содержать в себе слова узконаправленной специальности, не должен содержать необычных сокращений и витиеватых сравнений. Содержание протокола исследования зависит прежде всего от того, выявлены ли в легких патологические изменения.

Изменение легочного рисунка прежде всего связано с тем, что вокруг кровеносных сосудов возникает воспаление, вызывая утолщение их стенок, что конечно же отражается на рентгенограмме.

Схематическое изображение патологий придумали Л. Д. Линденбратен и Л. Б. Наумов. Эти изображения они описали в 1984 году в книге «Медицинская рентгенология». По их мнению, все рентгенологические изменения при различных заболеваниях органов дыхания можно детализировать в зависимости от их протяженности, формы, структуры, очертаний. Например: затенение, диссеминация, просветление. Эти рентгеновские симптомы врачи-рентгенологи на флюорографическом изображении схематически рисуют определенным образом [6]. На рисунке 7а и 7б изображены общепринятые схемы.

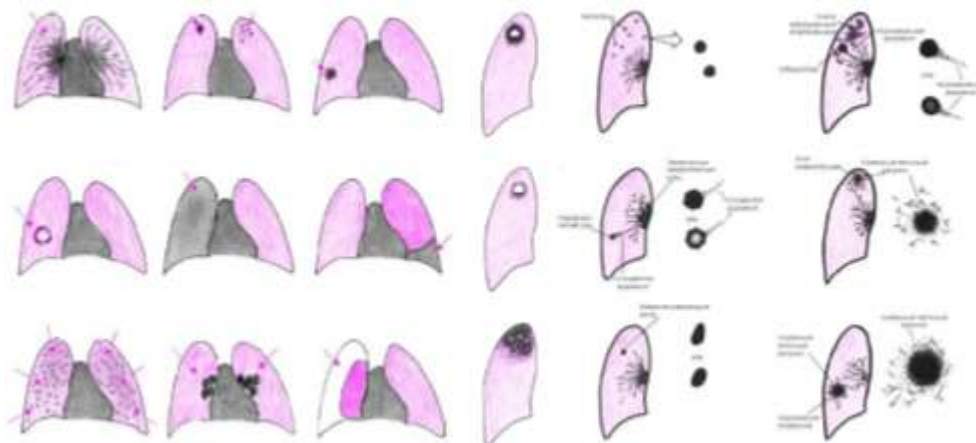


Рисунок 7а – Схемы патологий

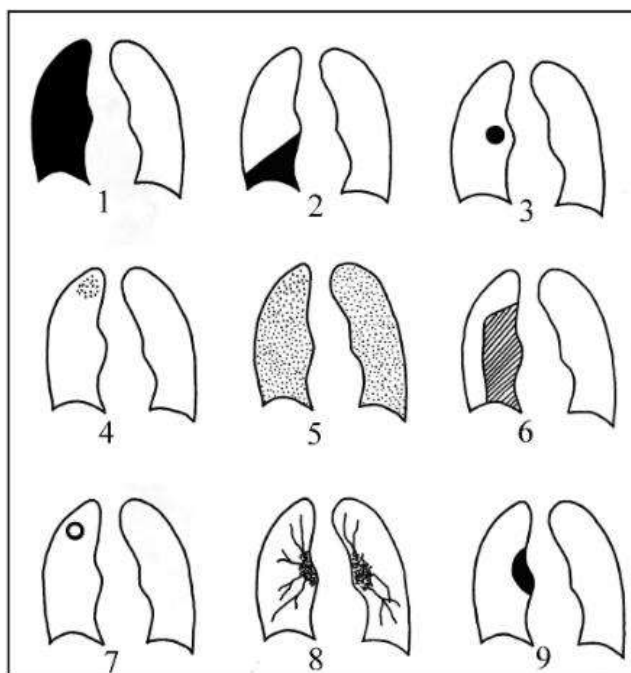


Рисунок 76 – Схематическое обозначение заболеваний легких

(1 – полное затемнение легочного поля, 2 – частичное затемнение, 3 – круглая тень, 4 – очаги и очаговая диссеминация, 5 – полная очаговая диссеминация, 6 – полное просветление, 7 – частичное просветление, 8 – изменение легочного рисунка, 9 – изменение корней легких)

Для автоматического создания протокола составляется таблица с описанием каждого симптома, из которой согласно схеме затем берутся данные.

2.1.1 Модель данных

Ядром совершенно любой базы данных является модель данных. Модель данных определяет логическую структуру базы данных и представляет объекты предметной области и их взаимосвязи.

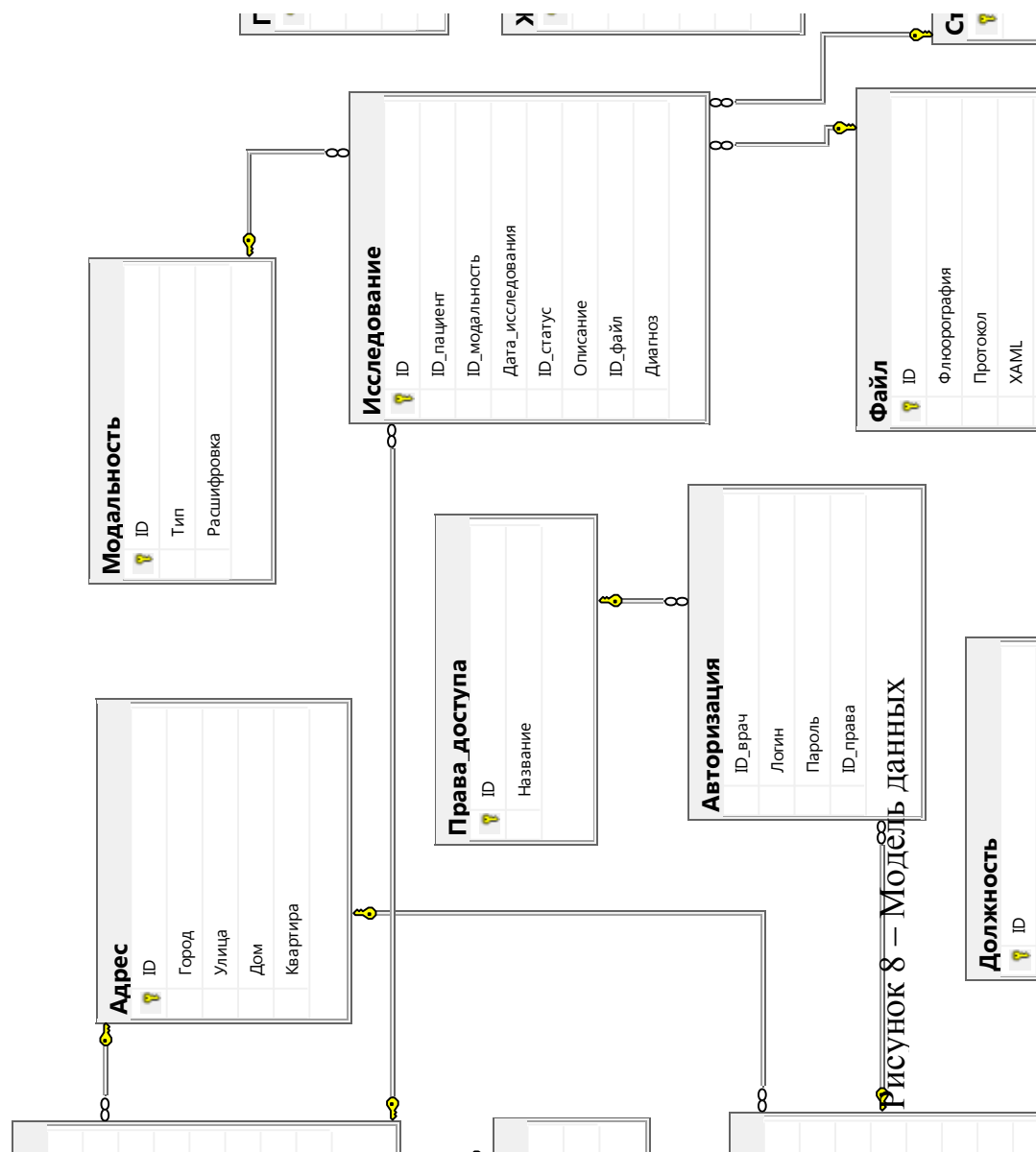
Стандартной концептуальной моделью на этапе проектирования является модель «сущность (Entity) – связь (Relation)» (ER-диаграмма). В данной диаграмме моделирование происходит на использовании графических средств. Диаграмма для программного обеспечения «Пульмон» сделана в программе Microsoft SQL Server Management Studio.

При рассмотрении бизнес-процессов моей предметной области был выявлен следующий перечень основных сущностей, которые отображены в таблице 7.

Таблица 7 – Сущность

№	Название	Краткое описание
1.	Пациент	Физическое лицо, обратившееся за медицинской помощью.
2.	Адрес	Адрес проживания пациента или врача.
3.	Пол	Мужчина или женщина
4.	Исследование	Таблица с обследованием пациентов
5.	Статус	Текущее состояние исследования
6.	Врач	Сотрудники, допущенные для пользования программным обеспечением
7.	Пиктограммы	Графическое отображение очагов, диссеминаций, теней
8.	Модальность	Классификация рентгенологического исследования
9.	Должность	Структурная единица штатного расписания медицинского учреждения
10.	Авторизация	Проверка логина и пароля сотрудника для доступа к ПО
11.	Права доступа	Условия доступа к определенным функциям
12.	Пиктограмма	Описание схематического изображения патологии
13.	Журнал посещений	История посещения программного обеспечения с сохранением даты и времени входа и выхода

ER-диаграмма служит для формирования общего взгляда на систему, для описания концептуальной схемы предметной области и для дальнейшей детализации информационной системы. ER-диаграмма имеет сущности, атрибуты и взаимосвязи проектируемой информационной системы. Далее на рисунке 8 представлена модель данных.



2.1.2 Информативно-справочная информация

Справочная информация в автоматизированной системе – это ядро единого информационного пространства.

Нормативно-справочная информация включает в себя: справочники, словари и классификаторы используемые в деятельности отделения.

На рисунке 9 показана какая нормативно-справочная информация используется в ПО «Пульмон»:

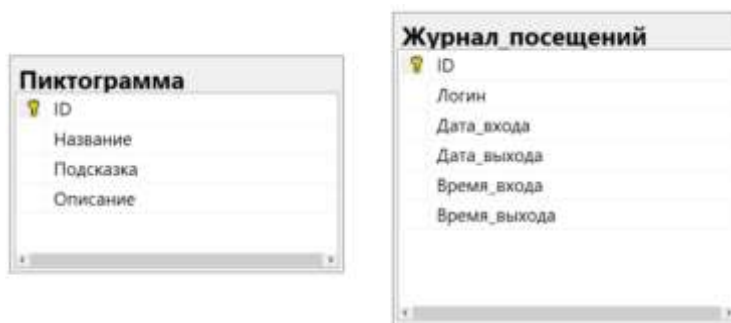


Рисунок 9 – Справочная информация для ПО

2.2 Характеристика результатов работы программного проекта

Результатом работы программы является:

1. Автоматически сформированное полное описание скиалогической картины флюорографического исследования. Пример представлен на рисунке 10.

На представленной рентгенограмме ОГК в прямой проекции: На обзорной рентгенограмме органов грудной клетки в нижней доле левого легкого (сегмент S10) на фоне деформированного легочного рисунка визуализируется ограниченное затемнение легочного поля. Диафрагма имеет куполообразную форму, четкий и ровный контур. Реберно-диафрагмальные синусы свободны, прозрачны, вершина синуса острая. Диафрагма не изменена. Положение диафрагмы нормальное. Тень средостения без особенностей, сердце и крупные сосуды без изменений. Легочный рисунок не деформирован, четкий. Корни легких структурны, не расширены.

Рисунок 10 – Пример задания

2. Сформированный предварительный диагноз пациента. Пример диагноза представлен на рисунке 11.

3.

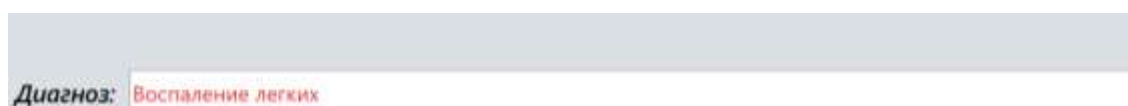


Рисунок 11 – Пример задания

4. Сформированный протокол флюорографического исследования (Приложение А). Данный протокол хранится в базе данных программы, который можно просмотреть, распечатать или скопировать на внешний носитель в любое время.

2.3 Требования к системе

2.3.1 Требования к функционированию и структуре системы

Информационная система должна включать в себя базу данных, хранящую информацию о пациентах, флюорографических исследований, фотографии, врачах, сотрудниках, заказчиках/поставщиках.

Интерфейс, программного обеспечения, должен быть удобным для заполнения баз данных информацией, для просмотра и создания отчетов. Должен обеспечивать комфортные условия пользователю в течение всего рабочего времени. Дизайн программы должен быть интуитивно понятным.

Функции, выполняемые системой:

- ввод в БД информацию о пациентах;
- ввод в БД информацию о врачах;
- формирование списка авторизации пользователей программы;
- сохранение документации в определенных каталогах;
- формирования списка исследований, необходимых для графического изображения скиалогической картины.

2.3.2 Требования к безопасности и надёжности ПО

Для обеспечения надежного и безопасного функционирования информационной системы должны выполняться следующие требования:

- система обязана сообщать пользователю об ошибках;
- система должна обеспечивать обработку ошибочных действий пользователя;
- ИС должна обеспечиваться контроль входной и выходной информации (авторизация пользователей).

2.4 Серверная часть программного обеспечения

Серверная часть базы данных выполняет управление и обслуживание данных, отвечает за сохранность и целостность информации. Для создания базы данных мною было выбрано MS SQL Server [7].

Для удобства было написано несколько хранимых процедур:

- поиск нужного пациента (листинг 5);
- создание отчета (листинг 6);
- выборка нужного ID флюорографического снимка (листинг 7);
- добавление исследования (листинг 3);
- добавление пациента (листинг 4);
- добавление адреса;
- добавление врача;
- удаление пациента;
- удаление врача.

Листинг 3 – Код добавления исследования

```
1 CREATE PROCEDURE [dbo].[Add_study]
2     @ID_пациент int,
3     @ID_модальность int,
4     @Дата_исследования nvarchar(30),
5     @ID_статус int,
6     @Описание nvarchar(4000),
7     @ID_файл int,
8     @Диагноз nvarchar(200),
9     @ID int out
10 AS
11 INSERT INTO Исследование (ID_пациент, ID_модальность,
12     Дата_исследования, ID_статус, Описание, Диагноз, ID_файл)
13 VALUES (@ID_пациент, @ID_модальность, @Дата_исследования,
14     @ID_статус, @Описание, @Диагноз, @ID_файл)
15 SET @ID=SCOPE_IDENTITY()
```

Листинг 4 – Код добавления пациента

```
2 CREATE PROCEDURE [dbo].[Add_patient]
3     @surname nvarchar(50),
4     @name nvarchar(20),
5     @patronymic nvarchar(30),
6     @date_of_birth date,
7     @id_gender int,
8     @id_address int,
9     @phone nvarchar(20)
10 AS
11 INSERT INTO Пациент
12 VALUES (@surname, @name, @patronymic, @date_of_birth, @id_gender,
13         @id_address, @phone)
```

Листинг 5 – Код поиска пациента

```
1 CREATE PROCEDURE [dbo].[Search]
2     @Фамилия varchar(30),
3     @Имя varchar(30),
4     @Отчество varchar(30),
5     @Дата varchar(30),
6     @Чек1 varchar(30)
7 AS
8 IF @Фамилия = '' Begin set @Фамилия = null end
9 IF @Имя = '' Begin set @Имя = null end
10 IF @Отчество = '' Begin set @Отчество = null end
11 IF @Дата = '' Begin set @Дата = null end
12 IF @Чек1 = '' Begin set @Чек1 = null end
13 SELECT s.ID, Фамилия, Имя, Отчество, Дата_рождения,
14         p.Название AS Пол, ID_адрес, Телефон FROM Пациент s
15 JOIN Пол p ON s.ID_пол=p.ID
16 where (Фамилия = @Фамилия or @Фамилия is null) And (Имя = @Имя or @Имя is null)
17         And (Отчество = @Отчество or @Отчество is null)
18         And (Дата_рождения = @Дата or @Дата is null) And (p.Название = @Чек1 or @Чек1 is null)
```

Листинг 6 – Код создания отчета

```
1 CREATE PROCEDURE [dbo].[GetReport]
2     @Beginning_Date DateTime,
3     @Ending_Date DateTime
4 AS
5 SELECT Диагноз, Дата_исследования, Пациент.Фамилия,
6         Пациент.Имя, Пациент.Отчество
7 FROM Исследование INNER JOIN Пациент ON Исследование.ID_пациент = Пациент.ID
8 WHERE Дата_исследования Between @Beginning_Date And @Ending_Date
```

Листинг 7 – Код выборки ID снимка

```
1 CREATE PROCEDURE [dbo].[GetImageByID]
2     @id int
3 AS
4 SELECT Фото FROM Файл WHERE ID = @id
```

2.5 Описание работы программы

2.5.1 Главная форма

1. Хранение документов

Первая часть документов, а именно протокол флюорографического обследования (word – файл) и файл разметки (.xaml), который хранит данные о графической схеме обследования хранится в корневом каталоге программы: «...\bin\Debug\Протокол» и «...\bin\Debug\XAML» соответственно.

Вторая часть документов хранится непосредственно в базе данных в таблице «Файл», это фото флюорографии и картинка с пиктограммами для веб-версии программы.

Снимок хранится в базе данных в поле, которое имеет тип image. Для сохранения картинки в таблицу используется код, представленный в листинге 8, а для его открытия из базы данных используется код, представленный в листинге 9 [8].

Листинг 8 – Сохранение снимка в БД

```
var path = Environment.CurrentDirectory + @"\Temp\" +
    pulmonForm.ID_FLUOROGRAPH.ToString() + ".bmp";
byte[] img = null;
System.IO.FileStream fs = new System.IO.FileStream
    (path, System.IO.FileMode.Open, System.IO.FileAccess.Read);
System.IO.BinaryReader br = new System.IO.BinaryReader(fs);
img = br.ReadBytes((int)fs.Length);
```

Листинг 9 – Открытие снимка из БД

```
System.IO.MemoryStream ms = new System.IO.MemoryStream(img);
System.Drawing.Bitmap bitmap = new System.Drawing.Bitmap(ms);
BitmapSource bitmapSource = System.Windows.Interop.Imaging.
    CreateBitmapSourceFromHBitmap(bitmap.GetHbitmap(), IntPtr.Zero,
    Int32Rect.Empty, BitmapSizeOptions.FromEmptyOptions());
ImageBrush image = new ImageBrush(bitmapSource);
inkCanvas1.Background = image;
image.Stretch = Stretch.Uniform;
```

2. Авторизация

Форма «Авторизации» представлена на рисунке 12.

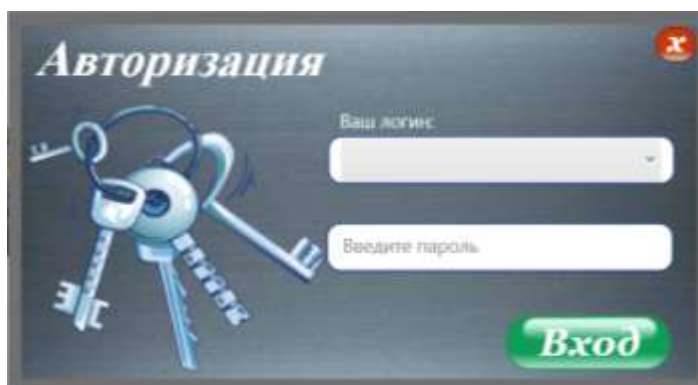


Рисунок 12 – Форма «Авторизация»

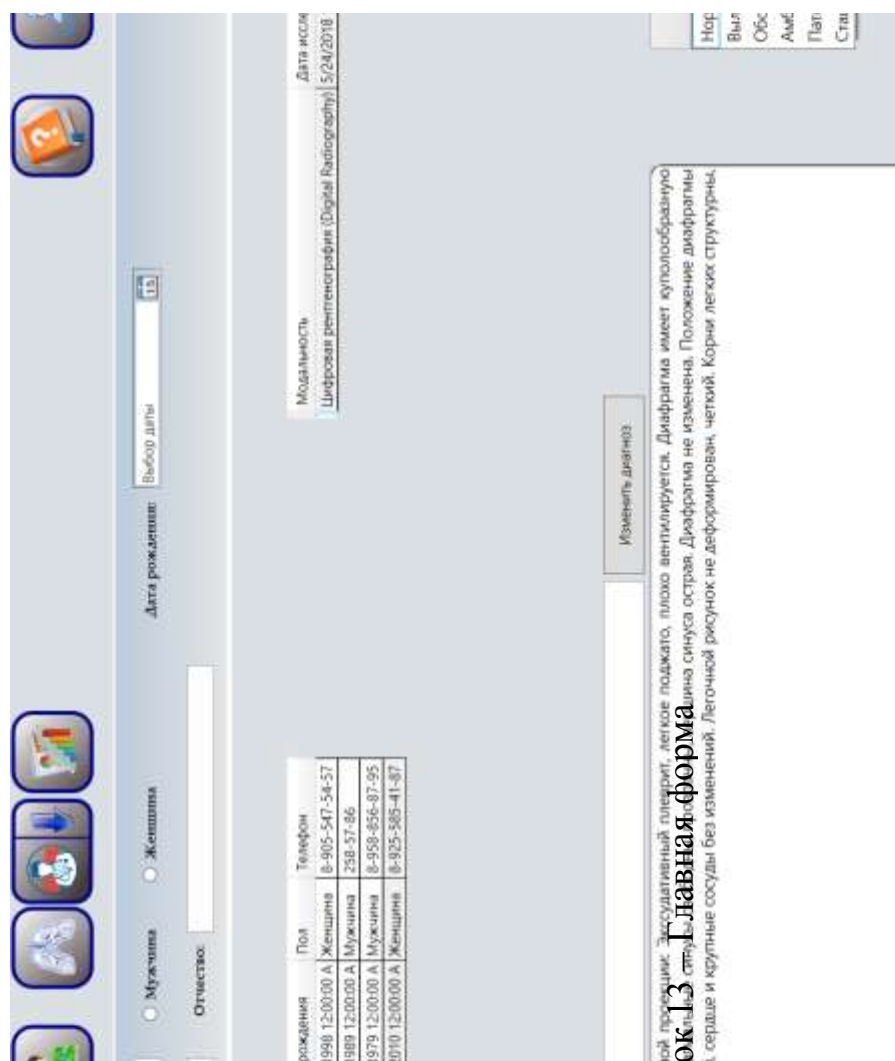
Чтобы пользователь начал работу с программным обеспечением ему необходимо пройти авторизацию, для этого необходимо ввести в соответствующие поля свои логин и пароль.

Логин и пароль выдается медицинскому работнику администратором, после того как он внес необходимую информацию о сотруднике в базу данных.

При нажатии кнопки «Вход», если логин и пароль введены правильно, происходит подключение к базе данных и запускается программа. Пароль хранится в базе данных в зашифрованном виде. Для этого используется хэш-код

SHA256, который является численным значением фиксированной длины и однозначно идентифицирует данные.

3. Главная форма. Главная форма изображена на рисунке 13.



Главная форма отображает следующую информацию:

- список пациентов;
- список исследований, который отображает обследование конкретного пациента;
- диагноз, который поставил врач – рентгенолог при обследовании;
- описание флюорографического обследования, если оно создано;
- для удобства создано быстрое открытие протокола флюорографического обследования, если он существует (рисунок 14);

Рисунок 13

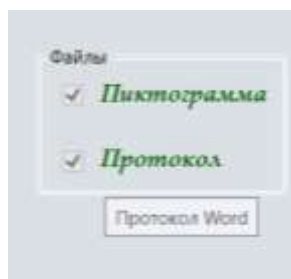


Рисунок 14 – Быстрый доступ к протоколу

Надпись пиктограмма отображает наличие схематического описания флюорографии пациента (зеленый цвет), если оно отсутствует, то надпись будет серого цвета.

Для облегчения пользования таблицей «Пациенты» сделана кнопка «Поиск» (рисунок 15), которая осуществляет поиск по фамилии, или по имени, или по отчеству, так же поиск может осуществляться либо по полу, либо по дате рождения пациента. Для сброса фильтра существует кнопка «Очистить форму».

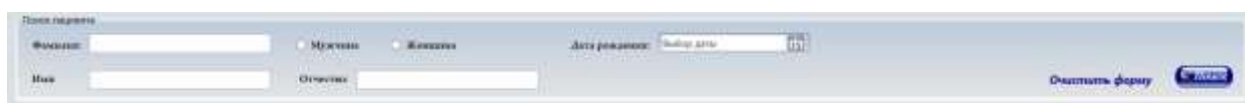


Рисунок 15 – Главная форма

4. Меню

Для того чтобы пользователю было удобно пользоваться, все кнопки имеют всплывающие подсказки, представлено на рисунке 16.



Рисунок 16 – Всплывающие подсказки

Переход осуществляется с помощью кнопок, расположенных сверху окна, которые представлены на рисунке 17.



Рисунок 17 –Переходы в программе

- при нажатии на «Меню» происходит отображение списка: «добавить пациента», «сменить пользователя», «закрыть файл» (рисунке 18).
- далее идут кнопки: «Добавить пациента», «Удалить пациента», «Изменить пациента»;
- следующая кнопка – «Создание протокола», которая переводит пользователя в редактор протоколов;
- при нажатии на «Исследование» происходит отображение списка (рисунок 19): «добавить исследование», «удалить исследование»;
- далее идет кнопка «Отчет»;
- затем кнопка «Сменить пользователя»;
- последняя кнопка – «Закрыть приложение».



Рисунок 18 – Отображение «Меню»

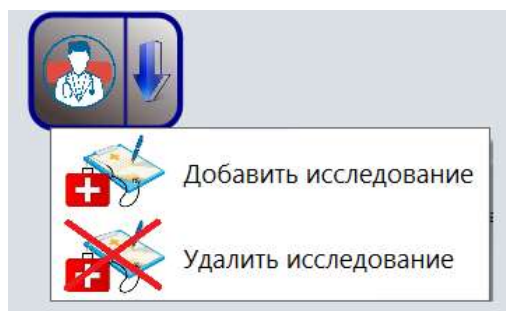


Рисунок 19 – Отображение «Исследование»

5. Пациент

При нажатии на кнопку «Пациент» откроется новая форма, с помощью которой можно добавить нового пациента. Данная форма представлена на рисунке 20.

Рисунок 20 – Форма «Добавить пациента»


В данной форме все поля являются обязательными для заполнения. Для удаления или редактирования пациента, в верхнем меню присутствуют соответствующие кнопки. Чтобы добавить адрес, необходимо щелкнуть на кнопку , после нажатия откроется новая форма, в которой можно будет добавлять, удалять и изменять адреса пациентов или врачей, рисунки 21, 22.



Рисунок 21 – Форма «Адреса»

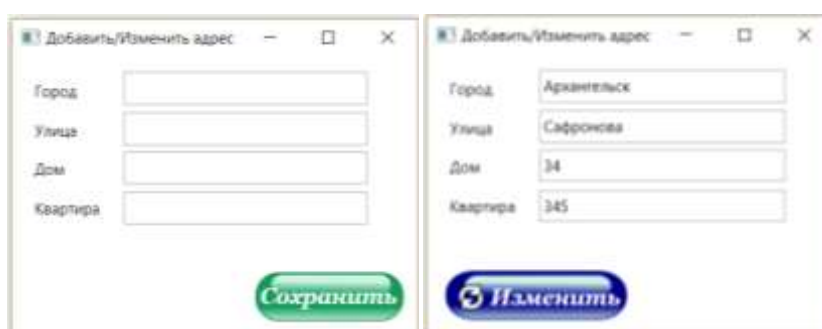


Рисунок 22 – «Добавить адрес» и «Изменить адрес»


6. Добавить исследование

Для того чтобы добавить исследование необходимо сначала выбрать пациента для которого создается исследование, а затем нажать на соответствующую кнопку «Добавить исследование», после чего откроется форма для создания исследования, рисунок 23.

Рисунок 23 – «Добавить исследование»

Все поля формы также обязательны для заполнения. При нажатии на кнопку «Удалить», происходит удаление выбранного исследования.

7. Отчет

С помощью кнопки «Отчет»  можно посмотреть статистику по диагнозам, с отбором конкретной даты, рисунок 24.

Диагноз	Дата исследования	Фамилия	Имя	Отчество	Диагноз
Закупорка сосудов левого плечевого	04.12.2017 0:00:00	Семязина	Оксана	Андреевна	1
	Всего				1
Пышечник	07.12.2017 0:00:00	Огурцов	Дмитрий	Валерьевич	1
	08.12.2017 0:00:00	Кудрешов	Максим	Витальевич	1
Всего					2
					3


Рисунок 24 – «Отчет» по диагнозам

С помощью кнопок, расположенных сверху формы (рисунок 25), можно отчет экспортировать в Excel, PDF, или в Word, либо просто его распечатать.



Рисунок 25 – кнопки «Отчет»

8. Сменить пользователя

При нажатии на кнопку «Сменить пользователя»  происходит закрытие и перезапуск программного продукта, чтобы пользователь мог войти под другим логином.

2.5.2 Администрирование программного проекта

В программе предусмотрена функция администрирования. Воспользоваться этой функцией может только администратор или пользователь, у которого есть такое право доступа.

Форма для администрации открывается, если пользователь вошел как «Администратор», представлено на рисунке 26.

Авторизация

Ваш логин:

Вход

Администрирование

Врачи:

Фамилия	Имя	Отчество	Дата рожден	Должность	Телефон
Добычин	Виктор	Сергеевич	12/15/1987 1:	Санитар	258-68-94
Арутюнян	Елена	Семеновна	12/16/1987 1:	Доктор	8-905-874-52-69
Сидоров	Иван	Викторович	9/20/2009 12:	Администратор	258-95-78

+ Добавить

Изменить

Удалить

Фамилия:

Имя:

Отчество:

Дата рождения: 15

Должность:


Телефон:

Адрес:

Журнал посещений **Логин и Пароль** **Запустить программу** **Сменить пользователя**

Рисунок 26 – форма «Администрирование»

На данной вкладке расположены кнопки: добавить, изменить, удалить, с помощью которых можно добавить или отредактировать врача. При нажатии «Удалить» происходит удаление выбранного врача, каскадно удаляется и его логин и пароль.

Рядом с выбором «Должность» есть кнопка , нажав на которую пользователь может добавить или отредактировать должность. Нажав на кнопку «Удалить», удаляется выбранная должность, рисунок 27.

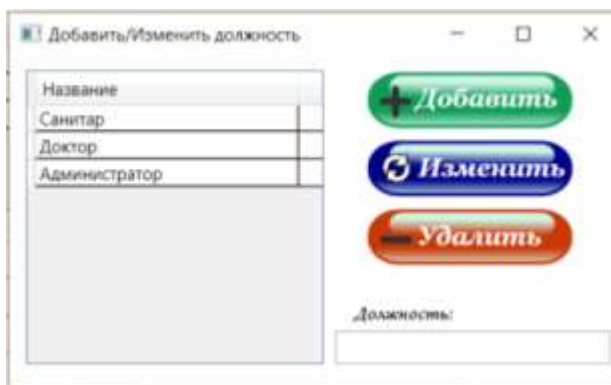


Рисунок 27 – «Добавить/Изменить» должность

При нажатии на кнопку «Логин и Пароль» происходит открытие формы, в которой администратор может редактировать или добавлять логин и пароль врача (рисунок 28).

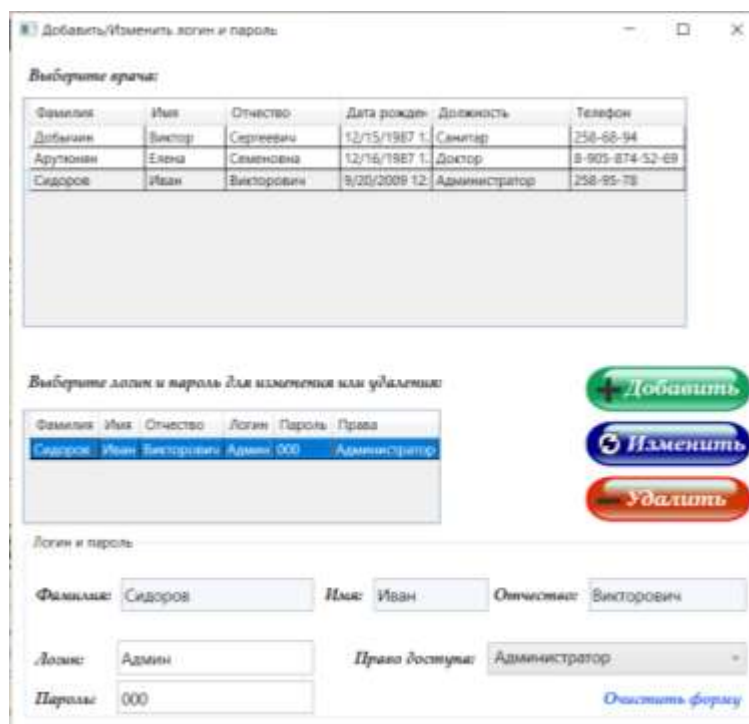
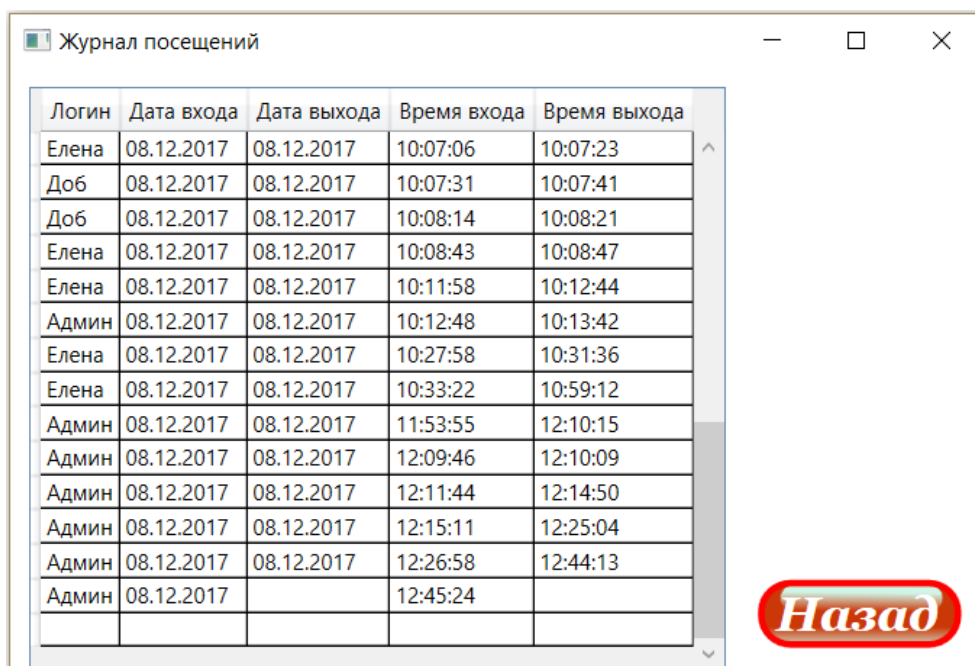


Рисунок 28 – Редактирование логина и пароля

В администрировании предусмотрен журнал посещений. Данный журнал хранит информацию о запускаящийся на компьютере программного продукта «Пульмон». Что самое главное – фиксируется точное время и день запуска, а также фиксируется точное время и дата выхода из программы. Чтобы его открыть необходимо выбрать пункт «Журнал посещений». Данный журнал представлен на рисунке 29.



Логин	Дата входа	Дата выхода	Время входа	Время выхода
Елена	08.12.2017	08.12.2017	10:07:06	10:07:23
Доб	08.12.2017	08.12.2017	10:07:31	10:07:41
Доб	08.12.2017	08.12.2017	10:08:14	10:08:21
Елена	08.12.2017	08.12.2017	10:08:43	10:08:47
Елена	08.12.2017	08.12.2017	10:11:58	10:12:44
Админ	08.12.2017	08.12.2017	10:12:48	10:13:42
Елена	08.12.2017	08.12.2017	10:27:58	10:31:36
Елена	08.12.2017	08.12.2017	10:33:22	10:59:12
Админ	08.12.2017	08.12.2017	11:53:55	12:10:15
Админ	08.12.2017	08.12.2017	12:09:46	12:10:09
Админ	08.12.2017	08.12.2017	12:11:44	12:14:50
Админ	08.12.2017	08.12.2017	12:15:11	12:25:04
Админ	08.12.2017	08.12.2017	12:26:58	12:44:13
Админ	08.12.2017		12:45:24	

Рисунок 29 – «Журнал посещений»

2.5.3 Редактор флюорографического протокола

Вторая часть программы – это создание протокола исследования. Прежде чем перейти по вкладке «Создать протокол» необходимо выбрать «Пациент» → «Исследование» (если нет, то создать), иначе перейти будет невозможно. Откроется форма «Создать протокол», представленная на рисунке 30.

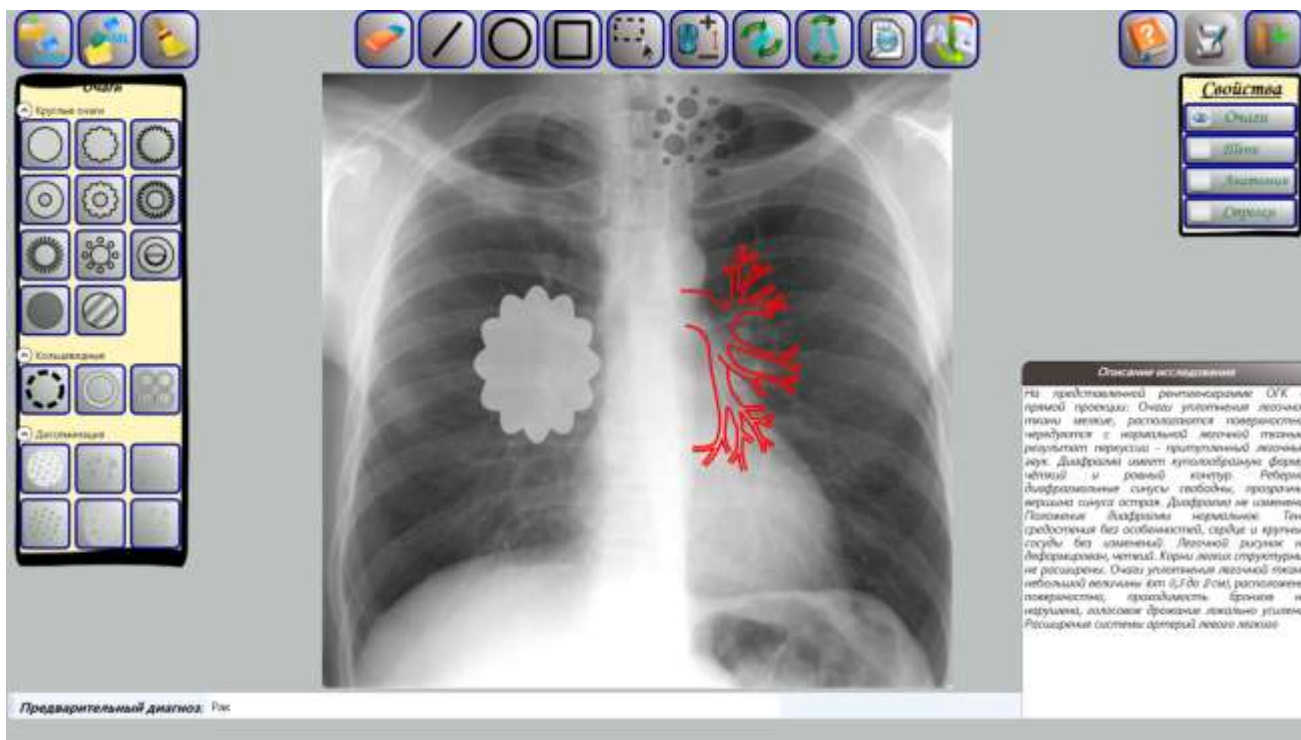


Рисунок 30 – «Создание протокола»

Для выбора пиктограмм, которые описывают патологию, необходимо нажать на одно из свойств, представленных на рисунке 31.



Рисунок 31 – Свойства пиктограмм

После выбора откроется набор кнопок с пиктограммами, каждая из которых имеет всплывающую подсказку. Данные кнопки показаны на рисунке 32 (а, б, в, г).

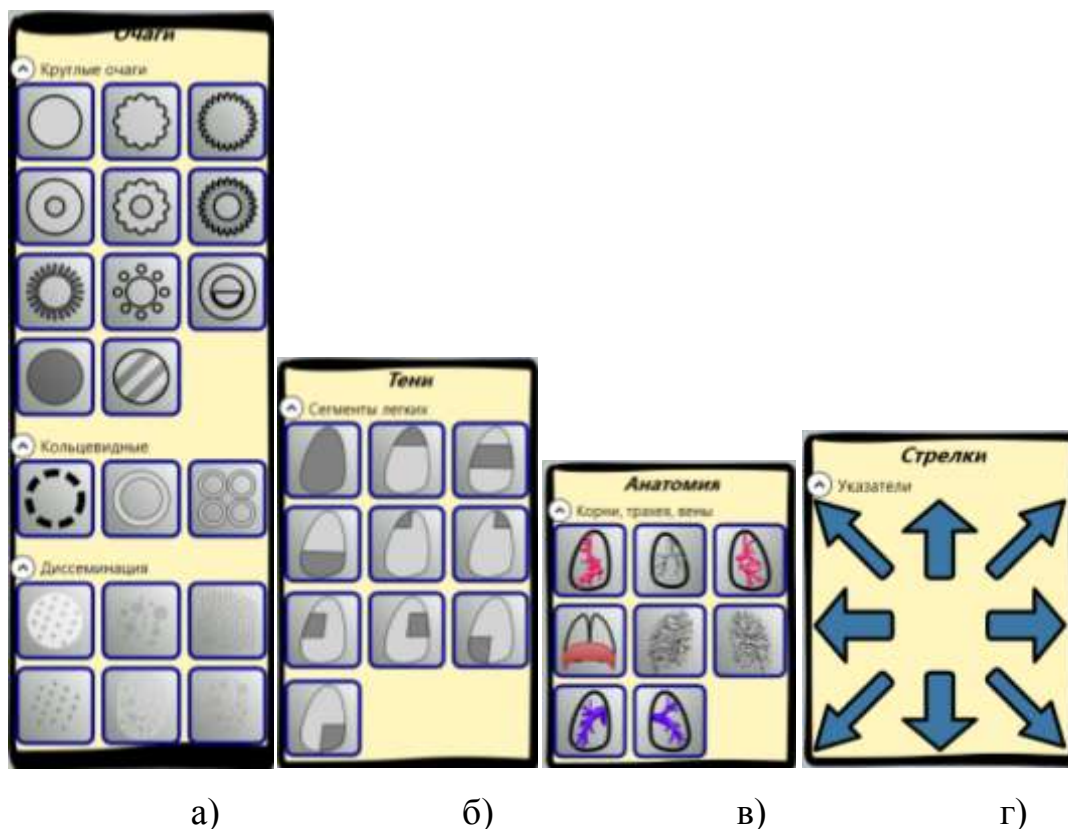




Рисунок 32 – Набор кнопок (а – очаги, б – тени, в – анатомия, г – стрелки)

Выбранная пиктограмма добавляется к флюорографии. Данную созданную схему можно сохранить (кнопка ) на диск в формате XAML или открыть (кнопка ) сохраненный файл для дальнейшей работы с ним. XAML – расширяемый язык разметки для приложений – основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft. Пример схемы показан на рисунке 33.

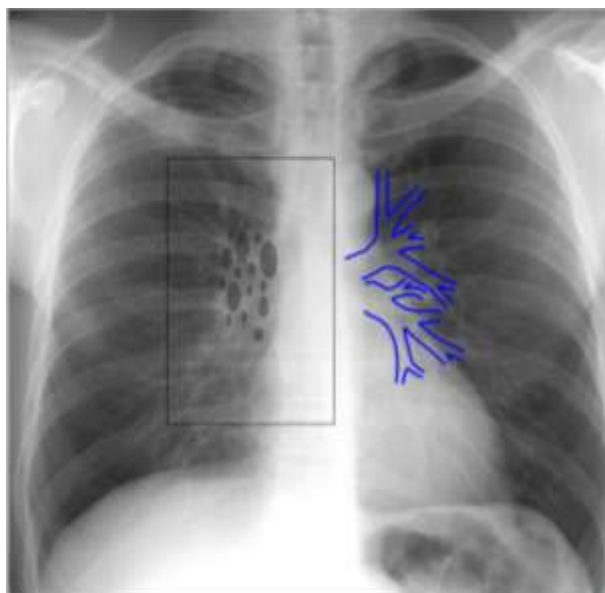


Рисунок 33 – Схематическое изображение патологий

После этого рентгенологу необходимо поставить предварительный диагноз, который записывается в строчку, расположенную снизу экрана, рисунок 34.



Рисунок 34 – Предварительный диагноз






Чтобы просмотреть предварительное флюорографическое описание рентгенологического исследования, необходимо нажать на , после чего автоматически сформируется описание. Данное описание можно редактировать, исправлять или дополнять. Результат показан на рисунке 35.



Рисунок 35 – Флюорографическое описание исследования

Данное окно можно свернуть, чтобы оно не мешало пользователю, для этого нужно нажать на кнопку: . У любой пиктограммы можно изменить размер, для этого следует нажать на кнопку:  и обвести либо выделить нужный элемент как на рисунке 36 (а, б). Также выделение необходимо если пользователь хочет удалить выделенный элемент. Для удаления по одному необходимо нажать на кнопку . Для очистки от всех пиктограмм существует кнопка .

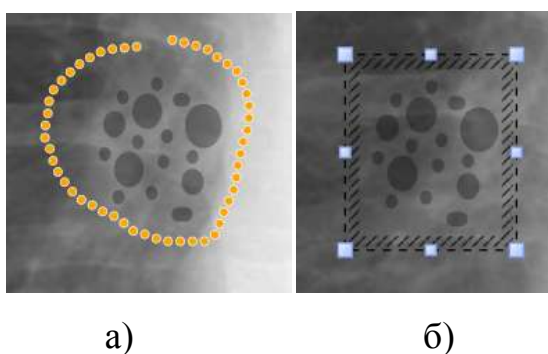



Рисунок 36 – Выделение схематического элемента (а – обвести, б – выделить)

Когда все подготовлено, можно приступить к созданию wordфайла флюорографического протокола рентгенологического исследования. Для его создания необходимо нажать на кнопку: , после этого программа свернется и откроется созданный файл протокола, который можно распечатать и отдать клиенту. Данный файл хранится в корневом каталоге программы. Пример данного файла показан в приложении А.

2.5.3 Web-версия информационной системы

Бывают ситуации, когда врачу необходимо проконсультироваться с другим специалистом, и чтобы его не звать к себе, была разработана Web-версия программного обеспечения, которая работает с одной базой данных, что и клиент –

серверное приложение. Интернет-версия разработана на платформе Web-приложений ASP.NET [7]. Данная платформа легко взаимодействует с базами данных.

Для входа в систему пользователю также как и на десктопном приложении необходимо авторизоваться. Форма авторизации представлена на рисунке 37.

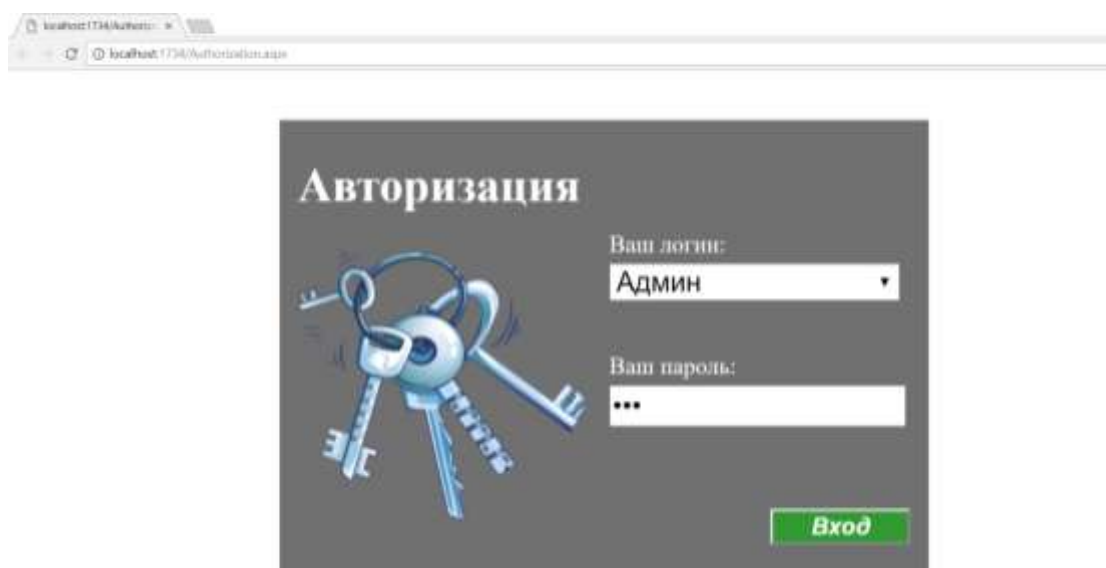


Рисунок 37 – Страница авторизации Web-версии приложения

Если пользователь ввел правильный логин и пароль он попадает на главную страницу системы. Здесь в зависимости от права доступа он может добавлять, удалять, редактировать пациентов и их исследования. Сверху находится удобный поиск, который позволяет быстро найти нужного пациента. Поиск осуществляется либо по фамилии, либо по имени, либо по дате рождения, либо сразу по нескольким вариантам. Главная страница информационной системы представлена на рисунке 38.

Программа помогает пользователю в случае ошибки всплывающими подсказками (рисунок 39). Для всплывающей подсказки был добавлен метод, который показан в листинге 10 [8].

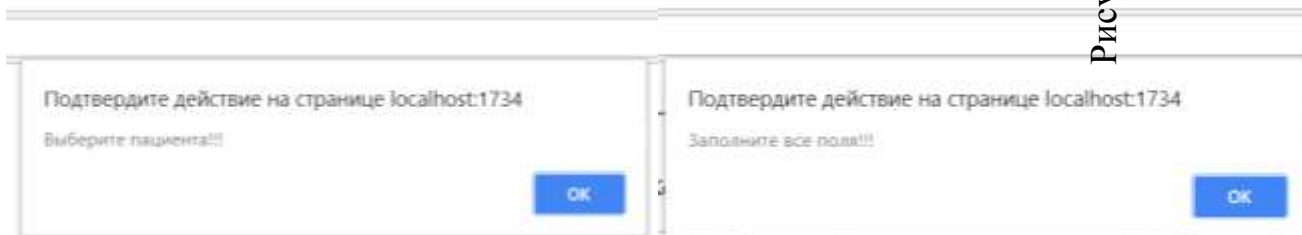


Рисунок 39 – Всплывающие подсказки приложения

Листинг 10 – Код, всплывающий подсказки приложения

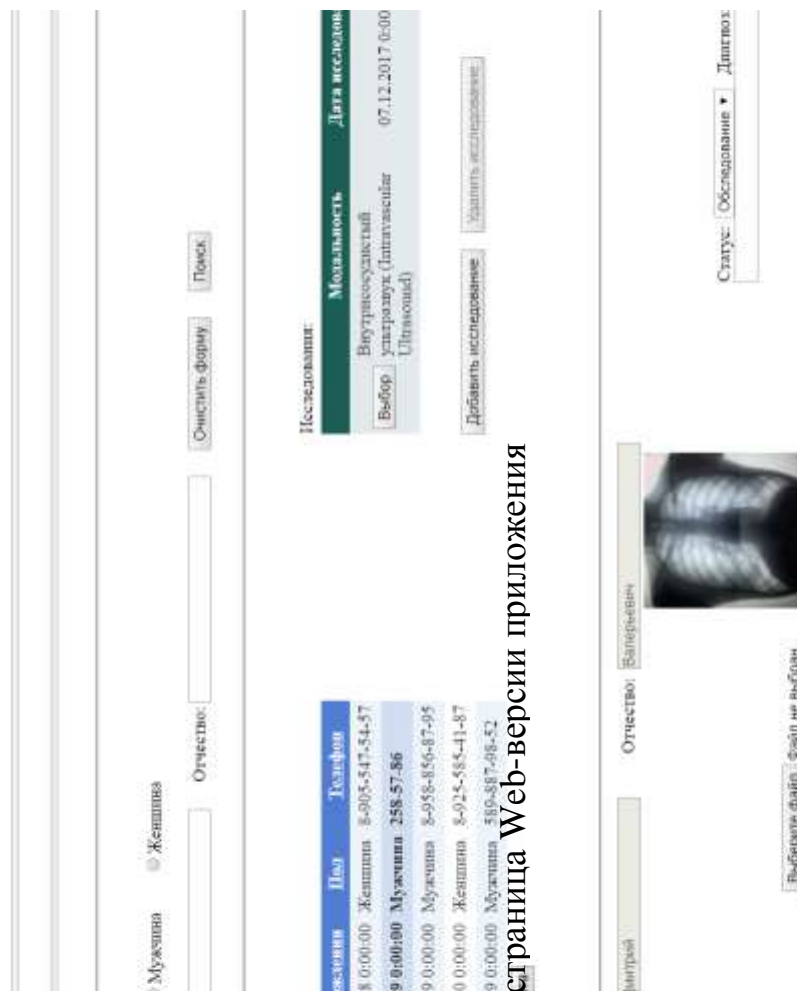


Рисунок 38 – Главная страница Web-версии приложения

```

209 <script type = "text/javascript">
210     function PatientDelete() {
211         var confirm_value = document.createElement("INPUT");
212         confirm_value.type = "hidden";
213         confirm_value.name = "confirm_value";
214         if (confirm("Вы хотите удалить пациента?")) {
215             confirm_value.value = "Yes";
216         }
217         else {
218             confirm_value.value = "No";
219             return;
220         }
221         document.forms[0].appendChild(confirm_value);
222     }
223 </script>

```

Здесь же можно добавить пациента, нажав на соответствующую кнопку, приложение откроет отдельную страницу для добавления пациента и его адреса (рисунок 40).

	Имя	Фамилия	Улица	Дом	Квартира
Выбор	Иванов	Иванов	Энгельса	53	45
Выбор	Минин	Минин	Ленина	4к	19
Выбор	Челышев	Краснопольский проспект		40	18
Выбор	Москал	ул. Пуршева		33	65
Выбор	Архангельск	Сафари		54	345
Выбор	Ветеринар	Пермякова		54	56
Выбор	Валюев	Горького		50к	130

Рисунок 40 – Страница для добавления пациента и адреса

Чтобы посмотреть описание и флюорографический снимок, необходимо выбрать пациента и его исследование, после чего нажать на кнопку «Посмотреть

полное описание протокола» на главной странице Web-приложения. Откроется новая страница с описанием и снимком необходимого исследования (рисунок 41).

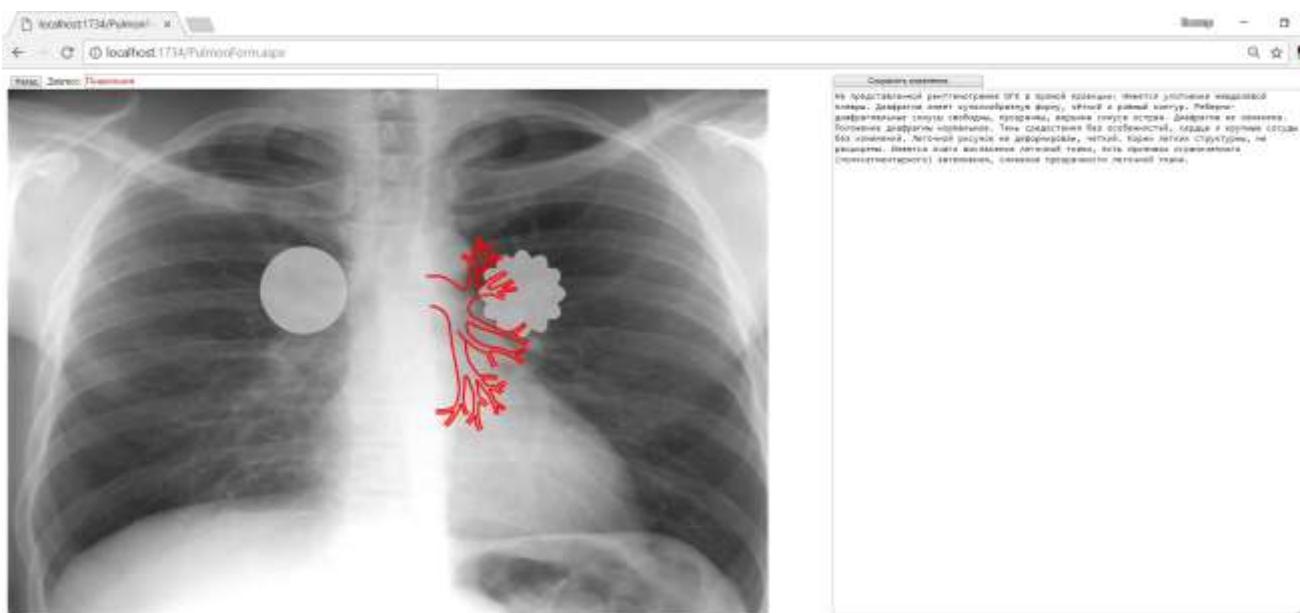


Рисунок 41 – Страница с описанием исследования

На этой странице врач может корректировать диагноз и описание флюорографического снимка, после чего нажав на кнопку «Сохранить изменения» данные будут сохранены в базе данных.

2.6 Структурная схема программы (дерево диалога)

Схема, описывающая дерево диалога, показывает собой схему использования программного обеспечения, в которой стрелками показан переход между формами, а в прямоугольниках показана, та форма, которая будет отображаться при переходе. Стрелки показывают переход при нажатии на соответствующую кнопку. С помощью овалов показаны кнопки, находящиеся на форме. Схема дерева диалога показана на рисунке 42.

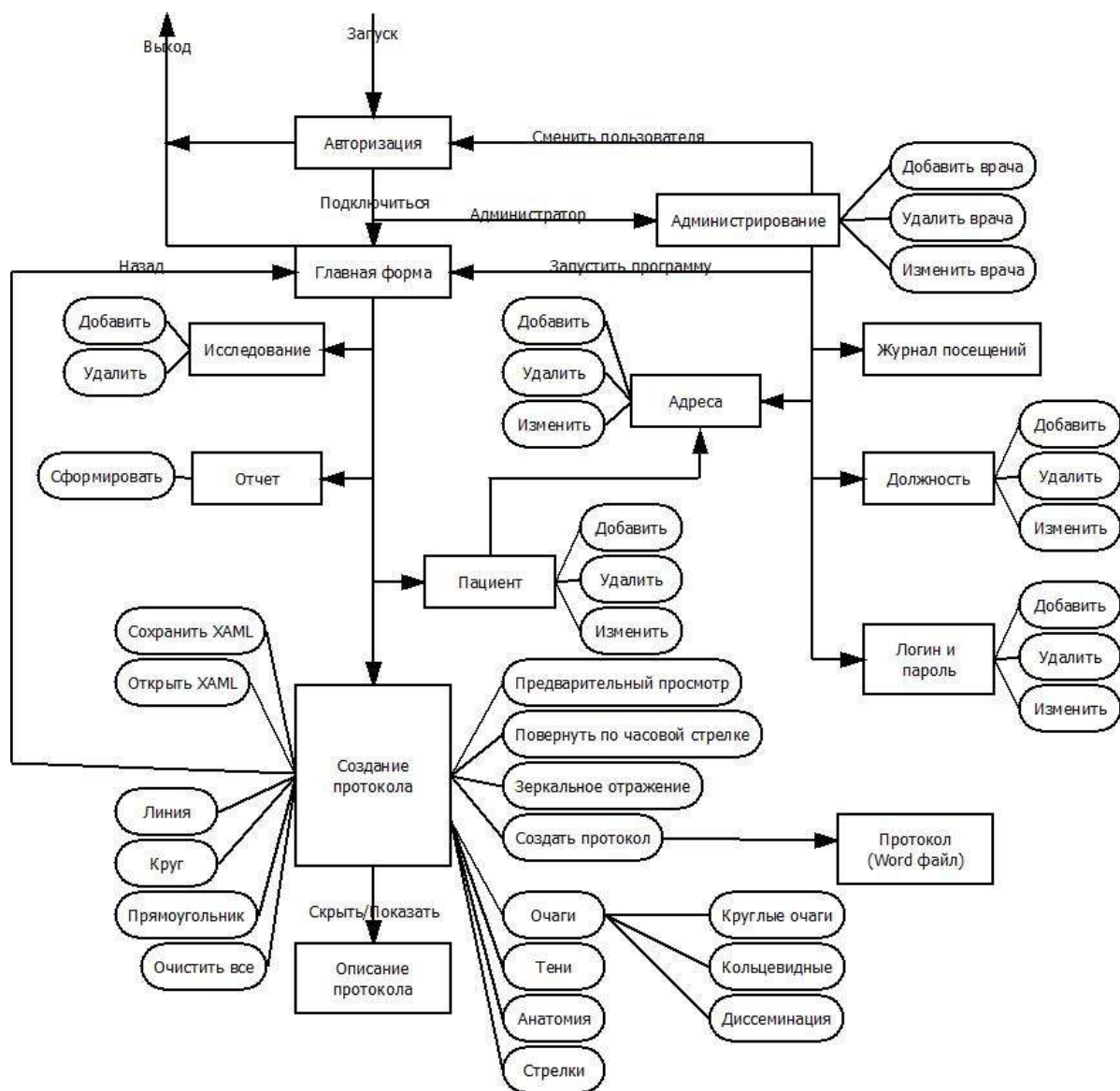


Рисунок 42 – Схема дерева диалога

2.7 Безопасность информационной системы

2.7.1 Обеспечение системы безопасности

Программное обеспечение имеет разграничение прав доступа на: гость, менеджер и администратор. Гость имеет право только на добавление пациентов и исследования. Он не может создавать, просматривать и редактировать протоколы. Эта функциональность подойдет для регистрации пациентов, для медсестер. Менеджер имеет право на все кроме администрирования, т.е. эта функциональность подходит для врачей-рентгенологов. Администратор включает в себя права дос-

тупа гостя и менеджера, плюс имеет право на редактирование, добавление, удаление пользователей, а также их логин и пароль.

Для целостности базы данных, а также для обеспечения безопасности информационной системы, предусмотрено следующее:

- пользователи системы имеют разграничения прав доступа;
- пароль пользователей системы хранится в зашифрованном виде;
- информация, которая вводится пользователем проверяется на сервере на соответствие типам и длине, для корректной работы базы данных;
- в базе данных поддерживается ссылочная целостность;
- только администратор базы данных может корректировать таблицы, которые не доступны для изменения пользователем;
- реализована система авторизации пользователей и фиксация их работы в системе;
- на все рабочие станции, использующие ПО, устанавливается антивирусная программа, которая защитит ПК от воздействия вирусов на информацию и саму систему.

2.7.2 Требования к прикладному обеспечению системы

Требования к информационной системе:

- система разработана для эксплуатации на ПК с установленной операционной системой Windows 7 и выше;
- библиотека .Net Framework должна содержать версию 3,5 и выше;
- для обеспечения безопасности необходимо установить антивирусную программу на все ПК, находящиеся в эксплуатации;
- система использует в качестве выходной информации документы, созданные в виде файла представляющего текст Word, следовательно, требуется установленный пакет MS Office.

2.7.1 Компоненты программного обеспечения

Программный продукт называется «Пульмон». Данное программное обеспечение включает в себя ряд программных единиц (по количеству форм), а также один модуль данных, на котором расположены компоненты доступа к базе данных. Программа включает в себя следующие формы:

- форма «ПульмонForm» является главной. Здесь заносится данные о пациентах и их обследовании;
- форма «MainWindow» служит для авторизации пользователей;
- форма «Add_patient» служит для добавления пациента в базу данных;
- форма «Add_address» служит для добавления адреса в базу данных;
- форма «Address» содержит информацию об адресах пациентов и врачей допущенных для работы с программой;
- форма «AuthorizationForm» служит для добавления в базу данных логина и пароля врача;
- форма «AdminForm» служит для администрирования;
- Форма «EditForm» служит для создания протокола рентгенологического исследования;
- форма «Position» служит для добавления должности в базу данных;
- форма «Visit_log» содержит информацию о посещении пользователей;
- форма «Study» служит для добавления в базу данных исследования;
- форма «Report» содержит отчет по диагнозам пациентов за конкретную дату.

Доступ к базе данных реализован через компонент `SqlConnection`, к нему подключен набор данных `SimpleDataSet`, каждому из которых, соответствует компонент `DataSource`.

2.8 Выбор технических средств для системы

Информационная система имеет клиент – серверную архитектуру, и, следовательно, для корректной работы ей необходим сервер системы, на котором будет располагаться и храниться база данных.

Ниже приведен состав и характеристики ПК, предназначенного для решений рассматриваемых задач:

1. Рабочее место:

Материнская плата: GIGABYTE GA-H110M-S2V, LGA 1151, Intel H110, mATX, Ret

процессор: Intel Core 2Duo E4700 2,2 ГГц;

жесткий диск: (320 – 500 Гбайт);

оперативная память: 4 Гбайт;

USB-порт;

SVGA дисплей.

2. Сервер:

процессор: CPU Intel Xeon E3-1240 V5 3.5 GHz/4core/1+8Mb/80W/8 GT/s
LGA1151

тип корпуса: ASUS TS300-E9-PS4 <90SV03EA-M04CE0>

количество процессоров: 1

интерфейс накопителя: Sata 6Gb/s

сетевая карта компьютера 4x10/100/1000Мбит/сек

частота процессора: 3500

материнская плата: S5520HCR "HanlanCreek"

жесткий диск: 600 Гб SAS 15000rpm 16Mb cache

привод: DVD-RW встроенный полноразмерный Serial ATA привод

Выводы по разделу два

На основании модели данных из первой главы была разработана база данных, предусмотрена целостность таблиц базы данных и составлены необходимые хранимые процедуры, для того чтобы логика решения оставалась на сервере.

Было разработано программное обеспечение с использованием технологии WPF. Для хранения паролей в базе данных, был выбран алгоритм SHA256 обеспечивающий хранение определенного хэш сообщения, которое сравнивается с зашифрованным хэш сообщением при вводе пароля пользователем.

Согласно книге «Медицинская рентгенология» Линденбратена Л.Д., Наумова Л.Б. были разработаны пиктограммы для обозначения патологий легких [6], а также разработано их описание, которое хранится в базе данных.

Для удобного удаленного просмотра обследования, была сделана Web-версия программного обеспечения, которое работает от одной базы данных что и клиент-серверное приложение.

3 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

Основная эффективность создания программного обеспечения:

- сокращение времени на создание протокола;
- срок хранения снимка и флюорографического протокола неограничен;
- сокращение времени на поиск необходимого файла;
- легко хранить и обмениваться между отделениями или медицинскими учреждениями.

Показатели эффективности бизнес-процессов представлены в таблице 8.

Таблица 8 – Показатели эффективности бизнес-процессов

Наименование \ Модель	AS-IS	TO-BE
1 время на составление флюорографического протокола (минут)	30	5 – 10
2 количество использованных материалов		
2.1 бумага А4 (упаковка)	15	7
2.2 картридж для печати (штук)	7	3
3 время на поиск нужной флюорографии (часов)	24 – 48	0,01
4 количество не принятых пациентов	40 – 60	0

Расчет расходов на разработку информационной системы

Все расходы по разработке информационной системы рассчитываются путем сложения всех расходов за месяц и умножения их на количество месяцев равных сроку разработки ИС.

Себестоимость разработки информационной системы сгруппирована по следующим статьям затрат:

- расходы на амортизацию;
- прочие расходы;

- расходы на материалы;
- расходы по зарплате.

1. Расходы на заработную плату

Расчет затрат по заработной плате разработчика рассчитываются по следующим двум формулам:

$$\text{ЗарботнаяПлата} = \text{Оклад} * \text{ЕСН} + \text{Оклад},$$

где ЕСН – Единый социальный налог, ЕСН = 0,3.

$$\text{Сумма} = \text{Загруженность} * \text{ЗарботнаяПлата}$$

Суммарные расходы по заработной плате программистов – разработчиков рассчитываются путем сложения расходов по заработной плате разработчика и расходов по заработной плате ведущего программиста – разработчика.

Расходы на заработную плату обоих разработчиков показан в таблице 9.

Таблица 9 – Расходы на заработную плату

Название	Оклад/руб.	ЕСН*Оклад/ руб.	Загруженность %	Общая сумма/руб.
Программист – разработчик	24 000	7 200	90	28 080
Ведущий программист – разработчик	36 000	10 800	10	4 680
Итого				32 760

2. Расходы на материалы

Расходы на материалы рассчитываются по следующей формуле:

$$\text{Сумма} = \text{ЗатратыНаЕдиницу} * \text{Количество}$$

Все затраты на бумагу и канцтовары рассчитываются путем сложения затрат на бумагу и затрат на канцелярские товары.

Подсчет себестоимости на материалы представлен в таблице 10.

Таблица 10 – Расчет материалов на разработку системы

Название	Ед. измерения	Расходы на ед./ руб.	Кол-во/шт.	Общая сумма/ руб.
Канцелярские товары (маркер, карандаш, ручка и т. д.)	комплект	120	1	120
Бумага А4	упаковка	280	1	280
Итого				400

3. Прочие расходы

Расходы на арендные платежи состоят из стоимости аренды и обслуживания помещения, управленческие расходы, а также бухгалтерские и прочие услуги

Арендные платежи рассчитываются по формуле:

$$\text{Аренда} = \text{ЗатратыНаАрендуЗа}1\text{м}^2 * S_{\text{м}}$$

где площадь рабочего места составляет 8 м²,

расходы на аренду за один кв. метр равны 600 рублей за месяц, следовательно

$$\text{Аренда} = 8 \text{ м}^2 * 600 \text{ руб./м}^2 = 4\,800 \text{ руб.}$$

4. Расходы на амортизацию

Расходы на амортизацию оборудования производятся по следующим двум формулам:

$$\text{Амортизация} = \text{НачальнаяСтоимость} / \text{СрокСлужбы(вМес.)}$$

$$\text{Сумма} = \text{Загруженность} * \text{АмортизацияМесяц}$$

ПК –ноутбук Acer Extensa EX2540-5325 стоимостью 32 600 руб., срок службы 4 года или 48 месяцев, следовательно, амортизация в месяц равна 32 600 руб./4*12 мес. = 679 руб./мес.

Программное обеспечение – Microsoft Windows 10 Pro, MS Office 2007, Microsoft SQL Server, Visual Studio 2013 Ultimate. Программное обеспечение, в состав которого входит: пакет Microsoft Windows 10 Pro + MSOffice 2007 стоимостью – 23 200 руб. Интегрированная среда разработки Visual Studio 2013 Ultimate – 27 500 руб., срок службы данного ПО составляет 1 год или 12 месяцев, следовательно, амортизация в месяц составляет (23 200 руб. + 27 500 руб.)/1*12 мес. = 3792 руб./мес.

Расходы на амортизацию оборудования представлен в таблице 11.

Таблица 11 – Расходы на амортизацию оборудования и ПО во время разработки информационной системы

Название	Стоимость/ руб.	Срок службы/ лет	Аморти- зация в мес., руб.	Загру- жен- ность/%	Общая сумма/ руб.
Ноутбук	32 600	4	679	90	611
ПО	50 700	1	4 225	90	3 802,5
Итого:					4 413,5

Общие расходы на разработку информационной системы

Общие расходы за 2 месяца разработки рассчитываются по следующей формуле:

$$\text{Затраты} = \text{ВремяРазработки} * \text{ЗатратыЗаМесяц}$$

Общие расходы на разработку информационной системы представлены в таблице 12.

Таблица 12 – Все расходы информационной системы

Название	Расходы в месяц/ руб.	Время разработ- ки/мес.	Общая сум- ма/руб.
Расходы на мате- риалы	400	2	800
Заработная плата	32 760	2	65 520
Амортизация	4 413,5	2	8 827
Прочие расходы	4 800	2	9 600
Итого			83 747

Выводы по разделу три

Хранение цифровых изображений на жестких дисках создает компактный и неограниченный рентгеновский архив отделения лучевой диагностики, обеспечивающий быстрый и удобный доступ через рабочее место одновременно нескольким врачам.

Цифровой архив хранится намного дольше, и вся информация всегда будет доступной и позволит тиражировать рентген снимок очень много, тем самым отпадает необходимость в повторных обследованиях, и проблема потери снимка больше не является проблемой.

После замены старого архива, который содержал пленочную и бумажную картотеку пациентов на современную цифровую картотеку, освобождается помещение медицинского учреждения для другого чего-нибудь и высвобождает обслуживающий персонал. Новый созданный современный цифровой архив (база данных) рентгенологического отделения позволяет контролировать объем различных видов обследований населения, позволяет выделить традиционные группы риска.

ЗАКЛЮЧЕНИЕ

Результатом моей дипломной работы по разработке автоматизированной системы для рентгенологического отделения: Модуль «создание формализованного протокола флюорографического исследования с использованием технологии WPF» является разработанное программное обеспечение «Пульмон».

Для этого было проведено исследование организационно-функциональной структуры рентгенологического отделения и построена ее AS-IS модель («как есть»). Также изучен новый материал по обширному API – интерфейсу, позволяющему создавать графические программы с насыщенным дизайном и интерактивностью.

В модели AS-IS были выявлены существенные недостатки, которые были подвержены реинжинирингу и устранены в модели TO-BE. Данные недостатки подверглись автоматизации, для этого была построена модель и диаграммы бизнес-процессов отделения.

В процессе разработки программного обеспечения была создана база данных в интегрированной среде разработки SQL Server, которая обрабатывает вводимую информацию и хранит ее. Было разработано несколько хранимых процедур.

Выполненная работа имеет экономическую эффективность и может быть использована в медицинских учреждениях, в качестве пробной версии. При необходимости программа может быть частично изменена и дополнена.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

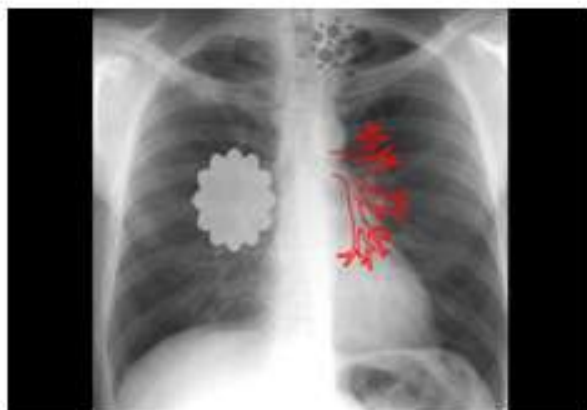
1. Описание функционала «Carestream Image Suite». – <https://www.carestream.com/en/us/medical/products/radiography/carestream-radiography-software/carestream-image-suite-v4>
2. Описание функционала программного комплекса «Pulmoscreen». – <http://www.pulmoscreen.ru>
3. Описание функционала АРМ «Ассистент диагноста». – <https://patientcards.ru>
4. Упражнения по системе моделирования и бизнес-процессов Ramus. – <http://www.intuit.ru/studies/courses/2195/55/lecture/15043>
5. Руководство по WPF. – <https://metanit.com/sharp/wpf>
6. Линденбрaтен Л.Д., Наумов Л.Б. Медицинская рентгенология. – Москва: 1984. – 384 с.
7. Сеть, ресурсы для разработчиков Microsoft. – <http://msdn.microsoft.com>
8. Описание разработки интерактивных приложений на языке C#. – <https://professorweb.ru>

ПРИЛОЖЕНИЕ А

Протокол флюорографического обследования

Челябинское Государственное учреждение здравоохранения

Отделение лучевой диагностики



№ исследования: 3

Дата исследования: 04.12.2017

ФИО пациента: Синичкина Оксана
Андреевна

Дата рождения: 18.02.1998

Пол: Женщина

Флюорографический протокол

в прямой проекции

На представленной рентгенограмме ОГК в прямой проекции: Очаги уплотнения легочной ткани мелкие, располагаются поверхностно, чередуются с нормальной легочной тканью; результат перкуссии - притупленный легочный звук. Диафрагма имеет куполообразную форму, чёткий и ровный контур. Реберно-диафрагмальные синусы свободны, прозрачны, вершина синуса острая. Диафрагма не изменена. Положение диафрагмы нормальное. Тень средостения без особенностей, сердце и крупные сосуды без изменений. Легочной рисунок не деформирован, четкий. Корни легких структурны, не расширены. Очаги уплотнения легочной ткани небольшой величины (от 0,5 до 2 см), расположены поверхностно, проходимость бронхов не нарушена, голосовое дрожание локально усилено. Расширение системы артерий левого легкого

Заключение: Рак

Врач-рентгенолог: Сидоров Иван Викторович

ПРИЛОЖЕНИЕ Б

Часть кода главной формы на языке C#

```
namespace Пульмон
{
    public partial class ПульмонForm : Window
    {
        string connectionString = "Data Source=.\SQLExpress; Initial Cata-
log=Пульмон; Integrated Security=True";
        MainWindow MW = new MainWindow();
        //ID Пациента
        public static int ID_patient;
        public int ID_PATIENT
        { get { return ID_patient; } }
        //ID - Фотографии(флюорографии) = ID исследования
        public static int id_fluorograph;
        public int ID_FLUOROGRAPH
        { get { return id_fluorograph; } }
        //ID - Файла
        public static int id_file;
        public int ID_FILE
        { get { return id_file; } }
        //Протокол
        public static string protokol = "";
        public string PROTOKOL
        { get { return protokol; } }
        //xaml_text
        public static string xaml_text = "";
        public string XAML_TEXT
        { get { return xaml_text; } }
        public static bool patient_edit = false;
        public bool PATIENT_EDIT
        { get { return patient_edit; } }
        //ID Адрес
        #region ID_ADDRESS
        public static int id_address;
        public int ID_ADDRESS
        { get { return id_address; }
          set { id_address = value; } }
        #endregion ID_ADDRESS
        public ПульмонForm()
        { InitializeComponent(); }
        SolidColorBrush myStroke1 = new SolidColorBrush(Colors.Silver);
        SolidColorBrush myStroke2 = new SolidColorBrush(Colors.DarkBlue);
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            if (MW.ID_RIGHTS != "1")
            {
                viewbox_admin.Visibility = System.Windows.Visibility.Hidden; }
            if (MW.ID_RIGHTS == "3")
```

```

{
Viewboxflur.Visibility = System.Windows.Visibility.Hidden;
viewbox_groupbox.Visibility = System.Windows.Visibility.Hidden;
button_diagnosis.Visibility = System.Windows.Visibility.Hidden;
button_status.Visibility = System.Windows.Visibility.Hidden;
    combobox3.Visibility = System.Windows.Visibility.Hidden;
}
Load();
SqlConnection con = new SqlConnection(connectionString);
try
{
    con.Open();
    SqlCommand cmd = new SqlCommand("SELECT * FROM Статус", con);
    SqlDataReader reader;
    reader = cmd.ExecuteReader();
    while (reader.Read())
    { combobox3.Items.Add(reader["Название"].ToString()); }
    reader.Close();
}
catch (Exception ex) { MessageBox.Show(ex.Message); }
finally { if (con != null) con.Close(); }
}
private void Load()
{
    SqlConnection conn = new SqlConnection(connectionString);
    conn.Open();
    SqlDataAdapter da = new SqlDataAdapter("SELECT s.ID, Фамилия,
Имя, Отчество, Дата_рождения, р.Название AS Пол, ID_адрес, Телефон FROM Па-
циент s JOIN Пол р ON s.ID_пол=р.ID", conn);
    DataSet ds = new DataSet();
    da.Fill(ds, "Пациент");
    Пациент.ItemsSource = ds.Tables["Пациент"].DefaultView;
    conn.Close();
    foreach (var oColumn in Пациент.Columns)
    {
        if (oColumn.Header.ToString() == "Фамилия")
            oColumn.Width = 100;
        if (oColumn.Header.ToString() == "Имя")
            oColumn.Width = 70;
        if (oColumn.Header.ToString() == "Отчество")
            oColumn.Width = 100;
        if (oColumn.Header.ToString() == "Дата_рождения")
        {
            oColumn.Width = 120;
            oColumn.Header = "Дата рождения";
        }
        if (oColumn.Header.ToString() == "Пол")
            oColumn.Width = 70;
    }
}

```

```

        if (oColumn.Header.ToString() == "Телефон")
            oColumn.Width = 100;
    }
    Пациент.Columns[0].Visibility = Visibility.Hidden;
    Пациент.Columns[6].Visibility = Visibility.Hidden;
}
private void Viebox1_MouseEnter(object sender, MouseEventArgs e)
{ Path1.Stroke = myStroke1; }
private void Viebox1_MouseLeave(object sender, MouseEventArgs e)
{ Path1.Stroke = myStroke2; }
private void Viewboxflur_MouseEnter(object sender, MouseEventArgs e)
{ Pathflur.Stroke = myStroke1; }
private void Viewbox1_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{ ComboBox1.IsDropDownOpen = true; }
private void image_help_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{ HelpExecuted(null, null); }
private void button_help_Click(object sender, RoutedEventArgs e)
{ HelpExecuted(null, null); }
private void Viewboxflur_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{ if (Исследование.SelectedIndex == -1)
    {
        MessageBox.Show("Выберите исследование!", "Внимание",
        MessageBoxButton.OK, MessageBoxImage.Question);
    }
    else
    {
        DataRowView rowView = Исследование.SelectedValue as DataRowView;
        id_fluorograph = (int)rowView[0];
        id_file = (int)rowView[7];
        EditForm ef = new EditForm();
        this.Hide();
        ef.ShowDialog();
        this.Show();
    }
}
private void Visit_Exit()
{ //Запись в журнал посещения-----
    MainWindow m = new MainWindow();
    SqlConnection con = new SqlConnection(connectionString);
    SqlCommand cmd1 = new SqlCommand("UPDATE Журнал_посещений SET
Дата_выхода = '" + DateTime.Today.ToString("d") + "', Время_выхода = '" +
DateTime.Now.ToLongTimeString() + "' WHERE ID = '" + m.ID_VISIT_LOG + "'",
con);

    con.Open();
    cmd1.ExecuteNonQuery();
    con.Close();
}

```

```

        m.Close(); }
private void Load_study()
{
    DataGridView rowView = Пациент.SelectedValue as DataGridView;
    ID_patient = (int)rowView[0];
    SqlConnection conn = new SqlConnection(connectionString);
    conn.Open();
    SqlDataAdapter da1 = new SqlDataAdapter("SELECT i.ID,
ID_пациент, m.Расшифровка AS Модальность, Дата_исследования AS " + "'Дата
исследования'" + ", s.Название AS Статус, Описание, Диагноз, ID_файл FROM
Исследование i JOIN Статус s ON i.ID_статус=s.ID JOIN Модальность m ON
i.ID_модальность=m.ID WHERE ID_пациент = " + ID_patient, conn);
    DataSet ds1 = new DataSet();
    da1.Fill(ds1, "Исследование");
Исследование.ItemsSource = ds1.Tables["Исследование"].DefaultView;
    conn.Close();
    foreach (var oColumn in Исследование.Columns)
    {
        if (oColumn.Header.ToString() == "ID_модальность")
            oColumn.Width = 120;
        if (oColumn.Header.ToString() == "Дата_исследвания")
            oColumn.Width = 120;
        if (oColumn.Header.ToString() == "Статус")
            oColumn.Width = 100;
        if (oColumn.Header.ToString() == "Описание")
            oColumn.Width = 100;
    }
    Исследование.Columns[0].Visibility = Visibility.Hidden;
    Исследование.Columns[1].Visibility = Visibility.Hidden;
    Исследование.Columns[7].Visibility = Visibility.Hidden;
}
private void vieboxremovepatient_MouseEnter(object sender,
MouseEventArgs e)
{ Pathremovepatient.Stroke = myStroke1; }
private void vieboxremovepatient_MouseLeave(object sender,
MouseEventArgs e)
{ Pathremovepatient.Stroke = myStroke2; }
#region Поиск пациента
private void ViewboxSearch_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    string check = "";
    if (radiobutton_men.IsChecked == true)
        check = "Мужчина";
    if (radiobutton_girl.IsChecked == true)
        check = "Женщина";
    SqlConnection con = new SqlConnection(connectionString);
    con.Open();

```

```

SqlCommand cmd = new SqlCommand("Search", con);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@Фамилия", textbox_family.Text);
cmd.Parameters.AddWithValue("@Имя", textbox_name.Text);
cmd.Parameters.AddWithValue("@Отчество", textbox_patronymic.Text);
cmd.Parameters.AddWithValue("@Дата", datepicker_birth.Text);
cmd.Parameters.AddWithValue("@Чек1", check);
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
DataTable dt = new DataTable();
adapter.Fill(dt);
Пациент.ItemsSource = dt.DefaultView;
foreach (var oColumn in Пациент.Columns)
{
    if (oColumn.Header.ToString() == "Фамилия")
        oColumn.Width = 100;
    if (oColumn.Header.ToString() == "Имя")
        oColumn.Width = 70;
    if (oColumn.Header.ToString() == "Отчество")
        oColumn.Width = 100;
    if (oColumn.Header.ToString() == "Дата_рождения")
    {
        oColumn.Width = 120;
        oColumn.Header = "Дата рождения";
    }
    if (oColumn.Header.ToString() == "Пол")
        oColumn.Width = 70;
    if (oColumn.Header.ToString() == "Телефон")
        oColumn.Width = 100;
}
Пациент.Columns[0].Visibility = Visibility.Hidden;
Пациент.Columns[6].Visibility = Visibility.Hidden;
con.Close();
}
#endregion Поиск пациента
private void viewboxuser_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    MessageBoxResult result = System.Windows.MessageBox.Show("Вы
действительно хотите сменить пользователя?", "Смена пользователя",
MessageBoxButton.YesNo, MessageBoxImage.Question);
    if (result == MessageBoxResult.No)
    { return; }
    else
    {
        Visit_Exit();
        Sys-
tem.Diagnostics.Process.Start(Application.ResourceAssembly.Location);
        Application.Current.Shutdown(); } }

```

ПРИЛОЖЕНИЕ В

Часть кода формы создания протокола на языке C#

```
namespace Пульмон
{
    public partial class EditForm : Window
    {
        SqlConnection con = new SqlConnection("Data Source=.\SQLExpress;
Initial Catalog=Пульмон; Integrated Security=True");
        SolidColorBrush myStroke1 = new SolidColorBrush(Colors.Silver);
        SolidColorBrush myStroke2 = new SolidColorBrush(Colors.DarkBlue);
        SolidColorBrush myStroke3 = new SolidColorBrush(Colors.Black);
        int chetnechet = 0;
        int chetnechet_anat = 0;
        int chetnechet_shadows = 0;
        int chetnechet_arrows = 0;
        int descript = 0;
        bool circle = false;
        bool rectangle = false;
        bool pensil = false;
        ПульмонForm pulmonForm = new ПульмонForm();
        Description description;
        List<UIElement> elements = new List<UIElement>();
        Line line;
        Ellipse ellipse;
        Rectangle rect;
        bool IsDrawing = false;
        Point point;
        bool zoom = false;
        int chetnechet_zoom = 0;
        Random rand = new Random();
        private int _angle = 90;
        private static int scaletransform = 1;
        public EditForm()
        {
            InitializeComponent();
            stackpanel2.Visibility = Visibility.Hidden;
            stackpanelanat.Visibility = Visibility.Hidden;
            stackpanelshadows.Visibility = Visibility.Hidden;
            stackpanelarrows.Visibility = Visibility.Hidden;
        }
        #region Загрузка формы
        private void EditForm1_Loaded(object sender, RoutedEventArgs e)
        {
            description = new Description();
            description.Show();
            InkCanvas.BackgroundImageAdd();
            if (pulmonForm.XAML_TEXT != "")
            {

```



```

        MessageBoxResult result =
System.Windows.MessageBox.Show("Имеется созданный файл пиктограмм", "Внима-
ние", MessageBoxButton.OK);
        if (result == MessageBoxResult.OK)
        {
            viewboxopen_MouseLeftButtonUp(null, null);
        }
    }
}
#endregion Загрузка формы
#region Ластик
private void vieboxeraser_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    inkCanvas1.Focus();
    SendKeys.SendWait("{DEL}");
}
private void vieboxeraser_MouseEnter(object sender, Sys-
tem.Windows.Input.MouseEventArgs e)
{
    Path_Eraser.Stroke = myStroke1;
}
#endregion Ластик
#region CopyPath / CopyViewbox
private Path CopyPath(Path p)
{
    string xaml = System.Windows.Markup.XamlWriter.Save(p);

    if (xaml == null)
    {
        return null;
    }
    using (System.IO.MemoryStream stream = new Sys-
tem.IO.MemoryStream(xaml.Length))
    {
        using (System.IO.StreamWriter sw = new System.IO.StreamWriter(stream))
        {
            sw.Write(xaml);
            sw.Flush();
            stream.Seek(0, System.IO.SeekOrigin.Begin);
            Path clonePath = System.Windows.Markup.XamlReader.Load(stream) as Path;
            return clonePath;
        }
    }
}
#endregion CopyPath / CopyViewbox
#region viewboxanatomia
private void viewboxanatomia_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)

```

```

{
    if (chetnechet_anat % 2 == 0)
    {
        stackpanelanat.Visibility = Visibility.Visible;
        imageanat.Source = new BitmapImage(new
Uri(@"Image/vision.png", UriKind.Relative));
        chetnechet_anat++;
    } else
    {
        stackpanelanat.Visibility = Visibility.Hidden;
        imageanat.Source = new BitmapImage(new
Uri(@"Image/unvision.png", UriKind.Relative));
        chetnechet_anat++;
    } }
#endregion viewboxanatomia
#region Zoom
private int scale = 0;
const double ScaleRate = 1.1;
private void inkCanvas1_MouseWheel(object sender,
MouseWheelEventArgs e)
{
    if (e.Delta > 0 && zoom == true)
    {
        scaleTransform1.CenterX = e.GetPosition(inkCanvas1).X;
        scaleTransform1.CenterY = e.GetPosition(inkCanvas1).Y;
        scaleTransform1.ScaleX *= ScaleRate;
        scaleTransform1.ScaleY *= ScaleRate;
        scale++;
    }
    else
    {
        if (scale != 0 && zoom == true)
        {
            scaleTransform1.CenterX = e.GetPosition(inkCanvas1).X;
            scaleTransform1.CenterY = e.GetPosition(inkCanvas1).Y;
            scaleTransform1.ScaleX /= ScaleRate;
            scaleTransform1.ScaleY /= ScaleRate;
            scale--;
        }
    }
}
#endregion Zoom
private void vieboxedit_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
{
    if (textbox_diagnosis.Text == "")
    {
        System.Windows.MessageBox.Show("Заполните предварительный диагноз");
        return;
    }
}

```

```

if (description.textbox1.Text == "")
    viewBox_see_MouseLeftButtonUp(null, null);
string name = "", surname = "", patronymic = "", date = "",
date2 = "", sex = "", conclusion = "", id = "";
int id_patient = 0, id_sex = 0;
SqlCommand cmd3 = new SqlCommand("SELECT * FROM Пол WHERE ID =
'" + id_sex + "'", con);
SqlDataReader reader3;
con.Open();
reader3 = cmd3.ExecuteReader();
while (reader3.Read())
{
    sex = reader3["Название"].ToString();
}
reader3.Close();
con.Close();
Foto_Ethernet();
try
{
    var path = Environment.CurrentDirectory + @"\Temp\" +
pulmonForm.ID_FLUOROGRAPH.ToString() + ".bmp";
byte[] img = null;
System.IO.FileStream fs = new System.IO.FileStream(path,
System.IO.FileMode.Open, System.IO.FileAccess.Read);
System.IO.BinaryReader br = new System.IO.BinaryReader(fs);
img = br.ReadBytes((int)fs.Length);
con.Open();
SqlCommand cmd6 = new SqlCommand("UPDATE Файл SET Фото =
@img WHERE ID = '" + pulmonForm.ID_FILE + "'", con);
cmd6.Parameters.Add(new SqlParameter("img", img));
cmd6.ExecuteNonQuery();
con.Close();
}
catch (Exception ex)
{
    System.Windows.MessageBox.Show(ex.Message);
}
MainWindow MW = new MainWindow();
var wordApp = new Word.Application();
conclusion = description.textbox1.Text;
string path2 = Sys-
tem.IO.Path.Combine(Environment.CurrentDirectory, "Протокол.docx");
var wordDocument = wordApp.Documents.Open(path2);
try
{
    description.Hide();
    pathdescription.Stroke = myStroke2;
    descript++;
}

```

```

ReplaceWordStub("{surname}", surname, wordDocument);
ReplaceWordStub("{name}", name, wordDocument);
ReplaceWordStub("{patronymic}", patronymic, wordDocument);
ReplaceWordStub("{id}", id, wordDocument);
ReplaceWordStub("{date}", date, wordDocument);
ReplaceWordStub("{date2}", date2, wordDocument);
ReplaceWordStub("{sex}", sex, wordDocument);
ReplaceWordStub("{diagnosis}", textbox_diagnosis.Text, wordDocument);
ReplaceWordStub("{doctor}", MW.NAME_DOCTOR, wordDocument);
wordApp.Selection.Find.Text = "{conclusion}";
wordApp.Selection.Find.Execute();
wordApp.Selection.TypeText(description.textbox1.Text);
var fileName = Environment.CurrentDirectory + @"\Temp\" +
pulmonForm.ID_FLUOROGRAPH.ToString() + ".bmp";
var range = wordDocument.Content;
Microsoft.Office.Interop.Word.InlineShape picture =
range.InlineShapes.AddPicture(fileName);
WindowState = WindowState.Minimized;
wordApp.Visible = true;
}
catch { System.Windows.MessageBox.Show("Невозможно сформировать
протокол"); }
//Сохранение в файл протокола
wordApp.ActiveDocument.SaveAs(Environment.CurrentDirectory +
@"\Протокол\" + pulmonForm.ID_FLUOROGRAPH.ToString() + ".docx");
try
{
    SqlCommand cmd5 = new SqlCommand("UPDATE Исследование SET
Описание = '" + description.textbox1.Text + "', Диагноз = '" + text-
box_diagnosis.Text + "' WHERE ID_файл = '" + pulmonForm.ID_FILE + "'",
con);

    con.Open();
    cmd5.ExecuteNonQuery();
    con.Close();
}
catch (Exception ex)
{ System.Windows.MessageBox.Show(ex.Message); }
//Сохранение адреса в БД
string path1 = Environment.CurrentDirectory + @"\Протокол\" +
pulmonForm.ID_FLUOROGRAPH.ToString() + ".docx";
try
{
    SqlCommand cmd4 = new SqlCommand("UPDATE Файл SET Протокол
= '" + path1 + "' WHERE ID = '" + pulmonForm.ID_FILE + "'", con);
    con.Open();
    cmd4.ExecuteNonQuery();
    con.Close();
}
catch (Exception ex)
{ System.Windows.MessageBox.Show(ex.Message); }

```

```

    }

    private void ReplaceWordStub(string stubToReplace, string text,
Word.Document wordDocument)
    {
        var range = wordDocument.Content;
        range.Find.ClearFormatting();
        range.Find.Execute(FindText: stubToReplace, ReplaceWith: text);
    }
    private void vieboxedit_MouseEnter(object sender, Sys-
tem.Windows.Input.MouseEventArgs e)
    { Path_Edit.Stroke = myStroke1; }
    private void vieboxedit_MouseLeave(object sender, Sys-
tem.Windows.Input.MouseEventArgs e)
    { Path_Edit.Stroke = myStroke2; }
    private void InkCanvasBackgroundImageAdd()
    {
        try
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("SELECT Флюорография FROM
Файл WHERE ID = '" + pulmonForm.ID_FILE + "'", con);
            SqlDataReader reader = cmd.ExecuteReader();
            reader.Read();
            if (reader.HasRows)
            {
                byte[] img = (byte[])reader[0];
                if (img == null)
                    return;
                else
                {
                    System.IO.MemoryStream ms = new System.IO.MemoryStream(img);
                    System.Drawing.Bitmap bitmap = new System.Drawing.Bitmap(ms);
                    BitmapSource bitmapSource = Sys-
tem.Windows.Interop.Imaging.CreateBitmapSourceFromHBitmap(bitmap.GetHbitmap
()), IntPtr.Zero, Int32Rect.Empty, BitmapSizeOptions.FromEmptyOptions());
                    ImageBrush image = new ImageBrush(bitmapSource);
                    inkCanvas1.Background = image;
                    image.Stretch = Stretch.Uniform;
                }
            }
            else
            { System.Windows.MessageBox.Show("Такой картинки нет"); }
            con.Close();
        }
        catch (Exception ex)
        { System.Windows.MessageBox.Show(ex.Message); }
    }
    private void Foto_Ethernet()

```

```
{
```

Окончание приложения В

```
        RenderTargetBitmap rtb = new
RenderTargetBitmap((int)inkCanvas1.ActualWidth,
(int)inkCanvas1.ActualHeight, 96d, 96d, PixelFormats.Default);
        rtb.Render(inkCanvas1);
        BmpBitmapEncoder encoder = new BmpBitmapEncoder();
        encoder.Frames.Add(BitmapFrame.Create(rtb));
        var location = Environment.CurrentDirectory + @"\Temp\";
        try
        {
            System.IO.FileStream fs = System.IO.File.Open(location +
pulmonForm.ID_FLUOROGRAPH.ToString() + ".bmp", System.IO.FileMode.Create);
            encoder.Save(fs);
            fs.Close();
        }
        catch { System.Windows.MessageBox.Show("Повторите попытку по-
позже. Файл закрывается!"); }
    }
    private void Viewboxexit_MouseLeftButtonUp(object sender,
MouseButtonEventArgs e)
    {
        MessageBoxResult result = System.Windows.MessageBox.Show("Вы
сохранили пиктограммы и протокол?", "Внимание",
MessageBoxButton.YesNoCancel, MessageBoxImage.Question);
        if (result == MessageBoxResult.Cancel)
        {
            return;
        }
        if (result == MessageBoxResult.No)
        {
            vieboxsave_MouseLeftButtonUp(null, null);
        }
        if (result == MessageBoxResult.Yes)
        {
            description.Close();
            pulmonForm.Close();
            this.Close();
        } }
    private void Viewboxexit_MouseEnter(object sender, Sys-
tem.Windows.Input.MouseEventArgs e)
    {
        Pathexit.Stroke = myStroke1;
    }

    private void Viewboxexit_MouseLeave(object sender, Sys-
tem.Windows.Input.MouseEventArgs e)
    {
        Pathexit.Stroke = myStroke2;
```

} } }