

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ПОДСИСТЕМЫ УЧЕТА
ПАЦИЕНТОВ В МЕДИЦИНСКОМ УЧРЕЖДЕНИИ**

Выпускная квалификационная работа
обучающегося по направлению подготовки 09.03.03 «Прикладная
информатика»
очной формы обучения, группы 07001304
Шохиной Кристины Сергеевны

Научный руководитель
ст. пр. Болгова Е.В.

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 АНАЛИТИЧЕСКАЯ ЧАСТЬ	7
1.1 Техничко-экономическая характеристика предметной области.....	7
1.1.1 Характеристика медицинской организации	7
1.1.2 Краткая характеристика подразделений и их деятельности.....	9
1.1.3 Структурно-функциональный анализ деятельности организации «КАК ЕСТЬ»	11
1.2 Постановка задачи.....	14
1.2.1 Цель и назначение автоматизированного варианта решения задачи	14
1.2.2 Общая характеристика организации решения задачи на ЭВМ.....	16
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи	18
1.4 Анализ существующих разработок и обоснование выбора технологии проектирования	21
1.5 Экономическая сущность задачи	32
2 ПРОЕКТНАЯ ЧАСТЬ	34
2.1 Обоснование проектных решений.....	34
2.1.1 Обоснование проектных решений по техническому обеспечению	34
2.1.2 Обоснование проектных решений по информационному обеспечению	38
2.1.3 Обоснование проектных решений по программному обеспечению	42
2.2 Структурно-функциональный анализ деятельности организации «КАК ДОЛЖНО БЫТЬ»	44
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ	
3.1 Разработка базы данных автоматизированной подсистемы.....	48
3.2 Программное обеспечение задачи разработки подсистемы	51
3.3 Проверка работоспособности разработанного приложения.....	57

3.4	Оценка экономической эффективности.....	64
	ЗАКЛЮЧЕНИЕ	70
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	72
	ПРИЛОЖЕНИЕ А	75
	ПРИЛОЖЕНИЕ Б.....	84
	ПРИЛОЖЕНИЕ В	86
	ПРИЛОЖЕНИЕ Г.....	89
	ПРИЛОЖЕНИЕ Д.....	108

ВВЕДЕНИЕ

На сегодняшний день современные организации, предоставляющие какие-либо услуги, уже невозможно представить без применения информационных технологий. Автоматизация процесса управления и контроля стала закономерным этапом развития таких структур. Создание информационных подсистем ориентировано, прежде всего, на повышение эффективности работы организации [3].

Одними из наиболее перспективных направлений деятельности какого-либо предприятия являются различные виды учета. От оперативности работы системы учета и возможности проведения точного анализа имеющихся данных зависит функционирование организации в целом. Именно поэтому тема автоматизации учета пациентов в медицинском центре «ОРМЕД», является очень актуальной.

Рассматривая информационные системы с точки зрения их использования в реализации медицинских заявок, можно отметить, что их роль также важна, как и в других крупных областях. Любое медицинское учреждение, которое ведёт активную рабочую деятельность, имеет потребность в автоматизации и упрощении протекающих информационных процессов.

В настоящее время одной из наиболее востребованных отраслей информационных технологий в структуре медицинских учреждений является автоматизация процесса управления системой приёма клиентов.

В данной выпускной квалификационной работе будут рассмотрены основные вопросы автоматизации процесса оформления заявок в медицинском центре «ОРМЕД».

Объектом исследования является медицинский центр «ОРМЕД», а предметом – процесс обслуживания пациентов при оформлении заявок.

Целью выпускной квалификационной работы является повышение эффективности деятельности работы медицинского центра. Увеличение производительности осуществляется за счет поэтапной работы разрабатываемой подсистемы, с помощью которой у сотрудников поликлиники появится возможность оперативно и точно просмотреть все необходимые им данные о медицинском обследовании, тем самым избежав возможных ошибок, которые было бы допущено при ручном выполнении данных операций.

Для достижения поставленной цели необходимо решить ряд задач:

- произвести анализ деятельности выбранной медицинской организации и выявить все существующие недостатки;
- спроектировать функциональные модели деятельности регистратуры «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- сформировать цели и назначение будущей автоматизированной подсистемы;
- провести анализ имеющихся разработок применительно к медицинским организациям и выявить их недостатки;
- спроектировать и реализовать прикладное решение;
- провести тестирование разработанного программного продукта;
- оценить показатели эффективности разработки.

Таким образом, выпускная квалификационная работа состоит из введения и основной части, которая в свою очередь состоит из трех взаимосвязанных разделов, заключения, списка используемых источников и приложений.

Во введении определяются объект и предмет исследования, указываются цель, задачи и методы исследования для достижения поставленной цели, а также актуальность выбранной темы.

Первый раздел содержит информацию о проведении анализа предметной области, а именно основных характеристик предприятия и их деятельности. Кроме того, производится анализ уже существующих

разработок, выявление их преимуществ и недостатков. Этот раздел также содержит подробное описание деятельности медицинского учреждения на основе функциональных моделей.

Второй раздел выпускной квалификационной работы включает подробное обоснование проектных решений, на основании которых производится выбор системы управления базами данных и среды разработки автоматизированной подсистемы.

В третьем разделе описываются этапы реализации автоматизированной подсистемы для работы с медицинскими заявками: характеристика моделей баз данных, интерфейс для работы с клиентом, разработка программных функций, тестирование разработанной автоматизированной системы, а также определение целесообразности разработки с экономической точки зрения.

В заключении сделаны выводы о результате выполнения поставленных целей и о проделанной работе в целом.

Выпускная квалификационная работа написана на 74 страницах, содержит 48 рисунков и 17 таблиц, 35 источников литературы и 5 приложений.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Техничко-экономическая характеристика предметной области

1.1.1 Характеристика медицинской организации

Медицинские организации – это организации, которые осуществляют разностороннюю деятельность в области здравоохранения или оказания медицинских услуг, при этом поддерживая развитие медицины как науки, занимающейся поддержанием здоровья и оказания помощи населению [20].

Медицинский центр «ОРМЕД» является крупным многопрофильным, специализированным лечебно-профилактическим учреждением, предназначенным оказывать медицинскую помощь и осуществлять комплекс профилактических мероприятий по оздоровлению населения и предупреждению заболеваний [21].

К функциям, которые выполняет данный медицинский центр, относятся:

- 1) оказание первой медицинской помощи при острых и внезапных заболеваниях, травмах;
- 2) проведение ультразвуковых исследований;
- 3) профилактика и лечение разного рода заболеваний;
- 4) качественное проведение амбулаторных исследований для выявления вируса;
- 5) экспертиза временной нетрудоспособности;
- 6) освобождение больных от работы;
- 7) выписка направления больных на санаторно-курортное лечение.

Работа центра «ОРМЕД» организована таким образом, чтобы врачи могли обеспечить комфортное и безопасное лечение своих пациентов

с соблюдением самых высоких современных стандартов. Повсеместно используются новейшие разработки для защиты пациентов и персонала по программам: анти - СПИД и анти – гепатит [21].

В медицинском центра функционирует терапевтическое отделение, в котором регулярно производится работа над повышением качества услуг, осваиваются новые технологии лечения, реализуются новые методы и подходы к работе с пациентами.

В задачи медицинского центра «ОРМЕД » входят:

- 1) ведение отчетной медицинской документации, утвержденной Министерством здравоохранение;
- 2) систематическое повышение квалификации врачей;
- 3) как можно более обширный охват населения;
- 4) использование современных технологий в лечении пациентов;
- 5) проведение профилактической и лечебно-диагностической работы с пациентами;
- б) предоставление информации по профилактике различного рода заболеваний.

Все врачи, которые работают в отделениях, предоставляющих платные услуги, имеют первую или высшую квалификационную категорию, и периодически повышают свой профессиональный уровень.

Абсолютно все врачи центра имеют работать с наиболее современными методами диагностики и лечения различного рода заболеваний у пациентов. Доктора проводят профилактику, диагностику и различные формы лечение заболеваний. Проводят диагностику пациентов и оказывают, в случае необходимости, неотложную медицинскую помощь в полном объеме в амбулаторных условиях [21].

Врачебный прием в центре ведется по специальностям:

- 1) общая терапия;
- 2) неврология;
- 3) лечебный и профилактический массаж;

- 4) ультразвуковая диагностика;
- 5) амбулаторное исследование.

Известно, что повышение эффективности и качества предоставления медицинских услуг во многом обусловлено состоянием материально-технической базы лечебно-профилактических учреждений. Своевременное развитие медицинских учреждений рассматривается, как один из наиболее важных факторов для формирования высокого уровня медицинского центра в целом [20].

Период развития характеризуется новыми услугами и технологиями, а также усовершенствование уже имеющихся процессов лечения пациентов, которые направлены на повышение уровня медицинской помощи и дальнейшие перспективы развития лечебно-профилактического центра. Любое развитие направлено на обеспечение полного удовлетворения потребностей населения в современных процессах получения медицинской помощи и улучшения ее качества в целом [25].

1.1.2 Краткая характеристика предприятий и их деятельности

В состав медицинского центра «ОРМЕД» входит ряд управленческих и лечебных подразделений, тесно взаимосвязанных между собой, которые непрерывно взаимодействуют друг с другом для качественного и своевременного оказания медицинской помощи пациентам [21].

Правильное распределение обязанностей между подсистемами напрямую влияет на функционирование организации в целом, то есть каждый отдел должен отвечать за определенный спектр обязанностей.

На рисунке 1.1 представлена структура медицинского центра «ОРМЕД».



Рисунок 1.1 – Структура медицинского центра «ОРМЕД»

На схеме представлена вся структура центра, включающая отдел кадров, бухгалтерию, организационно-методический отдел и другие. Отдел кадров отвечает за прием всего штатного персонала на работу в конкретной должности. Бухгалтерия подразделяется на два отдела: кассовый отдел и отдел бухгалтерской отчетности. Кассовый отдел занимается работой с денежными средствами, приемом платежей за услуги медицинского центра, а также закупку расходных материалов. Отдел бухгалтерской отчетности в свою очередь отвечает за сведение всех финансовых документов в единый бухгалтерский отчет.

Организационно – методический отдел отвечает за проведение инструктажа, своевременное повышение квалификации работников медицинского центра, закупка нового технологического оборудование и другие операции, связанные с обеспечением активной работы лечебно – профилактического учреждения.

Отдел медицины подразделяется на три основных отдела:

1) отдел лабораторных исследований, который занимается выявлением заболеваний посредством проведения амбулаторных обследований;

2) отдел врачебного приема, включает в себя прием и консультации врачей разного рода специализаций;

3) отдел обследования занимается проведением технических обследований, связанных с использованием конкретного оборудования.

Остановимся подробнее на том отделе, который непосредственно относится к подсистеме учета пациентов в медицинском центре – отдел регистрации пациентов, который выделен на рисунке пунктирной линией. Данный отдел отвечает за оформления новых пациентов в базе данных медицинского центра «ОРМЕД», первичную консультацию пациентов и формирование заявок на конкретные услуги.

1.1.3 Функциональный анализ деятельности организации «КАК ЕСТЬ»

Перед началом проектирования системы разработчик должен ясно представлять, какие функциональные возможности будут заложены в систему, и как будет организована работа внутри системы. Разработка функциональных моделей деятельности медицинского центра помогла определить объекты, автоматизацию которых нужно выполнить. Построение происходило путём разбиения контекстной диаграммы, отображающей деятельность организации в целом, на функциональные подсистемы. Данные системы в свою очередь делились на подфункции и конкретные процедуры.

В ходе анализа была разработана функциональная диаграмма «КАК ЕСТЬ» на основе методологии IDEF0 с помощью CASE-средства AllFusion Process Modeler 7. Стандарт IDEF0 предназначен для разработки функциональных моделей, отображающих структуру и функции системы, а также связывающие их потоки [13].

На рисунке 1.2 изображена контекстная диаграмма деятельности медицинского центра «ОРМЕД». Входными данными диаграммы являются информация о врачах и пациентах, сведения о предоставляемых услугах и список участков медицинского центра. Управление осуществляется на основе устава медицинского учреждения и нормативных документов. Механизмами представлены сотрудники, работающие в регистратуре. Выходной поток данных – это заявки на прием.

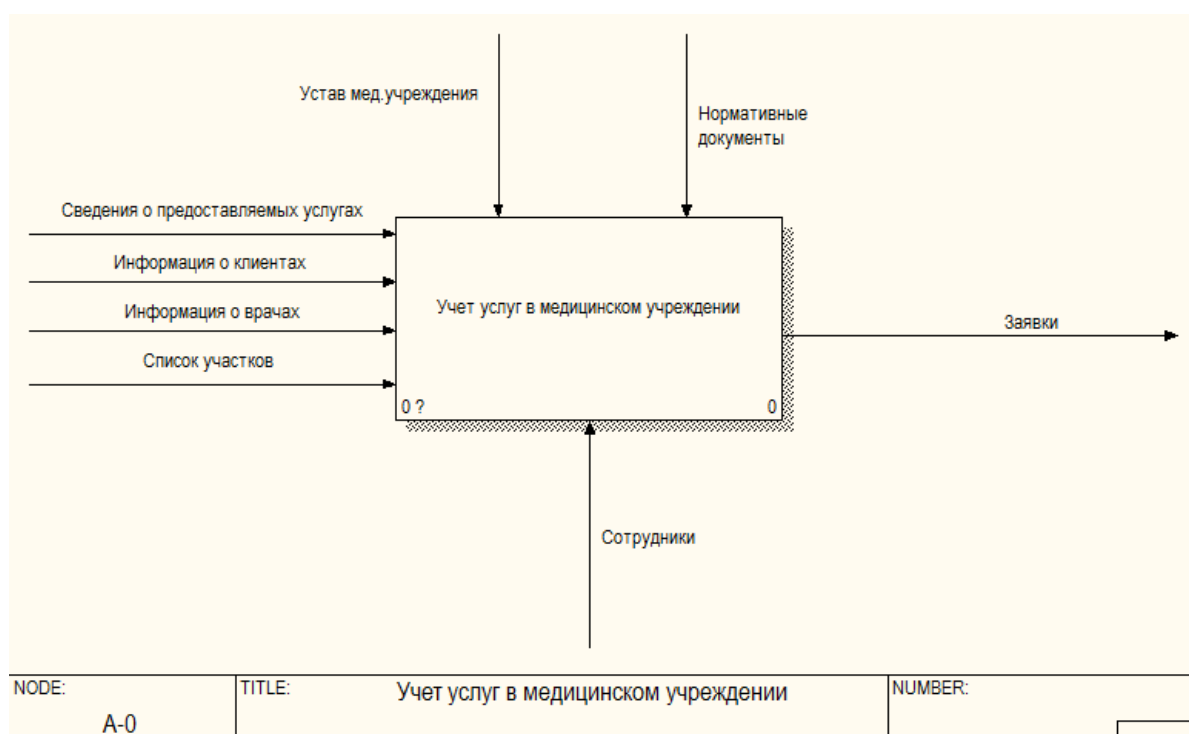


Рисунок 1.2 – Контекстная диаграмма деятельности медицинского центра «КАК ЕСТЬ»

Контекстная диаграмма деятельности детализирована четырьмя функциональными блоками, отражающими основные действия при начальном обслуживании пациентов:

1) ввод данных о пациенте предусматривает предоставление клиентом поликлиники паспорта и страхового полиса, по которому будет происходить регистрация;

- 2) выбор консультации включает в себя список всех имеющихся услуг, которые предоставляет медицинское учреждение;
- 3) подбор врача предоставляет пациентам возможность выбора доктора, который работает по выбранному виду услуги;
- 4) формирование заявки на прием – письменное упорядочивание всех указанных параметров в заявку (см. рисунок 1.3).

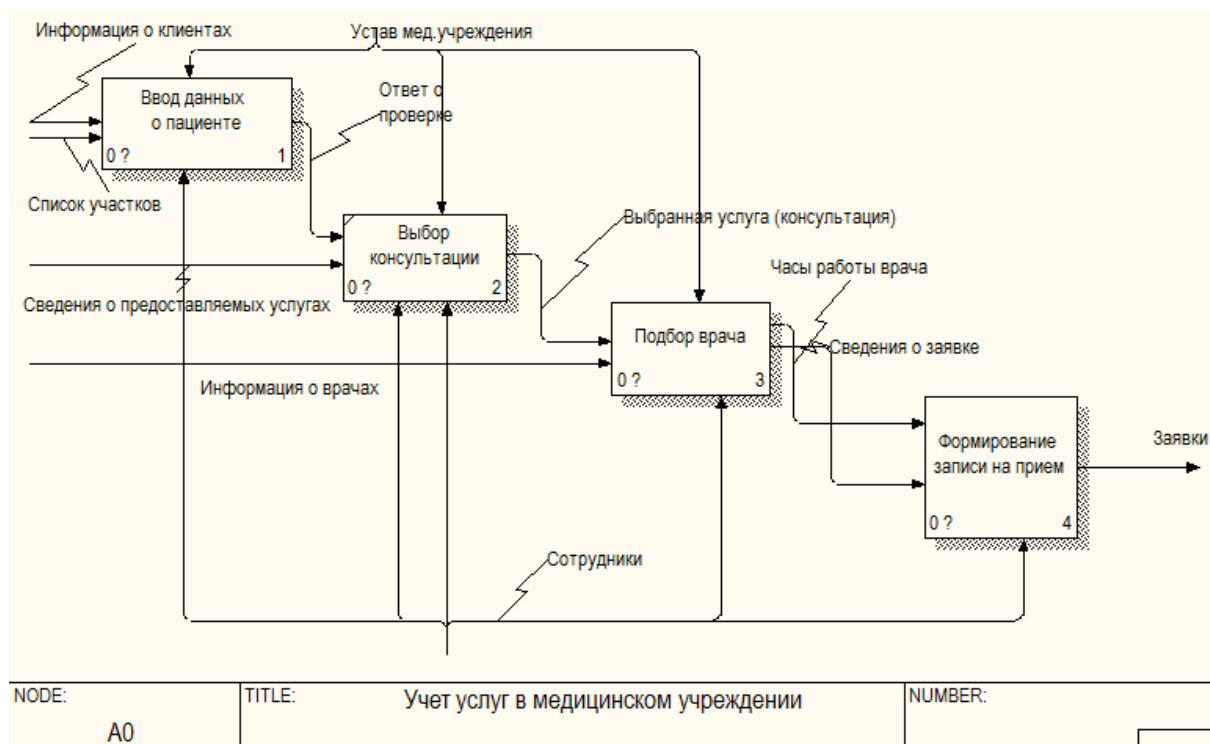


Рисунок 1.3 – Декомпозиция контекстной диаграммы деятельности медицинского центра «КАК ЕСТЬ»

Таким образом, прежде всего, сотрудник регистратуры должен ввести данные о пациенте и указать консультацию, которую планирует посетить пациент. Следующим шагом происходит подбор врача, который работает с данным видом услуг, затем уточняет дату и время и формирует письменную заявку.

Каждый функциональный блок диаграммы-декомпозиции деятельности медицинского центра на текущий момент выполняется вручную сотрудником регистратуры и заполняется в письменный отчет.

Разработанная модель говорит о том, что необходимо оптимизировать ручное оформление заявки на прием пациента. Для этого было принято решение разработать подсистему для автоматизации процесса учета пациентов. Кроме того, в подавляющем большинстве случаев, это позволит существенно облегчить работу с ведением отчетности и формированием анализа работы медицинского центра.

1.2 Постановка задачи

1.2.1 Цель и назначение автоматизированного варианта решения задачи

Целью выполнения данной выпускной квалификационной работы является разработка подсистемы учета пациентов в медицинском центре «ОРМЕД», создание учета всех поступающих заявок, их оплата и прохождение медицинского обследования.

Перед тем, как к разрабатываемой подсистеме автоматизации выставить требования, необходимо проанализировать схему процесса работы медицинского центра по оказанию работы, связанной с учетом пациентов.

В результате анализа предметной области и функционального проектирования организации на текущий момент, были выделены ряд требований и задач:

- 1) ведение данных о докторах, их отпусках и графиках работы, чтобы отображать эти данные в подсистеме и сотрудники регистратуры не допускали возможных ошибок при оформлении заявки пациента;
- 2) хранение данных об имеющихся пациентах и возможность добавления новых;

- 3) ведение информации о ценах на предоставляемые медицинскими центром услуги;
- 4) возможность формирования заявок на прием пациентов с выбором врача;
- 5) регистрация факта оплаты услуги через кассового работника;
- 6) формирование больничных листов через кабинет врача;
- 7) возможность проведение анализа динамики посещения медицинского центра, как в целом, так и по отдельно-выбранным параметрам;
- 8) возможность разграничить права для пользователей подсистемы, тем самым каждый сотрудник будет обладать только теми правами, которыми его наделит руководитель;
- 9) формирование различного рода отчетности.

Цель частичной автоматизации медицинского центра «ОРМЕД» заключается в упрощении работы с данными пациентов, повышение эффективности работы отдела регистратуры тем самым переход на новый уровень обслуживания пациентов.

В автоматизированной подсистеме должны быть реализованы следующие функции:

- 1) максимальная автоматизация оформления пациентов и их заявок на прием в медицинском центре;
- 2) хранение всех данных о пациентах и сотрудниках организации;
- 3) формирование отчетных данных по указанным параметрам за конкретный период;
- 4) обеспечение целостности хранимой информации в базе данных.

За счет выполнения перечисленных функций медицинский центр достигает сокращения трудоемкости и стоимостных затрат по формированию планов организации. Появление новых аналитических показателей в виде отчетов по указанным статистическим данным.

Также автоматизация работы отдела планирования сводится к повышению оперативности построения отчетов, структуризации информации по планированию в рамках БД за предыдущие периоды, исключения задержки необходимой информации, вследствие несвоевременности предоставления плановой информации. Кроме того, у медицинского центра появляется перспектива повышения активности в отдельных сферах, связанных с областью планирования. У уполномоченных сотрудников, занимающихся анализом деятельности предприятия, имеется возможность получения достоверных результатов, исключая несоответствия данных, используемых в различных отчетах, исключая дублирования потоков информации, при этом снижая трудоемкость обработки информации. Автоматизированная информационная подсистема позволяет усовершенствовать процесс сбора, передачи, обработки, хранения, защиты целостности и секретности информации и процессов выдачи результатов аналитики конечному пользователю.

1.2.2 Общая характеристика организации решения задачи на ЭВМ

Ряд задач, который поставлен перед выпускной квалификационной работой, будет реализован в виде автоматизированной подсистемы учета пациентов. На данном рабочем месте задача учета пациентов и их заявок на получение медицинских консультаций реализуется за счет заполнения данных в простую печатную форму.

Так как данный комплекс задач, относится к классу задач, для которых четко установлена последовательность выполнения действий по обработке данных, выбран диалоговый режим обработки данных, это связано

с тем, что данный метод является наиболее эффективным в выбранной работе.

В основу диалогового режима, необходимого для обработки данных заложена система, основанная на динамическом взаимодействии человека с электронно-вычислительной машиной, посредством введения необходимых параметров в подсистему. Обмен данными между человеком и ЭВМ, во время которого осуществляется решение конкретной задачи, называется технологией обработки информации в режиме диалога. Необходимость применения такой технологии обусловлена тем, что ее применение приводит к наилучшему сочетанию взаимодействия человека и ЭВМ в процессе обработки и решения какой-либо установленной задачи. При разработке с такой технологией обеспечивается:

- 1) обеспечивается быстрый и удобный поиск необходимых для пользователя данных;
- 2) быстрая обработка принятых сообщений от пользователя и немедленная передача результата в виде обработанных сообщений;
- 3) способность пользователя оказывать полноценное влияние на выполнение технологических операций по обработке данных;
- 4) возможность перебора произвольных комбинаций для поисковых запросов;
- 5) быстрый поиск информации;
- 6) улучшение базовых характеристик вводимых данных;
- 7) удобный доступ к редко используемой информации;
- 8) оперативная обработка и анализ информации на выходе[31].

Решение комплексных задач будет реализовано с помощью диалогового режима, таким образом при решении конкретной задачи будет происходить обмен сообщениями между пользователем и автоматизированной подсистемой.

Пользователь и подсистема выполняют роль активного элемента попеременно. Информационная система активна от момента завершения

ввода информации и команд заданных пользователем до завершения обработки последнего запроса. Далее пользователь производит анализ результатов обработки запроса и может осуществлять ввод данные для следующего запроса.

Пользовательский интерфейс, который используется в диалоговом режиме представляется в следующем виде: в формате меню, в формате команд или графически. При выборе из меню пользователю предоставляются на экране набор доступных для него операций.

Принцип заключается в однозначном выборе пользователя нужной для него команды. Если после задания команды какая-либо информация введена некорректно, то есть, например, текущий пользователь не обладает правами на работу с выбранной им информацией, то система потребует уточнения.

Системы, основанные на работе в графическом режиме, могут не содержать стандартных команд для работы с подсистемой. Управление процессом обработки обеспечивается движением курсора по экрану с помощью либо функциональных клавиш, либо курсора мыши. В процессе работы с данными пользователь переходит на объекты и выбирает конкретные параметры, нажимая клавиши.

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

Вычислительная техника является устройством, выполняющим последовательность ранее заданных операций с высокой скоростью. При этом оборудование получает исходные данные, инструкции, которые необходимы при их обработке. После чего происходит обработка данных соответственно инструкциям и получение результата в той форме, которую

выбрал разработчик. Данная информация затем используется человеком для решения каких-либо задач или для работы с другими данными[31].

Внедрение компьютеров в работу вместо человека имеет ряд значительных преимуществ. Рассмотрим наиболее очевидные из них:

- скорость обработки данных является одним из самых главных преимуществ, скорость работы с данными современными компьютерами очень высока, и в настоящее время достигает миллиарды операций в секунду;
- большой объем памяти, который позволяет хранить большое количество информации с целью дальнейшей работы с ней;
- компьютер позволяет решать сверхсложные задачи, с которыми не в силу справится человеку;
- компьютеры не поддаются эмоциям при обработке и анализе введенных данных, все факторы учитываются максимально объективно, и на их основе строятся реалистичная модель предметной области [31].

На основании вышеизложенных характеристик вычислительной техники, можно сделать вывод, что наиболее эффективным будет являться использование компьютеров при решении задач, которые требуют быстрого решения, а также для автоматизации повторяющейся рутинной работы. К сожалению, операции требующие изобретательности или каких-либо других качеств, пока еще невозможно поручить вычислительной технике. Тем не менее, область искусственного интеллекта постоянно развивается и постепенно начинает внедряться во все области человеческой деятельности.

Развитие электронно-вычислительной техники, ее внедрение в сферы деятельности различных организаций осуществляется все чаще в современных условиях работы предприятий. Выполнение медицинских центром всех функций работы невозможно без использования автоматизированных средств основных расчетов.

Также необходимо учесть, что на сегодняшний день в медицинском центре «ОРМЕД» основная работа является ручной, что имеет множество недостатков. К этим недостаткам можно отнести следующее:

- низкая оперативность работы;
- большое количество несовершенств при организации сбора и регистрации информации;
- сложность ведения учета и анализа динамики предприятия;
- недостоверность полученных отчетов из-за возможных ошибок;
- низкая производительность труда;
- необходимость задействования большого числа сотрудников, при выполнении более сложных задач.

Кроме этого, постоянно возрастающее количество информации, которое необходимо обрабатывать, для ведения корректных отчетов о деятельности организации, необходимость проведения результатов динамики работы медицинского центра на ближайшие перспективы указывают на необходимость внедрения электронно-вычислительной техники. Так как в бумажном варианте вести подобные отчеты достаточно сложно, а на проведение какого-либо анализа может уйти очень большое количество времени. При этом анализ можно провести, например, по динамике работы организации в целом, но не по отдельным его сегментам. Именно поэтому использование ЭВМ в данном случае значительно упростит работы всего медицинского учреждения.

С внедрением автоматизированной подсистемы, которая будет отвечать за выполнение указанных выше функций и устранение имеющихся недостатков. Основными преимуществами после введения ЭВМ являются следующие:

- 1) возможность проведения более разносторонних форм динамического учета;
- 2) практически полная независимость сотрудников от формирования отчета за любой период;
- 3) разделение функциональных обязанностей между сотрудниками;
- 4) оперативная обработка имеющихся данных;
- 5) сокращение возникающих ошибок при работе с данными.

1.4 Анализ существующих разработок и обоснование выбора технологии проектирования

На данный момент в компании практически не существует никакой автоматизированной системы, единственное, что применяется из программных средств это входящие в состав офисного пакета Microsoft Office – Microsoft Word и Microsoft Excel. Данные программные продукты используются для хранения всех данных, например, таких как сведения о пациентах и их заявках.

В работе бухгалтерии организация использует систему 1С:Бухгалтерия, но аспект автоматизации бухгалтерского учета не рассматривается в задаче, которую решает данная выпускная квалификационная работа.

В настоящее время бурно развиваются различные классы программных продуктов, например, ERP (корпоративные информационные системы), BPM (системы управления процессами), CRM (системы управления взаимоотношениями с клиентами), HD (системы управления технической поддержкой сотрудников), HRM (системы управления персоналом) и другие. Каждые из них можно использовать в работе медицинского центра, автоматизируя при этом какой-либо организационный отдел.

В каждой организации используются свои правила по выбору информационной системы, поэтому только после анализа особенностей медицинского учреждения можно сформулировать требования к информационной системе. К основным требованиям медицинских информационных систем можно отнести:

- наличие функциональной возможности для организации полноценной работы предприятия, т.е. возможность хранения, обработки,

учета определенных параметров и формирование различных форм отчетности;

- организация доступа сотрудников к документации и поддержка её параллельных изменений;

- возможность ограничения доступа к информации сотрудников с разными должностными обязанностями [19].

Не секрет, что уже существуют и успешно применяются данного рода системы.

Проанализировав работу медицинских учреждений, с точки зрения протекающих в них процессов и принципов работы сотрудников, выбраны следующие информационные системы, подходящие для автоматизации работы в данной сфере. К выбранным медицинским информационным системам относятся:

- 1) РИСТАР - набор программ, интернет-приложений и сервисов, работающих с общей базой данных и предназначенных для комплексной автоматизации медицинских учреждений [17];

- 2) MedSoft (МедСофт - Клиника) – медицинская информационная система от компании Medexis. МИС МедСофт дает возможность создать гибкую структуру документов, настраивать шаблоны для работы с клиентами [18];

- 3) М-АПТЕКА ЛПУ – медицинская информационная система от компании ЭСКЕЙП. Модуль «М-АПТЕКА ЛПУ» позволяет формировать и обеспечивать потребности прикрепленного населения льготных категорий в лекарственных препаратах и медицинских изделиях, автоматизирует выписку рецептов по всем программам льготного лекарственного обеспечения, создавать и вести региональные регистры льготников [1];

- 4) ТрантМед:Здоровье – медицинская информационная система предназначена для комплексной автоматизации деятельности персонала лечебно-профилактических учреждений, включающих амбулаторно-поликлинические подразделения [29];

5) МЕДИАЛОГ - медицинская информационная система, все модули которой системы работают в единой среде данных, при этом каждый модуль отвечает за определённую функциональность и может быть приобретен отдельно, таким образом, система будет масштабироваться вместе с ростом клиники или расширением задач по её автоматизации [15];

6) MedWork- медицинская информационная система, в которой ведется персонализированный учет оказываемых пациентам услуг (включая учет накладных расходов на услуги), с автоматическим формированием счетов на оплату для контрагентов учреждения и выгрузкой данных для 1С [34];

7) 1С:Медицина. Поликлиника - медицинская информационная система предназначена для автоматизации основных процессов медицинских организаций различных организационно-правовых форм, оказывающих медицинскую помощь в амбулаторно-поликлинических условиях [35];

8) Clinic365 - удобная медицинская информационная система, которая позволяет вести базу пациентов. Clinic365 можно разместить как в облаке, так и на сервере клиники [32];

9) Инфоклиника - полнофункциональная медицинская информационная система: управление поликлиникой, больницей, медицинским центром и сетью медицинских учреждений [10];

10) AKSi-клиника - программа для частных клиник, от компании АКСИМЕД. МИС AKSi-клиника — это типовая платформа для автоматизации лечебно-профилактических учреждений, тиражируемая и внедряемая в промышленных масштабах [16].

На основе анализа деятельности медицинских организаций были выделены отдельные признаки, разносторонне характеризующие информационные системы.

В первом классе признаков выделяются признаки позитивного характера, то есть те признаки медицинских информационных систем,

которые положительным образом влияют на эффективность его использования. К таким признакам относятся следующие:

- функциональность использования – данная система удобна для её использования в медицинской сфере;
- интеграция – возможность интеграции с другими системами;
- формы отчетности – наличие выбора различных форм отчетности;
- гибкость системы – способность системы адаптироваться к нововведениям;
- интерфейс - является понятным для пользователя, как работать в данной системе;
- разграничение прав – возможность разграничить права между сотрудниками;
- автоматические обновления - наличие в МИС автоматических обновлений;

Второй класс признаков качества медицинской информационной системы объединяет совокупность признаков негативного характера, отрицательно влияющих на использование данного ресурса. Среди признаков данного класса можно выделить следующие:

- 1) система защиты – возможность защиты информации, которая содержится в системе;
- 2) стоимость - использование данной системы в выбранной предметной области является бесплатным или за него необходимо платить.

Для выделенных признаков, определяющих качество информационных подсистем, устанавливается градационная шкала, каждый уровень которой будет отражать степень влияния конкретного признака на общую эффективность использования. Шкала градаций устанавливается таким образом, чтобы потребитель мог экспертным путем оценить степень влияния признака, выбирая соответствующую градацию.

К выше перечисленным признакам можно предложить следующие шкалы градаций:

Признак «функциональность использования» можно разделить на следующие градации:

- 1) «неудобна» - ИС неудобна для использования в данной предметной области;
- 2) «частично удобна» - система имеет как плюсы, так и минусы в своём использовании;
- 3) «удобна» - данная ИС удобна для работы в данной области.

Признак «интеграция» можно разделить на следующие градации:

- 1) «система неинтегрируемая» - в системе присутствует один язык;
- 2) «низкая степень интеграции» - система имеет наличие двух языков;
- 3) «высокая степень интеграции» - система написана на большом количестве языков;

Признак «формы отчетности» можно разделить на следующие градации:

- 1) «не имеет отчетности» - подсистема не позволяет формировать отчеты;
- 2) «наличие 1-2 форм отчетности» - возможность формирование нескольких отчетов;
- 3) «наличие более 2 форм отчетности» - подсистема предоставляет широкий выбор форм отчетности.

Признак «гибкость системы» можно разделить на следующие градации:

- 1) «не адаптируется» - ИС не адаптируется к новым условиям;
- 2) «частично адаптируется» - система имеет возможность адаптироваться только к некоторым изменяемым условиям;
- 3) «адаптируется» - данная ИС приспособлена к адаптации.

Признак «интерфейс» можно разделить на следующие градации:

- 1) «непонятен» - интерфейс системы непонятен.
- 2) «частично понятен» - ИС не всегда понятна в работе с ней.
- 3) «полностью понятен» - данная ИС понятна в работе.

Признак «разграничение прав» можно разделить на следующие градации:

- 1) «отсутствие разграничений» - нет возможность разграничить права между пользователями;
- 2) «наличие разграничений» - имеется возможность на разграничение доступа.

Признак «автоматические обновления» можно разделить на следующие градации:

- 1) «обновления отсутствуют» - отсутствие обновлений в системе
- 2) «обновления выполняются не всегда» - выполняются только некоторые обновления
- 3) «наличие всех автоматических обновлений» - имеются автоматические обновления в данной ИС.

Негативные признаки:

Признак «защита системы» можно разделить на следующие градации:

- 1) «имеется защита» - существует защита системы;
- 2) «защита отсутствует» - никакой защиты система не имеет.

Признак «стоимость» можно разделить на следующие градации:

- 1) «бесплатная» - система предоставляет свои услуги бесплатно;
- 2) «средняя стоимость» - система предоставляет свои услуги по средней цене;
- 3) «высокая стоимость» - система предоставляет свои услуги по достаточно высокой цене.

На следующем этапе определяются количественные значения каждого из уровней шкалы, градационное разбиение отдельных признаков качества называют коэффициентом весомости каждого признака.

Для каждого позитивного признака были установлены следующие коэффициенты весомости:

- функциональность использования (P_{ϕ}) – $k_{\phi}=5$;
- интеграция ($P_{и}$) – $k_{и}=4$;
- формы отчетности ($P_{\phiо}$) – $k_{\phiо}=2$;
- гибкость системы ($P_{г}$) – $k_{г}=3$;
- интерфейс ($P_{ис}$) – $k_{ис}=1$;
- разграничение прав ($P_{р}$) – $k_{р}=3$;
- автоматические обновления ($P_{а}$) – $k_{а}=2$.

Для негативных признаков устанавливаются следующие коэффициенты весомости:

- защита системы ($P_{з}$) – $k_{з}=3$;
- стоимость ($P_{с}$) – $k_{с}=4$;

Самая наивысшая оценка получилась у признака функционального использования системы. Таким образом, можно сделать вывод, что данный признак является главным, в связи с этим, в случае, если подсистема не будет давать возможность выполнять конкретные действия, то нет смысла использовать данную систему в работе медицинского центра. Самую наименьшую оценку получил признак интерфейса системы. Это связано с тем, что из всех указанных параметров, его значимость в работе с системой самая низкая.

Исходя из полученных данных, были рассмотрены коэффициенты нормирования для каждого класса признаков:

$$N+ = 100 / 20 = 5$$

$$N- = 100 / 7 = 14$$

По формулам: $r_i^+ = \text{entier} \{ k_i^+ * N^+ \}$, $r_j^- = \text{entier} \{ k_j^- * N^- \}$ определяются ранги для каждого признака.

Для первого класса признаков:

$$r_{\phi} = \text{entier} \{ k_{\phi} * N+ \} = \text{entier} \{ 5 * 5 \} = 25;$$

$$r_{и} = \text{entier} \{ k_{и} * N+ \} = \text{entier} \{ 4 * 5 \} = 20;$$

$$r_{\phi_0} = \text{entier} \{ k_{\phi_0} * N+ \} = \text{entier} \{ 2 * 5 \} = 10;$$

$$r_r = \text{entier} \{ k_r * N+ \} = \text{entier} \{ 3 * 5 \} = 15;$$

$$r_{ис} = \text{entier} \{ k_{ис} * N+ \} = \text{entier} \{ 1 * 5 \} = 5;$$

$$r_p = \text{entier} \{ k_p * N+ \} = \text{entier} \{ 3 * 5 \} = 15;$$

$$r_a = \text{entier} \{ k_a * N+ \} = \text{entier} \{ 2 * 5 \} = 10.$$

Для второго класса признаков:

$$r_3 = \text{entier} \{ k_3 * N- \} = \text{entier} \{ 3 * 14 \} = 42;$$

$$r_c = \text{entier} \{ k_c * N- \} = \text{entier} \{ 4 * 14 \} = 56.$$

По формулам: $h_i = \text{entier} \{ r_i^+ / (q_i^+ - 1) \}$, $h_j = \text{entier} \{ r_j^- / (q_j^- - 1) \}$

рассчитывается шаг градации для каждого признака.

Для первого класса признаков:

$$h_{\phi} = \text{entier} \{ r_{\phi} / (q_{\phi} - 1) \} = \text{entier} \{ 25 / (3 - 1) \} = 12,5;$$

$$h_{и} = \text{entier} \{ r_{и} / (q_{и} - 1) \} = \text{entier} \{ 20 / (3 - 1) \} = 10;$$

$$h_{\phi_0} = \text{entier} \{ r_{\phi_0} / (q_{\phi_0} - 1) \} = \text{entier} \{ 10 / (3 - 1) \} = 5;$$

$$h_r = \text{entier} \{ r_r / (q_r - 1) \} = \text{entier} \{ 15 / (3 - 1) \} = 7,5;$$

$$h_{ис} = \text{entier} \{ r_{ис} / (q_{ис} - 1) \} = \text{entier} \{ 5 / (3 - 1) \} = 2,5;$$

$$h_p = \text{entier} \{ r_p / (q_p - 1) \} = \text{entier} \{ 15 / (2 - 1) \} = 15;$$

$$h_a = \text{entier} \{ r_a / (q_a - 1) \} = \text{entier} \{ 10 / (3 - 1) \} = 5.$$

Для второго класса признаков:

$$h_3 = \text{entier} \{ r_3 / (q_3 - 1) \} = \text{entier} \{ 42 / (2 - 1) \} = 42;$$

$$h_c = \text{entier} \{ r_c / (q_c - 1) \} = \text{entier} \{ 56 / (3 - 1) \} = 28.$$

По формулам: $P_i^+(s = q_i^+) = r_i^+$;

$P_i^+(s) = (s - 1) * h_i$, при $s = 1, 2 \dots (q_i^+ - 1)$ рассчитываются значения

уровней градации для каждого признака.

Для первого класса признаков:

$$P_{\phi}(1) = 0, \quad P_{\phi}(2) = 12,5, \quad P_{\phi}(3) = 25;$$

$$P_{и}(1) = 0, \quad P_{и}(2) = 10, \quad P_{и}(3) = 20;$$

$$P_{\phi_0}(1) = 0, \quad P_{\phi_0}(2) = 5, \quad P_{\phi_0}(3) = 10;$$

$$P_r(1) = 0, \quad P_r(2) = 7,5, \quad P_r(3) = 15;$$

$$P_{ис}(1) = 0, \quad P_{ис}(2) = 2,5, \quad P_{ис}(3) = 5;$$

$$P_p(1) = 0, \quad P_p(2) = 15;$$

$$P_a(1) = 0, \quad P_a(2) = 5, \quad P_a(3) = 10;$$

Для второго класса признаков:

$$P_3(1) = 0, \quad P_3(2) = 42;$$

$$P_c(1) = 0, \quad P_c(2) = 28, \quad P_c(3) = 56;$$

В таблице 1.1 представлена оценка эффективности использования информационной системы «РИСТАР».

Таблица 1.1 – Экспертная оценка системы «РИСТАР»

Признак	Функциональность использования			Оценка, %
1	2			3
Градация	Неудобна	Частично удобна	Удобна	12,5
Значение	0	12,5	25	
Признак	Интеграция			Оценка, %
Градация	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	10
Значение	0	10	20	
Признак	Формы отчетности			Оценка, %
Градация	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
Значение	0	5	10	
Признак	Гибкость системы			Оценка, %
Градация	Не адаптируется	Частично адаптируется	Адаптируется	0
Значение	0	7,5	15	
Признак	Интерфейс			Оценка, %
Градация	непонятен	Частично понятен	Полностью понятен	5
Значение	0	2,5	5	
Признак	Разграничение прав			Оценка, %
Градация	Отсутствие разграничений	Наличие разграничений		15
Значение	0	15		
Признак	Автоматические обновления			Оценка, %
Градация	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	10
Значение	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				57,5

Продолжение таблицы 1.1

1	2			3
Признак	Защита системы			Оценка, %
Градация	Имеется защита	Защита отсутствует		0
Значение	0	42		
Признак	Стоимость			Оценка, %
Градация	Бесплатная	Средняя стоимость	Высокая стоимость	28
Значение	0	28	56	

Общая суммарная оценка негативных свойств $\sum P_j^-(s)$ в %	28
Общая суммарная оценка эффективности использования Информационного ресурса $\Theta = \sum P_i^+(s) - \sum P_j^-(s)$ в %	29,5

Таблицы оценки эффективности использования по всем выбранным системам представлены в приложении А.

В таблице 1.2 представлен результат проведения сравнительного анализа информационных систем.

Таблица 1.2 – Сравнительный анализ ИС

МИС Характеризующие признаки	РИСТАР	MedSoft	М-АПТЕКА ЛПУ АПТЕКА ЛПУ	ТрантМед:Здоровье	МЕДИАЛОГ	MedWork	1С:Медицина. Поликлиника	Сlinic365	Иифоклиника	AKSi-клиника
	Положительные признаки									
Функциональность использования	12,5	12,5	25	12,5	25	12,5	25	12,5	12,5	0
Интеграция	10	20	10	0	20	10	20	10	0	10

Продолжение таблицы 1.2

Формы отчетности	5	5	5	5	10	5	10	5	5	5
Гибкость системы	0	0	0	7,5	7,5	7,5	15	0	7,5	0
Интерфейс	5	5	5	5	2,5	5	5	2,5	5	2,5
Разграничение прав	15	15	0	15	15	15	15	15	15	0
Автоматические обновления	10	10	5	10	10	5	10	5	0	5
Отрицательные признаки										
Система защиты	0	0	0	0	0	0	0	42	42	0
Стоимость	28	28	28	28	56	28	56	28	0	28
Оценка										
Результат	29,5	39,5	22	27	34	32	44	-20	3	-5,5

По результатам сравнительного анализа медицинских информационных систем наибольшую оценку получили системы «MedSoft» и «1С: Медицина. Поликлиника», в связи с этим можно сделать вывод, что использование данных систем в работе медицинских учреждений значительно упростит работу как сотрудников, так и всей организации в целом.

Системы, которые получили наименьшую оценку, уступают другим медицинским информационным системам в комплексной работе, то есть их функциональные возможности менее развиты, чем у других систем.

Учитывая специфические особенности работы медицинского центра «ОРМЕД», принято решение разработать автоматизированную подсистему по учету пациентов и оформленных ими заявок на медицинские услуги.

Данное решение принято с целью устранения некоторых недостатков имеющихся систем и ориентировать разработанный продукт на работу конкретного медицинского учреждения. Это позволит, с одной стороны упростить работу сотрудников, используя в автоматизированной подсистеме только самые необходимые документы, справочники и отчеты, а с другой

стороны появляется возможность получения оперативной информации всеми участниками информационных отношений в сфере учета. Также при необходимости будет возможность доработки разрабатываемой подсистемы в ходе ее введения в работу организации.

1.5 Экономическая сущность задачи

Экономической сущностью задачи автоматизации работы отдела регистратуры в медицинском центра «ОРМЕД» является повышение результативности труда посредством автоматизации рутинной деятельности, а также деятельности, которую можно ускорить путем введение автоматизированной подсистемы.

Вследствие ускорения процесса делопроизводства возможно будет или повысить количество новых пациентов, тем самым увеличить количество предоставляемых услуг, а в конечном итоге и прибыль всей организации [25].

Или, если количество новых пациентов не будет расти, имеется возможность сокращения нескольких штатных сотрудников, которые в период до внедрения автоматизации, занимались анализом динамики деятельности предприятия, что в конечном итоге приведет к сокращению расходов на начисления заработной платы сотрудникам организации.

В случае если сокращение сотрудников не будет одним из вариантов по какой-либо причине, то возможно увеличение количества предоставляемых услуг, для привлечения новых пациентов, но данный вариант включает в себя дополнительные затраты на приобретения нового оборудования для новых видов услуг.

В конечном итоге цель данной автоматизации является сокращение издержек при работе медицинского центра, сокращение затрат и увеличение

прибыли. А также по возможности расширение сферы деятельности и ассортимента предоставляемых услуг медицинского центра «ОРМЕД».

В первом разделе выпускной квалификационной работы была проанализирована характеристика деятельности медицинского центра «ОРМЕД», на основе которой были построены функциональные диаграммы деятельности организации «КАК ЕСТЬ». Данные диаграмма позволили описать процесс функционирования отдела, занимающегося регистрацией клиентов и оформлением заявок на предоставление услуг. Также были выявлены недостатки в работе центра и проведен анализ имеющихся систем, которые используются в работе медицинских учреждений, позволяя упростить их функционирование и сократить количество выполняемых вручную операций. Выделены цели и задачи для выпускной квалификационной работы.

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Обоснование проектных решений

2.1.1 Обоснование проектных решений по техническому обеспечению

Для эффективного решения задачи, поставленной в ходе написания выпускной квалификационной работы, необходимо соответствующее техническое обеспечение.

Техническое обеспечение представляет собой совокупность технических средств, компьютерной техники, средств передачи информации, используемых в автоматизированных системах[17].

Необходимое техническое обеспечение для выполнения данной работы включает в себя ЭВМ (системный блок), монитор, клавиатуру, мышь, принтер, а также необходимость наличия сетевого оборудования.

При выборе ЭВМ необходимо руководствоваться следующим рядом характеристик:

- надежность;
- стоимость;
- производительность;
- объем памяти.

От значения указанных параметров зависит скорость и результат работы с выбранными программными средствами, и как следствие успех в разработке системы в целом.

На сегодняшний день в мире существуют ЭВМ трех классов: большие, мини-ЭВМ и микро-ЭВМ. Большие ЭВМ, как правило, имеют достаточно высокую себестоимость. Высокая цена данного оборудования в

свою очередь обеспечивает высокое быстродействие, такие ЭВМ предназначены для решения сложных задач, которые требуют большого количества вычислений. Подобный класс вычислительных машин применяют при работе с фундаментальными научными исследованиями и другими отраслями сложных вычислений. Типичным представителем класса микро-ЭВМ являются персональные электронно-вычислительные машины (ПЭВМ). Мини-ЭВМ занимают промежуточное место между большими и микро-ЭВМ [31].

Для решения стандартных экономических задач средней сложности больше всего подходят персональные ЭВМ, так как они имеют среднюю себестоимость, высокие показатели быстродействия и надежности, большие объемы памяти и, что немаловажно, небольшие размеры.

Таким образом, они могут использоваться фактически на любом предприятии и в любой сфере деятельности, в частности в работе медицинских учреждений.

При выборе ПЭВМ для реализации поставленных задач необходимо учитывать следующие характеристики:

- возможность подключение периферийных устройств;
- достаточный объем жесткого диска для возможности хранения большого количества данных;
- объем оперативной памяти, который влияет на скорость обработки запросов;
- тактовая частота процессора, позволяющая за малые промежутки времени обрабатывать входные данные.

Рабочая зона, на которой будет устанавливаться разрабатываемая подсистема автоматизации учета пациентов в медицинском учреждении, будет подключена в общую компьютерную сеть. Это необходимо для быстрого и точного обмена данными между сотрудниками медицинского центра «ОРМЕД».

Компьютерной сеть представляет собой комплекс территориально распределенных ЭВМ, которые связаны между собой каналами передачи данных. Компьютеры, обладающие сеть, в свою очередь обладают также суммарным вычислительным потенциалом, при этом они дают гарантию на повышение надежности работы системы в целом за счет создания дубликатов технических и информационных ресурсов [31]. Необходимость создания локальной сети обусловлена следующими признаками:

- создание возможности использования информационных баз данных, расположенных на территориально разграниченных компьютерах различных пользователей;
- достижение высокой скорости обработки информации, за счет использования ресурсов сразу нескольких электронно-вычислительных машин;
- возможность работы с разрабатываемым программным продуктом сразу несколькими пользователями;
- распределение нагрузки между компьютерами, соединенными в одну локальную сеть, что позволит избежать серьезной нагрузки на отдельные компьютеры [4].

Также помимо персональной ЭВМ и локальной сети, для распечатки документов необходим принтер. Существует несколько типов принтеров:

- 1) матричные, которые на сегодняшний день практически не используются;
- 2) струйные, чаще всего используются для распечатки цветных изображений или фотографий;
- 3) лазерные имеют достаточно высокую скорость печати [4].

Для работы с разрабатываемой подсистемой выбран лазерный принтер.

Исходя из анализа основных характеристик персональной ЭВМ и печатающих устройств, рабочая станция будет обладать следующими техническими характеристиками, представленными в таблице 2.1.

Таблица 2.1 – Технические характеристики

Рабочая станция	
1	2
Процессор	Intel Core i5-4440, 3.5 ГГц, 3100МГц, 5ГТ/с, S1155, iH61 Express
Материнская плата	ASUS, LGA1155, DDR3L, DDR3, PCI-E, SATA II
Жесткий диск	SATA II, 500 ГБ, 7200rpm
Корпус	INWIN EMR002, MicroATX, ATX 12V 450 Вт, 2 USB 2.0, 2 miniJack, 192x357x429 мм, 5.1 кг
Память	DIMM, DDR3 1600 (PC12800), 2 ГБ
Сеть	10/100Мбит, Atheros AR8152
Принтер	
Название	Samsung ml-2160
Максимальный формат бумаги	A4
Скорость черно-белой печати А4	20 листов в минуту
Максимальное разрешение черно-белой печати	1200 x 1200 dpi
Время выхода первого отпечатка	8 секунд
Стандартный объем памяти	9 МВ
Емкость входного лотка для бумаги	180 листов
Емкость выходного лотка для бумаги	100 листов
Тип интерфейса соединения	Hi-Speed USB 2.0
Монитор	
Название	AL1916W
Диагональ	19 дюймов

Продолжение таблицы 2.1

1	2
Разрешение экрана	1440x900
Подсветка	WLED
Угол обзора	160 градусов
Яркость	300 кд/м2
Время реакции	5 мс
Тип матрицы	TN
Количество цветов	16.7 млн. цветов
Интерфейс подключения	VGA (D-Sub)

2.1.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение представляет собой совокупность единой системы классификации информации, схем информационных потоков, используемых в организации, а также методология построения баз данных [24].

Одно из наиболее важных требований к информационному обеспечению является достоверность информации, хранящейся в базе данных. Необходимая достоверность данных в базе достигается за счет высокого уровня контроля на всех стадиях работы с информацией.

В состав подсистемы также должна входить отдельная специализированная подсистема, позволяющая осуществлять восстановление потерянных данных и резервное копирование информации.

Структура и состав организации данных в подсистеме определяются на этапе технического проектирования подсистемы. Хранение информации должно быть основано на работе с современными реляционными базами

данных. Для обеспечения целостности данных работа с СУБД должна осуществляться квалифицированными специалистами. Доступ к хранящейся в базе информации предоставляется только авторизованным пользователям с учетом уровня их полномочий.

Выбор оптимальной СУБД для работы с базой является одним из главных этапов разработки информационной подсистемы.

Для решения задачи автоматизации учета пациентов в медицинском центре «ОРМЕД» была выбрана система управления базами данных Firebird.

Данная СУБД позволяет обеспечить полный контроль над процессом работы с данными, их обработкой и своевременным использованием. Также существенно упрощает структуризацию и обработку больших объемов данных, которые хранятся в таблицах[33].

СУБД Firebird сервера обеспечивает возможность параллельной работы на многопроцессорном оборудовании. При работе с Borland Builder C++ предоставляется ряд функциональных компонентов InterBase eXpress (IBX), позволяющих осуществлять полноценную работу с этим сервером [23].

Разнообразные средства для работы с данными, обеспечивают выполнение трех основных функций: определение, обработка и управление данными:

- определение данных позволяет определить то, какая информация будет храниться в базе данных, тип этих данных и их возможные связи;
- обработка данных предоставляет возможность обрабатывать полученную информацию различными способами: выбирать конкретные поля данных, фильтровать и сортировать информацию, объединять данные для получения каких-либо итоговых значений. Также можно, выбрав некоторые данные, изменить их, удалить, скопировать их в другую таблицу или создать для них абсолютно новую таблицу;

– управление данными позволяет ограничить доступ к информации, то есть назначать разрешение на возможность просмотра, добавления или изменения каких-либо данных в базе [33].

Выбор СУБД включает в себя определение многих факторов, но главным из них является возможность работы с построенной моделью данных. Имеются системы, используемые для работы с сетевыми и иерархическими моделями, однако большинство систем управления базами данных для персональных ЭВМ осуществляют работу с реляционной моделью. Реляционные СУБД для персональных ЭВМ различаются набором реляционных операций, которые может выполнять выбранная СУБД [22].

Все обрабатываемые данные в реляционных СУБД представляются в виде информационных таблиц. Информация о каждом конкретном объекте хранится в таблице, каждый столбец которой включает в себя значения характеристики данных объектов, а строка представляет собой описание конкретного объекта. Также предоставляется возможность объединить информацию из нескольких таблиц или запросов для удобства работы с данными [2].

Для выделения основных сущностей разрабатываемой подсистемы автоматизации и определения связей между ними, была построена диаграмма «сущность-связь» (ERD), представленная на рисунке 2.1.



Рисунок 2.1 - ER-модель базы данных автоматизированной подсистемы

ER-диаграмма отражает самые главные для выбранной предметной области информационные объекты, которые на диаграмме называются сущностями, их свойства – атрибуты и связи между данными объектами [24].

ER-модель базы данных включает следующие сущности:

- таблица «Расписание» представляет собой график работы сотрудников, в котором указывается день недели и часы работы сотрудника в этот день;

- таблица «Сотрудники» включает в себя персональные данные человека, который является по базе сотрудником организации, с указанием его должности;

- таблица «Должности» - перечень всех имеющихся должностей в медицинском центре;

- таблица «Люди» представляет собой перечень всех людей, персональные данные которых указаны в базе. Данная таблица используется для того, чтобы не сохранять в базе одинаковые данные, так как сотрудник также может быть и клиентом медицинского центра;

- таблица «Отпуска сотрудников» - данные, содержащие период отсутствия сотрудника, для исключения его из графика на время отпуска.

- таблица «Пациенты» включает в себя данные всех клиентов зарегистрированных в базе медицинского центра;

- таблица «Талоны» предназначена для указания данных заявки;

- таблица «Запись на прием» - основная таблица, непосредственно предназначенная для формирования заявок, в которых будет указана вся необходимая информация о предоставлении услуг;

- таблица «Вид услуги» используется для разбиения всех услуг на отдельные категории, это используется для удобства разграничения обязанностей между докторами;

- таблица «Прием-услуга» предназначена в базе для хранения данных по заявке на предоставление услуг с указанием их количества;

– таблица «Услуги» включает в себя наименование всех услуг, которые предоставляет медицинский центр с указанием стоимости каждой из них.

2.1.3 Обоснование проектных решений по программному обеспечению

В настоящее время развитие информационных технологий в различных областях экономики приводит к постоянному усложнению используемых операций, для того, чтобы в свою очередь упростить пользователям работу с данными. Современные проекты по разработке информационных систем характеризуются следующими особенностями:

- сложность описания системы, это связано с тем, что разрабатываемые системы включают в себя большое количество функций, элементов и процессов, которым необходимо поэтапный анализ данных;
- отсутствие аналогов ограничивают возможность использования подобных проектных решений;
- необходимость интеграции разрабатываемых приложений [3].

Для успешной реализации проекта, объект исследования должен быть тщательно изучен, должны быть построены информационные и функциональные модели будущей автоматизированной подсистемы.

Построение моделей напрямую связано с отдельным классом программно – технологических средств - CASE-средств. Данные средства используются для сопровождения информационных систем. На сегодняшний день понятие CASE-технологии охватывает весь процесс разработки ИС. В области современного программирования под термином CASE-средства понимаются программные средства, поддерживающие процессы разработки и сопровождения информационных систем, включая анализ требований,

возможность проектирования прикладного ПО и баз данных, генерацию кода, тестирование, документирование, а также другие процессы [3].

Проектирование подсистемы автоматизации учета пациентов в медицинском центре «ОРМЕД» производилось с использованием следующих CASE-средств:

- ERwin - средство моделирования базы данных, использующее методологию IDEF1X. При помощи ERwin выполняется проектирование схемы баз данных и реинжиниринг существующей базы. ERwin выпускается в нескольких различных конфигурациях, каждая из них ориентирована на наиболее распространенные средства разработки приложений 4GL [24];

- AllFusion Process Modeler 7 (ранее BPWin) представляет собой средство функционального моделирования, реализующее методологию IDEF0, DFD. С помощью данного программного продукта можно проводить моделирование, анализ и описание бизнес-процессов. Графическое изображение схемы позволяют более точно представить работу конкретной организации или отдельно взятого процесса [13].

Также с помощью BPwin можно организовать подробное документирование всех аспектов бизнес-процессов, способов контроля над ними и впоследствии возможность визуализировать полученные в ходе анализа данные. С помощью BPwin можно увидеть полную картину организации деятельности предприятия: от количества работы в небольших подразделениях предприятия до сложных функций организации предприятия [13].

2.2 Структурно функциональный анализ деятельности организации «КАК ДОЛЖНО БЫТЬ»

Задача выпускной квалификационной работы заключается в разработке автоматизированной подсистемы учета пациентов в медицинском центре. Разработка позволит упростить работу регистрационного отдела, за счет того, что максимально автоматизирует процесс оформления заявок.

Исходя из требований, которые были определены в предыдущих разделах, реализована модель функциональных модулей подсистемы, представленная на рисунке 2.2.

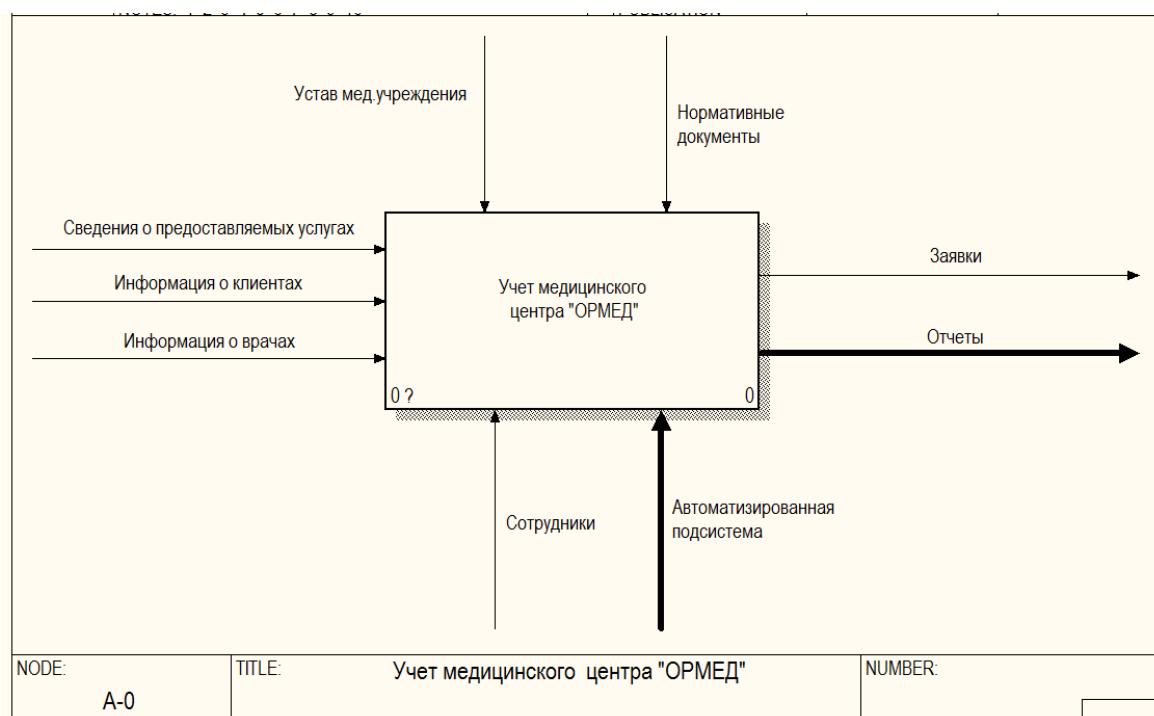


Рисунок 2.2 – Контекстная диаграмма «КАК ДОЛЖНО БЫТЬ»

На представленной диаграмме видно, что в процесс учета пациентов медицинского центра включается автоматизированная подсистема, которая будет выполнять основную работу по формированию заявок. Также подсистема предоставит возможность формирования отчетов, выполнение которых ранее было невозможным. Подобные отчеты можно будет

формировать как по деятельности всей организации в целом, так и по отдельным параметрам, подлежащим анализу.

Далее на рисунке 2.3 представлена диаграмма декомпозиции после выполнения автоматизации.

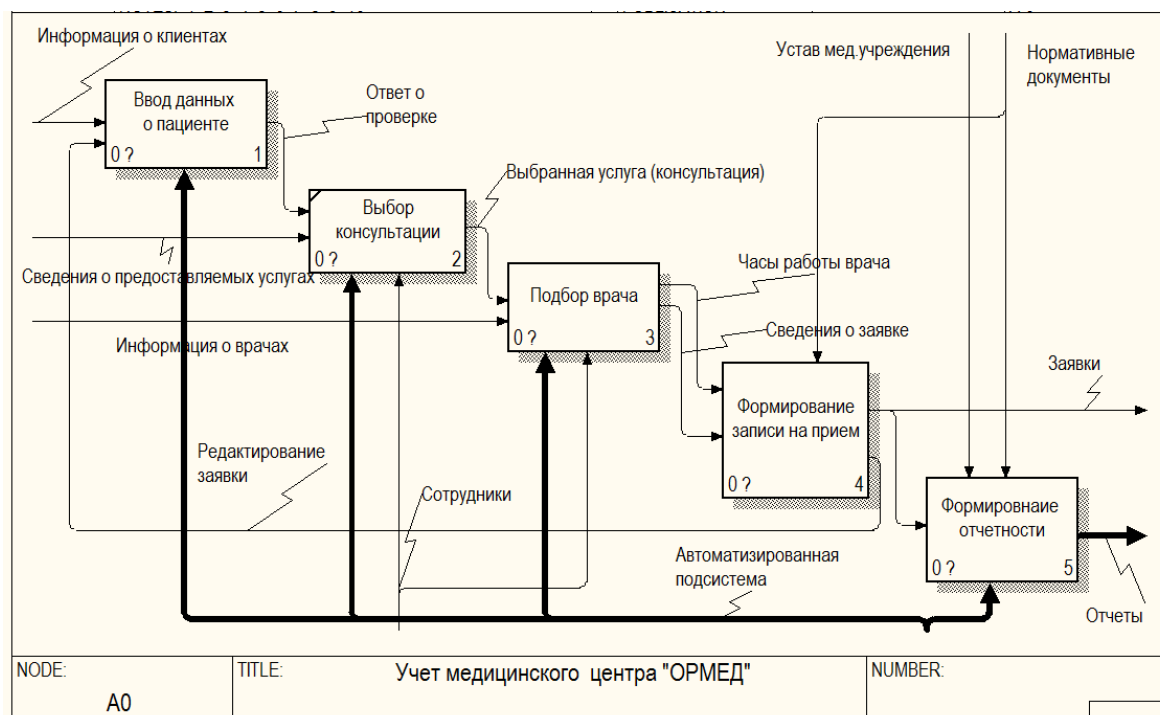


Рисунок 2.3 – Диаграмма декомпозиции «КАК ДОЛЖНО БЫТЬ»

С точки зрения взаимодействия можно выделить пять основных групп пользователей:

- системный администратор, в обязанности которого входит настройка и мониторинг работоспособности технических средств, установка и настройка программного обеспечения, ведение учетных записей пользователей;

- руководитель, которому необходим доступ к подсистеме для проведения анализа работы организации. Руководство медицинского центра получает от данной подсистемы всевозможные формы отчетности и анализа, как организации в целом, так и отдельных функциональных объектов, работа которых происходит в данной подсистеме;

- регистратор непосредственно выполняет добавление новых клиентов и занимается оформлением заявок на прием;
- кассир имеет доступ ко всем сформированным заявкам и возможность изменения статуса оплаты оформленных услуг;
- врач использует данную подсистему для просмотра назначенных на него заявок на прием.

Таким образом, в зависимости от того, кто входит в систему будут определены ограничения связанные с работой различного рода функций подсистемы.

Модель формирования записи на прием, представленная на рисунке 2.4, выполнена в методологии DFD. Данная методология предоставляет возможность использовать в функциональных моделях хранилища данных, которые в подсистеме представлены в виде отдельных таблиц базы данных [14].

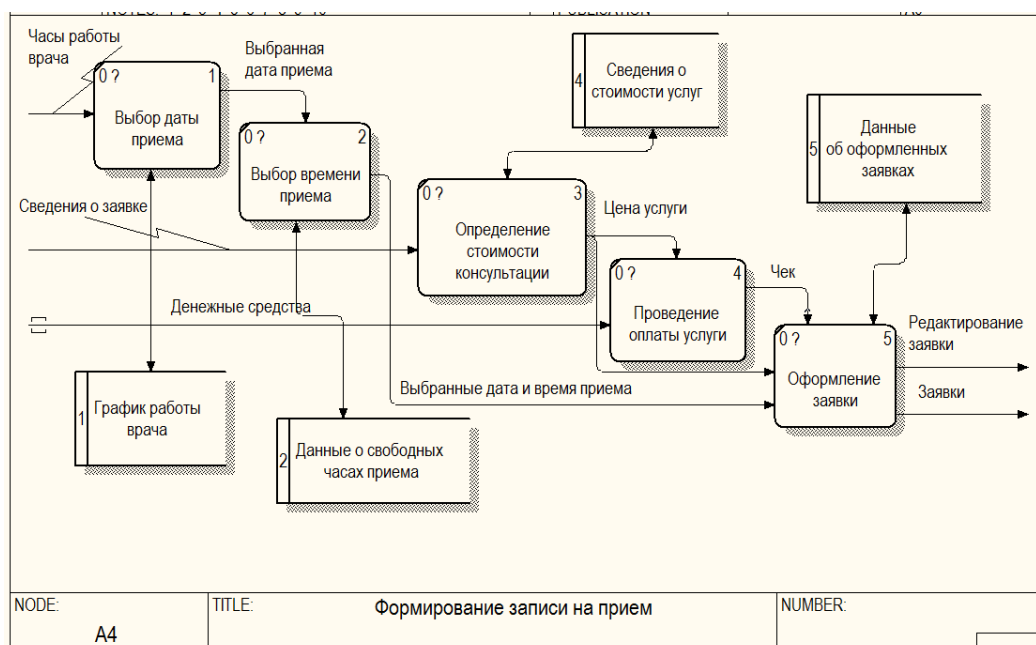


Рисунок 2.4 – Функциональная модель формирования записи на прием

Детализация других функциональных блоков диаграммы декомпозиции представлены в приложении Б.

Во втором разделе выпускной квалификационной работы были обоснованы проектные решения по техническому, информационному и программному обеспечению. Также был произведен выбор программных средств, которые будут использованы при разработке подсистемы учета пациентов в медицинском центре «ОРМЕД». Сформированы функциональные модели деятельности организации «КАК ДОЛЖНО БЫТЬ», позволяющие показать работу регистрационного отдела после введения автоматизации. Произведено описание логического уровня базы данных для подсистемы.

3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ

3.1 Разработка базы данных автоматизированной подсистемы

Следующим этапом после моделирования является непосредственно создание базы данных, состоящей из 11 таблиц, представленных выше на рисунке 2.1, и заполнение их информацией с использованием утилиты IVExpert. На рисунке 3.1 представлено окно параметров для создания базы данных.

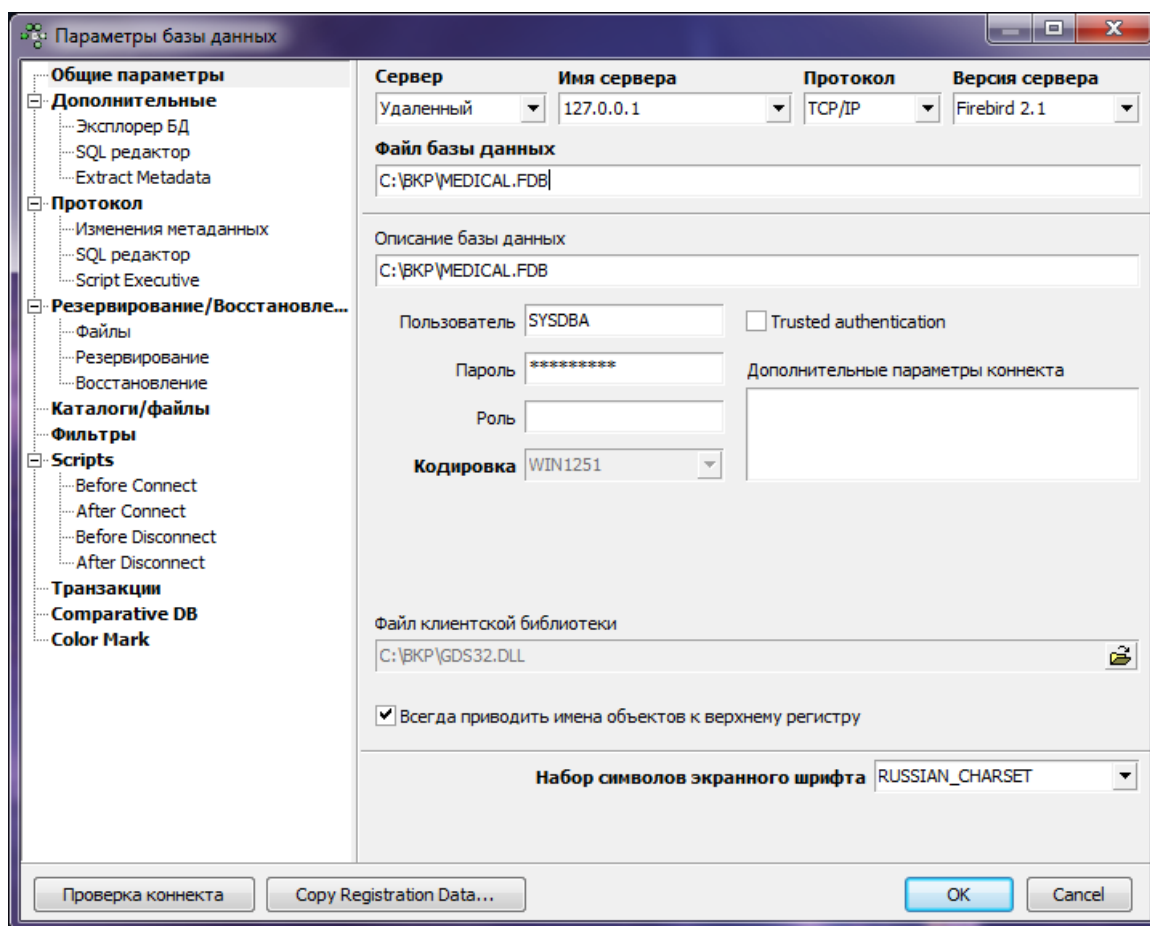


Рисунок 3.1 – Окно параметров базы данных

Также для упрощения работы с данными были созданы:

- домены - глобальное описание типа поля, доступное всей базе данных;
- индексы для некоторых полей таблиц, которые предназначены для ускорения получения доступа к записям, отбираемым в соответствии с какими-либо условиями;
- генераторы, которые представляет собой механизм, создающий уникальную последовательность чисел и автоматически заполняющий заданное поле при вставке или обновлении записей;
- представления, в которых выполнялись запросы к таблицам, объединение данных, группировка и использование агрегатных функций. Просмотры (представления) позволяют возвращать наборы данных, удовлетворяющие нужды конкретных пользователей или групп (рисунок 3.2) [8];

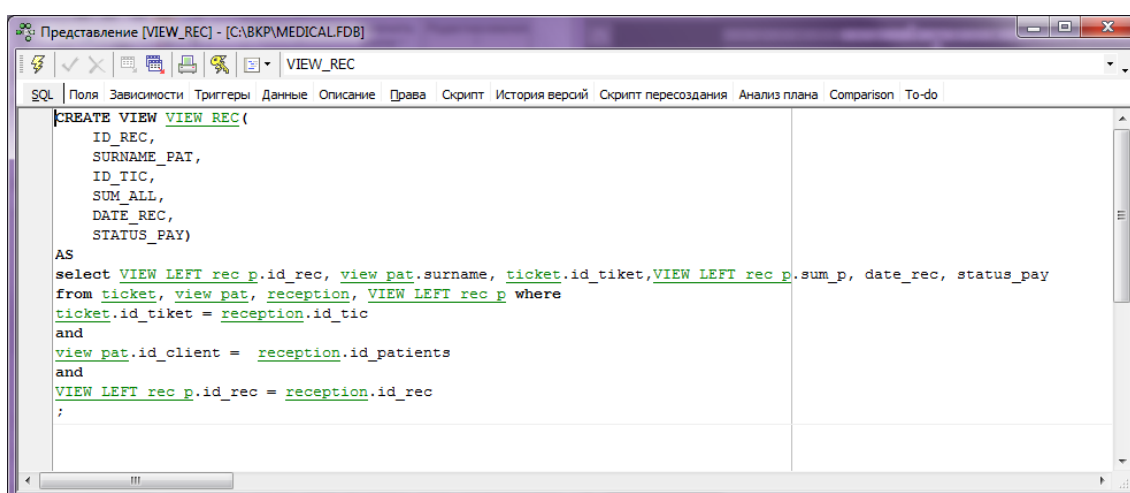


Рисунок 3.2 – Создание представления на объединение параметров из нескольких таблиц

- хранимые процедуры выполнения для ввода, модификации и удаления информации в таблицах БД и хранимые процедуры выборки [2]. Хранимая процедура представляет собой программу, хранящуюся на сервере в двоичном виде, как часть базы данных [5].

Пример создание процедуры добавление представлен на рисунке 3.3.

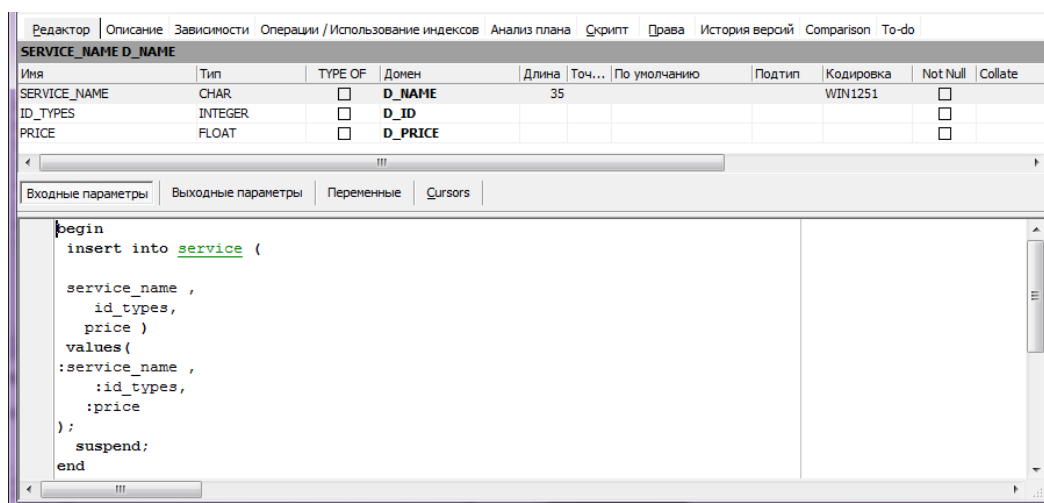


Рисунок 3.3 – Создание хранимой процедуры добавление записей в таблицу

Хранимые процедуры на изменение записей в таблице используется в подсистеме для исправления ошибок, допущенных при вводе данных. На рисунке 3.4 представлен пример создание процедуры изменения данных в таблице [28].

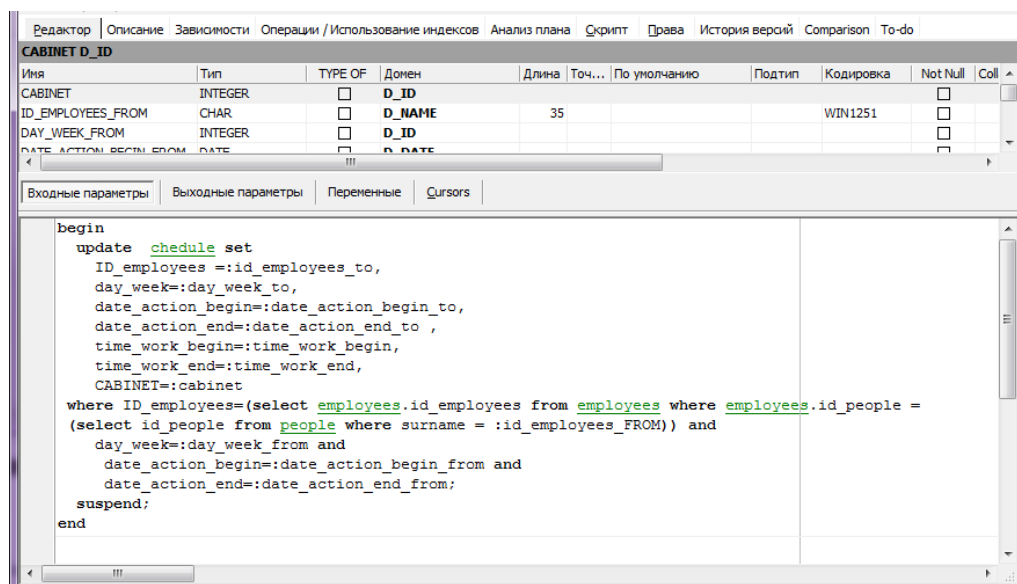


Рисунок 3.4 – Создание хранимой процедуры на изменение данных в таблице

SQL код создания всех элементов баз данных, которые использовались в работе подсистемы, приведен в приложении В.

3.2 Программное обеспечение задачи разработки подсистемы

Одной из основных целей выпускной квалификационной работы является разработка программного продукта для медицинского центра «ОРМЕД», который позволит максимально автоматизировать процесс учета пациентов.

В качестве системы для разработки программного продукта была взята интегрированная среда разработчика Borland C++. Разработка автоматизированной подсистемы основана на использовании одного проектирующего модуля, то есть программа состоит из одной формы, на которой размещены разделы. В каждом разделе происходит выполнение конкретного рода операций [27].

DataModule – контейнере для компонентов и доступа к данным. Главными компонентами, которые располагаются на DataModule являются:

- компонент IBDataBase служит для установления соединения с базой данных расположенной на сервере Firebird;
- компонент IBTransaction предоставляет свойства и методы для управления транзакциями, связывается с компонентом IBDataBase. Транзакцией называют логически связанный с базой данных блок операций, выполняющийся как единой целое или вовсе невыполняющийся;
- IBTable необходим для подключения таблиц;
- DataControls являются элементами управления и представления данных;
- IBQuery представляет собой набор данных, используемых для выполнения запросов;
- компонент IBStoredProc служит для выполнения хранимых процедур;
- DataSource необходим для соединения таблиц с DBGrid [26].

На рисунке 3.5 представлен DataModule9 со всеми используемыми в процессе разработки прикладного решения компонентами.

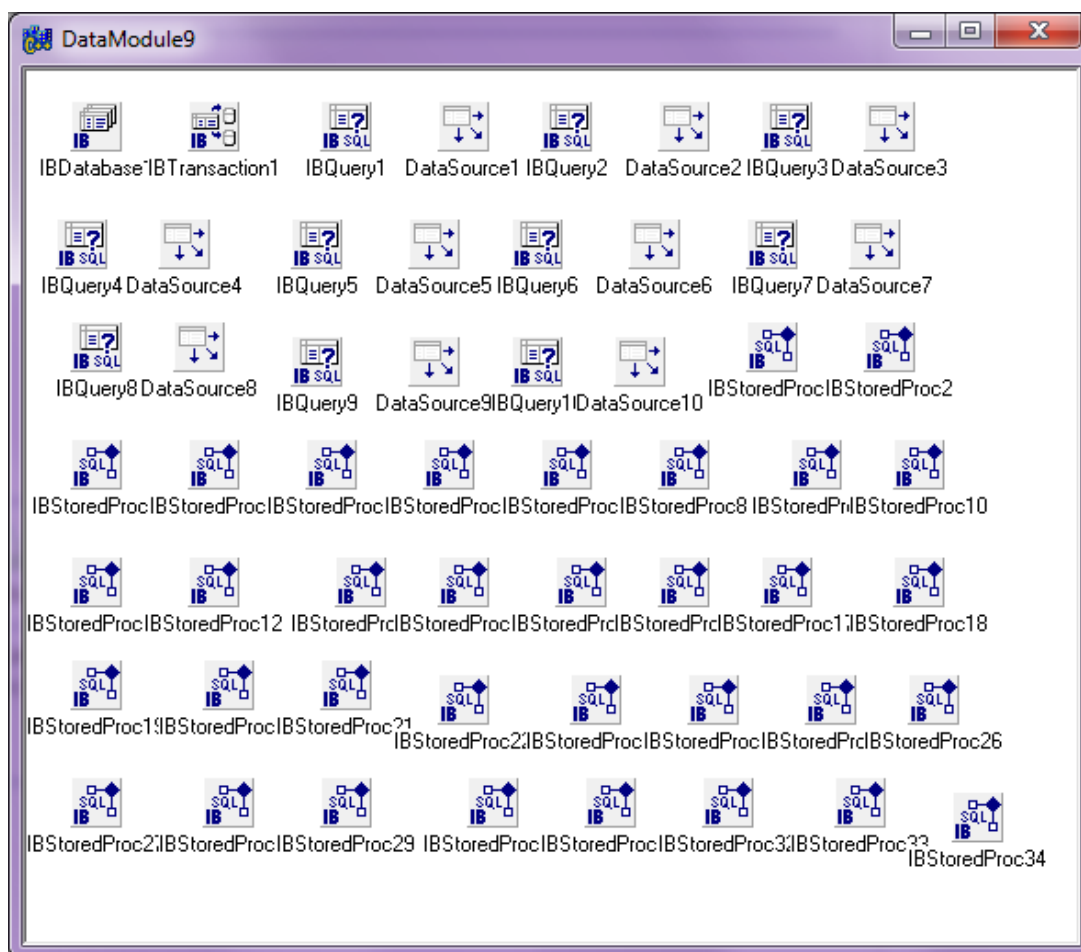


Рисунок 3.5 – DataModule9

Первый этап разработки приложения заключается в подключении, созданной в IBExpert, базы данных, для получения доступа к информации, которая уже заполнена. Подключение к базе происходит через окно «Database Component Editor», в котором прописывается путь к базе, а также логин и пароль, которые были указаны при создании [11].

Настройка доступа к базе данных представлена на рисунке 3.6.

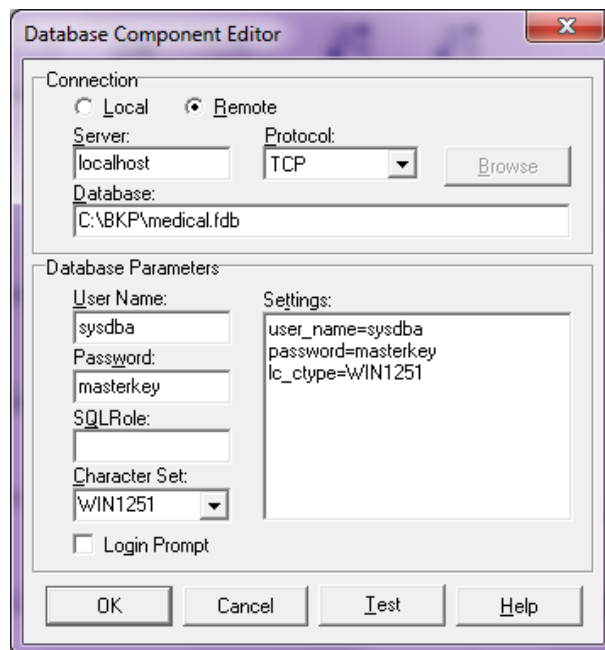


Рисунок 3.6 – Установка соединения с базой данных

Для корректной работы с функциями подсистемы был разработан этап авторизации сотрудников, получающих доступ к данным медицинского центра. Окно выполнения авторизации представлено на рисунке 3.7, оно включает в себя следующие компоненты: Label, ComboBox, Edit, Button [30].

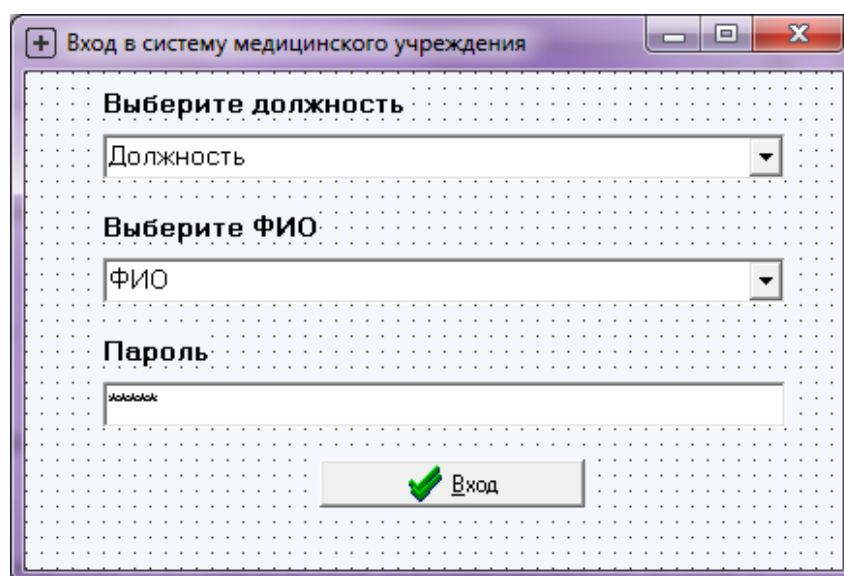


Рисунок 3.7 – Окно входа в систему

Далее проходила разработка главного окна подсистемы, которое представлено на рисунке 3.8. Основная форма содержит компонент MainMenu, позволяющий выполнять навигацию – переход на другие формы, которые выполняют определенный функционал [9].

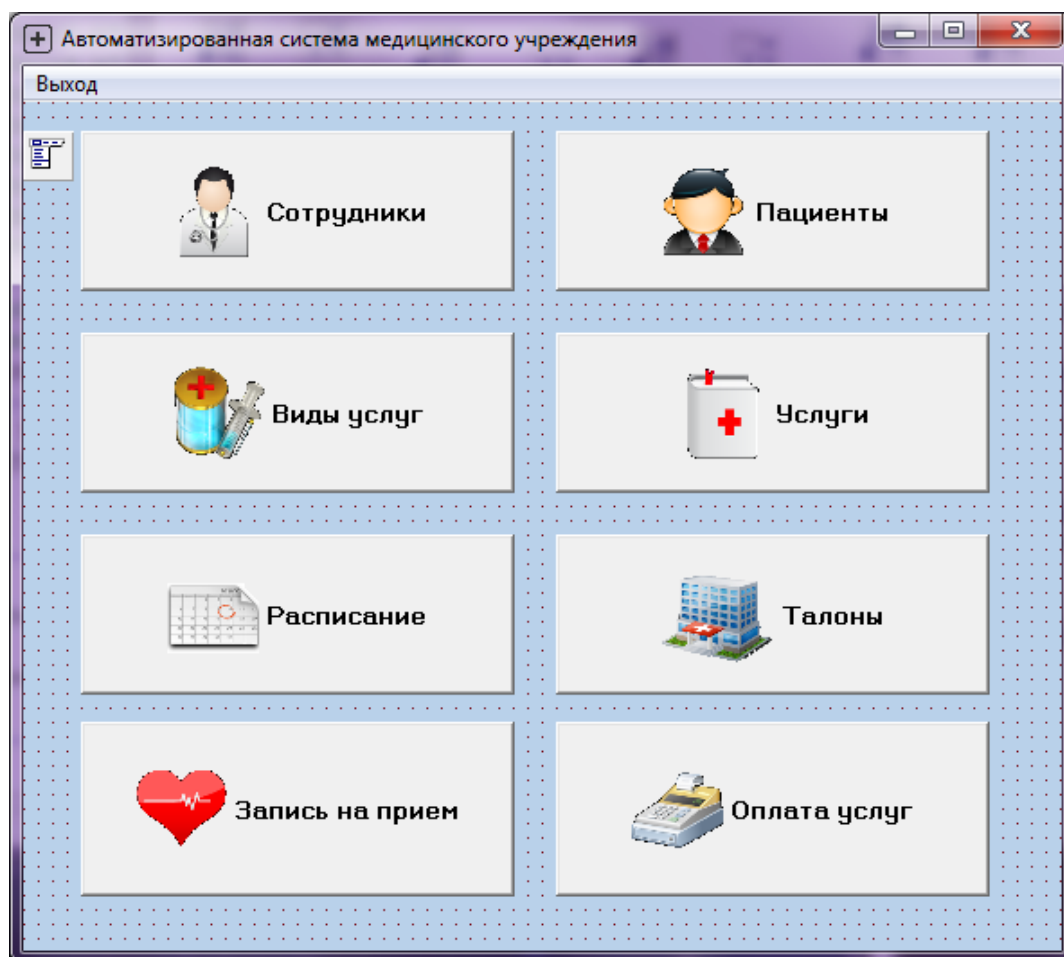


Рисунок 3.8 – Основная форма разрабатываемой подсистемы

Также вид данной формы будет зависеть от того, кто из сотрудников вошел в систему, так как права доступа на выполнения каких-либо операций напрямую связаны с должностью сотрудника, вошедшего в учетную запись. Например, если для сотрудника ограничен доступ на добавление новых услуг, но кнопка «добавление» для него будет неактивна.

Следующая форма, изображенная на рисунке 3.9, отвечает за работу со списками сотрудников медицинского центра, также данная форма

предоставляет возможность осуществлять разного рода сортировки и фильтрации для удобства работу с данными [6].

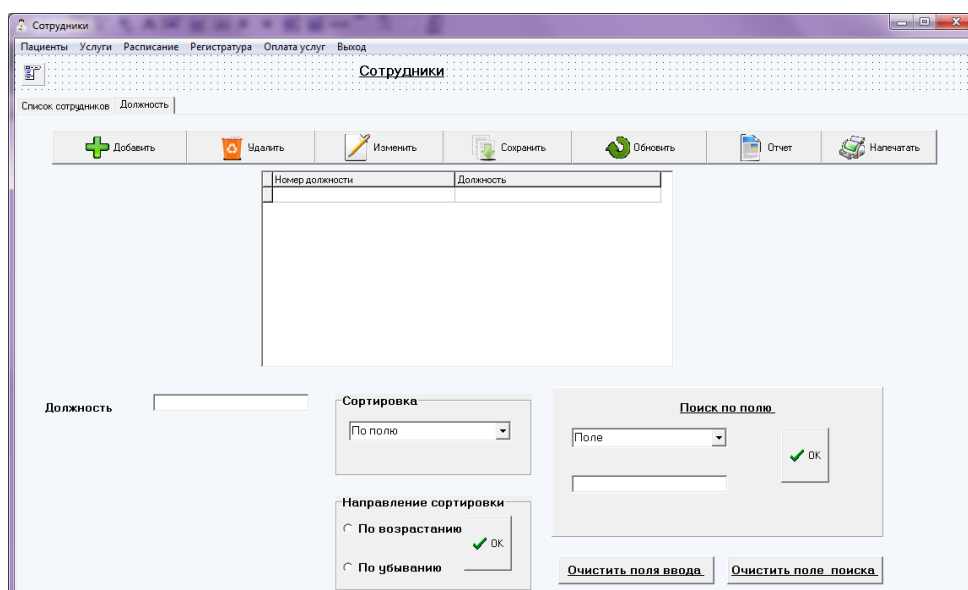


Рисунок 3.9 – Разрабатываемая форма «Сотрудники»

Форма для регистрации новых пациентов, возможности сортировки и поиска клиентских данных представлена на рисунке 3.10.

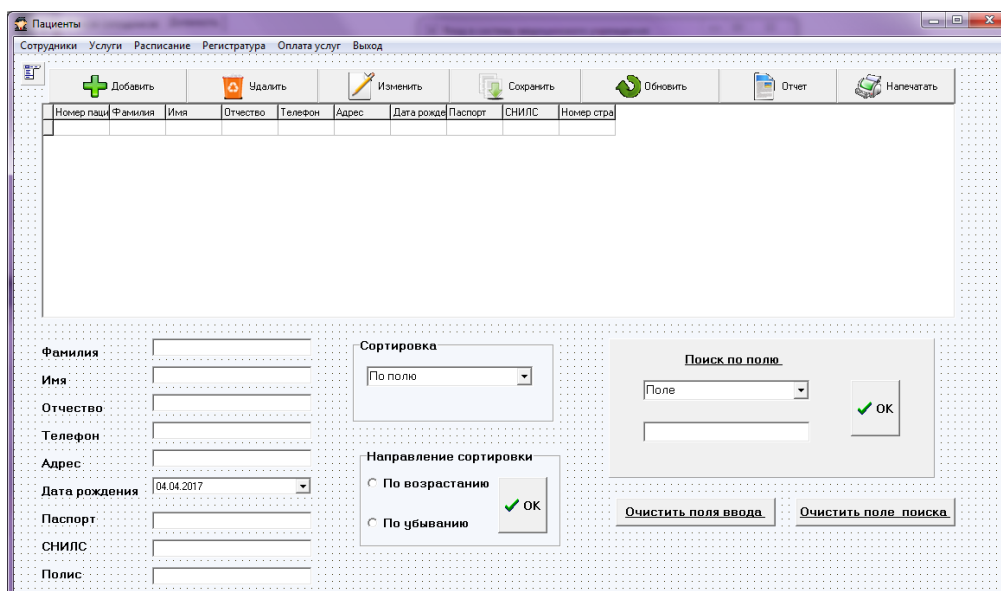


Рисунок 3.10 – Форма работы с клиентскими данными

На форме, которая занимается формированием отчетов (см. рисунок 3.11), используются следующие компоненты:

- QuickRep представляет собой основу для создания отчета;
- ITable – компонент, предназначенный для выбора информации из таблиц в отчеты;
- DataSource – данный компонент необходим для установления связи между компонентами на форме и таблицей базы данных [7].

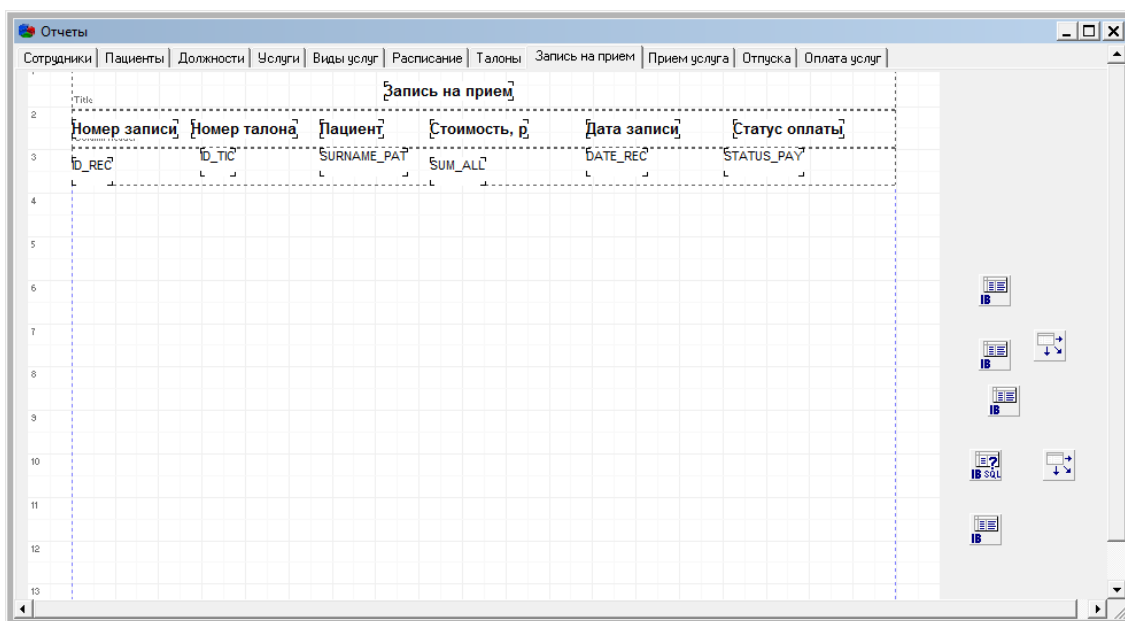


Рисунок 3.11 – Форма для работы с отчетами

Программный код, использованный при разработке программного продукта, и скриншоты созданных форм представлены в приложении Г.

3.3 Проверка работоспособности разработанного приложения

В процессе тестирования автоматизированной подсистемы была проведена проверка работоспособности приложения. Для работы с системой необходимо запустить файл Project1.exe и в появившемся окне входа в систему указать должность, ФИО и пароль сотрудника, который на данный момент входит в подсистему (см. рисунок 3.12).

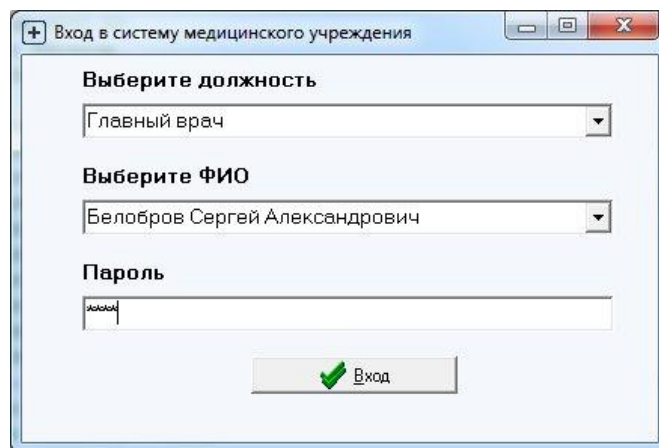


Рисунок 3.12 – Окно входа в подсистему

После авторизации работа подсистемы переходит в основное меню для выбора операции, необходимой сотруднику. На рисунке 3.13 представлена главное окно подсистемы под авторизацией главного врача.

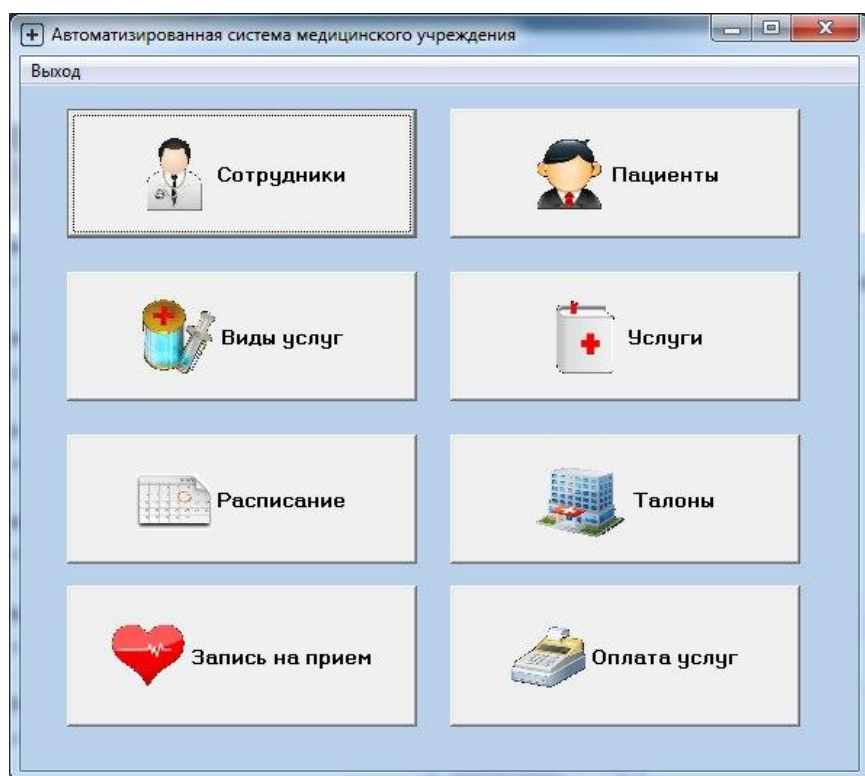


Рисунок 3.13 – Главное окно автоматизированной подсистемы

Каждая кнопка на главной странице при нажатии переводит клиента в меню, которое осуществляет определенный рабочий функционал.

Нажав на кнопку «Сотрудники» пользователь переходит в меню работы с данными работающего персонала медицинского центра «ОРМЕД». Обладая правами на работу с этими данными, пользователь может добавлять новых и удалять уже имеющихся сотрудников, также осуществлять сортировку и поиск данных по сотрудникам. На рисунке 3.14 представлен пример работы сортировки.

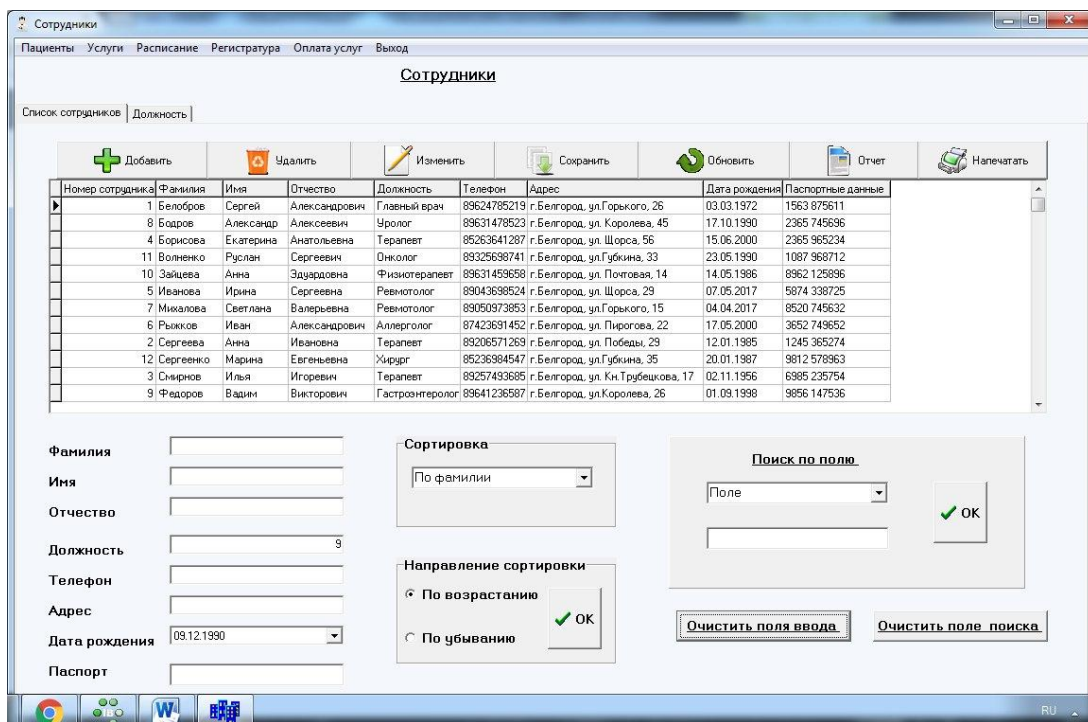


Рисунок 3.14 – Сортировка сотрудников по фамилии

При переходе из главного меню в режим работы с услугами медицинского учреждения пользователь имеет возможность выполнять операции связанные с добавлением, удалением или изменением услуг. Также пользователь может сортировать данные и использовать поиск для нахождения конкретной услуги. Используя в данном режиме кнопку «Просмотреть рейтинг услуг» пользователю предоставляется возможность отследить, процентное соотношение услуг, которые формируются пациентами, для проведения какого-либо сводного анализа (см. рисунок 3.15).

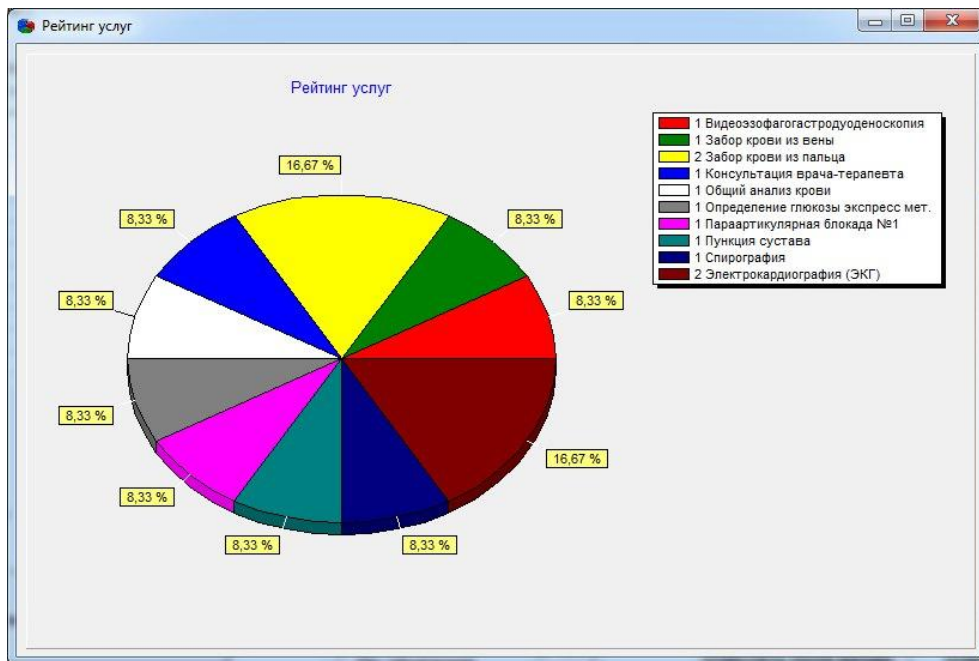


Рисунок 3.15 – Рейтинг услуг

На рисунке 3.16 представлен пример работы поиска услуги при указании вида услуги через выпадающий список и наименование услуги.

Номер услуги	Наименование услуги	Вид услуги	Стоимость, р
7	Забор крови из пальца	Лабораторные исследования	50

Услуга:

Вид услуги:

Стоимость, р:

Сортировка:

Направление сортировки: По возрастанию По убыванию

Поиск по полю:

Рисунок 3.16 – Поиск услуги

Кнопка «Запись на прием», расположенная в главном меню, переводит клиента в режим формирования заявки на прием, представленный на рисунке 3.17.

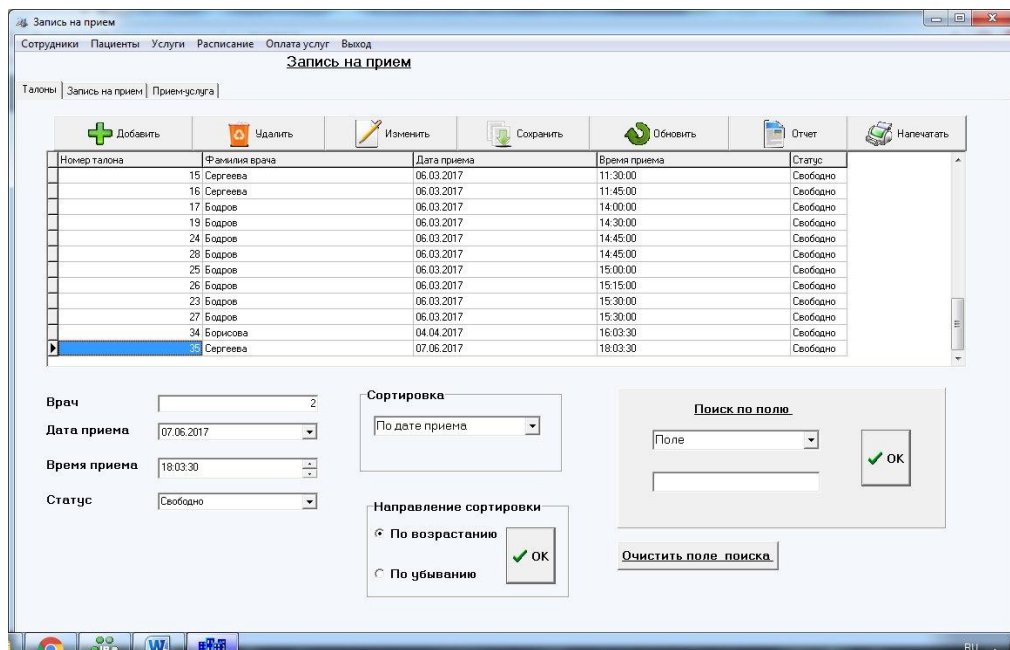


Рисунок 3.17 – Формирование записи на прием

Формирование записи осуществляется в несколько этапов:

- 1) добавляем пациента и списка всех клиентов медицинского центра (см. рисунок 3.18);

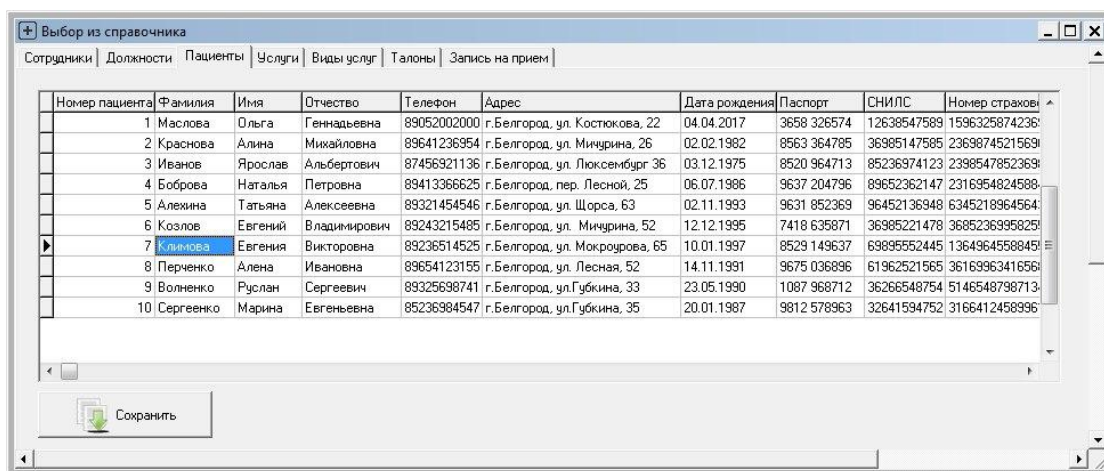


Рисунок 3.18 – Выбор пациента из списка

2) формируем талончик с указанием даты и времени приема;

При формировании заявки статус оплаты по умолчанию указывается «не оплачен», его меняет кассир при оплате квитанции.

3) запись добавлена, при этом поле стоимости приема – пустое;

4) переходим на вкладку «Выбор услуги», указываем из справочника необходимые услуги и их количество;

5) после ввода данных по услугам в поле «Стоимость приема» автоматически подтянется нужная сумма, которая будет рассчитываться из учета стоимости и количества указанных в заявке услуг (см. рисунок 3.19).

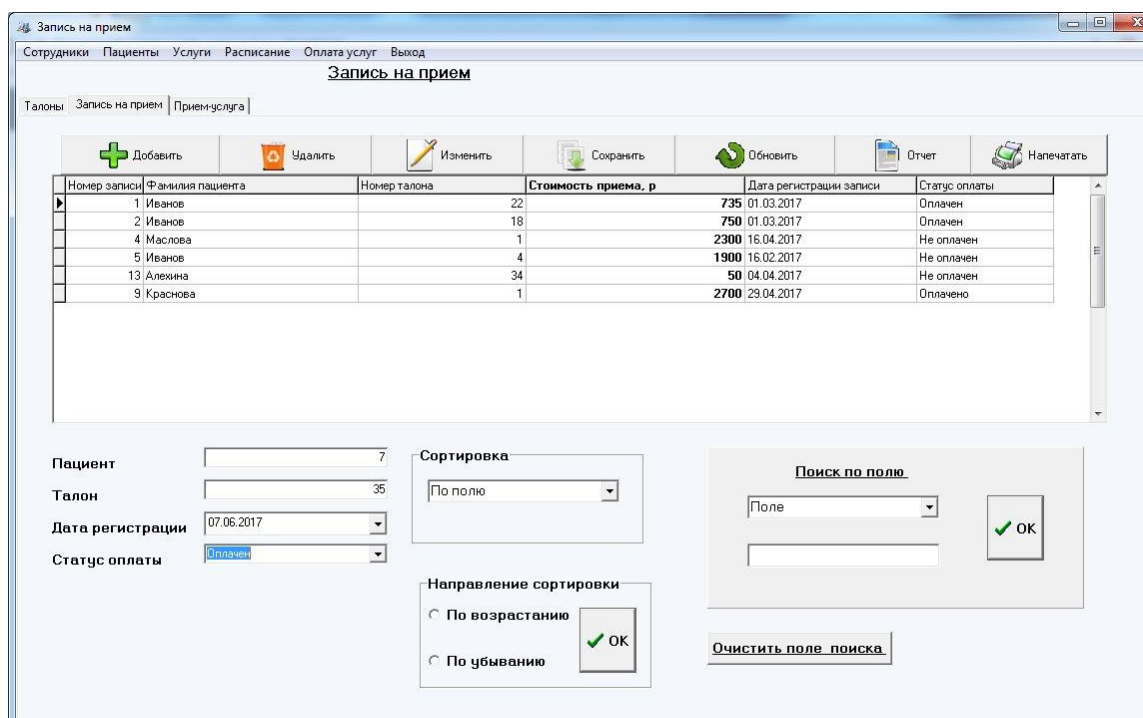


Рисунок 3.19 – Сформированные заявки на прием

б) после того, как в заявке подтянулась стоимость приема, необходимо распечатать клиенту квитанцию для оплаты, которая осуществляется через кассу медицинского центра.

Кассир, который также имеет доступ к подсистеме, после оплаты меняет статус заявки на «Оплачено». На рисунке 3.20 представлен список записей на прием с указанием статуса заявки на текущий момент.

Номер записи	Фамилия пациента	Номер талона	Стоимость приема	Дата	Статус оплаты
1	Иванов	22	735	01.03.2017	Оплачен
2	Иванов	18	750	01.03.2017	Оплачен
4	Маслова	1	2300	16.04.2017	Не оплачен
5	Иванов	4	1900	16.02.2017	Не оплачен
13	Алехина	34	50	04.04.2017	Не оплачен
14	Климова	35		07.06.2017	
9	Краснова	1	2700	29.04.2017	Оплачено

Рисунок 3.20 – Список заявок

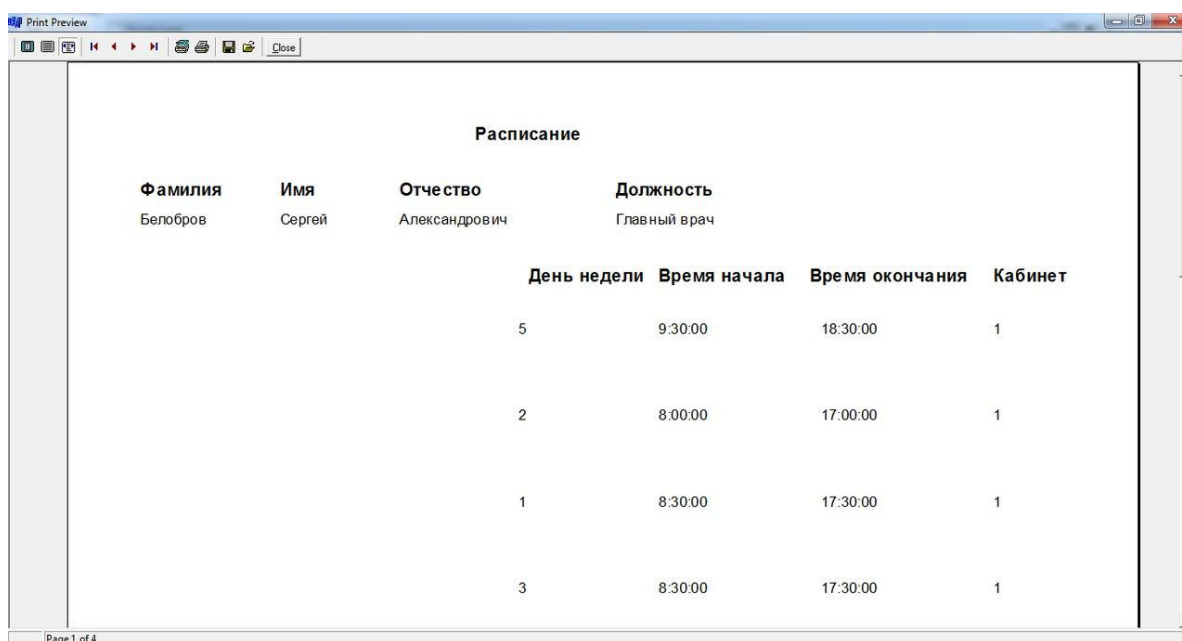
Квитанция на оплату услуг, выдаваемая пациенту медицинского центра, представлена на рисунке 3.21.

Оплата услуг					
Номер записи	Номер талона	Пациент	Стоимость, р	Дата записи	Статус оплаты
9	1	Краснова	2700	29.04.2017	Оплачено
Услуга					Количество
Электрoкардиография (ЭКГ)					1
Видеоэзофагогастродуоденоскопия					2
Время и дата оплаты		07.06.2017 0:48:01		ФИО Сотрудника _____	
				Подпись _____	

Рисунок 3.21 – Квитанция на оплату

При работе с данными подсистема позволяет формировать отчеты по всем разделам. Отчеты предназначены для просмотра данных в более удобном формате. Также подсистема дает возможность при необходимости распечатать данные отчеты.

На рисунке 3.22 представлено расписание приемных часов главного врача в виде отчета.



The screenshot shows a 'Print Preview' window with a table titled 'Расписание' (Schedule). The table contains personal information and a list of reception hours.

Фамилия	Имя	Отчество	Должность	День недели	Время начала	Время окончания	Кабинет
Белобров	Сергей	Александрович	Главный врач				
				5	9:30:00	18:30:00	1
				2	8:00:00	17:00:00	1
				1	8:30:00	17:30:00	1
				3	8:30:00	17:30:00	1

Рисунок 3.22 – Расписание, представленное в виде отчета

Скриншоты разработанной автоматизированной подсистемы, которые не представлены в данной главе, приведены в приложении Д.

3.4 Оценка экономической эффективности

Экономическая эффективность определяется, как показатель соотношения экономического результата и затрат, породивших этот результат [43].

Эффективность автоматизированной подсистемы можно определить качеством и эффективностью процесса разработки и ее сопровождения. Качество программного средства определяется:

- качеством с точки зрения пользователя, который непосредственно работает с данной подсистемой;

- качеством использования имеющихся данных и их дальнейшей обработки и анализа;
- полным выполнением поставленных требований от заказчика.

Для расчета экономической эффективности автоматизированной подсистемы, разработанной в ходе выполнения выпускной квалификационной работы, проводим подсчет времени, которое было затрачено на ее разработку. Для этого были определены показатели трудоемкости разработки, представленные в таблице 3.1.

Таблица 3.1 – Показатели трудоёмкости разработки

Этапы	Содержание этапов	Доля этапа трудоемкости
1. Подготовительный этап	Исследование деятельности медицинского центра и существующих методов работы в сфере учета пациентов.	30 дней
2. Этап теоретических разработок	Разработка алгоритма решения поставленных задач, обоснование цели разработки и составление расчетов.	25 день
3. Построение схем	Формирование диаграмм, описывающих работу организации до и после автоматизации.	30 дней
4. Разработка базы данных	Создание базы данных и заполнение ее информацией.	35 дней
5. Формирование интерфейса	Создание максимально интуитивно-понятного для пользователя интерфейса.	30 дней
6. Программирование	Программная реализация подсистемы учета пациентов в медицинском центре	65 дней
7. Проверка работоспособности	Тестирование и оптимизация разработанного программного продукта.	25 дней
8. Техническая отчетность	Определение экономического эффективности разработанного проекта	20 дней
9. Заключительный этап	Оформление результатов по проделанной работы	10 дней
	ИТОГО:	270 дней

После расчета показателей трудоемкости разработки было определено время, которое было затрачено на создание подсистемы учета пациентов в медицинском учреждении. Далее были рассчитаны эксплуатационные расходы, которые представлены в таблице 3.2.

Таблица 3.2 – Исходные данные

№ п/п	Наименование показателя	Ед. измер.	Значения показателей	
			до разработки подсистемы	после разработки подсистемы
1.	Коэффициент приведенных затрат к единому году	%	-	25
2.	Коэффициент отчислений	%	30	30
3.	Стоимость электроэнергии	руб.	2,53	2,53
4.	Заработная плата программиста	руб.	12000	12000
5.	Заработная плата аналитика	руб.	9000	-
6.	Заработная плата инженера	руб.	7000	-
7.	Заработная плата системного администратора	руб.	11000	11000
8.	Время на разработку	мес.	9	9

Произведем расчет эксплуатационных расходов на функционирование подсистемы, если потребление электроэнергии за месяц равно 200 кВт/ч.

$$P_{\text{ЭН}} = 200 * 9 * 2.53 = 4554 \text{ руб.}$$

Таким образом, расходы за электроэнергию, включая весь период разработки автоматизированной подсистемы учета пациентов в медицинском центре, составили 4554 рубля.

До создания подсистемы в штат медицинского центра входили: инженер, аналитик, системный администратор и программист. Следовательно, заработная плата основного персонала составляла:

$$P_{\text{ОСН}} = (12000.00 + 9000.00 + 7000.00 + 11000.00) * 9 * 0.3 = 105300.00 \text{ руб.}$$

Отсюда вычисляем заработную плату административно-управленческого персонала:

$$P_{\text{ауп}} = 0.25 * 105300.00 = 26325.00 \text{ руб.}$$

В таблице 3.3 приведены расходы на материалы до разработки автоматизированной подсистемы.

Таблица 3.3 – Расходы на материалы до разработки подсистемы

Материалы	Ед. измер.	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	пачки	3	180	540
Канцтовары	-		-	150
Картридж	шт.	1	300	300
Итого				760

Расходы на материалы за девять месяцев составляют:

$$P_{\text{м}} = (540 + 150) * 9 + 300 * 3 = 7200 \text{ руб.}$$

Поскольку $P_{\text{общ}} + P_{\text{пр}} = 4554 + 7200 = 11754.00 \text{ руб.}$, то

$$P_{\text{пр}} = 0.60 * 11754.00 = 7052.40 \text{ руб.},$$

Следовательно, общепроизводственные расходы будут равны:

$$P_{\text{общ}} = 11754.00 + 7052.40 = 18806.40 \text{ руб.}$$

Отсюда следует, что эксплуатационные расходы на выполнение задачи до разработки автоматизированной подсистемы составляют:

$$P_0 = 105300.00 + 26325.00 + 18806.40 = 150431.40 \text{ руб.}$$

Далее необходимо произвести расчет затрат после разработки автоматизированной подсистемы.

Разработкой занимается программист в течении семи месяцев, при этом он получает заработную плату в размере 12000 рублей каждый месяц, также в течении двух месяцев разработки системный администратор работает над администрированием подсистемы, созданием учетных записей и разграничением доступа к данным, хранящимся в базе.

Таким образом, капитальные затраты на разработку подсистемы равны:

$$K_{\text{пр}} = 12000.00 * 7 * 0.30 + 11000.00 * 2 * 0.30 = 31800.00 \text{ рублей}$$

Расходы на используемые в работе материалы после разработки автоматизированной подсистемы представлены в таблице 3.4.

Таблица 3.4 – Расходы на материалы после разработки подсистемы

Материалы	Ед. измер.	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	пачки	2	180	360
Канцтовары	-		-	100
Итого				460

Так как после внедрения в работу автоматизированной подсистемы использование компьютера увеличится, отсюда увеличатся и расходы на электроэнергию и будут равны 250 кВт/ч.

$$P_{\text{эн}} = 250 * 9 * 2.53 = 5692.50 \text{ руб.}$$

Расходы на материалы за девять месяцев составят:

$$P_{\text{м}} = 460 * 9 = 4140.00 \text{ руб.}$$

$$P_{\text{общ}} + P_{\text{пр}} = 5692.50 + 4140.00 = 9832.50 \text{ руб.}$$

$$P_{\text{пр}} = 0,60 * 9832.50 = 5899.50 \text{ руб.}$$

$$P_{\text{общ}} = 9832.50 + 5899.50 = 15735.00 \text{ руб.}$$

После внедрения в медицинский центр «ОРМЕД» автоматизированной подсистемы в штате организации продолжили работать программист и системный администратор, таким образом, за следующие девять месяцев расходы составили:

$$P_{\text{осн}} = (12000 + 11000) * 9 * 0.30 = 62100.00 \text{ руб.}$$

$$P_{\text{ауп}} = 0.25 * 62100.00 = 15525.00 \text{ руб.}$$

Таким образом, эксплуатационные расходы на решение функциональных задачи после внедрения подсистемы составляют:

$$P_1 = 62100.00 + 15525.00 + 15732.00 = 93357.00 \text{ руб.}$$

Рассчитываем приведенные затраты:

$$З = 93357.00 + 0.25 * 31800.00 = 101307.00$$

Условный экономический эффект:

$$150431.40 - 93357.00 = 57074.40 \text{ руб.}$$

Срок окупаемости:

$$T_p = 31800.00 / 57074.40 = 0,55$$

Расчетный коэффициент:

$$E_p = 57074.40 / 31800.00 = 1,79$$

Результаты итоговых расчетов представлены в таблице 3.5.

Таблица 3.5 – Результаты расчетов

№ п/п	Наименование показателей	Условные обозначения	Значение показателей
	Приведенные затраты	руб.	101307.00
	Расходы до внедрения	руб.	150431.40
	Расходы после внедрения	руб.	93357.00
	Экономический эффект	руб.	57074.40
	Срок окупаемости	мес.	6
	Расчетный коэффициент	%	1,79

По результатам проведенных расчетов можно сделать вывод, что затраты после внедрения автоматизированной подсистемы в работу медицинского центра сократились. Также уменьшилось количество штатных сотрудников, в связи с этим снизились расходы, выделяемые на заработную плату сотрудникам. Получен условный экономический эффект в размере 57074.40 рублей. Срок окупаемости данной разработки составляет 6 месяцев. Расчетный коэффициент $E_n \leq E_p = 1.7 \leq 1.79$, что доказывает целесообразность разработки автоматизированной подсистемы и ее внедрение в работу организации.

ЗАКЛЮЧЕНИЕ

В первом разделе выпускной квалификационной работе был произведен анализ предметной области, построены диаграммы деятельности медицинского центра, на основе которых были выявлены недостатки и сформулированы цели и задачи работы по разработке автоматизированной подсистемы учета пациентов. Также были рассмотрены имеющиеся информационные системы, используемые в данной сфере, и проведена экспертная оценка данных систем.

Во втором разделе выпускной квалификационной работы было произведено обоснование проектных решений, построены диаграммы, отображающие работу медицинского центра после внедрения автоматизированной подсистемы, и представлена логическая модель базы данных, которая использовалась при разработке программного продукта.

В третьем разделе выпускной квалификационной работы была создана база данных, разработан и запрограммирован интерфейс автоматизированной подсистемы учета пациентов. Представлены расчеты экономической эффективности разработанной подсистемы.

В процессе выполнения выпускной квалификационной работе были решены следующие задачи:

- проанализирована деятельность медицинского центра и выявлены имеющиеся в его работе недостатки;
- спроектированы функциональные модели, описывающие работу организации, «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- сформированы цели и назначения будущего автоматизированного варианта решения поставленной задачи;
- проанализированы существующие разработки и выявлены их недостатки, применительно к работе медицинского центра;
- разработано и протестировано созданное прикладное решение;

– произведена оценка показателей эффективности разработки.

Разработанный программный продукт позволит существенно повысить эффективность работы медицинского центра в плане учета пациентов за счет автоматизации повторяющихся действий и сокращению возможных ошибок при работе с клиентскими данными. Экономический анализ, проведенный в последней главе выпускной квалификационной работе показал, что подсистема является экономически эффективной, так как чистая прибыль за год от использования разработанной подсистемы превышает затраты и уже за два месяца покрывает ее себестоимость. В будущем предполагается дальнейшее развитие и совершенствование разработанной автоматизированной подсистемы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «АСУЛОН М – Аптека плюс ЛПУ» [Электронный ресурс] – Режим доступа: <http://www.miacso.ru/index.php/materialy/programmnoe-obespechenie/>, свободный.
2. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем [Текст] / А.М. Вендров. – Москва: Вильямс, 2008. – 298 с.
3. Гахов, Р.П. Компьютерное моделирование экономических процессов [Текст]: учебное пособие для студентов вузов / Р.П. Гахов, Н.В. Щербинина. - Белгород: ИД Белгород, 2014. - 88 с.
4. Гахова, Н.Н. Инструментальные средства информационных систем [Электронный ресурс] / Н.Н. Гахова - Белгород: НИУ БелГУ, 2012. - Режим доступа: <http://pegas.bsu.edu.ru/course/view.php?id=5188>, свободный.
5. Груббер, М. Введение в язык SQL [Текст] / М. Груббер. – Москва: ЛОРИ, 2012. – 644 с.
6. Гультияев, А.К. Проектирование и дизайн пользовательского интерфейса [Текст] / А.К. Гультияев, В.А. Машин. – СПб.: КОРОНАпринт, 2015. – 349 с.
7. Золотов, С.Ю. Проектирование информационных систем [Текст] / С.Ю. Золотов. – Томск: Эль Контент, 2013. - 88 с.
8. Илюшечкин, В.М. Основы использования и проектирования баз данных [Текст] / В.М. Илюшечкин. – Москва: Юрайт, 2017. – 214 с.
9. Купер, А. Интерфейс. Основы проектирования [Текст] / А. Купер, Р. Рейман, Д. Кронин, К. Носсел. – Санкт-Петербург: Питер, 2012. – 289 с.
10. Иифоклиника.RU [Электронный ресурс] – Режим доступа: <https://infoclinica.ru/>, свободный.
11. Коваленко, В.В. Проектирование информационных систем [Текст] / В.В. Коваленко. – Москва: Форум, 2012. – 320 с.

12. Култыгин, О.П. Администрирование баз данных СУБД MS SQL Server [Текст] / О.П. Култыгин. – Москва: Московская финансово-промышленная академия, 2012. – 232 с.
13. Маклаков, С.В. Моделирование бизнес-процессов с BPwin 4.0 [Текст] / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2012. – 224 с.
14. Маторин, С.И. Теория систем и системный анализ [Текст] / С.И. Маторин, О.А. Зимовец. - Белгород: ИД Белгород, 2012. – 154 с.
15. МЕДИАЛОГ [Электронный ресурс] – Режим доступа: <http://www.kampas.ru/news/news-kampas/410/>, свободный.
16. Медицинская информационная система AKSi-клиника [Электронный ресурс] – Режим доступа: <http://medbuy.ru/adverts/77>, свободный.
17. Медицинская информационная система МИС - РИСТАР [Электронный ресурс] – Режим доступа: <http://ristar.ru/>, свободный.
18. Медицинская информационная система MEDEXIS («МЕДСОФТ - Клиника») [Электронный ресурс] – Режим доступа: <http://www.medsoft.ru/index-2-med.html>, свободный.
19. Медицинские информационные системы [Электронный ресурс] – Режим доступа: <http://itm.consef.ru/main.mhtml?Part=84>, свободный
20. Медицинская организация [Электронный ресурс] – Режим доступа: <http://www.opencentre.ru/terminologiya-osnovnye-ponyatiya/>, свободный.
21. Медицинский центра «ОРМЕД» [Электронный ресурс] – Режим доступа: <http://www.ormed31.ru/>, свободный.
22. Михелев, В.М. Базы данных и СУБД [Текст] / В.М. Михелев. – Белгород: Издательство БелГУ, 2007. – 200 с.
23. О работе с InterBase/Firebird. [Электронный ресурс] - Режим доступа: <http://www.ibase.ru/devinfo/ibrusfaq.htm>, свободный.
24. Построение инфологической модели [Электронный ресурс] - Режим доступа: <http://citforum.ru/database/dbguide/5-2.shtml>, свободный.

25. Права и обязанности медицинских организаций [Электронный ресурс] - Режим доступа: <http://pravo.kormed.ru/prava-pacienta/prava-i-obyazannosti-medicinskoj-organizacii/prava-i-obyazannosti-medicinskih-organizacij/>, свободный.
26. Румбешт, В.В. Программирование информационных систем: Учебно-методическое пособие по выполнению лабораторных работ [Текст] / В.В. Румбешт, Г.Г. Банчук. – Белгород: Кооперативное образование, 2006. – 165 с.
27. Страуструп, Б. Язык программирования С++. Стандарт С++ [Текст] / Б. Страуструп. – Москва: Бинوم, 2017. – 176 с.
28. Стружкин, Н.П. Базы данных: Проектирование [Текст] / Н.П.Стружкин, В.В. Годин. – Москва: Юрайт, 2017. – 292 с.
29. «ТрансМед:Здоровье»[Электронный ресурс] – Режим доступа: <http://www.trustmed.ru/>, свободный.
30. Чистов, Д.В. Проектирование информационных систем. Учебник и практикум [Текст] / Д.В. Чистов, П.В. Мельников. – Москва: Юрайт, 2017, – 216 с.
31. Шугарева В.Г. Программное обеспечение и операционные системы ПК / В.Г. Шугарева. – Ростов-на-Дону: Феникс, 2013. -382 с.
32. Clinic365 [Электронный ресурс] – Режим доступа: <http://www.clinic365.ru/features/>, свободный
33. Firebird. Руководство разработчика баз данных. [Электронный ресурс] - Режим доступа: <http://coollib.com/b/240145/read>, свободный.
34. MedWork [Электронный ресурс] – Режим доступа: <https://www.medwork.ru/>, свободный.
35. 1С:Медицина. Поликлиника [Электронный ресурс] – Режим доступа: <http://www.pulsar.ru/medical/>, свободный.

ПРИЛОЖЕНИЕ А

Экспертная оценка информационных систем

Таблица А.1 – Экспертная оценка системы «MedSoft»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	12,5
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	20	10
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	0
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений		Наличие разграничений	15
<i>Значение</i>	0		15	
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех обновлений	10
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\sum P_i^+(s)$ в %				67,5
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита		Защита отсутствует	0
<i>Значение</i>	0		42	
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\sum P_j^-(s)$ в %				28
Общая суммарная оценка эффективности использования Информационного ресурса $\Theta = \sum P_i^+(s) - \sum P_j^-(s)$ в %				39,5

Таблица А.2 – Экспертная оценка системы «М-АПТЕКА ЛПУ АПТЕКА ЛПУ»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	20
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	10
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	0
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений	Наличие разграничений		0
<i>Значение</i>	0	15		
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	5
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				50
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита	Защита отсутствует		0
<i>Значение</i>	0	42		
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				28
Общая суммарная оценка эффективности использования Информационного ресурса $\Theta = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				22

Таблица А.3 – Экспертная оценка системы «ТрантМед:Здоровье»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	12,5
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	0
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	7,5
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений	Наличие разграничений		15
<i>Значение</i>	0	15		
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	10
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				55
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита	Защита отсутствует		0
<i>Значение</i>	0	42		
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				56
Общая суммарная оценка эффективности использования Информационного ресурса $\Theta = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				27

Таблица А.4 – Экспертная оценка системы «МЕДИАЛОГ»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	25
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	20
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	10
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	7,5
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	2,5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений		Наличие разграничений	15
<i>Значение</i>	0		15	
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	10
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				90
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита		Защита отсутствует	0
<i>Значение</i>	0		42	
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	56
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				56
Общая суммарная оценка эффективности использования Информационного ресурса $\mathcal{E} = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				34

Таблица А.5 – Экспертная оценка системы «MedWork»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	12,5
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	10
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	7,5
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений	Наличие разграничений		15
<i>Значение</i>	0	15		
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	5
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				60
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита	Защита отсутствует		0
<i>Значение</i>	0	42		
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				28
Общая суммарная оценка эффективности использования Информационного ресурса $\mathcal{E} = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				32

Таблица А.6 – Экспертная оценка системы «1С:Медицина. Поликлиника»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	25
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	20
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	10
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	15
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений		Наличие разграничений	15
<i>Значение</i>	0		15	
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	10
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				72
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита		Защита отсутствует	0
<i>Значение</i>	0		42	
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				28
Общая суммарная оценка эффективности использования Информационного ресурса $\mathcal{E} = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				44

Таблица А.7 – Экспертная оценка системы «Clinic365»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	12,5
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	10
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	0
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	2,5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений		Наличие разграничений	15
<i>Значение</i>	0		15	
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	5
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				50
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита		Защита отсутствует	42
<i>Значение</i>	0		42	
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				70
Общая суммарная оценка эффективности использования Информационного ресурса $\mathcal{E} = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				-20

Таблица А.8 – Экспертная оценка системы «Иифоклиника»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	12,5
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	0
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	7,5
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений		Наличие разграничений	15
<i>Значение</i>	0		15	
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	0
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				45
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита		Защита отсутствует	42
<i>Значение</i>	0		42	
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	0
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				42
Общая суммарная оценка эффективности использования Информационного ресурса $\Theta = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				3

Таблица А.9 – Экспертная оценка системы «AKSi-клиника»

<i>Признак</i>	Функциональность использования			Оценка, %
<i>Градация</i>	Неудобна	Частично удобна	Удобна	0
<i>Значение</i>	0	12,5	25	
<i>Признак</i>	Интеграция			Оценка, %
<i>Градация</i>	Система неинтегрируемая	Низкая степень интеграции	Высокая степень интеграции	10
<i>Значение</i>	0	10	20	
<i>Признак</i>	Формы отчетности			Оценка, %
<i>Градация</i>	Не имеет отчетности	Наличие 1-2 форм отчетности	Наличие более 2 форм отчетности	5
<i>Значение</i>	0	5	10	
<i>Признак</i>	Гибкость системы			Оценка, %
<i>Градация</i>	Не адаптируется	Частично адаптируется	Адаптируется	0
<i>Значение</i>	0	7,5	15	
<i>Признак</i>	Интерфейс			Оценка, %
<i>Градация</i>	непонятен	Частично понятен	Полностью понятен	2,5
<i>Значение</i>	0	2,5	5	
<i>Признак</i>	Разграничение прав			Оценка, %
<i>Градация</i>	Отсутствие разграничений		Наличие разграничений	0
<i>Значение</i>	0		15	
<i>Признак</i>	Автоматические обновления			Оценка, %
<i>Градация</i>	Обновления отсутствуют	Обновления выполняются не всегда	Наличие всех автоматических обновлений	5
<i>Значение</i>	0	5	10	
Общая суммарная оценка позитивных свойств $\Sigma P^+_i(s)$ в %				22,5
<i>Признак</i>	Защита системы			Оценка, %
<i>Градация</i>	Имеется защита		Защита отсутствует	0
<i>Значение</i>	0		42	
<i>Признак</i>	Стоимость			Оценка, %
<i>Градация</i>	Бесплатная	Средняя стоимость	Высокая стоимость	28
<i>Значение</i>	0	28	56	
Общая суммарная оценка негативных свойств $\Sigma P^-_j(s)$ в %				28
Общая суммарная оценка эффективности использования Информационного ресурса $\Theta = \Sigma P^+_i(s) - \Sigma P^-_j(s)$ в %				-5,5

ПРИЛОЖЕНИЕ Б

Функциональные диаграммы деятельности медицинского центра «КАК ДОЛЖНО БЫТЬ»

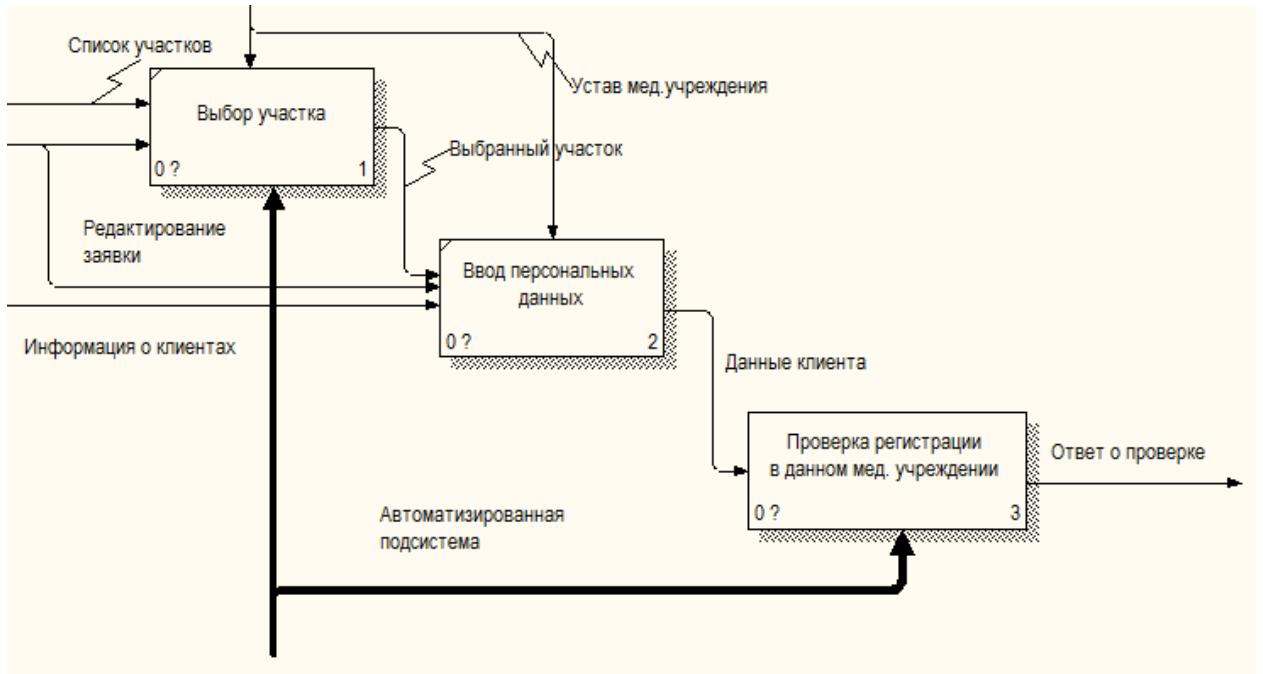


Рисунок Б.1 – Детализация блока «Ввод данных о пациенте»

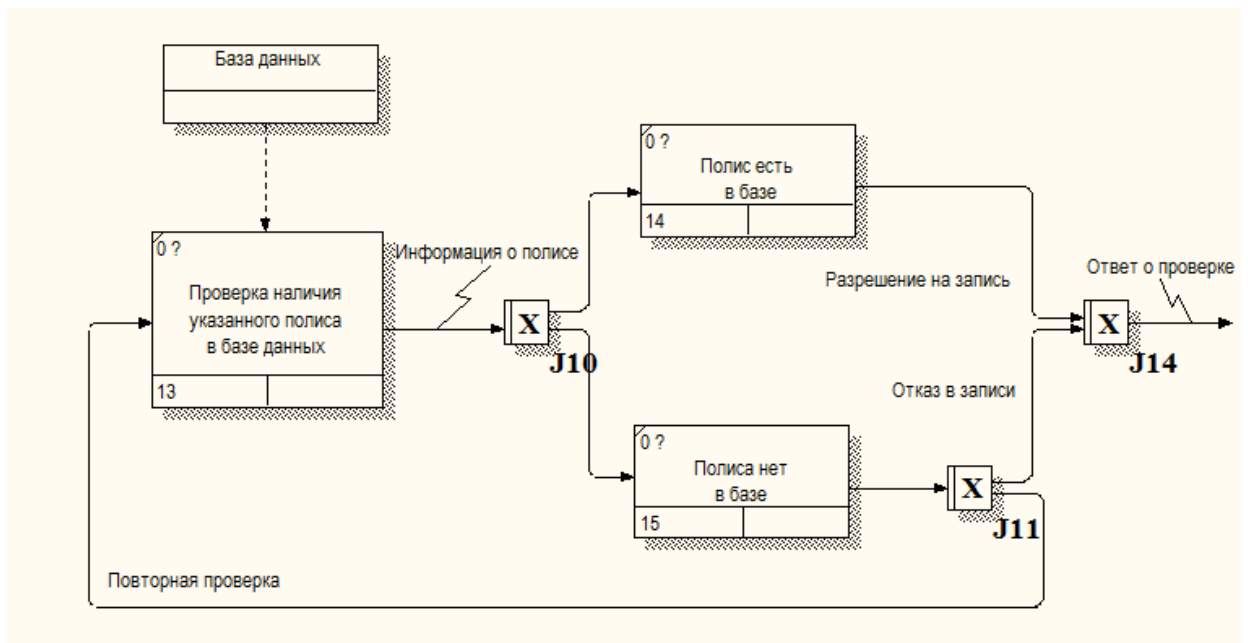


Рисунок Б.2 – Детализация блока «Проверка наличия клиента в базе»

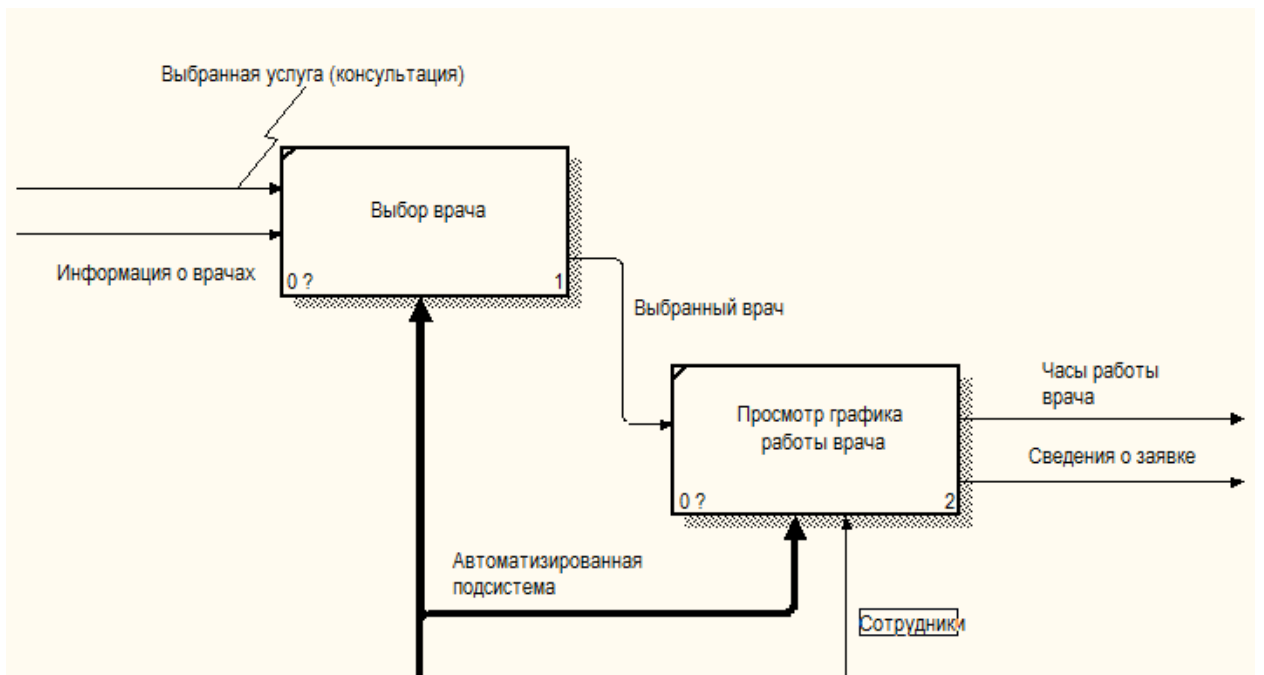


Рисунок Б.3 – Детализация блока «Выбор врача»

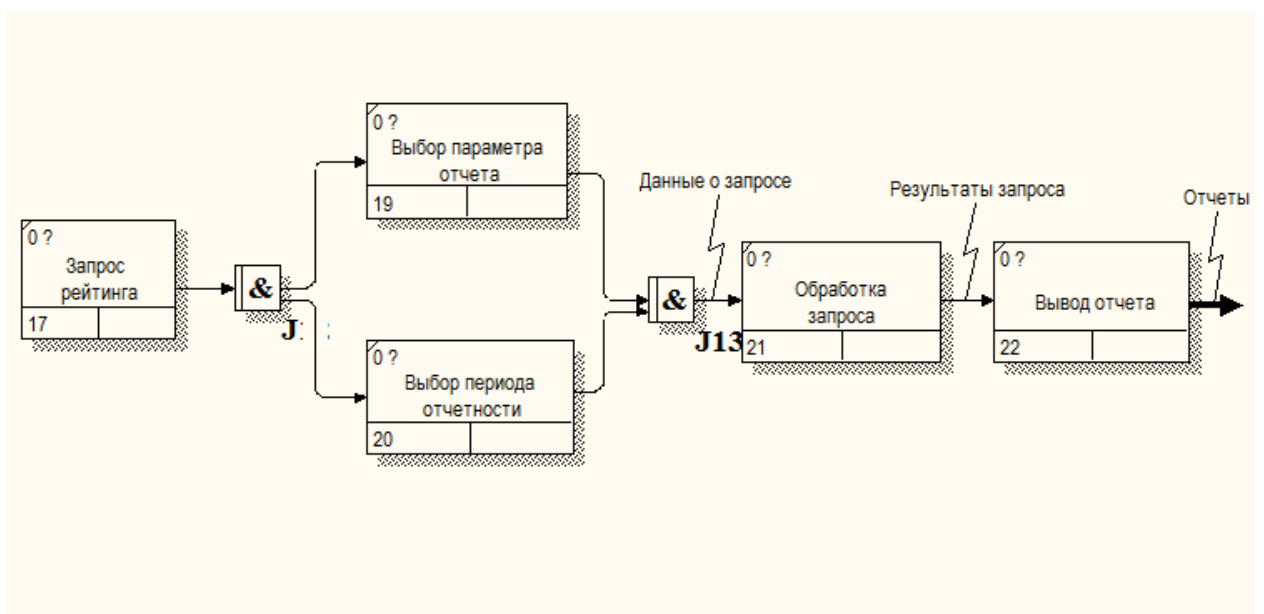


Рисунок Б.4 – Детализация блока «Формирование отчетности (рейтинга)»

ПРИЛОЖЕНИЕ В

Разработка базы данных автоматизированной подсистемы

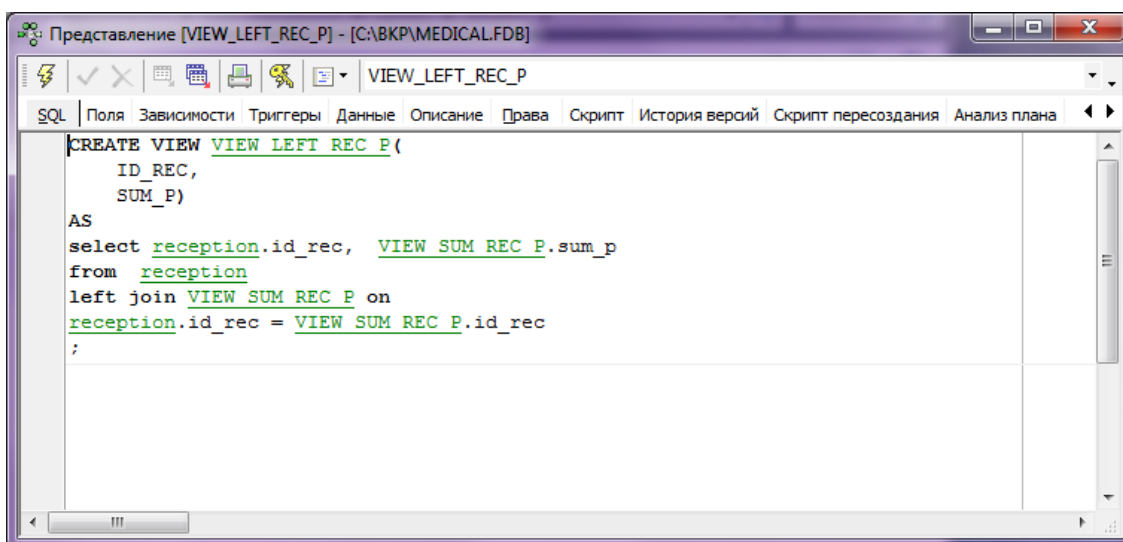


Рисунок В.1 – Создание представления на подсчет суммарной стоимости заявки

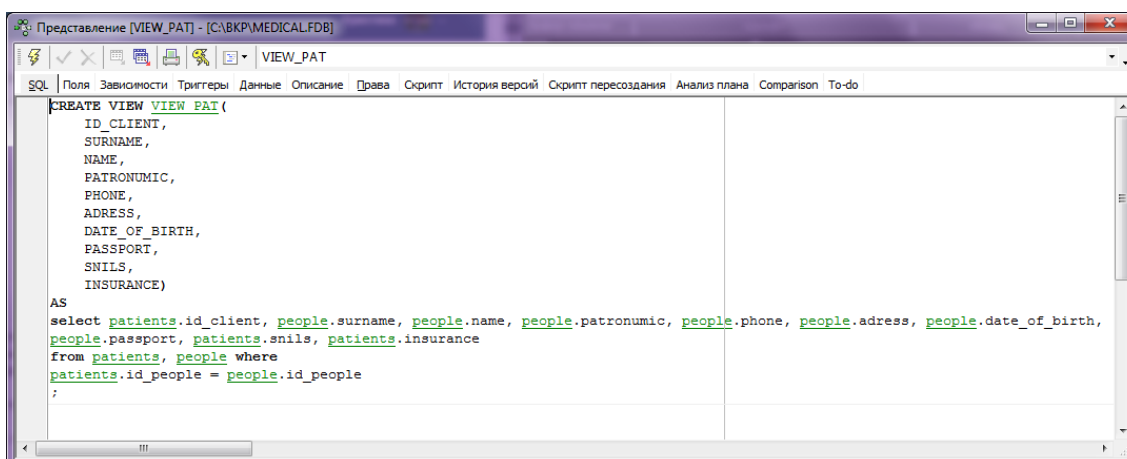


Рисунок В.2 – Создание представления на вывод всех данных о пациенте

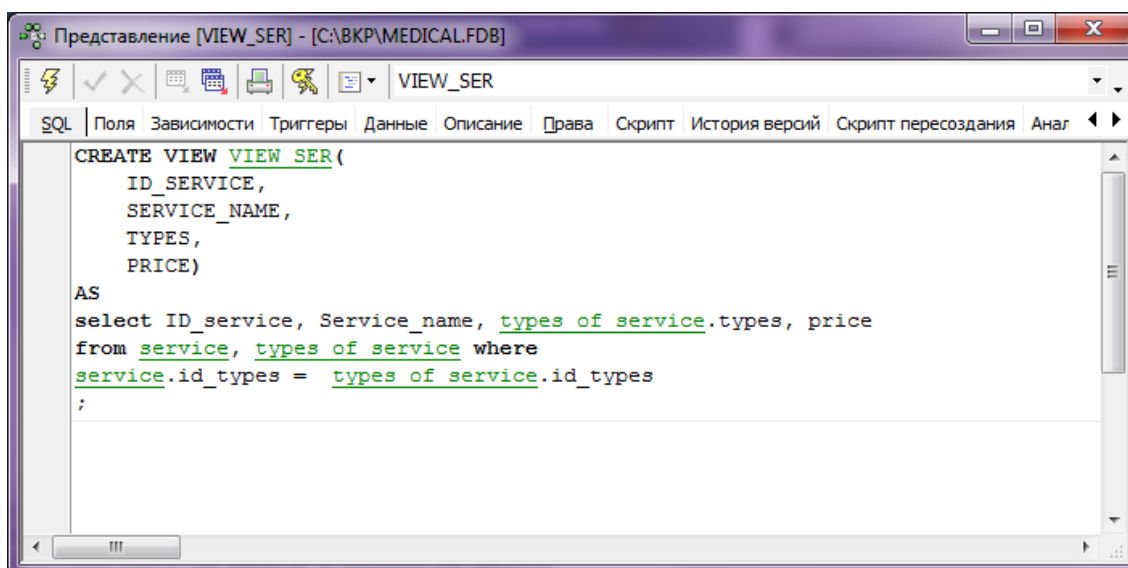


Рисунок В.3 – Создание представления на вывод информации по услугам

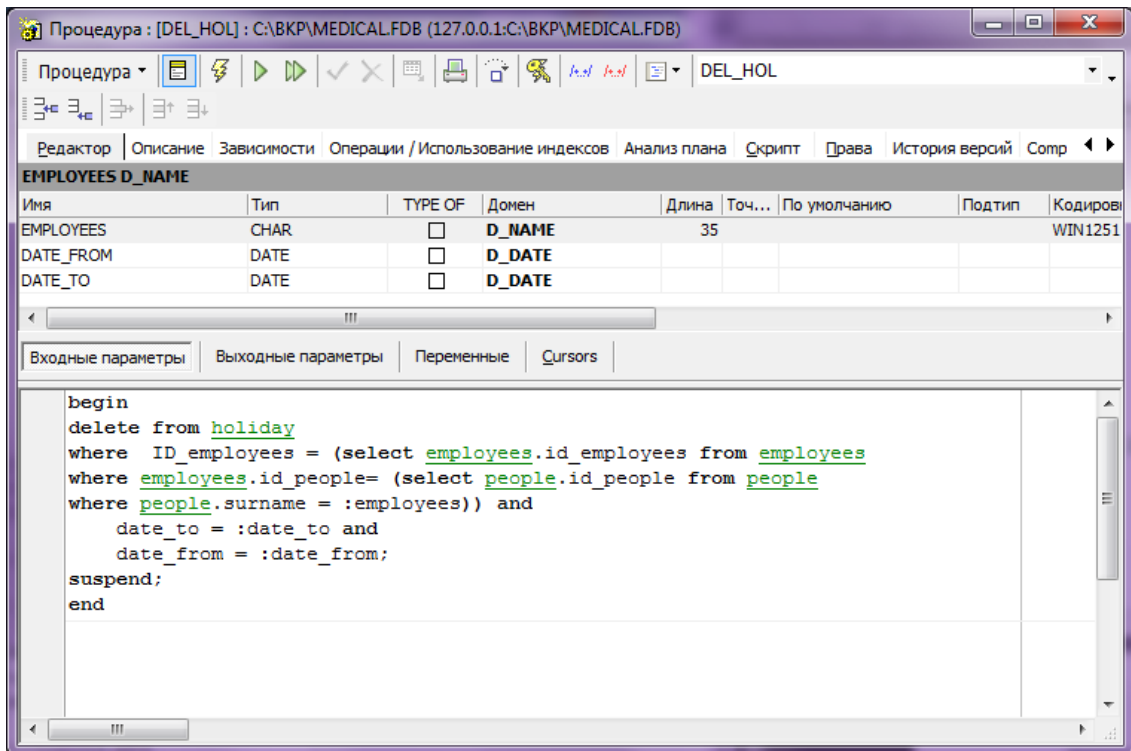


Рисунок В.4 – Создание процедуры удаления данных из таблицы отпусков

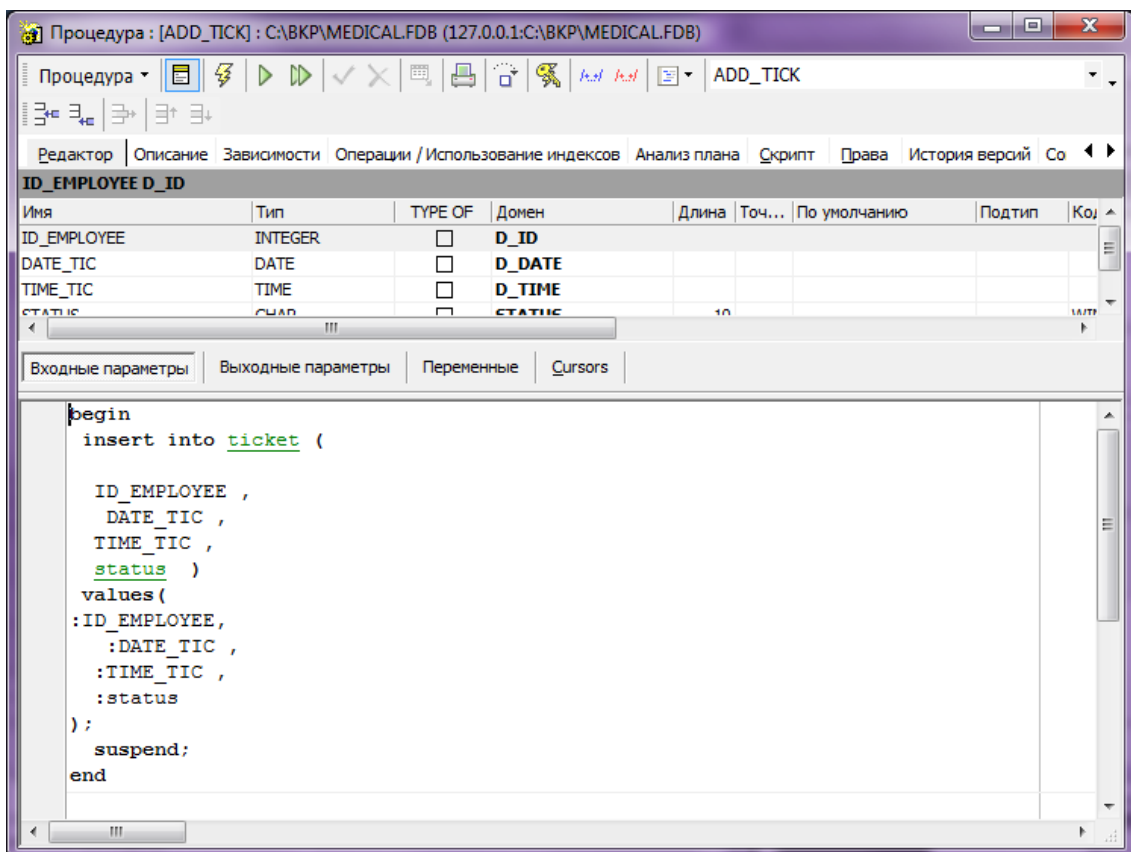


Рисунок В.5 – процедура добавления данных в таблицы «Талоны»

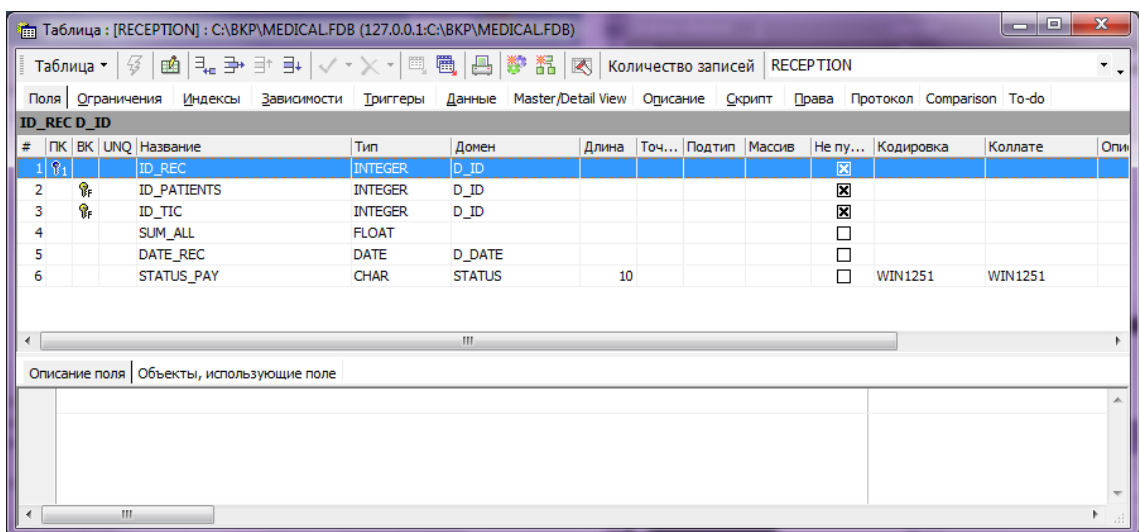
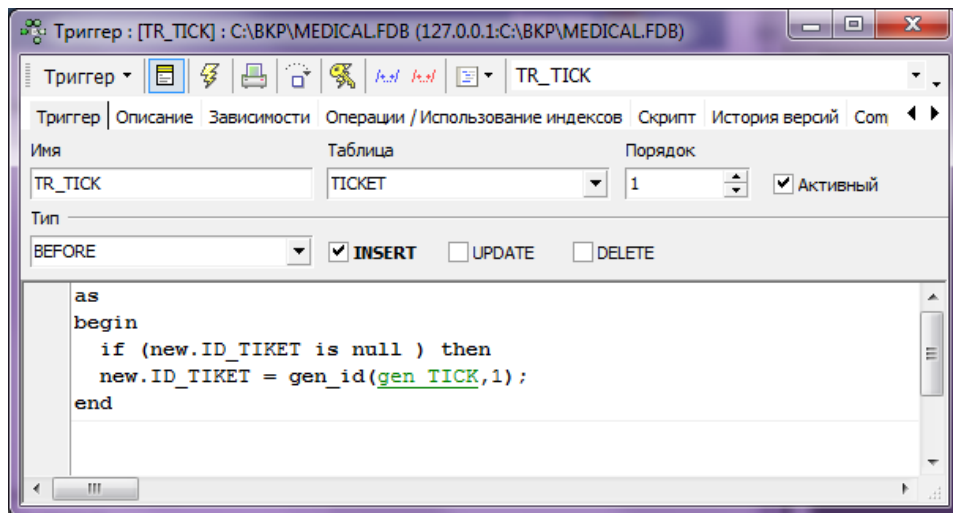
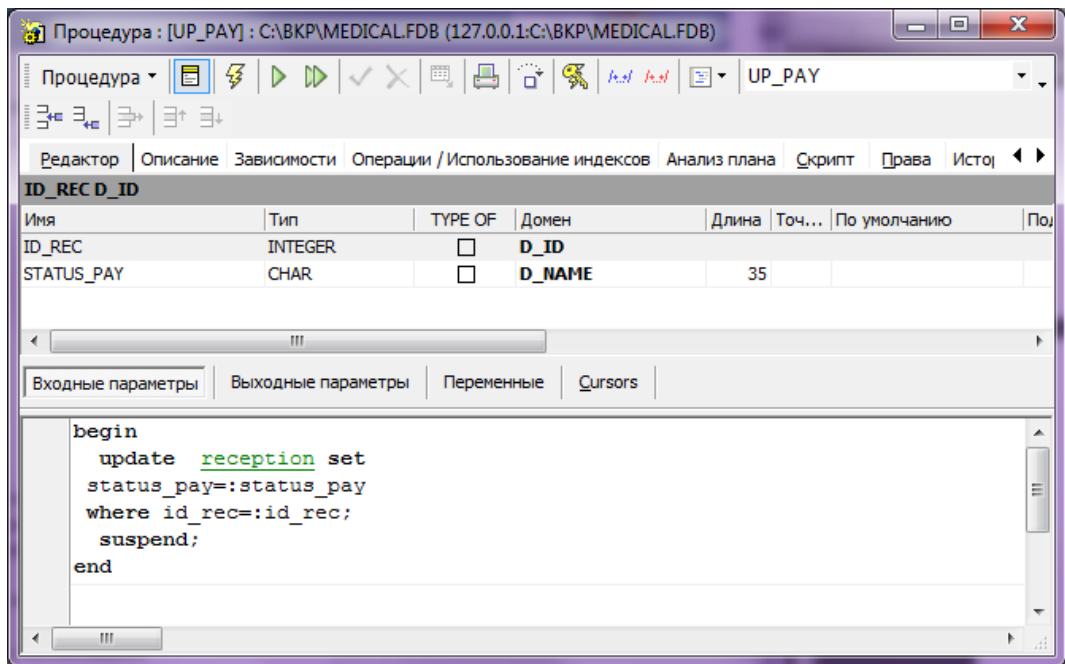


Рисунок В.8 – Созданная таблица «Услуги»

ПРИЛОЖЕНИЕ Г

Программный код разработки

Программный код файла Project1.cpp:

```
#include <vcl.h>
#pragma hdrstop
//-----
USEFORM("Unit1.cpp", Form1);
USEFORM("Unit2.cpp", Form2);
USEFORM("Unit3.cpp", Form3);
USEFORM("Unit4.cpp", Form4);
USEFORM("Unit5.cpp", Form5);
USEFORM("Unit6.cpp", Form6);
USEFORM("Unit7.cpp", Form7);
USEFORM("Unit8.cpp", Form8);
USEFORM("Unit9.cpp", DataModule9); /* TDataModule: File Type */
USEFORM("Unit10.cpp", Form10);
USEFORM("Unit11.cpp", Form11);
USEFORM("Unit12.cpp", Form12);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    try
    {
        Application->Initialize();
        Application->CreateForm(__classid(TForm3), &Form3);
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TForm2), &Form2);
        Application->CreateForm(__classid(TForm4), &Form4);
        Application->CreateForm(__classid(TForm5), &Form5);
        Application->CreateForm(__classid(TForm6), &Form6);
        Application->CreateForm(__classid(TForm7), &Form7);
        Application->CreateForm(__classid(TForm8), &Form8);
        Application->CreateForm(__classid(TDataModule9), &DataModule9);
        Application->CreateForm(__classid(TForm10), &Form10);
        Application->CreateForm(__classid(TForm11), &Form11);
        Application->CreateForm(__classid(TForm12), &Form12);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
        catch (Exception &exception)
        {
            Application->ShowException(&exception);
        }
    }
    return 0 }
}
```

Программный код Unit1.cpp:

```
#include <vcl.h>
#pragma hdrstop
#include "File1.h"
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
```

```

#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
#include "Unit10.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
int flag;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
Form2->Show();
Form2->TabSheet1->Show();
if ((flag == 1) || (flag == 5)){
    Form2->N2->Visible=true;
    Form2->N3->Visible=true;
    Form2->N4->Visible=true;
    Form2->N5->Visible=true;
    Form2->N6->Visible=true;
    Form2->N7->Visible=true;
    Form2->N8->Visible=true;
    Form2->N9->Visible=true;
    Form2->N10->Visible=true;
    Form2->N11->Visible=true;
    Form2->N12->Visible=true;
    Form2->N13->Visible=true;
}
Form1->Visible = false;
}
//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
Form3->Close();
}
//-----

void __fastcall TForm1::N1Click(TObject *Sender)
{
Form1->Visible = false;
Form3->Show();
}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
Form4->Show();
if ((flag == 1) || (flag == 5)){
    Form4->N3->Visible=true;
    Form4->N4->Visible=true;
    Form4->N5->Visible=true;
    Form4->N6->Visible=true;
    Form4->N7->Visible=true;
    Form4->N8->Visible=true;
    Form4->N9->Visible=true;
}
}

```

```

        Form4->N10->Visible=true;
        Form4->N11->Visible=true;
        Form4->N12->Visible=true;
        Form4->N13->Visible=true;
    } else if ((flag == 2) || (flag == 4)){
    Form4->N1->Visible=false;
    Form4->N12->Visible=false;
    }
    Form1->Visible = false;
    }
    //-----
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
    if ((flag == 2) || (flag == 4)){
    Form5->N1->Visible=false;
    Form5->N12->Visible=false;
    }
    Form5->Show();
    Form5->TabSheet2->Show();
    Form1->Visible = false;
    }
    //-----
void __fastcall TForm1::BitBtn4Click(TObject *Sender)
{
    if ((flag == 2) || (flag == 4)){
    Form5->N1->Visible=false;
    Form5->N12->Visible=false;
    }
    Form5->Show();
    Form5->TabSheet1->Show();
    Form1->Visible = false;
    }
void __fastcall TForm1::BitBtn5Click(TObject *Sender)
{
    if ((flag == 2) || (flag == 4)){
    Form6->N1->Visible=false;
    Form6->N12->Visible=false;
    }
    Form6->Show();
    Form6->TabSheet1->Show();
    Form1->Visible = false;
    }

void __fastcall TForm1::BitBtn6Click(TObject *Sender)
{
    if ((flag == 2) || (flag == 4)){
    Form7->N1->Visible=false;
    Form7->N12->Visible=false;
    }
    Form7->Show();
    Form7->TabSheet1->Show();
    Form1->Visible = false;
    }
    //-----
void __fastcall TForm1::BitBtn7Click(TObject *Sender)
{
    if ((flag == 2) || (flag == 4)){
    Form7->N1->Visible=false;
    Form7->N12->Visible=false;
    }
    Form7->Show();
    Form7->TabSheet2->Show();
    Form1->Visible = false;

```

```

}
//-----
void __fastcall TForm1::BitBtn8Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form8->N1->Visible=false;
}else if ( flag == 3){
Form8->N1->Visible=false;
Form8->N2->Visible=false;
Form8->N3->Visible=false;
    Form8->N4->Visible=false;
    Form8->N5->Visible=false;
    Form8->N6->Visible=false;
    Form8->N7->Visible=false;
    Form8->N8->Visible=false;
    Form8->N9->Visible=false;
    Form8->N10->Visible=false;
    Form8->N11->Visible=false;
    Form4->N13->Visible=true;
}
Form8->Show();
Form1->Visible = false;
}

```

Программный код Unit2.cpp:

```

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
#include "Unit10.h"
#include "Unit12.h"
#include "File1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
void __fastcall TForm2::FormCreate(TObject *Sender)
{
//DataModule9->DataSource1->DataSet->Delete();
switch (flag)
{
    case 2:
        Form2->N13->Visible = false;
        break;
    //case 3:
    //case 4:
    //case 5:
}
DateTimePicker1->Date=Now().DateString();
}
//-----
void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
if ((Form2->Edit1->Text != "" ) && (Form2->Edit2->Text != "" ) && (Form2->Edit3->Text != "" ) && (Form2->Edit4->Text != "" ) && (Form2->Edit16->Text != "" ) && (Form2->Edit6->Text != "" )){
DataModule9->IBStoredProc1->ParamByName ("SURNAME")->AsString=Edit1->Text;
}
}

```

```

DataModule9->IBStoredProc1->ParamByName ("NAME")->AsString=Edit2->Text;
DataModule9->IBStoredProc1->ParamByName ("PATRONUMIC")->AsString=Edit3->Text;
DataModule9->IBStoredProc1->ParamByName ("PHONE")->AsString=Edit16->Text;
DataModule9->IBStoredProc1->ParamByName ("ADDRESS")->AsString=Edit4->Text;
DataModule9->IBStoredProc1->ParamByName ("DATE_OF_BIRTH")->AsString=DateTimePicker1->Date;
DataModule9->IBStoredProc1->ParamByName ("PASSPORT")->AsString=Edit6->Text;
DataModule9->IBStoredProc1->Prepare();
DataModule9->IBStoredProc1->ExecProc();
DataModule9->IBStoredProc2->ParamByName ("ID_POST")->AsString=DBEdit1->Text;
DataModule9->IBStoredProc2->Prepare();
DataModule9->IBStoredProc2->ExecProc();
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->Open();
}else{
    ShowMessage("Заполните все поля!");
}
}
void __fastcall TForm2::N2Click(TObject *Sender)
{
Form4->Show();
Form2->Visible = false;
}
//-----
void __fastcall TForm2::N4Click(TObject *Sender)
{
Form2->Visible = false;
Form5->Show();
Form5->TabSheet1->Show();
}
//-----
void __fastcall TForm2::N5Click(TObject *Sender)
{
Form2->Visible = false;
Form5->Show();
Form5->TabSheet2->Show();
}
//-----
void __fastcall TForm2::N7Click(TObject *Sender)
{
Form2->Visible = false;
Form6->Show();
Form6->TabSheet1->Show();
}
//-----
void __fastcall TForm2::N8Click(TObject *Sender)
{
Form2->Visible = false;
Form6->Show();
Form6->TabSheet2->Show();
}
//-----
void __fastcall TForm2::N12Click(TObject *Sender)
{
Form2->Visible = false;
Form8->Show();
}
//-----

void __fastcall TForm2::DBEdit1Click(TObject *Sender)
{
Form10->Show();
Form10->TabSheet2->Show();
}
//-----

```

```

void __fastcall TForm2::N16Click(TObject *Sender)
{
Form2->Visible = false;
Form7->Show();
Form7->TabSheet1->Show();
}
//-----
void __fastcall TForm2::N10Click(TObject *Sender)
{
Form2->Visible = false;
Form7->Show();
Form7->TabSheet2->Show();
}
//-----
void __fastcall TForm2::N11Click(TObject *Sender)
{
Form2->Visible = false;
Form7->Show();
Form7->TabSheet3->Show();
}
//-----
void __fastcall TForm2::ComboBox1Click(TObject *Sender)
{
Form2->ComboBox2->Text="Поле";
Edit7->Clear();
}
//-----
void __fastcall TForm2::ComboBox2Click(TObject *Sender)
{
Form2->ComboBox1->Text="По полю";
}
//-----
void __fastcall TForm2::Button2Click(TObject *Sender)
{
Edit1->Clear();
Edit2->Clear();
Edit3->Clear();
Edit16->Clear();
Edit4->Clear();
Edit6->Clear();
}
//-----
void __fastcall TForm2::Button1Click(TObject *Sender)
{
Edit7->Clear();
}
//-----
void __fastcall TForm2::ComboBox3Click(TObject *Sender)
{
Form2->ComboBox4->Text="Поле";
Edit14->Clear();
}
//-----
void __fastcall TForm2::ComboBox4Click(TObject *Sender)
{
Form2->ComboBox3->Text="По полю";
}
//-----
void __fastcall TForm2::Button3Click(TObject *Sender)
{
Edit8->Clear();
}

```

```

//-----
void __fastcall TForm2::Button4Click(TObject *Sender)
{
Edit14->Clear();
}
//-----
void __fastcall TForm2::BitBtn8Click(TObject *Sender)
{
AnsiString d;
switch (RadioGroup1->ItemIndex)
{case 0: d = " "; break;
case 1: d = " DESC";
}
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->SQL->Clear();
DataModule9->IBQuery1->SQL->Add("select * from view_emp");
switch (ComboBox1->ItemIndex)
{case 0: d= "ORDER BY ID_EMPLOYEES"+ d; break;
case 1: d= "ORDER BY SURNAME"+ d; break;
case 2: d= "ORDER BY NAME"+ d; break;
case 3: d= "ORDER BY PATRONUMIC"+ d; break;
case 5: d= "ORDER BY DATE_OF_BIRTH"+ d; break;
case 4: d= "ORDER BY POST_NAME"+ d; break;
}
DataModule9->IBQuery1->SQL->Add(d);
DataModule9->IBQuery1->Open();
}
//-----
void __fastcall TForm2::BitBtn9Click(TObject *Sender)
{
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->SQL->Clear();
switch (ComboBox2->ItemIndex)
{
case 0:
DataModule9->IBQuery1->SQL->Add("select * from view_emp where ID_EMPLOYEES like '%" + Edit7-
>Text + "%' ");
break;
case 1:
DataModule9->IBQuery1->SQL->Add("select * from view_emp where SURNAME like '%" + Edit7->Text +
"%' ");
break;
case 2:
DataModule9->IBQuery1->SQL->Add("select * from view_emp where NAME like '%" + Edit7->Text + "%'
");
break;
case 3:
DataModule9->IBQuery1->SQL->Add("select * from view_emp where PATRONUMIC like '%" + Edit7-
>Text + "%' ");
break;
case 5:
DataModule9->IBQuery1->SQL->Add("select * from view_emp where DATE_OF_BIRTH like '%" + Edit7-
>Text + "%' ");
break;
case 4:
DataModule9->IBQuery1->SQL->Add("select * from view_emp where POST_NAME like '%" + Edit7->Text
+ "%' ");
break;
}
DataModule9->IBQuery1->Open();
if ((DataModule9->IBQuery1->RecordCount==0) || (Form2->Edit7->Text == ""))
{
ShowMessage("Не найдено!");
}
}

```

```

    }
}
//-----
void __fastcall TForm2::BitBtn5Click(TObject *Sender)
{
    DataModule9->IBQuery1->Close();
    DataModule9->IBQuery1->SQL->Clear();
    DataModule9->IBQuery1->SQL->Add("select * from view_emp");
    DataModule9->IBQuery1->Open();
    Form2->ComboBox2->Text="Поле";
    Edit7->Clear();
    Form2->ComboBox1->Text="По полю";
    Label5->Font->Color=clWindowText;
    DBEdit1->Font->Color=clWindowText;

}
//-----
void __fastcall TForm2::BitBtn17Click(TObject *Sender)
{
    AnsiString d;
    switch (RadioGroup2->ItemIndex)
    {case 0: d = " "; break;
    case 1: d = " DESC";
    }
    DataModule9->IBQuery2->Close();
    DataModule9->IBQuery2->SQL->Clear();
    DataModule9->IBQuery2->SQL->Add("select * from post");
    switch (ComboBox3->ItemIndex)
    {case 0: d= "ORDER BY ID_post"+ d; break;
    case 1: d= "ORDER BY name"+ d; break;
    }
    DataModule9->IBQuery2->SQL->Add(d);
    DataModule9->IBQuery2->Open();
}
//-----
void __fastcall TForm2::BitBtn18Click(TObject *Sender)
{
    DataModule9->IBQuery2->Close();
    DataModule9->IBQuery2->SQL->Clear();

    switch (ComboBox4->ItemIndex)
    {
    case 0:
        DataModule9->IBQuery2->SQL->Add("select * from post where ID_post like '%" + Edit14->Text + "%'");
        break;
    case 1:
        DataModule9->IBQuery2->SQL->Add("select * from post where NAME like '%" + Edit14->Text + "%'");
        break;
    }
    DataModule9->IBQuery2->Open();
    if ((DataModule9->IBQuery2->RecordCount==0) || (Form2->Edit14->Text == ""))
    {
        ShowMessage("Не найдено!");
    }
}
//-----

void __fastcall TForm2::BitBtn14Click(TObject *Sender)
{
    DataModule9->IBQuery2->Close();
    DataModule9->IBQuery2->SQL->Clear();
    DataModule9->IBQuery2->SQL->Add("select * from post");
    DataModule9->IBQuery2->Open();
}

```



```

Form2->ComboBox4->Text="Поле";
Edit14->Clear();
Form2->ComboBox3->Text="По полю";
}
//-----
void __fastcall TForm2::BitBtn10Click(TObject *Sender)
{
if (Form2->Edit8->Text != "" ) {
DataModule9->IBStoredProc3->ParamByName ("NAME")->AsString=Edit8->Text;
DataModule9->IBStoredProc3->Prepare();
DataModule9->IBStoredProc3->ExecProc();
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->Open();
}else{
    ShowMessage("Заполните все поля!");
}
}
//-----

void __fastcall TForm2::BitBtn2Click(TObject *Sender)
{
DataModule9->IBStoredProc12->ParamByName ("id_employees")->AsString=DBEdit2->Text;
DataModule9->IBStoredProc12->Prepare();
DataModule9->IBStoredProc12->ExecProc();
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->Open();
}
//-----

void __fastcall TForm2::BitBtn11Click(TObject *Sender)
{
DataModule9->IBStoredProc13->ParamByName ("id_post")->AsString=DBEdit3->Text;
DataModule9->IBStoredProc13->Prepare();
DataModule9->IBStoredProc13->ExecProc();
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->Open();
}
//-----

void __fastcall TForm2::BitBtn3Click(TObject *Sender)
{
ShowMessage("Внимание! Вы перешли в режим редактирования! Заполните обязательные поля, выделенные
КРАСНЫМ цветом. Для отмены нажмите кнопку ОБНОВИТЬ. ");
Label5->Font->Color=clRed;
DBEdit1->Font->Color=clRed;
}
//-----

void __fastcall TForm2::BitBtn4Click(TObject *Sender)
{
DataModule9->IBStoredProc22->ParamByName ("id_people")->AsString=DBEdit2->Text;
if (Edit1->Text != ""){
    DataModule9->IBStoredProc22->ParamByName ("SURNAME")->AsString=Edit1->Text;
}else{
    DataModule9->IBStoredProc22->ParamByName ("SURNAME")->AsString=DBEdit4->Text;
}
if (Edit2->Text != ""){
    DataModule9->IBStoredProc22->ParamByName ("NAME")->AsString=Edit2->Text;
}else{
    DataModule9->IBStoredProc22->ParamByName ("NAME")->AsString=DBEdit5->Text;
}
if (Edit3->Text != ""){
    DataModule9->IBStoredProc22->ParamByName ("PATRONUMIC")->AsString=Edit3->Text;
}else{
    DataModule9->IBStoredProc22->ParamByName ("PATRONUMIC")->AsString=DBEdit6->Text;
}
}
}

```

```

if (Edit16->Text != ""){
    DataModule9->IBStoredProc22->ParamByName ("PHONE")->AsString=Edit16->Text;
}else{
    DataModule9->IBStoredProc22->ParamByName ("PHONE")->AsString=DBEdit8->Text;
}
if (Edit4->Text != ""){
    DataModule9->IBStoredProc22->ParamByName ("ADRESS")->AsString=Edit4->Text;
}else{
    DataModule9->IBStoredProc22->ParamByName ("ADRESS")->AsString=DBEdit9->Text;
}
if (DateTimePicker1->Date != Now().DateString()){
    DataModule9->IBStoredProc22->ParamByName ("DATE_OF_BIRTH")->AsString=DateTimePicker1-
>Date;
}else{
    DataModule9->IBStoredProc22->ParamByName ("DATE_OF_BIRTH")->AsString=DBEdit10->Text;
}
if (Edit6->Text!=""){
    DataModule9->IBStoredProc22->ParamByName ("PASSPORT")->AsString=Edit6->Text;
}else{
    DataModule9->IBStoredProc22->ParamByName ("PASSPORT")->AsString=DBEdit11->Text;
}
DataModule9->IBStoredProc22->Prepare();
DataModule9->IBStoredProc22->ExecProc();
DataModule9->IBStoredProc23->ParamByName ("id_employees")->AsString=DBEdit2->Text;

DataModule9->IBStoredProc23->ParamByName ("post")->AsString=DBEdit1->Text;
DataModule9->IBStoredProc23->Prepare();
DataModule9->IBStoredProc23->ExecProc();
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->Open();
DateTimePicker1->Date = Now();
Label5->Font->Color=clWindowText;
DBEdit1->Font->Color=clWindowText;
}
//-----
void __fastcall TForm2::BitBtn12Click(TObject *Sender)
{
ShowMessage("Внимание! Вы перешли в режим редактирования! Для отмены нажмите кнопку
ОБНОВИТЬ.");
}
//-----
void __fastcall TForm2::BitBtn13Click(TObject *Sender)
{
DataModule9->IBStoredProc26->ParamByName ("id_post")->AsString=DBEdit3->Text;
if (Edit8->Text != ""){
    DataModule9->IBStoredProc26->ParamByName ("post")->AsString=Edit8->Text;
}else{
    DataModule9->IBStoredProc26->ParamByName ("post")->AsString=DBEdit12->Text;
}

DataModule9->IBStoredProc26->Prepare();
DataModule9->IBStoredProc26->ExecProc();
DataModule9->IBQuery2->Close();
DataModule9->IBQuery2->Open();
DataModule9->IBQuery1->Close();
DataModule9->IBQuery1->Open();
}
//-----
void __fastcall TForm2::BitBtn6Click(TObject *Sender)
{
Form12->QuickRep1->Preview();
}
//-----

```

```

void __fastcall TForm2::BitBtn7Click(TObject *Sender)
{
Form12->QuickRep1->Print();
}
//-----
void __fastcall TForm2::BitBtn15Click(TObject *Sender)
{
Form12->QuickRep3->Preview();
}
//-----
void __fastcall TForm2::BitBtn16Click(TObject *Sender)
{
Form12->QuickRep3->Print();
}
//-----

void __fastcall TForm3::ComboBox1Change(TObject *Sender)
{
if (ComboBox1->ItemIndex == 0){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Белобров Сергей Александрович");
}
else if (ComboBox1->ItemIndex == 1){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Сергеева Анна Ивановна");
    ComboBox2->Items->Add("Смирнов Илья Игоревич");
    ComboBox2->Items->Add("Борисова Екатерина Анатольевна");
    ComboBox2->Items->Add("Иванова Ирина Сергеевна");
    ComboBox2->Items->Add("Михалова Светлана Валерьевна");
    ComboBox2->Items->Add("Рыжков Иван Александрович");
    ComboBox2->Items->Add("Федоров Вадим Викторович");
    ComboBox2->Items->Add("Бодров Александр Алексеевич");
    ComboBox2->Items->Add("Зайцева Анна Эдуардовна");
    ComboBox2->Items->Add("Сергеенко Марина Евгеньевна");
    ComboBox2->Items->Add("Волненко Руслан Сергеевич");
}
else if (ComboBox1->ItemIndex == 2){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Петрова Юлия Александровна");
}
else if (ComboBox1->ItemIndex == 3){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Солнцев Игорь Сергеевич");
}
else if (ComboBox1->ItemIndex == 4){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Шохина Кристина Сергеевна");
}

ComboBox2->Text="ФИО";

}
//-----

void __fastcall TForm3::ComboBox2Change(TObject *Sender)
{
Edit1->Enabled = true;
//else{
//ShowMessage("Ошибка, отсутствует введенный сотрудник! Выберите сотрудника из предложенного выпадающего списка");
}
}

```

```
//}  
}
```

Программный код Unit3.cpp:

```
void __fastcall TForm3::BitBtn1Click(TObject *Sender)  
{  
    Form3->Visible = false;  
    if ((ComboBox1->ItemIndex==0)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0001")){  
        Form1->Show();  
        Form1->BitBtn1->Enabled = true;  
        Form1->BitBtn2->Enabled = true;  
        Form1->BitBtn3->Enabled = true;  
        Form1->BitBtn4->Enabled = true;  
        Form1->BitBtn5->Enabled = true;  
        Form1->BitBtn6->Enabled = true;  
        Form1->BitBtn7->Enabled = true;  
        Form1->BitBtn8->Enabled = true;  
        flag = 1;// если глав. врач  
    }  
    else if ((ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==1)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==2)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==3)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==4)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==5)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==6)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==7)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==8)&& (Edit1->Text == "0002")  
    || (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==9)&& (Edit1->Text == "0002")){  
        flag = 2;// узкий. спец.  
        Form1->Show();  
        Form1->BitBtn1->Enabled = false;  
        Form1->BitBtn2->Enabled = true;  
        Form1->BitBtn3->Enabled = true;  
        Form1->BitBtn4->Enabled = true;  
        Form1->BitBtn5->Enabled = true;  
        Form1->BitBtn6->Enabled = true;  
        Form1->BitBtn7->Enabled = true;  
        Form1->BitBtn8->Enabled = false;  
    }else if ((ComboBox1->ItemIndex==2)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0003")){  
        flag = 3; //кассиp  
        Form1->Show();  
        Form1->BitBtn1->Enabled = false;  
        Form1->BitBtn2->Enabled = false;  
        Form1->BitBtn3->Enabled = false;  
        Form1->BitBtn4->Enabled = false;  
        Form1->BitBtn5->Enabled = false;  
        Form1->BitBtn6->Enabled = false;  
        Form1->BitBtn7->Enabled = false;  
        Form1->BitBtn8->Enabled = true;  
    }else if ((ComboBox1->ItemIndex==3)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0004")){  
        flag = 4;//регистратор  
        Form1->Show();  
        Form1->BitBtn1->Enabled = false;  
        Form1->BitBtn2->Enabled = true;  
        Form1->BitBtn3->Enabled = true;  
        Form1->BitBtn4->Enabled = true;  
        Form1->BitBtn5->Enabled = true;  
        Form1->BitBtn6->Enabled = true;  
        Form1->BitBtn7->Enabled = true;  
        Form1->BitBtn8->Enabled = false;  
    }else if ((ComboBox1->ItemIndex==4)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0000")){
```

```

        flag = 5;//сисадмин
        Form1->Show();
        Form1->BitBtn1->Enabled = true;
        Form1->BitBtn2->Enabled = true;
        Form1->BitBtn3->Enabled = true;
        Form1->BitBtn4->Enabled = true;
        Form1->BitBtn5->Enabled = true;
        Form1->BitBtn6->Enabled = true;
        Form1->BitBtn7->Enabled = true;
        Form1->BitBtn8->Enabled = true;
    }else{
        ShowMessage("Вы не прошли авторизацию! Проверьте корректность данных!");
        Form3->Show();
    }
}
//-----

void __fastcall TForm3::FormCreate(TObject *Sender)
{
//Form3->BorderStyle=bsSizeToolWin;
}

```

Программный код Unit5.cpp:

```

void __fastcall TForm5::DBEdit1Click(TObject *Sender)
{
Form10->Show();
Form10->TabSheet5->Show();
}
//-----

void __fastcall TForm5::N14Click(TObject *Sender)
{
Form5->Visible = false;
Form2->Show();
Form2->TabSheet1->Show();
}
//-----

void __fastcall TForm5::N15Click(TObject *Sender)
{
Form5->Visible = false;
Form2->Show();
Form2->TabSheet2->Show();
}
//-----

void __fastcall TForm5::N2Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form5->N1->Visible=false;
Form5->N12->Visible=false;
}
Form5->Visible = false;
Form4->Show();
}
//-----

void __fastcall TForm5::N7Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form6->N1->Visible=false;
Form6->N12->Visible=false;
}
}

```

```

}
Form5->Visible = false;
Form6->Show();
Form6->TabSheet1->Show();
}
//-----

void __fastcall TForm5::N8Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form6->N1->Visible=false;
Form6->N12->Visible=false;
}
Form5->Visible = false;
Form6->Show();
Form6->TabSheet2->Show();
}
//-----

void __fastcall TForm5::N16Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form7->N1->Visible=false;
Form7->N12->Visible=false;
}
Form5->Visible = false;
Form7->Show();
Form7->TabSheet1->Show();
}
//-----

void __fastcall TForm5::N10Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form7->N1->Visible=false;
Form7->N12->Visible=false;
}
Form5->Visible = false;
Form7->Show();
Form7->TabSheet2->Show();
}
//-----

void __fastcall TForm5::N11Click(TObject *Sender)
{
if ((flag == 2) || (flag == 4)){
Form7->N1->Visible=false;
Form7->N12->Visible=false;
}
Form5->Visible = false;
Form7->Show();
Form7->TabSheet3->Show();
}
//-----

void __fastcall TForm5::N12Click(TObject *Sender)
{
Form5->Visible = false;
Form8->Show();
}
//-----

void __fastcall TForm5::N13Click(TObject *Sender)
{

```

```

Form5->Visible = false;
Form3->Show();
}
//-----
void __fastcall TForm5::FormClose(TObject *Sender, TCloseAction &Action)
{
Form3->Close();
}
//-----
void __fastcall TForm5::BitBtn8Click(TObject *Sender)
{
AnsiString d;
switch (RadioGroup1->ItemIndex)
{case 0: d = " "; break;
case 1: d = " DESC";
}
DataModule9->IBQuery4->Close();
DataModule9->IBQuery4->SQL->Clear();
DataModule9->IBQuery4->SQL->Add("select * from VIEW_SER");

switch (ComboBox1->ItemIndex)
{case 0: d= "ORDER BY ID_service"+ d; break;
case 1: d= "ORDER BY service_name"+ d; break;
case 2: d= "ORDER BY types"+ d; break;
case 3: d= "ORDER BY price"+ d; break;
}
DataModule9->IBQuery4->SQL->Add(d);
DataModule9->IBQuery4->Open();
}
//-----
void __fastcall TForm5::ComboBox1Click(TObject *Sender)
{
Form2->ComboBox2->Text="Поле";
Edit7->Clear();
}
//-----
void __fastcall TForm5::Button2Click(TObject *Sender)
{
Edit1->Clear();
Edit3->Clear();

}
//-----
void __fastcall TForm5::Button1Click(TObject *Sender)
{
Edit7->Clear();
}
//-----

void __fastcall TForm5::BitBtn9Click(TObject *Sender)
{
DataModule9->IBQuery4->Close();
DataModule9->IBQuery4->SQL->Clear();
switch (ComboBox2->ItemIndex)
{
case 0:
DataModule9->IBQuery4->SQL->Add("select * from VIEW_SER where ID_service like '%" + Edit7->Text +
"% '");
break;
case 1:
DataModule9->IBQuery4->SQL->Add("select * from VIEW_SER where service_name like '%" + Edit7-
>Text + "% '");
break;
}
}
}

```

```

case 2:
    DataModule9->IBQuery4->SQL->Add("select * from VIEW_SER where types like '%" + Edit7->Text + '%"
");
    break;
case 3:
    DataModule9->IBQuery4->SQL->Add("select * from VIEW_SER where price like '%" + Edit7->Text + '%"
");
    break;
}
DataModule9->IBQuery4->Open();
if ((DataModule9->IBQuery4->RecordCount==0) || (Form5->Edit7->Text == ""))
{
    ShowMessage("Не найдено!");
}
}
//-----
void __fastcall TForm5::BitBtn5Click(TObject *Sender)
{
    DataModule9->IBQuery4->Close();
    DataModule9->IBQuery4->SQL->Clear();
    DataModule9->IBQuery4->SQL->Add("select * from VIEW_SER");
    DataModule9->IBQuery4->Open();
    Form5->ComboBox2->Text="Поле";
    Edit7->Clear();
    Form5->ComboBox1->Text="По полю";
    Label3->Font->Color=clWindowText;
    DBEdit1->Font->Color=clWindowText;
}
//-----

void __fastcall TForm5::ComboBox2Click(TObject *Sender)
{
    Form5->ComboBox1->Text="По полю";
}
//-----

void __fastcall TForm5::BitBtn17Click(TObject *Sender)
{
    AnsiString d;
    switch (RadioGroup2->ItemIndex)
    {case 0: d = " "; break;
    case 1: d = " DESC";
    }
    DataModule9->IBQuery5->Close();
    DataModule9->IBQuery5->SQL->Clear();
    DataModule9->IBQuery5->SQL->Add("select * from TYPES_OF_SERVICE");

    switch (ComboBox3->ItemIndex)
    {case 0: d= "ORDER BY ID_types"+ d; break;
    case 1: d= "ORDER BY types"+ d; break;
    }
    DataModule9->IBQuery5->SQL->Add(d);
    DataModule9->IBQuery5->Open();
}
//-----

void __fastcall TForm5::BitBtn18Click(TObject *Sender)
{
    DataModule9->IBQuery5->Close();
    DataModule9->IBQuery5->SQL->Clear();

    switch (ComboBox4->ItemIndex)
    {
        case 0:

```



```

        DataModule9->IBQuery5->SQL->Add("select * from TYPES_OF_SERVICE where ID_types like '%" +
Edit9->Text + "%' ");
        break;
    case 1:
        DataModule9->IBQuery5->SQL->Add("select * from TYPES_OF_SERVICE where types like '%" + Edit9-
>Text + "%' ");
        break;
    }
DataModule9->IBQuery5->Open();
if ((DataModule9->IBQuery5->RecordCount==0) || (Form5->Edit9->Text == ""))
{
    ShowMessage("Не найдено!");
}
}
//-----
void __fastcall TForm5::Button4Click(TObject *Sender)
{
    Edit10->Clear();
}
//-----
void __fastcall TForm5::Button3Click(TObject *Sender)
{
    Edit9->Clear();
}
//-----
void __fastcall TForm5::ComboBox3Click(TObject *Sender)
{
    Form2->ComboBox4->Text="Поле";
    Edit9->Clear();
}
//-----

void __fastcall TForm5::ComboBox4Click(TObject *Sender)
{
    Form5->ComboBox3->Text="По полю";
}
//-----

void __fastcall TForm5::BitBtn14Click(TObject *Sender)
{
    DataModule9->IBQuery5->Close();
    DataModule9->IBQuery5->SQL->Clear();
    DataModule9->IBQuery5->SQL->Add("select * from TYPES_OF_SERVICE");
    DataModule9->IBQuery5->Open();
    Form5->ComboBox4->Text="Поле";
    Edit9->Clear();
    Form5->ComboBox3->Text="По полю";
}
//-----

void __fastcall TForm5::BitBtn1Click(TObject *Sender)
{
    if ((Form5->Edit1->Text != "" ) && (Form5->Edit3->Text != "" ) ){
        DataModule9->IBStoredProc5->ParamByName ("service_name")->AsString=Edit1->Text;
        DataModule9->IBStoredProc5->ParamByName ("id_types")->AsString=DBEdit1->Text;
        DataModule9->IBStoredProc5->ParamByName ("price")->AsString=Edit3->Text;
        DataModule9->IBStoredProc5->Prepare();
        DataModule9->IBStoredProc5->ExecProc();
        DataModule9->IBQuery4->Close();
        DataModule9->IBQuery4->Open();
    }else{
        ShowMessage("Заполните все поля!");
    }
}

```

```

}
}
//-----

void __fastcall TForm5::BitBtn10Click(TObject *Sender)
{
if (Form5->Edit10->Text != "") {
DataModule9->IBStoredProc6->ParamByName ("types")->AsString=Edit1->Text;
DataModule9->IBStoredProc6->Prepare();
DataModule9->IBStoredProc6->ExecProc();
DataModule9->IBQuery5->Close();
DataModule9->IBQuery5->Open();
}else{
    ShowMessage("Заполните все поля!");
}
}
//-----

void __fastcall TForm5::BitBtn2Click(TObject *Sender)
{
DataModule9->IBStoredProc15->ParamByName ("id_service")->AsString=DBEdit2->Text;
DataModule9->IBStoredProc15->Prepare();
DataModule9->IBStoredProc15->ExecProc();
DataModule9->IBQuery4->Close();
DataModule9->IBQuery4->Open();
}
//-----

void __fastcall TForm5::BitBtn11Click(TObject *Sender)
{
DataModule9->IBStoredProc16->ParamByName ("id_types")->AsString=DBEdit3->Text;
DataModule9->IBStoredProc16->Prepare();
DataModule9->IBStoredProc16->ExecProc();
DataModule9->IBQuery5->Close();
DataModule9->IBQuery5->Open();
}
//-----

void __fastcall TForm5::BitBtn4Click(TObject *Sender)
{
DataModule9->IBStoredProc27->ParamByName ("id_ser")->AsString=DBEdit2->Text;
if (Edit1->Text != ""){
    DataModule9->IBStoredProc27->ParamByName ("name")->AsString=Edit1->Text;
}else{
    DataModule9->IBStoredProc27->ParamByName ("name")->AsString=DBEdit4->Text;
}
DataModule9->IBStoredProc27->ParamByName ("id_types")->AsString=DBEdit1->Text;

if (Edit3->Text != ""){
    DataModule9->IBStoredProc27->ParamByName ("price")->AsString=Edit3->Text;
}else{
    DataModule9->IBStoredProc27->ParamByName ("price")->AsString=DBEdit6->Text;
}
DataModule9->IBStoredProc27->Prepare();
DataModule9->IBStoredProc27->ExecProc();
DataModule9->IBQuery4->Close();
DataModule9->IBQuery4->Open();
Label3->Font->Color=clWindowText;
DBEdit1->Font->Color=clWindowText;
}
//-----

void __fastcall TForm5::BitBtn3Click(TObject *Sender)

```

```

{
ShowMessage("Внимание! Вы перешли в режим редактирования! Заполните обязательные поля, выделенные
КРАСНЫМ цветом. Для отмены нажмите кнопку ОБНОВИТЬ. ");
Label3->Font->Color=clRed;
DBEdit1->Font->Color=clRed;
}
//-----

void __fastcall TForm5::BitBtn12Click(TObject *Sender)
{
ShowMessage("Внимание! Вы перешли в режим редактирования! Для отмены нажмите кнопку ОБНОВИТЬ.
");
}
//-----

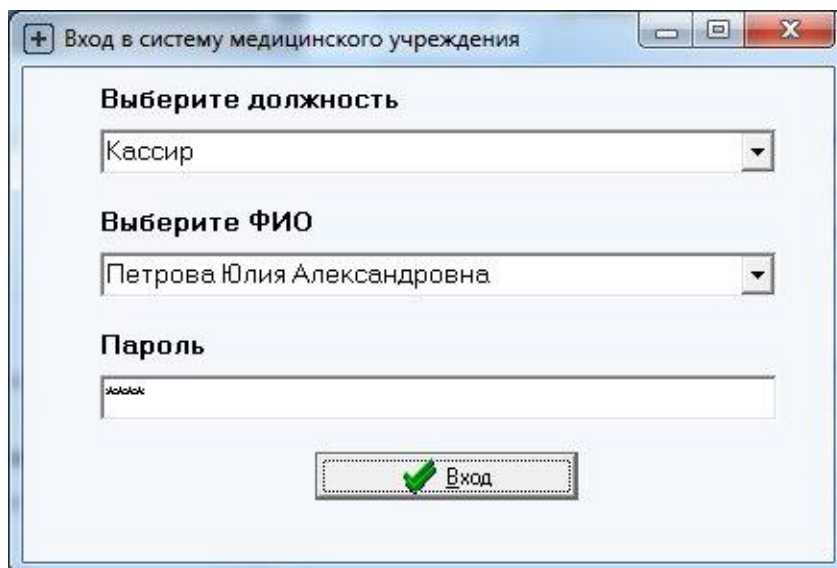
void __fastcall TForm5::BitBtn13Click(TObject *Sender)
{
DataModule9->IBStoredProc28->ParamByName ("id_types")->AsString=DBEdit3->Text;
if (Edit10->Text != ""){
    DataModule9->IBStoredProc28->ParamByName ("types")->AsString=Edit10->Text;
}else{
    DataModule9->IBStoredProc28->ParamByName ("types")->AsString=DBEdit7->Text;
}
DataModule9->IBStoredProc28->Prepares();
DataModule9->IBStoredProc28->ExecProc();
DataModule9->IBQuery5->Close();
DataModule9->IBQuery5->Open();
DataModule9->IBQuery4->Close();
DataModule9->IBQuery4->Open();
}
}

```

ПРИЛОЖЕНИЕ Д

Экранные форма разработанной подсистемы

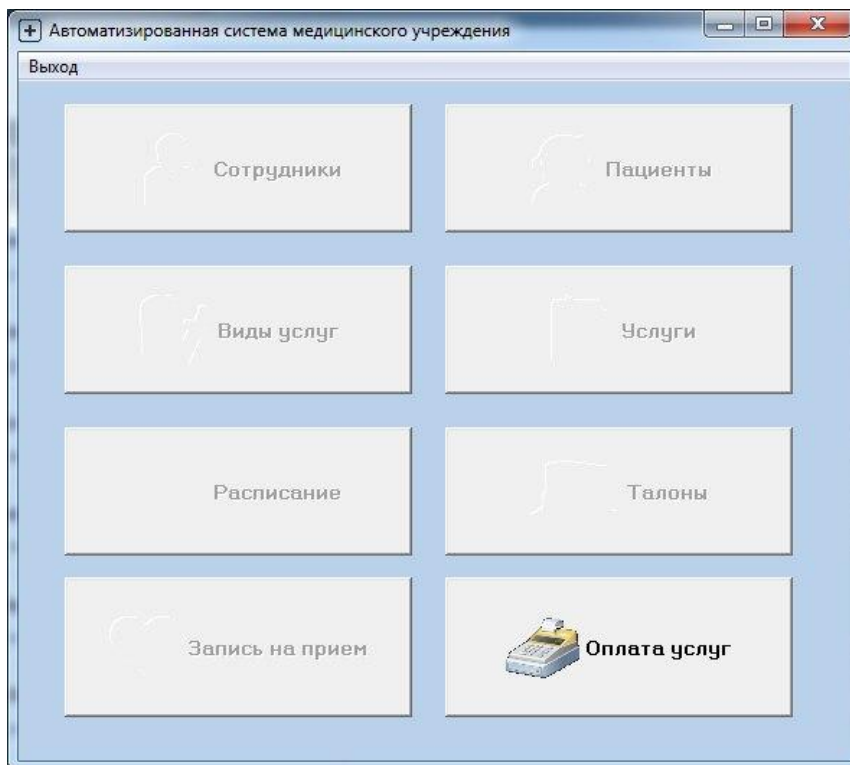
На рисунке Е1 представлена авторизация кассира при входе в приложение.



The screenshot shows a window titled "Вход в систему медицинского учреждения" (Login to the medical institution system). It contains three input fields: "Выберите должность" (Select position) with a dropdown menu showing "Кассир" (Cashier); "Выберите ФИО" (Select full name) with a dropdown menu showing "Петрова Юлия Александровна"; and "Пароль" (Password) with a masked input field. Below the fields is a button labeled "Вход" (Login) with a green checkmark icon.

Рисунок Д.1 – Окно авторизации

На рисунке Е2 изображено окно главного меню с активными кнопками, доступными для кассира.



The screenshot shows a window titled "Автоматизированная система медицинского учреждения" (Automated medical institution system) with a "Выход" (Exit) button in the top left. The main area contains eight buttons arranged in a 4x2 grid: "Сотрудники" (Employees), "Пациенты" (Patients), "Виды услуг" (Types of services), "Услуги" (Services), "Расписание" (Schedule), "Талоны" (Receipts), "Запись на прием" (Appointment), and "Оплата услуг" (Payment of services) which includes a calculator icon.

Рисунок Д.2 – Главное меню с активными кнопками

На рисунке Е3 представлена авторизация доктора при входе в приложение.

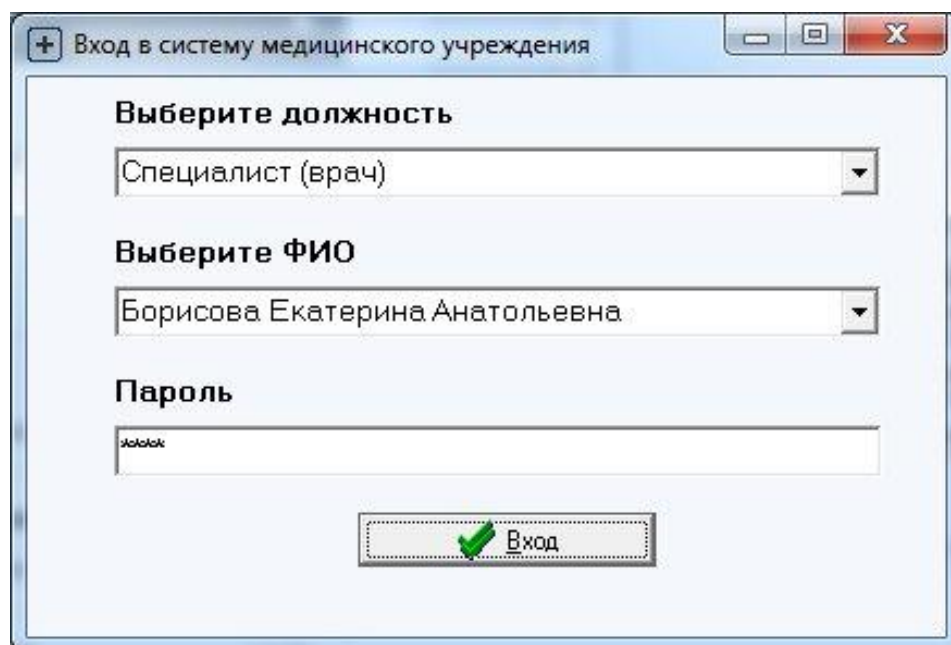


Рисунок Д.3 – Окно авторизации

На рисунке Е4 изображено окно главного меню с активными кнопками, доступными для специалиста (врача).

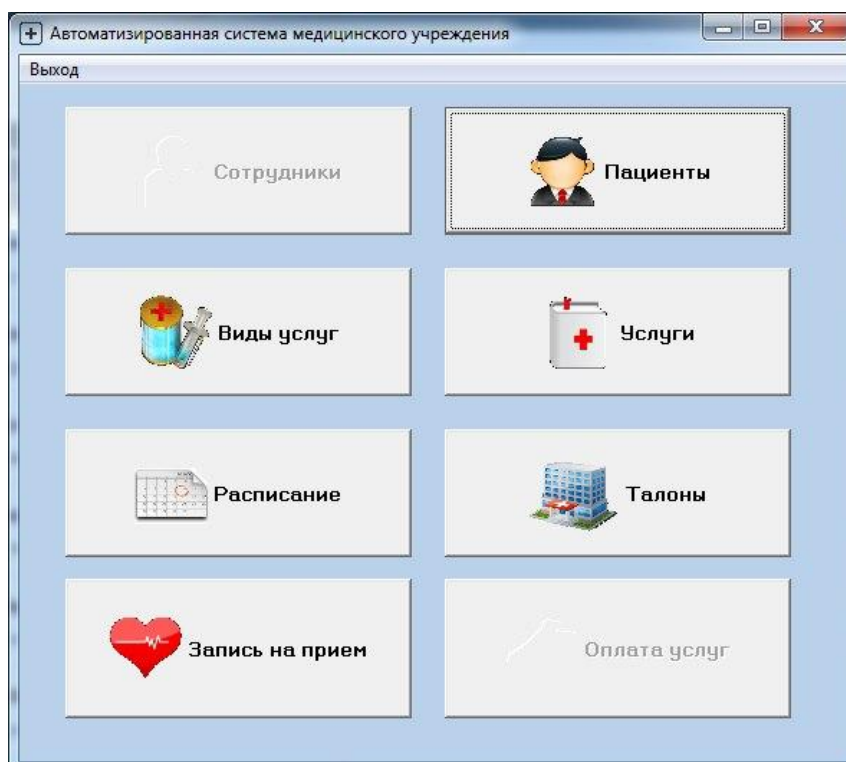


Рисунок Д.4 – Главное меню с активными кнопками

Услуги			
№	Наименование услуги	Вид услуги	Стоимость, р
1	Консультация врача-терапевта	Консультация	550
2	Консультация врача-эндокринолога	Консультация	600
3	Консультация врача-невролога	Консультация	600
4	Консультация врача-гастроэнтеролога	Консультация	600
5	Вызов на дом врача-специалиста	Консультация	1200
6	Забор крови из вены	Лабораторные исследования	100
7	Забор крови из пальца	Лабораторные исследования	50
8	Определение глюкозы экспресс мет.	Лабораторные исследования	85
9	Тест определения вирус гепат. В	Лабораторные исследования	165
10	Тест определения вирус гепат. С	Лабораторные исследования	165

258% Page 1 of 2

Рисунок Д.5 – Отчет по услугам

Сотрудники							
№	Фамилия	Имя	Отчество	Телефон	Адрес	Паспорт	Должность
1	Белобров	Сергей	Александрович	89624785219	г.Белгород, ул.Горького, 26	1563 875611	Главный врач
2	Сергеева	Анна	Ивановна	89206571269	г.Белгород, ул. Победы, 29	1245 365274	Терапевт
3	Смирнов	Илья	Игоревич	89257493685	г.Белгород, ул. Кн.Трубецкого, 17	6985 235754	Терапевт
4	Борисова	Екатерина	Анатольевна	85263641287	г.Белгород, ул. Щорса, 56	2365 965234	Терапевт
5	Иванова	Ирина	Сергеевна	89043698524	г.Белгород, ул. Щорса, 29	5874 338725	Ревматолог
6	Рыжков	Иван	Александрович	87423691452	г.Белгород, ул. Пирогова, 22	3652 749652	Аллерголог
7	Михалова	Светлана	Валерьевна	89050973853	г.Белгород, ул.Горького, 15	8520 745632	Ревматолог
8	Бодров	Александр	Алексеевич	89631478523	г.Белгород, ул. Королева, 45	2365 745696	Уролог
9	Федоров	Вадим	Викторович	89641236587	г.Белгород, ул.Королева, 26	9856 147536	Гастроэнтеролог
10	Сидорова	Анна	Сергеевна	89024456850	г.Белгород, ул. Победы, 11	8602 405000	Физиотерапевт

109% Page 1 of 1

Рисунок Д.6 – Отчет по сотрудникам

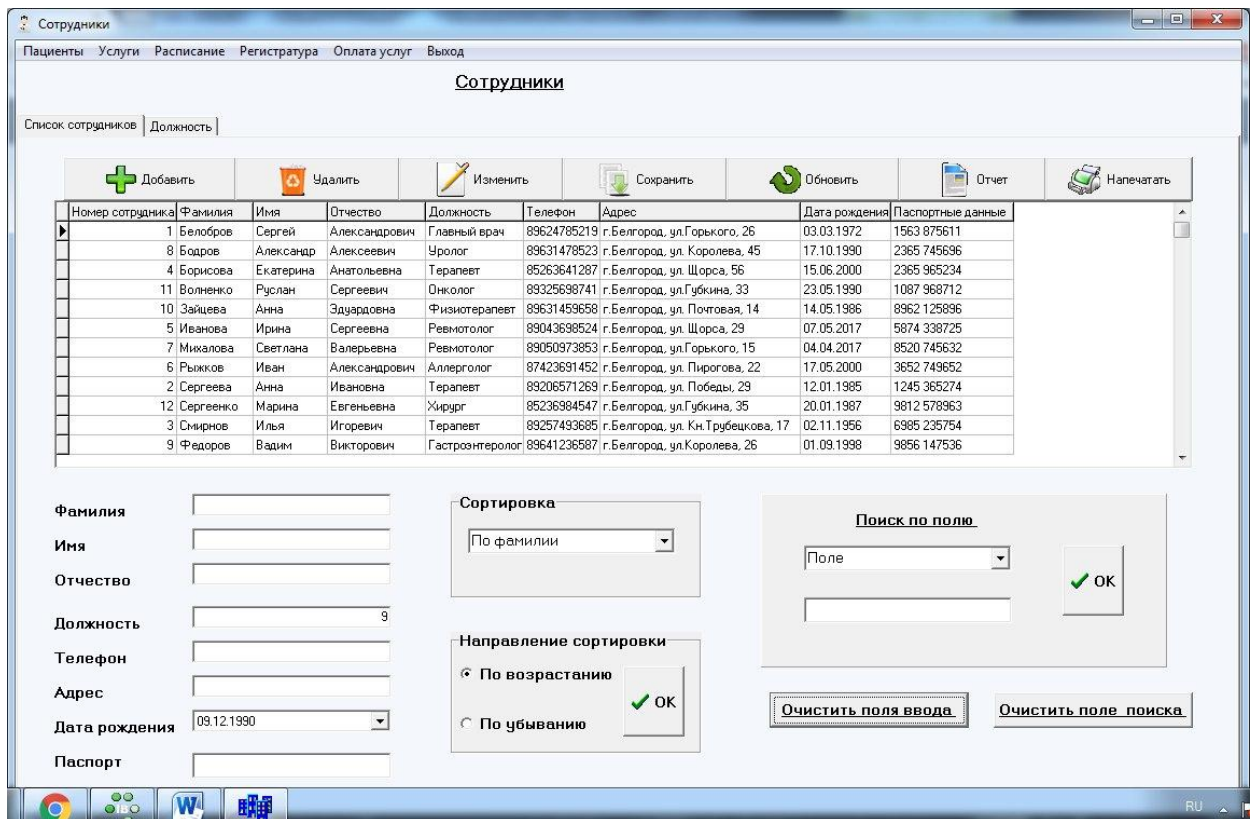


Рисунок Д.7 – Сортировка списка сотрудников по фамилии

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« » _____ 2017 г.

Студент _____ Шохина К.С.