

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

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой, д.т.н., с.н.с.
/ Б.М. Суховилов /
« ____ » _____ 20__ г.

Создание информационной системы для частной клиники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 09.03.03.2021.301/252.ВКР

Консультант, _____
_____/_____/_____
« ____ » _____ 2021 г.

Руководитель, доцент
_____/ В.А. Конов /
« ____ » _____ 2021 г.

Автор
студент группы ЭУ – 582
Д.С. Пяткин
« ____ » _____ 2021 г.

Нормоконтролер, доцент
_____/Е.А. Конова/
« ____ » _____ 2021 г.

Челябинск 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1 Изучение предметной области	8
2 ПОСТАНОВКА ЗАДАЧИ	10
3 Анализ существующих систем	12
1.1 MedIdea.....	12
1.2 Medesk.....	14
1.3 Медицинская информационная система qMS.	15
1.1 Медицинская система Archimed+	16
1.2 Вывод.....	19
4 РАЗРАБОТКА ДИАГРАММЫ ПОТОКОВ ДАННЫХ.....	21
5 ПРАКТИЧЕСКАЯ ЧАСТЬ	25
5.1 Разработка базы данных	25
5.2 Разработка приложения.....	31
6 РАЧЕТ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ ВНЕДРЕНИЯ ПРОГРАММНОГО ПРОДУКТА.....	39
6.1 Затраты на разработку продукта	39
6.2 Определение доходов от внедрения в работу программного продукта.	43
ЗАКЛЮЧЕНИЕ	47
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	48
ПРИЛОЖЕНИЯ	
ПРИЛОЖЕНИЕ А. Код хранимых процедур базы данных.....	50
ПРИЛОЖЕНИЕ Б. Код приложения.....	51

ВВЕДЕНИЕ

Информационные технологии прочно вошли в жизнь современного общества. На данный момент мы можем видеть их присутствие в повседневной жизни. Многие процессы можно упростить с помощью данных технологий, ускорить обмен данными. Уменьшить временные затраты, трудоемкость процесса, упростить сбор информации.

Сфера по предоставлению медицинских услуг несомненно нуждается во внедрении таких технологий. Это связано со сбором и хранением информации, которая сейчас в основной своей массе хранится на бумажных носителях, использование которых подвержено многим рискам. В основном это человеческий фактор. Примером будет являться утеря носителя, как такового. Так же при больших объемах информации появляется проблема с хранением такого объема. Становится необходимость какой-то системы хранения, что несет в себе затраты. При этом поиск необходимой информации становится более долгим и трудоемким процессом. Возьмем в пример простую поликлинику, в которую вы пришли на повторный прием, когда врач не вернул карту в регистратуру, но еще и принимает в другом кабинете, а карта в закрытом, но об этом еще пока никто не знает. А результаты ваших анализов случайно положили в коробочку не с вашим участком. Тогда время препровождения в медучреждении становится больше, чем хотелось бы. Оно становится не комфортным не только для вас, но и для окружающих.

В решении данных проблем может помочь внедрение информационной системы в медицинские организации. Будь это частная организация, например, кабинет частного врача, либо муниципальное учреждение. Такая система снизит риск утери информации, существенно ускорит поиск необходимой.

Существует множество организаций, которые предоставляют услуги по внедрению медицинских информационных систем в соответствующие организации. Предлагается множество функций и услуг, которые по заявлению

разработчиков программного обеспечения могут обеспечить стабильное хранение данных и оперативное предоставление информации.

Предстоит выделить несколько крупных, наиболее популярных информационных систем и подробнее изучить их. Это необходимо для более детального понимания того, что присутствует на рынке информационных услуг. Какие функции могут выполнять МИС. На сколько хорошо они могут помочь во взаимодействии персонала, начальства и пациентов. На сколько вообще развит рынок информационных систем именно в медицинской сфере и на сколько удалось автоматизировать процессы в клинике. А также рассмотреть какие это процессы и на что нацелены.

Так же необходимо провести анализ потребностей для дальнейшей разработки информационной системы. В частности, его функционала и ориентированности.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Изучение предметной области

Предметной областью будет являться частная клиника предоставляющая медицинские услуги на коммерческой основе.

Обычно в основе любого, даже самого маленького медицинского учреждения лежит персонал, обслуживающий людей. В первую очередь это такие персоны, как администратор, который вас встречает при входе, либо который ответит на ваш звонок. Далее идет врач, к которому вас направляет администратор. Это будет базовый минимум персонала, для работы такого рода деятельности бизнеса. Хотя, бывают случаи, когда работает один только врач, который и является администратором.

Когда пациент приходит в первый раз в клинику, записываются его данные в виде карточки, далее они хранятся в клинике. При повторном приеме необходимо найти уже имеющуюся карту и передать ее врачу, который будет принимать пациента. Либо выдать карту пациенту на руки, что может привести к потере этой информации, если клиент не дойдет до врача. Так же при большом количестве карт увеличивается время их поиска администратором, что сказывается на скорости приема клиента. Этот фактор становится критичным, по мере увеличения числа пациентов.

Рассмотрим это со стороны врача. Ко врачу приходит пациент. Отдает карту, если с внешней стороны все понятно, то внутри, скорее всего, уже не все так хорошо. Врачу приходится разбираться в почерке других врачей, а если карта уже потерялась, то информация будет не актуальна. Обычно пациенту еще приходится разбираться, что же такое написано в рецепте.

Так же стоит учитывать материалы. Особенно если это какой-нибудь кабинет стоматолога или в организации в принципе присутствует предоставление процедурных услуг. Ведь учет лекарственных средств трудоемкий и кропотливый процесс при котором очень просто допустить ошибку или упустить что-то из виду. Стоит учитывать, что даже на обычный прием врача, при осмотре, уходит пара

смотровых перчаток, пара бахил, маска и дезинфицирующее средство. А при обычном УЗИ дополнительно понадобится еще и гель, одноразовая простыня и несколько салфеток. Если часто допускать ошибки, то какой-нибудь из материалов может закончиться не в подходящий момент, что негативно скажется на обслуживании клиентов.

Для бизнеса важна скорость и качество обслуживания клиентов. В этом и может помочь внедрение информационной системы. С помощью ИС можно автоматизировать часть процессов, которые делались человеком вручную. Снизить шанс допущения ошибки. Ускорить работу, путем снижения ввода информации.

2 ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать информационную систему для сбора, хранения, обработки данных небольшой частной клиники. А также вывода этой информации в отредактированном виде.

Необходимо вести учет личных данных пациентов, сотрудников, материалов, предоставляемых услуг, ведение карт пациентов.

Так же важен доступ к информации средствами интернет технологий.

Работа будет осуществляться на рабочих станциях. Так же будет осуществлен доступ к информации на сайте.

Для реализации поставленных задач потребуется выбор СУБД и разработка базы данных для хранения, накопления информации и доступа к ней.

Принято решение использовать систему управления реляционными базами данных SQL Server. Выбор именно этого программного продукта сделан не спроста. Эта система является одной из самых популярных в мире. Может использоваться как в небольших проектах, так и в крупномасштабных. Эта система с 16 версии доступна на Linux системах, что делает ее доступней. Что очень важно, SQL server имеет высокую надежность и безопасность, т.к. предоставляет шифрование данных, а это безопасность личных данных клиентов клиники. Высокая скорость работы даже с большими объемами данных. Существует несколько способов подключения.

Неотъемлемым преимуществом перед некоторыми другими системами управления БД является то, что есть как бесплатные версии, так и версии, за которые придется заплатить. Существует 2 вида бесплатных версий. Они интересуют больше всего. Первая версия - SQL Server 2019 Developer. Она имеет полный набор функций, созданные базы данных предназначены для разработки и тестирования и не годятся для применения в рабочей среде. Вторая версия - SQL Server 2019 Express. Подходит для разработки приложений, использования на персональном компьютере, веб-серверах и небольших серверах.

Стоит отметить легкий перенос в облачный сервис Azure. При увеличении объема данных, или финансовая нецелесообразность держать сервер поможет виртуальная машина Azure SQL Server.

Для рабочей станции решено написать приложение на языке программирования *c#*. Этот язык выбран потому что он несколько проще в написании, чем *c++*, так же он является современным объектно-ориентированным языком, относящимся к семейству *си*. Имеет большой функционал, прост в работе, а также приложения на этом языке считаются надежными и безопасными.

Для удаленного доступа необходимо написать веб-сайт. Выбор упал на такой язык, как ASP.NET. Упрощает работу то, что можно использовать платформу Visual studio. В графическом интерфейсе платформы отлично видно, как происходит построение написанного кода в реальном времени.

3 АНАЛИЗ СУЩЕСТВУЮЩИХ СИСТЕМ

Медицинская информационная система (МИС) — система автоматизации документооборота для медицинских учреждений, в которой объединены система поддержки принятия врачебных решений, электронные медицинские карты пациентов, данные медицинских исследований в цифровой форме, данные мониторинга состояния пациента с медицинских приборов, средства общения между сотрудниками, финансовая и административная информация.

МИСами обычно используется стандарт передачи данных HL7, Health Level 7, описывающий процедуры и механизмы обмена, управления и интеграции электронной медицинской информации.

МИС можно классифицировать по направлению деятельности медицинского учреждения:

1. МИС для стационаров
2. МИС для поликлиник и амбулаторий
3. МИС для стоматологических клиник
4. МИС для санаториев (лечебно-профилактических учреждений)

Рассмотрим 5 информационных систем, направленных на автоматизацию медицинских учреждений.

1.1 MedIdea.

Она характеризует себя как современная медицинская информационная система для клиник. Идеально подойдет как для крупного медицинского центра, так и для частного кабинета врача. Интуитивно понятный интерфейс программы, не требующий обучения. Комплекс обеспечит: сокращение расходов, структурирование документов, сокращение времени на приём, онлайн-запись пациентов, увеличение прибыли.

Включает в себя большой функционал, такой как:

1. Модуль работы в ОМС
2. Модуль складского учета

3. Модуль постановки задач
4. Модуль сетки расписания
5. Модуль телефония
6. Модуль отчетов
7. Модуль рассылки
8. Модуль истории болезни
9. Модуль онлайн записи
10. Модуль кассы
11. Модуль сертификаты и абонементы
12. Модуль лист ожидания
13. Модуль скидки
14. Интеграция с Prodoctorov

К положительным сторонам можно отнести гибкость, которая заключается в модульности данной МИС. Все это основывается на коммерческой основе. Каждый модуль продается отдельно от базового набора функций. Цена МИС складывается из стоимости модулей и количества рабочих мест, которые будут обеспечиваться данной системой. Это сделано для того, чтобы охватить больше потенциальных клиентов, будь то отдельный врачебный кабинет или сеть клиник.

Минусами данной системы является стоимость как самих модулей, так и цена для каждого рабочего места. Например, за 1 рабочее место компания просит 4500 рублей плюс цена основного модуля составит еще 50 тысяч рублей, что не приемлемо в наших рамках. Если понадобится дополнительный модуль для расширения функционала, то, например, за модуль расчета заработной платы придется выложить 17000, а за модуль «Касса» еще 25000 рублей. Итого за 10 рабочих мест, базовый модуль, модуль касса и модуль расчета заработной платы придется отдать 137 000 рублей. Но и это еще не все. Технической поддержки за эти деньги не будет, она продается отдельно. За 1 месяц придется отдать 6500 рублей, но при покупке 12 мес, мы можем получить 2 месяца в подарок. В конечном итоге получаем 14 мес технической поддержки за 78000 рублей и итоговую

стоимость в 215000 рублей. Можно отнести к плюсам бесплатные внедрение системы, перенос базы данных, обучение сотрудников и доработку программы под нужды.

В заключение можно сказать, что система будет удобна для разных компаний по объему рабочих мест и потребностям, но не удобна в финансовом плане, так как нужно еще ежегодно оплачивать техническую поддержку, да и все остальное стоит не дешево. Так же можно отметить, что существует пробный период, на котором можно посмотреть, действительно ли необходима эта программа.

1.2 Medesk.

Характеризует себя как надежная платформа для простого и оперативного управления частной клиникой. Помогает анализировать и улучшать работу каждого сотрудника. Предлагает множество возможностей, таких как:

1. Удаленный доступ к системе, запись и прием.
2. Аналитика клиники, которая включает более 40 готовых отчетов по различным параметрам, конструктор отчетов и т.д.
3. Управление расписанием, синхронизация с календарем.
4. Единый источник данных обо всех пациентах.
5. Инструменты коммуникации с пациентами, рассылки.
6. Обеспечение защиты данных, разграничение прав сотрудников.

Базовый комплект для клиники включает множество функций, но не все они будут востребованы в нашем случае.

Минусом является множество лишних функций, заложенных в базовую версию, за которую придется платить. Так же невозможность самостоятельно на сайте узнать хотя бы приблизительную стоимость системы без оставления заявки с номером своего телефона.

Плюсом можно назвать то, что техническая поддержка функционирует всегда, когда оплачена система.

1.3 Медицинская информационная система qMS.

Позиционирует себя, как полнофункциональная МИС, которая объединяет пациентов, руководителей, персонал, лаборатории, страховые организации в одном информационном пространстве. Которое может выступать для организация различного профиля. Так же упоминается о поддержке различных устройств.

МИС qMS структурирует все основные информационно-аналитические процессы учреждения и адаптируется под клинику любого масштаба. Система обеспечивает прозрачность бизнес-процессов, а также оптимизацию материальных и трудовых ресурсов, что ведет к повышению качества и безопасности оказания медицинской помощи.

Из дополнительных функций можно отметить интеграцию с государственными системами здравоохранения, интеграцию с таким оборудованием, как фискальные регистраторы, терминалы самозаписи, сканеры, лабораторное оборудование, реанимационное оборудование, ЭКГ и т.д. Так же присутствует приложение для мобильных, которое повторяет личный кабинет пациента.

Для правильного функционирования серверной части системы требуется специализированные серверы.

Серверная часть сделана по принципу основной сервер + репликация. Характеристики машин должны быть одинаковыми. Для повышения отказоустойчивости советуется использовать внешнюю систему хранения данных.

Для рабочей станции требования представлены на рис. 1.

Минимальные системные требования:

1. Процессор Intel Celeron G440 или AMD Athlon 64 X2 3800+
2. Оперативная память DDR3 2Gb 1333Mhz
3. Тип накопителя – HDD, необходимый объем свободного пространства – 40Gb
4. Операционная система Microsoft Windows 7
5. Офисный пакет Apache OpenOffice 4.1.6 или Microsoft Office 2003
6. Монитор с соотношением сторон экрана 16:9, разрешение 1366x768 (WXGA) пикселей

Рекомендуемые системные требования (для анализа DICOM изображений – требуемые):

1. Процессор Intel i3 6th Gen или AMD Phenom 2 X4
2. Оперативная память DDR4 4Gb 1866Mhz
3. Тип накопителя – SSD, необходимый объем свободного пространства – 40Gb
4. Операционная система Microsoft Windows 10
5. Офисный пакет Apache OpenOffice 4.1.6 или Microsoft Office 2013
6. Монитор с соотношением сторон экрана 16:9, разрешение 1920x1080 (FullHD) пикселей (в случае с DICOM – DICOM Part 14 совместимые)

Рисунок 1 – системные требования рабочей станции qMS

Так же система предлагает использование тонких клиентов. Требования к монитору такие же, как и к рабочей станции. Перенаправление COM и USB портов. Возможность использования аппаратных ключей аутентификации.

1.1 Медицинская система Archimed+

Является одной из лучших медицинских систем на рынке.

Медицинская система для частных медицинских центров и стоматологических клиник.

С этой МИС работают все работники клиники. Медсестры, врачи, лаборанты, бухгалтер, регистратор.

Для каждой категории есть свой набор инструментов.

Для медрегистратора есть система быстрого поиска карты пациента, а также получение информации и внесение изменений. Присутствуют формы печати амбулаторной карты 025/у и стоматологической карты 043/у. Есть возможность печати любой документации и выгрузки базы в Excel. Присутствует функция отслеживания графика работы врачей. Особенно актуально для частных клиник или муниципальных, когда врач проходящий. Система записи пациентов помогает упростить этот процесс, особенно новым сотрудникам. Разработчики системы

стараятся сделать программу наиболее простой, удобной и интуитивно понятной. Вместе с программой предоставляется огромная база знаний со скриншотами и подробными инструкциями, а также видеоматериалами.

Для руководителя так же присутствует не мало инструментов.

Расчет заработной платы сотрудников. Позволяет создавать правила начисления заработной платы, устанавливать процент оплаты в зависимости от оказанных врачом услуг, рассчитывать оклад сотрудников и т.д.

Работа с филиалами. Наличие единой базы позволяет эффективно работать из одного места сразу со многими филиалами. Это позволяет получать отчетность со всех филиалов компании, возможность записать пациента любого филиала в любой филиал.

Так же присутствуют функции работы с агентами/партнерами, управление бизнес-процессами, клиентский сервис, предоставление 80 видов отчетности как на рабочем месте, так и посредством WEB. Присутствует маркетинговая и финансовая отчетности.

Для врача предоставлено множество удобных функций, ускоряющих его работу за счет уменьшения бумажной волокиты.

Электронная история болезни. Снимки, хронические болезни, посещенные специалисты, результаты анализов и исследований, все это хранится в электронной истории болезни и теперь не сможет потеряться как информация на бумажном носителе.

Выписка больничных листов облегчается тем, что присутствует автоподстановка данных. Есть возможность печати на стандартных бланках больничных листов и передачи электронного больничного листа. Ведется журнал больничных листов.

С помощью шаблонов врачу не придется печатать на клавиатуре вручную, ему достаточно будет выбрать конечный пункт в дереве фраз.

Удобство добавляют готовые шаблоны, благодаря которым у врача остается больше времени на пациентов и меньше времени уходит на бумажную работу. А

пациент получает на фирменном бланке всю необходимую информацию, без необходимости расшифровки почерка врача.

Весь перечень функций ArchiMed+ представлен на рисунках 2 и 3.

Регистратура Касса	Врач	Приказ 29Н
<ul style="list-style-type: none">База пациентовРасписание врачейЗапись пациентов на приемCALL- центрIP телефонияРабота с платными услугамиПодключение онлайн-кассАвансовые оплатыДисконтная системаПодарочные сертификатыБонусная системаРабота со страховыми компаниямиРасчет заработной платыОтчетность	<ul style="list-style-type: none">Электронная история болезниЭлектронная медицинская картаШаблоны, протоколы осмотровВыписка электронных больничных листовГотовые фразы, быстрый вводМедицинские стандартыПечать протоколов осмотров и рекомендацийЭлектронная подписьНаблюдение за пациентомНазначение леченияЛабораторные исследованияРекомендации	<ul style="list-style-type: none">Калькуляция профосмотровЗагрузка списка сотрудниковОсмотры врачейВредные факторыБегунокРезультаты анализовЗаключение профпатологаСправки, медицинские книжкиЗаключительный актПаспорт здоровьяИндивидуальные профосмотры

Рисунок 2 - перечень функций ArchiMed+

Маркетинг и отчетность	Лаборатория	Другие функции
<ul style="list-style-type: none">Онлайн запись с сайтаТелеграмм бот (запись в клинику)Личный кабинет пациентаОплата с сайта (Яндекс.Касса)Сайт для вашей клиникиТерминалы самозаписиВывод информации на плазменную панельСМС-рассылкаEmail -рассылкаCRM - системаМаркетинговая отчетностьУправленческая отчетностьФинансовая отчетностьОтчетность по зарплатеWEB - отчетностьСтатистика по пациентам, диагнозам, возрасту	<ul style="list-style-type: none">Забор анализовКонтроль норм анализовРезультаты анализов на сайте клиникиСвязь с лабораторным оборудованиемИнтеграции с внешними лабораториями (ХЕЛИКС, ИНВИТРО, СИТИЛАБ, КДЛ, ЛИС АЛЬФАЛАБ)Журнал лабораторных исследованийОтчетность	<ul style="list-style-type: none">Подключение филиаловВзаиморасчеты с юридическими лицамиВзаиморасчеты со страховыми компаниямиВнутренний онлайн-чатСвязь с диагностическим оборудованиемСкладской учет материаловAPI (в разработке)Интеграция с RoiStat (в разработке)

Рисунок 3 - перечень функций ArchiMed+

Цена зависит от количества рабочих мест и функций МИС.

Поэтому система доступна как частным врачам, так и крупным сетевым клиникам. Для каждого подбирается оптимальная конфигурация.

Настоящую стоимость определить не удалось. Для того чтобы что-то узнать, необходимо ответить на 4 вопроса и указать свой номер телефона. Иначе получить информацию о стоимости не представляется возможным. Зато говорится, что никакой абонентской платы нет, система покупается раз и навсегда.

Можно получить демо версию программы.

Внедрение системы происходит в основном удаленно. Работы по настройке и установке производятся сотрудниками системы. После внедрения предоставляется сотрудник. Он проводит обучение персонала и настройку программы до тех пор, пока клиника не начнет работать в системе полноценно.

1.2 Вывод

На сегодняшний день существует множество информационных медицинских систем, которые предлагают богатый функционал. Связано это с потребностью ускорения документооборота, так как эта характеристика непосредственно влияет на продуктивность клиники и скорость принятия решений руководителям. Так же обусловлено скоростью и качеством обслуживания пациентов внедрением электронных карт.

Централизованное хранение данных помогает собирать данные с любых филиалов клиники, и посредством технологий интернет позволяет получать эту информацию, как руководителю удаленно, так и регистратору другого филиала, даже если этот человек там ни разу не был. Это сильно уменьшает вероятность утери данных, если бы это было на бумажном носителе и пришлось бы выдавать пациенту карту на руки. Врач может видеть необходимую информацию прямо со своего компьютера и не разбираться в чем-то почерке.

Оперативное получение актуальной информации руководителями клиник помогает принимать необходимые эффективные решения по управлению клиникой. Помогает бухгалтерским отделам вести финансовый учет проще и

эффективнее. Рассчитывать зарплату работникам становится на много проще если присутствуют соответствующие функции.

4 РАЗРАБОТКА ДИАГРАММЫ ПОТОКОВ ДАННЫХ

Средством разработки проекта были выбраны диаграммы потоков данных.

При разработке диаграммы потоков данных определены следующие сущности:

- Регистратор
- Клиент
- Процедуры
- Материалы
- Доктор

На рисунке 4 представлен общий вид взаимодействия клиники с остальными внешними объектами. Входящие и выходящие потоки данных показаны стрелками с подписями. В ходе разработки данной диаграммы были определены сущности:

- Clinic – непосредственно сама внутренняя сущность, в которую поступает информация

- Клиент – пациент, который приходит получить услуги, сообщает какую-то информацию о себе

- Доктор – это персонал, который работает в клинике, дает информацию о себе

- Регистратор – это тот, кто может вносить данные, редактировать их, удалять, изменять для корректного ввода.

- Материалы – информация о требуемых для клиники материалах и их учете. Выходными данными будут служить данные о клиенте, прайс лист, и т.п.

Сущности показаны в виде прямоугольников с подписями внутри.

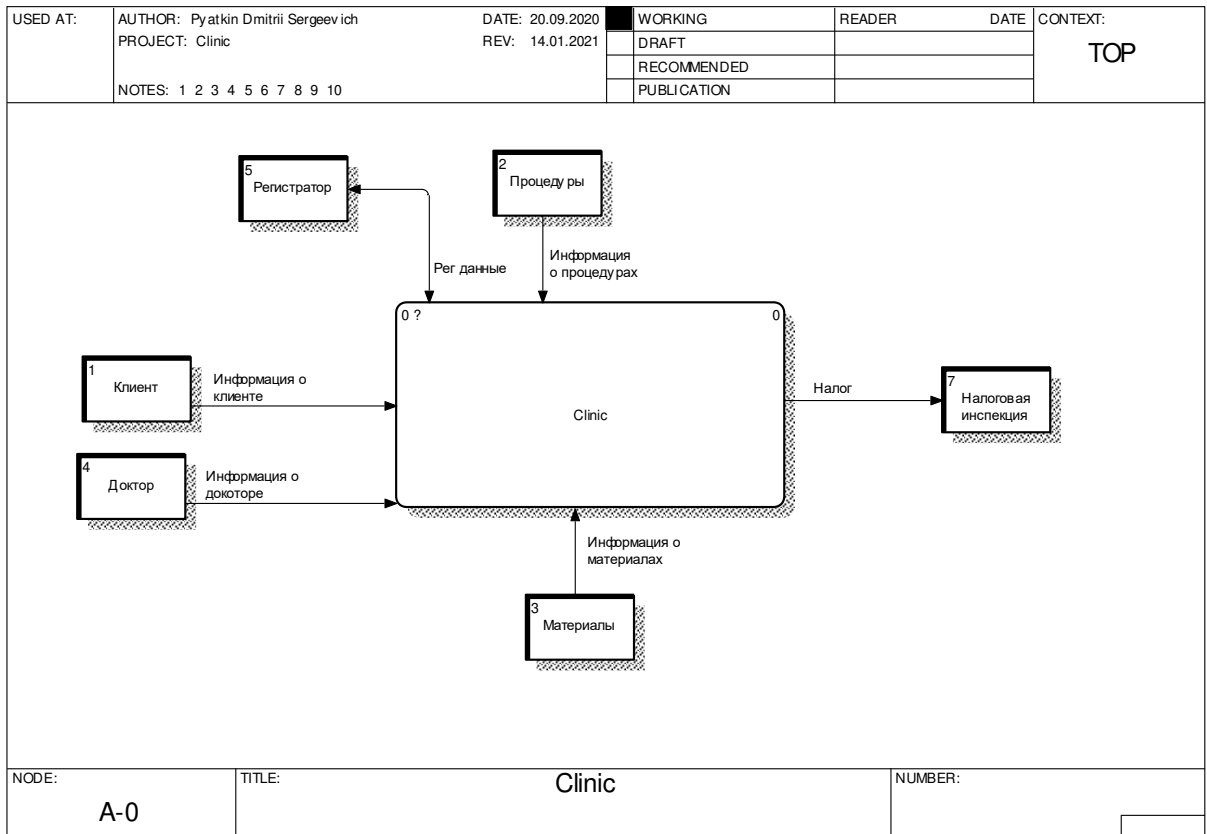


Рисунок 4 – общий вид.

На рисунке 5 показаны внутренние потоки данных.

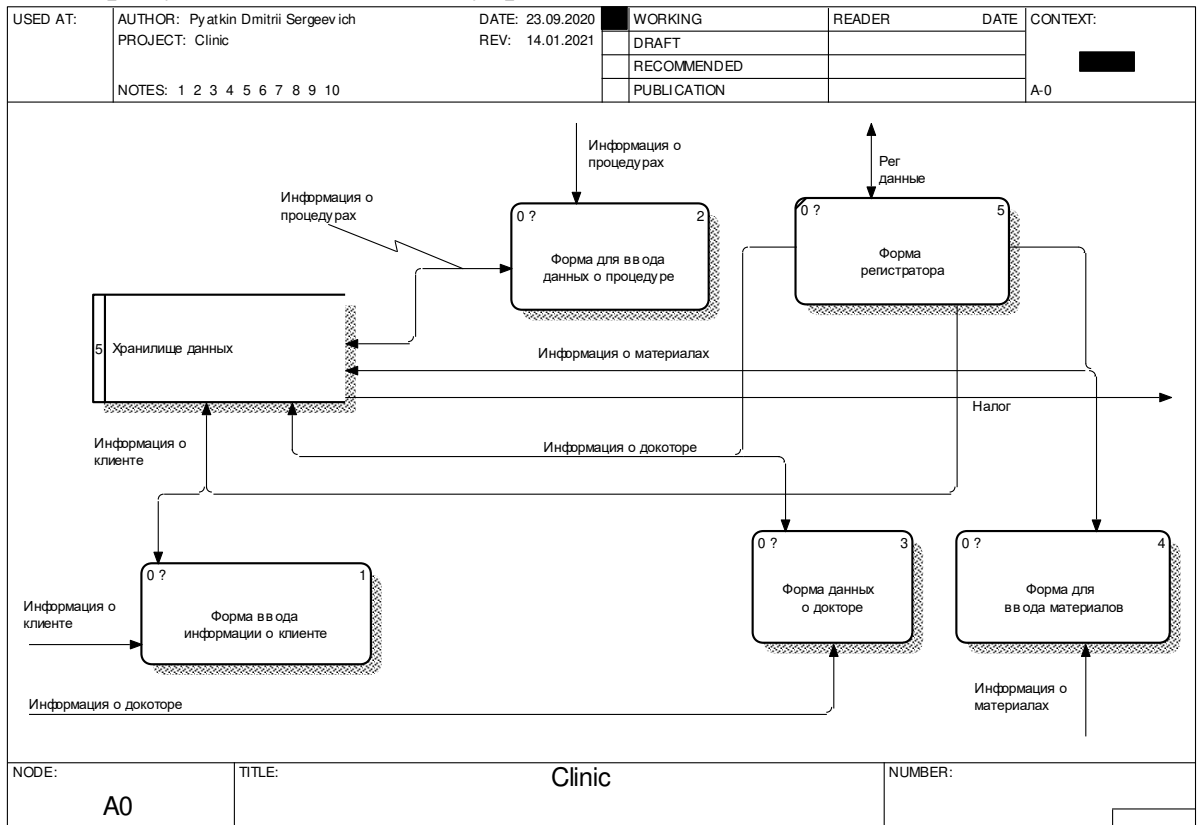


Рисунок 5 – потоки данных в клинике.

На рисунке 6 потоки данных для формы ввода информации о клиенте.

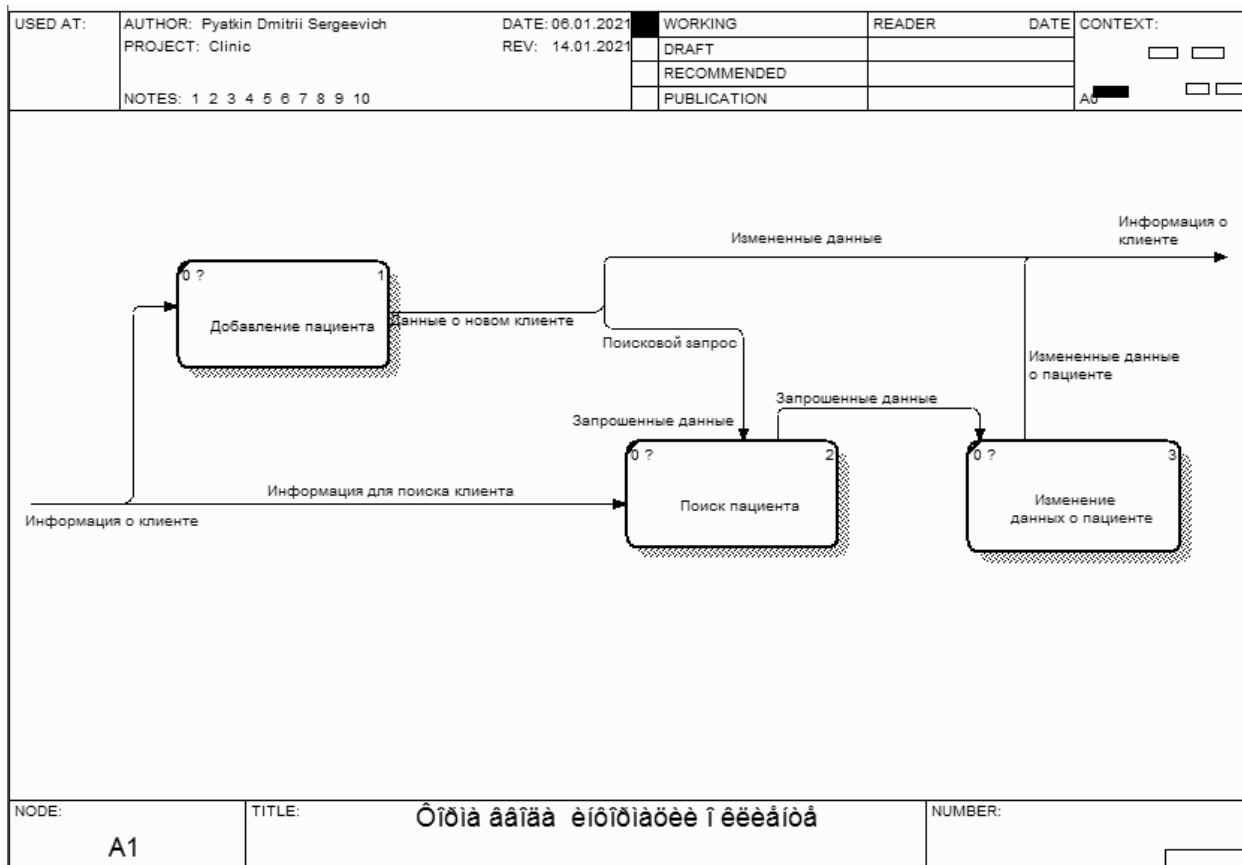


Рисунок 6 – форма ввода информации о клиенте.

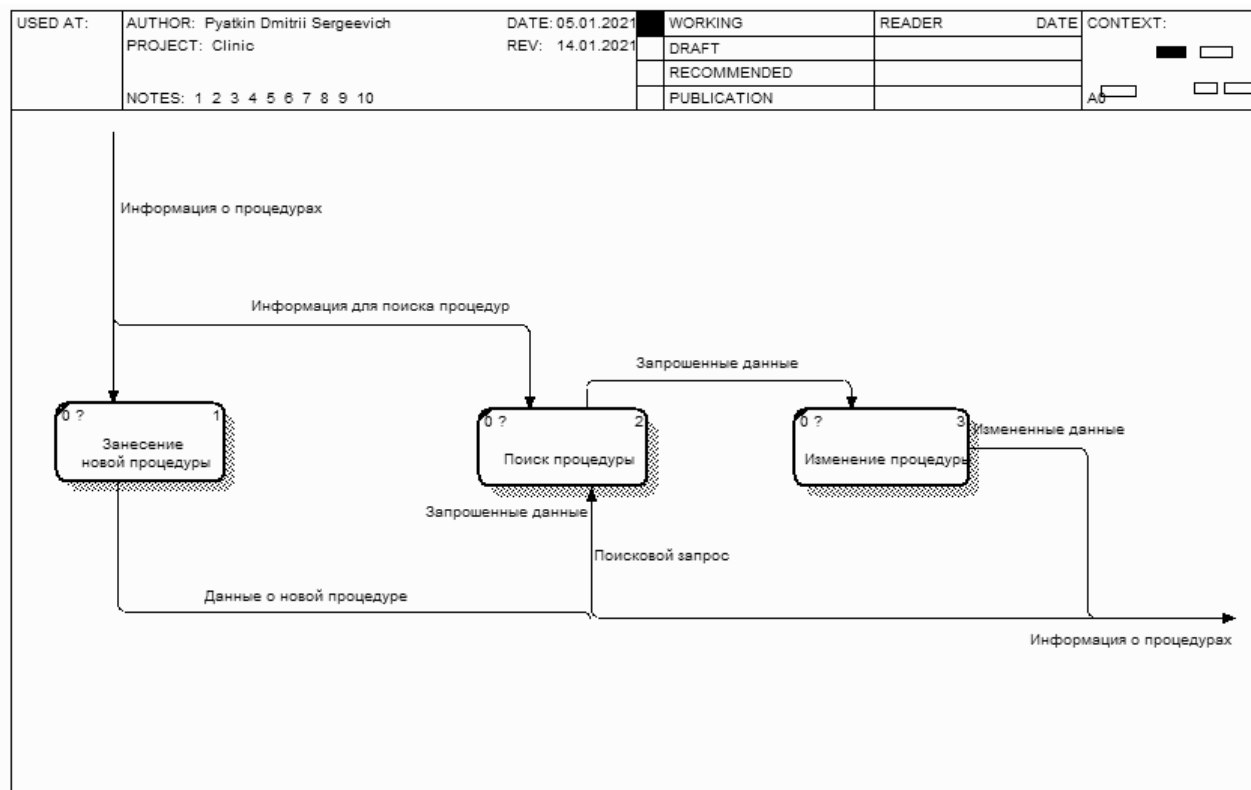


Рисунок 7 – форма ввода информации о процедурах

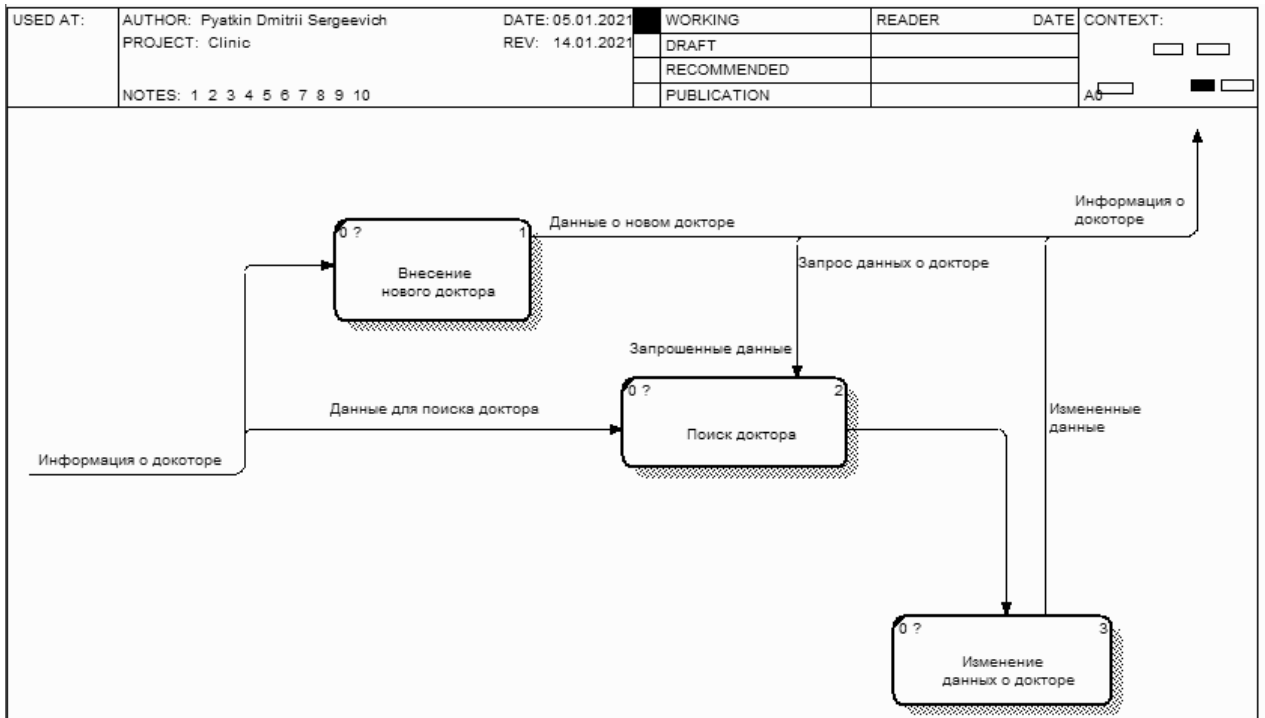


Рисунок 8 – форма ввода информации о врачах

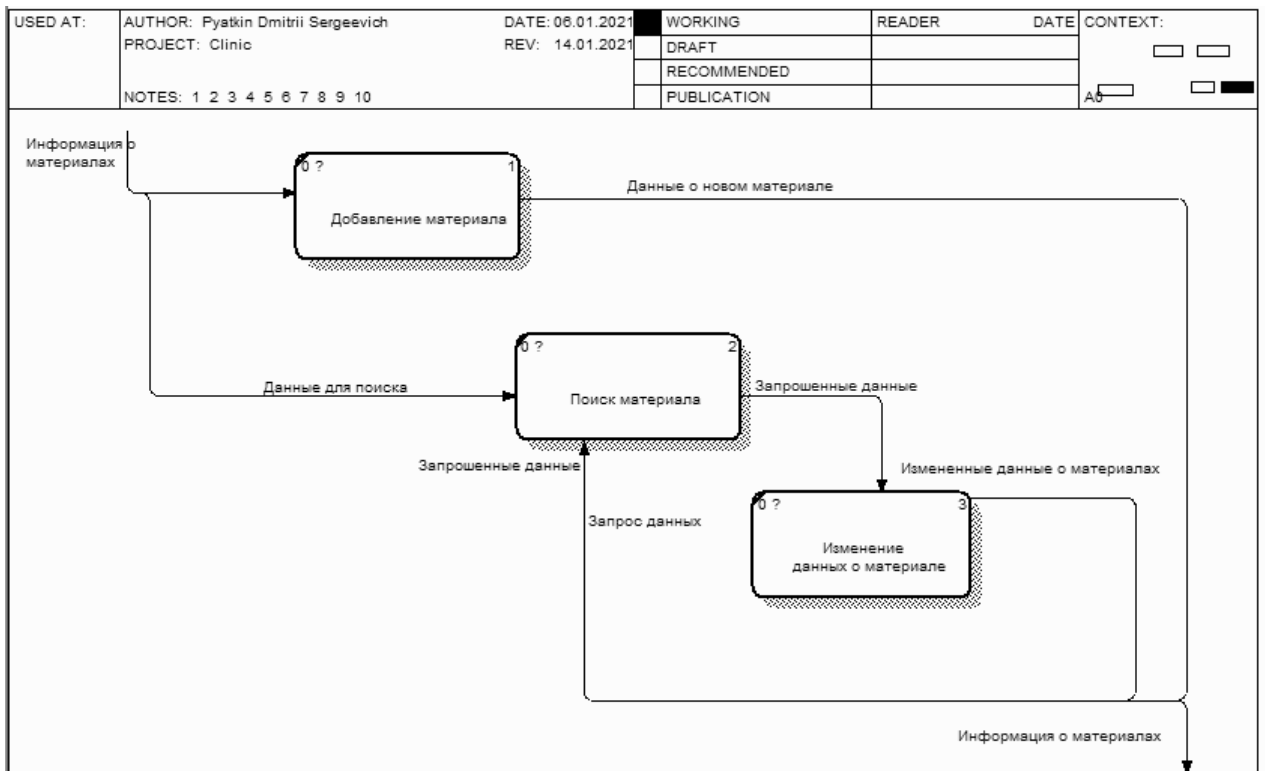


Рисунок 9 – форма ввода информации о материалах

5 ПРАКТИЧЕСКАЯ ЧАСТЬ

5.1 Разработка базы данных

Для разработки базы данных требуется программное обеспечение Microsoft SQL Server Management Studio с помощью которой и будет происходить создание базы данных и всевозможные манипуляции.

В ходе разработки определены и созданы следующие таблицы:

1. Должности
2. ЕдИзм
3. Материалы
4. Материалы_на_процедуру
5. Отделения
6. Пациенты
7. Персонал
8. Процедуры
9. Роли
10. Журнал процедур

Для таблицы «Должности» созданы столбцы, в которых определен тип информации, которая будет храниться. Отображено в таблице 5.1.

Таблица 5.1 – Типы данных столбцов таблицы «Должности»

Имя столбца	Тип данных
id	int
Должность	nvarchar(MAX)
Ставка	float
Должностные обязанности	nvarchar(MAX)
Комментарии	nvarchar(MAX)
id_Отделения	int

Для столбцов «Должностные обязанности» и «Комментарии» разрешены пустые поля, для остальных определено, что они пустыми быть не могут.

Рассмотрим таблицу «ЕдИзм». Это таблица – справочник, которая хранит в себе данные о различных единицах измерения, например, штуки, литры, коробки и так далее. Данная таблица нужна в проекте для того, чтобы уменьшить объемы данных при хранении базы данных. На нее будет ссылаться таблица «Материалы». В таблице 5.2 указаны ее столбцы с отображением типов данных.

Таблица 5.2 – Типы данных столбцов таблицы «ЕдИзм»

Имя столбца	Тип данных
id	int
Name	nvarchar(10)

Таблица 3 отображает в себе типы данных столбцов таблицы базы данных «Материалы», которая содержит в себе наименования и количество имеющихся материалов.

Таблица 5.3 – Типы данных столбцов таблицы «Материалы»

Имя столбца	Тип данных
id	int
Название	nvarchar(MAX)
ЕдИзм	int
Количество	float
Цена	money

Таблица под названием «Материалы_на_процедуру» хранит в себе информацию о затрачиваемых материалах на каждую процедуру. В данной таблице хранятся только id процедур, материалов и их количество. Данные о типах данных столбцов показаны в таблице 5.4.

Таблица 5.4 – Типы данных столбцов таблицы «Материалы_на_процедуру»

Имя столбца	Тип данных
id	int
ид_процедуры	int
ид_материала	int
кол_материала	float

Таблица «Отделения» хранит наименования отделений клиники. Типы данных столбцов представлены в таблице 5.5.

Таблица 5.5 - Типы данных столбцов таблицы «Отделения»

Имя столбца	Тип данных
id	int
name	nvarchar(30)

Вся информация о пациентах хранится в таблице под названием «Пациент». Включает в себя паспортные данные о пациентах клиники, данные о полисе, адресе проживания, номер телефона. Поля с соответствующими типами данных указаны в таблице 5.6.

Таблица 5.6 - Типы данных столбцов таблицы «Пациент»

Имя столбца	Тип данных
id	int
Фамилия	nvarchar(MAX)
Имя	nvarchar(MAX)
Отчество	nvarchar(MAX)
Паспорт_серия_номер	char(10)
Паспорт_кем_выдан	nvarchar(MAX)
Город	nvarchar(MAX)
Улица	nvarchar(MAX)
Дом	nvarchar(16)
Квартира	char(5)
Полис	char(16)
Телефон	char(12)
Дата_рождения	date

Таблица «Персонал» так же включает в себя паспортные данные, но уже о сотрудниках клиники. Каждый сотрудник имеет должность. Должность не указывается непосредственно, а указывается id должности, соответствующий ее названию. В таблице 5.7 указаны типы данных столбцов.

Таблица 5.7 - Типы данных столбцов таблицы «Персонал»

Имя столбца	Тип данных
id	int
Фамилия	nvarchar(MAX)
Имя	nvarchar(MAX)
Отчество	nvarchar(MAX)
Дата_рождения	date
Паспорт_серия_номер	char(10)
Паспорт_кем_выдан	nvarchar(MAX)
Город	nvarchar(MAX)
Район	nvarchar(MAX)
Улица	nvarchar(MAX)
Дом	nvarchar(MAX)
Квартира	char(5)
Полис	char(16)
Телефон	char(12)
ИНН	char(13)
Образование	nvarchar(MAX)
Комментарий	nvarchar(MAX)
Должность	int

В таблице «Процедуры» хранятся названия непосредственно процедур и их стоимость. Типы данных и названия столбцов указаны в таблице 5.8.

Таблица 5.8 - Типы данных столбцов таблицы «Процедуры»

Имя столбца	Тип данных
id	int
Название	nvarchar(MAX)
Стоимость	money

Таблица под названием «роли» хранит в себе информацию с логинами и паролями. С помощью нее будет определяться доступный функционал, как сайта,

так и клиентского приложения. В таблице 5.9 показаны типы данных столбцов таблицы «роли».

Таблица 9 - Типы данных столбцов таблицы «роли»

Имя столбца	Тип данных
id	int
log_in	nvarchar(MAX)
passw	nvarchar(MAX)

Таблица «Журнал_процедур» хранит данные о том, когда и какому пациенту проводились или будут проводиться процедуры, информацию о том, кто провел определенную процедуру. Имена столбцов с типами данных отображены в таблице 5.10.

Таблица 5.10 - Типы данных столбцов таблицы «Журнал_процедур»

Имя столбца	Тип данных
id	int
id_пациента	int
id_персонала	int
id_процедуры	int
Дата	date
Статус	bit

Диаграмма, на которой видны все таблицы базы данных и их связи, показаны на рисунке 10.

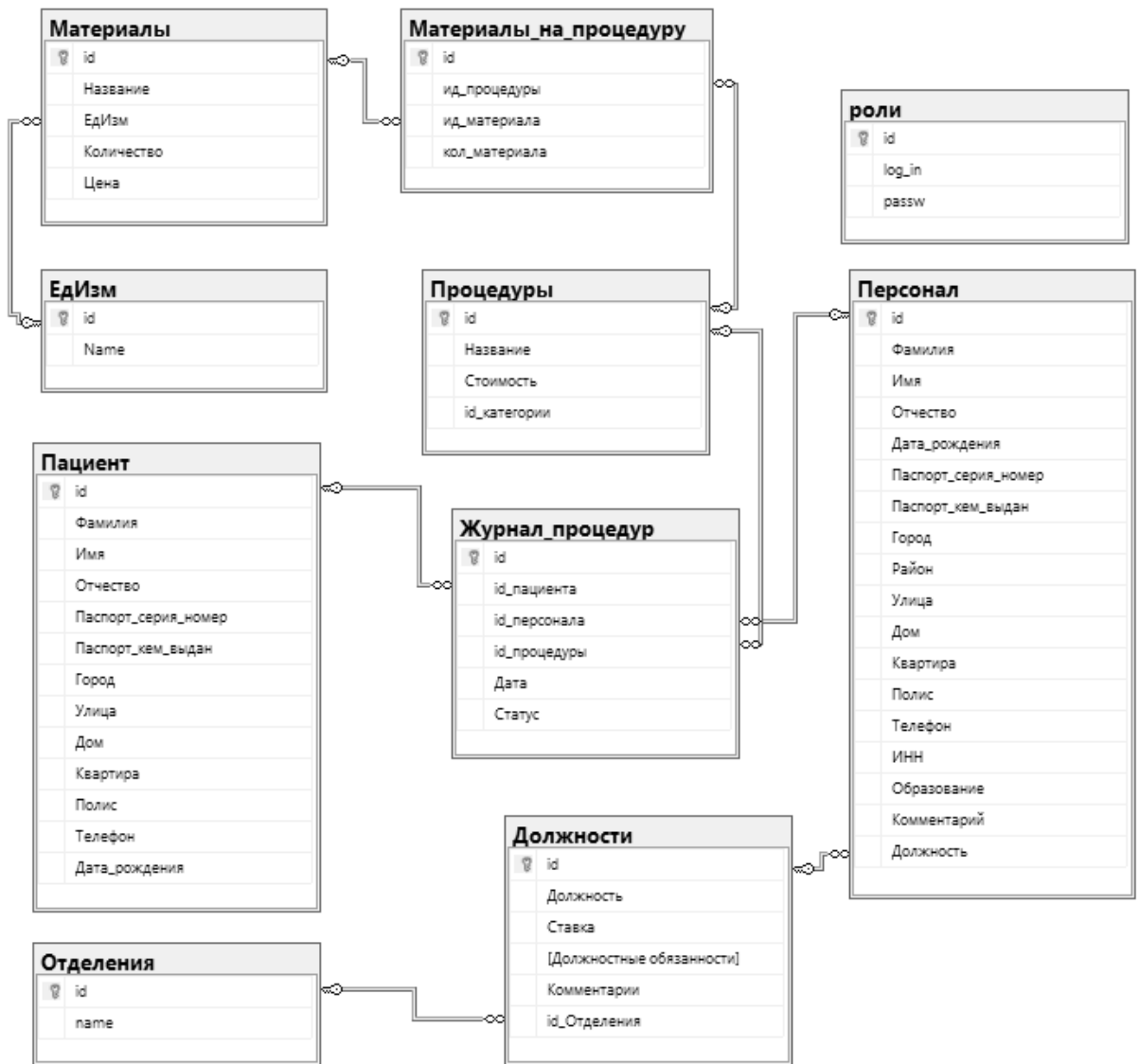


Рисунок 10 – диаграмма базы данных Clinic.

5.2 Разработка приложения

5.2.1 Авторизация

В ходе разработки приложения были реализованы отдельные классы подключения к базе данных. Сделано это для повышения безопасности и удобства внедрения в рабочую среду, так как можно быстро и удобно сменить только строку подключения к базе.

```
public static SqlConnection GetDBConnection()
{
    string datasource = @"DESKTOP-A9Q0I3R";
    string database = "Clinic";
    string user = "DESKTOP-A9Q0I3R/Дмитрий";
    return DBSQLServerUtils.GetDBConnection(datasource, database,
user);
}
public static SqlConnection
    GetDBConnection(string datasource, string database, string user)
{
    string connString = @"Data Source=" + datasource + ";Initial
Catalog=" + database + ";Integrated Security=True;User ID=" + user;
    SqlConnection conn = new SqlConnection(connString);
    return conn;
}
```

Для доступа к программе разработана система авторизации пользователей. Для того чтобы получить доступ к приложению, необходимо ввести логин и пароль. За авторизацию отвечает класс `Autorized.cs`. Для авторизации в базе данных используется хранимая процедура, в которую из данного класса передаются параметры в виде логина и пароля, обратно возвращается `id` пользователя, по которому и определяется доступный функционал. Листинг кода этого класса приведен ниже.

```
SqlConnection conn = DBUtils.GetDBConnection();
    conn.Open();
    SqlCommand cmd = new SqlCommand("log_in", conn);
    cmd.CommandType = CommandType.StoredProcedure;
```

```

cmd.Parameters.AddWithValue("@login", log);
cmd.Parameters.Add("@passw", SqlDbType.NVarChar).Value =
passw;

// Добавить параметр
cmd.Parameters.Add(new SqlParameter("@id", SqlDbType.Int));
// Зарегистрировать параметр
cmd.Parameters["@id"].Direction =
ParameterDirection.Output;

cmd.ExecuteNonQuery();
// Получить выходные значения.
id = (int)cmd.Parameters["@id"].Value;
conn.Close();
cmd.Dispose();

```

Форма авторизации показана на рисунке 11.

The image shows a standard Windows-style dialog box with the title 'Авторизация пользователя'. Inside the dialog, there are two text input fields. The first is labeled 'Логин' and the second is labeled 'Пароль:'. Below these fields is the instruction 'Введите логин и пароль'. At the bottom of the dialog, there are two buttons: 'Вход' on the left and 'Выход' on the right. The dialog has standard window controls (minimize, maximize, close) in the top right corner.

Рисунок 11 – форма авторизации пользователя.

5.2.2 Роли

Роль администратора в приложении включает весь функционал приложения. Администратор может управлять данными пациентов, персонала, материалами, процедурами, должностями и так далее. Внешний вид окна приложения можно увидеть на рисунке 12.



Рисунок 12 – вид окна приложения.

Во вкладке «Файл» доступны функции выхода из приложения и смены пользователя. При смене пользователя происходит закрытие формы просмотра и редактирования данных и открытие формы авторизации пользователей с возможностью войти в приложение под другим пользователем.

Пункт меню «Пациенты» открывает форму с таблицей, которая содержит в себе данные о пациентах с возможностью их редактирования. Таблица с пациентами показана на рисунке 13.

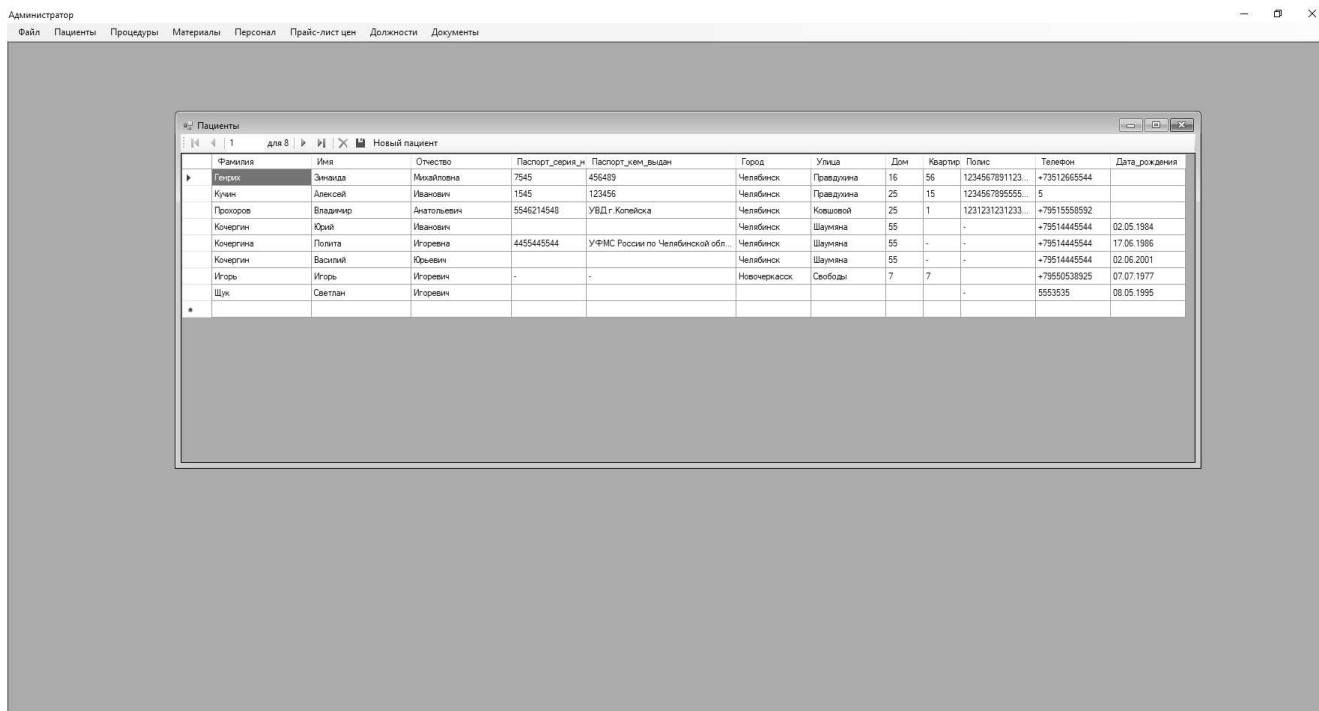


Рисунок 13 – форма «Пациенты»

При нажатии на кнопку «Новый пациент» открывается окно создания с пустыми полями для ввода данных. Данное окно показано на рисунке 14.

Рисунок 14 – форма ввода нового пациента.

Пункт меню процедуры открывает форму в которой содержатся 2 таблицы. Первая таблица содержит в себе данные о процедуре, вторая показывает расходные материалы, которые требуются для выполнения процедуры. Данные обеих таблиц можно редактировать непосредственно в таблице. Вид данной формы показан на рисунке 15.

The screenshot shows a software window titled 'Процедуры' (Procedures). It contains two tables. The first table lists various medical procedures and their costs. The second table, located below the first, lists materials used for a procedure, including their names and quantities. There are also buttons for 'Сохранить' (Save) and 'Удалить' (Delete) between the two tables.

Название	Стоимость
УЗИ органов брюшной полости	700,0000
УЗИ почек	600,0000
УЗИ желудка	700,0000
Первичный прием врача-терапевта	1300,0000
Вторичный прием врача-терапевта	700,0000
Прием (осмотр, консультация) врача-офтальмолога первичный	1648,0000
Прием (осмотр, консультация) врача-офтальмолога повторный	979,0000
Профилактический прием врача-офтальмолога	527,0000
Прививка от covid2019	1900,0000

Название	Материал	Количество
УЗИ органов брюшной полости	Бахилы Прочные, 1000шт	2
УЗИ органов брюшной полости	Гель для УЗИ "Медигель" 0,25 кг средней вязкос...	0,5
УЗИ органов брюшной полости	Перчатки смотровые нестерильные XL, 50 пар	1
УЗИ органов брюшной полости	Простыня 70*80 см, нестерильная, SMS голубая/б...	1
*		

Рисунок 15 – форма «Процедуры»

Пункты меню «Материалы», «Персонал», и «Должности» открывают формы, которые содержат таблицы с соответствующей информацией с возможностью добавления и редактирования записей. Эти формы показаны на рисунке 16.

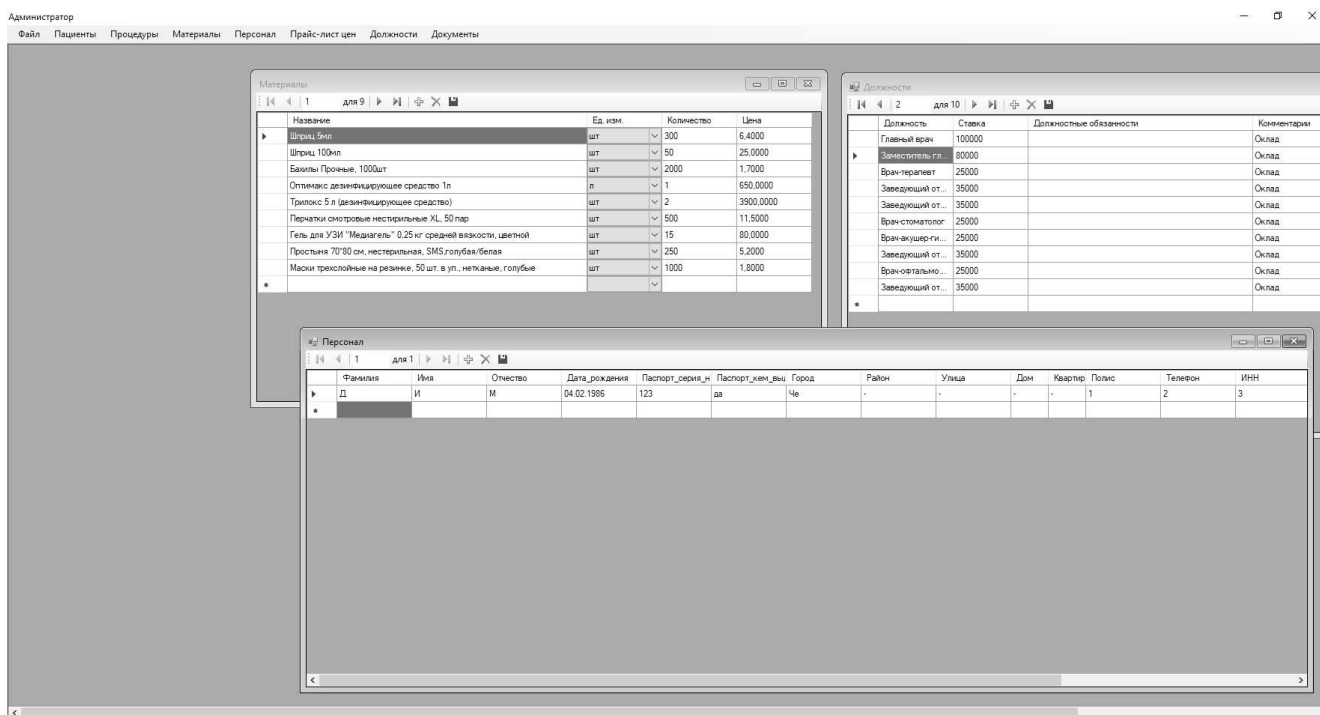


Рисунок 16 – формы «Материалы», «Персонал», и «Должности».

«Прайс лист цен» открывает форму, в которой указаны названия процедур и их стоимость. Есть возможность печати данных, которые выводятся в документ Microsoft Word. Данная форма показана на рисунке 17.

Цены на услуги

Услуга	Стоимость
УЗИ органов брюшной полости	700,0000
УЗИ почек	600,0000
УЗИ желудка	700,0000
Первичный прием врача-терапевта	1300,0000
Вторичный прием врача-терапевта	700,0000
Прием (осмотр, консультация) врача-офтальмолога первичный	1648,0000
Прием (осмотр, консультация) врача-офтальмолога повторный	979,0000
Профилактический прием врача-офтальмолога	527,0000
Прививка от covid2019	1900,0000
Прогревание	350,0000
Антицеллюлитный	500,0000
Прививка от краснухи	190,0000
*	

Печать

Рисунок 17 – форма «Прайс лист цен».

При нажатии на пункт меню «Запись на процедуру», открывается окно, в котором отображена информация о пациентах, которые записаны на процедуры, либо приемы и уже проведенные процедуры. Данное изображено на рисунке 18.

№	Пациент фамилия	Имя	Услуга	Сотрудник	Стоимость	Дата	Статус
3	Генрих	Зинаида	Вторичный прием врача-терапевта	И	700,0000	26.06.2021	<input checked="" type="checkbox"/>
4	Кочергин	Юрий	УЗИ почек	И	600,0000	27.06.2021	<input checked="" type="checkbox"/>
6	Кочергина	Лолита	Прививка от covid2019	И	1900,0000	27.06.2021	<input checked="" type="checkbox"/>
7	Кучин	Алексей	Первичный прием врача-терапевта	И	1300,0000	28.09.2021	<input type="checkbox"/>
8	Прохоров	Владимир	УЗИ желудка	И	700,0000	27.06.2021	<input type="checkbox"/>
*							<input type="checkbox"/>

Рисунок 18 – форма «Запись на процедуру».

Далее идет пункт меню «Документы». Он включает в себя некоторый перечень документов, которые необходимы для работы с пациентами. Это согласие на обработку персональных данных, как форма для заполнения, так и пример. Информированное медицинское вмешательство и договор на оказание услуг.

Для других пользователей функционал усечен в рамках потребностей.

6 РАЧЕТ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ОТ ВНЕДРЕНИЯ ПРОГРАММНОГО ПРОДУКТА

Разработка программного обеспечения сопряжена с большим вложением денежных средств, как на приобретение техники и программного обеспечения, так и на разработку проектов, выполнение подготовительных работ и т.д. Поэтому необходимо экономическое обоснование целесообразности ведения разработок.

Себестоимость программного продукта - это выраженные в денежной форме текущие расходы предприятия на его производство и сбыт. Расходы на производство образуют производственную (заводскую) себестоимость, а расходы на производство и сбыт - полную себестоимость. Расчет себестоимости программного продукта по статьям затрат называется калькуляцией. Калькулирование себестоимости программного продукта осуществляется в соответствии с «Типовым положением по планированию, учету и калькулированию себестоимости продукции (работ, услуг) в промышленности»

Главным при разработке программного обеспечения является обеспечение максимальной экономической эффективности, т.е. проектирование с минимальными затратами труда и денежных средств.

6.1 Затраты на разработку продукта

В данном разделе определяются необходимые инвестиции для разработки и реализации программного продукта. Для этого необходимо произвести расчет единовременных (таблица 6.1) и текущих (таблица 6.5) затрат.

Таблица 6.1 – Расчет единовременных затрат

№	Наименование технического средства и ПО	Модель, Марка	Стоимость, р.
1	Персональный компьютер	ACER Aspire TC-390	30990
2	Монитор	SAMSUNG C24RG50FQI 23.5"	12990
3	Клавиатура	A4TECH KV-300H	1990
4	Компьютерная мышь	Canyon 3145	900
5	Операционная система	Windows 10 Home; 64 бит	8900
6	Компьютерный стол	Каспер Лоредо,	5780
7	Компьютерное кресло	Бюрократ СН-1300N	2270
Итого			63820

Общая сумма единовременных затрат составляет 63 820р. В нее входят затраты на приобретение программного и технического обеспечения.

Также необходимо произвести расчет заработной платы для сотрудников. Для этого необходимо составить график основных плановых работ, он представлен в таблице 6.2.

Таблица 6.2 Календарный план-график основных этапов выполнения дипломной работы

Наименование этапа	Дата начала	Длительность, недель	Длительность в часах
Подготовительный этап	5 апреля 2021 г.	1 неделя	40
Теоретическая разработка	12 апреля 2021 г.	1 неделя	40
Разработка базы данных	26 апреля 2021 г.	2 недели	80
Разработка приложения	17 мая 2021 г.	2 недели	80
Расчеты	6 июня 2021 г.	1 неделя	40
Итого затрат на полную реализацию проекта		7 недель	280

Расчеты затрат на разные этапы разработки рассчитываются по формуле (1)

$$Z_i = T_z * T_c, \quad (6.1)$$

где T_z - время затраченное на выполнение проекта;

T_c - Тарифная ставка.

Расчет заработной платы представлен в таблице 6.3

Таблица 6.3 Расчет заработной платы

Наименование работ	Обозначение	Тарифная ставка	Длительность работ (T_z)	Стоимость работ (Z_i), Р
Подготовительный этап (Зп)	Зп	110	40	4 400,00
Теоретическая разработка (Зт)	Зт	110	40	4 400,00
Разработка базы данных (Збд)	Збд	210	80	16 800,00
Разработка приложения (Зрп)	Зрп	210	80	16800,0
Расчеты, резюмирование (Зрр)	Зрр	120	40	4 800,00
Общие затраты на оплату труда			280	47 200,00

Таким образом, полная оплата труда складывается из сумм оплаты всех этапов разработки программного продукта по формуле (2).

$$Z_{\text{сум}} = Z_{\text{п}} + Z_{\text{т}} + Z_{\text{ср}} + Z_{\text{рп}} + Z_{\text{уч}} + Z_{\text{рр}}, \quad (6.2)$$

$$Z_{\text{сум}} = 4400 + 4400 + 16800 + 16800 + 4800 = 47200 \text{ руб.}$$

Неотъемлемой статьёй расходов являются затраты на накладные расходы, которые связаны с проектированием и отладкой программного продукта. Расчет по данному пункту представлен в таблице 6.4.

Таблица 6.4 – Потребности

Материалы	Потребность	Стоимость одной ед., руб.	Общая стоимость, руб.	Комментарий
Бумага	100 листов	0,5	50,00	Бумага для печати
Расходные материалы для принтера	1 шт	1050	1 050,00	Печать необходимых данных
Перезаписываемый флэш-накопитель	1 шт	510	510,00	Резервирование
Сеть интернет	2 месяца	450	900,00	Поиск информации
Итого			2 510,00	

Для выполнения работ необходимо материалов на сумму 2 510 р.

Оплата за пользование электричеством составляет

$$Z_{\text{э}} = 0,5 \text{ кВт} \cdot 280 \text{ часов} \cdot 3,36 \text{ руб} = 470,4 \text{ руб.}$$

Текущие затраты (себестоимость) (С) включают затраты на постановку задачи, изучение предметной области, разработку и оснащение необходимым содержанием, рассчитываются по формуле:

$$C = Z_{\text{сум}} + N_{\text{з}} + Z_{\text{э}} + Z_{\text{ф}}, \quad (6.3)$$

где $N_{\text{з}}$ - затраты накладные;

$Z_{\text{ф}}$ - Отчисления в фонды с заработной платы труда.

Отчисления в фонды с заработной платы труда рассчитываются по формуле

$$Z_{\text{ф}} = 30,2\% \cdot Z_{\text{сум}}, \quad (6.4)$$

$$47200 \cdot 0,302 = 14254,4$$

Все учтенные расходы представлены в общей таблице 6.5

Таблица 6.5 – Текущие затраты

Затраты на разработку ПП	Сумма
Заработная плата	47200
Отчисления в фонды с заработной платы труда (ФФОМС, ПФР, ФСС)	14254,4
Накладные расходы	2510
Расходы на электричество при пользовании вычислительной техникой	470,4
Итого затрат	64 434,8

После расчета единовременных и текущих затрат необходимо составить общую таблицу потребности в инвестициях

Таблица 6.6 – Текущие затраты

Единовременные затраты	1 квартал	2 квартал	3 квартал	4 квартал
		63820		
Заработная плата	47200	47200	47200	47200
Отчисления в фонды с заработной платы труда (ФФОМС, ПФР, ФСС)	14254,4	14254,4	14254,4	14254,4
Накладные расходы	2510	2510	2510	2510
Расходы на электричество при пользовании вычислительной техникой	470,4	470,4	470,4	470,4
Итого затрат	128 254,8	64 434,8	64 434,8	64 434,8

6.2 Определение доходов от внедрения в работу программного продукта.

При отсутствии программного обеспечения в клинике на данный момент уходит больше времени на ручной ввод данных пациентов, учет материалов. Все это может занять довольно долгое время. Возьмем зарплату администратора клиники в 24000 рублей в месяц, учтем затраты времени на ручной ввод данных 2 часа в день. Получаем зарплату в 800 рублей в день и с учетом того, что клиника работает ежедневно, то за месяц выходит 2,5 дня рабочего времени, что эквивалентно ($2,5 \cdot 800 = 2000$) 2000 рублей в месяц или 6000 рублей за квартал. При филиальном устройстве компании, при 5 филиалах данный показатель увеличится до 30000 рублей.

За счет внедрения информационной системы должна ускориться работа персонала, повыситься качество работы и увеличиться пропускная способность клиники, из-за чего увеличится прибыль.

Доходы от внедрения информационной системы отображены в таблице 6.7

Таблица 6.7 – Доходы

Доходы/Месяцы	1 квартал, руб.	2 квартал, руб.	3 квартал, руб.	4 квартал, руб.
Снижение затрат на ручную обработку информации, руб.	30000	30000	30000	30000
Привлечение дополнительных клиентов, руб.	30000	35000	40000	45000
Итого доходов тыс. руб.:	36000	41000	46000	51000

Расчет экономической эффективности проводится по следующим показателям:

- чистый дисконтированный доход (ЧДД) или интегральный эффект
- индекс доходности (ИД)

- внутренняя ставка доходности (ВСД)
- срок окупаемости.

Чистый дисконтированный доход - превышение интегральных результатов над интегральными затратами. Определяется как сумма текущих эффектов за весь расчетный период, приведенная к начальному шагу.

Метод дисконтированных денежных потоков предполагает анализ потоков инвестиционного капитала, причем как затратных, так и прибыльных, с учетом их временно-стоимостной оценки.

Выбор данного метода был обусловлен следующими причинами:

- необходимость инфляционной корректировки показателей;
- обеспечение наглядности и простоты расчета срока окупаемости;
- стабильность и предсказуемость денежных потоков;
- равномерность денежных потоков.

Этот метод нашел широкое применение в зарубежной практике в ходе оценки инвестиций в материальные и нематериальные активы, особенно в случаях нестабильной макроэкономической ситуации (инфляция, возможность кризисных ситуаций и т. д.).

Суть этого метода заключается в отнесении предполагаемых денежных потоков (и положительных, и отрицательных) к временным интервалам, начиная с момента начала инвестирования. Как правило, при малых показателях рентабельности продукта сроком анализа этого метода выбирают нормативный «срок жизни» продукта, или срок окупаемости продукта-субститута. В данном случае для установления срока окупаемости достаточно ограничиться сроком в 3 года.

К недостаткам данного метода можно отнести то, что он не позволяет учесть случайные риски (как внешние, так и внутренние), а также известную условность расчета срока окупаемости, связанную с выбором ставки дисконтирования.

Чистый дисконтированный доход определяется как сумма текущих эффектов за весь расчетный период по формуле 6.6:

$$ЧДД = \sum_{t=1}^5 (D_t - P_t) \cdot \frac{1}{(1 + \alpha)^t} \quad (6.6)$$

Где D_t - результаты, достигаемые на t -ом шаге расчета;

P_t - затраты, осуществляемые на том же шаге;

$D_t - P_t$ - эффект достигаемый на t -ом шаге.

$1/(1 + \alpha)^t$ - коэффициент приведения по времени результатов и затрат;

α - норма дисконта, равная приемлемой для инвестора норме дохода на капитал -0,14.

Предполагаемые доходы от продаж (D) определяются по формуле 6.7:

$$D = Ц_{пр} * N \quad (6.7)$$

где $Ц_{пр}$ - цена продажи ПП, руб;

N - объем продаж по периодам в соответствии с исследованиями рынка, шт.

Расходы (P) включают, кроме текущих затрат, расходы на тиражирование и рекламу, формулы 6.8 и 6.9:

$$P = З_{тир} * N + З_p \quad (6.8)$$

$$P = З_{тир} * N + З_p * n \text{ мес} \quad (6.9)$$

где n - количество месяцев в рассматриваемом периоде

Таблица 6.8 – Техничко-экономические показатели работы

Периоды	Показатели		$1/(1 + \alpha)^t$, руб.	Дисконти рованные, руб.	Годовая экономическая эффективность	ЧДД с нарастающим итоном
	Доходы, руб.	Расходы, руб.				
3 квартал 2021	60000	128254,8	0,8772	52631,58	112504,21	-59 872,63
4 квартал 2021	65000	64434,8	0,7695	50015,39	49580,49	-59 437,73
1 квартал 2022	70000	64434,8	0,6750	47248,01	43491,65	-55 681,38
2 квартал 2022	75000	64434,8	0,5921	44406,02	38150,57	-49 425,93
3 квартал 2022	80000	64434,8	0,5194	41549,49	33465,42	-41 341,85
4 квартал 2022	85000	64434,8	0,4556	38724,86	29355,63	-31 972,62
1 квартал 2023	90000	64434,8	0,3996	35967,36	25750,55	-21 755,82

Продолжение таблицы 6.8

2 квартал 2023	95000	64434,8	0,3506	33303,11	22588,20	-11 040,91
3 квартал 2023	100000	64434,8	0,3075	30750,79	19814,21	-104,33
4 квартал 2023	105000	64434,8	0,2697	28323,10	17380,89	10 837,88
Итого	825000	708168	5,21	402919,71	392081,82	

Срок окупаемости - минимальный временной интервал (от начала осуществления проекта), за пределами которого интегральный эффект становится положительным и в дальнейшем остается неотрицательным. Это период, измеряемый в месяцах, кварталах или годах, начиная с которого первоначальные вложения и другие затраты, связанные с проектом, покрываются суммарными результатами его осуществления.

Определим срок окупаемости графически. Для этого необходимо построить график срока окупаемости проекта он представлен на рисунке 19.

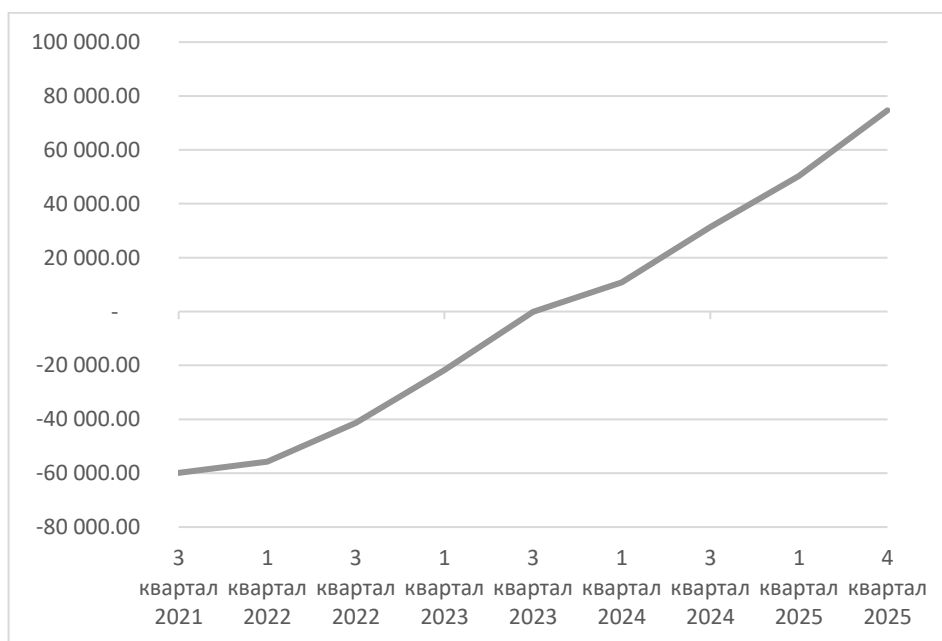


Рисунок 19 – График окупаемости.

Как видно из графика, данный продукт окупит себя в четвертом квартале 2023 года и начнет приносить прибыль. Из этого можно сделать вывод, что проект экономически целесообразен для разработки и внедрения в организацию.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была спроектирована и реализована информационная система для клиники.

Был изучен рынок продуктов в данном сегменте и определено, что на данный момент уже существует множество информационных систем подобного типа, но не все они подходят по каким-то параметрам тем, кто задумался об столь важном приобретении. Будь это стоимость, функционал или если система слишком крупна. Тем не менее, конкуренция очень высока. Но проблема состоит в том, что даже при наличии такого большого количества предложений, очень много организаций не пользуются ими, в том числе и муниципальные.

В ходе выполнения работ по проектировке и реализации был приобретен важный опыт по созданию такого рода проектов. Несомненно, в данной работе допущено не мало ошибок, на которых стоит поучиться, а масштаб разработки можно увеличивать до невероятных размеров.

Так же пришлось определяться с выбором СУБД. Эта задача не оказалась слишком сложной, но и не была слишком простой, так как уже существует множество решений. Выбор пал на Microsoft SQL Server, потому что это проверенный поставщик и данный продукт обеспечивает хорошую защиту данных и высокую производительность.

В ходе экономических исследований стало выявлено, что внедрение данного программного продукта является целесообразным и выгодным вложением.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. СТО ЮУрГУ 21–2008 Стандарт организации. Система управления качеством образовательных процессов. Курсовая и выпускная квалификационная работа. Требования к содержанию и оформлению / составители: Т.И. Парубочая, Н.В. Сырейщикова, А.Е. Шевелев, Е.В. Шевелева. – Челябинск: Изд-во ЮУрГУ, 2008. – 55 с.
2. В. И. Горбаченко / Проектирование информационных систем с СА ERwin Modeling Suite 7.3 : учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с.
3. Интернет ресурс Яндекс Дзен, автор – пользователь MedTelegraph, 5 инновационных Медицинских Информационных Систем для вашей клиники – <https://zen.yandex.ru/media/medtelegraph/5-innovacionnyh-medicinskih-informacionnyh-sistem-dlia-vashei-kliniki-5aedc7bcd7bf215715282b6c>
4. Лучшие медицинские информационные системы – <https://a2is.ru/catalog/meditsinskie-informatsionnye-sistemy>
5. Медицинская система Archimed+ – <https://archimed.pro>
6. Медицинская информационная система – [qMShttps://sparm.com/products/qms](https://sparm.com/products/qms)
7. Введение в MS SQL Server и T-SQL – <https://metanit.com/sql/sqlserver/1.1.php>
8. Краткий обзор языка C# – <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>
9. Microsoft Azure – <https://azure.microsoft.com/ru-ru/products/azure-sql/>
10. Microsoft SQL Server 2019 – <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019>
11. Сколько стоят услуги программистов? Цены студий и фрилансеров – <https://www.kadrof.ru/articles/46641>

- 12.МЕТАНИТ.СОМ – Сайт о программировании.
<https://metanit.com/sql/sqlserver/>
- 13.Бизнес-план медицинского центра –
<https://www.business.ru/article/1291-biznes-plan-meditsinskogo-tsentra>
- 14.Учебник по языку SQL (DDL, DML) на примере диалекта MS SQL Server – <https://habr.com/ru/post/255361/>

ПРИЛОЖЕНИЕ А

Код хранимых процедур базы данных.

Процедура добавления нового клиента.

```
CREATE PROCEDURE [dbo].[AddClient]
    @Фамилия nvarchar(max),
    @Имя      nvarchar(max),
    @Отчество nvarchar(max),
    @Паспорт_СН char(10),
    @Паспорт_КВ nvarchar(max),
    @Город      nvarchar(max),
    @Улица      nvarchar(max),
    @Дом        nvarchar(16),
    @Квартира  char(5),
    @Полис     char(16),
    @Телефон   char(12),
    @id        int output,
    @ДР       date
AS
BEGIN
    INSERT INTO Пациент (Фамилия, Имя, Отчество, Паспорт_серия_номер,
Паспорт_кем_выдан, Город, Улица, Дом, Квартира, Полис, Телефон, Дата_рождения)
VALUES (@Фамилия, @Имя, @Отчество, @Паспорт_СН, @Паспорт_КВ, @Город, @Улица, @Дом,
@Квартира, @Полис, @Телефон, @ДР);
    SET @id = @@IDENTITY
END
```

Процедура авторизации.

```
CREATE PROCEDURE [dbo].[log_in]
    @login nvarchar(max) = '',
    @passw nvarchar(max) = '',
    @id int = '' output
AS
BEGIN
    set @id = (select id from поли
where log_in = @login and passw = @passw)
END
```

Процедура списания материалов

```
CREATE PROCEDURE [dbo].[SpisanieMat]
    @id int,
    @kol float
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE Материалы
    SET Количество = @kol
    WHERE id = @id
END
```

ПРИЛОЖЕНИЕ Б

Код приложения

Обработчики событий.

Обработчик кнопки «ОК» формы авторизации.

```
private void btn_ok_Click(object sender, EventArgs e)
{
    lb_message.Text = "";
    string login, passw;
    int id = -1;
    //проверка ввода
    if (string.IsNullOrEmpty(tb_login.Text) || string.IsNullOrEmpty(tb_pa
ssw.Text))
    {
        lb_message.Text = "Не введен логин/пароль";
        lb_message.ForeColor = System.Drawing.Color.Red;
    }
    else
    {
        login = tb_login.Text;
        passw = tb_passw.Text;
        try
        {
            {
                id = Authorized.Auth(login, passw);
                if (id == 1)
                {
                    Form reg = new FAdministrator();
                    reg.Show();
                    Hide();
                }
                else lb_message.Text = "Не верная пара логин/пароль"; lb_message.
ForeColor = System.Drawing.Color.Red;
            }
            catch (Exception a)
            {
                MessageBox.Show("Ошибка: " + a.Message, "Ошибка!", 0);
            }
        }
    }
}
```

Открытие документов.

```
private void примерToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process myProcess = new Process();
    myProcess.StartInfo.FileName = "C:/Users/Дмитрий/YandexDisk/Учеба ЮУрГУ/Д
иплом/Проект/Clinic/Docs/obrazec-soglasie-na-obrabotku-personalnyh-
dannyh.doc"; //"c:\\MyFile.doc";
    myProcess.StartInfo.Verb = "Open";
}
```



```

        myProcess.StartInfo.CreateNoWindow = false;
        myProcess.Start();
    }

```

Другие документы открываются аналогично.

Пример открытия форм.

```

FClients newMDIChild = new FClients();
    newMDIChild.MdiParent = this;
    newMDIChild.Show();

```

Смена пользователя

```

{
    Form1 f1 = new Form1();
    f1.Show();
    this.Hide();
    SqlConnection conn = DBUtils.GetDBConnection();
    conn.Close();
}

```

Выход из приложения.

```

private void Administrator_FormClosed(object sender, FormClosedEventArgs e)
{
    //закрываем подключение и программу
    SqlConnection conn = DBUtils.GetDBConnection();
    conn.Close();
    Application.Exit();
}

```

Добавление клиента.

```

private void BtnOk_Click(object sender, EventArgs e)
{
    //проверка нулевых значений обязательных полей
    string name = TbName.Text;
    string surname = TbSurname.Text;
    string phone = TbPhone.Text;
    string otch = TbOtchestvo.Text;
    string paspsn = TbPasspSN.Text;
    string paspkv = TbPassKV.Text;
    string city = TbCity.Text;
    string street = TbStreet.Text;
    string house = TbHouse.Text;
    string apartament = TbApartamentNumb.Text;
    string polis = TbPolis.Text;
    string DR = TbDR.Text;
    if (string.IsNullOrEmpty(phone) || string.IsNullOrEmpty(surname) || string.IsNullOrEmpty(name))
    {
        MessageBox.Show("Обязательно заполните поля обозначенные звездочкой!", "Ошибка выполнения");
    }
}

```

```

else
{
    try
    {
        SqlConnection conn = DBUtils.GetDBConnection();
        conn.Open();
        SqlCommand cmd = new SqlCommand("AddClient", conn);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Фамилия", surname);
        cmd.Parameters.AddWithValue("@Имя", name);
        cmd.Parameters.AddWithValue("@Отчество", otch);
        cmd.Parameters.AddWithValue("@Паспорт_СН", paspsn);
        cmd.Parameters.AddWithValue("@Паспорт_КВ", paspkv);
        cmd.Parameters.AddWithValue("@Город", city);
        cmd.Parameters.AddWithValue("@Улица", street);
        cmd.Parameters.AddWithValue("@Дом", house);
        cmd.Parameters.AddWithValue("@Квартира", apartment);
        cmd.Parameters.AddWithValue("@Полис", polis);
        cmd.Parameters.AddWithValue("@Телефон", phone);
        cmd.Parameters.AddWithValue("@ДР", DR);
        // Добавить параметр
        cmd.Parameters.Add(new SqlParameter("@id", SqlDbType.Int));
        // Зарегистрировать параметр
        cmd.Parameters["@id"].Direction = ParameterDirection.Output;
        cmd.ExecuteNonQuery();
        int idClient = (int)cmd.Parameters["@id"].Value;
        conn.Close();
        cmd.Dispose();
        this.Close();
        DialogResult = DialogResult.OK;
    }

    catch (Exception a)
    {
        MessageBox.Show("Ошибка: " + a.Message, "Ошибка!");
    }
}
}

```

Обработчик нажатия кнопки «НОВЫЙ КЛИЕНТ»

```

private void toolStripButton1_Click(object sender, EventArgs e)
{
    AddClient AC = new AddClient();
    AC.ShowDialog();
    if (AC.DialogResult == DialogResult.OK)
    {
        пациентTableAdapter.Update(clinicDataSet1);
        clinicDataSet1.Clear();
        пациентTableAdapter.Fill(clinicDataSet1.Пациент);
    }
}

```

При нажатии на кнопку «Списание» формы «Материалы».

```
private void toolStripButton1_Click(object sender, EventArgs e)
{
    SpisanieMaterialov SM = new SpisanieMaterialov();
    SM.ShowDialog();
    if (SM.DialogResult == DialogResult.OK)
    {
        материалыTableAdapter.Update(clinicDataSet1.Материалы);
        clinicDataSet1.Clear();
        едИзмTableAdapter.Fill(clinicDataSet1.едИзм);
        материалыTableAdapter.Fill(clinicDataSet1.Материалы);
    }
}
```

При изменении выбора строки в форме «Материалы».

```
private void материалыBindingSource_CurrentChanged(object sender, EventArgs e)
{
    DataRowView drv = материалыBindingSource.Current as DataRowView;
    if (drv != null)
    {
        Data.idMat = Convert.ToInt32(drv.Row["id"]);
        Data.KolvoMat = Convert.ToSingle (drv.Row["Количество"]);
    }
}
```

Форма «Списание», нажатие на кнопку «ОК».

```
private void BtnOk_Click(object sender, EventArgs e)
{
    string Kolvo = textBox1.Text;
    if (string.IsNullOrEmpty (Kolvo))
    {
        MessageBox.Show("Заполните количество списываемого материала", "Внимание!");
    }
    else
    {
        float k = Data.KolvoMat - Convert.ToSingle(Kolvo);
        if (k > 0)
        {
            try
            {
                SqlConnection conn = DBUtils.GetDBConnection();
                conn.Open();
                SqlCommand cmd = new SqlCommand("SpisanieMat", conn);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@id", Data.idMat);
                cmd.Parameters.AddWithValue("@kol", k);
                cmd.ExecuteNonQuery();
                conn.Close();
            }
            catch { }
        }
    }
}
```

```

        cmd.Dispose();
        DialogResult = DialogResult.OK;
    }
    catch (Exception a)
    {
        MessageBox.Show("Ошибка: " + a.Message, "Ошибка!");
    }
}
else
{
    MessageBox.Show("Нельзя списать больше, чем имеется", "Ошибка спи
сания");
    DialogResult = DialogResult.Cancel;
}
}
}

```

Проверка и исключение символов поля ввода.

```

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (Char.IsNumber(e.KeyChar) | (e.KeyChar == Convert.ToChar(",")) | e.Key
Char == '\b') return;
    else
        e.Handled = true;
}

```

Отмена формы «Списание».

```

private void BtnCancel_Click(object sender, EventArgs e)
{
    SqlConnection conn = DBUtils.GetDBConnection();
    conn.Close();
    DialogResult = DialogResult.Cancel;
    this.Close();
}

```

Вывод на печать прайса из формы «Прайс лист».

```

private void button1_Click(object sender, EventArgs e)
{
    WordApp = new Word.Application();
    WordApp.Visible = true;
    WordDocuments = WordApp.Documents;
    WordDocument = WordDocuments.Add();
    WordParagraphs = WordDocument.Content.Paragraphs;
    WordParagraph = WordParagraphs[1];
    WordParagraph.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCente
r;

    WordRange = WordParagraph.Range;
    WordRange.InsertAfter("Прайс лист услуг клиники\n");
    WordRange.Font.Bold = 1;
    WordRange.Font.Size = 16;
    WordRange.Collapse(Word.WdCollapseDirection.wdCollapseEnd);
}

```

```

WordRange = WordParagraph.Range;
// Добавим текст, он будет выделенным участком.
WordRange.InsertAfter("по состоянию на " +
    DateTime.Now.ToLongDateString() + "\n");
// Сделаем шрифт выделенного участка нежирным
WordRange.Font.Bold = 0;
// Сделаем размер шрифта выделенного участка равным 14
WordRange.Font.Size = 14;
int i = 0;
// Цикл по записям таблицы
for (процедурыBindingSource.MoveFirst();
    i < процедурыBindingSource.Count;
    процедурыBindingSource.MoveNext())
{
    DataRowView drv1 = процедурыBindingSource.Current as DataRowView;
    // Получаем значения полей
    int id = Convert.ToInt32(drv1.Row["id"]);
    string Название = drv1.Row["Название"].ToString();
    string Стоимость = drv1.Row["Стоимость"].ToString();
    // Добавим параграф
    WordParagraph = WordParagraphs.Add();
    // Устанавливаем выравнивание по левой границе
    WordParagraph.Alignment =
        Word.WdParagraphAlignment.wdAlignParagraphLeft;
    WordRange = WordParagraph.Range;
    WordRange.Font.Bold = 0;
    WordRange.Font.Size = 12;
    // Добавим текст в новый параграф
    WordRange.InsertAfter("\t" + Convert.ToString(Название) + " - " +
        Convert.ToString(Math.Truncate(System.Convert.ToDecimal(Стоим
ость))) + " руб.");
    i++;
}

```

Выборка материалов на процедуру.

```

private void процедурыBindingSource_CurrentChanged(object sender, EventArgs e)
{
    DataRowView drv = процедурыBindingSource.Current as DataRowView;

    if (drv != null)
    {
        idProc = Convert.ToInt32(drv.Row["id"]);
        try
        {
            this.материалыНаПроцедуруОтПрTableAdapter.Fill(this.clinicDataSet
1.МатериалыНаПроцедуруОтПр, idProc);
        }
        catch (System.Exception ex)
        {
            System.Windows.Forms.MessageBox.Show(ex.Message);
        }
    }
}

```

Классы для подключения к базе данных

```
public static SqlConnection GetDBConnection()
{
    string datasource = @"DESKTOP-A9Q0I3R";
    string database = "Clinic";
    string user = "DESKTOP-A9Q0I3R/Дмитрий";
    return DBSQLServerUtils.GetDBConnection(datasource, database, user);
}

public static SqlConnection
    GetDBConnection(string datasource, string database, string user)
{
    // Data Source=DESKTOP-
A9Q0I3R;Initial Catalog=Clinic;Integrated Security=True
    string connString = @"Data Source=" + datasource + ";Initial Catalog=" +
database + ";Integrated Security=True;User ID=" + user;
    SqlConnection conn = new SqlConnection(connString);
    return conn;
}
```