

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка автоматизированной системы управления платными услугами (на примере ГБУЗ СО «Жигулевская ЦГБ»)»

Студент

А.Д. Свистов

(И.О. Фамилия)

(личная подпись)

Руководитель

д.т.н, доцент, С.В. Мкртычев

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

АННОТАЦИЯ

Выпускная квалификационная работа на тему: «Разработка автоматизированной системы управления платными услугами (на примере ГБУЗ СО «Жигулевская ЦГБ»)».

Объект исследования бакалаврской работы – бизнес-процесс управления платными услугами ГБУЗ СО «Жигулевская ЦГБ».

Предмет исследования бакалаврской работы – информационная система управления платными услугами ГБУЗ СО «Жигулевская ЦГБ».

Сущность выполненной работы заключается в разработке автоматизированной информационной системы управления платными услугами на примере Жигулевской ЦГБ.

Полученная разработка не несет новизны – существуют подобные аналоги медицинского программного обеспечения, однако разработанный продукт является уникальным, так как реализует исключительно нужный функционал.

Внедрение программного продукта является эффективным. В дальнейшем он может применяться в других медицинских учреждениях.

В первой главе проанализирована предметная область. Во второй главе разработаны и реализованы проектные решения. В третьей главе произведена оценка и обоснование экономической эффективности проекта.

ОГЛАВЛЕНИЕ

Введение.....	5
Глава 1 Анализ предметной области.....	7
1.1 Технико-экономическая характеристика предметной области.....	7
1.1.1 Характеристика предприятия ГБУЗ СО «Жигулевская ЦГБ»	7
1.1.2 Краткая характеристика подразделения и его видов деятельности... 8	
1.1.3 Сущность задачи автоматизации.....	10
1.2 Концептуальное моделирование предметной области	12
1.3 Постановка задачи.....	16
1.3.1 Цель и назначение автоматизированного варианта решения задачи	16
1.3.2 Общая характеристика организации решения задачи на ЭВМ	18
1.3.3 Формализация расчетов подзадач	18
1.4 Анализ существующих разработок	19
1.4.1 Определение критериев анализа существующих разработок	19
1.4.2 Сравнительная характеристика существующих разработок	19
Выводы по главе 1.....	23
Глава 2 Разработка и реализация проектных решений	24
2.1 Логическое моделирование предметной области.....	24
2.1.1 Логическая модель и ее описание	24
2.1.2 Используемые классификаторы и системы кодирования.....	28
2.1.3 Характеристика нормативно-справочной и входной оперативной информации	29
2.1.4 Характеристика базы данных.....	29
2.1.5 Характеристика результатной информации.....	32
2.2 Физическое моделирование АИС.....	33
2.2.1 Выбор архитектуры АИС	33
2.2.2 Функциональная схема проекта.....	34
2.2.3 Структурная схема проекта.....	36

2.2.4	Описание программных модулей	39
2.2.5	Схема взаимосвязи программных модулей и информационных файлов.....	39
2.3	Технологическое обеспечение задачи.....	40
2.3.1	Организация технологии сбора, передачи, обработки и выдачи информации	40
2.3.2	Схема технологического процесса сбора, передачи, обработки и выдачи информации.....	40
2.4	Контрольный пример реализации проекта и его описание	41
	Выводы по главе 2.....	49
Глава 3	Оценка и обоснование экономической эффективности проекта	50
3.1	Выбор и обоснование методики расчета экономической эффективности	50
3.2	Расчет показателей экономической эффективности проекта.....	51
	Выводы по главе 3.....	59
	Заключение	60
	Список используемой литературы	62
	Приложение А – Фрагмент исходного кода.....	66

ВВЕДЕНИЕ

Развитие вычислительной техники, а также глобальная информатизация являются предпосылками проблемы автоматизации производственной деятельности за счет внедрения современных инновационных способов взаимодействия с информационными системами.

Одной из сфер, требующих внедрения автоматизации, является сфера управления платными услугами в ГБУЗ СО «Жигулевской ЦГБ». Данный факт обусловлен тем, что на сегодняшний день в ГБУЗ СО «Жигулевской ЦГБ» имеется два способа учета платных услуг:

- при помощи ЕМИАС;
- при личном обращении.

Автоматизация учета платных услуг при личном обращении граждан позволит решить ряд проблем:

- повысить показатели производительности труда;
- сократить временные издержки;
- уменьшить объем однообразной и монотонной работы [4].

Актуальность бакалаврской работы обусловлена необходимостью повышения эффективности управления платными услугами ГБУЗ СО «Жигулевская ЦГБ».

Объект исследования бакалаврской работы – бизнес-процесс управления платными услугами ГБУЗ СО «Жигулевская ЦГБ».

Предмет исследования бакалаврской работы – информационная система управления платными услугами ГБУЗ СО «Жигулевская ЦГБ».

Цель бакалаврской работы - разработка информационной системы управления, обеспечивающей повышение эффективности управления платными услугами ГБУЗ СО «Жигулевская ЦГБ».

Чтобы достичь поставленную цель необходимо решить ряд задач:

– проанализировать предметную область автоматизации и выполнить постановку задачи на разработку информационной системы управления (ИСУ) платными услугами ГБУЗ СО «Жигулевская ЦГБ»;

– провести сравнительный анализ существующих ИТ-решений в данной области;

– спроектировать ИСУ платными услугами ГБУЗ СО «Жигулевская ЦГБ»;

– выполнить программную реализацию ИСУ платными услугами ГБУЗ СО «Жигулевская ЦГБ» и обосновать ее экономическую эффективность.

Структурно работы состоит из введения, трех разделов основной части, заключения, списка использованной литературы и приложений. Первая глава работы является аналитической, вторая – практической, третья содержит описание экономической эффективности.

ГЛАВА 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия ГБУЗ СО «Жигулевская ЦГБ»

История государственного бюджетного учреждения здравоохранения Самарской области «Жигулевской центральной городской больницы» началась в 1952 году вместе с введением в строй больницы «Куйбышевгидростроя». В 1958 году, по завершению строительства Волжской ГЭС, ее переименовали в Жигулевскую городскую больницу, что обеспечило возможность развития поликлинических и специализированных стационарных служб. По прошествии долгого развития, в 1988 году все лечебно-профилактические учреждения города стали единым территориальным объединением, которое впоследствии было реорганизовано в Жигулевскую центральную городскую больницу.

ГБУЗ СО «Жигулевская ЦГБ» оказывает следующие виды медицинских услуг:

- первичная медико-санитарная помощь, куда также входят: первичная доврачебная, врачебная и специализированная;
- паллиативная медицинская помощь;
- специализированная медицинская помощь.

В настоящее время Жигулевская ЦГБ является единственным медицинским учреждением в городе, где объединяются отделения стационаров разного профиля на 282 койки, в том числе хоспис и отделение сестринского ухода, 6 поликлиник на 1731 посещения в смену, а также стоматологическая и детская поликлиники, противотуберкулезное, психо-наркологическое отделения.

1.1.2 Краткая характеристика подразделения и его видов деятельности

На рисунке 1 представлена организационная структура отделения платных услуг ГБУЗ СО «Жигулевской ЦГБ».



Рисунок 1 - Организационная структура отделения платных услуг ГБУЗ СО «Жигулевской ЦГБ»

Учреждение оказывает платные медицинские услуги на основании договоров, которые регламентируют сроки и условия предоставления услуг, права, обязанности и ответственность сторон, а также порядок расчетов.

Учреждение оказывает услуги также в рамках Добровольного медицинского страхования (ДМС).

Основанием для оказания платных медицинских услуг являются:

- добровольное желание пациента получить платную медицинскую услугу;

- постановление Правительства РФ от 4 октября 2012 г. №1006 (Об утверждении правил предоставления платных медицинских услуг населению медицинским учреждением).

Медицинские организации, которые принимают участие в программе имеют право осуществлять реализацию платных медицинских услуг:

- на прочих условиях, нежели предусмотрено программой, целевыми программами и (или) территориальными программами, по желанию заказчика, включая в том числе: установление индивидуального поста медицинского наблюдения в случае лечения в стационаре и использование лекарственных препаратов, которые не входят в список важнейших и жизненно необходимых лекарственных средств, в том случае, когда их назначение и использование не обусловлено жизненными показаниями или заменой по причине индивидуальной непереносимости лекарственных средств, которые входят в указанный список, а также использование медицинских изделий, лечебного питания, в том числе специализированных продуктов лечебного питания, не соответствующих стандартам медицинской помощи;
- лицам, не имеющим Российского гражданства, а также лицам без гражданства (кроме лиц, обладающих полисом обязательного медицинского страхования, и граждан Российской Федерации, не проживающих постоянно на территории страны и не являющихся застрахованными по обязательному медицинскому страхованию), в тех случаях, когда это не покрывается международными договорами Российской Федерации;
- в случае анонимного предоставления медицинских услуг, за исключением ситуаций, предусмотренных законодательством Российской Федерации;
- в случаях самостоятельного обращения за получением медицинских услуг, за исключением ситуаций и порядка, предусмотренных статьей 21 Федерального закона «Об основах охраны здоровья граждан в

Российской Федерации», а также случаев оказания скорой, в том числе скорой специализированной, медицинской помощи и медицинской помощи, оказываемой в экстренной или неотложной форме.

1.1.3 Сущность задачи автоматизации

Основой автоматизации ГБУЗ СО «Жигулевской ЦГБ» является единая медицинская информационно-аналитическая система (ЕМИАС), которая успешно реализует процесс учета платных услуг, обращения по которым поступают посредством сети Интернет.

Однако есть пациенты, которые не имеют доступом к системе ЕМИАС, поэтому они вынуждены обращаться в поликлинику лично. Для обработки таких заявок в ГБУЗ СО «Жигулевской ЦГБ» используется пакет приложений Microsoft Office. Справочники и документы хранятся в форматах Excel и Word.

На рисунке 2 представлена диаграмма существующего процесса.

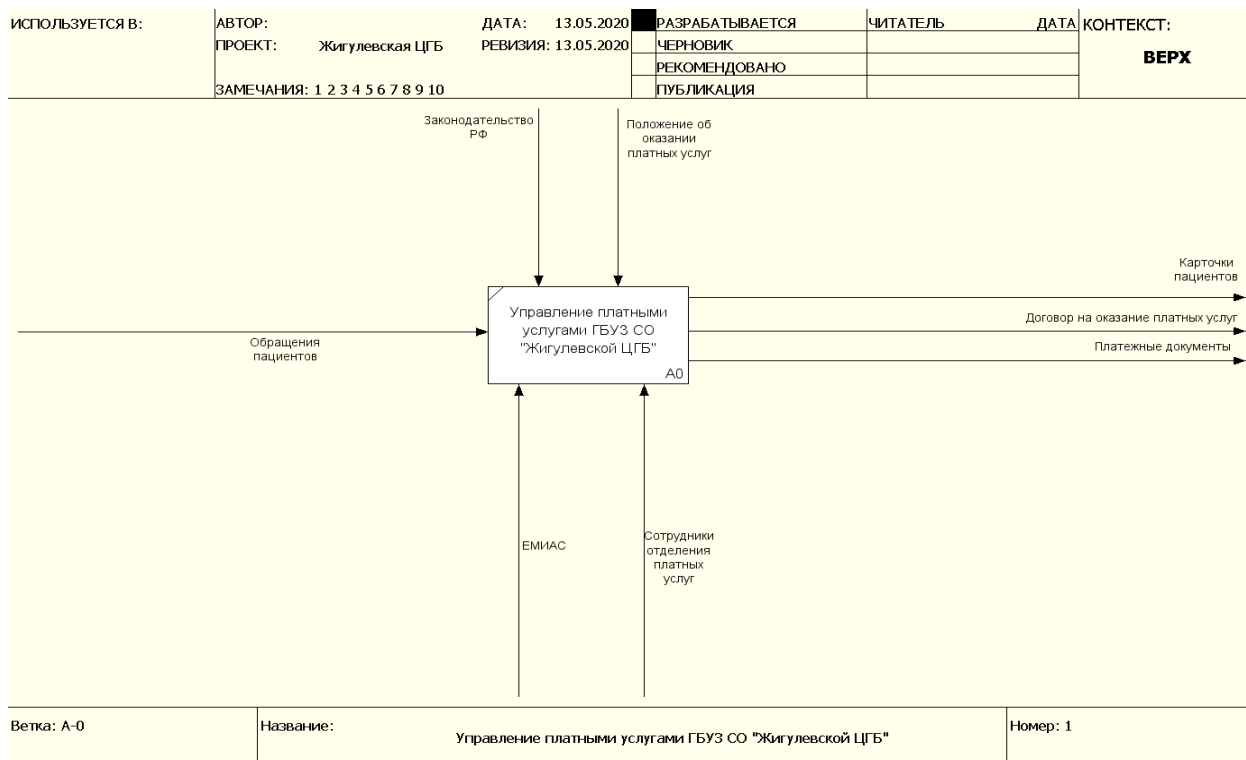


Рисунок 2 - Контекстная диаграмма процесса управления платными услугами в методологии IDEF0 («как есть»)

На рисунке 3 представлена декомпозиция существующего процесса управления платными услугами поликлиники [31].

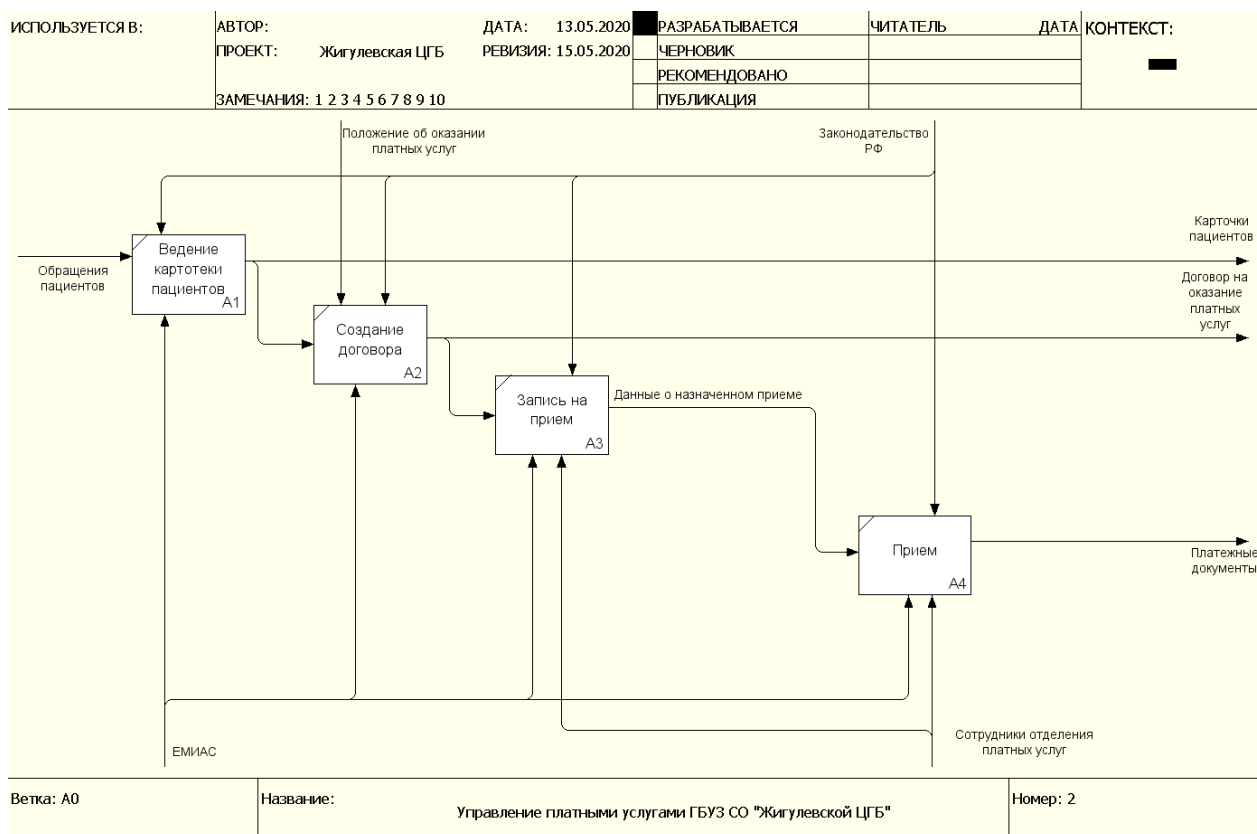


Рисунок 3 - Диаграмма декомпозиции существующего процесса управления платными услугами поликлиники

Главным недостатком существующего процесса является отсутствие возможности автоматизированного создания документов (договора, направления к специалисту, платежей) и их регистрации.

Разработка ИСУ для автоматизации процесса управления платными услугами ГБУЗ СО «Жигулевской ЦГБ» позволит оптимизировать деятельность по ведению единой картотеки историй болезни пациентов, обратившихся за платными услугами, вести информацию о состоявшихся приемах пациентов врачами-специалистами, производить формирование аналитической отчетности, сократить временные затраты на выполнение

основных технологических операций, сократить вероятность возникновения ошибок вследствие влияния человеческого фактора, уменьшить время поиска информации по истории болезни пациентов, что в конечном итоге позволит более рационально организовать деятельность медицинского учреждения в целом.

1.2 Концептуальное моделирование предметной области

Контекстная диаграмма процесса управления платными услугами поликлиники ГБУЗ СО «Жигулевская ЦГБ» в методологии IDEF0 приведена на рисунке 4 [27].

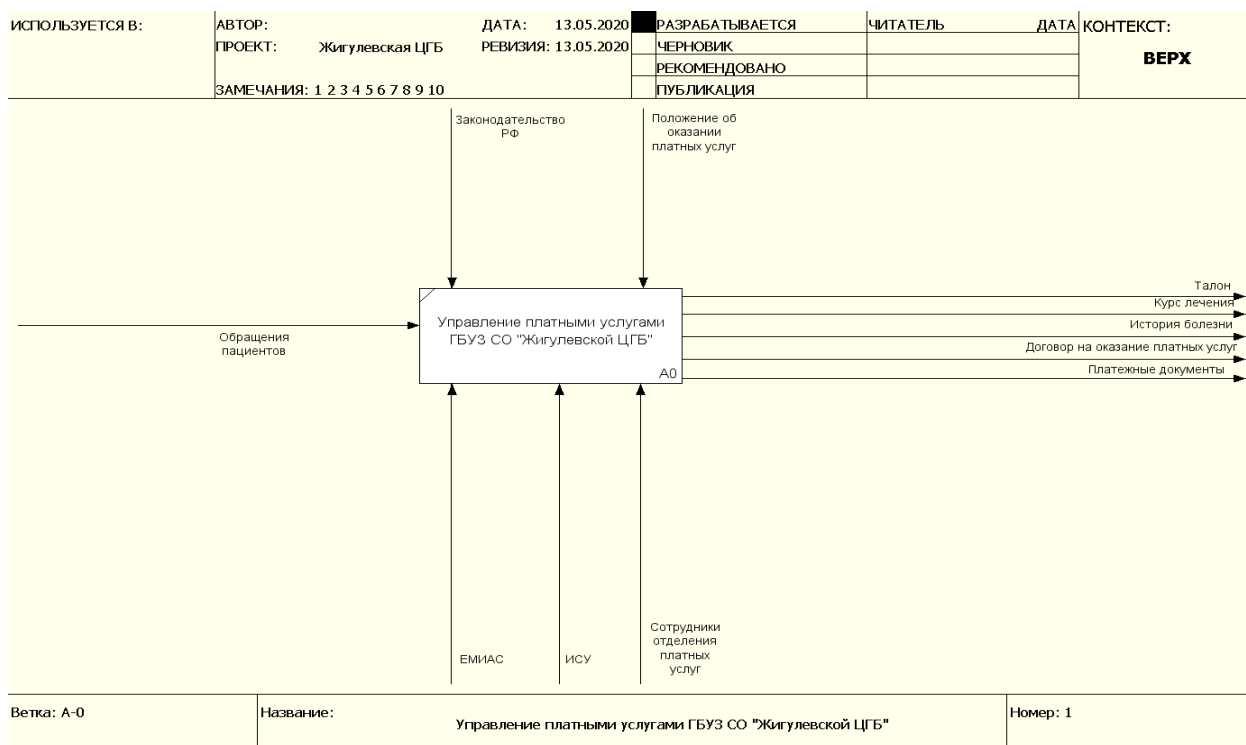


Рисунок 4 - Контекстная диаграмма процесса управления платными услугами в методологии IDEF0 («как должно быть»)

Из рисунка видно, что в рамках процесса предоставления платных услуг в состав входных информационных потоков входят обращения пациентов, на основании которых реализуется заполнение данных согласно

представленных документов, а также данных о причинах обращения, после чего выбирается время приема к профильному специалисту.

При этом обращения доступны как по сети (посредством ЕМИАС), так и при личном обращении [14].

В состав результатных информационных потоков входят:

- талон на предстоящий прием;
- курс лечения, назначаемый врачом-специалистом;
- история болезни пациента.

На рисунке 5 представлена декомпозиция процесса управления платными услугами поликлиники [31].

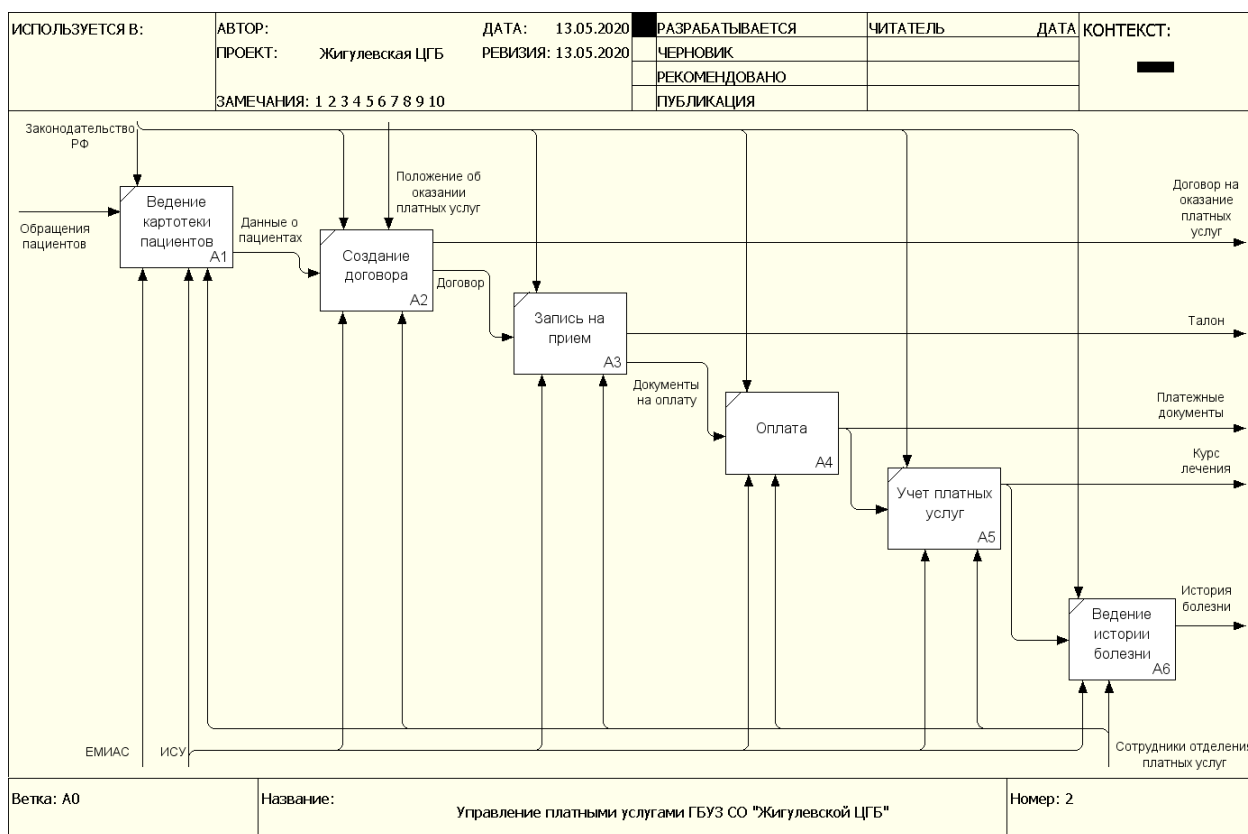


Рисунок 5 - Диаграмма декомпозиции процесса управления платными услугами поликлиники

Из данного рисунка видно, что к основным технологическим этапам технологии ведения учета платных услуг относятся:

- ведение картотеки пациентов;

- выбор профильных специалистов и времени приема;
- учет назначенных курсов лечения;
- учет изменений в истории болезни.

Указанные выше процессы тесно связаны с задачами автоматизации проекта. На рисунках 6-8 представлены диаграммы декомпозиции первого уровня [9, 25].

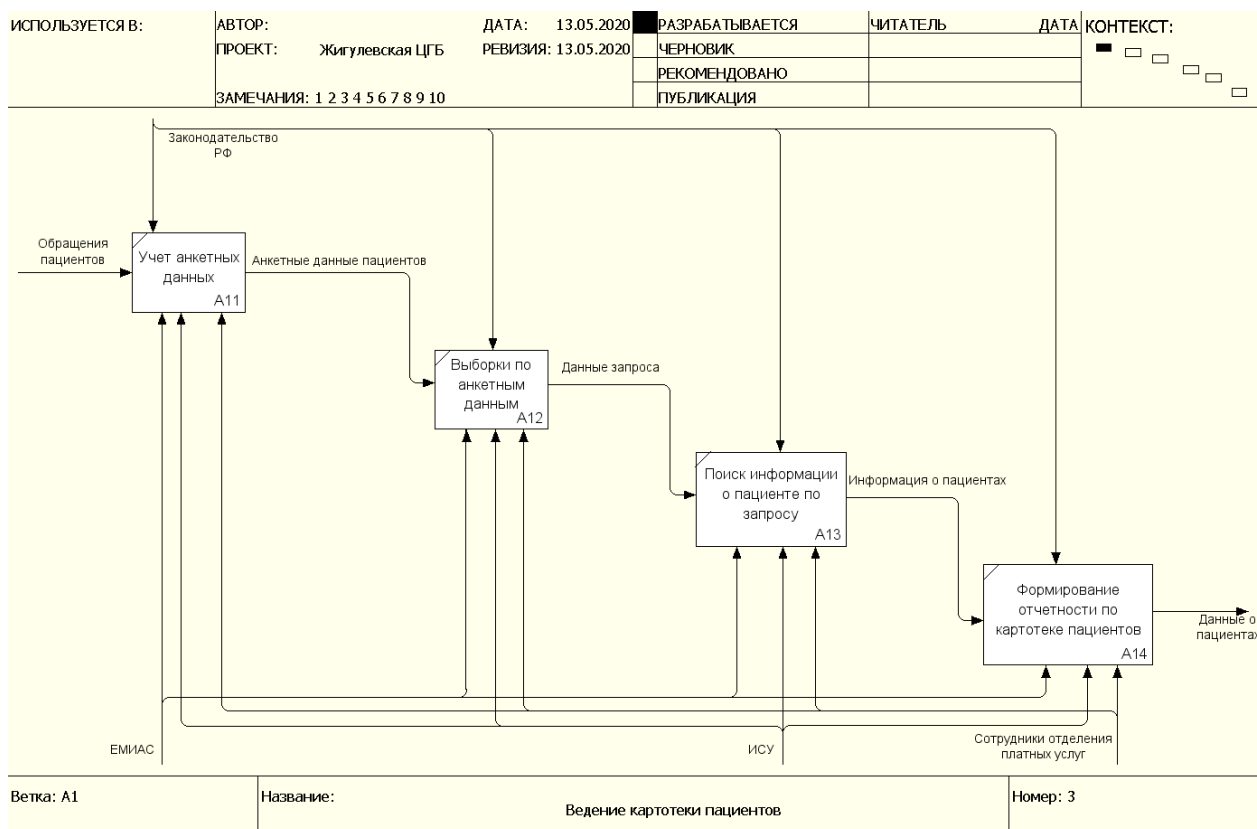


Рисунок 6 - Диаграмма декомпозиции ведения картотеки пациентов

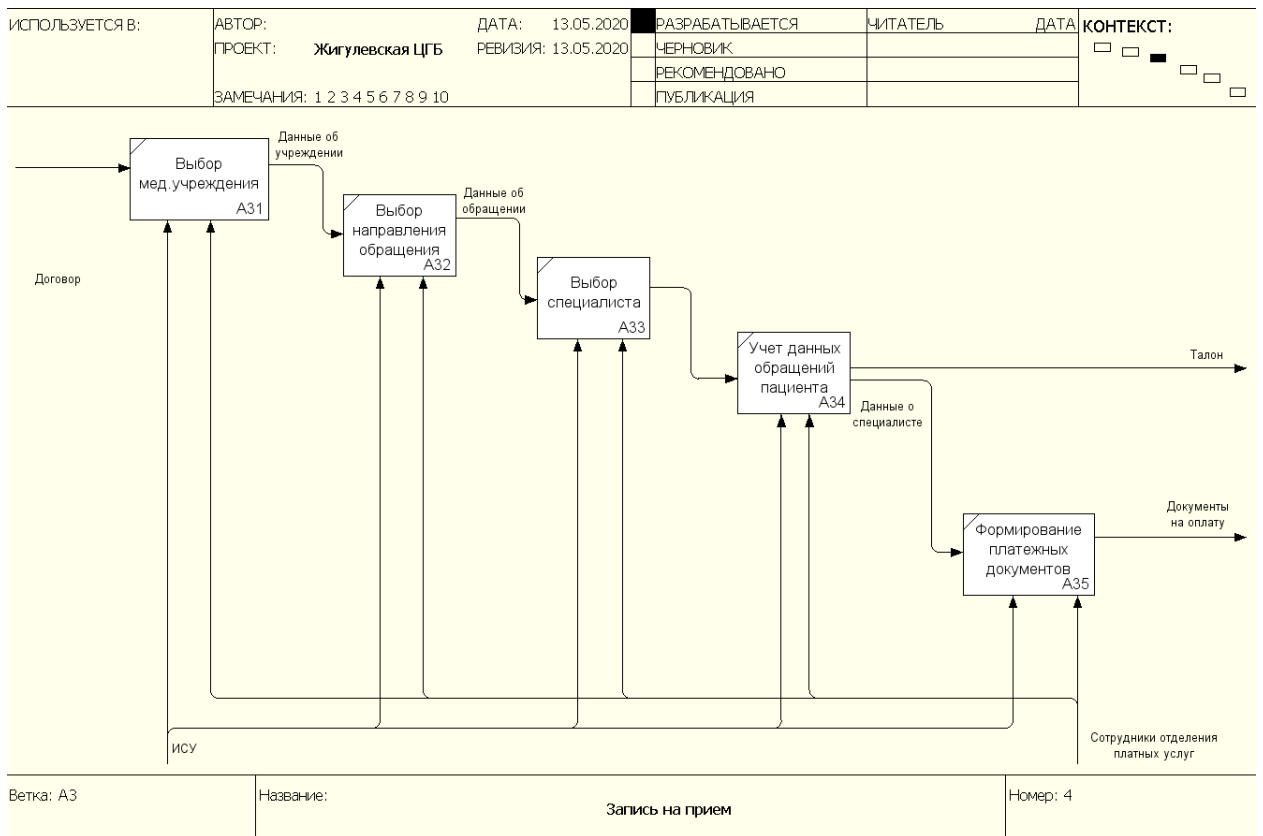


Рисунок 7 - Диаграмма декомпозиции процесса записи на прием

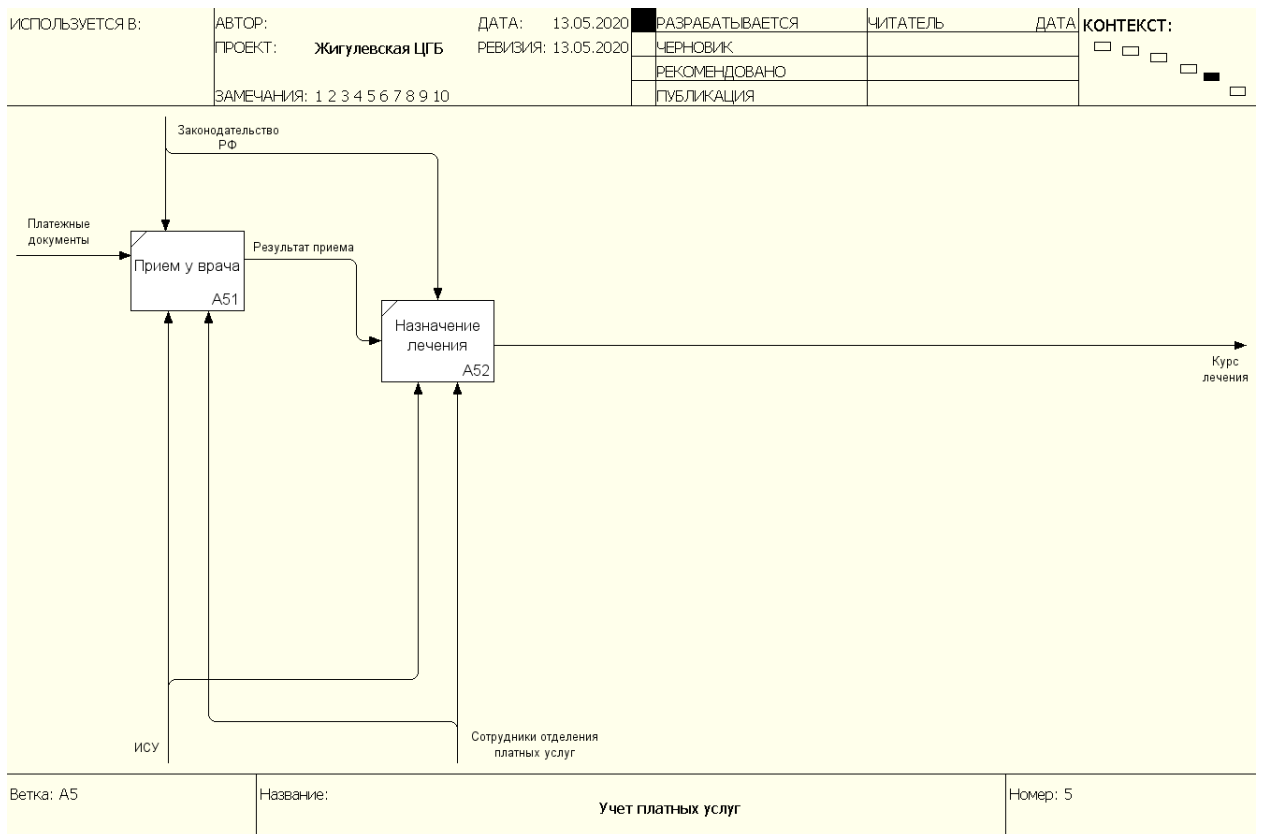


Рисунок 8 – Диаграмма декомпозиции процесса платных услуг

Из данных рисунков видно, что ведение картотеки пациентов в рамках технологии ведения электронной истории болезни предполагает учет некоторых анкетных данных граждан, а также выборки по этим данным с целью формирования списков граждан, подлежащих профилактическим осмотрам. Кроме того, здесь же предусмотрена возможность формирования отчетности по картотеке пациентов.

В рамках электронной истории болезни технологический этап «Запись на прием» предполагает учет всех обращений конкретного пациента ко врачу.

Таким образом, в качестве предметной области в рамках поставленной задачи учета пациентов ГБУЗ СО «Жигулевской ЦГБ» является множество фрагментов, взаимосвязанных друг с другом.

Например, пациенты и журнал сведений о пациентах, содержащий историю болезни и поставленные диагнозы, сведения об оказанных медицинских услугах, врачах-специалистах, времени приема, классификаторе диагнозов.

1.3 Постановка задачи

1.3.1 Цель и назначение автоматизированного варианта решения задачи

Разрабатываемая информационная система должна выполнять решение следующих задач:

- ведение картотеки пациентов, в том числе с данными о поставленных диагнозах, истории обращения в поликлинику и назначенных процедурах;
- ведение учета врачей-специалистов, в том числе времени их приема;
- ведение используемого классификатора медицинских диагнозов;

- учет приема пациентов и назначенных для них процедур;
- выдача талонов на прием к врачам-специалистам;
- ведение полной истории болезни.

Формирование картотеки пациентов предполагает учет атрибутов:

- идентификационные данные – фамилия, имя, отчество;
- дата рождения;
- серия и номер паспорта;
- адрес прописки;
- серия и номер медицинского полиса;
- данные о социальном статусе;
- контактные данные.

Справочник врачей-специалистов должен хранить и предоставлять следующую информацию:

- идентификационные данные – фамилия, имя, отчество;
- специализация;
- данные о наличии ученой степени и квалификационной категории.

Классификаторе медицинских диагнозов служит для хранения информации о диагнозе с его кратким описанием.

Журнал приема пациентов предполагает наличие формы открытия курса лечения, назначенных процедур, а также отметок врача о состоянии пациента.

Учет выдачи талонов на прием к врачам-специалистам предполагает работу с формой выбора пациента, врача-специалиста и автоматический поиск талонов на прием.

Разрабатываемая информационная система предполагается к использованию сотрудниками регистратуры, врачами-специалистами, а также администраторами базы данных.

Сотрудники регистратуры осуществляют выдачу пациентам талонов на прием к врачам-специалистам, учет пациентов, а также печать списка выданных талонов на прием.

Врачи-специалисты проводят открытие курса лечения, заполняют информацию о назначенных процедурах, а также истории болезни.

Администраторы системы имеют доступ к редактированию справочной информации, а также отвечают за разграничение прав доступа к информационной системе.

1.3.2 Общая характеристика организации решения задачи на ЭВМ

Разработка информационной системы позволит изменить существующий подход к обработке входной информации так, что все используемые данные будут храниться в едином формате в рамках централизованной базы данных.

Первый этап внедрения полученной системы потребует дополнительного времени, которое потребуется для переноса существующих данных поликлиники в базу. Однако последующий ввод информации будет автоматизирован посредством разработанной информационной системы.

1.3.3 Формализация расчетов подзадач

Разработка информационной системы может быть описана последовательностью этапов:

- изучение деятельности ГБУЗ СО «Жигулевская ЦГБ»
- постановка задачи;
- анализ существующих систем, способных решить задачу;
- анализ средств разработки;
- определение ресурсов будущей системы;
- определение входных и выходных данных;

- разработка информационной модели;
- разработка базы данных;
- разработка приложения;
- тестирование полученного приложения;
- обоснование экономической эффективности проекта.

1.4 Анализ существующих разработок

1.4.1 Определение критериев анализа существующих разработок

Разрабатываемая информационная система должна стать частью общей системы, используемой в деятельности ГБУЗ СО «Жигулевской ЦГБ» и должна легко интегрироваться с остальными частями существующей системы.

Рассматриваемая система служит для автоматизации деятельности учреждения здравоохранения на новом качественном уровне.

Основные функции, которые должны предоставляться информационной системой:

- формирование и ведение базы данных пациентов ГБУЗ СО «Жигулевской ЦГБ» и картотеки оказываемых медицинских услуг;
- проведение расчетов по оказанным медицинским услугам;
- ведение учета приемов пациентов и статистики по оказанным медицинским услугам;
- ведение истории болезни в электронном формате.

1.4.2 Сравнительная характеристика существующих разработок

Ранее было сказано, что в ГБУЗ СО «Жигулевской ЦГБ» для управления платными услугами используется система ЕМИАС, основным недостатком которой является необходимость наличия интернета. Пользователи, не обладающие достаточными навыками взаимодействия с

компьютером, не способны создать обращение в поликлинику, поэтому основная задача работы – автоматизировать процесс управления платными услугами тех пациентов, которые лично обращаются в поликлинику.

В данном разделе представлен краткий анализ аналогов разрабатываемой информационной системы.

На сегодняшний день на рынке информационных технологий представлено целое множество различных разработок в области автоматизации деятельности поликлиник. Наиболее популярными из них являются: ПО «ArhiMed+», «Учет пациентов», «Дентал-Софт».

Программный продукт «ArhiMed+» представляет собой программное обеспечение на базе клиент-серверной архитектуры, которое применяется с целью автоматизации работы учреждений здравоохранения.

На рисунке 9 представлен интерфейс данного программного обеспечения.

Основные функции, реализуемые данной программой:

- ведение картотеки пациентов;
- выписка больничных листов;
- автоматизация лаборатории;
- связь с медицинским оборудованием;
- внутренний чат специалистов;
- взаимодействие со страховыми компаниями;
- адресная SMS и e-mail-рассылка.

Интерфейс программного продукта «Учет пациентов», разработанного компанией Clinic host, представлен на рисунке 10.

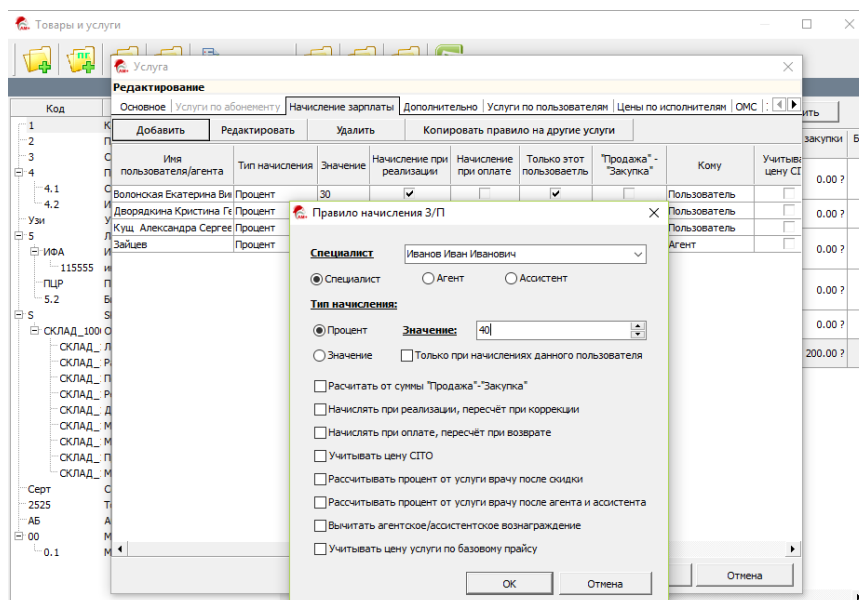


Рисунок 9 - Интерфейс ПО «Archimed+»

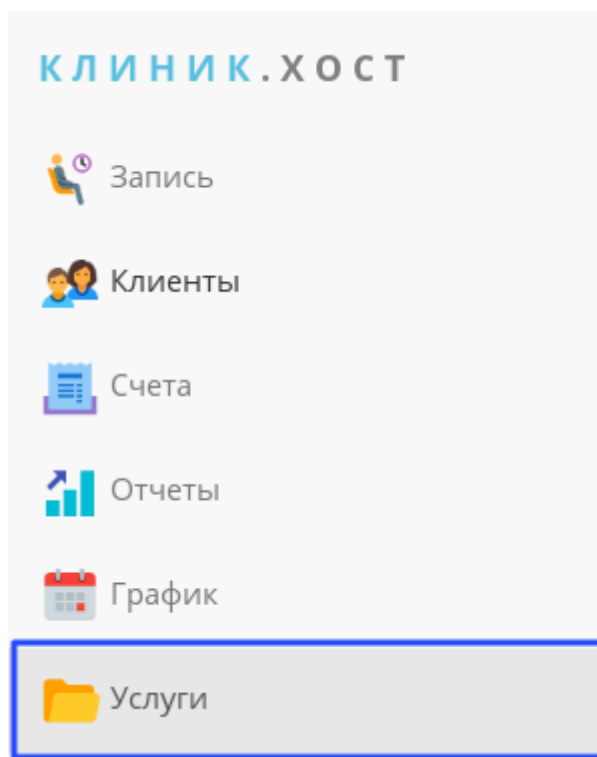


Рисунок 10 - Интерфейс ПО «Учет пациентов»

Данное приложение реализует следующие функции:

- централизованное ведение базы данных пациентов;
- организация рабочего времени специалистов;
- автоматизация приема;
- маркетинг и коммуникация с клиентами;

– анализ деятельности учреждения.

«Дентал-Софт» представляет собой информационную систему для частных практикующих врачей клиник, оказывающих как платные услуги, так и имеющих договоры со страховыми компаниями и принимающими пациентов по полисам ОМС. На рисунке 11 изображен интерфейс данного приложения.

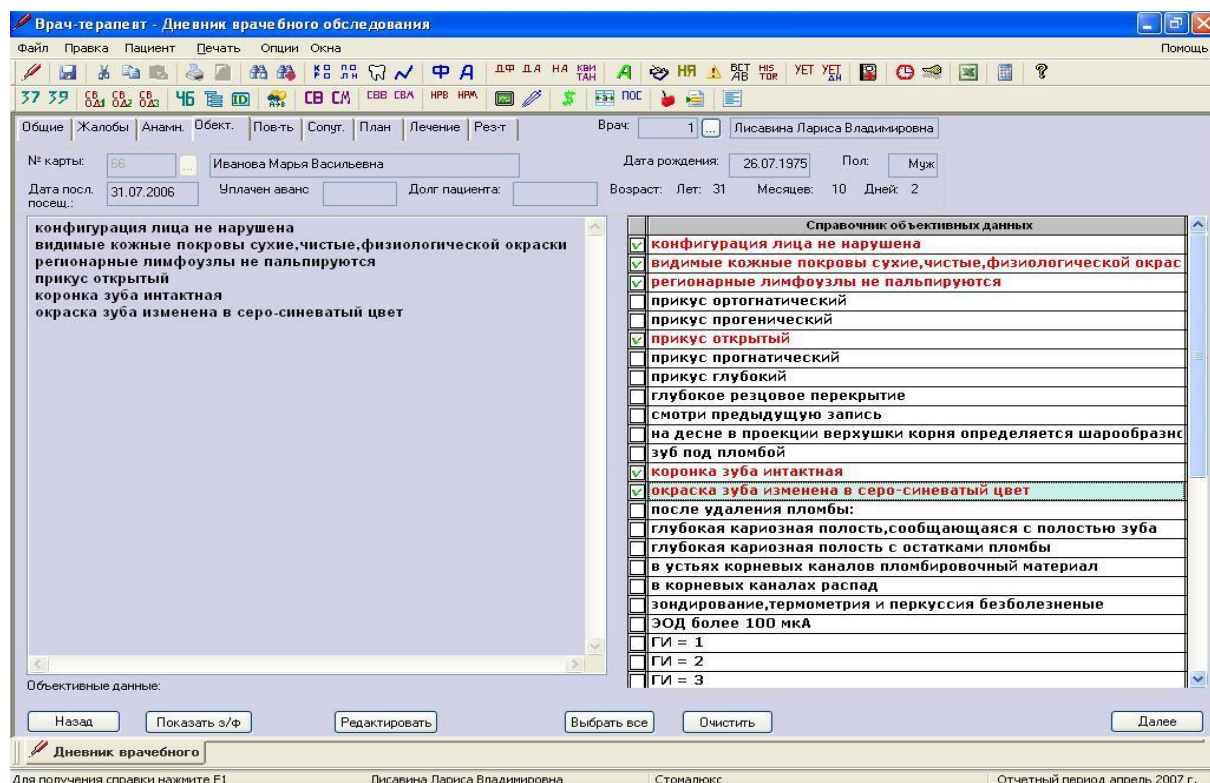


Рисунок 11 - Интерфейс ПО «Дентал-Софт»

Данная информационная система служит для ведения электронных медицинских карт пациентов, а также автоматизации медицинского документооборота. В основе данной программы лежит язык программирования MS Visual C++ 2010. Хранилищем данных может выступать одна из следующих СУБД: MS SQL Server, MS Access или MySQL Server.

В таблице 1 представлено сравнение описанных программных продуктов.

Таблица 1 – Сравнение аналогов

Программный продукт	Стоимость лицензии	Ведение картотеки	Автоматизация приема	Выгрузка отчетов
ArhiMed+	От 3000 рублей	+	+	+
Учет пациентов	От 3000 рублей	+	+	+
Дентал-Софт	От 4200 рублей	+	+	+

Стоит отметить, что все рассмотренные информационные системы обладают своими достоинствами, соответствующими технологиям работы ГБУЗ СО «Жигулевской ЦГБ», однако внедрение данных систем не является целесообразным по причинам:

- несоответствия данных систем существующей архитектуре автоматизированной системы;
- несоответствия специфике работы поликлиники, в результате чего необходима доработка каждого программного продукта.

Самостоятельная разработка позволит учесть специфику исследуемого медицинского учреждения.

Требования, предъявляемые к разрабатываемому программному решению:

- отсутствие необходимости установки дополнительного программного обеспечения и системных библиотек;
- удобный пользовательский интерфейс;
- высокая скорость обработки данных;
- формирование отчетной информации путем ее выгрузки в OpenOffice.org, используемым в качестве офисного пакета на рабочих станциях специалистов медицинского учреждения ГБУЗ СО «Жигулевская ЦГБ».

Выводы по главе 1

Данная глава работы – аналитическая. В ней приводится описание деятельности ГБУЗ СО «Жигулевской ЦГБ», ее организационная структура, а также основные функции.

Отдельное внимание в рамках данной главы уделено информационным системам, которые могут использоваться в медицине с целью автоматизации.

ГЛАВА 2 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОЕКТНЫХ РЕШЕНИЙ

2.1 Логическое моделирование предметной области

2.1.1 Логическая модель и ее описание

В процессе анализа предметной области были выделены следующие объекты [33]:

- справочник врачей-специалистов;
- справочник диагнозов;
- справочник видов процедур и программ лечения;
- картотека пациентов;
- журнал курсов лечения;
- журнал истории болезни, поставленных диагнозов;
- журнал приемов пациента.

Сущность «Врачи-Специалисты» характеризуется следующими атрибутами [19]:

- фамилия, имя, отчество;
- специальность;
- номер кабинета;
- квалификационная категория.

К атрибутам сущности «Диагнозы» относятся:

- наименование;
- описание;
- программа лечения.

Сущность «Виды процедур» описывается двумя атрибутами:

- наименование;
- стоимость.

Сущность «Программы лечения» также описывается двумя атрибутами:

- наименование;
- количество процедур.

Для сущности «Пациент» выделены следующие атрибуты [5]:

- фамилия, имя, отчество;
- дата рождения;
- номер медицинского полиса;
- адрес;
- контактные данные;
- социальный статус.

Сущность «Курсы лечения» описывается тремя атрибутами:

- фамилия, имя, отчество пациента;
- дата начала;
- дата завершения.

Для описания сущности «История болезни» используются следующие атрибуты:

- фамилия, имя, отчество пациента;
- фамилия, имя, отчество врача;
- дата;
- запись в истории болезни.

Сущность «Прием пациента» характеризуется следующими атрибутами:

- фамилия, имя, отчество пациента;
- фамилия, имя, отчество врача;
- дата и время приема.

В результате анализа предметной области были выделены следующие ограничения целостности:

- запись на прием может осуществляться исключительно в рабочее время поликлиники;
- ограничения на значения даты рождения пациентов;
- стоимость услуги должна задаваться положительным числом;
- на одно время к одному специалисту можно записать только одного пациента;
- нельзя записать на прием с прошедшей датой [6].

На рисунке 12 представлена модель данных.

На рисунке 13 представлена UML-диаграмма вариантов использования.

На рисунке 14 представлена UML-диаграмма последовательности.

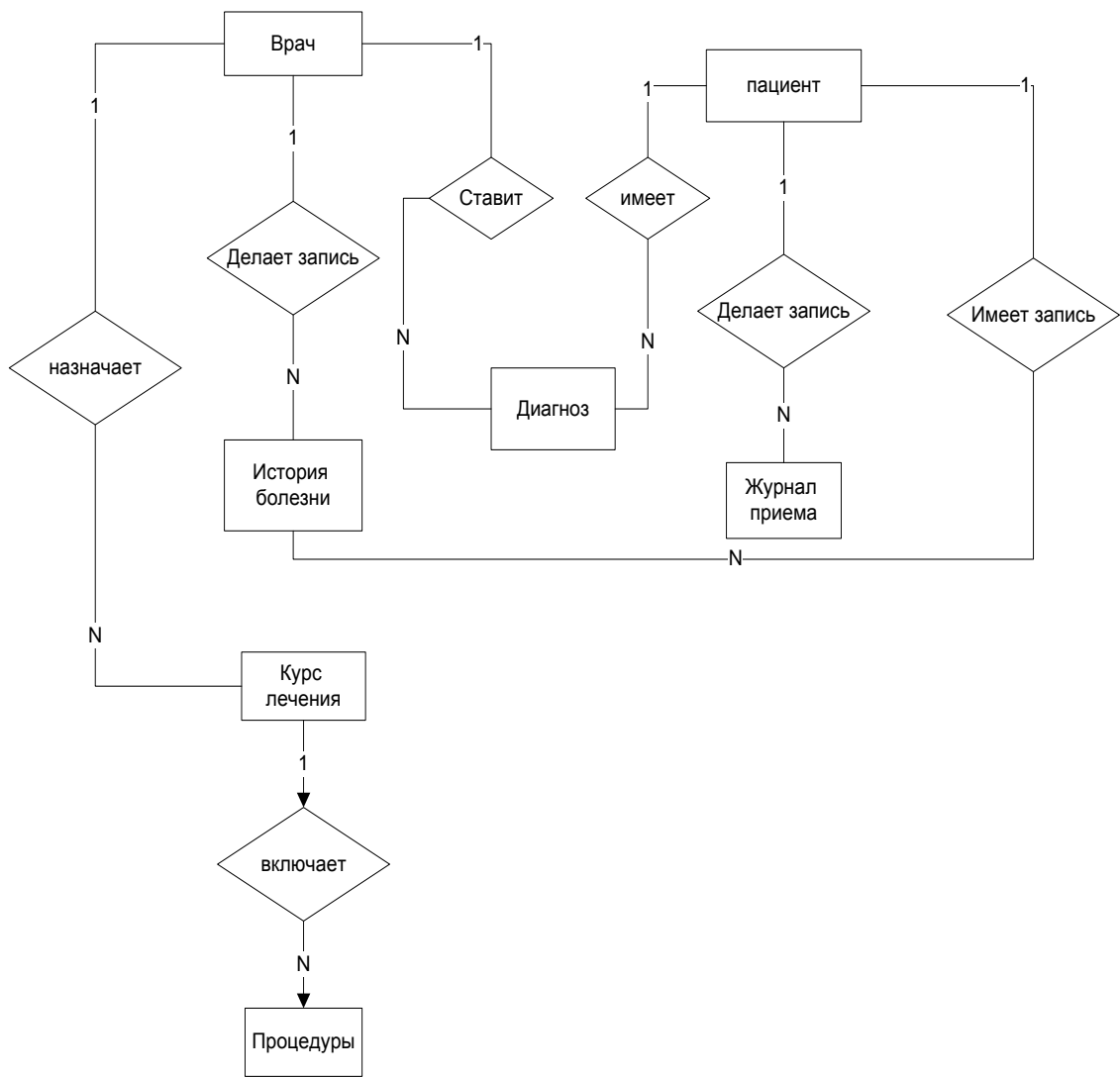


Рисунок 12 – Модель данных

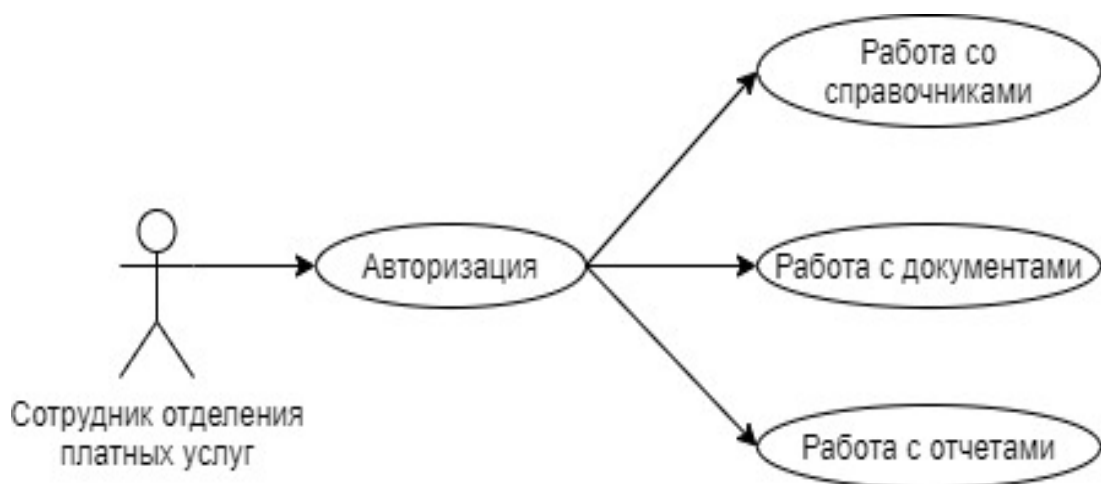


Рисунок 13 - UML-диаграмма вариантов использования

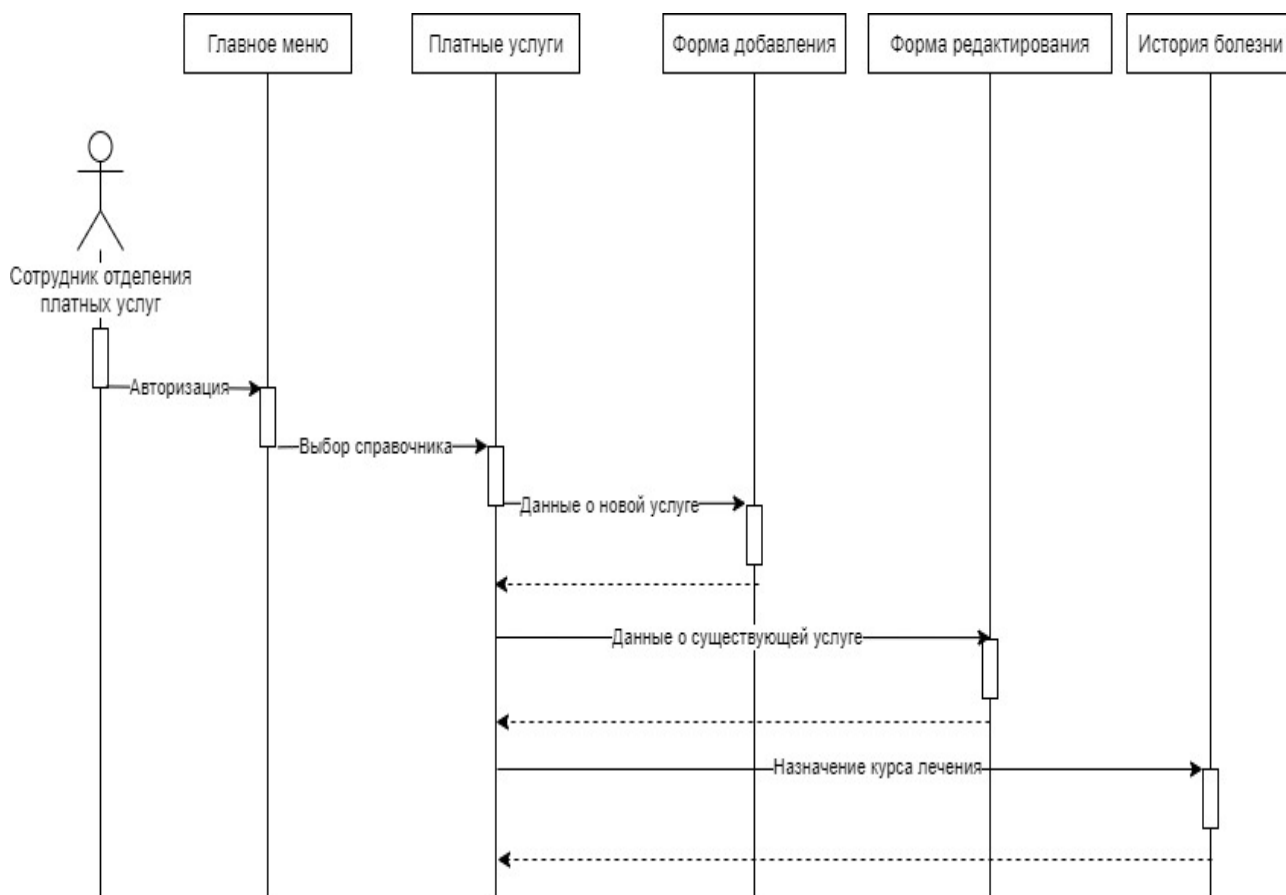


Рисунок 14 - UML-диаграмма вариантов последовательности

2.1.2 Используемые классификаторы и системы кодирования

В сфере разработке информационных систем кодированием «называется процесс присвоения объектам некоторых кодовых обозначений. Главной задачей данного процесса является создание однозначного обозначения объектов, а также обеспечение требуемого уровня достоверности кодируемой информации» [39].

Выбор конкретной системы кодирования определяется количеством классификационных признаков и разработанной системой классификации. В данном случае под системой классификации понимается совокупность правил распределения объектов множества на подмножества.

В рамках поставленной задачи в качестве кодов (уникальных идентификаторов) используются натуральные числа [30].

2.1.3 Характеристика нормативно-справочной и входной оперативной информации

В основе нормативно-справочной информации лежат данные Жигулевской ЦГБ. Сюда относятся данные о врачах, диагнозах и пациентах.

2.1.4 Характеристика базы данных

В данном разделе представлено описание структуры таблиц созданной базы данных.

В таблице 2 содержится структура таблицы с данными справочника пациентов.

Таблица 2 - Структура справочника пациентов

Наименование поля	Тип данных	Размер поля
code_pac	int	
fio	char	50
adr	char	50
tel	char	20
soc_pol	char	20
polis	char	20

Структура таблицы с данными справочника курсов лечения приведена в таблице 3.

Таблица 3 - Справочник курсов лечения

Наименование поля	Тип данных	Размер поля
code_pac	int	
day1	date	
day2	date	
code_vr	int	

В таблице 4 отражена структура таблицы с данными справочника программ лечения.

Таблица 4 - Справочник программ лечения

Наименование поля	Тип данных	Размер поля
code_prog	int	
nam	char	50
stm	money	

Структура таблицы с данными справочника процедур отражена в таблице 5.

Таблица 5 - Справочник процедур

Наименование поля	Тип данных	Размер поля
code_proc	int	
nam	char	50
stm	money	

В таблице 6 содержится структура таблицы с данными детализации программ лечения.

Таблица 6 - Детализация программ лечения

Наименование поля	Тип данных	Размер поля
code_proc	int	
code_prog	int	
kol	int	

В таблице 7 приведена структура таблицы с данными справочника врачей-специалистов.

Таблица 7 - Справочник «Врачи-специалисты»

Наименование поля	Тип данных	Размер поля
code_vr	int	
spec	char	50
cab	int	
kateg	char	50

Структура таблицы с данными журнала историй болезни представлена в таблице 8.

Таблица 8 - Журнал историй болезни

Наименование поля	Тип данных	Размер поля
code_pac	int	
code_vr	int	
day	date	
commnt	char	200

Таблица 9 отражает структуру таблицы с данными журнала отпущенных процедур [23, 24].

Таблица 9 - Журнал отпущенных процедур

Наименование поля	Тип данных	Размер поля
code_pac	int	
code_proc	int	
day	date	
stm	money	

Схема полученной базы данных приведена на рисунке 15 [39].

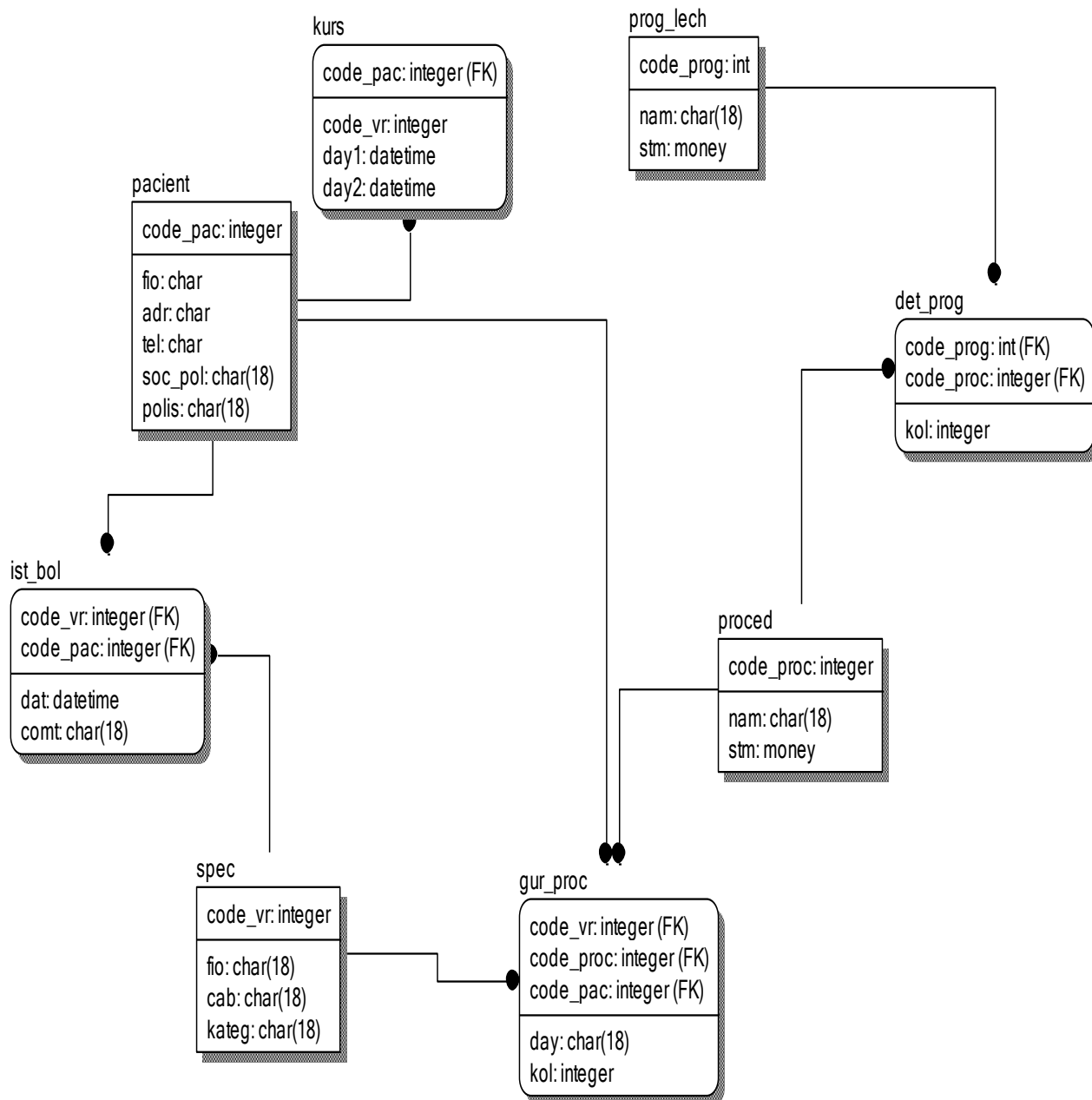


Рисунок 15 - Схема базы данных

2.1.5 Характеристика результатной информации

В качестве результатной информации выступают отчеты, получаемые пользователями при работе с системой. Сюда относятся:

- данные справочников;
- талоны записи.

2.2 Физическое моделирование АИС

2.2.1 Выбор архитектуры АИС

Вопрос выбора архитектуры АИС очень важен. При этом необходимо рассмотреть следующие возможные варианты в зависимости от основы системы:

- файл-серверная архитектуры – данная архитектура не предполагает сетевое разделение компонент, поэтому для реализации функций обработки данных, а также диалогов, используется компьютер пользователя. В этом случае данные извлекаются из файлов, поэтому подключение и запросы остальных пользователей не оказывают большого влияния на загрузку процессора. Основным недостатком данной архитектуры является наличие возможности большой загрузки процессора в случае выполнения сложных запросов [3];
- клиент-серверная архитектура – предназначена для решения проблем файл-серверной архитектуры путем разделения компонентов приложения и их перемещения туда, где они будут максимально эффективно функционировать. Характерной чертой данной архитектуры является применение выделенного сервера БД, который отвечает за реализацию всех операций с базой данных. Кроме того, клиент-серверная архитектура характеризуется лучшими показателями масштабируемости [16];
- многоуровневая архитектура – системы с многоуровневой архитектурой состоят из трех уровней:
 - нижний, на котором располагаются приложения пользователей;
 - средний – сервер приложений, который реализует прикладную логику;
 - верхний – удаленный сервер, в задачи которого входит реализация файловых операций [29];

- технологии интернет/интернет – эти системы используются для браузерных приложений.

Таким образом, в результате анализа существующих технологий, можно сделать вывод о том, что для решения поставленной задачи достаточно реализовать файл-серверную архитектуру, которая не потребует дополнительных трудозатрат.

2.2.2 Функциональная схема проекта

На рисунке 16 представлена функциональная схема приложения [37].

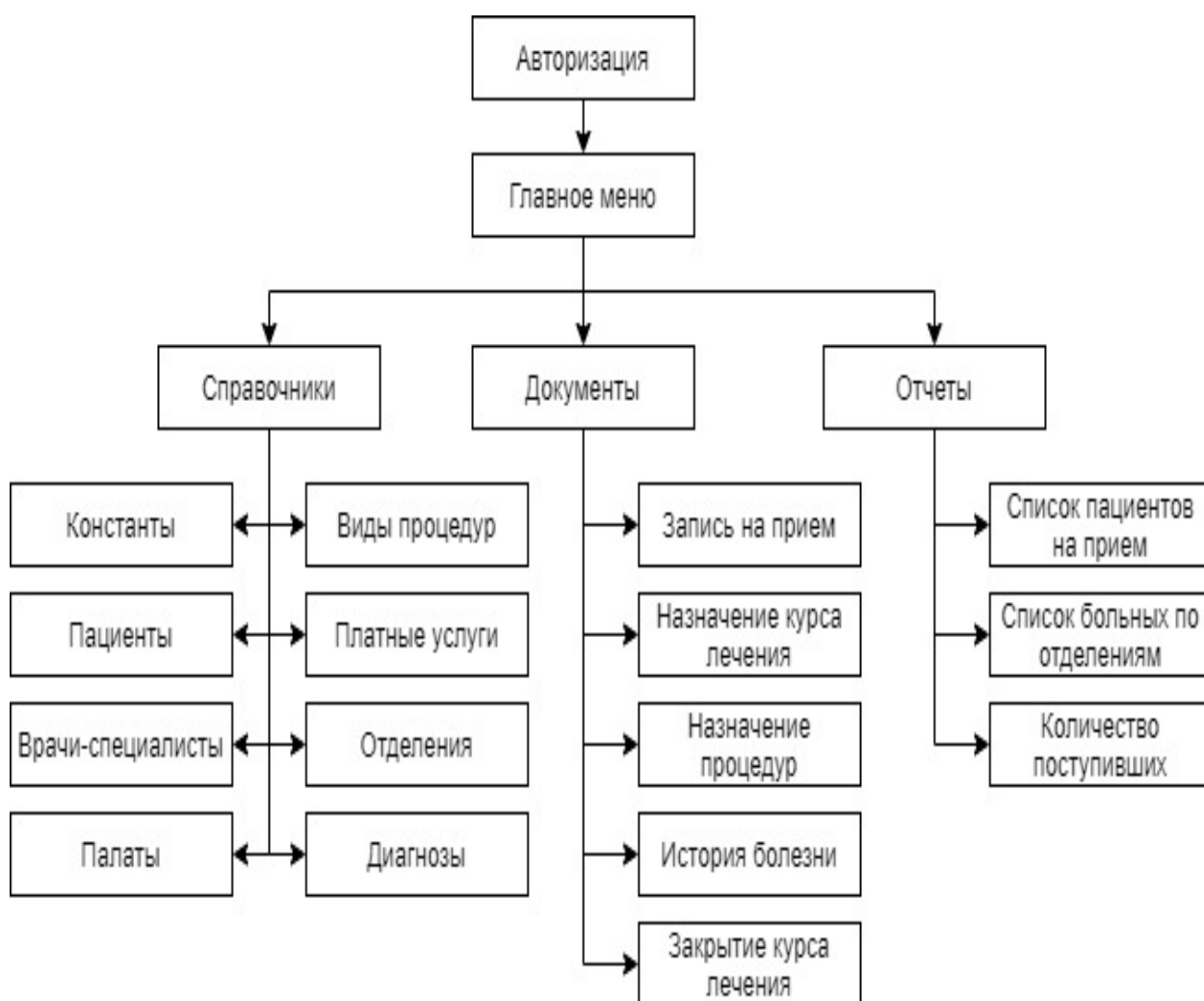


Рисунок 16 - Функциональная схема приложения

Из данного рисунка видно, что после успешной авторизации в системе, приложение при помощи главного меню предоставляет ряд функций, а именно:

- работа со справочниками:
 - константы – просмотр основных данных об организации;
 - пациенты – работа с данными физических лиц, которым оказываются услуги;
 - врачи-специалисты – работа с данными врачей (возможность редактирования должности, специализации, кабинета);
 - палаты – перечень палат клиники;
 - виды процедур – перечень процедур, реализуемых в клинике;
 - платные услуги – работа со списком платных услуг;
 - отделения – перечень доступных отделений;
 - диагнозы – список диагнозов, упрощающий работу врачей-специалистов в процессе заполнения данных;
- работа с документами:
 - запись на прием – реализация поиска свободного врача в конкретную дату;
 - назначение курса лечения – процедура закрепления за пациентом конкретного курса лечения с помещением его в определенную палату;
 - назначение процедур – добавление некоторого количества процедур конкретному пациенту;
 - история болезни – просмотр данных о болезнях конкретного пациента;
 - закрытие курса лечения – завершение лечения;
- просмотр отчетов – данный функционал не является обязательным и заложен для последующего улучшения информационной системы [8].

2.2.3 Структурная схема проекта

Структурно проект состоит из ряда модулей, именуемых в языке Delphi юнитами. Их функциональное назначение описано ниже [40]:

- Unit1 – главная форма приложения;
- Unit2 – справочник «Физические лица»;
- Unit3 – форма создания нового физического лица;
- Unit4 – форма редактирования данных о физическом лице;
- Unit5 – справочник «Комнаты»;
- Unit6 – форма добавления новой палаты;
- Unit7 - форма редактирования данных о палате;
- Unit8 - справочник «Специалисты»;
- Unit9 - форма добавления нового врача-специалиста;
- Unit10 - форма редактирования данных о враче;
- Unit11 - справочник «Виды процедур» (платные услуги);
- Unit12 - форма добавления новой платной услуги;
- Unit13 - форма редактирования данных о платной услуге;
- Unit14 - справочник «Отделения больницы»;
- Unit15 - форма добавления нового отделения больницы;
- Unit16 - форма редактирования данных о программе лечения;
- Unit17 – форма отображения данных о клинике;
- Unit18 – форма назначения курса лечения;
- Unit19 - справочник «Виды процедур»;
- Unit20 - форма добавления новой процедуры;
- Unit21 - форма просмотра журнала курсов лечения;
- Unit22 – форма назначения процедур;
- Unit23 – форма просмотра истории болезни;
- Unit24 – форма авторизации;
- Unit25 – форма установки прав пользователей;

- Unit26 – форма создания записи в истории болезней;
- Unit27 – форма закрытия курса лечения;
- Unit28 - справочник «Диагнозы»;
- Unit29 - форма добавления нового диагноза;
- Unit30 - форма редактирования данных о диагнозе;
- Unit31 – форма записи на прием.

Структурная схема проекта с логической группировкой модулей представлена на рисунке 17 [2].

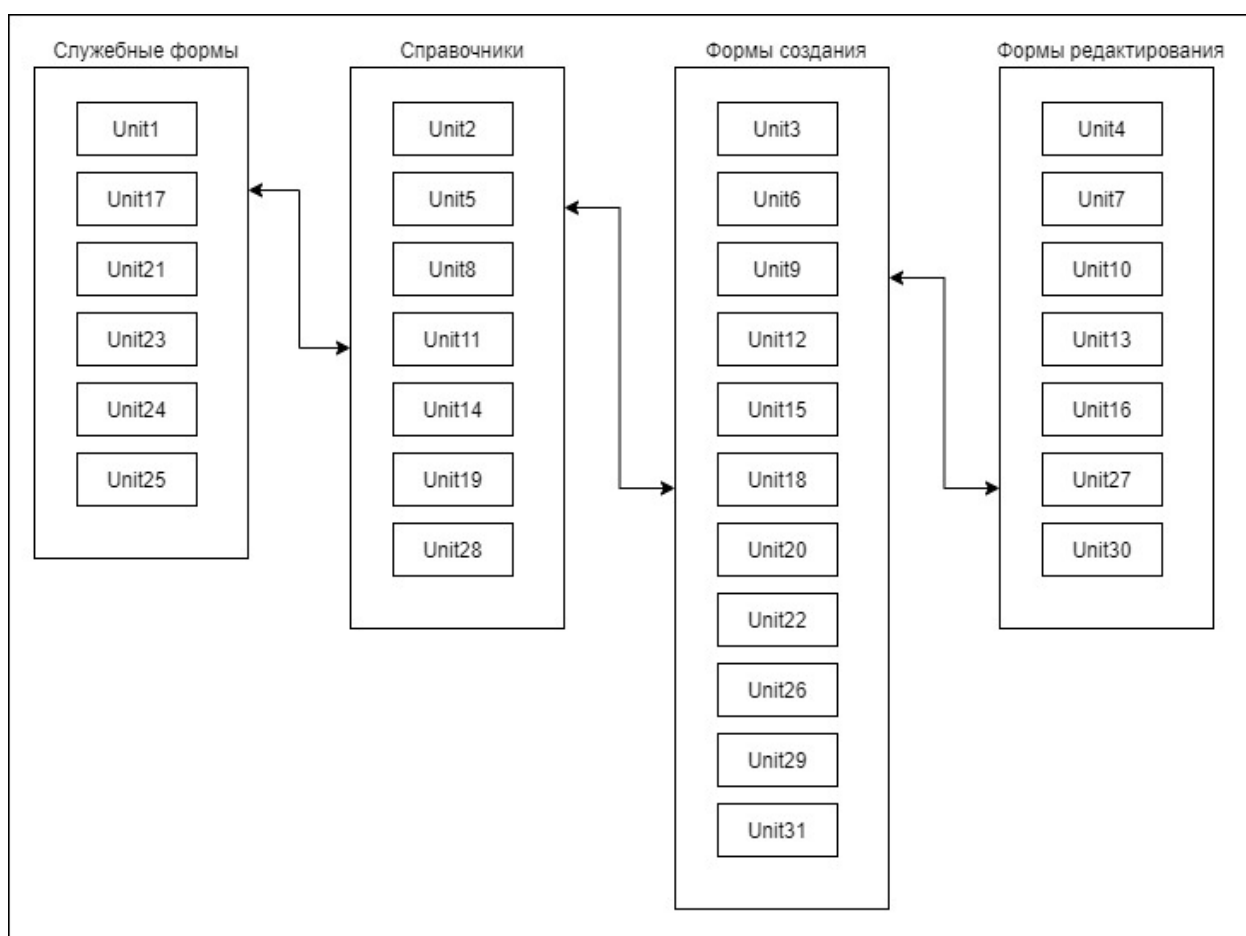


Рисунок 17 - Структурная схема проекта

Отдельно стоит отметить, что все группы взаимодействуют между собой.

На рисунке 18 представлена диаграмма классов приложения.

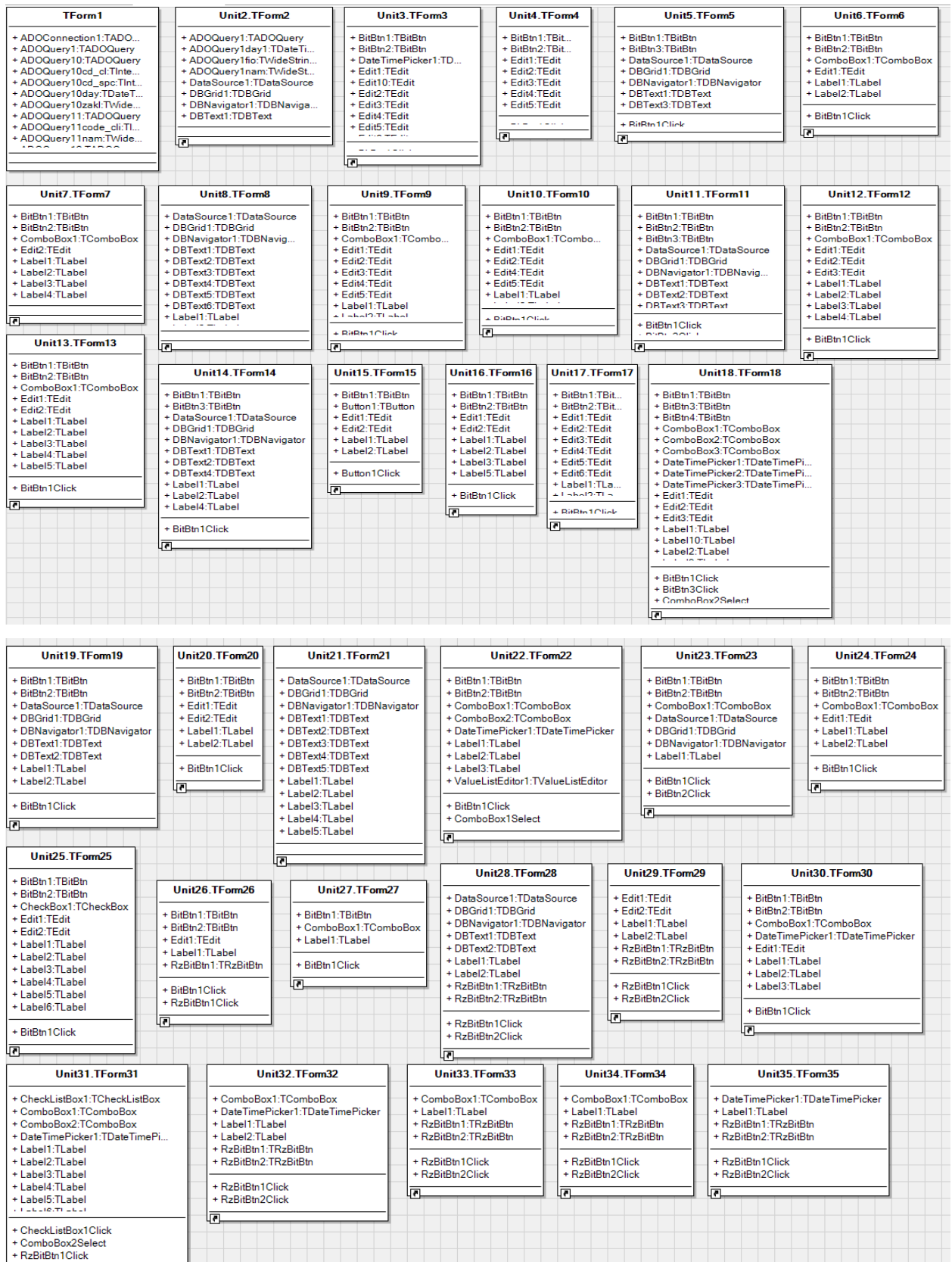


Рисунок 18 – Диаграмма классов приложения

2.2.4 Описание программных модулей

Основная логика работы информационной системы построена на взаимодействии с базой данных посредством sql-запросов.

Основные типы запросов, используемые в системе:

- insert into – добавление записи в базу данных. Алгоритм данной процедуры может быть описан двумя шагами:
 - заполнение исходных данных;
 - выполнение запроса;
- update – обновление записи в базе данных. Алгоритм процедуры обновления аналогичен процедуре создания;
- select – выборка данных. Данный вид запроса требует заполнения ряда параметров для отбора, после чего также выполняется команда [20].

2.2.5 Схема взаимосвязи программных модулей и информационных файлов

«В силу того, что информационный файл программы – база данных в формате mdb, схема взаимосвязи программных модулей простейшая и представлена на рисунке 19» [1].

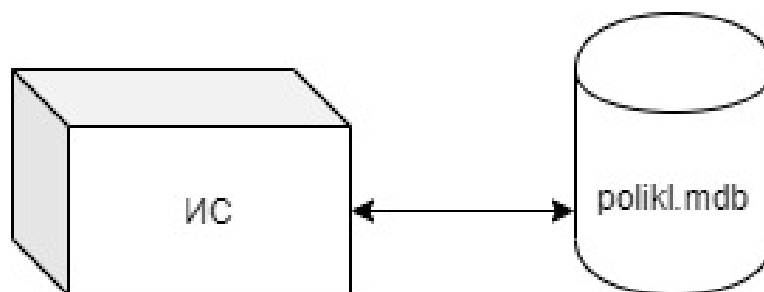


Рисунок 19 – Схема взаимосвязи программных модулей и информационных файлов

2.3 Технологическое обеспечение задачи

2.3.1 Организация технологии сбора, передачи, обработки и выдачи информации

В начале работы с информационной системой на экран выводится главное окно приложения, в котором доступны только две возможности:

- авторизация;
- выход.

Работа с информационной системой осуществляется в событийных и диалоговых режимах. В данном случае диалогом называется предоставление пользователю некоторого количества альтернатив с последующей обработкой его выбора [10].

Событиями именуется процессы, которые активизируются действиями пользователя, а также программные события – получение фокуса определенным полем, а также редактирование или потеря фокуса ввода. На основании указанных событий запускаются процедуры контроля корректности данных.

В основе разработанной информационной системы лежит интерактивный режим ввода данных. Исходными ресурсами (базой данных) приложения являются файлы оперативной информации, которые подгружаются по запросу [17].

2.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

На рисунке 20 изображена схема технологического обеспечения.



Рисунок 20 – Схема технологического процесса сбора, передачи, обработки и выдачи информации

Сбор информации реализуется путем ввода данных на формах создания и редактирования. Эти данные программным путем обрабатываются, а затем передаются в базу данных в виде, соответствующих SQL-запросов.

Выдача информации реализуется посредством служебных форм и форм справочников [28].

2.4 Контрольный пример реализации проекта и его описание

В данном разделе описывается тестирование разработанного программного обеспечения. Установка программы производится из файла дистрибутива. Запуск программы проводится с помощью ярлыка, созданного установщиком на рабочем столе.

Для корректной работы информационной системы в части формирования отчетов на рабочих станциях пользователей должен быть установлен OpenOffice.org [34].

На рисунке 21 приведена главная форма приложения [7].

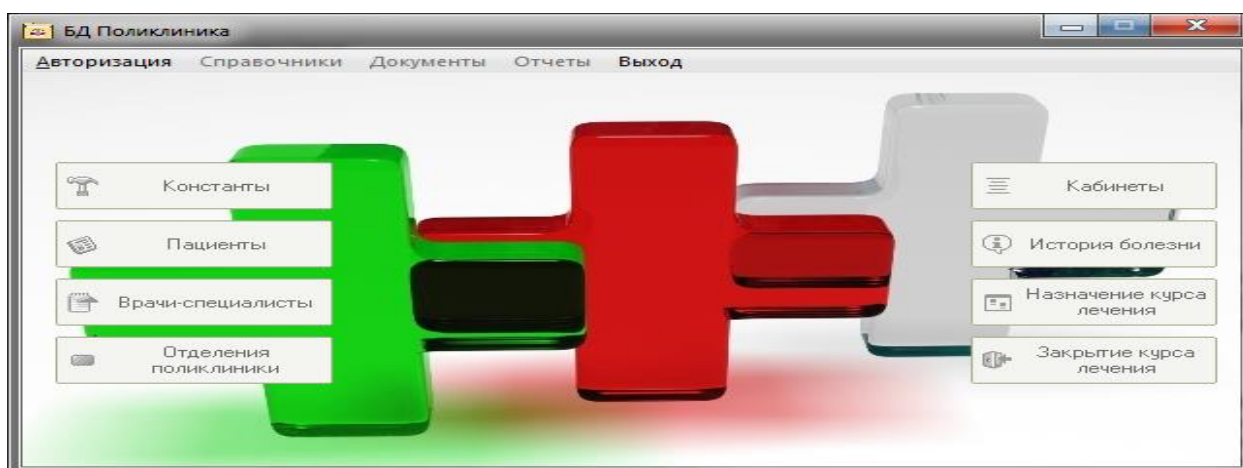


Рисунок 21 - Главная форма приложения

Для входа в систему необходимо зарегистрироваться, нажав кнопку «Авторизация», выбрать ФИО пользователя и ввести пароль. В случае успешной авторизации программа открывает режимы, доступные пользователю в соответствии с его полномочиями. Первоначально для всех пользователей установлен пароль «1» (см. рисунок 22).

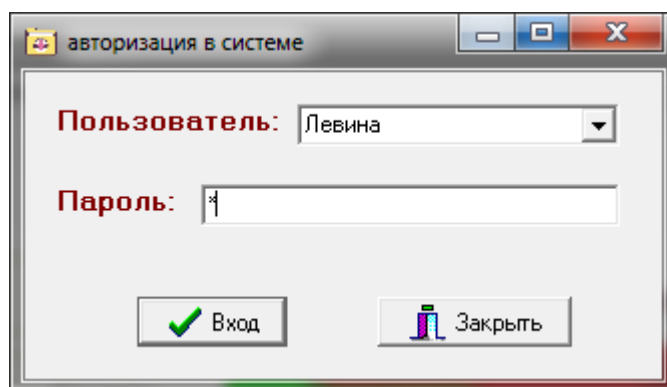


Рисунок 22 - Окно авторизации

На рисунке 23 показан режим работы с данными о медицинском учреждении.

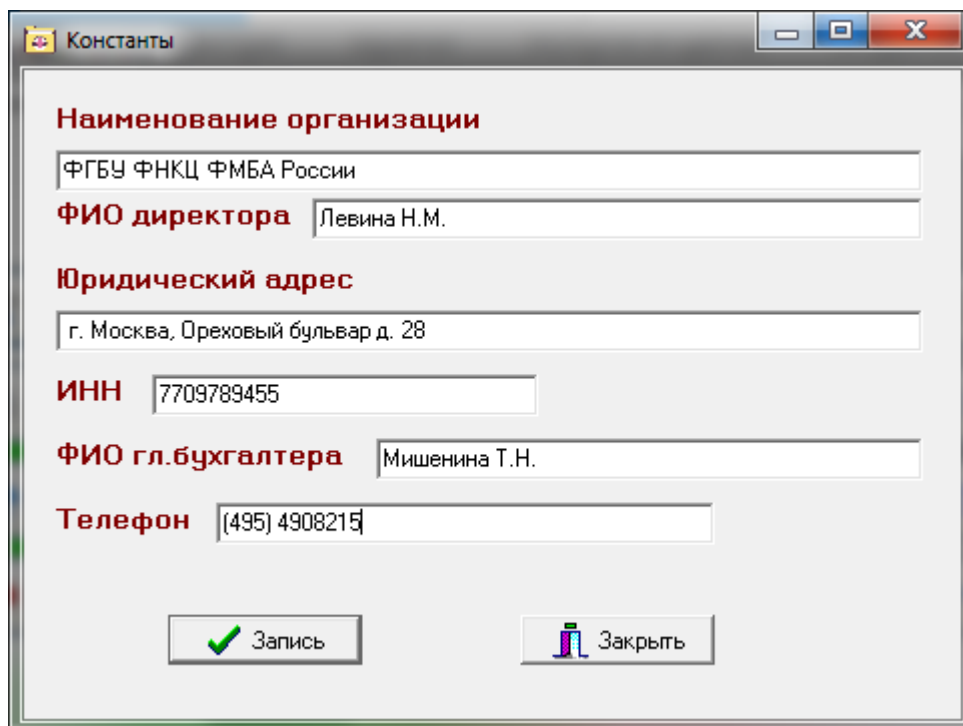


Рисунок 23 - Справочник «Информация о поликлинике»

Режим справочника пациентов приведен на рисунке 24. В данном режиме доступен просмотр истории болезни и установленных диагнозов.

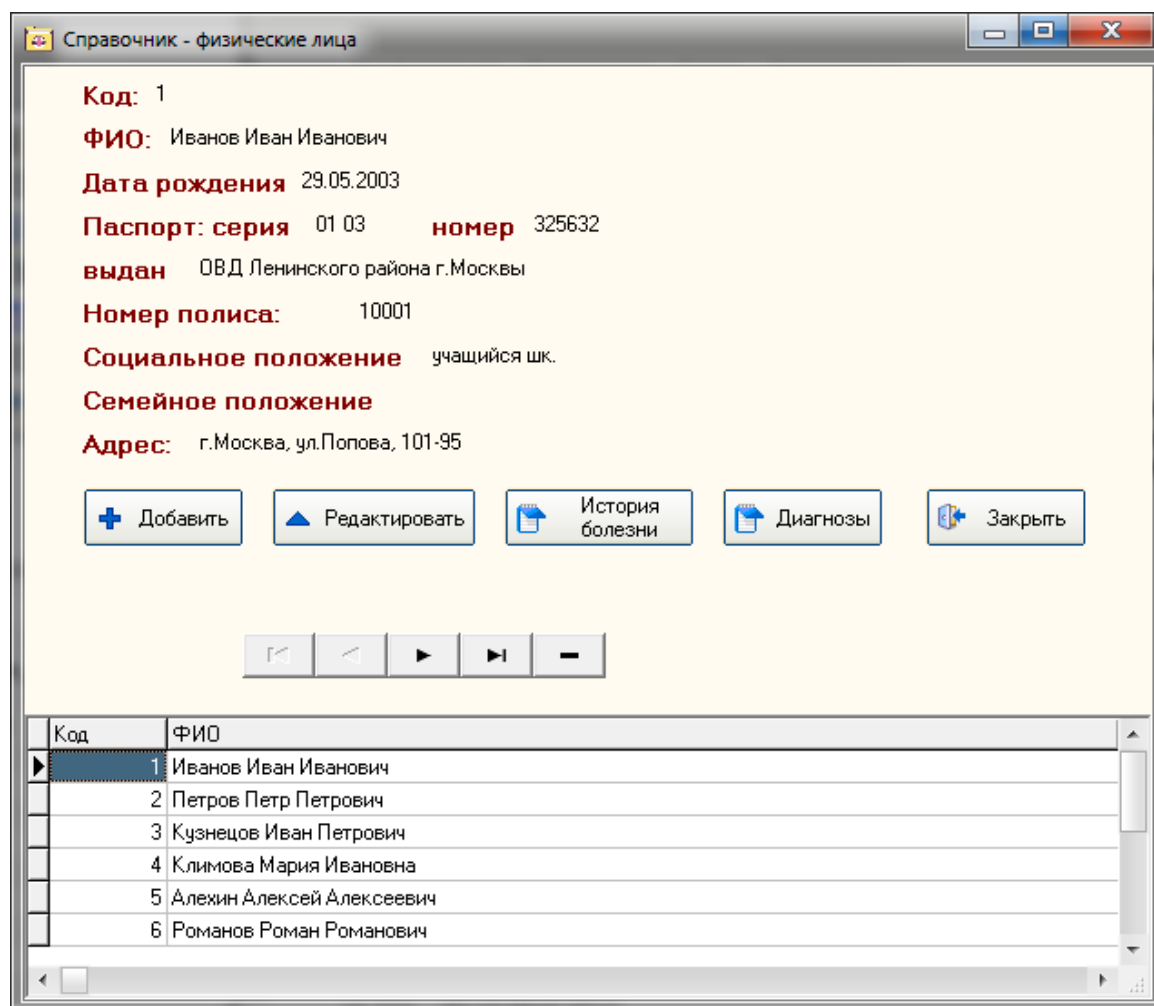


Рисунок 24 - Справочник «Пациенты»

На рисунке 25 приведен режим добавления записей в справочник физических лиц. Из режима справочника пациентов доступен просмотр истории болезни и диагнозов текущего пациента.

На рисунке 26 показан режим печати списка диагнозов пациента.

На рисунке 27 показан режим печать истории болезни пациента.

На рисунках 28-29 показан режим справочника специалистов.

На рисунке 30 показан справочник видов процедур.

На рисунке 31 представлен справочник диагнозов.

На рисунке 32 показан режим записи на прием, после подбора врача и желаемой даты приема в правой части формы отображаются доступные

талоны на прием, которые после выбора времени приема могут быть распечатаны (рисунок 33).

Рисунок 25 - Добавление новой записи в справочник «Пациенты»

ФИО
пациента: Кузнецов Иван Петрович

№	Дата установления	ФИО специалиста	Диагноз
1	22.12.2016	Левина Наталия Марковна	Артериальная гипертензия

Рисунок 26 - Просмотр списка диагнозов пациента

ФИО пациента: Кузнецов Иван Петрович
Адрес Кузнецов Иван Петрович

№	Дата	ФИО специалиста	Заключение
1	22.12.2016 16:56:46	Левина Наталия Марковна	Поступление с АД 160/100

Рисунок 27 - История болезни

На рисунке 34 показан режим назначения курса лечения.

На рисунках 35-36 показан отчет о принятых пациентах [22, 26].

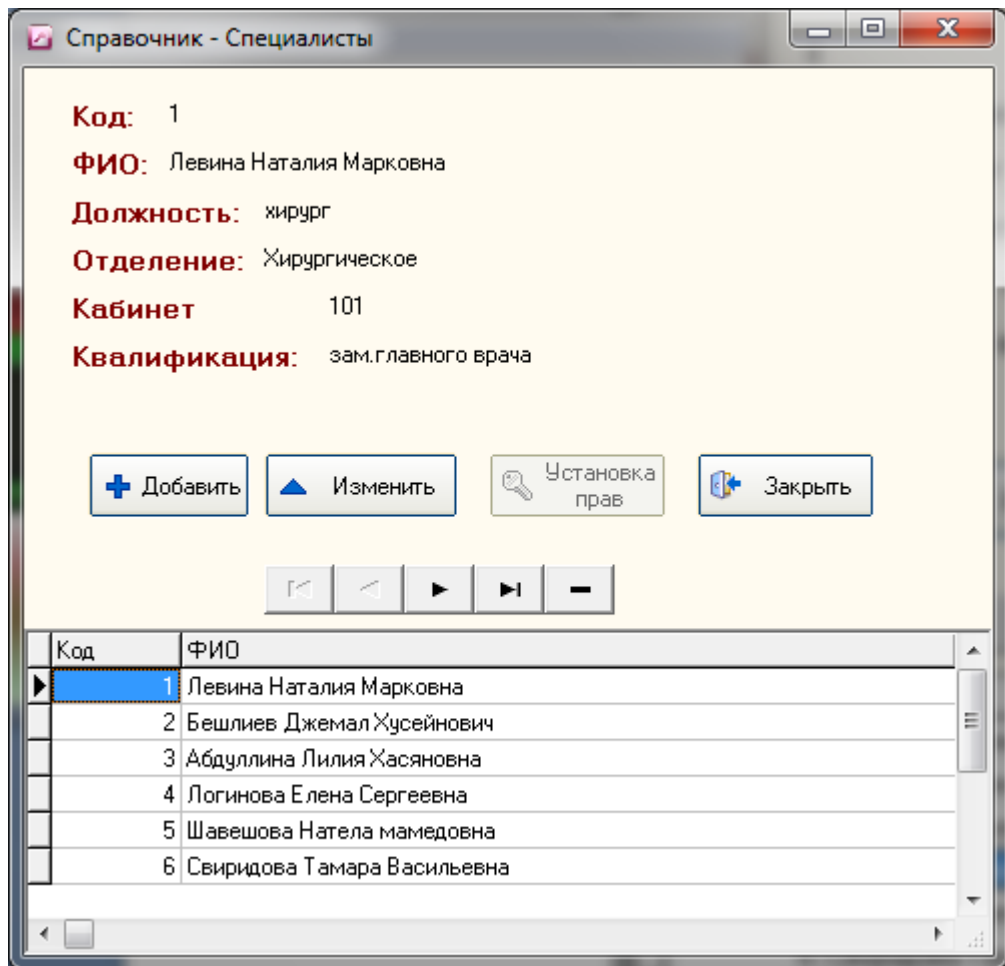


Рисунок 281 - Справочник «Специалисты»

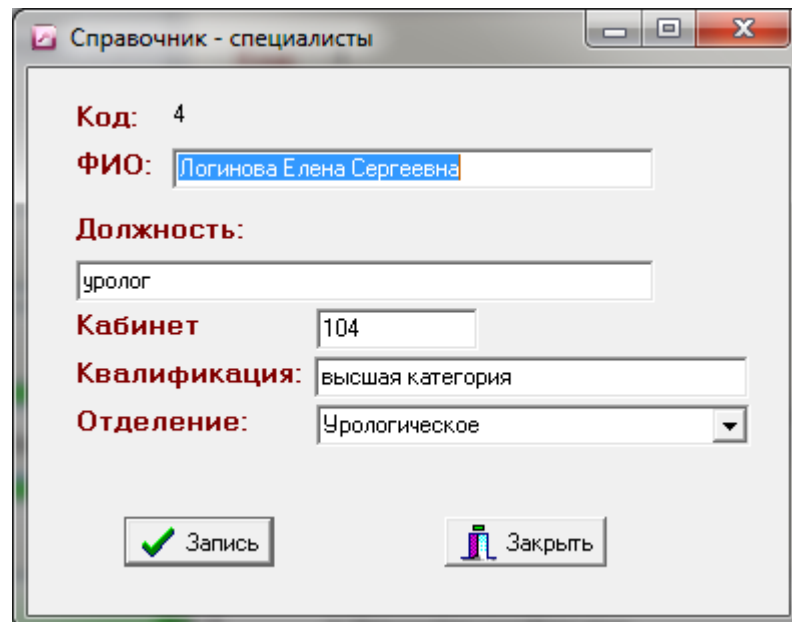


Рисунок 29 - Изменение текущей записи в справочник «Специалисты»

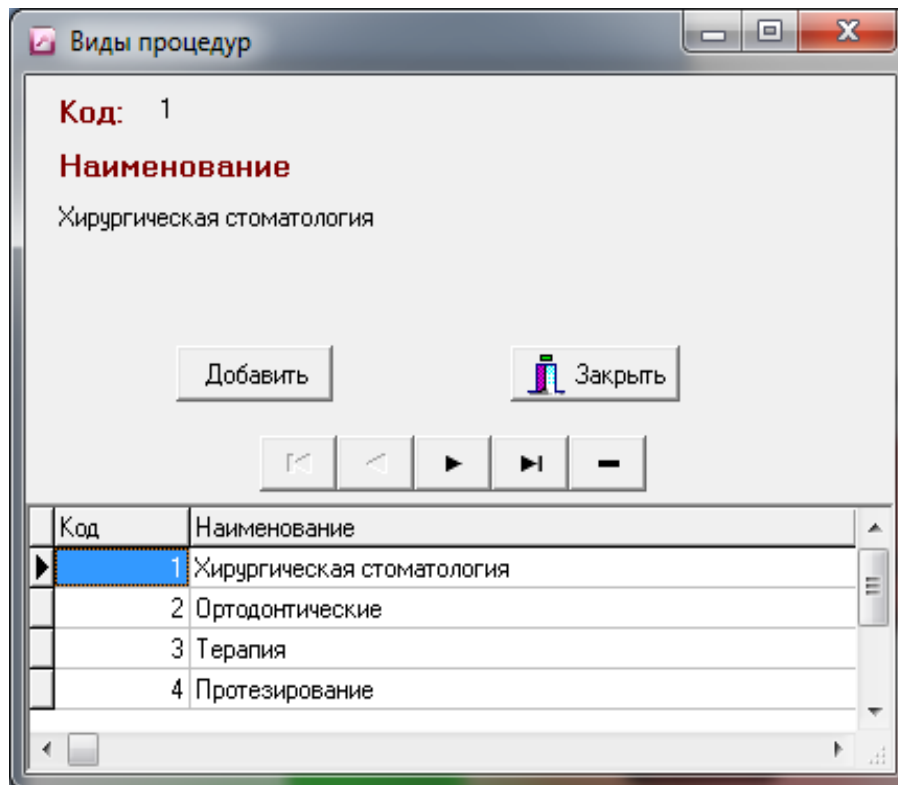


Рисунок 30 - Виды процедур

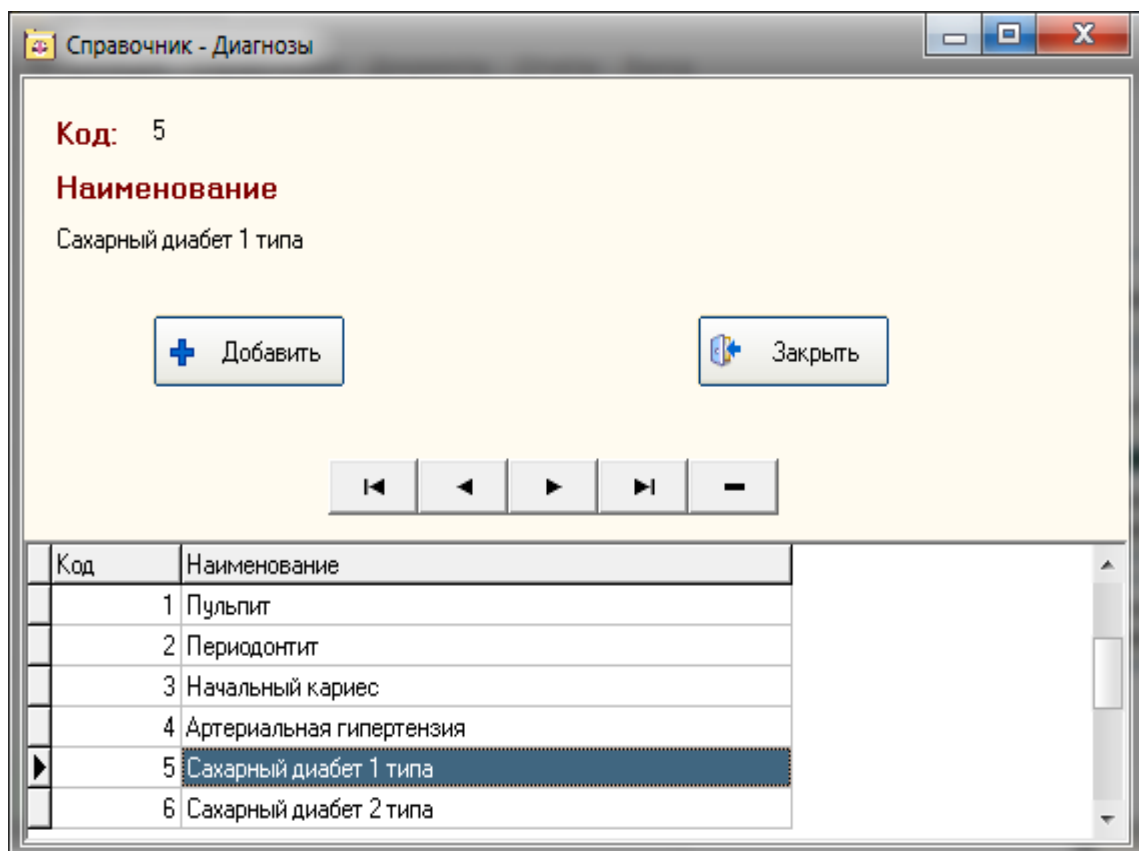


Рисунок 31 - Справочник диагнозов

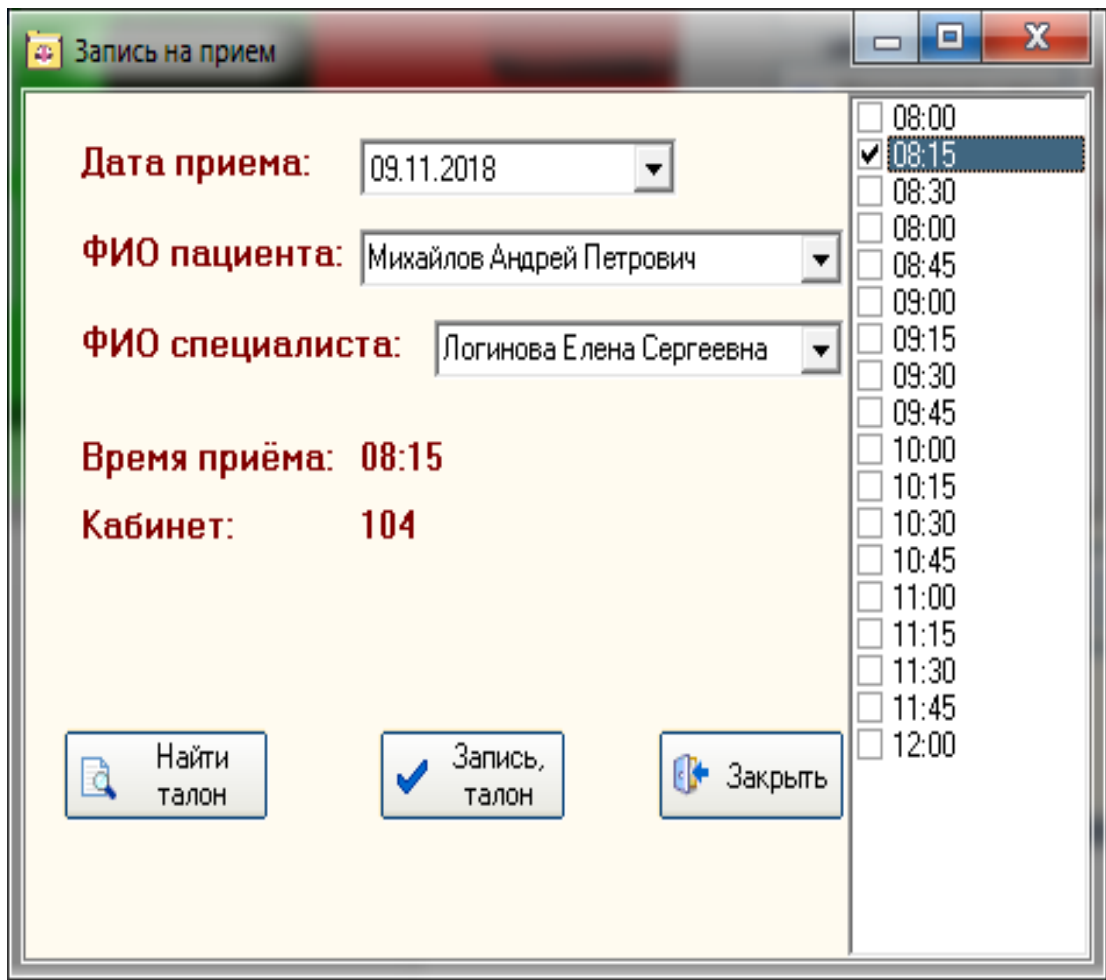


Рисунок 32 - Режим «Запись на прием»

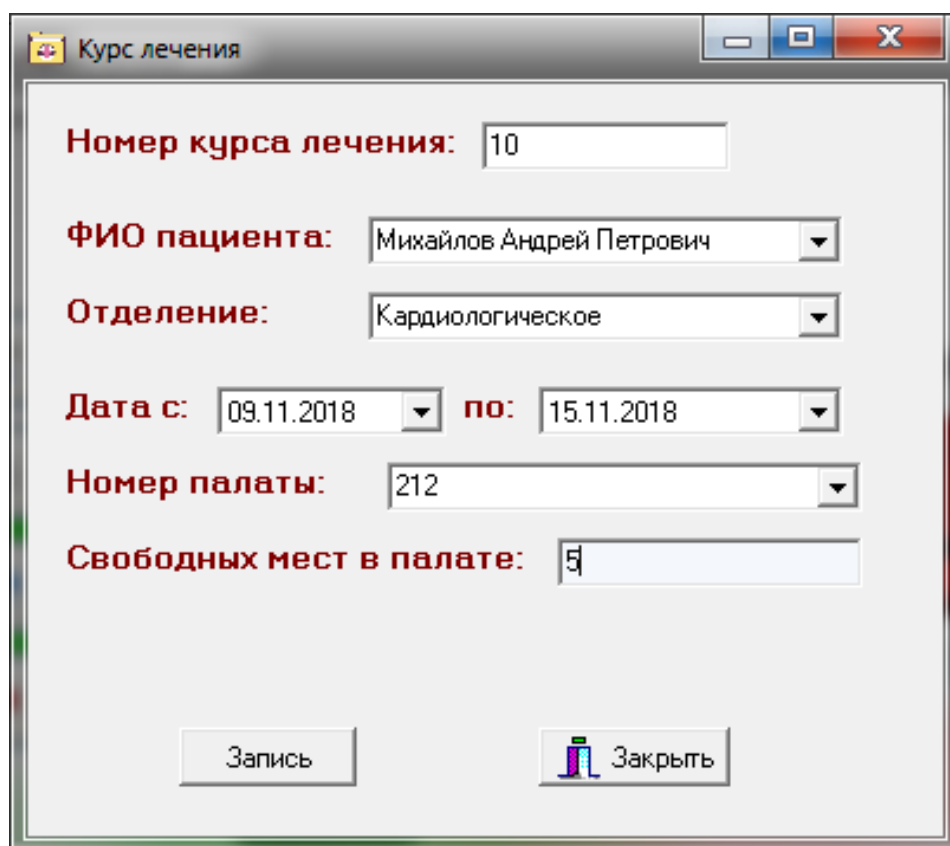
ФГБУ ФНКЦ ФМБА России

г. Москва, Ореховый бульвар д. 28

Талон на приём

ФИО пациента:	Михайлов Андрей Петрович
Дата:	09.11.2018
Время:	104
ФИО врача:	Логинова Елена Сергеевна
Кабинет:	08:15

Рисунок 33 – Талон на прием ко врачу



Курс лечения

Номер курса лечения: 10

ФИО пациента: Михайлов Андрей Петрович

Отделение: Кардиологическое

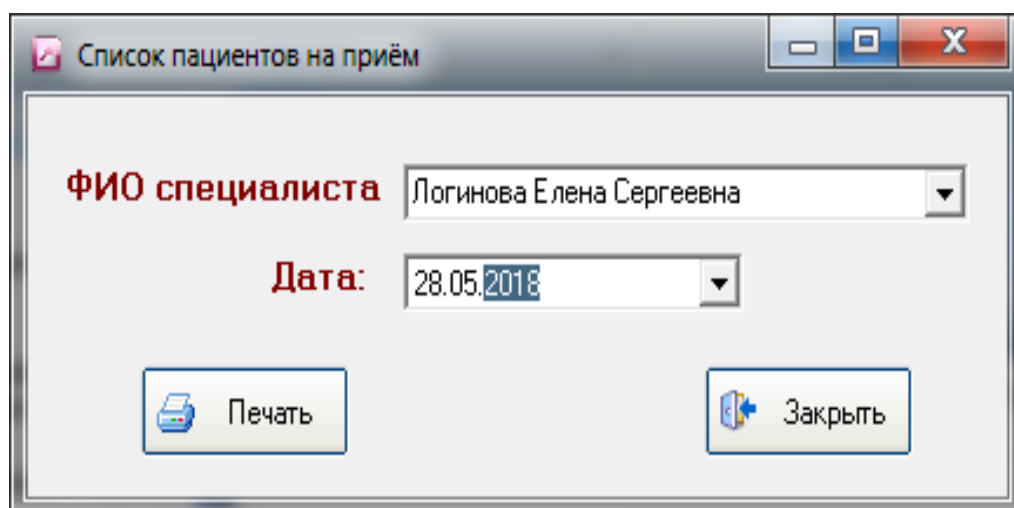
Дата с: 09.11.2018 по: 15.11.2018

Номер палаты: 212

Свободных мест в палате: 5

Запись Закреть

Рисунок 34 - Назначение курса лечения



Список пациентов на приём

ФИО специалиста: Логинова Елена Сергеевна

Дата: 28.05.2018

Печать Закреть

Рисунок 35 - Запрос отчета о принятых пациентах

Дата:

28.05.2018

ФИО специалиста: Логинова Елена Сергеевна

Время	ФИО пациента
09:30	Кузнецов Иван Петрович
09:45	Иванов Иван Иванович
10:00	Томина Марина Михайловна
10:30	Попов Виктор Михайлович

Рисунок 36 - Отчет о пациентах

Выводы по главе 2

Данная глава содержит описание практической части диплома, а именно:

- логическое моделирование предметной области;
- физическое моделирование предметной области;
- технологическое обеспечение задачи;
- контрольный пример реализации.

ГЛАВА 3 ОЦЕНКА И ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1 Выбор и обоснование методики расчета экономической эффективности

Показатель экономической эффективности проекта используется для выявления необходимости внедрения рассматриваемого проекта. В основе расчета экономической эффективности лежит сопоставление существующего и внедряемого методов обработки данных.

В большинстве случаев базой сравнения является ручной способ реализации работ.

Достижимый эффект характеризуется путем сопоставления экономии от использования информации с затратами на ее получение.

Трудоемкость работ по реализации проекта вычисляется с учетом срока завершения работ, используемого языка программирования, а также объема реализуемых функций.

ГОСТ 19.102-77 «Единая система программной документации» определяет комплекс работ по реализации проекта. При этом руководитель формулирует постановку задачи и отвечает за работу по созданию информационной системы.

В обязанности исполнителя входит ответственность за проектирование информационного и методического обеспечения, организация разработки программного обеспечения, ответственность за корректную работу системы.

Для разработки «нового программного продукта трудоемкость вычисляется на основе трудоемкости разработки аналогичного ПО с учетом отличительных черт данного проекта, отражаемых использованием специальных поправочных коэффициентов» [15].

3.2 Расчет показателей экономической эффективности проекта

Чтобы вычислить затраты, необходимые на разработку проекта Q_p применяется формула 1:

$$Q_p = \sum t_i \quad (1)$$

где, t_i – i-й этап проекта.

В таблице 10 отражены основные этапы проекта.

Таблица 10 - Этапы проекта

Этап	№ работы	Содержание работы	Трудоемкость	
			(чел-час)	(чел-дни)
Изучение архитектуры Жигулевской ЦГБ	1	Изучение бизнес-процессов отдела платных услуг Жигулевской ЦГБ	24	3
	2	Изучение используемых программных средств	56	7
Проектирование базы данных	1	Сбор данных, необходимых для описания сущностей	24	3
	2	Построение логической модели БД	32	4
Разработка автоматизированной информационной системы	1	Сбор первоначального списка требований	24	3
	2	Выбор средств разработки приложения	8	1
	3	Проектирование будущей информационной системы	56	7
	4	Написание программного кода	168	21
Тестирование полученной информационной системы	1	Разработка тестовых сценариев для созданной системы	24	3
	2	Проведение тестов для функций системы, а также сбор результатов тестирования	56	7
	3	Анализ полученных результатов тестирования	16	2
	4	Внесение изменений в программный код (в случае необходимости)	56	7

Чтобы определить количество времени, которое понадобится на реализацию каждой работы, применяется технология *planning poker*,

относящаяся к методологии scum. При помощи данной технологии, каждый разработчик самостоятельно определяет время, которое понадобится ему на выполнение конкретной задачи, опираясь на собственный опыт.

Чтобы вычислить количество чел-часов, количество чел-дней умножается на рабочее время (стандартный восьмичасовой рабочий день).

Получается, $Q_p = 68$ чел-дней = 544 чел-часов.

Число исполнителей, которое требуется для реализации проекта, можно вычислить по формуле 2 [13]:

$$N = \frac{Q_p}{F}, \quad (2)$$

«где Q_p – затраты труда, необходимые для реализации проекта, F – фонд рабочего времени, величина которого определяется по формуле 3» [39]:

$$F = T \cdot F_M, \quad (3)$$

«где T – число месяцев, необходимое для реализации проекта, F_M – фонд времени в рамках конкретного месяца. Значение F_M вычисляется по формуле 4» [39]:

$$F_M = \frac{t_p \cdot (D_K - D_B - D_{\Pi})}{12} \quad (4)$$

«где t_p – продолжительность рабочего дня, D_K – число рабочих дней в году, D_B – число выходных дней в году, D_{Π} – число праздничных дней в году» [39]. Получается, $F_M = \frac{8(365-118)}{12} \approx 165$.

Таким образом, $F = 2 * 165 = 330$. В данном случае 2 – количество месяцев, затраченных на реализацию проекта.

Тогда число исполнителей $N = \frac{544}{330} \approx 1$.

В таблице 11 представлена информация о работах, необходимых для построения сетевой модели, и времени, затрачиваемом на реализацию указанных выше работ.

На рисунке 37 изображен график сетевой модели.

Критический путь модели охватывает вершины:

0-1-2-3-4-5-6-7-8-9-10.

Таблица 11 - Работы проекта

№	Событие	Код работы	Работа	t, чел-дни
0	Начало проекта	0-1	Изучение бизнес-процессов отдела платных услуг Жигулевской ЦГБ	3
1	Бизнес-процессы изучены	1-2	Изучение используемых программных средств	7
2	Используемые программные средства изучены	2-3	Сбор данных, необходимых для описания сущностей	3
3	Данные о сущностях будущей системы собраны	3-4	Построение логической модели БД	4
4	Требования к системе собраны	4-5	Сбор первоначального списка требований	3
5	Модель базы данных построена	3-6	Выбор средств разработки приложения	1
6	Определены инструменты для разработки системы	6-7	Проектирование будущей информационной системы	7
7	Составлен проект клиентской части информационной системы	7-8	Написание программного кода	21
8	Завершено написание программного кода	3-9	Разработка тестовых сценариев для созданной системы	3
9	Получены результаты тестирования информационной системы	8-9	Проведение тестов для функций системы, а также сбор результатов тестирования	2
10	Проведен анализ результатов тестирования	9-10	Анализ полученных результатов тестирования	7
11	Внесены изменения в код (в случае необходимости)	10-11	Ожидание внесения изменений	0

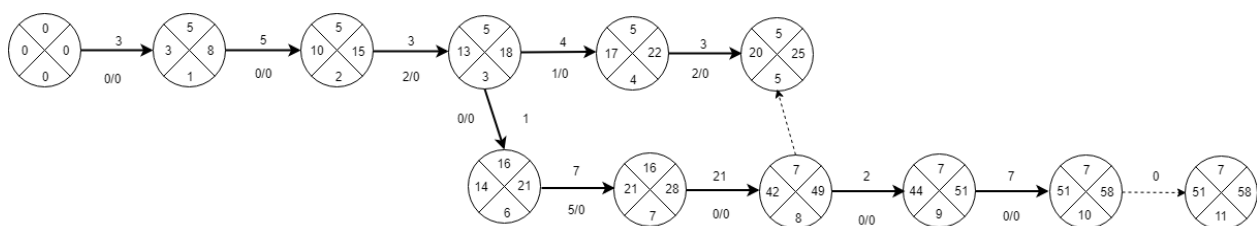


Рисунок 37 - Сетевая модель

Для определения данных, отраженных на графике сетевой модели, были проведены расчеты ранних и поздних сроков совершения событий, а также резервов времени событий.

Расчёт ранних сроков наступления событий:

$$T_0^p = 0 \quad (5)$$

$$T_1^p = 0 + 3 = 3 \quad (6)$$

$$T_2^p = 3 + 7 = 10 \quad (7)$$

$$T_3^p = 10 + 3 = 13 \quad (8)$$

$$T_4^p = 13 + 4 = 17 \quad (9)$$

$$T_5^p = 17 + 3 = 20 \quad (10)$$

$$T_6^p = 13 + 1 = 14 \quad (11)$$

$$T_7^p = 14 + 7 = 21 \quad (12)$$

$$T_8^p = 21 + 21 = 42 \quad (13)$$

$$T_9^p = 42 + 2 = 44 \quad (14)$$

$$T_{10}^p = 44 + 7 = 51 \quad (15)$$

$$T_{11}^p = 51 + 0 = 51 \quad (16)$$

Расчёт поздних сроков наступления событий:

$$T_{11}^n = 58 \quad (17)$$

$$T_{10}^n = 58 \quad (18)$$

$$T_9^n = 58 - 7 = 51 \quad (19)$$

$$T_8^n = 51 - 2 = 49 \quad (20)$$

$$T_7^n = 49 - 21 = 28 \quad (21)$$

$$T_6^n = 28 - 7 = 21 \quad (22)$$

$$T_5^n = 25 \quad (23)$$

$$T_4^n = 25 - 3 = 22 \quad (24)$$

$$T_3^n = 22 - 4 = 18 \quad (25)$$

$$T_2^n = 18 - 3 = 15 \quad (26)$$

$$T_1^{\Pi} = 15 - 7 = 8 \quad (27)$$

$$T_0^{\Pi} = 0 \quad (28)$$

На рисунке 38 изображена диаграмма Ганта, построенная на базе таблицы 10, которая отображает календарный график работ.

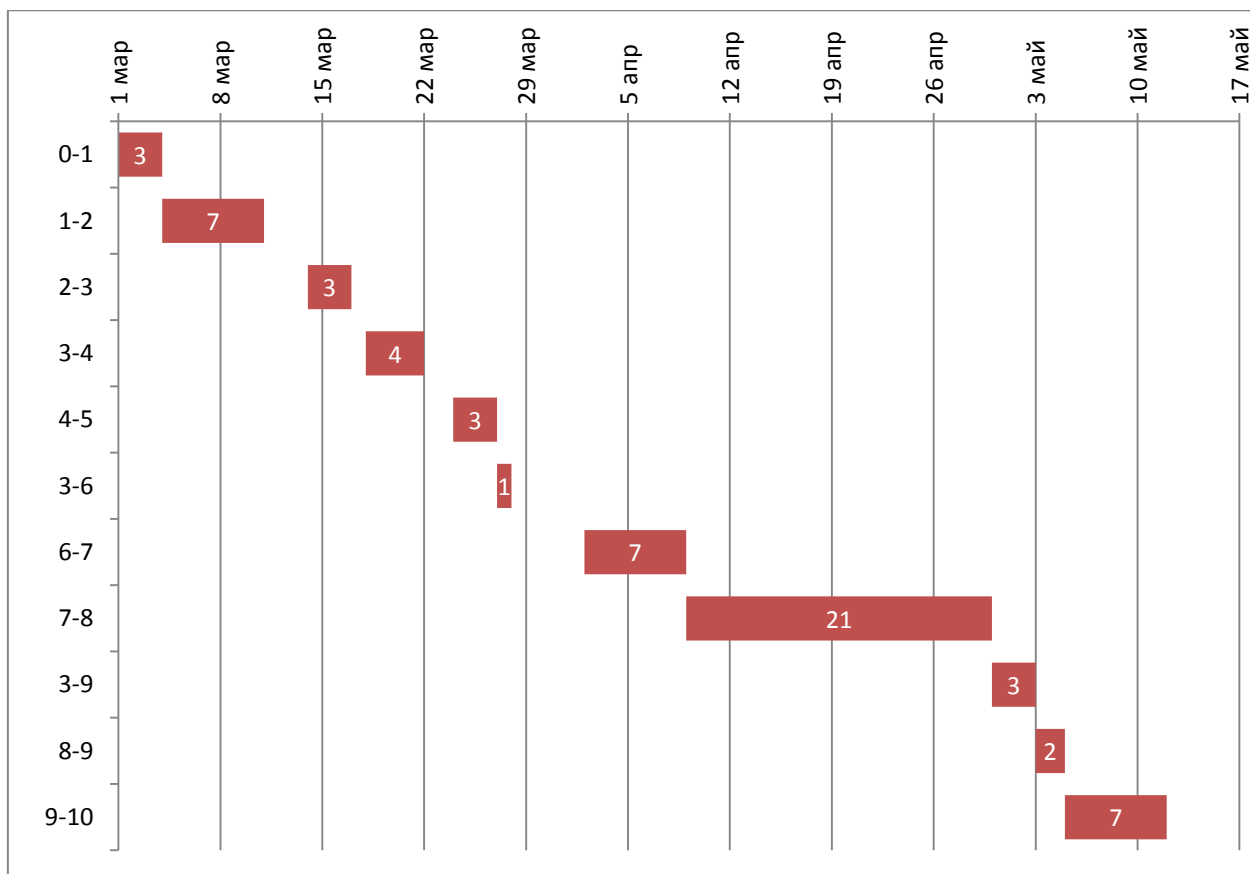


Рисунок 38 - Диаграмма Ганта

Затраты на реализацию проекта вычисляются по формуле 29:

$$K = C_{\text{ЗАРП}} + C_{\text{ОБ}} + C_{\text{ОРГ}} + C_{\text{НАКЛ}} \quad (29)$$

где, K – общие затраты, $C_{\text{ЗАРП}}$ – заработная плата исполнителей проекта, $C_{\text{ОБ}}$ – затраты, необходимые на оборудование, $C_{\text{ОРГ}}$ – затраты, необходимые для организации рабочих мест, $C_{\text{НАКЛ}}$ – накладные расходы.

$C_{\text{ЗАРП}}$ определяется из следующего соотношения:

$$C_{\text{ЗАРП}} = C_{\text{З.ОСН}} + C_{\text{З.ДОП}} + C_{\text{З.ОТЧ}} \quad (30)$$

где, $C_{з.осн}$ и $C_{з.доп}$ – основная и дополнительная заработная плата соответственно, $C_{з.отч}$ – отчисления с заработной платы.

Для вычисления $C_{з.осн}$ используется формула 31:

$$C_{з.осн} = T_{зан} \cdot O_{дн} \quad (31)$$

где $T_{зан}$ – число дней, которое отработал исполнитель, $O_{дн}$ – дневной оклад исполнителя.

Дневной оклад исполнителя вычисляется по формуле 32:

$$O_{дн} = \frac{O_{мес} \cdot 8}{F_M} \quad (32)$$

где $O_{мес}$ – оклад за месяц, F_M – фонд рабочего времени.

$$O_{мес} = 18000 \cdot 1,13 = 20340 \text{ р.} \quad (33)$$

Таким образом, $O_{дн} = \frac{20340 \cdot 8}{165} \approx 986,2$ р.

$$C_{з.осн} = 986,2 \cdot 68 = 67060,36 \text{ р.} \quad (34)$$

В таблице 12 представлены данные о затратах на основную заработную плату.

Таблица 12 - Основная заработная плата

№	Должность	Оклад	Дн. оклад	Труд. затраты	Зар. плата
1	Программист	18000	986,2	68	104614,2

Определим дополнительную зарплату:

$$C_{з.доп} = 0,2 \cdot 67060,36 = 13412,07 \text{ р.} \quad (35)$$

Для определения размера отчислений используем следующую формулу:

$$C_{з.отч} = (67060,36 + 13412,07) \cdot 0,3 = 24141,73 \text{ р.} \quad (36)$$

Где 0,3 – размер единого социального налога в 2019 году.

Следовательно, $C_{зарп}$ определяется по следующей формуле:

$$C_{зарп} = 67060,36 + 13412,07 + 24141,73 = 104614,2 \text{ р.} \quad (37)$$

Определим затраты на оборудование $C_{об}$:

$$C_{об} = 6000 \text{ р.} \quad (38)$$

Затраты на организацию рабочих мест $C_{ОРГ}$ определяются по следующей формуле:

$$C_{ОРГ} = \frac{C_{КВМ}}{12} S \frac{T_{АР} \cdot 8}{F_M} \quad (40)$$

где $C_{КВМ}$ – стоимость за аренду квадратного метра площади в год, S – общая площадь рабочего помещения, $T_{АР}$ – период аренды. Получается,

$$C_{ОРГ} = 1041 \cdot 40 \cdot \frac{68 \cdot 8}{165} = 137285,8 \text{ р} \quad (41)$$

Величина накладных расходов задается формулой 42:

$$C_{НАКЛ} = 0,6 \cdot C_{З.ОСН} \quad (42)$$

Таким образом, $C_{НАКЛ} = 0,6 \cdot 67060,36 = 40236,22 \text{ р}$.

Таким образом, суммарные затраты на выполнение проекта составляют:

$$K = 67060,36 + 6000 + 137285,8 + 40236,22 = 250582,4 \text{ р} \quad (43)$$

Затраты на внедрение разработанной автоматизированной информационной системы вычисляются по формуле 44:

$$C_{ВН} = C_{ВН.ЗАРП} + C_{ВН.ОБ} + C_{ВН.ОРГ} + C_{ВН.НАКЛ} \quad (44)$$

«где $C_{ВН.ЗАРП}$ – заработная плата всех исполнителей, принимающих участие во внедрении, $C_{ВН.ОБ}$ – затраты на оборудование, $C_{ВН.ОРГ}$ – затраты на организацию рабочих мест сотрудников» [39], взаимодействующих с внедренным программным обеспечением, $C_{ВН.НАКЛ}$ – накладные расходы в процессе внедрения».

В силу того, что использование разрабатываемой системы не предполагает закупку нового оборудования и модернизацию рабочих мест сотрудников, параметры $C_{ВН.ОБ}$ и $C_{ВН.ОРГ}$ равны нулю. Получается, нужно вычислить $C_{ВН.ЗАРП}$ и $C_{ВН.НАКЛ}$:

$$C_{ВН.ЗАРП} = 986,2 \cdot 22 + 0,2 \cdot 986,2 \cdot 22 + \\ + (986,2 \cdot 22 + 0,2 \cdot 986,2 \cdot 22) \cdot 0,3 = 33845,76 \text{ р} \quad (45)$$

где 22 дня – время, необходимое на внедрение системы.

$$C_{ВН.НАКЛ} = 986,2 \cdot 22 \cdot 0,6 = 13017,6 \text{ р} \quad (46)$$

Получается, затраты на внедрение составляют:

$$K_{ВН} = 33845,76 + 0 + 0 + 13017,6 = 46863,36 \text{ р.} \quad (47)$$

Чтобы вычислить величину общих затрат, нужно сложить затраты на разработку и внедрение системы:

$$K_{ОБ} = K_{ВН} + K$$

Получается, значение $K_{ОБ}$ равно:

$$K_{ОБ} = 46863,36 + 250582,4 = 297445,8 \text{ р.} \quad (48)$$

Далее представлены расчеты, которые необходимы для планирования цены и прогнозирования прибыли от разрабатываемой системы.

Чтобы вычислить стоимость программной системы применяется формула 49:

$$K_{ПО} = (\Delta K + K_{ВН}) \cdot (1 + D_{ПРИБ}) \quad (49)$$

где ΔK – цена одной копии программы, $K_{ВН}$ – затраты, необходимые на внедрение информационной системы, $D_{ПРИБ}$ – процент прибыли от продажи проекта, который заранее закладывается в стоимость.

Для определения $D_{ПРИБ}$ применяется формула 50:

$$D_{ПРИБ} = \left(\frac{K_{ПР}}{\Delta K + K_{ВН}} - 1 \right) \cdot 100\% \quad (50)$$

Таким образом, $D_{ПРИБ} = 15\%$.

Для вычисления прибыли от продажи каждой установки информационной системы, применяется формула 51:

$$C_{ПРИБ} = K_{ПР} \cdot D_{ПРИБ} \cdot (1 - N_{НДС}) \quad (51)$$

где $N_{НДС}$ – размер налога на добавочную стоимость.

Получается, $C_{ПРИБ}$ равна:

$$C_{ПРИБ} = 9000 \text{ р.} \quad (52)$$

В таблице 13 приведен фрагмент таблицы прибыли и расходов по проекту.

Таблица 13 - Фрагмент таблицы расходов и прибыли по проекту

Период расчета	Разработка (проект)				Прибыль	
	Частичная стоимость	Сальдо начальное по кредиту	Погашение кредита	Сальдо конечное по кредиту	Расчетная прибыль	Чистая прибыль
1-4. 200х	88906,12	-320062,03	88906,12	-231155,9	13779,38	11023,51
5-8. 200х	88906,12	-231155,9	88906,12	-142249,79	13779,38	11023,51
9-12. 200х	88906,12	-142249,79	88906,12	-53343,67	13779,38	11023,51
1-3. 200х	88906,12	-53343,67	55006,12	0	49341,83	39473,47
4-6. 200х	88906,12	0	0	0	102685,5	82148,4

В результате представленных расчетов можно говорить о том, что проектируемая система представляет собой эффективное решение задачи автоматизации системы управления платными услугами Жигулевской ЦГБ.

Выводы по главе 3

Данная глава служит для оценки и обоснования экономической эффективности проекта. На базе представленных расчетов можно говорить о том, что проект экономически эффективен.

ЗАКЛЮЧЕНИЕ

В рамках выполнения данной работы была рассмотрена тема «Разработка автоматизированной системы управления платными услугами (на примере ГБУЗ СО «Жигулевская ЦГБ»)».

Первая часть работы носит аналитический характер. В ней представлена организационная структура ГБУЗ СО «Жигулевская ЦБ», а также выделены и описаны основные функции, среди которых:

- защита прав и свобод гражданина в сфере здравоохранения;
- проведение единой государственной политики в сфере здравоохранения;
- организация, обеспечение и реализация федерального государственного санитарно-эпидемиологического надзора;
- организация и осуществление контроля в сфере здравоохранения, в том числе за соблюдением требований технических регламентов данной сферы.

Отдельное внимание в рамках первой главы уделяется концептуальному моделированию предметной области.

Кроме того, здесь же представлены результаты сравнительной характеристики аналогов, существующих в сфере медицинского ПО, а именно:

- ArhiMed+;
- Учет пациентов;
- Дентал-Софт.

Вторая глава работы – практическая. В процесс создания базы данных были выделены следующие сущности, описывающие предметную область:

- справочник врачей-специалистов;
- справочник диагнозов;
- справочник видов процедур и программ лечения;

- картотека пациентов;
- журнал курсов лечения;
- журнал истории болезни, поставленных диагнозов;
- журнал приемов пациента.

Физическое моделирование производилось в СУБД Microsoft Access.

Реализация информационной системы была создана с помощью языка программирования Delphi.

Отдельное внимание в рамках практической части уделено тестированию полученного программного обеспечения, что подтверждает работоспособность системы.

Третья глава работы посвящена оценке и обоснованию экономической эффективности проекта. В результате проведенных расчетов можно сделать вывод о том, что проект успешен и эффективен.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Абдусаламов Р. А. Информационные системы и технологии: учебное пособие / Р. А. Абдусаламов. - Махачкала: Изд-во ДГУ, 2015. - 236 с.
2. Ахметшин Д.А. Проектирование информационных систем / Д.А. Ахметшин, Н.К. Нуриев, С.Д. Старыгина, З.Х. Шакирова. – Казань: Отечество, 2016. – 172 с.
3. Баканов М.В. Базы данных. Системы управления базами данных: учебное пособие / М.В. Баканов, В.В. Романова, Т.П. Крюкова. Кемеровский технологический институт пищевой промышленности. – Кемерово, 2015. – 166 с.
4. Балдин К.В. Информационные системы в экономике / К.В. Балдин, В.Б. Уткин. – М.: Дашков и К, 2015. – 395 с.
5. Бекаревич Ю.Б. Самоучитель Access 2010 / Ю.Б. Бекаревич, Н.В. Пушкина. – СПб.: БХВ-Петербург, 2015. – 432 с.
6. Боченина Н.В. Информационные технологии: Учебное пособие / С.В. Синаторов, О.В. Пикулик, Н.В. Боченина. - М.: Альфа-М: ИНФРА-М, 2016. - 336 с.
7. Боровская Е.В. Программирование в среде Delphi - 3-е изд., (эл.) - М.: БИНОМ. ЛЗ, 2015. - 241 с.
8. Варфоломеева А.О. Информационные системы предприятия / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. – М.: НИЦ ИНФРА-М, 2017. – 283 с.
9. Ветрова О.А. Операционные системы и базы данных: метод. указ. / О. А. Ветрова. - М.: МГУДТ, 2017. - 40 с.
10. Гагарина Л.Г. Информационные технологии: Учебное пособие / Е.Л. Румянцева, В.В. Слюсарь; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2016. - 256 с.
11. Гвоздева В.А. Базы и банки данных / В.А. Гвоздева. - М.: Альтаир-МГАВТ, 2015, - 76 с.

12. Голицына О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: ФОРУМ: ИНФРА-М, 2016. – 352 с.
13. Гофман, В.Э. Работа с базами данных в Delphi: Пособие / Хомоненко А.Д., Гофман В.Э., - 3-е изд., перераб. и доп. - СПб:БХВ-Петербург, 2015. - 628 с.
14. Громов Ю.Ю. Технология программирования / Ю.Ю. Громов, О.Г. Иванова, М.П. Белев, Ю.В. Минин. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2017. – 172 с.
15. Гущин А.Н. Базы данных. 2-е изд., испр. и доп.: учебно-методическое пособие. М.-Берлин: Директ-Медиа, 2016. – 311 с.
16. Дадян, Э.Г. Современные базы данных. Часть 2: практические задания: Учебно-методическое пособие / Дадян Э.Г. - М.: НИЦ ИНФРА-М, 2017. - 68 с
17. Долженко А.И. Управление информационными системами. – Ростов-на-Дону: Изд-во РГУ, 2017. – 191 с.
18. Евгениев Г.Б. Основы автоматизации технологических процессов и производств / Г.Б. Евгениев, С.С. Гаврюшин, А.В. Грошев, М.В. Овсянников, П.С. Шильников. – Москва: Изд-во МГТУУ им. Н.Э. Баумана, 2015. – 441 с.
19. Елинова Г.Г. Информационные технологии в профессиональной деятельности: Краткий курс лекций. Оренбург: ГОУ ОГУ, 2016. – 39 с.
20. Затонский, А.В. Информационные технологии: разработка информационных моделей и систем: Учеб. пос. / А.В.Затонский - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2017 - 344с.
21. Карпузова, В.И. Информационные технологии в менеджменте: Учебное пособие / В.И. Карпузова, Э.Н. Скрипченко, К.В. Чернышева, Н.В. Карпузова. - 2-е изд., доп. - М.: Вузовский учебник: НИЦ ИНФРА-М, 2017.
22. Колдаев, В.Д. Структуры и алгоритмы обработки данных: Учебное пособие / В.Д. Колдаев. - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2016. – 296 с.

23. Коннолли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение: теория и практика. - Москва: Вильямс, 2017. – 1439 с.
24. Коряковский А.В. Информационные системы предприятия: Учебное пособие. - М.: НИЦ ИНФРА-М, 2016. - 283 с.
25. Краснянский М.Н. Проектирование информационных систем управления документооборотом научно-образовательных учреждений / М.Н. Краснянский, С.В. Карпушкин, А.В. Остроух. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 216 с.
26. Кумскова И.А. Базы данных: учебник – 2-е изд., стер. – М.: КНОРУС, 2017. – 488 с.
27. Литвинов В.А. Информационные технологии в юридической деятельности: Учебное пособие. Стандарт третьего поколения. – СПб.: Питер, 2018. – 320 с.
28. Макарова Н.В. Информатика: Учебник для вузов / Н.В. Макарова, В.Б. Волков. – СПб.: Питер, 2016. – 576 с.
29. Мартиросова Т.М. Основы проектирования баз данных. Практикум: учеб. пособие. – Волгоград: ИУНЛ ВолгГТУ, 2016. – 48 с.
30. Михеева Е.В. Информатика / Е.В. Михеева, О.И. Титова. – М.: Издательский центр «Академия», 2015. – 352 с.
31. Мокеев В.В. Бизнес-информатика / В.В. Мокеев, Е.В. Бунова, О.С. Буслаева. – Челябинск: издательский центр ЮУрГУ, 2015. – 67 с.
32. Нестеров С.А. Базы данных: учеб. пособие. – СПб.: Изд-во Политех. ун-та, 2018. – 150 с.
33. Одинцов Б.Е. Информационные системы управления эффективностью бизнеса. - Люберцы: Юрайт, 2015. - 206 с.
34. Рудаков А.В. Технология разработки программных продуктов. – М.: Академия, 2016. – 190 с.
35. Самигуллин М. Ф., Читалин Н. А., Лучкин Г. С. Информационные технологии в медицине: / М. Ф. Самигуллин, Н. А. Читалин, Г. С. Лучкин. - Казань; Шумерля: [б. и.], 2017. - 101 с.

36. Титоренко Г.А. Информационные системы в экономике/ 2-е изд. - М.: ЮНИТИ-ДАНА, 2015. - 463 с.
37. Фомин В. В., Фомина И. К. Системная и программная инженерия. Информационные системы и технологии: учебное пособие / В. В. Фомин, И. К. Фомина. - Санкт-Петербург: Изд-во РГПУ им. А. И. Герцена, 2017. - 139 с.
38. Черников, Б.В. Информационные технологии управления: Учебник / Б.В. Черников. - 2-е изд., перераб. и доп. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2015. - 368 с.
39. Шеховцов О. И. Корпоративные информационные системы: учебное пособие / О. И. Шеховцов. - Санкт-Петербург: СПбГУТ, 2016. - 59 с.
40. John Sharp. Microsoft® Visual C#® 2012 Step by Step, 2012. - 844 p.
41. Lorenz Patrick A. C Sharp Shortcut - Programmieren im .NET-Framework mit der C Sharp Beta 2, 2016. - 296 p.
42. Mike McGrath. C# programming in easy steps, 2016. - 357 p.
43. Sam A Abolrous. Learn C-Sharp - Includes the C-Sharp 3.0 Features, 2017. - 425 p.
44. Svetlin Nakov & Co. Fundamentals of computer programming with C#, 2013. - 1122 p.

Приложение А
Фрагмент исходного кода

```
unit Unit1;

var
  Form1: TForm1;

implementation

uses Unit2, Unit5, Unit8, Unit11, Unit14, Unit17, Unit18, Unit19, Unit21,
  Unit22, Unit23, Unit24, Unit27, Unit28, Unit31, Unit32, Unit33, Unit34,
  UOOCalc,
  Unit35;

{$R *.dfm}

procedure TForm1.N3Click(Sender: TObject);
begin
  with adoquery1 do
    begin
      close;
      sql.Clear;
      sql.Add('select * from clients');
      active:=true;
    end;
  form2.DBGrid1.Columns[0].title.caption:='Код';
  form2.DBGrid1.Columns[1].title.caption:='ФИО';
  form2.DBGrid1.Columns[2].title.caption:='Дата рождения';
  form2.DBGrid1.Columns[3].title.caption:='Серия';
  form2.DBGrid1.Columns[4].title.caption:='Номер';
  form2.DBGrid1.Columns[5].title.caption:='Паспорт выдан';
  form2.DBGrid1.Columns[6].title.caption:='Полис';
  form2.DBGrid1.Columns[7].title.caption:='Социальное положение';
  form2.DBGrid1.Columns[8].title.caption:='Семейное положение';
  form2.DBGrid1.Columns[9].title.caption:='Адрес';
  form2.Show;
end;

procedure TForm1.N5Click(Sender: TObject);
begin
  with adoquery2 do
    begin
```

Продолжение Приложения А

```
close;
sql.Clear;
sql.Add('select * from rooms');
active:=true;
end;
form5.DBGrid1.Columns[0].Title.caption:='Номер палаты';
form5.DBGrid1.Columns[1].Title.caption:='Отделение';
form5.Show;
end;

procedure TForm1.N4Click(Sender: TObject);
begin
  with adoquery3 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from spec');
    active:=true;
  end;
  if label6.Caption='Режим администратора' then form8.rzBitBtn3.Enabled:=true
  else form8.rzBitBtn3.Enabled:=false;
  form8.DBGrid1.Columns[0].Title.caption:='Код';
  form8.DBGrid1.Columns[1].Title.caption:='ФИО';
  form8.DBGrid1.Columns[2].Title.caption:='Специальность';
  form8.DBGrid1.Columns[3].Title.caption:='Отделение';
  form8.Show;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
  with adoquery4 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from proced');
    active:=true;
  end;
  form11.DBGrid1.Columns[0].title.caption:='Код';
  form11.DBGrid1.Columns[1].title.caption:='Наименование';
  form11.DBGrid1.Columns[2].title.caption:='Цена';
  form11.Show;
end;
```

Продолжение Приложения А

```
procedure TForm1.N7Click(Sender: TObject);
begin
  with adoquery5 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from prog_lech');
    active:=true;
  end;
  form14.DBGrid1.Columns[0].Title.caption:='Код';
  form14.DBGrid1.Columns[1].Title.caption:='Наименование';
  form14.Show;
end;
```

```
procedure TForm1.N2Click(Sender: TObject);
begin
  with form1.ADOQuery6 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from const');
    open;
  end;
  with form17 do
  begin
    edit1.Text:=form1.ADOQuery6nam.AsString;
    edit2.Text:=form1.adoquery6fiodir.AsString;
    edit3.Text:=form1.adoquery6adr.AsString;
    edit4.Text:=form1.ADOQuery6inn.AsString;
    edit5.Text:=form1.adoquery6gbuh.AsString;
    edit6.Text:=form1.adoquery6tel.AsString;
    show;
  end;
end;
```

```
procedure TForm1.N9Click(Sender: TObject);
var i,j:integer;
begin
  with form1.adoquery1 do
  begin
    close;
    sql.Clear;
```

Продолжение Приложения А

```
    sql.Add('select * from clients');
    open;
end;
form18.ComboBox1.Clear;
for i:=1 to form1.adoquery1.RecordCount do
begin
form18.ComboBox1.Items.Add(trim(form1.adoquery1 fio.AsString));
form1.adoquery1.Next;
end;
with form1.ADOQuery5 do
begin
    close;
    sql.Clear;
    sql.add('select * from prog_lech');
    open;
end;
form18.ComboBox2.Clear;
for i:=1 to form1.adoquery5.RecordCount do
begin
form18.ComboBox2.Items.Add(trim(form1.adoquery5nam.AsString));
form1.adoquery5.Next;
end;
with form1.adoquery7 do
begin
    close;
    sql.Clear;
    sql.Add('select * from putevki');
    open;
end;
j:=0;
for i:=1 to form1.adoquery7.RecordCount do
begin
    if          j<form1.adoquery7code.AsInteger          then
j:=form1.adoquery7code.AsInteger;
    form1.adoquery7.Next;
end;
form18.Edit1.Text:=inttostr(j+1);
with form1.adoquery2 do
begin
    close;
    sql.Clear;
    sql.Add('select * from rooms');
```

open;

Продолжение Приложения А

```
end;  
form18.ComboBox3.Clear;  
for i:=1 to form1.adoquery2.RecordCount do  
begin  
    form18.ComboBox3.Items.Add(trim(form1.ADOQuery2code.AsString));  
    form1.adoquery2.Next;  
end;  
form18.Label9.Caption:='0';  
form18.Edit3.Text:='0';  
form18.DateTimePicker1.Date:=now;  
form18.DateTimePicker2.Date:=now;  
form18.Show;  
end;
```

```
procedure TForm1.N10Click(Sender: TObject);  
begin  
    with form1.adoquery8 do  
    begin  
        close;  
        sql.Clear;  
        sql.Add('select * from vid_proc');  
        active:=true;  
    end;  
    form19.DBGrid1.Columns[0].title.caption:='Код';  
    form19.DBGrid1.Columns[1].title.caption:='Наименование';  
    form19.Show;  
end;
```

```
procedure TForm1.N11Click(Sender: TObject);  
begin  
    with adoquery7 do  
    begin  
        close;  
        sql.Clear;  
        sql.Add('select * from putevki');  
        active:=true;  
    end;  
    form21.show;  
end;
```

```
procedure TForm1.N12Click(Sender: TObject);
```

```
var i:integer;
```

Продолжение Приложения А

```
begin
  with form1.adoquery8 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from vid_proc');
    open;
  end;
  form22.ComboBox1.clear;
  for i:=1 to form1.adoquery8.RecordCount do
  begin
    form22.ComboBox1.Items.Add(form1.adoquery8nam.AsString);
    form1.adoquery8.Next;
  end;
  with form1.adoquery1 do
  begin
    close;
    sql.Clear;
    sql.add('select * from clients');
    open;
  end;
  form22.combobox2.clear;
  for i:=1 to form1.ADOQuery1.RecordCount do
  begin
    form22.ComboBox2.Items.add(trim(form1.ADOQuery1 fio.AsString));
    form1.adoquery1.next;
  end;
  with adoquery3 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from spec');
    active:=true;
  end;
  form22.Show;
end;

procedure TForm1.N13Click(Sender: TObject);
var i:integer;
begin
  with form1.adoquery10 do
```

```
begin
```

Продолжение Приложения А

```
    close;
    sql.Clear;
    sql.Add('select * from ist_bol');
    open;
end;
with form1.adoquery1 do
begin
    close;
    sql.Clear;
    sql.Add('select * from clients');
    open;
end;
form23.ComboBox1.Clear;
for i:=1 to form1.adoquery1.recordcount do
begin
    form23.ComboBox1.Items.Add(trim(form1.ADOQuery1 fio.AsString));
    form1.adoquery1.Next;
end;
form23.Show;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
n1.Enabled:=false;
n8.Enabled:=false;
n19.Enabled:=false;
label2.Visible:=false;
label3.Visible:=false;
label4.Visible:=false;
label5.Visible:=false;
label6.Visible:=false;
label6.Caption:="";
rzb1btn1.Enabled:=false;
rzb1btn2.Enabled:=false;
rzb1btn3.Enabled:=false;
rzb1btn4.Enabled:=false;
rzb1btn5.Enabled:=false;
rzb1btn6.Enabled:=false;
rzb1btn7.Enabled:=false;
rzb1btn8.Enabled:=false;
end;
```


Продолжение Приложения А

```
procedure TForm1.N14Click(Sender: TObject);
var i:integer;
begin
  with adoquery3 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from spec');
    open;
  end;
  form24.ComboBox1.Clear;
  for i:=1 to adoquery3.RecordCount do
  begin
    form24.ComboBox1.Items.Add(trim(form1.adoquery3login.asstring));
    form1.ADOQuery3.Next;
  end;
  form24.show;
end;

procedure TForm1.N16Click(Sender: TObject);
begin
  form1.Close;
end;

procedure TForm1.N15Click(Sender: TObject);
var i:integer;
begin
  with form1.adoquery1 do
  begin
    close;
    sql.Clear;
    sql.Add('select * from clients');
    open;
  end;
  form27.ComboBox1.Clear;
  for i:=1 to form1.adoquery1.recordcount do
  begin
    form27.ComboBox1.Items.Add(trim(form1.ADOQuery1.fio.AsString));
    form1.adoquery1.Next;
  end;
  form27.Show;
```

end;

Продолжение Приложения А

```
procedure TForm1.N17Click(Sender: TObject);
```

```
begin
```

```
  with form1.ADOQuery11 do
```

```
  begin
```

```
    close;
```

```
    sql.Clear;
```

```
    sql.add('select * from diag');
```

```
    active:=true;
```

```
  end;
```

```
  form28.DBGrid1.Columns[0].Title.caption:='Код';
```

```
  form28.DBGrid1.Columns[1].Title.caption:='Наименование';
```

```
  form28.Show;
```

```
end;
```

```
procedure TForm1.N18Click(Sender: TObject);
```

```
var i:integer;
```

```
begin
```

```
  with adoquery1 do
```

```
  begin
```

```
    close;
```

```
    sql.Clear;
```

```
    sql.Add('select * from clients');
```

```
    open;
```

```
  end;
```

```
  form31.ComboBox1.Clear;
```

```
  for i:=1 to adoquery1.RecordCount do
```

```
  begin
```

```
    form31.ComboBox1.Items.add(trim(form1.ADOQuery1 fio.AsString));
```

```
    adoquery1.Next;
```

```
  end;
```

```
  form31.ComboBox2.clear;
```

```
  with adoquery3 do
```

```
  begin
```

```
    close;
```

```
    sql.Clear;
```

```
    sql.Add('select * from spec');
```

```
    open;
```

```
  end;
```

```
  for i:=1 to form1.ADOQuery3.RecordCount do
```

```
  begin
```

```
    form31.ComboBox2.Items.add(trim(form1.ADOQuery3 fio.AsString));
```

```
adoquery3.Next;
```

Продолжение Приложения А

```
end;  
form31.CheckListBox1.clear;  
form31.label5.Caption:="";  
form31.label7.Caption:="";  
form31.DateTimePicker1.date:=now;  
form31.show;  
end;  
  
procedure TForm1.N20Click(Sender: TObject);  
var i:integer;  
begin  
    with adoquery3 do  
        begin  
            close;  
            sql.Clear;  
            sql.Add('select * from spec');  
            open;  
        end;  
        form32.ComboBox1.Clear;  
        for i:=1 to form1.ADOQuery3.RecordCount do  
            begin  
                form32.ComboBox1.Items.add(trim(form1.ADOQuery3fio.AsString));  
                adoquery3.Next;  
            end;  
        form32.Show;  
    end;  
  
procedure TForm1.RzBitBtn1Click(Sender: TObject);  
begin  
    n5.Click;  
end;  
  
procedure TForm1.RzBitBtn3Click(Sender: TObject);  
begin  
    n3.click;  
end;  
  
procedure TForm1.RzBitBtn4Click(Sender: TObject);  
begin  
    n4.click;  
end;
```

Продолжение Приложения А

```
procedure TForm1.RzBitBtn5Click(Sender: TObject);
begin
    n7.Click;
end;
```

```
procedure TForm1.RzBitBtn6Click(Sender: TObject);
begin
    n13.click;
end;
```

```
procedure TForm1.RzBitBtn7Click(Sender: TObject);
begin
    n9.Click;
end;
```

```
procedure TForm1.RzBitBtn8Click(Sender: TObject);
begin
    n15.click;
end;
```

```
procedure TForm1.RzBitBtn2Click(Sender: TObject);
begin
    n2.click;
end;
```

```
procedure TForm1.N21Click(Sender: TObject);
var i:integer;
begin
    form33.ComboBox1.clear;
    with form1.adoquery5 do
    begin
        close;
        sql.Clear;
        sql.Add('select * from prog_lech');
        open;
    end;
    for i:=1 to form1.ADOQuery5.RecordCount do
    begin
        form33.ComboBox1.Items.Add(trim(form1.ADOQuery5nam.AsString));
        form1.ADOQuery5.Next;
    end;
```

```
form33.Show;
```

Продолжение Приложения А

```
end;
```

```
procedure TForm1.N22Click(Sender: TObject);  
var i:integer;  
begin  
  with adoquery2 do  
    begin  
      close;  
      sql.Clear;  
      sql.Add('select * from rooms');  
      open;  
    end;  
    form34.ComboBox1.clear;  
    for i:=1 to form1.ADOQuery2.RecordCount do  
      begin  
        form34.ComboBox1.Items.Add(form1.ADOQuery2code.AsString);  
        form1.ADOQuery2.Next;  
      end;  
    form34.Show;  
  end;
```

```
procedure TForm1.N23Click(Sender: TObject);  
var calc:toocalc; i:integer;  
begin  
  calc:=toocalc.Create('c:\polikl\slt\sv_pal.slt');  
  with form1.ADOQuery2 do  
    begin  
      close;  
      sql.Clear;  
      sql.add('select * from rooms');  
      open;  
    end;  
    for i:=1 to form1.ADOQuery2.RecordCount do  
      begin  
        calc.cell[4,1]:=form1.ADOQuery2code.AsString;  
        with form1.ADOQuery7 do  
          begin  
            close;  
            sql.Clear;  
            sql.add('select * from putevki where  
(cdroom='+form1.ADOQuery2code.AsString+') and (act=true)');
```

open;

Продолжение Приложения А

```
end;  
calc.Cell[4,2]:=5-form1.ADOQuery7.RecordCount;  
if i<form1.ADOQuery2.RecordCount then calc.InsertRow(4,true);  
form1.ADOQuery2.Next;  
end;  
  
calc.Visible:=true;  
end;  
  
procedure TForm1.N24Click(Sender: TObject);  
begin  
    form35.show;  
end;  
  
end.
```