

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «Южно-Уральский государственный университет» (НИУ)
Институт «Высшая школа экономики и управления»
Кафедра «Информационные технологии в экономике»

ПРОЕКТ ПРОВЕРЕН

Рецензент, к.т.н., доцент

_____/А.И. Демченко /

« ____ » _____ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.т.н., с.н.с.

_____/ Б.М. Суховилов /

« ____ » _____ 2017 г.

Разработка системы автоматизации проведения диспансеризации.
Подсистема сбора первичных данных

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ–09.04.03.2016.121.ПЗ ВКР

Консультант, к.т.н., доцент

_____/ Е.М. Саргасов /

« ____ » _____ 2017 г.

Руководитель, к.т.н., доцент

_____/ О. И. Галичин /

« ____ » _____ 2017 г.

Автор

студент группы ЭУ–292

_____/ Е.С. Манеева /

« ____ » _____ 2017 г.

Нормоконтролер, доцент

_____/ Е.А. Конова /

« ____ » _____ 2017 г.

Челябинск 2017

АННОТАЦИЯ

Манеева Е.С. «Разработка системы автоматизации проведения диспансеризации. Подсистема сбора первичных данных». – Челябинск: ЮУрГУ, ЭУ-292, 2017 – 74 с., 39 ил., библиогр. список – 14 наим.

Объектом исследования является процесс проведения диспансеризации в лечебно-профилактических учреждениях страны.

Цель исследования: проектирование и разработка автоматизированной информационной системы проведения диспансеризации взрослого населения, анализ существующих подходов к решению данной задачи и их оценка.

Методы исследования: теоретический, эмпирический, системный и сравнительный анализ информации.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1 ПОСТАНОВКА ЗАДАЧИ.....	8
1.1 Технические требования к подсистеме	8
1.2 Требования к организации защиты данных	9
1.3 Анализ существующих подходов к решению задачи и их оценка.....	10
1.4 Выбор и обоснование программных средств разработки системы.....	17
Вывод по разделу один	19
2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	20
2.1 Разработка и описание модели приложения.....	20
2.2 Проектирование и разработка базы данных в MS SQL Server	21
2.3 Описание используемых таблиц базы данных	22
2.4 Алгоритмы решения поставленных задач	30
Вывод по разделу два.....	32
3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА	33
3.1 Разработка интерфейса и описание работы приложения	33
Вывод по разделу три.....	68
4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	69
4.1 Расчет затрат	69
Вывод по разделу четыре.....	70
ЗАКЛЮЧЕНИЕ	71
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	72

ВВЕДЕНИЕ

Актуальность темы. В соответствии с приказами Министерства здравоохранения, реализующих исполнение государственной программы в рамках системы ОМС, медицинские организации, осуществляют масштабную деятельность по диспансеризации отдельных групп взрослого населения. Вся эта деятельность способствует повышению общего уровня здоровья населения, ранней профилактики заболеваний, но требует привлечения значительных ресурсов медицинской организации. В силу большого объема учетных работ и достаточно сложной отчетности, проведение диспансеризации может быть эффективно подвергнуто автоматизации [1].

Диспансеризация – это метод активного динамического наблюдения за состоянием здоровья пациента с целью выявления хронических заболеваний, складывающийся из нескольких этапов: первичный скрининг, осмотры врачей специалистов, распределение пациентов по группам здоровья.

Применение информационных технологий, в том числе внедрение программно-аппаратных комплексов позволяет автоматизировать длительный медико-технологический процесс диспансеризации взрослого населения, оптимизировать учет полученных данных и формирование отчетности [2].

Научная новизна заключается в многоуровневом сопровождении диспансеризации с вызовом пациента на ее прохождение согласно графика ее проведения, автоматизированной генерации списка обследований первого этапа и формировании расписания прохождения этих исследований.

Цель работы – спроектировать и автоматизировать процессы подсистемы сбора первичных данных.

Задачи работы:

- выявить и проанализировать существующие подходы к автоматизации диспансеризации, оценить существующие на рынке системы по определенным критериям, в том числе успешность их внедрения;

- провести анализ нормативно-правовых актов, определяющих порядок проведения диспансеризации, изучить методические рекомендации по проведению диспансеризации определенных групп взрослого населения;
- определить общие требования к системе, а также конкретные задачи в рамках подсистемы;
- определить функционал подсистемы, планируемый к реализации;
- оценить проблемы и особенности по реализации технических требований в рамках разрабатываемой подсистемы;
- произвести выбор инструментария для решения поставленной задачи и обоснование этого выбора;
- спроектировать базу данных, определить ее структуру и взаимосвязи таблиц;
- определить требования к организации политики конфиденциальности и защиты данных от несанкционированного доступа.

Для создания системы использовались языки программирования C#, в качестве систем управления базами данных был использован MS SQL Server 2012 Express. Основной используемый инструментарий для разработки и администрирования приложения – Microsoft Visual Studio 2015.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Технические требования к подсистеме

Программный комплекс должен быть предназначен для автоматизации медико-технологического процесса при диспансеризации взрослого населения в лечебно-профилактических учреждениях, учета полученных данных и формирования статистики проведенных осмотров.

Для этого система должна состоять из двух связанных между собой общим источником данных приложений: клиентского приложения для сбора и автоматизированной обработки данных и интернет-приложения для мониторинга процессов диспансеризации и отображения статистики.

В клиентском приложении выделены две подсистемы:

1. Подсистема сбора первичных данных.
2. Подсистема автоматической обработки результатов осмотра пациентов.

Таким образом, необходимо разработать подсистему сбора первичных данных, которая будет соответствовать следующим требованиям:

- наличие процедуры вызова пациента лечебного учреждения на диспансеризацию согласно графика ее проведения по годам;
- наличие удобного пользовательского интерфейса для ввода персональных данных пациента (ФИО, адрес, телефон, № полиса, № паспорта и т.д.);
- наличие интерфейса для проведения анкетирования пациентов на выявление хронических неинфекционных заболеваний с автоматическим определением факторов риска их развития (хранение вопросов анкеты и ответов пациентов в базе данных);
- наличие удобного интерфейса карты внесения диагнозов диспансерного наблюдения;
- автоматическое распределение плана диспансеризации первого этапа по возрасту и полу пациента (генерация списка необходимых к прохождению базовых медицинских исследований);

- удобное отображение списка показанных по возрасту и полу осмотров и исследований, из них уже пройденных, незавершенных, отказов;
- наличие удобного интерфейса для внесения данных первичных обследований (антропометрия, измерения, анализы);
- наличие удобного интерфейса для автоматизированного статистического анализа с использованием методов статистики (изменение процента заболеваемости в зависимости от года; зависимость заболеваемости от выявленных факторов риска развития этих заболеваний по отдельным категориям граждан).

1.2 Требования к организации защиты данных

Разрабатываемая подсистема должна быть защищена от несанкционированного доступа к данным, представляющим собой конфиденциальную информацию (данные о заболеваниях пациентов), которая ни при каких условиях не должна попасть в руки к третьим лицам.

В связи с этим в системе необходимо реализовать соответствующую политику, по которой доступ к конфиденциальным данным могут получить только те лица, кому эти данные принадлежат, а также врач, заполняющий эти данные в программе.

В программном комплексе предусмотрены следующие роли, разграничивающие права доступа к конфиденциальным данным:

1. Пациент (просмотр персональной информации, персональных данных по проведенным обследованиям и выявленным в рамках диспансеризации заболеваниям).

2. Врач-терапевт (просмотр и изменение персональной информации пациентов, заполнение и редактирование данных обследований, выявленных заболеваний, просмотр статистических данных).

3. Системный администратор (создание и редактирование пользователей, выдача и изменение прав пользователей).

В рамках разрабатываемой подсистемы реализации подлежат роли врача-терапевта и администратора.

Для этого в системе предусмотрена регистрация врачей администратором и присвоение им соответствующих занимаемой должности ролей. Пароль для аутентификации и входа в систему устанавливаются сами пользователи, и вместо самого пароля хранится его контрольная сумма, вычисляемая с помощью специального алгоритма. Такая сумма называется "хэшем", но проблема заключается в том, что хэш одинаковых паролей совпадает. Для этого при создании/изменении пароля генерируется определенное уникальное число, называемое "солью". Это число присоединяется к паролю и вычисляется хэш не самого пароля, а строки, полученной присоединением соли к паролю. Такой хэш называется "подсоленным", и именно он и хранится в системе вместо пароля.

Таким образом, процесс аутентификации пользователя в системе представляет собой вычисление хэша строки, получаемой присоединением соли к вводимому каждый раз пользователем паролю, и сравнение этого хэша с тем, что хранится в базе данных.

1.3 Анализ существующих подходов к решению задачи и их оценка

В настоящее время на рынке существует не так много решений по автоматизации проведения диспансеризации. Некоторые из них имеют свои достоинства и недостатки в том или ином аспекте. В данном разделе рассмотрены несколько примеров существующих решений данной задачи.

1.3.1 ТрастМед: Диспансеризация

TrustMedTM — полнофункциональное решение задач автоматизации лечебно-профилактических и аптечных учреждений, персонифицированного учета оказания медицинской помощи, а также процессов обеспечения необходимыми лекарственными средствами отдельных категорий граждан (ОНЛС).

В состав комплекса TrustMed входит набор программ, которые способны функционировать как самостоятельно, так и в составе интегрированного комплекса, обеспечивая единое информационное пространство здравоохранения региона [10].

SofTrust разрабатывает программное обеспечение в соответствии с индивидуальными требованиями и потребностями заказчика, использует творческий подход в решении самых нестандартных задач, а также гарантирует заказчику реализацию проекта в срок и в рамках запланированного бюджета.

Преимущества:

1) Система «ТМ: Диспансеризация» входит в программный комплекс «ТМ: МИС (Медицинская Информационная Система)» и предназначена для автоматизации работы при проведении диспансеризации различных групп населения.

2) Система обеспечивает контроль и мониторинг прохождения диспансеризации со стороны работодателей и организационных групп населения.

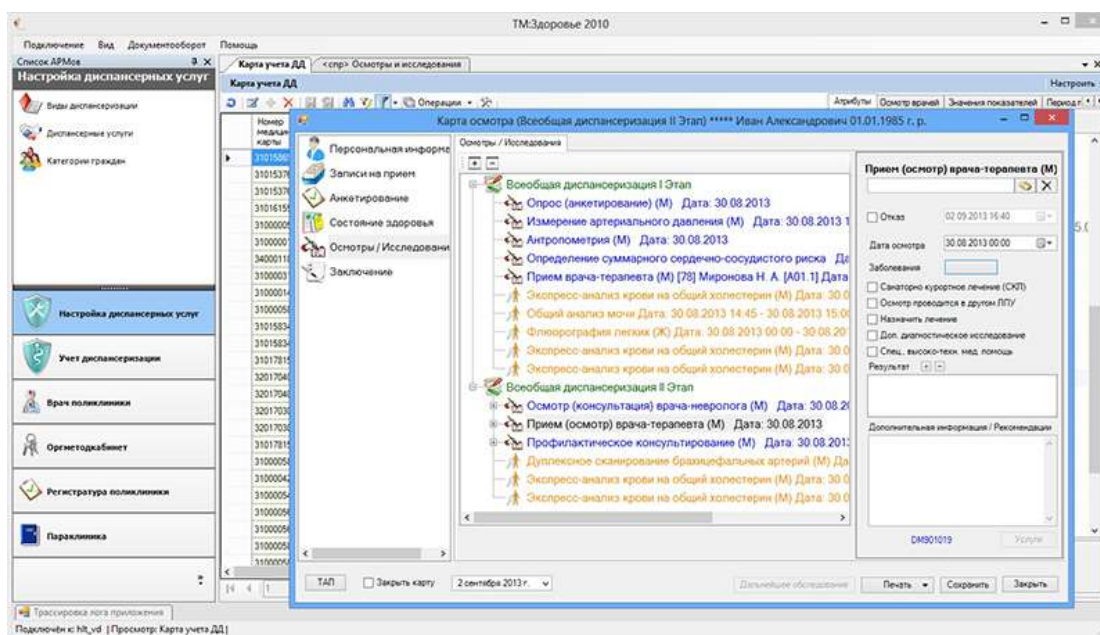


Рисунок 1.1 – ИС «ТМ:Диспансеризация». Карта осмотра

Недостатки:

1) Система выполнена по старым нормативным актам, не соответствующим новым требованиям и стандартам.

2)Отсутствует процедура вызова пациента на диспансеризацию согласно графика ее проведения по годам.

3)Отсутствует возможность редактирования даты и времени приёма обследований пациента.

Стоимость лицензии ТрастМед:

- Стоимость лицензии на одно рабочее место составляет 6000 рублей.
- Внедрение составляет 24000 рублей на одно рабочее место и включает: установку, настройку, техническую поддержку в течение гарантийного срока.
- Обучение пользователя составляет 8000 рублей на человека.
- Обучение администратора составляет 30000 рублей.
- Техническая поддержка одного рабочего места составляет 1000 рублей в месяц и включает: линию консультаций, устранение ошибок, предоставление обновлений.

1.3.2 Система «САДиП»

САДиП - это программная система для автоматизации диспансеризации и профилактических осмотров. Программное обеспечение создавалось на основе рекомендаций и пожеланий действующих врачей-терапевтов, совместно с ФГБУ "ГНИЦПМ" Минздрава России. Это позволило учесть все профессиональные нюансы и создать продукт, который призван облегчить задачу врачей отделений медицинской профилактики на всех этапах проведения диспансеризации и профилактических осмотров [12].

Преимущества:

- 1) Возможность импорта реестра пациентов и оказанных услуг.
- 2) Возможность прохождения анкетирования конфиденциально на сайте с получением штрих-кода, возможность ввода анкеты в базу сканером штрих-кода.

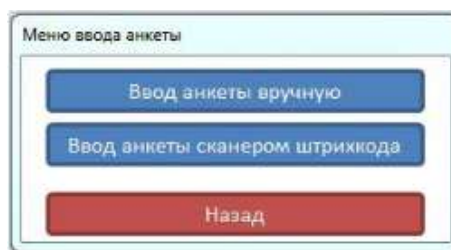


Рисунок 1.2 – ИС «САДиП». Меню ввода анкеты

Список обследований по выбранному пациенту

Наименование обследования	Справка	Дата проведения	Заболевания (отклонения)	Отказ
Опрос (анкетирование), направленный на выявление хронических неинфекционных заболеваний		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Антропометрия (измерение роста стоя, массы тела, окружности талии, бедра)		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Измерение артериального давления		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Определение уровня общего холестерина в крови		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Определение уровня глюкозы в крови экспресс-методом		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Определение абсолютного суммарного сердечно-сосудистого риска		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Электрокардиография (в покое)			<input type="checkbox"/>	<input checked="" type="checkbox"/>
Флюорография легких		17.04.2014	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Клинический анализ крови		20.02.2016	<input type="checkbox"/>	<input type="checkbox"/>
Общий анализ мочи		29.07.2015	<input type="checkbox"/>	<input type="checkbox"/>
Исследование кала на скрытую кровь иммунохимическим методом		17.06.2015	<input type="checkbox"/>	<input type="checkbox"/>
Измерение внутриглазного давления		20.02.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Прием (осмотр) врача-терапевта (I этап)		20.02.2016	<input type="checkbox"/>	<input type="checkbox"/>

Сохранить Печать направлений Печать отказов Отмена

Рисунок 1.3 – ИС «САДиП». Список обследований по пациенту

Недостатки:

- 1) Отсутствует формирование расписания обследований пациента и возможность редактирования даты и времени приёма обследований.
- 2) Отсутствие единого интерфейса карты учёта диспансеризации по пациенту для отображения поэтапного прохождения диспансеризации.
- 3) Отсутствует процедура вызова пациента на диспансеризацию согласно графика ее проведения по годам.
- 4) Суммарная стоимость информационно-технического обслуживания 15 рабочих мест (5 + 3 + 3 + 2 + 2) составит: 60 000 + 54 000 + 54 000 + 48 000 + 48 000 = 264 000 за год.

1.3.3 Подсистема «КМИС – Диспансеризация и профилактические осмотры»

Компания "Комплексные медицинские информационные системы" (сокращенно К-МИС) - это специализированная Российская IT-компания, предлагающая современные информационные системы для автоматизации здравоохранения собственного производства, а также различный спектр услуг по их внедрению и техническому сопровождению[13].

Проекты внедрения программных продуктов КМИС Решения компании "Комплексные медицинские информационные системы" внедряются по всей стране. На сегодняшний день клиентская база компании составляет свыше 80 заказчиков и более 300 лечебно-профилактических учреждений разного профиля, формы собственности и специализации. Количество пользователей, работающих с программными продуктами компании, превышает 12 тыс. человек.

Подсистема диспансеризации и профилактических осмотров предназначена для полной автоматизации процесса организации, выполнения и подведения итогов различных медицинских осмотров и профосмотров, в том числе диспансеризации взрослого населения медосмотров несовершеннолетних, дополнительной диспансеризации и т.д [14].

Преимущества:

- 1) Наличие специальных справочников системы для хранения настроек диспансеризации и профосмотров.
- 2) Настройка периодичности различных мероприятий, предусмотренных диспансеризацией и профосмотрами, а также их автоматическое назначение и планирование.



Рисунок 1.4 – Подсистема «КМИС – Диспансеризация и профилактические осмотры». Центральный справочник МИС



Рисунок 1.5 – Подсистема «КМИС – Диспансеризация и профилактические осмотры». Диспансеризация 1й этап.

Недостатки:

- 1) Система выполнена по старым нормативным актам, не соответствующим новым требованиям и стандартам.
- 2) Отсутствует процедура вызова пациента на диспансеризацию согласно графика ее проведения по годам.
- 3) Отсутствует формирование расписания обследований пациента и возможность редактирования даты и времени приёма обследований.

В таблицу 1 сведены сравнительные характеристики рассмотренных систем, выявлены их достоинства и недостатки в сравнении с выполненным проектом.

Таблица 1 – Сравнительный анализ существующих аналогов

№ п.п.	Программа	Нормативные акты, требования и стандарты	Интерфейс	Дополнительные возможности
1	ТрастМед: Диспансеризация	Система выполнена по старым нормативным актам, не соответствующим новым требованиям и стандартам.	Наличие отдельного модуля для контроля и мониторинга прохождения диспансеризации со стороны работодателей и организационных групп населения.	1.Отсутствует процедура вызова пациента на диспансеризацию согласно графика ее проведения по годам. 2. Отсутствует возможность редактирования расписания обследований пациента.
2	«САДиП»	Система выполнена по старым нормативным актам, не соответствующим новым требованиям и стандартам.	Отсутствие единого интерфейса карты учёта диспансеризации по пациенту для отображения поэтапного прохождения диспансеризации.	1.Отсутствует процедура вызова пациента на диспансеризацию согласно графика ее проведения по годам. 2. Отсутствует возможность редактирования расписания обследований пациента.
3	Подсистема «КМИС – Диспансеризация и профилактические осмотры»	Система выполнена по старым нормативным актам, не соответствующим новым требованиям и стандартам.	1. Наличие специальных справочников системы для хранения настроек диспансеризации. 2. Настройка периодичности различных мероприятий, предусмотренных диспансеризацией.	1.Отсутствует процедура вызова пациента на диспансеризацию. 2. Отсутствует возможность редактирования расписания обследований пациента.

1.4 Выбор и обоснование программных средств разработки системы

Выбор системы управления базами данных является одним из важных этапов при разработке автоматизированной системы диспансеризации.

Microsoft SQL Server – одна из наиболее мощных СУБД архитектуры клиент-сервер. Эта СУБД позволяет удовлетворять такие требования, предъявляемые к системам распределенной обработки данных, как тиражирование данных, параллельная обработка, поддержка больших баз данных на относительно недорогих аппаратных платформах при сохранении простоты управления и использования.

MS SQL Server не предназначен непосредственно для разработки пользовательских приложений, а выполняет функции управления базой данных. Сервер имеет средства удаленного администрирования и управления операциями, организованные на базе объектно-ориентированной распределенной среды управления.

Microsoft SQL Server предназначен исключительно для поддержки систем, работающих в среде клиент-сервер. Он поддерживает широкий спектр средств разработки и максимально прост в интеграции с приложениями, работающими на ПК.

В процессе разработки программного продукта было принято решение об использовании СУБД – Microsoft SQL Server 2012.

В качестве инструментальной среды разработки выбор был сделан в пользу MS Visual Studio 2015.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server) [3].

В качестве языка программирования выбран язык C# с поддержкой технологии .NET.Framework.

C# – объектно-ориентированный язык программирования. Разработан в 1998–2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [4].

В него входит много полезных особенностей — простота, объектная ориентированность, типовая защищённость, «сборка мусора», поддержка совместимости версий и многое другое. Данные возможности позволяют быстро и легко разрабатывать приложения.

.NET Framework – программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

Основной идеей при разработке .NET Framework являлось обеспечение свободы разработчика за счёт предоставления ему возможности создавать приложения различных типов, способные выполняться на различных типах устройств и в различных средах.

Программа для .NET Framework, написанная на любом поддерживаемом языке программирования, сначала переводится компилятором в единый для .NET промежуточный байт-код Common Intermediate Language (CIL) (ранее назывался Microsoft Intermediate Language, MSIL). В терминах .NET получается сборка, англ. assembly. Затем код либо исполняется виртуальной машиной Common Language Runtime (CLR), либо транслируется утилитой NGen.exe в исполняемый код для конкретного целевого процессора. Использование виртуальной машины предпочтительно, так как избавляет разработчиков от необходимости заботиться об особенностях аппаратной части. В случае использования виртуальной машины CLR встроенный в неё JIT-компилятор «на лету» (just in time) преобразует промежуточный байт-код в машинные коды нужного процессора. Современная технология динамической компиляции позволяет достигнуть высокого уровня быстродействия. Виртуальная машина CLR также сама заботится о базовой безопасности, управлении памятью и системе исключений, избавляя разработчика от части работы.

Архитектура .NET Framework описана и опубликована в спецификации Common Language Infrastructure (CLI), разработанной Microsoft и утверждённой ISO и ECMA. В CLI описаны типы данных .NET, формат метаданных о структуре программы, система исполнения байт-кода и многое другое [5].

Вывод по разделу один

В разделе представлены примеры существующих подходов и разработок в области автоматизации диспансеризации, определены их возможности, выявлены преимущества и недостатки. Сделан и обоснован выбор программных средств разработки.

2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Разработка и описание модели приложения

Медико-технологический процесс диспансеризации взрослого населения начинается с информирования пациентов о возможности ее прохождения. Пациенты, изъявившие желание пройти диспансеризацию, получают индивидуальные маршрутные карты обследований первого и второго этапа, в зависимости от определенных нормативными актами критериев. В свою очередь, врачи ведут карты учёта диспансеризации, редактируют автоматически сформированные расписания обследований пациента, вносят результаты обследований и заполняют карты.

На рисунке 2.1 показана схема процесса диспансеризации в общем виде со стороны пациента и медицинской организации.

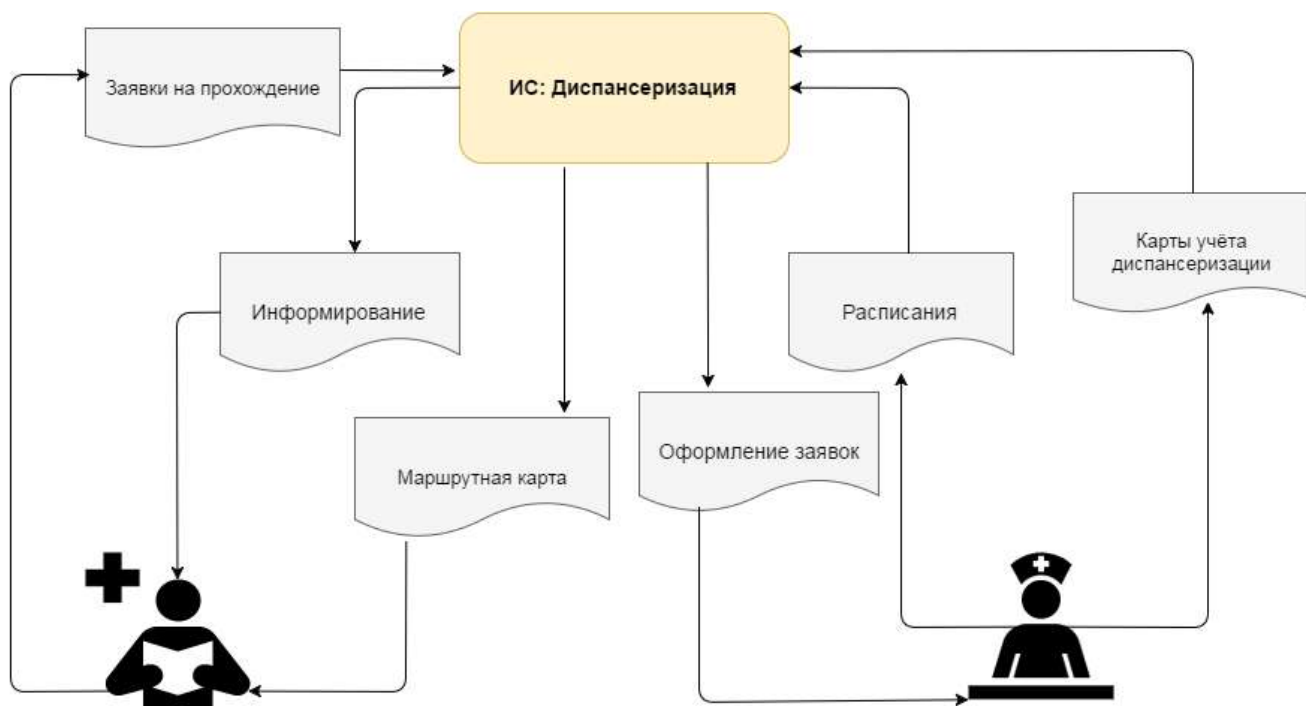


Рисунок 2.1 – Схема процесса диспансеризации

2.2 Проектирование и разработка базы данных в MS SQL Server

На этапе проектирования базы данных выполняется построение информационно-логической или инфологической модели (ИЛМ) предметной области, называемой также концептуальной моделью [15].

К числу объектов или сущностей, имеющих важное функциональное назначение в данной подсистеме, относится справочник пациентов, карта учета диспансеризации, справочник обследований и проведенные обследования по карте учёта.

Схема разработанной базы данных информационной системы представлена на рисунке 2.2.

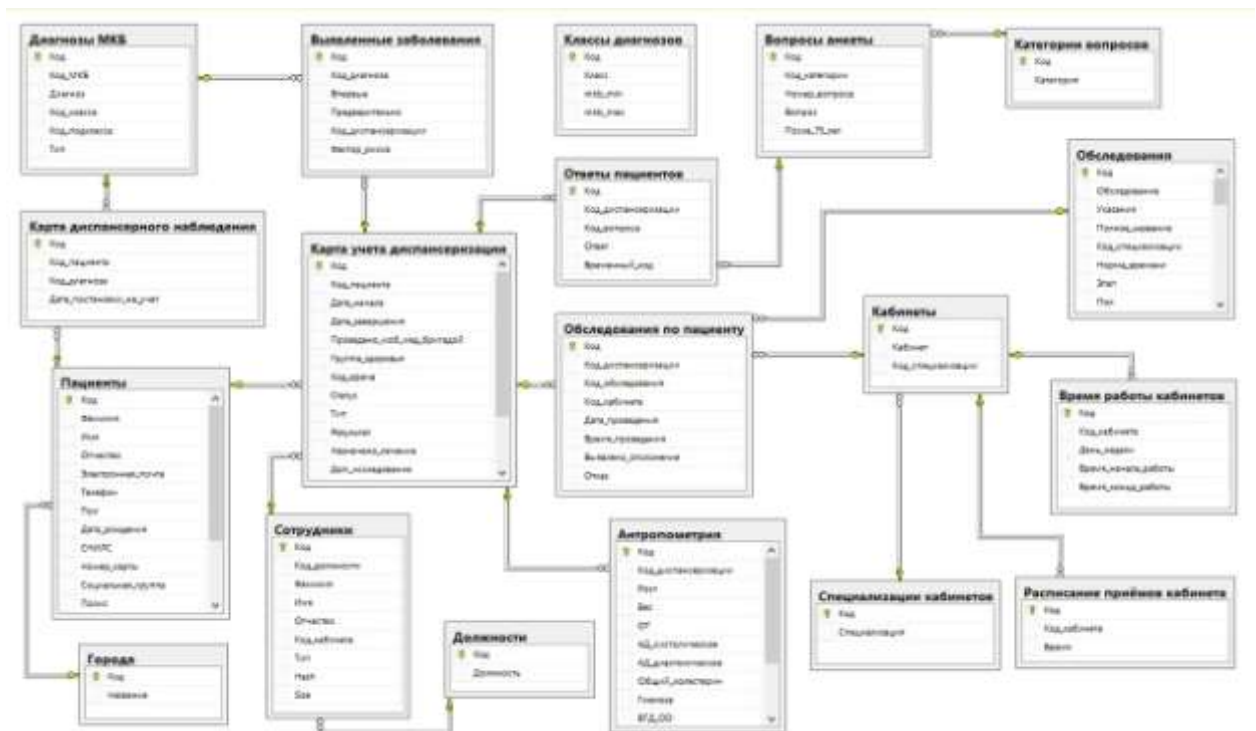


Рисунок 2.2 – Схема базы данных системы

Разработанная база данных приведена к третьей нормальной форме.

2.3 Описание используемых таблиц базы данных в реализуемой подсистеме

Разработанная база данных содержит таблицы, в которых хранится информация, необходимая для функционирования системы диспансеризации.

В реализуемой подсистеме сбора первичных данных пациентов используются следующие таблицы:

1. Таблица «Пациенты»

	Имя столбца	Тип данных
▶	Код	bigint
	Фамилия	varchar(150)
	Имя	varchar(150)
	Отчество	varchar(150)
	Электронная_почта	varchar(150)
	Телефон	varchar(17)
	Пол	varchar(1)
	Дата_рождения	date
	СНИЛС	varchar(30)
	Номер_карты	varchar(30)
	Социальная_группа	varchar(250)
	Полис	varchar(30)
	Код_города	int
	Местность	bit
	Адрес	varchar(250)
	Hash	varchar(200)
	Size	int

Рисунок 2.3 – Таблица «Пациенты»

Таблица необходима для хранения персональных данных пациентов. Таблица «Пациенты» является основной таблицей базы данных и содержит поля «Код», «Фамилия», «Имя», «Отчество», «Телефон», «Пол», «Дата рождения», «СНИЛС», «Номер карты», «Социальная группа», «Полис», «Код_города», «Местность», «Адрес».

2. Таблица «Вопросы анкеты»


	Имя столбца	Тип данных	Разрешить ...
	Код	int	<input type="checkbox"/>
	Код_категории	int	<input checked="" type="checkbox"/>
	Номер_вопроса	varchar(5)	<input checked="" type="checkbox"/>
	Вопрос	varchar(500)	<input checked="" type="checkbox"/>
	После_75_лет	bit	<input checked="" type="checkbox"/>

Рисунок 2.4 – Таблица «Вопросы анкеты»

Таблица «Вопросы анкеты» предназначена для хранения вопросов анкетирования пациентов на первом этапе диспансеризации и содержит поля: поле «Код» – уникальный идентификатор, поле «Код категории» – номер раздела вопроса анкетирования, поле «Номер_вопроса» – номер вопроса в указанном разделе, поле «Вопрос» – формулировка вопроса, поле «После_75_лет» – имеет значение True или False для отметки вопросов, которые входят в анкетирование для людей, старше 75 лет, так как для данной возрастной категории перечень вопросов меняется.

3. Таблица «Ответы пациентов»


	Имя столбца	Тип данных
	Код	bigint
	Код_диспансеризации	bigint
	Код_вопроса	int
	Ответ	int

Рисунок 2.5 – Таблица «Ответы пациентов»

Таблица «Ответы пациентов» предназначена для хранения ответов анкетирования пациентов для последующего автоматического выявления хронических неинфекционных заболеваний. Данная таблица содержит поля: поле «Код» – уникальный идентификатор, поле «Код_диспансеризации» – содержит код таблицы «Карта учёта диспансеризации», т.е. конкретной карты пациента, поле «Код_вопроса» – содержит код вопроса из таблицы «Вопросы анкеты», «Ответ» – предназначен для хранения ответа на вопрос анкетирования в виде определенных установленных цифр.

4. Таблица «Антропометрия»


	Имя столбца	Тип данных
	Код	bigint
	Код_диспансеризации	bigint
	Рост	float
	Вес	float
	ОТ	float
	АД_систолическое	int
	АД_диастолическое	int
	Общий_холестерин	float
	Глюкоза	float
	ВГД_OD	float
	ВГД_OS	float
	ССР	varchar(20)
	процент_ССР	int

Рисунок 2.6 – Таблица «Антропометрия»

Таблица предназначена для хранения результатов первичных исследований первого этапа диспансеризации. Таблица «Антропометрия» содержит поля: поле «Код» – уникальный идентификатор, поле «Код_диспансеризации» – содержит код таблицы «Карта учёта диспансеризации», поле «Рост» – рост пациента, поле «Вес» – вес пациента, поле «ОТ» – обхват талии, поле «АД_систолическое» – артериальное давление систолическое, поле «АД_диастолическое» – артериальное давление диастолическое, поле «Общий_холестерин» – результат анализа холестерина, поле «Глюкоза» – результат анализа глюкозы, поле «ВГД_OD» и «ВГД_OS» – внутриглазное давление, поле «ССР» – сердечно-сосудистый риск, поле «процент_ССР» – процент сердечно-сосудистого риска.

5. Таблица «Обследования»

	Имя столбца	Тип данных
▶	Код	int
	Обследование	varchar(150)
	Указания	varchar(1000)
	Полное_название	varchar(500)
	Код_специализации	int
	Норма_времени	int
	Этап	int
	Пол	varchar(1)
	[21]	bit
	[24]	bit
	[27]	bit
	[30]	bit
	[33]	bit
	[36]	bit
	[39]	bit
	[42]	bit
	[45]	bit
	[48]	bit
	[51]	bit
	[54]	bit

Рисунок 2.7 – Таблица «Обследования»

Таблица «Обследования» предназначена для хранения перечня обследований в соответствии с этапом прохождения диспансеризации, а также с полом и возрастом пациента. Таблица содержит поля: поле «Код» – уникальный идентификатор, поле «Обследование» – содержит название обследования, поле «Полное название обследования», поле «Указания» – содержит указания для подготовки к обследованию, поле «Код_специализации» – содержит код таблицы «Специализации кабинетов» для определения номера кабинета для проведения обследования, поле «Норма_времени» – содержит установленное количество затрачиваемых минут на проведение данного обследования, «Этап» – содержит номер этапа, к которому относится обследование, «Пол» – предназначено для идентификации обследований по половому признаку. Далее, поля возрастов диспансеризации, которые имеют значение True или False – предназначено для определения необходимых обследований для данной возрастной категории. Таким образом, для каждого исследования определен этап, пол и отмечены те возраста диспансеризации, которым необходимо пройти данное обследование (рис. 2.8).

Код	Обследование	Укл.	План	К.	Нр.	Э.	Рис	25	26	27	28
1	Опрос (анкет.)	MSL	Опрос	9	20	1	MSL	True	True	True	True
4	Антропометрия	MSL	Антр.	9	3	1	MSL	True	True	True	True
5	Измерение арт.	MSL	Измер.	9	5	1	MSL	True	True	True	True
6	Определение у.	Кров.	Опред.	3	15	1	MSL	True	True	True	True
7	Определение у.	Кров.	Опред.	3	15	1	MSL	True	True	True	True
8	Определение	MSL	Опред.	9	5	1	MSL	True	True	True	True
9	Определение	MSL	Опред.	9	5	1	MSL	MSL	MSL	MSL	MSL
10	Электрокард.	Гар.	Элект.	3	15	1	ж	MSL	MSL	MSL	MSL
11	Электрокард.	Гар.	Элект.	3	15	1	ж	MSL	MSL	MSL	MSL
12	Синтез фальд.	Заб.	Синтез	4	15	1	ж	True	True	True	True
13	Флюорограф.	MSL	Флюор.	14	15	1	MSL	True	True	True	True
14	Манометрия	MSL	Ман.	7	20	1	ж	MSL	MSL	MSL	MSL
15	Клинический	Кров.	Клини.	3	15	1	MSL	True	True	True	True
16	Клинический	Кров.	Клини.	3	15	1	MSL	MSL	MSL	MSL	MSL
17	Анализ крови	Кров.	Анализ	3	15	1	MSL	MSL	MSL	MSL	MSL
18	Общий анализ	Гар.	Общ.	4	MSL	1	MSL	True	True	True	True
20	Исследование	Диаг.	Иссл.	4	MSL	1	MSL	MSL	MSL	MSL	MSL
21	УЗИ на предст.	MSL	Ультр.	2	15	1	MSL	MSL	MSL	MSL	MSL
24	УЗИ в щитов. ж.	MSL	Ультр.	2	15	1	ж	MSL	MSL	MSL	MSL
25	Измерение вкл.	MSL	Измер.	8	15	1	MSL	MSL	MSL	MSL	MSL
27	Проба (слюн.)	MSL	Проб.	9	15	1	MSL	True	True	True	True
28	Дуплексное ск.	MSL	Дупл.	2	15	2	MSL	True	True	True	True
29	Синтез (ультр.)	MSL	Синтез	10	15	2	MSL	True	True	True	True
30	Эхокардиограф.	MSL	Эхокард.	11	15	2	MSL	True	True	True	True

Рисунок 2.8 – Таблица «Обследования»

6. Таблица «Обследования по пациенту»

	Имя столбца	Тип данных
🔑	Код	bigint
	Код_диспансеризации	bigint
	Код_обследования	int
	Код_кабинета	int
	Дата_проведения	date
	Время_проведения	time(7)
	Выявлено_отклонение	bit
	Отказ	bit

Рисунок 2.9 – Таблица «Обследования по пациенту»

Таблица «Обследования по пациенту» предназначена для хранения информации о перечне обследований по каждому пациенту. Таблица содержит поля: «Код» – уникальный идентификатор, «Код_диспансеризации» – код таблицы «Карта учёта диспансеризации» для конкретного пациента, поле «Код_обследования» – код из таблицы «Обследования», «Код_кабинета» – код из таблицы «Кабинеты» для проведения указанного обследования, поле «Дата проведения», поле «Время проведения» – содержит время проведения обследования, «Выявлено отклонение» – для сигнализации об отклонениях по результатам обследования, «Отказ» – для отображения отказа от обследования.

7. Таблица «Карта учёта диспансеризации»

	Имя столбца	Тип данных
🔑	Код	bigint
	Код_пациента	bigint
	Дата_начала	date
	Дата_завершения	date
	Проведено_моб_мед_б...	bit
	Группа_здоровья	varchar(10)
	Код_врача	bigint
	Статус	varchar(200)
	Тип	bit
	Результат	varchar(250)
	Назначено_лечение	bit
	Доп_исследование	bit
	К_сосудистому_хирургу	bit
	К_психиатру	bit
	Спец_мед_помощь	bit
	Санаторное_лечение	bit

Рисунок 2.10 – Таблица «Карта учёта диспансеризации»

Таблица «Карта учёта диспансеризации» – центральный объект системы. Таблица хранит всю важную информацию по прохождению пациентом диспансеризации. Таблица содержит поля: поле «Код» – уникальный идентификатор, поле «Код пациента» – код из таблицы «Пациенты», «Дата начала» – дата создания карты учёта диспансеризации, «Дата завершения» – дата завершения диспансеризации по данной карте учёта, поле «Проведено мобильной медицинской бригадой» – в значении True или False, «Группа здоровья» – окончательно определенная группа здоровья пациента в результате прохождения диспансеризации, поле «Назначено лечение», поле «Доп_исследование» – в значении True, если врач назначил пациенту дополнительные исследования, поле «К_сосудистому_хирургу» – в значении True, если врач направил пациента к сосудистому хирургу, поле «К_психиатру» – в значении True, если врач направил пациента к психиатру, поле «Спец_мед_помощь», поле «Санаторное лечение» – в значении True, если пациент направлен на санаторное лечение.

8. Таблица «Кабинеты»

	Имя столбца	Тип данных
🔑	Код	int
	Кабинет	varchar(10)
	Код_специализации	int

Рисунок 2.11 – Таблица «Кабинеты»

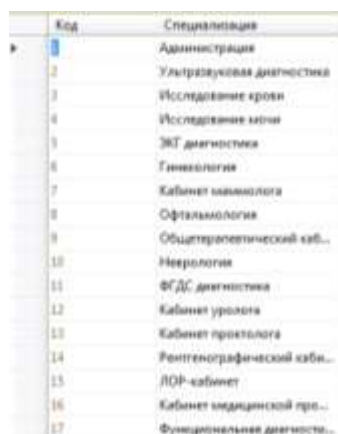
Таблица «Кабинеты» предназначена для хранения информации о кабинетах для прохождения диспансеризации. Таблица содержит поля: поле «Код» – уникальный идентификатор, поле «Кабинет» – номер кабинета, поле «Код_специализации» – содержит код таблицы «Специализации кабинетов».

9. Таблица «Специализации кабинетов»

	Имя столбца	Тип данных
🔑	Код	int
▶	Специализация	varchar(100)

Рисунок 2.12 – Таблица «Специализации кабинетов»

Таблица предназначена для определения специализаций кабинетов и специализаций самих обследований для того, чтобы определить, какие обследования в каких кабинетах проводятся. Таблица содержит поля: поле «Код» – уникальный идентификатор, поле «Специализация». Специализации выделены в соответствии с обследованиями, которые необходимо проводить в рамках диспансеризации (рис.2.13).



Код	Специализация
1	Администрация
2	Ультразвуковая диагностика
3	Исследование крови
4	Исследование мочи
5	ЭКГ диагностика
6	Гинекология
7	Кабинет маммолога
8	Офтальмология
9	Общепроветический каб...
10	Неврология
11	ФГДС диагностика
12	Кабинет уролога
13	Кабинет проктолога
14	Рентгенографической каби...
15	ЛОП-кабинет
16	Кабинет медицинской пре...
17	Функциональная диагно...

Рисунок 2.13 – Таблица «Специализации кабинетов»

10. Таблица «Время работы кабинетов»

	Имя столбца	Тип данных
🔑	Код	int
	Код_кабинета	int
	День_недели	varchar(50)
▶	Время_начала_работы	time(7)
	Время_конца_работы	time(7)

Рисунок 2.14 – Таблица «Время работы кабинетов»

Таблица предназначена для хранения информации о диспансерном дне недели и времени работы кабинетов для диспансеризации. Таблица содержит поля: поле «Код» – уникальный идентификатор, поле «Код_кабинета» – код таблицы «Кабинеты», поле «День_недели» – диспансерный день работы кабинета, поле «Время_начала_работы» и поле «Время_конца_работы» – часы работы кабинета.

11. Таблица «Расписание приёмов кабинета»

	Имя столбца	Тип данных
🔑	Код	int
	Код_кабинета	int
▶	Время	time(7)

Рисунок 2.15 – Таблица «Расписание приёмов кабинета»

Таблица предназначена для хранения возможных значений времени для обследований в данном кабинете. Таблица используется для определения занятого и свободного времени на обследования в данном кабинете в случае изменения времени определенного обследования у пациента.

2.4 Алгоритмы решения поставленных задач

Диспансеризация проводится бесплатно для взрослых граждан, начиная с 21 года и до 90 лет, каждые три года. Прохождение диспансеризации является добровольным, а если человек отказывается от неё, врач фиксирует этот факт в амбулаторной карте пациента.

Для своевременного прохождения диспансеризации необходима система оповещений пациентов, подлежащих диспансеризации согласно установленному плану.

Для реализации процедуры вызова пациента лечебного учреждения на диспансеризацию согласно графика ее проведения по годам в приложении используется проверка соответствия пациентов больницы указанным возрастам, в случае соответствия даты рождения пациента, происходит рассылка оповещения на электронную почту о возможности прохождения диспансеризации в текущем году.

Для проведения анкетирования пациентов на выявление хронических неинфекционных заболеваний с автоматическим определением факторов риска их развития используются формы заполнения анкеты врачом-терапевтом. Ответы анкетирования каждого пациента хранятся в базе данных для последующего определения факторов риска.

Далее, исходя из персональных данных пациента, заполненных на форме карточки пациента, происходит *автоматическое* распределение плана диспансеризации по возрасту и полу, т.е. генерируется список необходимых к прохождению для каждого пациента базовых медицинских исследований:

- Определение относительного суммарного сердечно-сосудистого риска у граждан в возрасте от 21 до 39 лет и абсолютного суммарного сердечно-сосудистого риска у граждан в возрасте от 40 до 65 лет, не имеющих заболеваний, связанных с атеросклерозом.

- Электрокардиография в покое (для мужчин в возрасте старше 35 лет, для женщин в возрасте 45 лет и старше, а для мужчин в возрасте до 35 лет и женщин в возрасте до 45 лет - при первичном прохождении диспансеризации).
- Осмотр фельдшером (акушеркой), включая взятие мазка (соскоба) с поверхности шейки матки (наружного маточного зева) и цервикального канала на цитологическое исследование (далее - мазок с шейки матки) (для женщин в возрасте от 21 года до 69 лет включительно).
- Флюорография легких (не проводится, если в течение предшествующего календарного года либо года проведения диспансеризации проводилась флюорография, рентгенография (рентгеноскопия) или компьютерная томография органов грудной клетки).
- Маммография обеих молочных желез (для женщин в возрасте от 39 до 75 лет).
- Клинический анализ крови (определение концентрации гемоглобина в эритроцитах, количества лейкоцитов и скорости оседания эритроцитов).
- Клинический анализ крови развернутый (в возрасте от 39 лет и старше с периодичностью 1 раз в 6 лет вместо клинического анализа крови).
- Анализ крови биохимический общетерапевтический (определение уровня креатинина, общего билирубина, аспартат-аминотрансаминазы, аланин-аминотрансаминазы, глюкозы, холестерина). В возрасте 39 лет и старше с периодичностью 1 раз в 6 лет.
- Общий анализ мочи.
- Исследование кала на скрытую кровь иммунохимическим методом (допускается проведение бензидиновой или гваяковой пробы) в возрасте от 48 до 75 лет.
- Ультразвуковое исследование (УЗИ) органов брюшной полости и малого таза на предмет исключения новообразований для граждан в возрасте 39 лет и старше с периодичностью 1 раз в 6 лет (для женщин УЗИ поджелу-

дочной железы, почек, матки и яичников; для мужчин УЗИ поджелудочной железы, почек и предстательной железы), а для мужчин, когда-либо куривших в жизни, также УЗИ брюшной аорты с целью исключения аневризмы однократно в возрасте 69 или 75 лет. (УЗИ органов брюшной полости не проводится, если в течение предшествующих 2 лет либо в год проведения диспансеризации проводилась МРТ или КТ органов брюшной полости).

- Измерение внутриглазного давления (в возрасте 39 лет и старше);

Перечень обследований первого этапа хранится в таблице «Обследования» с учётом половых и возрастных категорий граждан.

Для внесения данных первичных обследований (антропометрия, измерения, анализы) используются формы внесения результатов, в результате сравнения определяются отклонения от установленных норм для дальнейшего определения списка обследований второго этапа.

Результаты обследований пациентов хранятся в отдельной таблице «Обследования по пациенту», там же отмечаются отклонения и фиксируются отказы. Каждое назначенное обследование сопровождается датой, временем и указанием кабинета для его прохождения. Для этого существует таблица «Время работы кабинета» для диспансеризации. У каждого кабинета существует определенная специализация. В соответствии с этим, у каждого обследования из общего перечня установлена специализация и определенный интервал времени на его проведение. Далее образуется очередь, т.е. каждому пациенту назначается своё время прохождения тех или иных исследований.

Вывод по разделу два

В разделе отображена схема базы данных, описаны таблицы, используемые при разработке пользовательского приложения, рассмотрены алгоритмы решения поставленных задач.

3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

3.1 Разработка интерфейса и описание работы приложения

Созданная база данных Microsoft SQL Server подключена как внешний источник данных к приложению в Visual Studio.

Процесс диспансеризации начинается с вызова пациента на ее прохождение. Для своевременного прохождения диспансеризации необходима система оповещений пациентов, подлежащих диспансеризации согласно установленному плану. Для реализации процедуры вызова пациента лечебного учреждения на диспансеризацию согласно графика ее проведения по годам в приложении используется проверка соответствия пациентов больницы указанным возрастам, в случае соответствия даты рождения пациента, происходит рассылка оповещения на электронную почту о возможности прохождения диспансеризации в текущем году.

Форма «Список пациентов» представлена на рисунке 3.1.

Пациент	Дата рождения	Пол	Телефон	Электронная почта	Номер карты	Город	Социальная группа
Курочкин И. С.	11.08.1983	м	+7(913)	ikura@mail.ru	34683945	Владимир	Работающее население
Воробьев П. А.	12.11.1978	м	+7(912)	pr6@mail.ru	34390453	Владимир	Обучающиеся в обл.
Земляков Р. А.	11.12.1980	м	+7(91)	4g@mail.ru	358F4572	Челябинск	Неработающее население
Бурочкина Ж. Н.	11.11.2000	ж	+7(912)	dfg4@mail.ru	4333X561	Орск	Неработающее население
Ромашкин Р. Р.	11.12.1990	м	+7(35)	romashkin_r@mail.ru	34767914	Челябинск	Работающее население
Шошвакин Д. Д.	13.11.1965	м	+7(351)		85436872	Челябинск	Неработающее население
Шинкин И. И.	03.12.1955	м	+7(35)		85384138	Учалы	Неработающее население
Мушкин М. И.	13.10.1999	ж	+7(351)		35347671	Челябинск	Неработающее население
Климов О. Д.	11.08.1970	м	+7(35)		34656321	Челябинск	Работающее население
Петрушкин И. И.	11.11.1980	ж	+7(921)	petrushkin@mail.ru	456R3452	Челябинск	Обучающиеся в обл.
Козушкин Э. В.	11.11.1981	м	+7(35)		987887A4	Минск	Работающее население
Седук Ж. П.	11.11.1980	м	+7(912)		340Y3214	Караганда	Работающее население
Кукушкина Т. Н.	11.11.1980	ж	+7(95)	ktamaek@mail.ru	909D4564	Челябинск	Неработающее население
Галактионов С. В.	11.12.1980	м	+7(351)		37C34291	Челябинск	Неработающее население
Прокорьев С. С.	12.12.1984	м	+7(91)	prokof@mail.ru	867y3413	Челябинск	Неработающее население
Чуркина Н. А.	28.01.1993	ж	+7(812)	nika@mail.ru	14232A36	Курган	Неработающее население
Шеста Р. В.	03.02.1993	м	+7(90)	sk1926@mail.ru	28333H71	Челябинск	Работающее население
Непривалый в. в.	11.11.1981	м	+7(912)		44383H32	Уфа	Неработающее население
Лушкин А. С.	11.11.1981	м	+7(91)		32332341	Челябинск	Работающее население

Рисунок 3.1 – Форма списка пациентов больницы

Для упрощения работы персонала в списке пациентов зеленым цветом выделены пациенты, которые в этом году уже проходят диспансеризацию, без выделения цветом отображены пациенты, которые не подвергаются диспансеризации.

ции в этом году, красным цветом выделены пациенты, которые подвергаются диспансеризации в этом году. Для этого происходит проверка пациентов: сначала на соответствие возрасту диспансеризации, затем, если пациент в этом году подходит по возрасту, происходит проверка на наличие карты диспансеризации в последние три года, так как карта учёта диспансеризации может быть создана для пациента каждые три года (т.е. пациент проходит диспансеризацию раз в три года). Если карта за последние три года уже есть, то новая не может быть создана. Данный отбор происходит при загрузке формы «ListOfPatients». Код обработчика события «ListOfPatients_Load» приведен в листинге 1.

Листинг 1 – Код обработчика события «ListOfPatients_Load»

```
int[] ages = { 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66,
69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99 };
for (int k = 0; k < zПациентыBindingSource.Count; k++)
    {
    DataRowView r = this.zПациентыBindingSource.Current as DataRowView;
    long kod = Convert.ToInt32(r.Row["Код"]);

    DateTime born = Convert.ToDateTime(r.Row["Дата_рождения"]);
    DateTime nowDate = DateTime.Today;
    int age = nowDate.Year - born.Year;
    for (int i = 0; i <= 26;)
    {
        int ok = 0;
        if (ages[i] == age)
            {
                DateTime date1;
                DateTime now_Date = DateTime.Today;
                int f = now_Date.Year - 3;

                карта_учета_диспансеризацииBindingSource.Filter = "Код_пациента=" + kod;
                if (карта_учета_диспансеризацииBindingSource.Count > 0)
                    {
                        for (int l = 0; l < карта_учета_диспансеризацииBindingSource.Count; l++)
                            {
                                DataRowView r1 = this.карта_учета_диспансеризацииBindingSource.Current as
                                DataRowView;

                                date1 = Convert.ToDateTime(r1.Row["Дата_начала"]);
                                int d = date1.Year;
                                if (d <= f) // за последние три года карты нет
                                    { ok = 1;
                                    }
                                else // уже есть карта за последние три года
                                    {
                                        ok = 2;
                                    }
                            }
                    }
            }
    }
```



```

карта_учета_диспансеризацииBindingSource.MoveNext();
}
if (ok == 1) // за последние три года карты нет
{
    int index = zПациентыDataGridView.CurrentRow.Index;

    zПациентыDataGridView.Rows[index].DefaultCellStyle.BackColor = Color.LightSalmon;
}
if (ok == 2) // уже есть карта за последние три года
{int index = zПациентыDataGridView.CurrentRow.Index;

zПациентыDataGridView.Rows[index].DefaultCellStyle.BackColor = Color.LightGreen;
}
ok = 0;
}
else //вообще нет карты
{
int index = zПациентыDataGridView.CurrentRow.Index;

zПациентыDataGridView.Rows[index].DefaultCellStyle.BackColor = Color.LightSalmon;
}

break;
}
else i++;
}
zПациентыBindingSource.MoveNext();
}
zПациентыBindingSource.MoveFirst();
}

```

Данная форма используется перед созданием карты учёта диспансеризации для выбора из пациентов больницы или создания нового пациента для диспансеризации. Для удобного поиска пациента предназначены фильтры по фамилии или по номеру больничной карты.

По кнопке «Новый» происходит добавление нового пациента для диспансеризации, т.е. пациента, который в этом году подвергается диспансеризации. Для этого при создании происходит проверка, а также, в случае наличия у созданного пациента электронной почты, происходит отправка пароля для входа в личный кабинет на сайте диспансеризации.

Код обработчика события «button1_Click» приведен в листинге 2.

Листинг 2 – Код обработчика события «button1_Click» «GetPass». Код функции представлен в листинге 3.

Листинг 3 – Код функции «GetPass»

```

public static string GetPass(int x)
{
    string pass = "";
    var r = new Random();
    while (pass.Length < x)
    {
        Char c = (char)r.Next(33, 125);
        if (Char.IsLetterOrDigit(c))
            pass += c;
    }
    return pass;
}

```

Для генерации хэша пароля используется функция «GenerateHash». Код функции представлен в листинге 4.

Листинг 4 – Код функции «GenerateHash»

```

public static string GenerateHash(string Password, int ISize)
{
    if (string.IsNullOrEmpty(Password)) return null;
    if (Password.Length < 1) return null;

    byte[] salt = new byte[_saltSize];
    byte[] key = new byte[_keySize];
    byte[] result = new byte[_saltSize + _keySize];

    try
    {
        using (RNGCryptoServiceProvider randBytes = new
RNGCryptoServiceProvider())
        {
            randBytes.GetBytes(salt);

            using (var hashBytes = new Rfc2898DeriveBytes(Password, salt,
ISize))
            {
                key = hashBytes.GetBytes(_keySize);
                Buffer.BlockCopy(salt, 0, result, 0, _saltSize);
                Buffer.BlockCopy(key, 0, result, _saltSize, _keySize);
            }
        }
        // returns salt/key pair
        return Convert.ToBase64String(result);
    }
    finally
    {
        if (salt != null)
            Array.Clear(salt, 0, salt.Length);
        if (key != null)
            Array.Clear(key, 0, key.Length);
        if (result != null)
            Array.Clear(result, 0, result.Length);
    }
}

```

```
}
```

Для отправки пароля используется функция «sendMail». Код функции представлен в листинге 5.

Листинг 5 – Код функции «GenerateHash»

```
void sendMail(string mail, string theme, string message)
{
    SmtplibClient Smtplib = new SmtplibClient("smtp.mail.ru", 25);

    Smtplib.Credentials = new NetworkCredential("micky_11_mouse@mail.ru",
        "1234567890q");
    Smtplib.EnableSsl = true;
    MailMessage msg = new MailMessage();
    msg.From = new MailAddress("micky_11_mouse@mail.ru");//от кого
    msg.To.Add(new MailAddress(mail));//кому
    msg.Subject = theme;
    msg.Body = message;
    Smtplib.Send(msg);
}
```

Для заполнения и изменения данных о пациентах создана форма редактирования персональных данных пациента. Форма «Карточка пациента» представлена на рисунке 3.2. Форма содержит необходимые к заполнению поля, по кнопке «ОК» происходит проверка корректного заполнения всех необходимых полей. В случае неверного заполнения или пустых значений появляется сообщение об ошибке.

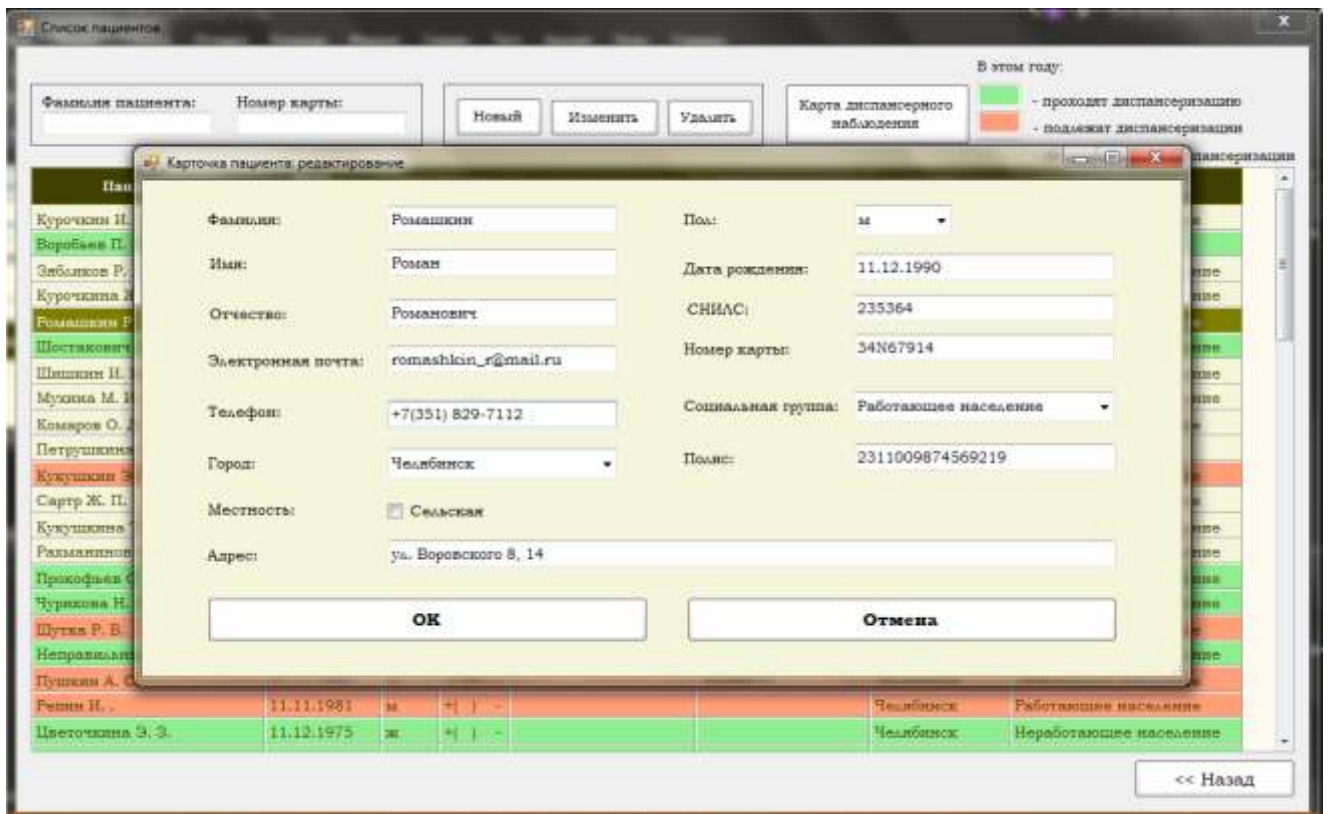


Рисунок 3.2 – Форма редактирования персональных данных пациента

Далее создаётся карта учёта, которая содержит всю информацию об этапах диспансеризации. Данная карта может быть создана на одного пациента каждые три года. Форма «Карта учёта диспансеризации» представлена на рисунке 3.3.

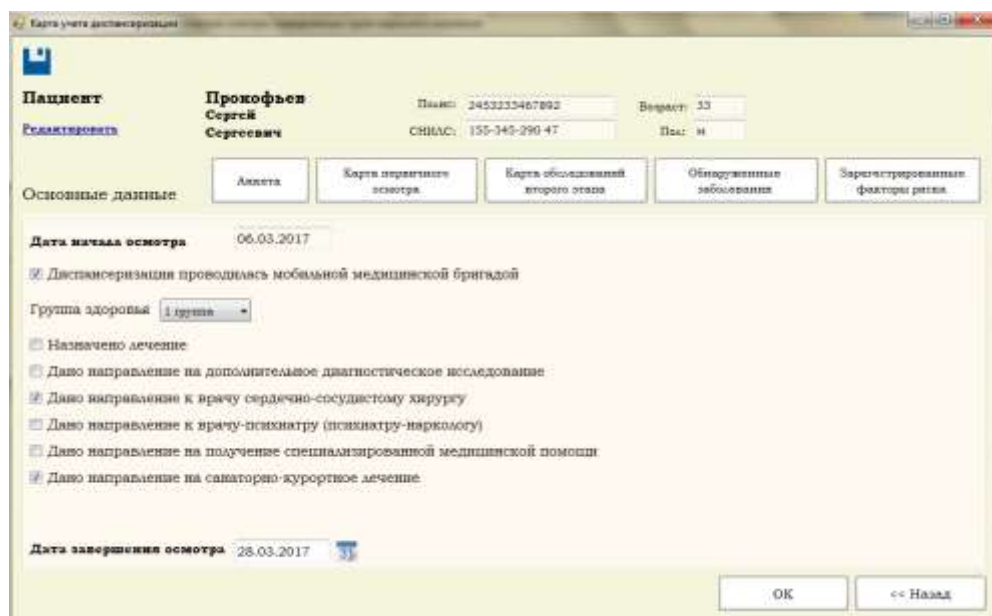
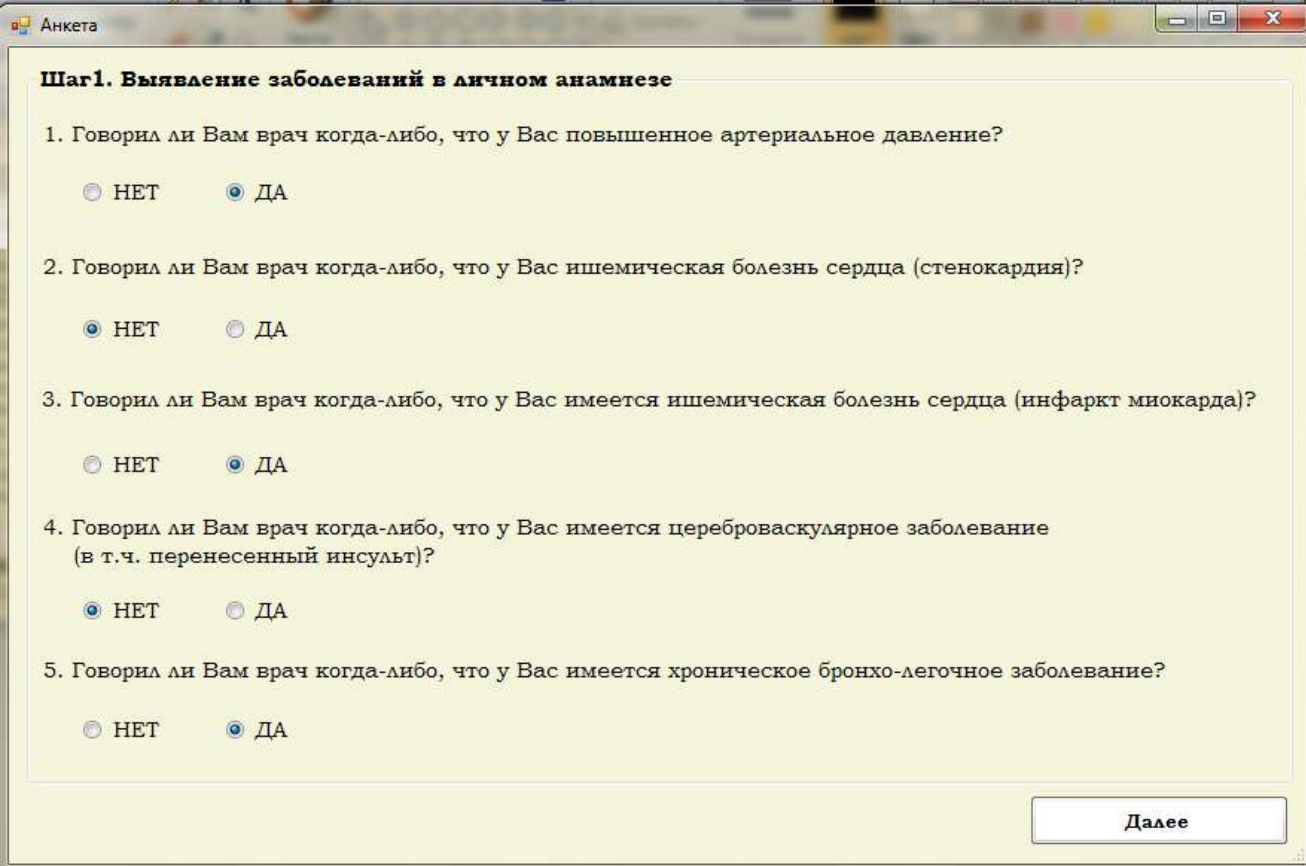


Рисунок 3.3 – Форма карты учёта диспансеризации

Карта учёта диспансеризации является центральным объектом системы и содержит информацию по всем этапам прохождения пациентом диспансеризации. В разрабатываемой подсистеме рассмотрим анкетирование и карту первичного осмотра.

Одной из составляющих первого этапа диспансеризации является анкетирование. Анкета содержит перечень вопросов по разным категориям и состоит из 11 разделов. Вопросы анкеты автоматически меняются в зависимости от пола и возраста пациента. Автоматизированное анкетирование необходимо для дальнейшего выявления факторов риска развития хронических неинфекционных заболеваний.

По кнопке «Анкета» вызывается форма анкетирования пациента (рис.3.4).



Анкета

Шаг 1. Выявление заболеваний в личном анамнезе

1. Говорил ли Вам врач когда-либо, что у Вас повышенное артериальное давление?

НЕТ ДА

2. Говорил ли Вам врач когда-либо, что у Вас ишемическая болезнь сердца (стенокардия)?

НЕТ ДА

3. Говорил ли Вам врач когда-либо, что у Вас имеется ишемическая болезнь сердца (инфаркт миокарда)?

НЕТ ДА

4. Говорил ли Вам врач когда-либо, что у Вас имеется цереброваскулярное заболевание (в т.ч. перенесенный инсульт)?

НЕТ ДА

5. Говорил ли Вам врач когда-либо, что у Вас имеется хроническое бронхо-легочное заболевание?

НЕТ ДА

Далее

Рисунок 3.4 – Форма анкетирования. Шаг 1

По кнопке «Далее» происходит запись ответов пациента на вопросы анкетирования в таблицу «Ответы пациентов». Код обработчика события `button1_Click` представлен в листинге 6.

Листинг 6 – Код обработчика события «button1_Click»

```
private void button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=1";

        if (ответы_пациентовBindingSource.Count == 0)
        {
            ответы_пациентовTableAdapter.Insert(Kard_Id, 1, 1, null);
        }
        else
        {
            DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
            int kod_ = Convert.ToInt32(g.Row["Код"]);
            ответы_пациентовTableAdapter.Update(Kard_Id, 1, 1, null,
kod_);
        }
    }
    else if (radioButton2.Checked == true)
    {
        ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=1";

        if (ответы_пациентовBindingSource.Count == 0)
        {
            ответы_пациентовTableAdapter.Insert(Kard_Id, 1, 2, null);
        }
        else
        {
            DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
            int kod_ = Convert.ToInt32(g.Row["Код"]);
            ответы_пациентовTableAdapter.Update(Kard_Id, 1, 2, null,
kod_);
        }
    }
    else { MessageBox.Show("Ответьте на первый вопрос!"); return; }

    if (radioButton5.Checked == true)
    {
        ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=2";

        if (ответы_пациентовBindingSource.Count == 0)
        {
            ответы_пациентовTableAdapter.Insert(Kard_Id, 2, 1, null);
        }
        else
        {

```

```

        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 2, 1, null,
kod_);
    }
}
else if (radioButton6.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=2";

    if (ответы_пациентовBindingSource.Count == 0)
    {
        ответы_пациентовTableAdapter.Insert(Kard_Id, 2, 2, null);
    }
    else
    {
        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 2, 2, null,
kod_);
    }
}
else { MessageBox.Show("Ответьте на второй вопрос!"); return; }

if (radioButton11.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=3";

    if (ответы_пациентовBindingSource.Count == 0)
    {
        ответы_пациентовTableAdapter.Insert(Kard_Id, 3, 1, null);
    }
    else
    {
        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 3, 1, null,
kod_);
    }
}
else if (radioButton12.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=3";

    if (ответы_пациентовBindingSource.Count == 0)
    {
        ответы_пациентовTableAdapter.Insert(Kard_Id, 3, 2, null);
    }
    else

```

```

        {
            DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
            int kod_ = Convert.ToInt32(g.Row["Код"]);
            ответы_пациентовTableAdapter.Update(Kard_Id, 3, 2, null,
kod_);
        }
    }
else { MessageBox.Show("Ответьте на третий вопрос!"); return; }

if (radioButton7.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=4";

    if (ответы_пациентовBindingSource.Count == 0)
    {
        ответы_пациентовTableAdapter.Insert(Kard_Id, 4, 1, null);
    }
    else
    {
        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 4, 1, null,
kod_);
    }
}
else if (radioButton8.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=4";

    if (ответы_пациентовBindingSource.Count == 0)
    {
        ответы_пациентовTableAdapter.Insert(Kard_Id, 4, 2, null);
    }
    else
    {
        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 4, 2, null,
kod_);
    }
}
else { MessageBox.Show("Ответьте на четвёртый вопрос!"); return; }

if (radioButton9.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=5";

    if (ответы_пациентовBindingSource.Count == 0)
    {

```



```

        ответы_пациентовTableAdapter.Insert(Kard_Id, 5, 1, null);
    }
    else
    {
        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 5, 1, null,
kod_);
    }
}
else if (radioButton10.Checked == true)
{
    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=5";

    if (ответы_пациентовBindingSource.Count == 0)
    {
        ответы_пациентовTableAdapter.Insert(Kard_Id, 5, 2, null);
    }
    else
    {
        DataRowView g = this.ответы_пациентовBindingSource.Current as
DataRowView;
        int kod_ = Convert.ToInt32(g.Row["Код"]);
        ответы_пациентовTableAdapter.Update(Kard_Id, 5, 2, null,
kod_);
    }
}
else { MessageBox.Show("Ответьте на пятый вопрос!"); return; }

this.Hide();
Anketa2 a = new Anketa2();
a.Pol = Pol;
a.Kard_Id = Kard_Id;
a.ShowDialog();
}
}

```

Для проведения анкетирования было создано 12 форм вопросов, включая дополнительные вспомогательные формы, а также 6 форм вопросов для возрастной категории «Старше 75 лет» в соответствии с нормативными документами о порядке проведения диспансеризации. На рисунке 3.5 представлена форма анкетирования с выбором ответа из дополнительной формы. Если ответ на 10 вопрос «Да», то появляется дополнительный вопрос и форма выбора ответа.

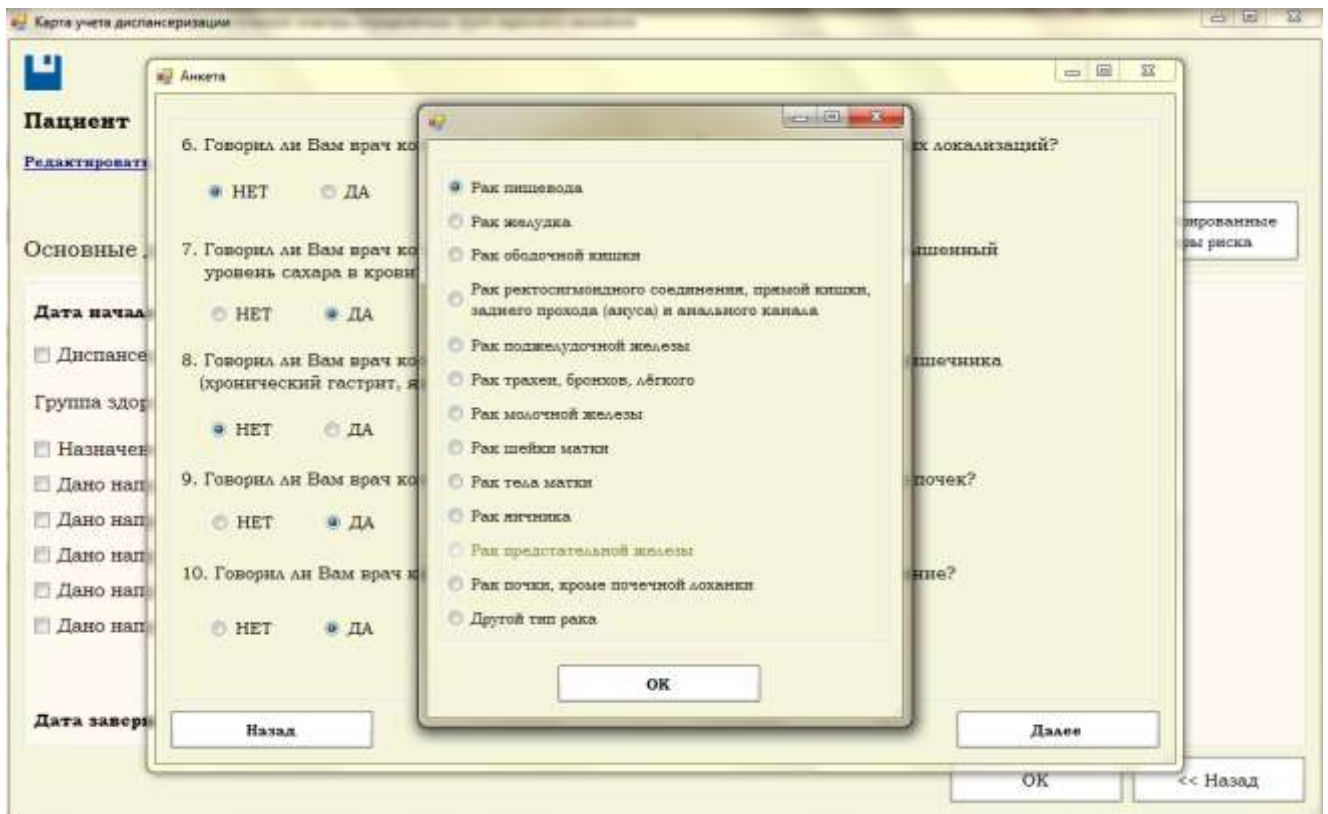


Рисунок 3.5 – Форма анкетирования.

Тип вопроса с выбором ответа из дополнительной формы

По кнопке «Назад» происходит заполнения соответствующих ответов, которые были даны, в случае необходимости изменения. Код обработчика события «Anketa1_Load» представлен в листинге 7.

Листинг 7 – Код обработчика события «Anketa1_Load»

```
private void Anketa1_Load(object sender, EventArgs e)
{
    rb_mas = new RadioButton[] { radioButton1, radioButton2, radioButton5,
    radioButton6, radioButton11, radioButton12, radioButton7, radioButton8,
    radioButton9, radioButton10 };

    this.f_Ответы_пациентов_по_карте_учетаTableAdapter.Fill(this.dispDataSet.f_Ответы_
    _пациентов_по_карте_учета, Kard_Id);
    f_Ответы_пациентов_по_карте_учетаBindingSource.Filter = "Код_вопроса
    > 0 AND Код_вопроса < 6";
    f_Ответы_пациентов_по_карте_учетаBindingSource.Sort = "Код_вопроса";
    if (f_Ответы_пациентов_по_карте_учетаBindingSource.Count > 0)
    {
```

```

int i = 0;
f_Ответы_пациентов_по_карте_учетаBindingSource.MoveFirst();
for (int k = 0; k <
f_Ответы_пациентов_по_карте_учетаBindingSource.Count; k++)
    {DataGridView dr =
f_Ответы_пациентов_по_карте_учетаBindingSource.Current as DataGridView;
    int answer;
    if (Int32.TryParse(dr.Row["Ответ"].ToString(), out answer))
    {
        if (answer == 1)
        {
            rb_mas[i].Checked = true;
        }
        else if (answer == 2)
        {
            rb_mas[i + 1].Checked = true;
        }
        i += 2;
    }
    f_Ответы_пациентов_по_карте_учетаBindingSource.MoveNext();
}
}

```

В анкете также реализованы связанные вопросы, в зависимости от ответа на один вопрос появляется или скрывается связанный с ним. На рисунке 3.6 представлены связанные вопросы. Если ответ на первый вопрос «Да», то появляется список «radioButton» и в базу записывается еще один ответ.

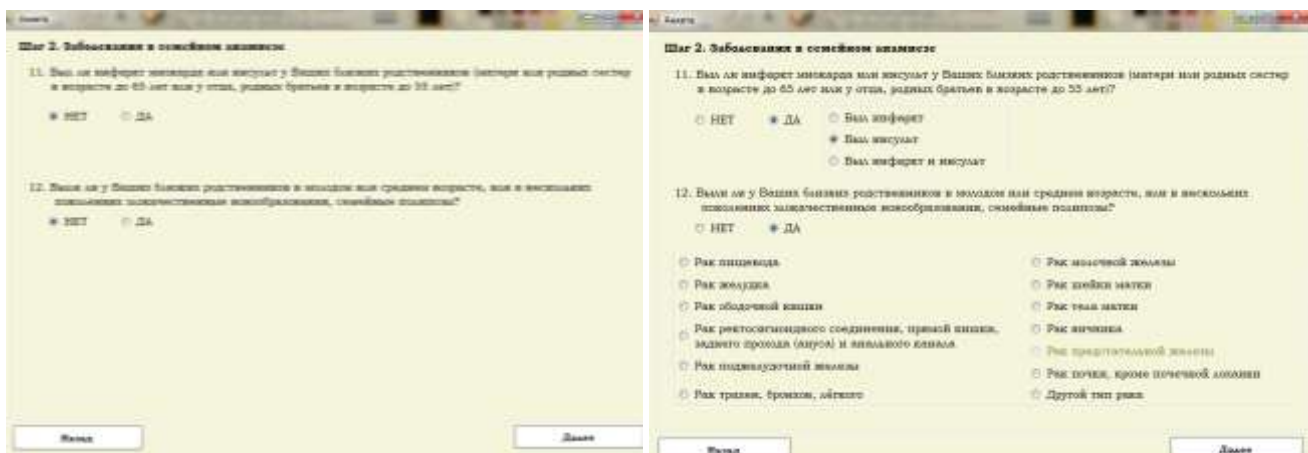


Рисунок 3.6 – Форма анкетирования. Связанные вопросы

На седьмом шаге анкетирования присутствуют связанные вопросы о курении. При выборе варианта ответа «Да» на 28ой вопрос появляются уточняющие вопросы с вводом ответа (рис. 3.7).

Анкета

25. Вызывает ли у Вас боль в области заднепроходного отверстия?

НЕТ ДА

26. Бывают ли у Вас кровянистые выделения с калом?

НЕТ ДА

27. Бывает ли у Вас неоформленный (полужидкий) черный или дегтеобразный стул?

НЕТ ДА

Шаг 7: Выявление табакокурения, показания к спирометрии и УЗИ брюшной аорты.

28. Курите ли Вы?

Нет, никогда не курил Курил в прошлом ДА

СКОЛЬКО СИГАРЕТ В ДЕНЬ?

СКОЛЬКО ЛЕТ?

Рисунок 3.7 – Форма анкетирования.
Связанные вопросы. Вопросы с вводом ответа

Также предусмотрена возможность изменения ответов на уже записанные вопросы, т.е. при возвращении назад поменять ответы пациента. Для этого используется процедура «ответы_пациентовTableAdapter.Update». Код обработчика события кнопки «Далее» (button1_Click) представлен в листинге 8.

Листинг 8 – Код обработчика события «button1_Click»

```
private void button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id + "AND
    Код_вопроса=27";
        if (ответы_пациентовBindingSource.Count == 0)
        {ответы_пациентовTableAdapter.Insert(Kard_Id, 27, 1, null);
        }
        else
        {
            DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
            int kod_ = Convert.ToInt32(g.Row["Код"]);
            ответы_пациентовTableAdapter.Update(Kard_Id, 27, 1, null, kod_);
        }
    }
    else if (radioButton2.Checked == true)
    {
```

```

    ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id + "AND
Код_вопроса=27";
    if (ответы_пациентовBindingSource.Count == 0)
    {ответы_пациентовTableAdapter.Insert(Kard_Id, 27, 2, null);
    }
    else
    {DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
    int kod_ = Convert.ToInt32(g.Row["Код"]);
    ответы_пациентовTableAdapter.Update(Kard_Id, 27, 2, null, kod_);
    }
    }
    else { MessageBox.Show("Ответьте на двадцать пятый вопрос!"); return; }
    if (radioButton3.Checked == true)
    { ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id +
"AND Код_вопроса=28";
    if (ответы_пациентовBindingSource.Count == 0)
    {ответы_пациентовTableAdapter.Insert(Kard_Id, 28, 1, null);
    }
    else
    {
    DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
    int kod_ = Convert.ToInt32(g.Row["Код"]);
    ответы_пациентовTableAdapter.Update(Kard_Id, 28, 1, null, kod_);
    }
    }
    else if (radioButton4.Checked == true)
    {ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id +
"AND Код_вопроса=28";
    if (ответы_пациентовBindingSource.Count == 0)
    {ответы_пациентовTableAdapter.Insert(Kard_Id, 28, 2, null);
    }
    else
    {DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
    int kod_ = Convert.ToInt32(g.Row["Код"]);
    ответы_пациентовTableAdapter.Update(Kard_Id, 28, 2, null, kod_);
    }
    }
    }
    else { MessageBox.Show("Ответьте на двадцать шестой вопрос!"); return; }
    if (radioButton5.Checked == true)
    {ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id +
"AND Код_вопроса=29";
    if (ответы_пациентовBindingSource.Count == 0)
    {ответы_пациентовTableAdapter.Insert(Kard_Id, 29, 1, null);
    }else
    {DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
    int kod_ = Convert.ToInt32(g.Row["Код"]);
    ответы_пациентовTableAdapter.Update(Kard_Id, 29, 1, null, kod_);
    }
    }
    }
    else if (radioButton6.Checked == true)
    {ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=29";
    if (ответы_пациентовBindingSource.Count == 0)
    {ответы_пациентовTableAdapter.Insert(Kard_Id, 29, 2, null);
    }
    }
}

```

```

else
{DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
int kod_ = Convert.ToInt32(g.Row["Код"]);
ответы_пациентовTableAdapter.Update(Kard_Id, 29, 2, null, kod_);
}}
else { MessageBox.Show("Ответьте на двадцать седьмой вопрос!"); return; }
if (radioButton7.Checked == true)
{ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id + "AND
Код_вопроса=33";
if (ответы_пациентовBindingSource.Count == 0)
{ответы_пациентовTableAdapter.Insert(Kard_Id, 33, 1, null);
}
}
else
{
DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
int kod_ = Convert.ToInt32(g.Row["Код"]);
ответы_пациентовTableAdapter.Update(Kard_Id, 33, 1, null, kod_);
ответы_пациентовTableAdapter.Delete(Kard_Id, 34);
ответы_пациентовTableAdapter.Delete(Kard_Id, 35);
}
}
else if (radioButton8.Checked == true)
{ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " +
Kard_Id + "AND Код_вопроса=33";
if (ответы_пациентовBindingSource.Count == 0)
{
ответы_пациентовTableAdapter.Insert(Kard_Id, 33, 2, null);
}
else
{
DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
int kod_ = Convert.ToInt32(g.Row["Код"]);
ответы_пациентовTableAdapter.Update(Kard_Id, 33, 2, null, kod_);
}
}
else if (radioButton9.Checked == true)
{ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id +
"AND Код_вопроса=33";
if (ответы_пациентовBindingSource.Count == 0)
{ответы_пациентовTableAdapter.Insert(Kard_Id, 33, 3, null);
}
}
else
{DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
int kod_ = Convert.ToInt32(g.Row["Код"]);
ответы_пациентовTableAdapter.Update(Kard_Id, 33, 3, null, kod_);
}
}
int kol, let;
try { kol = Convert.ToInt32(textBox2.Text); } catch { MessageBox.Show("Введите
количество сигарет в день!"); return; };
try { let = Convert.ToInt32(textBox1.Text); } catch { MessageBox.Show("Введите
количество лет!"); return; };
ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id + "AND
Код_вопроса=34";
if (ответы_пациентовBindingSource.Count == 0)
{ответы_пациентовTableAdapter.Insert(Kard_Id, 34, kol, null);
}

```

```

}
else
{DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
int kod_ = Convert.ToInt32(g.Row["Код"]);
ответы_пациентовTableAdapter.Update(Kard_Id, 34, kol, null, kod_);
}
ответы_пациентовBindingSource.Filter = "Код_диспансеризации = " + Kard_Id + "AND
Код_вопроса=35";
if (ответы_пациентовBindingSource.Count == 0)
{ответы_пациентовTableAdapter.Insert(Kard_Id, 35, let, null);
}
}
else
{DataRowView g = this.ответы_пациентовBindingSource.Current as DataRowView;
int kod_ = Convert.ToInt32(g.Row["Код"]);
ответы_пациентовTableAdapter.Update(Kard_Id, 35, let, null, kod_);
}
}
}
else { MessageBox.Show("Ответьте на двадцать восьмой вопрос!"); return; }
this.Hide();
Anketa8 a = new Anketa8();
a.Kard_Id = Kard_Id;
a.Pol = Pol;
a.ShowDialog();
}

```

После прохождения анкетирования, из главной формы «Карта учета диспансеризации» вызывается форма «Карта первичного осмотра» по соответствующей кнопке на панели. Данная форма является основной для первого этапа диспансеризации и содержит перечень полей для заполнения данных по антропометрии и другим первичным исследованиям, а также список обследований первого этапа (рисунок 3.7).

Обследования	Дата	Время проведения	Кабинет	Выполнено	Отложено	Отказ
Общий анализ мочи	03.04.2017	09:00	116	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Исследование мочи на окраску крива	03.04.2017	09:00	116	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Определение уровня общего холестерина в крови	03.04.2017	09:30	110	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Определение уровня глюкозы в крови экспресс-методом	03.04.2017	09:45	110	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Клинический анализ крови	03.04.2017	10:00	110	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Флюорография легких	04.04.2017	09:30	119	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Манжетграфия обеих верхних конечностей	04.04.2017	09:30	299	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Осмотр фаллопиевых (маточных)	04.04.2017	10:45	576	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Электрореоэнцефалография (в покое)	05.04.2017	08:00	389	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Измерения внутриглазного давления	06.04.2017	08:15	432	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Опрос (интервьюирование) на выявление хронической никотинозависимости	07.04.2017	12:40	223	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Антропометрия	07.04.2017	12:45	223	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Измерения артериального давления	07.04.2017	12:50	223	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Прямая (осмотр) кривизна позвоночника (в стоя)	07.04.2017	13:00	223	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 3.7 – Форма карты первичного осмотра

На рисунке 3.8 представлены поля, которые записываются в таблицу «Антропометрия».

Рост (см): Вес (кг):

Окружность талии (см):

Индекс массы тела: Избыточная масса тела

АД систолическое (мм рт. столба): Повышенное АД

АД диастолическое (мм рт. столба):

Общий холестерин (ммоль/л): Повышенный ХС

Глюкоза (ммоль/л): Повышенный сахар

Внутриглазное давление (OD):

Внутриглазное давление (OS): Повышенное ВГД

Относительный суммарный сердечно-сосудистый риск:
 label124

Рисунок 3.8 – Конструктор формы карты первичного осмотра.

Поля антропометрии

По кнопке «ОК» происходит проверка корректности заполнения необходимых полей. Код обработчика события нажатия кнопки «ОК» представлен в листинге 9.

Листинг 9 – Код обработчика события «button1_Click»

```
private void button1_Click(object sender, EventArgs e)
{
    float result = 0;
    int r;
    if (String.IsNullOrEmpty(textBox1.Text) || !Single.TryParse(textBox1.Text, out result))
    {
        DialogResult = 0;
        MessageBox.Show("Проверьте правильность ввода роста!");
        textBox1.Focus();
        return;
    }
    if (String.IsNullOrEmpty(textBox2.Text) || !Single.TryParse(textBox2.Text, out result))
    {
        DialogResult = 0;
        MessageBox.Show("Проверьте правильность ввода веса!");
        textBox2.Focus();
    }
}
```



```

return;
}
if (String.IsNullOrEmpty(textBox3.Text) || !Single.TryParse(textBox3.Text, out
result))
{DialogResult = 0;
  MessageBox.Show("Проверьте правильность ввода ОТ!");
  textBox3.Focus();
  return;
}
if (String.IsNullOrEmpty(textBox5.Text) || !Int32.TryParse(textBox5.Text, out r)
|| String.IsNullOrEmpty(textBox6.Text) || !Int32.TryParse(textBox6.Text, out r))
{DialogResult = 0;
  MessageBox.Show("Проверьте правильность ввода АД!");
  textBox5.Focus();
  textBox6.Focus();
  return;
}
if (String.IsNullOrEmpty(textBox8.Text) || !Single.TryParse(textBox8.Text, out
result))
{DialogResult = 0;
  MessageBox.Show("Проверьте правильность ввода холестерина!");
  textBox8.Focus();
  return;
}
if (String.IsNullOrEmpty(textBox9.Text) || !Single.TryParse(textBox9.Text, out
result))
{DialogResult = 0;
  MessageBox.Show("Проверьте правильность ввода глюкозы!");
  textBox9.Focus();
  return;
}
try
{VGD_od = Convert.ToSingle(textBox10.Text);
  VGD_os = Convert.ToSingle(textBox11.Text);
}
catch { if (textBox10.Visible == true) { DialogResult = 0;
MessageBox.Show("Проверьте правильность ввода ВГД!"); textBox10.BackColor = Col-
or.Red; textBox11.BackColor = Color.Red; return; } }
if(comboBox2.Enabled == true && (String.IsNullOrEmpty(comboBox2.Text) ||
String.IsNullOrEmpty(comboBox2.Text)))
{DialogResult = 0;
  MessageBox.Show("Определите сердечно сосудистый риск!");
  comboBox2.Focus();
  return;
}
if (textBox10.Visible == false) VGD_od = VGD_os = null;
антропометрияBindingSource.Filter = "Код_диспансеризации = " + Kard_ID;
if (антропометрияBindingSource.Count == 0)
{this.антропометрияTableAdapter.Insert(Height, Weight, ОТ, AD_sis, AD_dia,
Holesterin, Glucose, VGD_od, VGD_os, Kard_ID, comboBox2.Text, percent);
DialogResult = DialogResult.OK;
}else
{int kod__;
  int антро_kod;
  for (int k = 0; k < антропометрияBindingSource.Count; k++)

```

```

{
DataRowView g = this.антропометрияBindingSource.Current as DataRowView;
kod__ = Convert.ToInt32(g.Row["Код_диспансеризации"]);
if (Kard_ID == kod__)
{antro_kod = Convert.ToInt32(g.Row["Код"]);
this.антропометрияTableAdapter.Update(Height, Weight, OT, AD_sis, AD_dia,
Holesterin, Glucose, VGD_od, VGD_os, Kard_ID, comboBox2.Text, percent,
antro_kod);
DialogResult = DialogResult.OK;
break;
}антропометрияBindingSource.MoveNext();
}}
Close();
}

```

Поле «Индекс массы тела» рассчитывается исходя из введенных значений роста и веса (листинг 10).

Листинг 10 – Код обработчика события «textBox1_Leave»

```

private void textBox1_Leave(object sender, EventArgs e)
{
float h = 1; float w;
try
{
h = Convert.ToSingle(textBox1.Text);
if (h < 100 || h > 280)
{
textBox1.BackColor = Color.Red;
}
else { textBox1.BackColor = Color.Beige; }
}
catch { MessageBox.Show("Неверно введено значение роста!");
textBox1.Text = ""; textBox1.Focus(); return; }
try
{
w = Convert.ToSingle(textBox2.Text);
float index = w / ((h/100) * (h/100));
textBox4.Text = index.ToString();
}
catch { }
}

```

В зависимости от значения индекса массы тела определяется характеристика телосложения и появляется идентификатор (label) с нужным уведомлением: Острый дефицит массы тела, недостаточная масса тела, нормальная масса тела, избыточная масса тела, ожирение I степени, ожирение II степени, ожирение III степени (листинг 11).

Листинг 11 – Код обработчика события «textBox4_TextChanged»

```

private void textBox4_TextChanged(object sender, EventArgs e)
{if (Convert.ToSingle(textBox4.Text)<=15)
{label12.Visible = true;

```

```

    label12.Text = "Острый дефицит массы тела!";}
if (Convert.ToSingle(textBox4.Text) > 15 && Convert.ToSingle(textBox4.Text) <20)
{label12.Visible = true;
label12.Text = "Недостаточная масса тела!";}
if (Convert.ToSingle(textBox4.Text) >= 20 && Convert.ToSingle(textBox4.Text) <
25)
{label12.Visible = true;
label12.Text = "Нормальная масса тела"; }
if (Convert.ToSingle(textBox4.Text) >= 25 && Convert.ToSingle(textBox4.Text) <
30)
{label12.Visible = true;
label12.Text = "Избыточная масса тела";}
if (Convert.ToSingle(textBox4.Text) >= 30 && Convert.ToSingle(textBox4.Text) <
35)
{label12.Visible = true;
label12.Text = "Ожирение I степени";}
if (Convert.ToSingle(textBox4.Text) >= 35 && Convert.ToSingle(textBox4.Text) <
40)
{label12.Visible = true;
label12.Text = "Ожирение II степени";}
if (Convert.ToSingle(textBox4.Text) >= 40)
{label12.Visible = true;
label12.Text = "Ожирение III степени";}
}

```

В зависимости от введенных значений артериального давления также появляются уведомления: Пониженное артериальное давление, нормальное артериальное давление, I категория гипертензии, II категория гипертензии, III категория гипертензии (листинг 12).

Листинг 12 – Код обработчика события «textBox5_Leave»

```

private void textBox5_Leave(object sender, EventArgs e)
{try
{if (Convert.ToInt32(textBox5.Text) < 50 || Convert.ToInt32(textBox5.Text) >
300)
{textBox5.BackColor = Color.Red;
}
else { textBox5.BackColor = Color.Beige; }
}catch { }
try
{if (Convert.ToInt32(textBox5.Text) < 100 && (Convert.ToInt32(textBox6.Text) <
80))
{label13.Visible = true;
label13.Text = "Пониженное артериальное\n давление";
}
else if (Convert.ToInt32(textBox5.Text) >= 100 && (Convert.ToInt32(textBox5.Text)
< 130) && (Convert.ToInt32(textBox6.Text) >= 80) && (Con-
vert.ToInt32(textBox6.Text) < 90))
{label13.Visible = true;
label13.Text = "Нормальное артериальное\n давление";}
}
}

```

```

else if (Convert.ToInt32(textBox5.Text) >= 140 && (Convert.ToInt32(textBox5.Text) < 160) || (Convert.ToInt32(textBox6.Text) >= 90) && (Convert.ToInt32(textBox6.Text) < 100))
{label13.Visible = true;
label13.Text = "I категория гипертензии";
}
else if (Convert.ToInt32(textBox5.Text) >= 160 && (Convert.ToInt32(textBox5.Text) < 180) || (Convert.ToInt32(textBox6.Text) >= 100) && (Convert.ToInt32(textBox6.Text) < 110))
{label13.Visible = true;
label13.Text = "II категория гипертензии";
}
else if (Convert.ToInt32(textBox5.Text) >= 180 || (Convert.ToInt32(textBox6.Text) >= 110))
{label13.Visible = true;
label13.Text = "III категория гипертензии";
}
else { label13.Visible = false; }
}catch { } }

```

Таким образом, по части первичного осмотра здесь фиксируются измерения, а также предусмотрены автоматические индикаторы об отклонениях (рисунок). Красным цветом выделяются аномально высокие или низкие значения для данного показателя.



Рисунок 3.9 – Форма карты первичного осмотра

Список обследований первого этапа. Исходя из персональных данных, заполненных на форме карточки пациента, происходит автоматическое распределение плана диспансеризации по возрасту и полу, т.е. генерируется список необходимых к прохождению для каждого пациента базовых медицинских исследований. Данные обследования выбираются из установленного перечня медицин-

ских мероприятий в рамках диспансеризации, которые определяются отдельно для мужчин и для женщин в разные возрастные периоды. На рисунке 3.10 представлен установленный план диспансеризации в соответствии с приказом №36ан от 27 февраля 2015 года об утверждении порядка проведения диспансеризации. В соответствии с этим планом была разработана таблица «Обследования», описанная во втором разделе. По данной таблице и происходит отбор обследований для каждого конкретного пациента.

Приложение № 14

Объем диспансеризации и число медицинских мероприятий, составляющее 100 % и 85 % от объема обследований, установленного для данного возраста и пола гражданина при прохождении диспансеризации

Раздел I. Перечень медицинских мероприятий, проводимых в рамках диспансеризации у мужчин в определенные возрастные периоды. Число медицинских мероприятий, составляющее 100 % и 85 % (84 %) от объема обследования, установленного для данного возраста **мужчины**

Осмотр, исследование, иное медицинское мероприятие	Возраст (лет)																											
	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90	93	96	99	
Первый этап диспансеризации																												
1. Опрос (анкетирование) на выявление хронических неинфекционных заболеваний, факторов риска их развития, потребления наркотических средств и психотропных веществ без назначения врача	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2. Антропометрия (измерение роста стоя, массы тела, окружности талии), расчет индекса массы тела	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
3. Измерение артериального давления	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
4. Определение уровня общего холестерина в крови (допускается экспресс-метод)	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
5. Определение уровня глюкозы в крови экспресс-методом (допускается лабораторный)	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Рисунок 3.10 – План диспансеризации в соответствии с приказом №36ан от 27 февраля 2015 года об утверждении порядка проведения диспансеризации

Для формирования перечня обследований для пациента в соответствии с полом и возрастом реализована фильтрация по таблице «Обследования» (листинг 13).

Листинг 13 – Формирование перечня обследований для пациента

```
обследованияBindingSource.Filter = "Этап=1 AND [" + ages[i].ToString() + "] = true AND Пол LIKE'" + Po1 + "%' OR Этап=1 AND [" + ages[i].ToString() + "] = true AND Пол is null";
```

Каждое назначенное обследование сопровождается датой, временем и указанием кабинета для его прохождения. Для этого было создано расписание работы кабинетов для диспансеризации. У каждого кабинета существует определенная специализация. В соответствии с этим, у каждого обследования из общего

перечня установлена специализация и определенный интервал времени на его проведение. Далее образуется очередь, т.е. каждому пациенту назначается своё время прохождения тех или иных исследований.

На рисунке 3.11 представлена форма «Расписание работы кабинетов для диспансеризации». На данной форме расположено представление ZКабинеты, которое содержит номер кабинета, специализацию, диспансерный день недели, часы работы кабинета.

Кабинет	Специализация	День недели	Время начала работы	Время конца работы
110	Исследование крови	Понедельник	08:00	15:00
116	Исследование мочи	Понедельник	09:00	11:00
223	Общепедиатрически...	Пятница	11:00	17:00
354	Ультразвуковая	Вторник	10:00	17:00
432	Офтальмология	Четверг	08:00	18:30
119	Рентгенографический	Вторник	09:00	18:00
289	Кабинет маммолога	Вторник	09:30	18:00
576	Гинекология	Вторник	10:30	18:00
389	ЭКГ диагностика	Среда	08:00	17:00
440	Неврология	Среда	09:00	17:30
322	ФГДС диагностика	Среда	10:00	18:00
512	Кабинет уролога	Четверг	09:00	17:00
514	Кабинет проктолога	Среда	10:30	18:00
432	ЛОР-кабинет	Четверг	10:00	17:30

Рисунок 3.11 – Форма расписания работы кабинетов для прохождения диспансеризации

Также предусмотрена возможность изменения часов работы кабинета (рис. 3.12). Код обработчика события «zКабинетыDataGridView_CellDoubleClick_1» представлен в листинге 14.

Листинг 14 – Код обработчика события «CellDoubleClick»

```
private void zКабинетыDataGridView_CellDoubleClick_1(object sender,
DataGridViewCellEventArgs e)
{Edit_cab a = new Edit_cab();
DataRowView g = this.zКабинетыBindingSource1.Current as DataRowView;
a.id = Convert.ToInt32(g.Row["Код"]);
a.day = g.Row["День_недели"].ToString();
try { a.time1 = DateTime.Parse(g.Row["Время_начала_работы"].ToString()); } catch
{ };
```

```

try { a.time2 = DateTime.Parse(g.Row["Время_конца_работы"].ToString()); } catch
{ };
DialogResult res = a.ShowDialog();
if (res == DialogResult.OK)
{zКабинетыTableAdapter.Fill(dispDataSet.zКабинеты);
}}

```

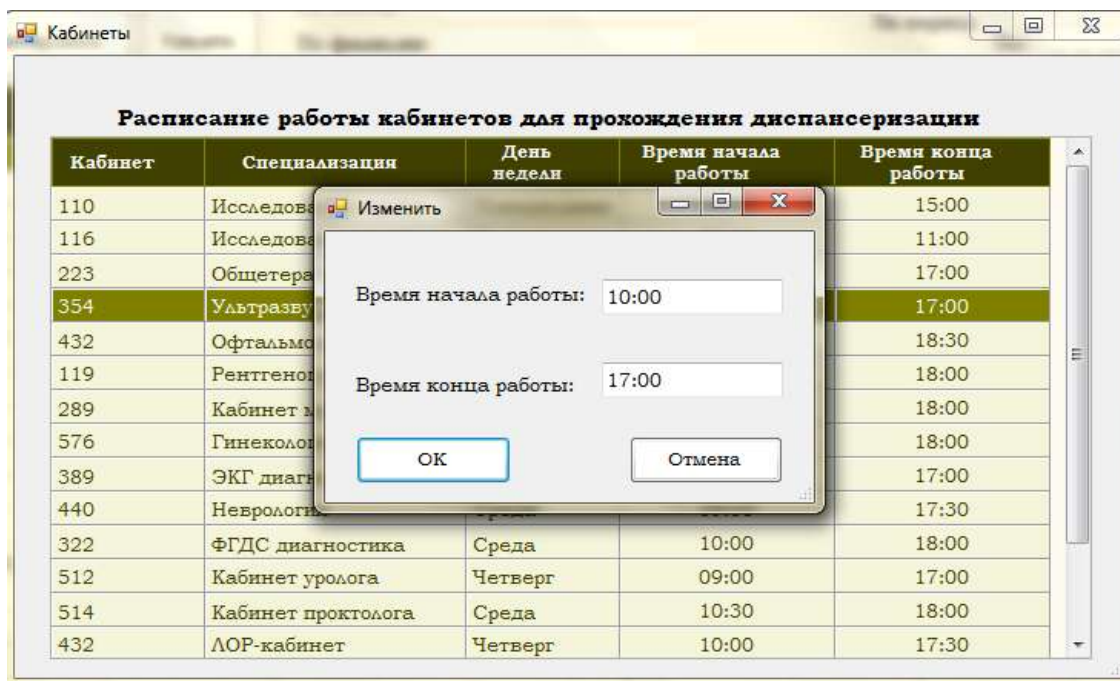


Рисунок 3.12 – Форма расписания работы кабинетов для прохождения диспансеризации

Формирование списка обследований с указанием даты, времени и кабинета прохождения происходит при загрузке формы «Карта первичного осмотра» (событие «Antropomertia_Load»).

Алгоритм формирования: в соответствии с указанной для каждого обследования специализацией в таблице «Обследования» выбирается кабинет с соответствующей специализацией (кабинетыBindingSource.Filter="Код_специализации = " + kod_spec1, где kod_spec1 – код специализации обследования). Далее, находится данный кабинет в таблице «Время работы кабинетов», устанавливается день недели, время начала работы кабинета. Затем необходимо установить последнее занятое время проведения обследования в данном кабинете и установить дату последнего проведения обследования. Для этого ис-

пользуется функция fПоследнее_занятое_время2. Код функции приведен в листинге 15.

Листинг 15 – Код функции «fПоследнее_занятое_время2»

```
ALTER FUNCTION [dbo].[fПоследнее_занятое_время2] ( @kod_cab int, @etap int)
RETURNS
@a TABLE
(
Код int,
Код_диспансеризации bigint,
Код_обследования int,
Код_кабинета int,
Дата_проведения datetime,
Время_проведения time(7),
Выявлено_отклонение bit,
Отказ bit,
Этап int
)
AS
BEGIN
INSERT INTO @a
SELECT TOP (1) dbo.[Обследования по пациенту].Код, dbo.[Обследования по пациен-
ту].Код_диспансеризации, dbo.[Обследования по пациенту].Код_обследования,
dbo.[Обследования по пациенту].Код_кабинета, dbo.[Обследования по пациен-
ту].Дата_проведения, dbo.[Обследования по пациенту].Время_проведения,
dbo.[Обследования по пациенту].Выявлено_отклонение, dbo.[Обследования по пациен-
ту].Отказ, dbo.Обследования.Этап
FROM dbo.[Обследования по пациенту] INNER JOIN
dbo.Обследования ON dbo.[Обследования по пациен-
ту].Код_обследования = dbo.Обследования.Код
WHERE (dbo.[Обследования по пациенту].Код_кабинета = @kod_cab AND
dbo.Обследования.Этап = @etap)
ORDER BY dbo.[Обследования по пациенту].Код DESC
RETURN
END
```

Если обследований в данном кабинете еще не проводилось, назначаем на выбранное обследования время начала работы данного кабинета и ближайшую дату на день недели, в который работает данный кабинет по диспансеризации. Для выбора следующего ближайшего указанного дня недели используется функция Next1. Код функции представлен в листинге 16.

Листинг 16 – Код функции «Next1»

```
public static DateTime Next1(DateTime start, DayOfWeek day)
{
do
start = start.AddDays(+1);
while (start.DayOfWeek != day);
```



```

        return start;
    }

```

Далее, проверяем существование других обследований у пациента в этот день и в это время (используем фильтр по таблице «Обследований по пациенту»):

```

обследования_по_пациентуBindingSource.Filter = @" Дата_проведения= " +
d1.ToShortDateString() + @" AND Код_диспансеризации =" + Kard_ID;

```

Если данное время уже занято, то сдвигаем время обследования на установленную норму.

Кроме того, необходимо проверка на конец возможных назначений обследований в конкретный день. Если часы работы закончены, то назначение прохождения обследования проходит на следующую неделю в соответствующий день недели. Полный код формирования списка обследований первого этапа представлен в листинге 17.

Листинг 17 – Формирование списка обследований первого этапа

```

int kod1, kod_spec1, norma = 0 ;
int kod_cab = 0; int kod_cab1 = 0;
TimeSpan time, at_time;
TimeSpan time_start, time_end;
DataRowView r1, r2, r3, r4;
DateTime date, d1;
string day;
for (int k = 0; k < обследованияBindingSource.Count; k++)
{
    r1 = this.обследованияBindingSource.Current as DataRowView;
    kod1 = Convert.ToInt32(r1.Row["Код"]);
    kod_spec1 = Convert.ToInt32(r1.Row["Код_специализации"]);
    try { norma = Convert.ToInt32(r1.Row["Норма_времени"]); } catch { };
    кабинетыBindingSource.Filter= "Код_специализации = " + kod_spec1;
    if (кабинетыBindingSource.Count>0)
    {r2 = this.кабинетыBindingSource.Current as DataRowView;
      kod_cab = Convert.ToInt32(r2.Row["Код"]);
    }

    время_работы_кабинетовBindingSource.MoveFirst();

    for (int p = 0; p < время_работы_кабинетовBindingSource.Count; p++)
    {r3 = this.время_работы_кабинетовBindingSource.Current as DataRowView;
      kod_cab1 = Convert.ToInt32(r3.Row["Код_кабинета"]);
      day = r3.Row["День_недели"].ToString();
      if (kod_cab == kod_cab1)
      {time_start = TimeSpan.Parse(r3.Row["Время_начала_работы"].ToString());

```

```

fПоследнее_занятое_время2TableAdapter.Fill(dispDataSet.fПоследнее_занятое_время2,
kod_cab,1);

if (fПоследнее_занятое_время2BindingSource.Count == 0)
{if (day == "Понедельник")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Monday);
}
else if (day == "Вторник")
{
d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Tuesday);
}
else if (day == "Среда")
{
d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Wednesday);
}
else if (day == "Четверг")
{
d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Thursday);
}
else
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Friday);
}
time = time_start;
обследования_по_пациентуBindingSource.Filter = @" Дата_проведения= '" +
d1.ToShortDateString() + @"' AND Код_диспансеризации =" + Kard_ID;
for (int b = 0; b < обследования_по_пациентуBindingSource.Count; b++)
{DataRowView q = this.обследования_по_пациентуBindingSource.Current as
DataRowView;
TimeSpan time1 = TimeSpan.Parse(q.Row["Время_проведения"].ToString());
if (time1== time)
{time = time + TimeSpan.FromMinutes(norma);
}
обследования_по_пациентуBindingSource.MoveNext();
}
}
else
{r4 = this.fПоследнее_занятое_время2BindingSource.Current as DataRowView;
at_time = TimeSpan.Parse(r4.Row["Время_проведения"].ToString());
date = DateTime.Parse(r4.Row["Дата_проведения"].ToString());
time_end = TimeSpan.Parse(r3.Row["Время_конца_работы"].ToString());
if (at_time >= time_end)
{if (date > DateTime.Now)
{time = time_start;
if (day == "Понедельник")
{d1 = dayofweek.Next1(date, DayOfWeek.Monday);
}
else if (day == "Вторник")
{
d1 = dayofweek.Next1(date, DayOfWeek.Tuesday);
}
else if (day == "Среда")
{
d1 = dayofweek.Next1(date, DayOfWeek.Wednesday);
}
else if (day == "Четверг")

```

```

        {
            d1 = dayofweek.Next1(date, DayOfWeek.Thursday);
        }
else
{d1 = dayofweek.Next1(date, DayOfWeek.Friday);
}
}
else
{time = time_start;
if (day == "Понедельник")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Monday);
}
else if (day == "Вторник")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Tuesday);
}
else if (day == "Среда")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Wednesday);
}
else if (day == "Четверг")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Thursday);
} else
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Friday);
}}
}
else
{if (date > DateTime.Now)
{d1 = date;
time = TimeSpan.FromMinutes(norma) + at_time;
}
else
{if (day == "Понедельник")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Monday);
}
else if (day == "Вторник")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Tuesday);
}
else if (day == "Среда")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Wednesday);
}
else if (day == "Четверг")
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Thursday);
}
else
{d1 = dayofweek.Next1(DateTime.Now, DayOfWeek.Friday);
}
time = time_start;
}}}}
if (kod1 == 19 || kod1 == 20)
{обследования_по_пациентуTableAdapter.Insert(Kard_ID, kod1, null, false,
time_start, d1, kod_cab);
}
else обследования_по_пациентуTableAdapter.Insert(Kard_ID, kod1, null, false,
time, d1, kod_cab);
break; }
время_работы_кабинетовBindingSource.MoveNext();}

```

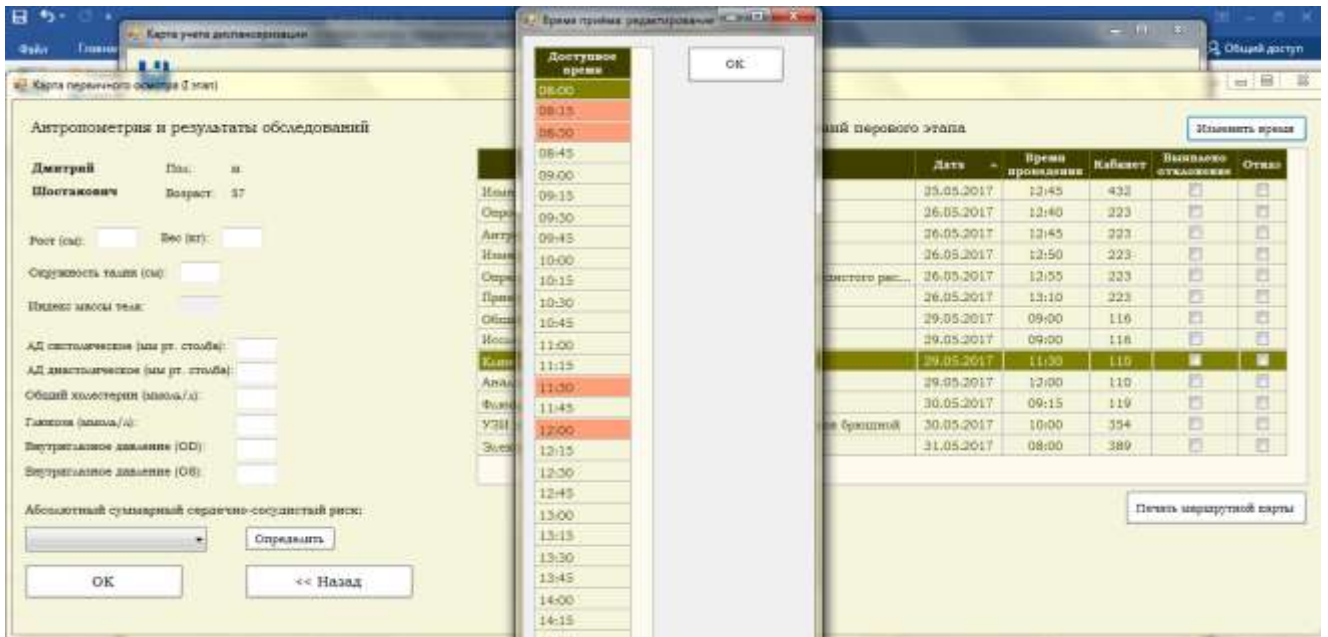



Рисунок 3.14 – Форма редактирования результатов обследований

Код обработчика события нажатия кнопки «Изменить время» представлен в листинге 18.

Листинг 18 – Код обработчика события «button3_Click»

```
private void button3_Click(object sender, EventArgs e)
{
    DataGridView v = fОбследованияПоКартеУчетаBindingSource.Current as
DataGridView;
    int kod = Convert.ToInt32(v.Row["Код_обследования"]);
    Edit_time a = new Edit_time();
    DataGridView g = this.fОбследованияПоКартеУчетаBindingSource.Current
as DataGridView;

    a.id = Convert.ToInt32(g.Row["Код"]);
    a.kod_obsled = Convert.ToInt32(g.Row["Код_обследования"]);
    a.kod_disp = Convert.ToInt32(g.Row["Код_диспансеризации"]);
    a.date = Convert.ToDateTime(g.Row["Дата_проведения"].ToString());
    a.time = TimeSpan.Parse(g.Row["Время_проведения"].ToString());
    a.cab = Convert.ToInt32(g.Row["Код_кабинета"]);

    if      ((DateTime.Parse(g.Row["Дата_проведения"].ToString()) <
DateTime.Now))
    {
        MessageBox.Show("Нельзя изменить время приёма!"); return;
    }
    else
    {
        DialogResult res = a.ShowDialog();
        if (res == DialogResult.OK)

```

```

    {
fОбследования_По_Карте_УчетаTableAdapter.Fill(dispatchDataSet.fОбследования_По_Карте_
Учета, Kard_ID, 1);
for (int k = 0; k < fОбследованияПоКартеУчетаBindingSource.Count; k++)
    {
DataRowView g1 = this.fОбследованияПоКартеУчетаBindingSource.Current as
DataRowView;
bool fail = Convert.ToBoolean(g1.Row["Отказ"]);
if (fail == true)
{int index = zОбследования_по_пациентуDataGridView.CurrentRow.Index;
zОбследования_по_пациентуDataGridView.Rows[index].DefaultCellStyle.BackColor =
Color.LightSalmon;
}
fОбследованияПоКартеУчетаBindingSource.MoveNext();
    }
}
}
}

```

Если дата проведения обследования уже прошла, то время проведения нельзя изменить (рисунок 3.15).

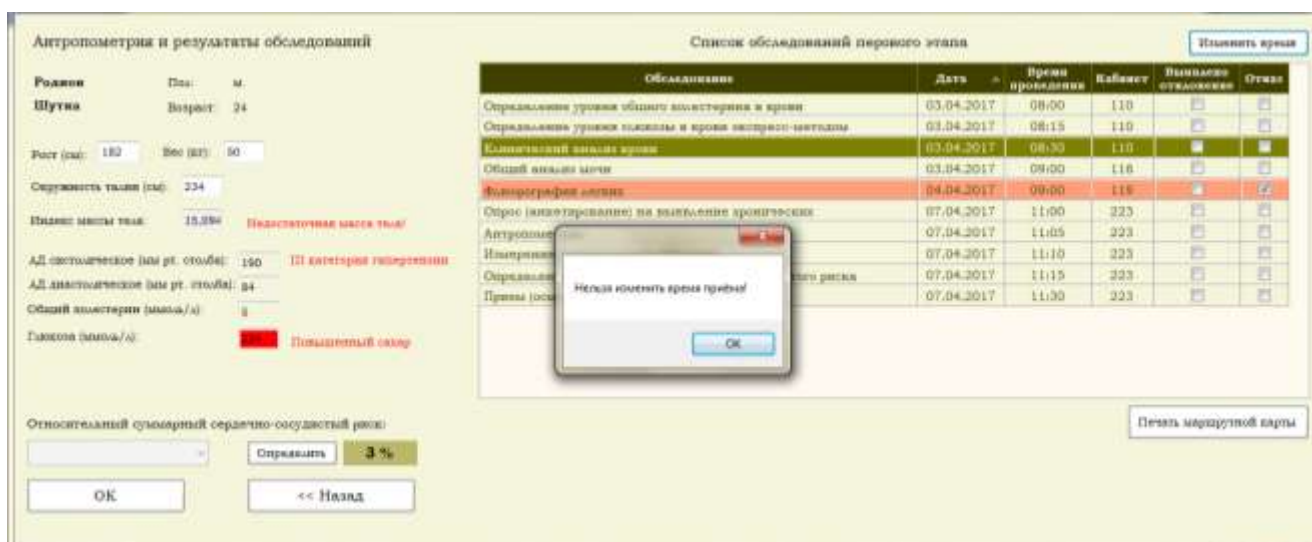


Рисунок 3.15 – Сообщение об ошибке изменения времени приёма

При загрузке формы изменения времени приёма происходит выделение занятых приёмов. Код данного отбора представлен в листинге 19.

Листинг 19 – Код обработчика события загрузки формы «Edit_time_Load»

```

private void Edit_time_Load(object sender, EventArgs e)
{
DataRowView r, r1;
TimeSpan time2, time1;
int kod_cab;
int index;
r = this.расписание_приёмов_кабинетаBindingSource.Current as DataRowView;

```

```

kod_cab = Convert.ToInt32(r.Row["Код_кабинета"]);

обследования_по_пациентуBindingSource.Filter = @" Дата_проведения= '' +
date.ToShortDateString() + @"' AND Код_кабинета = '' + kod_cab + @'";

for (int k = 0; k < обследования_по_пациентуBindingSource.Count; k++)
{
r1 = this.обследования_по_пациентуBindingSource.Current as DataRowView;
time1 = TimeSpan.Parse(r1.Row["Время_проведения"].ToString());
расписание_приёмов_кабинетаBindingSource.MoveFirst();
for (int p = 0; p < расписание_приёмов_кабинетаBindingSource.Count; p++)
{
r = this.расписание_приёмов_кабинетаBindingSource.Current as DataRowView;
time2 = TimeSpan.Parse(r.Row["Время"].ToString());
if (time2 == time1)
{
index = zОбследования_по_пациентуDataGridView.CurrentRow.Index;
zОбследования_по_пациентуDataGridView.Rows[index].DefaultCellStyle.BackColor =
Color.LightSalmon;
break;
}
else расписание_приёмов_кабинетаBindingSource.MoveNext();
}
обследования_по_пациентуBindingSource.MoveNext();
}
расписание_приёмов_кабинетаBindingSource.MoveFirst();
}

```

По кнопке «ОК» происходит проверка, если выбрано занятое время, то появляется сообщение об ошибке (рис.3.16). Сообщение об ошибке также появляется в случае, если выбранное время занято другим обследованием у данного пациента (обследованием другой специализации в другом кабинете). В случае, если выбор сделан верно, происходит изменение времени прохождения обследования в таблице «Обследования по пациенту». Код обработчика события `button1_Click` представлен в листинге 20.

Листинг 20 – Код обработчика события «`button1_Click`»

```

private void button1_Click(object sender, EventArgs e)
{
    DataRowView r=this.расписание_приёмов_кабинетаBindingSource.Current
as DataRowView;
    int index = zОбследования_по_пациентуDataGridView.CurrentRow.Index;
    int f = 0;
    TimeSpan time2 = TimeSpan.Parse(r.Row["Время"].ToString());
    if
(zОбследования_по_пациентуDataGridView.Rows[index].DefaultCellStyle.BackColor ==
Color.LightSalmon)
    {
        DialogResult = 0;
        MessageBox.Show("Выбранное время занято!"); return;
    }
}

```

```

    }
    обследования_по_пациентуBindingSource.Filter = @" Дата_проведения= '" +
    date.ToShortDateString() + @"' AND Код_диспансеризации = " + kod_disp;
    for (int k = 0; k < обследования_по_пациентуBindingSource.Count; k++)
    {
        DataRowView r1 = this.обследования_по_пациентуBindingSource.Current as
        DataRowView;
        TimeSpan time1 = TimeSpan.Parse(r1.Row["Время_проведения"].ToString());
        if(time1==time2)
        {
            f = 1;
            break; }
        else обследования_по_пациентуBindingSource.MoveNext();
        {
            if(f==1)
            {
                DialogResult = 0;
                MessageBox.Show("Выбранное время занято другим обследованием у данного пациен-
                та!"); return;
            }
            else
            {
                обследования_по_пациентуTableAdapter.Update(kod_disp, kod_obsled, null, false,
                time2, date, cab, id);
            }
            Close();
        }
    }
}

```

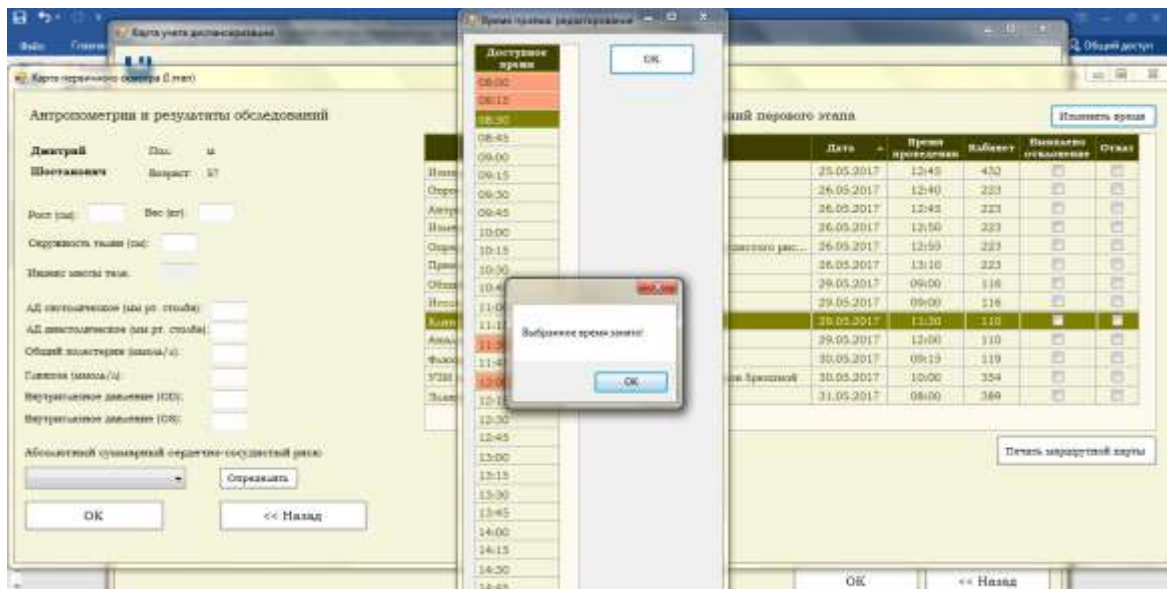


Рисунок 3.16 – Сообщение об ошибке изменения времени приёма

В результате для пациента реализуется печатная форма маршрутной карты с указанием списка необходимых к прохождению обследований (рисунок 3.17).

Код обработчика события нажатия кнопки «Печать маршрутной карты» представлен в листинге 21.

Листинг 21 – Код обработчика события «button4_Click»

```
private void button4_Click(object sender, EventArgs e)
{
    Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();
        ExcelApp.Visible = false;
        Excel.Workbook ExcelWorkBook;
        Excel.Worksheet ExcelWorkSheet;
    ExcelWorkBook = ExcelApp.Workbooks.Add(System.Reflection.Missing.Value);

    ExcelWorkSheet = (Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);
    ExcelWorkSheet.Cells[1, 1] = "Маршрутная карта";
        var cell = ExcelWorkSheet.get_Range("A1");

        cell.EntireRow.Font.Size = 11;
        cell.EntireRow.Font.Name = "Bookman Old Style";
        cell.EntireRow.Font.Bold = true;
        cell.EntireRow.Font.ColorIndex = 5;

    for (int i = 1; i < zОбследования_по_пациентуDataGridView.Columns.Count-6; i++)
        {
        ExcelWorkSheet.Cells[3, i] = zОбследования_по_пациентуDataGridView.Columns[i -
        1].HeaderText;
        }
    ExcelWorkSheet.Cells[3, 1].EntireRow.Font.Bold = true;
    ExcelWorkSheet.Cells[3, 1].EntireRow.Font.Size = 11;

    for (int i = 0; i < zОбследования_по_пациентуDataGridView.Rows.Count; i++)
    {ExcelWorkSheet.Cells[i + 4, 3].NumberFormat = "чч:мм";
    for (int j = 0; j < zОбследования_по_пациентуDataGridView.Columns.Count-7; j++)
        {
        ExcelWorkSheet.Cells[i + 4, j + 1] =
        zОбследования_по_пациентуDataGridView.Rows[i].Cells[j].Value.ToString();
        }
    }
    var count = zОбследования_по_пациентуDataGridView.Rows.Count + 3;
    var cells = ExcelWorkSheet.get_Range("A3", "D" + count);
    cells.WrapText = false;
    cells.Font.Name = "Bookman Old Style";

    cells.Borders[Microsoft.Office.Interop.Excel.XlBordersIndex.xlInsideVertical].LineStyle
    = Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous
    cells.Borders[Microsoft.Office.Interop.Excel.XlBordersIndex.xlInsideHorizontal].LineStyle
    = Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
    cells.Borders[Microsoft.Office.Interop.Excel.XlBordersIndex.xlEdgeTop].LineStyle
    = Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
    cells.Borders[Microsoft.Office.Interop.Excel.XlBordersIndex.xlEdgeRight].LineStyle
    = Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
    cells.Borders[Microsoft.Office.Interop.Excel.XlBordersIndex.xlEdgeLeft].LineStyle
    = Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
```

```
cells.Borders[Microsoft.Office.Interop.Excel.XlBordersIndex.xlEdgeBottom].LineStyle = Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
```

```

ExcelWorkSheet.Cells[3, 1].EntireColumn.ColumnWidth = 64.57;
ExcelWorkSheet.Cells[3, 2].EntireColumn.ColumnWidth = 11.86;
ExcelWorkSheet.Cells[3, 3].EntireColumn.ColumnWidth = 23;
ExcelWorkSheet.Cells[3, 4].EntireColumn.ColumnWidth = 10.71;
ExcelApp.Visible = true;
}

```

Маршрутная карта			
Обследование	Дата	Время проведения	Кабинет
Общий анализ мочи	03.04.2017	09:00	116
Исследование кала на скрытую кровь	03.04.2017	09:00	116
Определение уровня общего холестерина в крови	03.04.2017	09:30	110
Определение уровня глюкозы в крови экспресс-методом	03.04.2017	09:45	110
Клинический анализ крови	03.04.2017	10:00	110
Флюорография легких	04.04.2017	09:30	119
Маммография обеих молочных желез	04.04.2017	09:50	289
Осмотр фельдшером (акушеркой)	04.04.2017	10:45	576
Электрокардиография (в покое)	05.04.2017	08:00	389
Измерение внутриглазного давления	06.04.2017	08:15	432
Опрос (анкетирование) на выявление хронических неинфекционных заболеваний	07.04.2017	12:40	223
Антропометрия	07.04.2017	12:45	223
Измерение артериального давления	07.04.2017	12:50	223
Прием (осмотр) врача-терапевта (I этап)	07.04.2017	13:05	223

Рисунок 3.17 – Маршрутная карта обследований первого этапа

Вывод по разделу три

В разделе реализации программного продукта разработано приложение для автоматизации подсистемы сбора первичных данных диспансеризации в среде Visual Studio, отвечающее поставленным требованиям.

4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

Разрабатываемая информационная система «Диспансеризация» предназначена для бюджетных учреждений. Данный проект не планируется внедрять на коммерческой основе.

При принятии решения о создании и внедрении информационной системы «Диспансеризация» необходимо рассчитать затраты на разработку.

4.1 Расчет затрат

В таблице 2 приведены затраты на основные и вспомогательные материалы, используемые при работе.

Таблица 2 – Затраты на основные и вспомогательные материалы

Наименование материала	Единицы измерения	Стоимость единицы материала, руб.	Кол-во	Общая стоимость материала, руб.
Бумага А4	Упаковка (500лист.)	250.00	1	250,00
Ручка шариковая	штук	20.00	2	40,00
Карандаш	штук	15.00	2	30,00
Запоминающее устройство USB-	штук	350.00	1	350,00
Заправка 1 картриджа для принтера марки HP (лазерный монохромный)	штук	300.00	1	300,00
Итого за основные материалы:				970,00
Вспомогательные материалы (10% от основных):				97,00
Итого за основные и вспомогательные материалы:				1067,00
Транспортно-заготовительные расходы (20% от суммы расходов на основные и вспомогательные материалы):				213,4
Итого затраты на основные и вспомогательные материалы с учетом транспортных расходов:				1280,4

Лицензионное программное обеспечение и используемый ПК являются бесплатными.

Расчет заработной платы работников, разрабатывающих и внедряющих проект, приведен в таблице 3.

Таблица 3 – Расчет заработной платы

Должность	Оклад, руб./мес.	Оплата, руб./день	Продолжительность работ, дни	Итого, руб.
Инженер-программист 1	25 000	1190	20	23 800
Инженер-программист 2	25 000	1190	20	23 800
Инженер-программист 3	25 000	1190	20	23 800
Итого:				71 400
Дополнительная заработная плата (20% от основной)				14 280
Основная и дополнительная заработная плата				85 680
Социальные платежи (30% от основной и дополнительной заработной платы)				25 704

В результате затраты на разработку и внедрение программного продукта составляют 112 664,4 руб.

Вывод по разделу четыре

В организационно-экономическом разделе проанализированы и рассчитаны статьи затрат на разработку и внедрение проекта.

ЗАКЛЮЧЕНИЕ

В процессе выполнения научно-исследовательской работы поставлены цели и задачи, обоснована актуальность и необходимость выбранной темы, проанализированы процессы, происходящие в предметной области, произведен обзор аналогов, обоснование выбора программных средств разработки, спроектирована база данных MS SQL Server, подробно рассмотрены созданные таблицы и их связи, а также алгоритм реализации поставленных задач.

Разработано приложение, автоматизирующее процессы подсистемы сбора первичных данных. Разработанный программный продукт отвечает всем поставленным требованиям и выполнен в соответствии с техническим заданием.

Таким образом, цель работы достигнута. Решена поставленная задача, а именно автоматизация первого этапа диспансеризации взрослого населения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бойцов, С.А. Организация проведения диспансеризации определенных групп взрослого населения. Методические рекомендации по практической реализации приказа Минздрава России от 3 февраля 2015 г. № 36ан «Об утверждении порядка проведения диспансеризации определенных групп взрослого населения». Утверждены главным специалистом по профилактической медицине Минздрава России 23 марта 2015 г. / Бойцов С.А., Ипатов П.В., Калинина А.М., Вергазова Э.К., Ткачева О.Н., Гамбарян М.Г., Еганян Р.А., Медик Я.В., Бабаева С.В., Соловьева С.Б., Старинский В.В., Биличенко Т.Н., Федоров Е.Д – М.: ФГБУ "ГНИЦПМ" Минздрава России – 2015. – 111 с.
2. Щербук, Ю.А. Автоматизация процесса диспансеризации (на примере Санкт-Петербурга) / Щербук, Ю.А., Кадыров Ф.Н., Симаходский А.С., Добрых Д.В. // Врач и информационные технологии – 2006. – №4.
3. Visual Studio 2010 для профессионалов / Н. Рендольф, Д. Гарднер, М. Минутилло, К. Андерсон. – М.: Диалектика, 2011. – 1184 с.
4. Скит, Дж. С# для профессионалов: тонкости программирования, 3-е издание, новый перевод / Дж. Скит. – М.: Вильямс, 2014. – 608 с.
5. Нейгел, К. С# 5.0 и платформа .NET 4.5 для профессионалов / К. Нейгел. – М.: Диалектика, 2013. – 1440 с.
6. Фримен, Э. Изучаем HTML, XHTML и CSS / Э. Фримен, Э. Фримен. – 1-е изд. – М.: Питер, 2010. – 656 с.
7. Шмитт, К. CSS. Рецепты программирования / К. Шмитт. – СПб.: БХВ-Петербург, 2007. – 592 с
8. Рейсиг, Д. Инструменты отладки и тестирования. JavaScript. Профессиональные приёмы программирования / Д. Рейсиг; пер. с англ. Н. Вильчинский. – СПб.: Питер, 2008. – 76 с.
9. Шандров, Б.В. Технические средства автоматизации. / Б.В. Шандров, А.Д. Чудаков. – М.: Академия, 2007. – 368 с.

10. ТрастМед – медицинские информационные системы, запись на прием к врачу, электронная медицинская карта, учет медицинских услуг, лекарственное обеспечение [Электронный ресурс]. – Режим доступа: <http://www.trustmed.ru/themes/trustmed2/content.asp?folder=1916>

11. Компания SofTrust [Электронный ресурс]. – Режим доступа: <http://softrust.ru>

12. САДиП – Система Автоматизации Диспансеризации и Профилактических осмотров [Электронный ресурс]. – Режим доступа: <http://sadip.ru/Doctors/>

13. КМИС (Комплексные медицинские информационные системы) [Электронный ресурс]. – Режим доступа: <http://www.kmis.ru/site.nsf/pages/about.htm>

14. КМИС – Подсистема диспансеризации и профилактических осмотров [Электронный ресурс]. – Режим доступа: http://www.kmis.ru/site.nsf/pages/pol_disp.htm