

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
(наименование профиля, специализации)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: « Разработка CRM системы для медицинского учреждения (на примере офтальмологической клиники «Кругозор»)»»

Студент

П. В. Борисов

(И.О. Фамилия)

(личная подпись)

Руководитель

В. Ф. Глазова

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

Тольятти 2021

Аннотация

ИНФОРМАЦИОННАЯ СИСТЕМА, УЧЕТ ПАЦИЕНТОВ КЛИНИКИ, БАЗА ДАННЫХ, SADT, IDEF0, ЛОКАЛЬНАЯ КОМПЬЮТЕРНАЯ СЕТЬ.

В данной выпускной квалификационной работе рассматривается задача разработки CRM системы для клиники на примере офтальмологической клиники «Кругозор».

В работе разобрана структура клиники «Кругозор», определены роли и основных подразделений и их участие в процессах офтальмологической клиники «Кругозор», выполнен технико-экономический анализ, выявлены основные проблемы.

Построены модели «КАК ЕСТЬ» с использованием методологии IDEF0. Сформулированы цель и задачи проектирования и требования к информационной системе. Произведен выбор и обоснование методов проектирования информационной системы, а также проектных решений по СУБД и ЛВС.

Построена функциональная модель компании «КАК ДОЛЖНО БЫТЬ», описано программное, компьютерное, сетевое и технологическое обеспечение автоматизированной информационной системы.

Разработана CRM система для клиники и описаны основные функциональные особенности данной системы.

Оценена экономическая эффективность реализации проекта. Работа находится на стадии внедрения (принято решение о внедрении, рассматривается вопрос о внедрении).

Работа представлена на 69 листах, рисунков – 25, таблиц – 18, литературных источников – 23, приложений – 2.

Оглавление

Введение.....	4
Глава 1 Анализ существующих CRM систем медицинских учреждений и разработка требований к создаваемой системе.....	6
1.1 Определение CRM системы и ее элементов	6
1.2 Техничко–экономическая характеристика существующих систем.....	10
1.3 Организационная структура и характеристика офтальмологической клиники «Кругозор»	14
1.4 Концептуальное моделирование предметной области	15
1.5 Постановка задачи для разработки автоматизированной информационной системы	21
Глава 2 Разработка автоматизированной информационной системы	22
2.1 Логическое моделирование предметной области.....	22
2.2 Информационное обеспечение системы	29
2.3 Выбор архитектуры и способов создания базы данных для разрабатываемой автоматизированной информационной системы	31
2.4 Физическое моделирование БД	31
2.5 Физическое моделирование разрабатываемой CRM системы для медицинского учреждения.....	39
2.6 Описание разработанной автоматизированной информационной CRM системы для медицинского учреждения	43
Глава 3 Расчет экономической эффективности разработанной CRM системы для медицинского учреждения	47
3.1 Выбор методики расчета экономической эффективности	47
3.2 Расчет показателей экономической эффективности, разработанной CRM системы для медицинского учреждения	51
Заключение	57
Список используемой литературы и используемых источников.....	59
Приложение А Программный код.....	61
Приложение Б Декомпозиция процесса администрирования приёма пациентов	69

Введение

Ежедневно в медицинских учреждениях решается множество серьезных проблем, связанных с регистрацией, обработкой и хранением медицинской информации, и управлением информационными потоками. Для упрощения работы по организации и систематизации работы сотрудников с данной информацией существуют Автоматизированные Информационные Системы (далее АИС).

Объектом исследований в рамках данной работы является бизнес-процессы в офтальмологической клинике "Кругозор" (далее Компания).

Предметом исследования является автоматизация процессов лечения пациентов.

Цели работы:

- анализ бизнес-процессов компании;
- создание логической концептуальной модели АИС;
- создание физической модели АИС;
- разработка приложения АИС и базы данных (далее БД).

Для достижения поставленных целей в работе необходимо решить следующие задачи:

- описать организационную структуру и основные бизнес-процессы офтальмологической клиники «Кругозор»;
- разработать модели бизнес-процессов клиники с определением задач, которые должны быть автоматизированы в рамках реинжиниринга бизнес-процессов;
- разработать требования к АИС;
- разработать логических моделей АИС с помощью нотации UML;
- разработать АИС;
- разработать информационного программного обеспечения (базы данных) АИС.

Последние три задачи должны быть практически решены с помощью соответствующих программных средств (систем CASE).

Бакалаврская работа состоит из введения, трех глав, заключения, списка используемых источников и приложений.

В первой главе работы приведено описание предмета исследования, рассмотрена организационная структура клиники «Кругозор», ее основные бизнес-процессы, идентифицированы роли основных подразделений, проведены сравнительный анализ наиболее распространенных АИС, существующих на рынке, и реинжиниринг бизнес-процессов, выполнена постановка задачи на разработку АИС.

Вторая глава посвящена логическому и физическому проектированию разрабатываемой АИС. В ней содержится описание информационного обеспечения компании, обоснован выбор архитектуры, методов и средств для разработки АИС, а также рассмотрены модули и функциональность разработанного приложения.

Третья глава дает оценку затрат, амортизации и экономической эффективности внедрения разработанной системы.

Результаты и выводы о проделанной работе кратко представлены в заключении.

Результатом данной работы является разработка АИС компании, применение которой значительно повышает качество и эффективность производственных процессов офтальмологической клиники «Кругозор».

Глава 1 Анализ существующих CRM систем медицинских учреждений и разработка требований к создаваемой системе

1.1 Определение CRM системы и ее элементов

«Управление взаимоотношениями с клиентами (CRM) – это сочетание практик, стратегий и технологий, которые компании используют для управления и анализа взаимодействий с клиентами. Цель применения CRM систем состоит в том, чтобы улучшить отношения клиентов со службой поддержки, а также помочь менеджерам компании в удержании клиентов и стимулировании роста продаж. Системы CRM собирают данные о клиентах по различным каналам связи между клиентом и компанией, которые могут включать веб-сайт компании, телефон, чат, прямую почтовую рассылку, маркетинговые материалы и социальные сети. Системы CRM могут также дать персонализированные и подробные сведения на клиента сотрудникам фирмы» [6].

«CRM включает в себя сочетание бизнес-стратегий, программного обеспечения и процессов, которые помогают выстраивать долгосрочные отношения между компаниями и их клиентами. Поэтому с помощью технологии CRM можно управлять стратегиями, методами, инструментами, которые использует предприятие для развития и удержания своих клиентов» [7].

CRM хранит подробную информацию об общей истории покупок, личной информации и даже моделях покупательского поведения, поскольку представляет возможность собирать данные о клиентах по нескольким каналам [3].

Развертывание программного обеспечения CRM выгодно для компаний любого размера. Простая в использовании CRM позволяет организациям малого и среднего бизнеса легко управлять всем циклом обслуживания клиента, не перегружая сотрудников ненужными сложными функциями.

Более того, внедрив решение с высокой степенью масштабируемости, малые предприятия смогут расширять возможности CRM вместе с ростом компании.

Следует отметить, что самое главное достоинство в CRM системе то, что она может принести пользу практически любому организационному подразделению – от продаж и обслуживания клиентов до набора персонала, маркетинга и развития бизнеса. Хорошее программное обеспечение CRM позволяет лучше управлять внешними отношениями.

Возможности CRM систем:

- хранение всей информации о клиентах в одном месте;
- регистрация проблем с обслуживанием;
- определение возможностей продаж;
- управление маркетинговыми кампаниями;
- планирование мероприятий;
- управление персоналом компании.

«Основная цель программного обеспечения CRM – сделать взаимодействие более эффективным и продуктивным. Автоматизированные процедуры в модуле CRM включают отправку маркетинговых материалов отделу продаж на основе выбора продукта или услуги клиентом. Программы также оценивают потребности клиента, чтобы сократить время, необходимое для выполнения запроса» [8].

CRM обеспечивает следующие технологии:

- автоматизация маркетинга;
- автоматизация продаж;
- автоматизация работы контакт–центра;
- технология геолокации или услуги на основе определения местоположения;
- автоматизация рабочего процесса;
- ведущий менеджмент;
- управление человеческими ресурсами;

- аналитика;
- искусственный интеллект;
- управление проектом;
- интеграция с другим ПО.

С помощью CRM можно добиться следующего:

1. Улучшить отношения с клиентами – CRM фиксирует каждое взаимодействие на пути клиента (перспектива, ведущая к клиенту) от нескольких команд, чтобы создать 360–градусное представление о клиенте.

2. Уменьшить ввод данных – CRM уменьшает рутинные задачи ввода данных, автоматически добавляя и обновляя записи о клиентах в базе данных.

3. Улучшить общение – CRM собирает всю информацию о клиентах в одном месте.

4. Визуализировать конвейеры – CRM помогает визуализировать потоки продаж, чтобы компания могла расставить приоритеты, над какими сделками работать. Визуализируя конвейер, менеджеры могут создавать контрольные показатели процента конверсии и сразу видеть, когда сделки не проходят.

5. Уменьшить отток – использование данных из CRM, например, требования, предпочтения, покупательские привычки и т.д.

6. Увеличить доход – анализ CRM поможет вам определить самых прибыльных клиентов и построить с ними более крепкие отношения.

7. Улучшить сотрудничество – без CRM бизнес–функции часто работают изолированно с отдельными частями данных о клиентах.

CRM системы бывают трех основных типов:

1. Операционная CRM является наиболее распространенным типом CRM и используется в различных отраслях промышленности. Этот тип CRM отлично подходит для управления коммуникациями с клиентами и помогает предприятиям управлять своими повседневными продажами, маркетингом и операциями по обслуживанию клиентов.

2. Аналитическая CRM собирает, хранит и анализирует информацию о своих клиентах. Обычно это данные о клиентах, маркетинговые данные, данные о продажах и данные об услугах.

3. Совместная CRM – позволяет организации собирать, организовывать и обмениваться информацией с клиентами по нескольким командам.

CRM системы имеют 2 основных типа решения – это облачные и локальные. Рассмотрим их более подробно.

«Облачные системы предоставляют данные в реальном времени торговым агентам в офисе и на местах, пока компьютер, смартфон, ноутбук или планшет подключен к Интернету. Такие системы обладают повышенным доступом к информации о клиентах и исключают иногда сложный процесс установки, связанный с другими продуктами или программным обеспечением CRM» [8]. Компании тщательно продумывают способ развертывания CRM системы, чтобы убедиться, что она соответствует их бизнес-целям. Хотя некоторые организации могут извлечь выгоду из локального программного обеспечения, развертывание облачной CRM дает множество преимуществ.

При использовании локальной CRM компании необходимо заранее приобрести лицензии. Учитывая, что программное обеспечение размещено на серверах компании, локальная CRM является хорошим выбором для организаций со строгой политикой безопасности, но в таком случае процесс внедрения требует больше времени и ИТ-ресурсов.

По результатам сравнительного анализа существующих типов CRM, было принято решение остановиться на разработке локальной версии CRM, поскольку в данном случае важно то, что локальное программное обеспечение CRM обладает высокой масштабируемостью на серверах компании, также данная версия обладает большей безопасностью хранимых данных, а это означает, что организация может легко расширять свою функциональность по мере роста бизнес-требований.

1.2 Техничко–экономическая характеристика существующих систем

Сегодня в пятерку крупнейших игроков на рынке CRM входят гигант облачных вычислений Salesforce, Microsoft, Oracle, Adobe Systems и 1С, но это ведущие разработчики универсальных систем, нас же интересует конкретно системы для медицинских клиник. Рассмотрим более подробно данные системы.

CRM для здравоохранения выходит за рамки управления пациентами и контактами. Также большое внимание уделяется маркетингу и продажам. При этом под продажами имеется ввиду помощь клиникам или другим медицинским организациям в поиске новых пациентов и расширении их практики.

В современных клиниках автоматизация стала привычным явлением: онлайн–расписание, электронные медицинские карты, бухгалтерский учет. Тем временем взаимоотношения с пациентами до сих пор остаются без внимания. Клиника с помощью CRM имеет возможность ведения базы пациентов, отслеживания всех этапов их взаимодействия с медицинским центром, а также может оставлять метки и напоминания для регистраторов.

Подобная CRM может быть задействована даже в таких вещах, как поддержка клиентов. Создание онлайн–ресурсных центров, формирование заявок и отслеживание проблем, предоставление единой панели управления для всех сообщений от потенциальных и повторных пациентов, а также решение многих других проблем возможно с применением CRM.

CRM для клиники должны включать в себя:

- управление отношениями с пациентом;
- управление медицинскими данными;
- вовлечение пациентов;
- прием новых пациентов;
- реферальное управление;
- управление оплатой;

- возможность сбыта препаратов;
- инструменты записи на прием;
- управление репутацией клиники.

Рассмотрим более подробно 5 самых распространённых CRM для медицинских клиник в России.

1. Medesk МИС – это система, совмещающая технологии и медицину, для прозрачного и простого управления частной клиникой. Есть онлайн–запись, телефония, складской учет и пр. Возможно масштабирование до 1000 клиник и индивидуальная настройка под бизнес–процессы.

2. MedicalCRM – облачная система для управления клиниками и центрами. Отличается простым и одновременно удобным интерфейсом с большим количеством функций. Бесплатная настройка и обучение. Отчеты в любых разрезах, выгрузка данных без ограничения. Гибкие права доступа. Виджет онлайн записи для сайта.

3. Клиника онлайн – простая система для управления клиникой экономит время доктора и администратора, повышает управляемость предприятием. Ведется прозрачный учет финансов, материалов, зарплаты. Можно формировать программы лояльности, стоимость зависит от модулей.

4. MEDIDEA – современная программа для клиник, медицинских центров и косметологических клиник с возможностью работы в системе ОМС, АТС, онлайн запись, электронная история болезни, учет и списание медикаментов, финансовая и экономическая аналитика клиники, расчет заработной платы.

5. ArchiMed+ – это современная медицинская информационная система для автоматизации всех процессов работы частных и сетевых медицинских и стоматологических клиник.

На основании этих данных создадим таблицы с характеристиками по следующим параметрам: используемые платформы для CRM, функционал, варианты развертывания и цену использования.

В таблице 1 приведены основные существующие платформы и возможность установки CRM на эти платформы.

Таблица 1 – Платформы, на которых можно ставить CRM системы

	Клиника онлайн	MedicalCRM	Medesk	ArchiMed+	MEDIDEA
Веб-приложение	+	+	+	+	-
Windows	-	-	+	+	+
Mac	-	-	+	+	-
Linux	-	-	+	+	-
Android	-	-	-	-	-
iOS	-	-	-	-	-

Далее составим таблицу в которой уточним возможный для использования функционал CRM систем (таблица 2).

Таблица 2 – Функционал, который можно использовать в CRM системах

	Клиника онлайн	Medical CRM	Medesk	ArchiMed+	MEDIDEA
Воронка продаж	-	+	-	+	-
База клиентов	+	+	+	+	+
Колл-центр и телефония	-	+	+	+	+
История взаимодействия с клиентом	+	+	+	+	+
Мониторинг эффективности персонала	-	+	+	+	+
Открытый исходный код	-	-	-	-	-
Отчёты	+	+	+	+	+

Анализ показывает, что такие CRM как Medesk и ArchiMed+ являются лидерами по платформам, на которых можно их разместить, поскольку они обладают как облачной версией программы, так и локальной.

Можно сделать вывод, что CRM Medesk и ArchiMed+ также являются лидерами по функционалу, который можно обеспечить для использования как докторам клиник с административным составом, так и клиентам, которые могут воспользоваться услугами клиники. Основные функциональные

достоинства данных CRM будут учтены при разработке новой автоматизированной информационной системы.

Создадим таблицу, в которой укажем возможные варианты развертывания CRM систем (таблица 3).

Таблица 3 – Варианты развертывания CRM систем

	Клиника онлайн	MedicalCRM	Medesk	ArchiMed+	MEDIDEA
Облако	+	+	+	+	-
Сервер	-	-	-	+	+
ПК	-	-	-	+	+

Рассмотрим цену использования данных CRM систем и внесем её в отдельную таблицу (таблица 4).

Таблица 4 – Тарифы на развертывание CRM систем

	Клиника онлайн	MedicalCRM	Medesk	ArchiMed+	MEDIDEA
Тарифы	От 1490 рублей в месяц.	От 1500 рублей в месяц.	От 3500 рублей в месяц.	От 3500 рублей в месяц.	1 месяц бесплатно, далее - 50000 рублей за лицензию.

Главными недостатками данных CRM систем является то, что они являются платными и имеют закрытый исходный код.

Далее рассмотрим организационную структуру и характеристику оказываемых услуг медицинского учреждения на примере офтальмологической клиники «Кругозор».

1.3 Организационная структура и характеристика офтальмологической клиники «Кругозор»

Академическая офтальмологическая клиника «Кругозор» в Москве была создана как клиническая база «Академии медицинской оптики и оптометрии».

Основным направлением деятельности клиники является научный подход в коррекции рефракционных нарушений зрения путём применения самых современных методов оптической коррекции зрения [9].

На рисунке 1 показана организационная структура клиники.



Рисунок 1 – Организационная структура офтальмологической клиники «Кругозор»

Офтальмологическая клиника «Кругозор» – современная офтальмологическая клиника, предлагающая высококвалифицированную помощь в лечении и диагностике рефракционных нарушений, а также:

- новейшие диагностические приборы, уникальные методы оптической коррекции, эффективные методы лечения осложнённой близорукости, гиперметропии, астигматизма, а также других

заболеваний глаза: глаукомы, катаракты, патология зрительного нерва и сетчатки и других;

- полное обследование зрения и постановку диагноза в день обращения;
- специализированное отделение лечения амблиопии, косоглазия, астигматизма и других осложнений рефракционных нарушений, как у взрослых, так и у детей;
- политику «доступных цен» для широкого круга пациентов;
- консультации от лучших врачей–офтальмологов Москвы и России;
- удобное время приема для каждого клиента.

После того как составили организационную структурную схему клиники и охарактеризовали оказываемые ею медицинские услуги, перейдем к созданию концептуальной модели.

1.4 Концептуальное моделирование предметной области

«Концептуальная модель – это определённое множество понятий и связей между ними, являющихся смысловой структурой рассматриваемой предметной области» [10].

«Концептуальная модель – модель предметной области, состоящей из перечня взаимосвязанных понятий, используемых для описания этой области, вместе со свойствами и характеристиками, классификацией этих понятий, по типам, ситуациям, признакам в данной области и законов протекания процессов в ней» [10].

«Концептуальная модель – это минимальный набор абстрактных понятий, описывающих отдельную область применения, класс задач или систем. Концептуальная модель служит основой для проектирования базы данных и разработки системы. При этом каждый новый объект этой базы данных должен быть классифицирован в соответствии с понятиями этой модели. Это же справедливо и в отношении каждой операции» [10].

В данной модели мы укажем минимальный набор абстрактных понятий, который показывает суть процесса работы клиники. Создадим модели «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ» а также опишем эти модели. Моделирование бизнес-процессов будет выполнено с использованием технологии IDEF0.

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ) [17].

На рисунке 2 представлена контекстная диаграмма бизнес-процесса



Рисунок 2 – Диаграмма IDEF0 «КАК ЕСТЬ» (A0) бизнес–процесса «Организация обслуживания пациентов офтальмологической клиники»

На диаграмме изображены следующие элементы:

1. Выходные данные: отчеты и акты выполненных работ.

2. Входные данные: обращение клиента.
3. Управляющие воздействия: законодательство РФ, нормативные акты клиники «Кругозор».
4. Исполнители: сотрудники клиники «Кругозор».

Для проведения анализа существующего процесса, следует создать поэтапную декомпозицию процесса до требуемого уровня детализации, в которой указать существующие отношения между пациентом клиники и предоставляемыми ему услугами.

Данная поэтапная композиция отображена на рисунке 3.

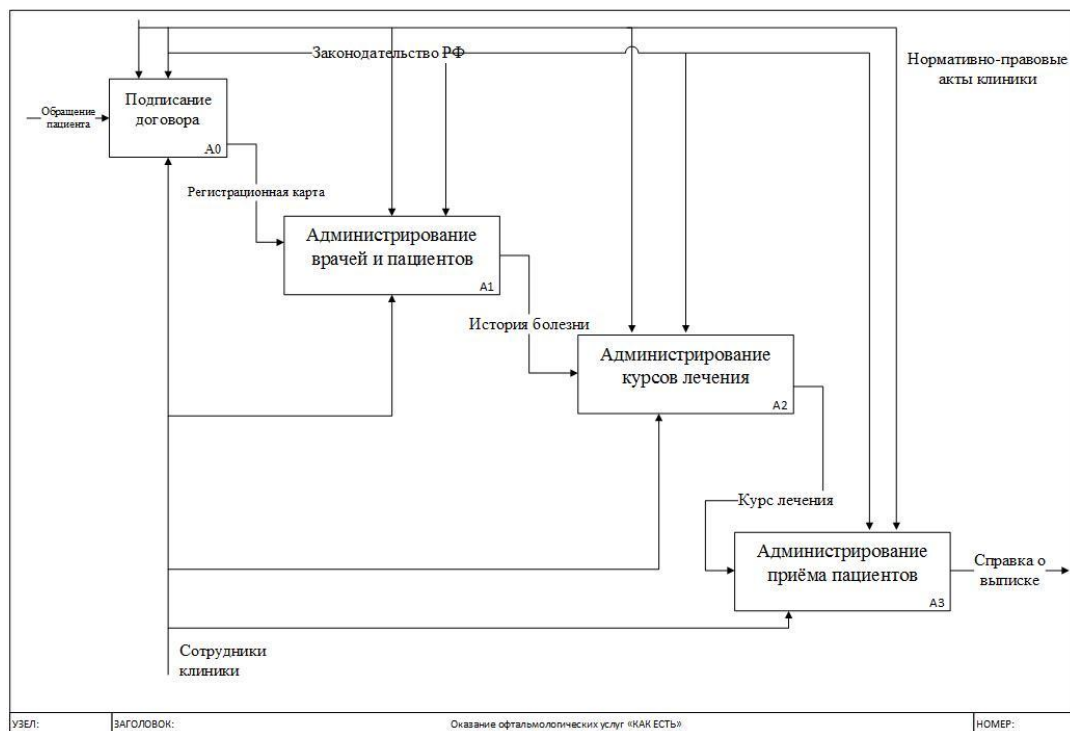


Рисунок 3 – Диаграмма IDEF0 «КАК ЕСТЬ» декомпозиции бизнес-процесса «Оказание офтальмологических услуг»

Данная модель является основой для анализа и дальнейшего совершенствования процесса.

A1: подписание договора. Сотрудники выясняют у клиента предмет и условия договора.

А2: администрирование врачей и пациентов. После подписания договора для клиента создаются регистрационная карта, история болезни и назначается доктор.

А3: администрирование курсов лечения. Назначенный доктор проводит осмотр клиента, даёт направление на сдачу анализов, на основании которых ставится диагноз и назначается курс лечения.

А4: администрирование приема пациентов. После проведенного курса лечения доктор готовит выписку для клиента и закрывает историю болезни.

Проанализировав модели бизнес–процессов оказания медицинских услуг на примере офтальмологической клиники «Кругозор», можно сделать вывод о необходимости автоматизированного учета клиентов, записи и последующего оказания медицинских услуг.

Рассмотрев модель «КАК ЕСТЬ», мы выявили основные недостатки процесса оказания услуг клиники:

- неэффективный документооборот, вынуждающий каждого сотрудника клиники вести списки пациентов и предоставляемых для них услуг;
- большое количество изменений в ходе оказания услуг, требующих постоянной фиксации, препятствуют оперативности обработки данных и их актуальности;
- неэффективное использование времени сотрудниками клиники;
- низкая эффективность использования медицинских препаратов;
- ошибки в постановке диагноза;
- в процессе исправления найденных ошибок могут возникнуть другие ошибки, которые связаны между собой и система может дать сбой в работе;
- есть вероятность потери данных.

Теперь необходимо построить диаграммы «КАК ДОЛЖНО БЫТЬ», в данном случае добавился дополнительный исполнитель – автоматизированная информационная система (АИС).

На рисунке 4 указана диаграмма IDEF0 верхнего уровня «КАК ДОЛЖНО БЫТЬ».



Рисунок 4 – Диаграмма IDEF0 верхнего уровня «КАК ДОЛЖНО БЫТЬ» бизнес–процесса «Оказание офтальмологических услуг»

В данном случае при выполнении любого процесса в клинике на него будет оказываться воздействие АИС, которая позволит упростить процесс работы квалифицированного персонала.

Отметим на схеме процесса лечения планируемые новшества, которые сделают более эффективной работу сотрудников клиники.

Данная поэтапная композиция процесса лечения пациентов «КАК ДОЛЖНО БЫТЬ» отображена на рисунке 6.

На рисунке 5 показана декомпозиция диаграммы «КАК ДОЛЖНО БЫТЬ» процесса «Оказание офтальмологических услуг».

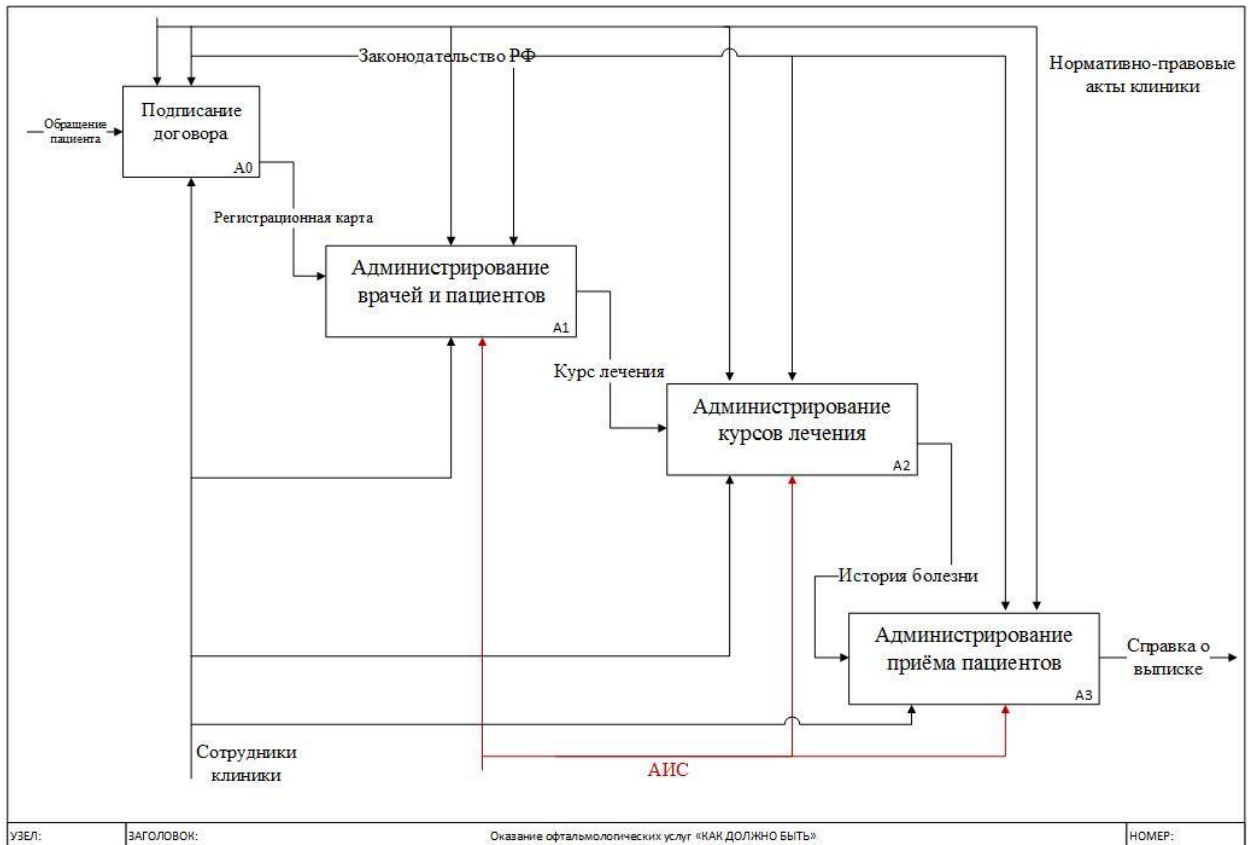


Рисунок 5 – Диаграмма IDEF0 «КАК ДОЛЖНО БЫТЬ» декомпозиции бизнес-процесса «Оказание офтальмологических услуг»

Для пункта администрирование приёма пациентов сделаем декомпозицию (ПРИЛОЖЕНИЕ Б).

С помощью декомпозиции процесса оказания офтальмологических услуг мы отметили влияние внедрения АИС в медицинское учреждение. Внедрение АИС в данном случае облегчит работу персонала клиники, а также уменьшит время на обработку информации и нормативных актов.

1.5 Постановка задачи для разработки автоматизированной информационной системы

Благодаря проведенному анализу бизнес-процессов клиники и существующих решений на рынке CRM систем для клиники были определены следующие задачи на разработку:

1. Разработать электронную базу данных компании, с помощью которой будет вестись учет клиентов и оказываемых им услуг.

2. Разработать приложение, с помощью которого будет проводиться работа с базой данных клиники для оказания офтальмологических услуг, учитывая проведенный ранее анализ CRM систем [18].

3. В разработанном приложении реализовать отдельные модули, позволяющие упростить обслуживание клиентов клиники и работы персонала.

После автоматизации процессов офтальмологической клиники «Кругозор» планируется достичь следующих результатов:

- уменьшение времени обработки медицинской документации;
- автоматизация рабочего процесса работников клиники;
- долговременное хранение данных о сотрудниках и клиентах клиники.

Выводы по главе 1

В данной главе было дано определение CRM системы, также проведен сравнительный анализ самых распространённых систем для клиники, основные достоинства которых планируется использовать в разрабатываемой системе. Построена организационная структура офтальмологической клиники «Кругозор». Были построены модели «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ» при использовании методологии IDEF0.

Глава 2 Разработка автоматизированной информационной системы

2.1 Логическое моделирование предметной области

Методы моделирования можно разделить на логические и физические. Использование комбинации этих методов позволяет создать наиболее полную модель.

Модель процесса логического уровня используется для моделирования подробных требований прикладной системы. В ходе анализа концептуальная модель процесса превращается в подробные функциональные требования или логическую модель процесса. Моделирование процесса на логическом уровне создает подробную модель процесса, используемую для описания функциональных требований прикладной системы, с помощью которой получаем физический проект [19].

Логическое моделирование процесса – это представление бизнес-процесса, детализирующее все действия в процессе от сбора исходных данных до достижения желаемого результата.

Логическое моделирование процессов включает в себя:

1. Описание функциональных требований
2. Определение всех процессов, включенных в систему.

Процесс – это скоординированный набор действий, направленных на достижение определенного результата.

Бизнес-процесс – это тип процесса, предназначенный для достижения определенной бизнес-цели. Бизнес-процессы представляют собой поток данных через серию задач, которые предназначены для достижения определенных бизнес-результатов.

Бизнес-процессы состоят из множества компонентов, в том числе:

- данные, необходимые для достижения желаемой бизнес-цели;
- индивидуальные рабочие задачи, которые манипулируют,

- просматривают или каким-либо образом воздействуют на данные;
- решения, влияющие на данные в процессе или способ его проведения;
- перемещение данных между задачами в процессе;
- отдельные лица и группы, выполняющие задачи.

Виды деятельности, описанные в логической модели процесса:

- сбор данных;
- контроль доступа к данным во время выполнения процесса;
- определение того, какая рабочая задача в процессе должна быть выполнена следующей;
- доставка соответствующего подмножества данных в соответствующую рабочую задачу;
- обеспечение наличия всех необходимых данных и выполнения всех необходимых действий для каждой задачи;
- обеспечение механизма, указывающего на принятие результатов процесса.

Методы моделирования логических процессов обеспечивают описание логического потока данных в бизнес-процессе. Они не обязательно предоставляют подробную информацию о том, как принимаются решения или как выбираются задачи во время выполнения процесса. Они могут быть либо ручными, либо электронными, либо сочетанием методов.

Основные форматы логического моделирования:

- письменное описание процесса;
- блок-схемы;
- диаграммы потоков данных;
- иерархии функций;
- модели в реальном времени или конечные автоматы;
- диаграммы функциональных зависимостей.

Построим логическую модель предметной области. Данную модель создадим, используя язык UML.

Язык UML используется для графического представления программного обеспечения для моделирования бизнес-процессов, проектирования систем и отображения организационной структуры [2].

Используя этот язык, отобразим основные бизнес-процессы, происходящие в клинике, и укажем их взаимодействие с окружающей средой.

На рисунке 6 представлены набор прецедентов, отражающих возможные варианты выполнения бизнес-процесса «AS-IS».

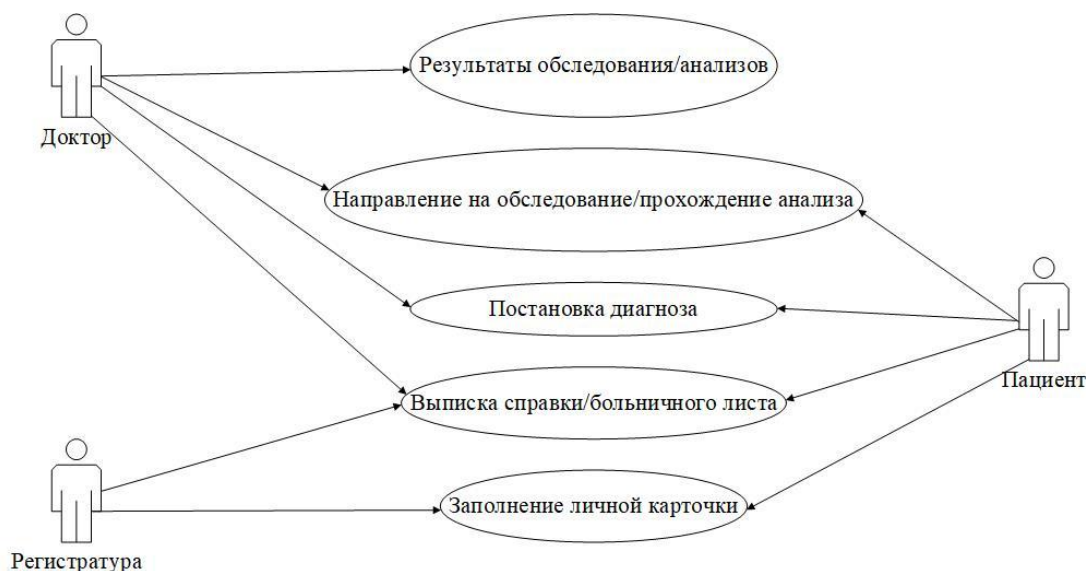


Рисунок 6 – Диаграмма вариантов использования

Поток событий прецедента «Личная карточка»:

1. Пациент обращается в регистратуру.
2. Регистратура создает личную карточку пациента.
3. Пациент получает личную карточку и записывается на прием к доктору.
4. Регистратура назначает точное время приема пациента доктором.

Поток событий прецедента «Направление на обследование/прохождение анализов»:

1. Доктор заполняет бланк направления на обследование и передает его пациенту.
2. Пациент получает бланк направления на обследование и проходит соответствующие процедуры.

Поток событий прецедента «Диагноз»:

1. Доктор на основании результатов обследования и анализов ставит соответствующий диагноз.
2. Доктор назначает курс лечения
3. Пациент ознакомляется с поставленным диагнозом и проходит курс лечения

Поток событий прецедента «Справка/больничный лист»:

1. Доктор закрывает историю болезни пациента.
2. Доктор выписывает справку или больничный лист
3. Пациент получает справку/больничный лист и регистрирует его в регистратуре

Диаграмма последовательностей бизнес-процесса оформления курса лечения пациента представлена на рисунке 7.

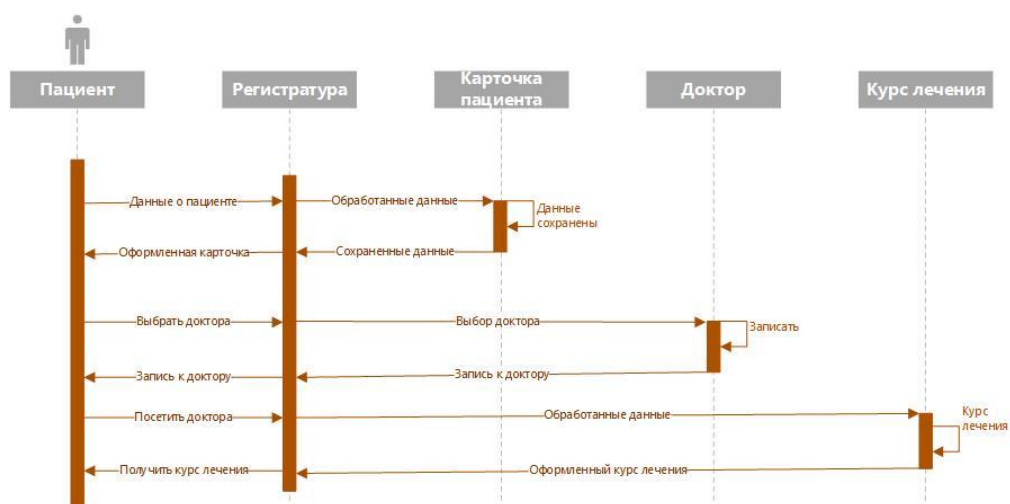


Рисунок 7 – Диаграмма последовательностей бизнес-процесса «Оформления курса лечения пациента»

С помощью диаграммы действий сделаем более подробное описание процессов нижнего уровня, в которых происходят действия, связанные с администрированием курсов лечения и работа с документами клиники.

Диаграмма бизнес-процесса «Администрирование курсов лечения» представлена на рисунке 8.

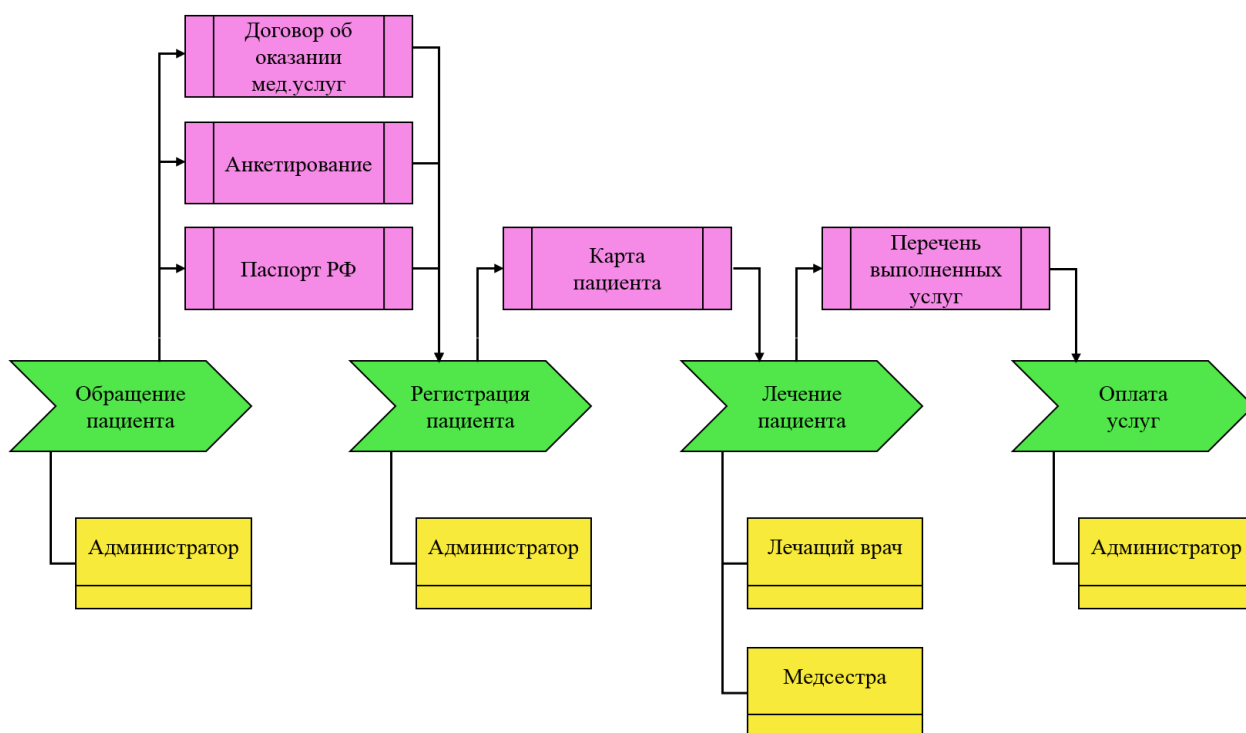


Рисунок 8 – Диаграмма «Администрирование курсов лечения»

На основании обработанных данных опишем созданную диаграмму классов. Диаграмма классов является одной из наиболее полезных типов диаграмм в UML, поскольку они четко отображают структуру конкретной системы путем моделирования ее классов, атрибутов, операций и отношений между объектами.

Диаграмма классов является частью логической модели системы. Представлена на рисунке 9.

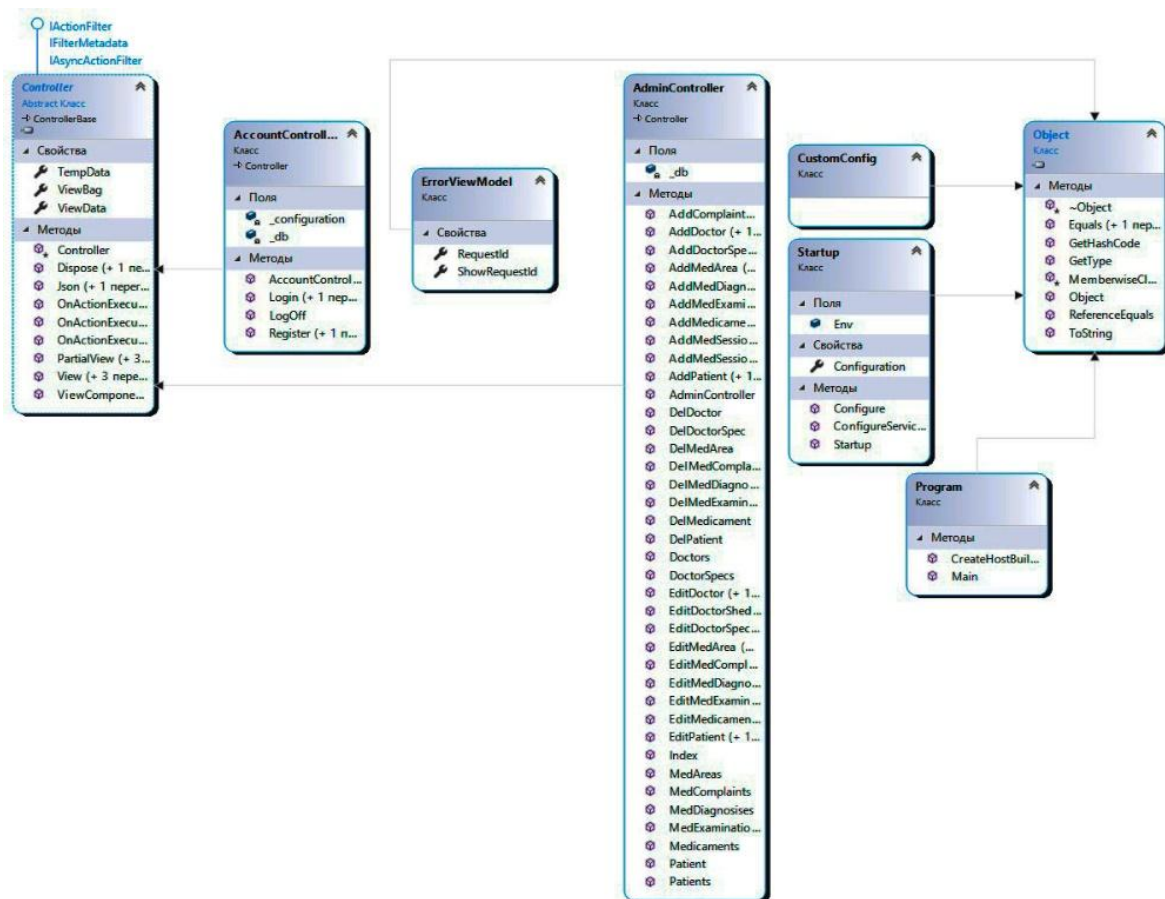


Рисунок 9 – Диаграмма классов АИС

В логической модели базы данных все сущности детализируются методом описания присвоенных им атрибутов, а также с помощью выявления идентифицирующих атрибутов (FK) и внешних ключей, с помощью которых происходит связь с другими таблицами БД. Логическая модель базы данных представлена на рисунке 10.

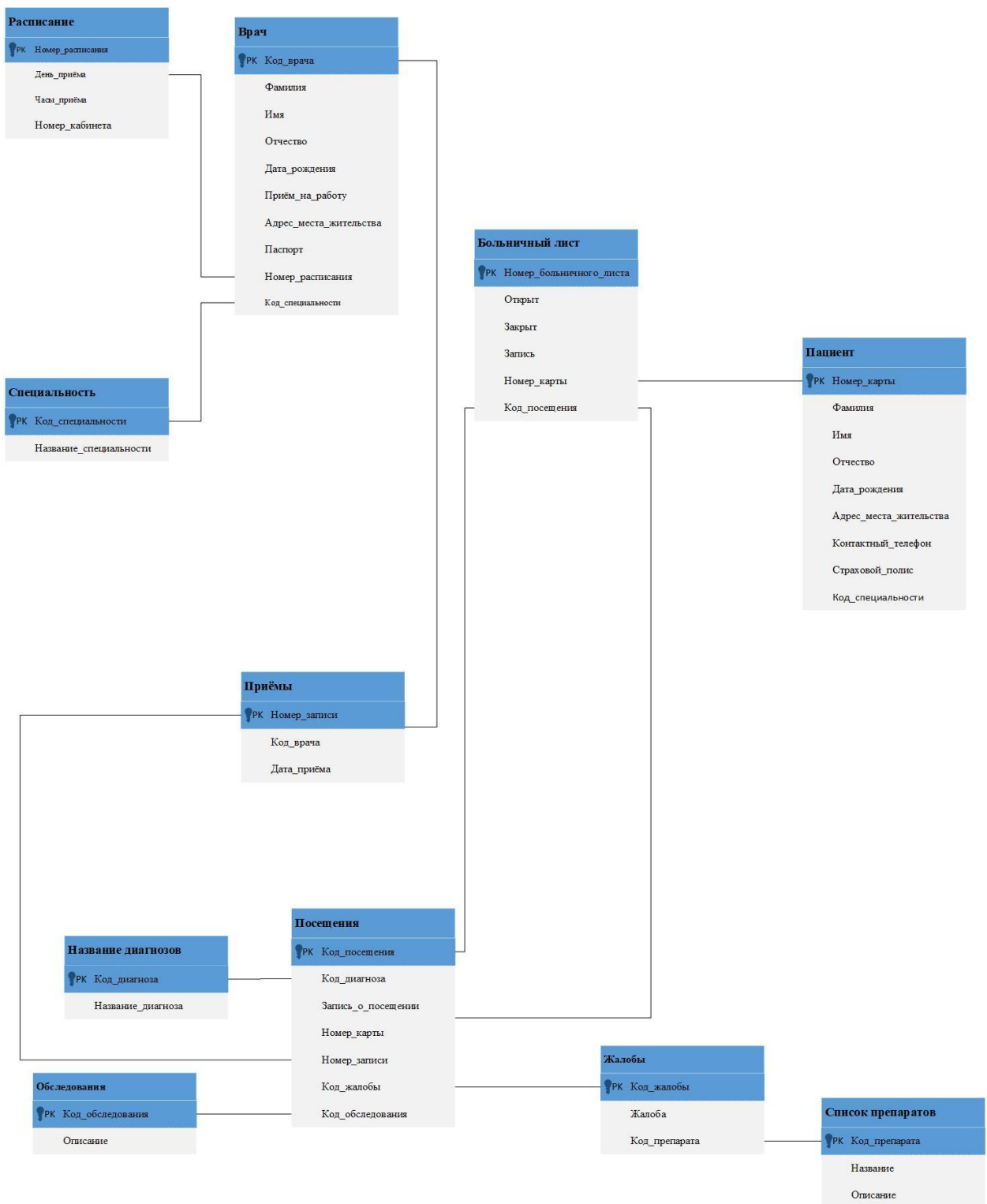


Рисунок 10 – Логическая модель базы данных

Далее опишем информационное обеспечение системы.

2.2 Информационное обеспечение системы

2.2.1 Используемые классификаторы и системы кодирования

«Классификатор – это систематизированный свод однородных наименований, объектов, предметов, явлений по классификационным признакам (номенклатура) и их кодовых обозначений. Код является условным обозначением объекта цифровым или алфавитно-цифровыми знаками по определенным правилам, установленным системами кодирования» [14].

Информационное обеспечение CRM системы представляет собой совокупность справочных и оперативных данных. Справочные данные, как правило, являются постоянными, а изменения крайне редки, в отличие от справочных данных, оперативные данные регулярно добавляются в хранилище данных.

Как часть информационного программного обеспечения CRM системы, справочные и операционные данные используют локальные классификаторы, предназначенные для идентификации соответствующих записей (таблица 5). «При создании записей в БД происходит автоматическое присваивание порядкового номера с помощью встроенных функций СУБД» [14].

Таблица 5 – Перечень классификаторов CRM системы

Наименование классификатора	Система кодирования	Система Классификации	Вид классификатора
Назначение	порядковая	линейная	локальный
Приём	порядковая	линейная	локальный
Анализы	порядковая	линейная	локальный
Диагноз	порядковая	линейная	локальный
Операция	порядковая	линейная	локальный

«Поскольку данные классификаторы имеют четырехзначную структуру, структурная формула классификаторов определяется с помощью соотношения: $\Phi = [XXXX]$ » [14].

2.2.2 Характеристика нормативно–справочной, входной и выходной информации

Входной информацией относительно CRM системы являются:

- обращение пациента;
- полис ОМС;
- паспортные данные;
- номер телефона;
- информация для заполнения справочника диагнозов и лекарств;
- критерии фильтрации и поиска информации по справочникам.

Выходными данными для CRM системы будут являться:

- таблицы справочников и учетные таблицы, содержащие данные в соответствии с запросами и фильтрами пользователей;
- выписки из амбулаторных карт пациента;
- оперативный отчет о курсе лечения (с возможностью поиска и фильтрации).

2.2.3 Требования к аппаратно–программному обеспечению информационной системы

Технический комплекс офтальмологической клиники «Кругозор» представляет собой: комплекс аппаратных средств в виде персональных компьютеров на которых установлена операционная система Windows 10.

Данным комплексом оснащены администратор клиники, врачебный персонал в кабинетах и регистратура.

Минимальные требования к рабочей станции:

- ПК на базе процессора Pentium Celeron или AMD с тактовой частотой от 2,3 ГГц;
- ОЗУ 6 ГБ с возможностью увеличения объема памяти;
- наличие свободного места на жестком диске от 10 ГБ или более;
- объем видеопамяти: от 1 ГБ или более;
- скорости интернет соединения от 10 Мбит/с;

- набор USB-контроллеров для подключения внешних периферийных устройств.

2.3 Выбор архитектуры и способов создания базы данных для разрабатываемой автоматизированной информационной системы

База данных была разработана с помощью PostgreSQL.

PostgreSQL – является объектно-реляционной базой данных, которая использует язык SQL. Главными достоинствами PostgreSQL является то, что она имеет открытый исходный код и набор функций с помощью которых происходит масштабирование и безопасное хранение данных [11].

Данная СУБД является многофункциональной, поскольку с помощью неё разработчики могут реализовать множество функций, которые упрощают создание приложения, а также защищают целостность данных и создают отказоустойчивые среды.

«Главными достоинствами PostgreSQL является то, что эта СУБД имеет открытый исходный код и обладает широкими возможностями расширения. Именно благодаря этому СУБД PostgreSQL получила большую популярность в среде программной разработки. Управление данной СУБД происходит с помощью программы PgAdmin» [12].

PgAdmin является программой, с помощью которой упрощается администрирование баз данных PostgreSQL.

2.4 Физическое моделирование БД

Опишем разработанную физическую модель базы данных. Данная модель изображена на рисунке 11.

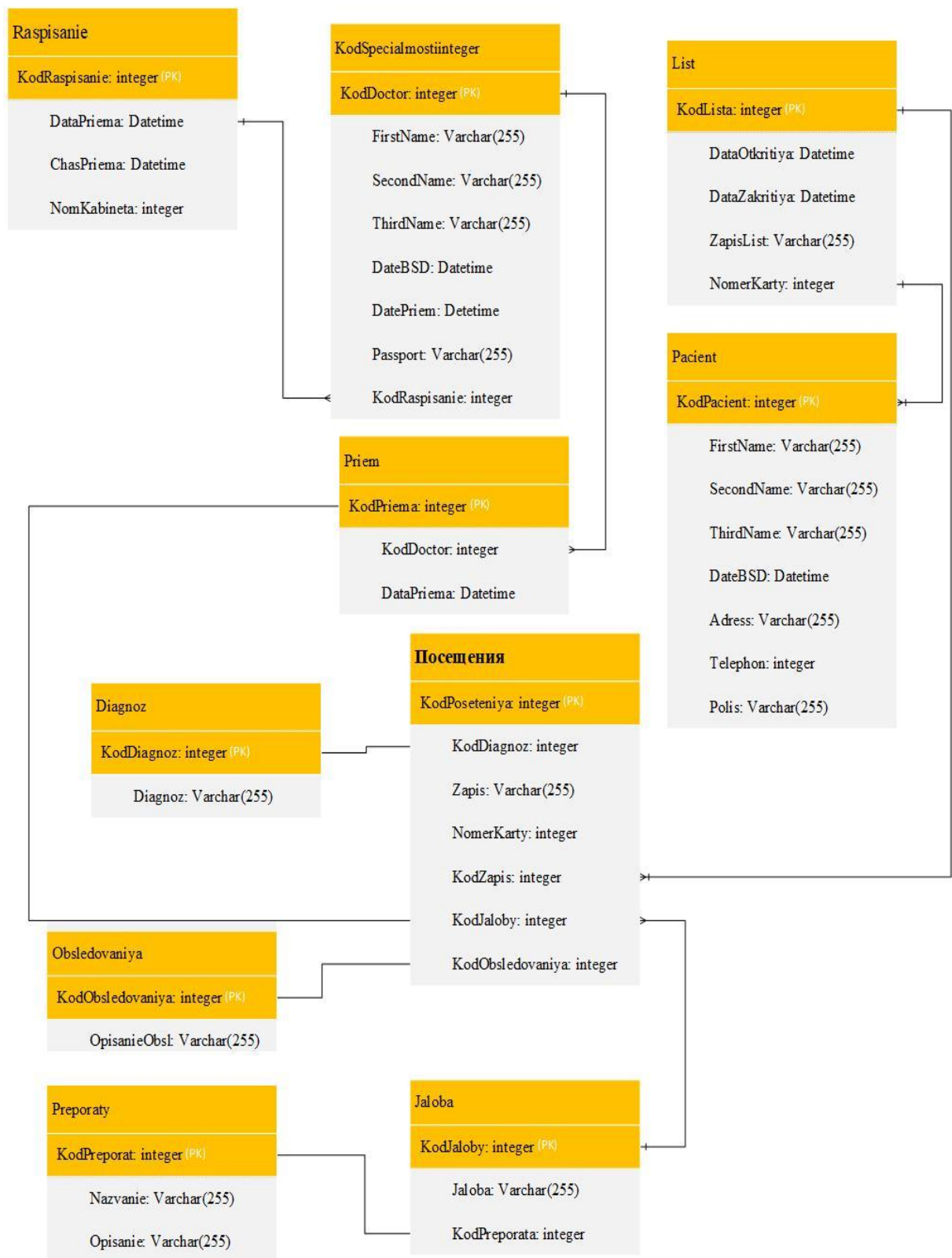


Рисунок 11 – Физическая модель базы данных

При рассмотрении предметной области было выбрано десять сущностей, атрибуты и типы данных этих атрибутов приведены в таблицах 6 – 15.

Таблица 6 – Сущность Raspisanie

Ключ	Атрибут	Тип данных	Примечание
PK	KodRaspisanie	integer	Код аббревиатуры
	ChasPriema	Datetime	Время приёма
	NomKabineta	integer	Номер кабинета

Таблица 7 – Сущность Doctor

Ключ	Атрибут	Тип данных	Примечание
PK	KodDoctor	integer	Код аббревиатуры
	FirstName	Varchar(255)	Фамилия
	SecondName	Varchar(255)	Имя
	ThirdName	Varchar(255)	Отчество
	DateBSD	Datetime	Дата рождения
	Passport	Varchar(255)	Паспортные данные
FK	KodRaspisanie	integer	Расписание
FK	KodSpecialmosti	integer	Специальность

Таблица 8 – Сущность Pacient

Ключ	Атрибут	Тип данных	Примечание
PK	KodPacient	integer	Код аббревиатуры
	FirstName	Varchar(255)	Фамилия
	SecondName	Varchar(255)	Имя
	ThirdName	Varchar(255)	Отчество
	DateBSD	Datetime	Дата рождения
	Adress	Varchar(255)	Адрес
	Polis	Varchar(255)	Данные полиса ОМС
	Telephon	integer	Контактный телефон

Таблица 9 – Сущность Poseteniya

Ключ	Атрибут	Тип данных	Примечание
PK	KodPoseteniya	integer	Код аббревиатуры
FK	KodDiagnoz	integer	Диагноз
	Zapis	Varchar(255)	Запись о посещении
FK	NomerKarty	integer	Карта пациента
FK	KodJaloby	integer	Данные о приеме пациента
FK	KodObsledovaniya	integer	Данные обследования

Таблица 10 – Сущность List

Ключ	Атрибут	Тип данных	Примечание
PK	KodLista	integer	Код аббревиатуры
	DataOtkritiya	Datetime	Дата открытия
	DataZakritiya	Datetime	Дата закрытия
FK	NomerKarty	integer	Номер карты пациента
FK	KodJaloby	integer	Посещение пациента
	ZapisList	Varchar(255)	Запись в листе

Таблица 11 – Сущность Priem

Ключ	Атрибут	Тип данных	Примечание
PK	KodPriema	integer	Код аббревиатуры
	DataPriema	Datetime	Дата приёма
FK	KodDoctor	integer	Назначенный доктор

Таблица 12 – Сущность Preporaty

Ключ	Атрибут	Тип данных	Примечание
PK	KodPreporata	integer	Код аббревиатуры
	Nazvanie	Varchar(255)	Название препарата
	Opisanie	Varchar(255)	Описание препарата

Таблица 13 – Сущность Jaloba

Ключ	Атрибут	Тип данных	Примечание
PK	KodJaloby	integer	Код аббревиатуры
	Jaloba	Varchar(255)	Описание состояния пациента
FK	KodPreporata	integer	Назначенное лекарство

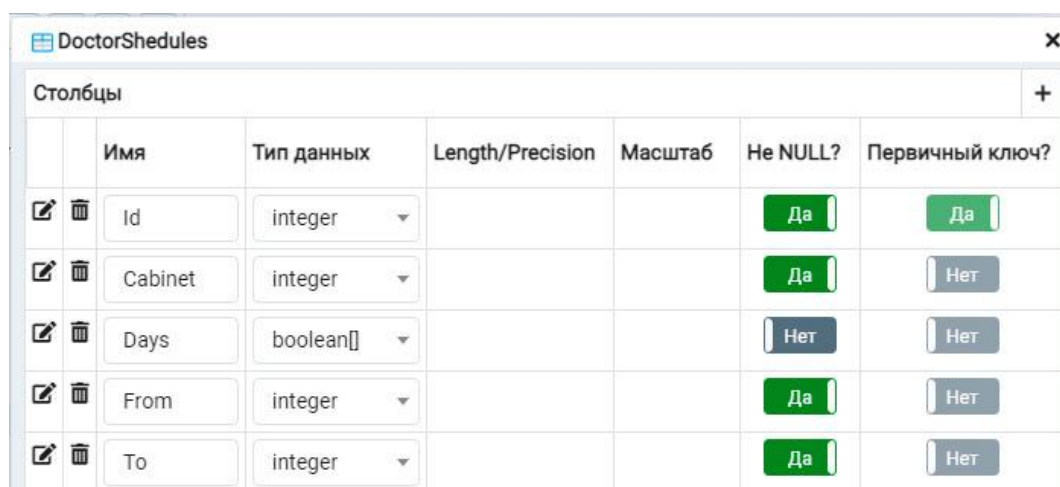
Таблица 14 – Сущность Obsledovaniya

Ключ	Атрибут	Тип данных	Примечание
PK	KodObsledovaniya	integer	Код аббревиатуры
	OpisanieObsl	Varchar(255)	Результаты обследования

Таблица 15 – Сущность Diagnoz

Ключ	Атрибут	Тип данных	Примечание
PK	KodDiagnoza	integer	Код аббревиатуры
	Diagnoz	Varchar(255)	Диагноз

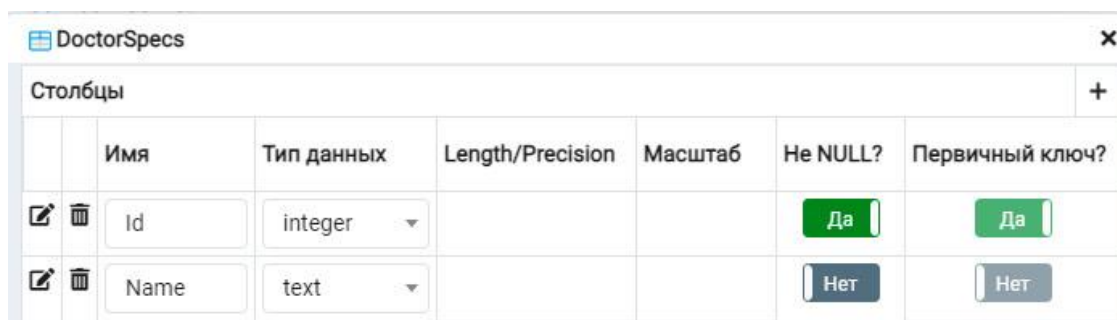
Таблица «DoctorSchedules» хранит информацию о расписании врачебного персонала. Скриншот данной таблицы показан на рисунке 12.



DoctorSchedules						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Cabinet	integer			Да	Нет
	Days	boolean[]			Нет	Нет
	From	integer			Да	Нет
	To	integer			Да	Нет

Рисунок 12 – Таблица «DoctorSchedules»

Таблица «DoctorSpecs» хранит информацию о специальностях врачебного персонала. Скриншот данной таблицы показан на рисунке 13.



DoctorSpecs						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Name	text			Нет	Нет

Рисунок 13 – Таблица «DoctorSpecs»

Таблица «Doctors» хранит информацию о врачебном персонале. Скриншот данной таблицы показан на рисунке 14.

Doctors						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	FirstName	text			Нет	Нет
	SecondName	text			Нет	Нет
	ThirdName	text			Нет	Нет
	Passport	text			Нет	Нет
	Date	timestamp without time zone			Да	Нет
	MedAreald	integer			Да	Нет
	DoctorSheduleId	integer			Да	Нет
	DoctorSpecId	integer			Да	Нет

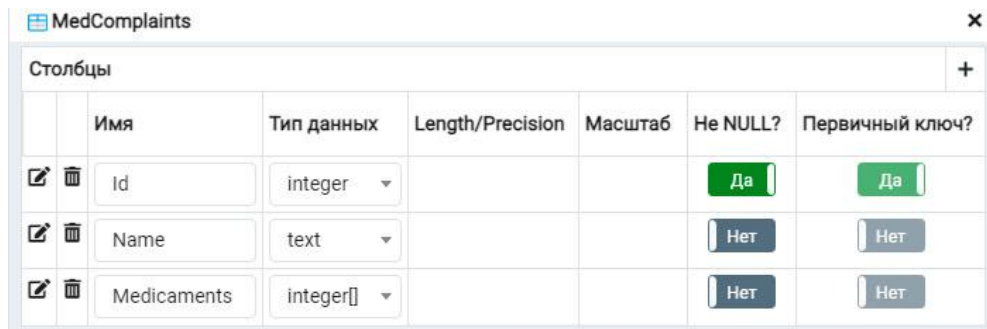
Рисунок 14 – Таблица «Doctors»

Таблица «MedAreas» хранит информацию об участках, за которыми закреплены представители медперсонала. Скриншот данной таблицы показан на рисунке 15.

MedAreas						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Name	text			Нет	Нет
	Adress	text			Нет	Нет

Рисунок 15 – Таблица «MedAreas»

Таблица «MedComplaints» хранит информацию о жалобах пациентов клиники. Скриншот данной таблицы показан на рисунке 16.



MedComplaints						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Name	text			Нет	Нет
	Medicaments	integer[]			Нет	Нет

Рисунок 16 – Таблица «MedComplaints»

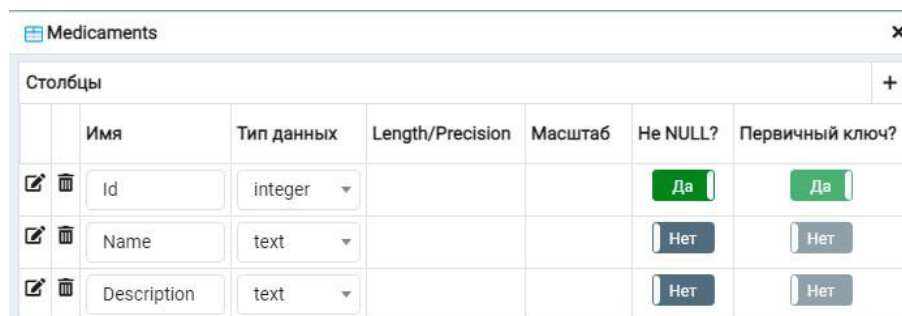
Таблица «MedDiagnosises» хранит информацию о диагнозах пациентов. Скриншот данной таблицы показан на рисунке 17.



MedDiagnosises						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Name	text			Нет	Нет

Рисунок 17 – Таблица «MedDiagnosises»

Таблица «Medicaments» хранит информацию о лекарствах, назначаемых пациентам. Скриншот данной таблицы находится на рисунке 18.



Medicaments						
Столбцы						
	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Name	text			Нет	Нет
	Description	text			Нет	Нет

Рисунок 18 – Таблица «Medicaments»

Таблица «Patients» хранит информацию о данных пациентов. Скриншот данной таблицы находится на рисунке 19.

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Araid	integer			Да	Нет
	FirstName	text			Нет	Нет
	SecondName	text			Нет	Нет
	ThirdName	text			Нет	Нет
	Adress	text			Нет	Нет
	Phone	text			Нет	Нет
	Policy	text			Нет	Нет
	BDate	timestamp without time zone			Да	Нет

Рисунок 19 – Таблица «Patients»

Таблица «MedSessionResults» хранит информацию о приеме доктором пациента. Скриншот данной таблицы находится на рисунке 20.

	Имя	Тип данных	Length/Precision	Масштаб	Не NULL?	Первичный ключ?
	Id	integer			Да	Да
	Araid	integer			Да	Нет
	FirstName	text			Нет	Нет
	SecondName	text			Нет	Нет
	ThirdName	text			Нет	Нет
	Adress	text			Нет	Нет
	Phone	text			Нет	Нет
	Policy	text			Нет	Нет
	BDate	timestamp without time zone			Да	Нет

Рисунок 20 – Таблица «MedSessionResults»

Далее опишем выбранные методы физического моделирования CRM системы для медицинского учреждения.

2.5 Физическое моделирование разрабатываемой CRM системы для медицинского учреждения

Методы физического моделирования определяют топологию, данные, роли и правила бизнес-процесса. Эта модель описывает такие предметы, как:

- рабочие задачи, которые необходимо выполнить в процессе;
- порядок выполнения задач;
- данные, необходимые для запуска процесса;
- данные, необходимые для начала и завершения каждой рабочей задачи;
- правила, необходимые для определения маршрутизации в процессе;
- методы обработки исключений;
- по крайней мере, один определенный бизнес-результат;
- роли и разрешения каждого участника процесса.

2.5.1 Выбор архитектуры, разрабатываемой автоматизированной информационной системы CRM

Локально-вычислительная сеть клиники «Кругозор» состоит из кабелей, точек доступа, коммутаторов, маршрутизаторов и других компонентов, которые позволяют устройствам подключаться к внутренним серверам, веб-серверам через глобальные сети [5].

Благодаря этому в клинике, в которой существуют отделы: бухгалтерия, ИТ-поддержка и администрация, компьютеры каждого отдела могут быть логически подключены к одному и тому же коммутатору, но сегментированы, чтобы вести себя так, как если бы они были отдельными.

Именно поэтому для разработанной АИС следует использовать архитектуру «клиент–сервер», с помощью которой происходит разделение функций приложения пользователя (называемого клиентом) и сервера.

С помощью архитектуры «клиент–сервер» происходит разделение обязанностей по обработке данных между сервером и клиентом. Сервером

обычно является высокопроизводительная рабочая станция, а клиентом обычно являются ПК.

Современные клиент-серверные архитектуры обмениваются сообщениями по локальным сетям.

База данных работает на собственном сервере (сервере базы данных), используя дополнительное дисковое пространство на подключенном к сети запоминающем устройстве.

Создание физической архитектуры АИС, происходит с помощью диаграммы развертывания UML [20]. Эта диаграмма обеспечивает единообразное представление различных компонентов системы. Данная диаграмма представлена на рисунке 21.

Архитектура разрабатываемой АИС включает:

- рабочую станцию в виде рабочего места сотрудника клиники;
- web-сервер, с помощью которого происходит взаимодействие между рабочей станцией и сервером базы данных;
- сервер базы данных с СУБД PostgreSQL.

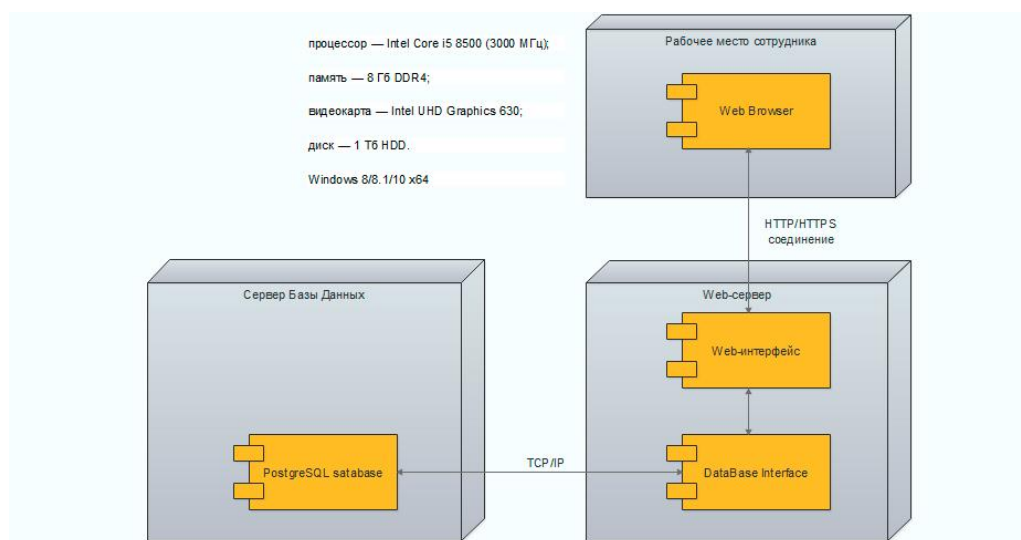


Рисунок 21 – UML-диаграмма развертывания АИС

Далее обоснуем выбор средств реализации АИС.

2.5.2 Обоснование выбора средств для реализации, разрабатываемой CRM для медицинского учреждения

Опишем язык и выбранную технологию, с помощью которой проводили разработку CRM системы. Были использованы язык программирования C# и технология NET CORE MVC.

Язык C# является современным объектно-ориентированным языком программирования. С помощью C# разработчики могут создавать множество типов безопасных и надежных приложений, работающих в экосистеме .NET [4].

В языке C# уделяется особое внимание управления версиями для обеспечения совместимости программ и библиотек при их изменении [5].

«Технология ASP.NET Core предлагает встроенную поддержку внедрения зависимостей, благодаря этому данная архитектура является наиболее оптимальным подходом к структурированию нетривиальных монолитных приложений» [13].

«При работе с монолитными приложениями проекты ядра приложения, инфраструктуры и пользовательского интерфейса выполняются как единое приложение. Следует также учесть, что одним из отличительных моментов платформы ASP.NET Core является применение паттерна MVC. Работа данного фреймворка происходит поверх платформы ASP.NET Core, и его предназначение в том, чтобы упростить создание приложения» [13].

Взаимодействие компонентов данной архитектуры представлены на рисунке 22.

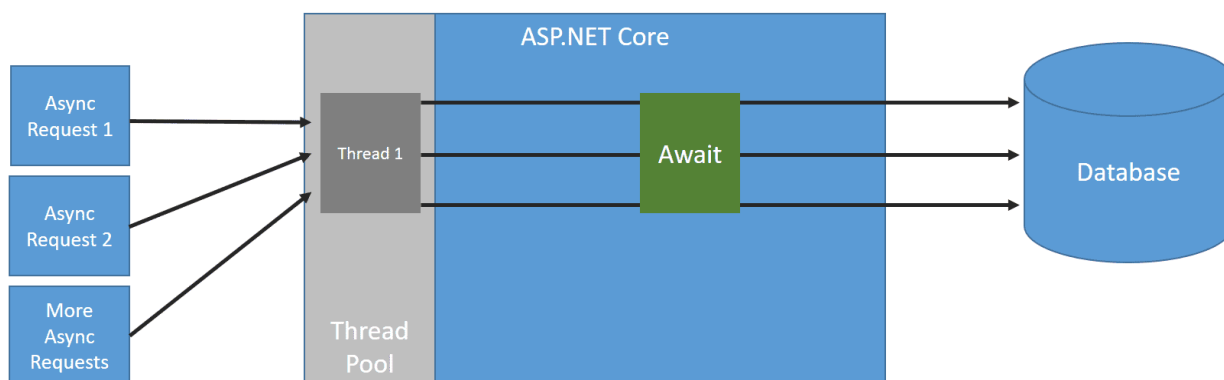


Рисунок 22 – Схема взаимодействия компонентов ASP.NET Core

Концепция паттерна MVC состоит из трех компонентов: модель, представления и контроллер. Рассмотрим более подробно каждый из них.

При помощи такого компонента как модель происходит описание используемых данных и связанной с ними логики. Хранение объектов моделей происходит в БД. Концепция паттерна MVC представляет данные модели двумя основными типами: модели представлений и модели домена. Модели представления используются для отображения и передачи данных, а с помощью модели домена описывается логика управления данными [13].

С помощью компонента представления происходит отображение, а также контроль за визуальной частью или пользовательским интерфейсом с помощью html-страниц.

Центральным компонентом является контроллер, поскольку данный компонент отвечает за связь между пользователем и приложением. В нём содержится логика обработки запросов между пользователем и приложением.

Отношения внутри паттерна MVC между его компонентами отображены на рисунке 23.

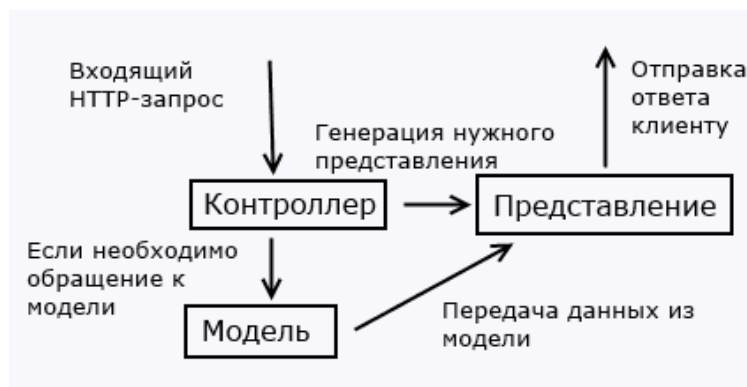


Рисунок 23 – Отношения внутри паттерна MVC между его компонентами

В данной схеме видно, что независимым компонентом является модель, поскольку изменение контроллера или представления не влияет на модель приложения.

Контроллер и представление являются относительно независимыми компонентами, поскольку присутствует возможность изменения данных компонентов независимо друг от друга.

Реализация концепции, которая разделяет ответственность, строго отвечая за свою сферу, возможна с помощью разграничения компонентов. Благодаря этому облегчается построение работы над каждым компонентом в отдельности, их разработка, а также в дальнейшем поддержка и тестирование.

Далее опишем процессы создания разработанной CRM системы для офтальмологической клиники «Кругозор».

2.6 Описание разработанной автоматизированной информационной CRM системы для медицинского учреждения

Рассмотрим подробнее содержание проекта, построенного с помощью технологии ASP.NET Core MVC и программы Microsoft Visual Studio Community. Опишем процесс разработки CRM [16].

2.6.1 Описание модулей приложения

В процессе разработки системы получены следующие результаты:

1. Создана доменная модель.
2. Созданы модели сущностей для базы данных.
3. Указаны поля сущностей и их связи для последующей генерации таблиц в БД.
4. Создан контроллер для учета пользователей (Авторизация/Регистрация).
5. Созданы базовые методы для прохождения регистрации и авторизации.
6. Созданы контроллер для панели администратора.
7. Созданы методы отображения и сортировки данных из БД.
8. Созданы методы создания/изменения/удаления записей из таблиц.
9. Созданы сложные методы для отображения связанных данных из разных таблиц.

Файловая структура созданного проекта включает:

- папка Properties – свойства проекта;
- папка wwwroot – корневая папка сайта, содержащая изображения, а также файлы css и js сайта;
- зависимости – подгруженные библиотеки и плагины;
- AccountController.cs – контроллер, отвечающий за регистрацию и вход пользователя в систему;
- AdminController.cs – контроллер, реализующий взаимосвязь работы программы и БД;
- Migrations – папка с файлами миграций с помощью которых создается и происходит работа с БД с помощью Entity Framework;
- папка Domain – данная папка содержит доменную модель, в которой представлены файлы форм (админ, доктор, пациент, диагноз и т.д.);

- Models – папка, содержащая классы моделей данных;
- Views – папка файлов html сайта с помощью которых отображаются созданные формы;
- appsettings.json – файл содержащий данные свойств проекта (имя сервера, номер порта, учетные данные для работы с БД);
- package.json – файл в котором приписана информация про версию ASP.NET;
- Program.cs – файл одноименного класса, содержащий метод Main, который является точкой входа приложения ASP.NET Core;
- Startup.cs – файл одноименного класса с помощью которого происходит конфигурация приложения и настройка сервисов необходимых для запуска программы. С помощью него также происходит регистрация набора ПО в промежуточном слое приложения.

2.6.2 Описание функциональности приложения

При запуске CRM системы появляется окно авторизации, с помощью которого сотрудник клиники может зайти в систему или зарегистрироваться в ней. Данные о регистрации хранятся в соответствующей таблице. Форма окна регистрации представлена на рисунке 24.

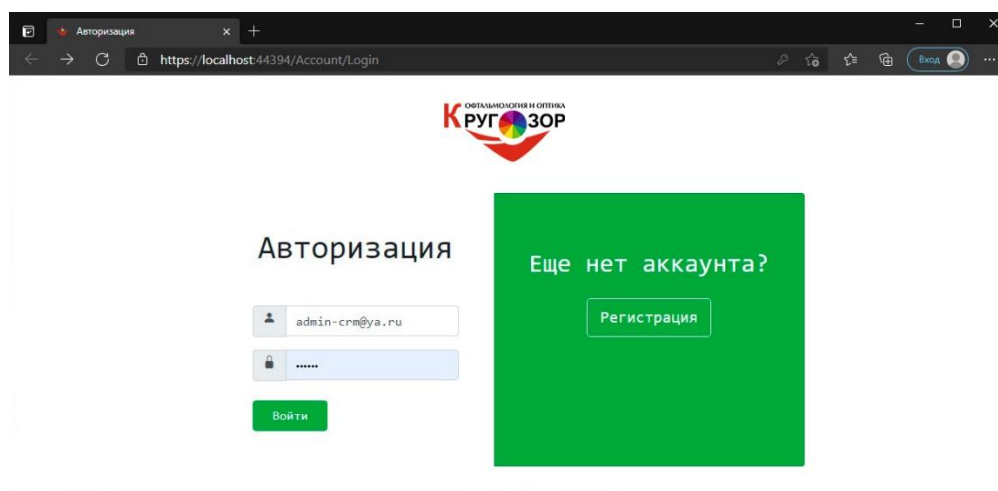


Рисунок 24 – Авторизация в системе

Далее после авторизации сотруднику клиники (например, доктор) назначают пациента, и сотрудник обладает возможностями курировать процесс лечения пациента и вводить данные о нём. Эти данные вводятся в форму истории болезни. Форма истории болезни пациента отображена на рисунке 25.

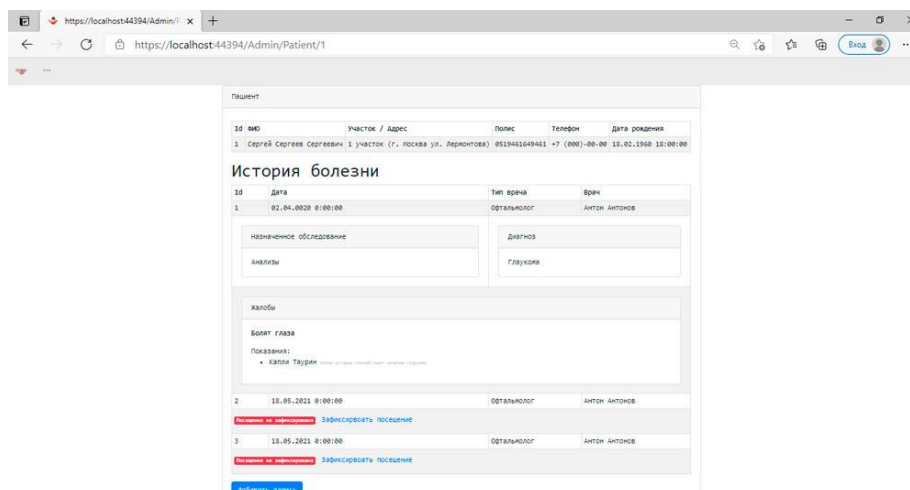


Рисунок 25 – История болезни пациента

Также отметим, что разработанную программу можно расширить функционалом и увеличить количество возможностей.

Выводы по главе 2

На основании диаграммы классов была разработана БД, состоящая из 11 таблиц. При создании БД была сформирована входная информация, с помощью которой запускается работа CRM системы.

При построении АИС был обоснован выбор языка программирования и архитектуры, с помощью которых были разработаны интерфейс программы и сама CRM система.

Глава 3 Расчет экономической эффективности разработанной CRM системы для медицинского учреждения

3.1 Выбор методики расчета экономической эффективности

Экономическая эффективность рассчитывается для того, чтобы показать насколько успешным является результат деятельности компании. Данная величина является показателем, отображающим соотношение итогов деятельности за определенный период и потраченными финансами на данную деятельность. Экономическая эффективность представляет собой соотношение затрат на:

- разработку и внедрение системы;
- эксплуатацию системы;
- модернизацию системы;
- полученной прибыли от её применения.

Для того, чтобы произвести оценку эффективности разработанной системы используются два типа показателей – главные и дополнительные (дифференцированные) показатели. Главные показатели называют обобщающими, дифференцированные показатели отображают в себе показатели с помощью которых производится функциональная характеристика, затрагивающая определённую сторону деятельности.

Для показателей экономической эффективности используется условно-годовая экономия с целью охарактеризовать общую эффективность внедряемого мероприятия, направленного на покрытие затрат. Условно-годовая экономия отображается расчетами за год. Текущая экономия учитывает в своём расчете время внедрения до конца календарного года.

Экономия текущих затрат определяется по тем статьям себестоимости, которые непосредственно затронуты данным мероприятием.

Амортизация затрат на разработку происходит после снижения текущих издержек. При освоении приобретенной техники амортизационные

отчисления начинают постепенно увеличиваться и только после полного ввода технических средств в эксплуатацию амортизационные отчисления уменьшаются из-за износа техники.

Прирост прибыли предприятия обусловлен экономией текущих затрат. Поскольку при подсчете прибыли учитывается соответствующий налог выплачиваемый и указанный в налоговой декларации предприятия не следует относить всю величину экономии текущих затрат на прирост чистой прибыли предприятия.

При оценке эффекта от функционирования подсистем бухгалтерского учета и управления финансами для которых характерен большой объём вычислений расчет экономии текущих затрат осуществляется с помощью ЭВМ.

Для того, чтобы произвести расчет текущей экономии следует годовую экономию разделить на 12 и умножить на число месяцев, оставшихся до конца года с момента полного внедрения мероприятия.

Во время эксплуатации основных фондов происходит их износ, и основные фонды теряют часть первичной стоимости. Существуют 2 вида износа основных фондов: моральный и физический.

Моральный износ появляется из-за развития технологий. Поскольку купленное ранее оборудование начинает терять в цене независимо от физического износа.

Физический износ возникает из-за природных и эксплуатационных факторов.

Теперь приступим к расчету годовой суммы амортизации. Амортизация подразумевает под собой финансовый показатель износа основных средств. Расчет амортизации производится с помощью систематического распределения стоимости объекта в течение срока полезного использования, превышающего временной промежуток 12 месяцев. Экономическая сущность состоит в том, что с помощью амортизационных отчислений предприятие в будущем обеспечивает для себя

возмещение потраченных средств на объект. Амортизационные отчисления включают в ежемесячные расходы.

Показатель величины суммарных годовых амортизационных отчислений рассчитывается по формуле (1):

$$A = \frac{\Phi * NA}{100\%} \quad (1)$$

где Φ – первоначальная величина стоимости основных фондов, руб.;

NA – показатель нормы амортизации, в %.

Амортизируемым оборудованием признается то, на которое предприятие имеет право собственности и использует его с целью получения дохода. К амортизируемому должно относиться оборудование со сроком полезного использования более 12 месяцев, первоначальная стоимость которого более 100 000 руб.

Амортизацию оборудования за период можно рассчитать по формуле (2):

$$A_{\text{факт}} = \frac{A_{\text{год}} * T_{\text{факт}}}{365} \quad (2)$$

где $A_{\text{год}}$ – амортизация оборудования за год,

$T_{\text{факт}}$ – период для начисления амортизации.

Проведем расчет затрат на заработную плату специалистов, которые занимаются разработкой, внедрением и поддержкой работоспособности CRM системы. Поскольку заработная плата различается между сотрудниками, работающими по системе почасовой ставки и производственных рабочих сдельщиков. У специалистов, которые работают по системе почасовой оплаты труда, сумма заработной платы определяется исходя из тарифной ставки, указанной за отработанное время, а также учитываются действующие положения премий и доплат. Для сотрудников, которые работают по системе

оплаты труда на сдельной основе, проводится расчет зарплаты исходя из объёма выполненной работы. Расчёт заработной платы не учитывает расходы, связанные со стоимостью необходимого оборудования его монтаж и наладку, а также затраты на необходимое обучение персонала предприятия.

Затраты на заработную плату рассчитываются по формуле (3):

$$ЗП_{\text{пр}} = \frac{ЗП_{\text{мес}} * T_{\text{факт}}}{Д} \quad (3)$$

где $З_{\text{пр}}$ – месячная зарплата исполнителя, руб.;

$T_{\text{факт}}$ – количество потраченных календарных дней на разработку системы;

$Д$ – количество дней в периоде.

Для того чтобы определить величину расходов на разработку следует произвести суммирование месячных расходов и умножить данный показатель на число, которое равно количеству месяцев разработки. Для расчета величины расходов происходит группирование затрат по экономическим элементам.

При расчете затрат на материалы учитывается стоимость данных материалов, а также финансы, которые были потрачены на оборудование и монтаж. Расчет затрат на разработку ПО должны включать в себя следующие виды расходов:

- материальные затраты;
- основная заработная плата;
- единый социальный налог;
- накладные расходы.

Проведем расчет суммы затрат на период разработки ПО по формуле (4):

$$З_{\text{пр}} = \frac{З_{\text{мес}} * T_{\text{факт}}}{Д}, \quad (4)$$

где $Z_{\text{мес}}$ – сумма ежемесячных затрат, руб.;

$T_{\text{факт}}$ – количество потраченных календарных дней на разработку CRM;

D – количество дней в периоде.

Приступим к расчету себестоимости продукции (калькуляции). Поскольку себестоимость продукции является важным показателем и одним из ключевых факторов деятельности организации, с помощью которой отображается правильность управленческих решений, сделаем выбор метода расчёта [13].

С помощью калькуляции можно определить фактическую или плановую себестоимость продукции. Калькуляция определяет затраты на производство единицы товара в денежной форме.

Для определения стоимости разработанной системы воспользуемся формулой:

$$Ц = \frac{C_{\text{сп}} * R}{100\%} + C_{\text{ст}}, \quad (5)$$

где $C_{\text{ст}}$ – показатель себестоимости разработки ПО;

R – планируемый уровень рентабельности.

После того, как произвели выбор методики расчета экономической эффективности, перейдем к расчету показателей экономической эффективности разработанной АИС.

3.2 Расчет показателей экономической эффективности, разработанной CRM системы для медицинского учреждения

Проведем расчет показателей экономической эффективности, разработанной CRM системы для офтальмологической клиники «Кругозор». При расчете учтем следующие виды расходов (таблица 16-17):

- расходы затрат на электроэнергию;

- заработная плата разработчика CRM системы;
- расходные и комплектующие материалы.

Проведём расчет затрат электроэнергии для стандартного размера рабочего дня, который равен 8 часам.

Тариф электроэнергии для предприятия 1 кВт/ч=7,4 руб.

Расход на электроэнергию в месяц составит $7,4 \cdot 372 = 2752,8$ рублей.

Заработная плата разработчика АИС составит 60000 рублей.

Таблица 16 – Расчет стоимости затрат электроэнергии на оборудование

Наименование оборудования	Количество	кВт/час	кВт/сутки	кВт/месяц
Компьютер	1	0,25	2	60
Освещение помещения	1	0,4	3,2	96
Система климат-контроля	1	0,9	7,2	216
ИТОГО:		1,55	12,4	372

Таблица 17 – Расчет затрат на создание АИС в месяц

Наименование затрат	Сумма, руб.	Налоги, руб.
Заработная плата разработчика	60000	18120
Расходы на электроэнергию	2752,8	
Расходы на услуги интернет-провайдера	1500	
Прочие расходы	2000	
ИТОГО:	66 252,8	18120
Всего расходов	84 372,8	

$RP_{\text{ост}} = 84\,372,8$ руб. – сумма постоянных месячных расходов при разработке АИС.

При расчете суммы годовых амортизационных отчислений используем формулу (1), результаты этих расчетов запишем в таблицу 18.

Таблица 18 – Расчет суммы амортизационных отчислений за год

Список основных фондов	Количество	Стоимость, руб.	Суммарная стоимость, руб.	Показатель нормы амортизации, %	Сумма амортизационных отчислений, руб.
Компьютер	1	50000	50000	20	10000
Система климат-контроля	1	25000	25000	20	5000
Рабочее помещение	20м ²	1500	30000	3	900
ИТОГО:					15900

Сумма амортизационных отчислений за годовой период составляет 15900 рублей.

Теперь проведем расчет суммы амортизационных отчислений по формуле (2):

$$A_{\text{факт}} = \frac{15900 * 10}{365} = 435,62 \text{ рублей}$$

Расчет затрат на период разработки АИС сделаем по формуле (4):

$$A_{\text{факт}} = \frac{84\,372,8 * 10}{22} = 38352 \text{ руб.}$$

Далее проведем расчет себестоимости. Расчёт себестоимости происходит по формуле (6):

$$C = M_3 + A_0 + P_{\text{от}} + O_{\text{гес}} + O_{\text{омс}} + O_{\text{оси}} + O_{\text{кр}} + Z_{\text{др}}, \quad (6)$$

где M_3 – сумма материальных затрат;

A_0 – величина амортизационных отчислений;

$P_{\text{от}}$ – расходы на заработную плату сотрудникам;

$O_{\text{гес}}$ – оплата социального страхования;

$O_{\text{омс}}$ – отчисления по обязательному медицинскому страхованию;

$O_{\text{оси}}$ – отчисления связанные с обязательным страхованием имущества;

$O_{\text{кр}}$ – плата по процентам за кредиты банков; $Z_{\text{др}}$ – другие затраты на производство и реализацию продукции.

Расчет себестоимости АИС будем делать по формуле (7):

$$C_{\text{ст}} = 3_{\text{пр}} + A \quad (7)$$

где $C_{\text{ст}}$ – показатель себестоимости разработки программы

$$C_{\text{ст}} = 38352 + 435.62 = 38\,787,62 \text{ рублей.}$$

Отметим, что при расчете себестоимости не учтены мелкие детали, которые существенно на итог не повлияют. Потому данный расчёт является приблизительным – $C_{\text{ст}} \approx 38800$ рублей.

Проведем расчет цены, разработанной нами АИС, используя формулу (5):

$$Ц = 38800 + 38800 * 20 / 100 = 46560 \text{ рублей.}$$

Сумма затрат на внедрение АИС составит 46560 рублей.

Теперь проведем расчеты экономической эффективности внедрения CRM системы. Рассчитаем прямой эффект трудовых показателей [15].

Рассчитаем абсолютное снижение трудовых затрат ΔT по формуле:

$$\Delta T = T_0 - T_1 \quad (8)$$

где T_0 – время, затрачиваемое на выполнение автоматизируемых операций в базовом варианте;

T_1 – время, затрачиваемое на выполнение автоматизируемых операций в проектном варианте;

$$\Delta T = 8 - 6 = 2$$

Далее вычислим коэффициент относительного снижения трудовых затрат K_T (в процентах), для расчета которого используется формула

$$K_T = \frac{\Delta T}{T_0} * 100\% \quad (9)$$

$$K_T = \frac{2}{8} * 100\% = 25\%$$

Далее рассчитаем индекс снижения трудовых затрат Y_T , рассчитываемый следующим образом:

$$Y_T = T_0 / T_1 \quad (10)$$

$$Y_T = \frac{8}{6} = 1.33$$

К стоимостным показателям относятся:

1. абсолютное снижение стоимостных затрат

$$\Delta C = C_0 - C_1 \quad (11)$$

где C_0 – стоимостные затраты на обработку информации по базовому варианту;

C_1 – стоимостные затраты на обработку информации по предлагаемому варианту;

$$\Delta C = 25000 - 20000 = 5000$$

2. коэффициент относительного снижения стоимостных затрат K_c (в процентах), определяемый по формуле

$$K_c = \frac{\Delta C}{C_0} * 100\% \quad (12)$$

$$K_c = \frac{5000}{25000} * 100\% = 20\%$$

3. индекс снижения стоимостных затрат Y_c рассчитываемый по формуле

$$Y_c = \frac{C_0}{C_1} \quad (13)$$

$$Y_c = \frac{25000}{20000} = 1,25$$

Коэффициент K_c и индекс Y_c характеризуют рост производительности труда за счет внедрения более экономичного варианта проектного решения.

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение проекта $T_{ок}$ (14).

$$T_{\text{ок}} = \frac{K_{\text{п}}}{\Delta C} \quad (14)$$

где $K_{\text{п}}$ – капитальные затраты на создание проекта.

$$T_{\text{ок}} = \frac{46560}{5000} = 9,3$$

На основании описанной методики необходимо выполнить расчет показателей экономической эффективности, заполнить таблицу (таблица 18) и сделать вывод об экономической обоснованности внедрения автоматизированной системы [16].

Таблица 18 – Показатели эффективности от внедрения проекта автоматизации

Трудоемкость	Затраты		Абсолютное изменение затрат ΔT	Коэффициент изменения затрат $K_{\text{т}}, \%$	Индекс изменения затрат $Y_{\text{т}}$
	Базовый вариант	Проектный вариант			
	T_0 , час	T_1 , час			
	8	6	2	20%	1,33
Стоимость	C_0 , (руб)	C_1 , (руб)	ΔC (руб)	$K_{\text{с}}, \%$	$Y_{\text{с}}$
	25000	20000	5000	20%	1,25

На основании данных расчетов можно сделать вывод, что система является экономически эффективной и срок её окупаемости составляет 9 месяцев.

Выводы по главе 3

В ходе выполнения третьей главы были произведены: выбор методики расчёта и произведен расчет показателей экономической эффективности, разработанной CRM системы для медицинского учреждения. На основании проведенного расчета делаем вывод что внедрение данной системы дает значительную экономическую выгоду сотрудникам клиники.

В заключении следует отметить, что более точные показатели практической и экономической эффективности будут получены после разработки и полного внедрения АСУ клиникой.

Заключение

В ходе выполнения первой главы была дана характеристика компании, построена ее организационная структура. Для бизнес-процесса «Организация обслуживания пациентов офтальмологической клиники» были построены модели «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ» с помощью методологии IDEF0.

С помощью анализа модели «КАК ЕСТЬ» были выявлены недостатки процесса оказания медицинских услуг. Далее был проведен анализ существующих решений на рынке АИС с медицинским уклоном, в которых были отмечены достоинства и недостатки, на основании которых были поставлены задачи последующей разработки новой АИС.

Во второй главе, которая посвящена разработке АИС, были рассмотрены объектно-функциональные подходы к проектированию АИС «Кругозор», построены логическая и физическая модель АИС, построена модель прецедентов, выбрана архитектура приложения.

В данной главе также дано описание информационного обеспечения для разработанной системы, обоснован выбор СУБД PostgreSQL для построения БД, и языка программирования, которым является C#. В конце главы были описаны модули приложения и функциональность.

В третьей главе произведен расчет затрат на разработку и амортизацию оборудования и с помощью данного расчета была посчитана экономическая эффективность.

Анализ экономической эффективности реализации проекта АИС «Кругозор» позволяет количественно оценить возможный экономический эффект: внедрение системы позволит снизить затраты на 25 %.

Сумма амортизационных отчислений за годовой период составляет 15900 рублей, окупаемость проекта АИС, которая составляет 9 месяцев и указывает на возврат основных средств. Себестоимость программы

составляет 38 787,62 рублей, а сумма затрат на внедрение АИС составит 46560 рублей.

Это также подтверждает рассчитанный годовой экономический эффект, который составляет 60000 рублей. Таким образом, приведенные расчеты экономического эффекта от внедрения системы позволяют сделать вывод о довольно высокой эффективности разработанного проекта АИС.

Проектирование и реализация информационной системы осуществлялись с применением современных методов разработки и проектирования.

Разработка всех моделей программного обеспечения UML и проектирование информационного программного обеспечения АИС были выполнены в среде разработки Visual Paradigm 13.2 [22]. Язык программирования высокого уровня C# в Visual Studio 2019 выбран для реализации приложения.

Таким образом, задачи, поставленные в начале работы над ВКР, были решены в полном объеме.

Список используемой литературы и используемых источников

Нормативно–правовые акты и стандарты

1. ГОСТ Р 51904–2002. Программное обеспечение встроенных систем. Общие требования к разработке и документированию.

Научная и методическая литература

2. Арлоу, Д. UML 2 и Унифицированный процесс. Практический объектно–ориентированный анализ и проектирование: пер. с англ. / Д. Арлоу, И. Нейштадт. – 2–е изд. – СПб.: Символ–Плюс, 2012. – 624 с.

3. Балдин, К.В. Информационные системы в экономике. / К.В. Балдин, В.Б. Уткин. – М.: Издательский центр Академия, 2005. – 288 с.

4. Вершинин, М. C# Enterprise Edition. Технологии проектирования и разработки. / М. Вершинин, Е. Иванова. – М.: ВHV, 2003 г. – 1088 с.

5. Инюшкина, О.Г. Проектирование информационных систем (на примере методов структурного анализа). Учебное пособие. Уральский федеральный университет им. Первого президента России Б.Н. Ельцина / О.Г. Инюшкина. – Екатеринбург: Форт–Диалог, 2014. – 240 с.

Электронные ресурсы

6. Что такое CRM-системы и как их правильно выбирать? // [Электронный ресурс] <https://habr.com/ru/company/trinion/blog/249633/>

7. Что такое CRM и для чего это нужно компании [Электронный ресурс] <https://www.klerk.ru/boss/articles/50404/>

8. Управление взаимоотношениями с клиентами [Электронный ресурс] https://nesrakonk.ru/customer_relation_management-2/

9. Глазная клиника «Кругозор» [Электронный ресурс] <https://moscow.krugozor-clinic.ru/>

10. Концептуальные, реляционные и «сущность-связь» модели; логическая структура моделей [Электронный ресурс] https://inep.sfedu.ru/wp-content/uploads/ehamt/learn/ombp_s/lecture_8.pdf

11. PostgreSQL: The world's most advanced open source database [Электронный ресурс] <https://www.postgresql.org/>
 12. PostgreSQL Интервью Вопросы [Электронный ресурс] <https://ru.education-wiki.com/6187298-postgresql-interview-questions>
 13. Внедрение зависимостей в ASP.NET Core Visual Studio [Электронный ресурс] <https://docs.microsoft.com/ru-ru/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-3.1>
 14. Разработка локального классификатора [Электронный ресурс] https://studwood.ru/1874845/informatika/razrabotka_lokalnogo_klassifikatora
 15. Экономические эффекты от внедрения CRM [Электронный ресурс] <https://www.cfin.ru/itm/crm/effects.shtml>
 16. Как рассчитать эффективность от внедрения CRM [Электронный ресурс] <https://web-regata.ru/blog/crm-efficiency/>
 17. Как оценить эффективность внедрения CRM-систем: [Электронный ресурс] <https://www.sekretariat.ru/article/210383-qqe-16-m5-kak-otsenit-effektivnost-vnedreniya-crm--sistem>
 18. Visual Studio [Электронный ресурс] <https://visualstudio.microsoft.com/ru/>
- Литература на иностранном языке*
19. IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12–1990
 20. IEEE Standard 830–1998, "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std., 1998
 21. RUP. Rational Unified Process, 2003, Version 2003.06.13
 22. UML 2.0. Superstructure specification. – Final adopted specification – ptc/03–08–02; 2003, 640 p.
 23. Visual Paradigm for UML 13. User's guide.: Software documentation –user's manual – 1485 pages

Приложение А

Программный код

Файл startup.cs

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using System.Collections.Generic;
using System.IO.Compression;
using Hangfire;
using Hangfire.MemoryStorage;
using Microsoft.AspNetCore.Http;
using Microsoft.EntityFrameworkCore;
using Microsoft.Net.Http.Headers;
using WebApp.Domain;
using WebApp.Models;
using WebMarkupMin.AspNetCore.Brotli;
using WebMarkupMin.AspNetCore.Common.Compressors;
using WebMarkupMin.Core;
using Internal.Configuration;
using Internal.Startup;
using WebMarkupMin.AspNetCore5;

namespace WebApp
{
    public class Startup
    {
        public static IWebHostEnvironment Env;
        public IConfiguration Configuration { get; }
    }
}
```

Продолжение Приложения А

Программный код

```
public Startup(IWebHostEnvironment _Env)
{
    Env = _Env;

    var _Builder = new ConfigurationBuilder()
        .SetBasePath(_Env.ContentRootPath)
        .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
        .AddJsonFile("appsettings.custom.json", optional: true,
reloadOnChange: true)
        .AddEnvironmentVariables(); // Производим подключение
конфигурационных файлов

    Configuration = _Builder.Build();

    var _OptionsBuilder = new DbContextOptionsBuilder<EfDbContext>();
    _OptionsBuilder.UseNpgsql(Configuration["ConnectionString"]); //
Подключаем БД

    using var _Db = new EfDbContext(_OptionsBuilder.Options);
    _Db.Database.Migrate(); // Инициализация БД
}

public void ConfigureServices(IServiceCollection _Services)
{
    _Services.AddHangfire(_Configuration => _Configuration
        .SetDataCompatibilityLevel(CompatibilityLevel.Version_170)
        .UseSimpleAssemblyNameTypeSerializer()
        .UseRecommendedSerializerSettings()
        .UseMemoryStorage());
}
```

Продолжение Приложения А

Программный код

```
_Services.AddHangfireServer(); // Производим подключение
планировщика задач

_Services.AddDbContext<EfDbContext>(_Options =>
    _Options.UseNpgsql(Configuration["ConnectionString"])
); // Производим подключение EF к PostgreSQL

_Services.ConfigureWritable<CustomConfig>(Configuration.GetSection("Main"),
"appsettings.custom.json"); // Делаем настройку файла конфигурации

_Services.AddResponseCaching(); // Производим подключение кеша

_Services.AddWebMarkupMin(_Options =>
{
    _Options.AllowMinificationInDevelopmentEnvironment = true;
    _Options.AllowCompressionInDevelopmentEnvironment = true;
})
.AddHtmlMinification(_Options =>
{
    HtmlMinificationSettings _Settings = _Options.MinificationSettings;
    _Settings.RemoveRedundantAttributes = true;
    _Settings.RemoveHttpProtocolFromAttributes = true;
    _Settings.RemoveHttpsProtocolFromAttributes = true;
})
.AddHttpCompression(_Options =>
{
    _Options.CompressorFactories = new List<ICompressorFactory>
    {
        new BrotliCompressorFactory(new BrotliCompressionSettings
        {
            Level = 1
```

Продолжение Приложения А

Программный код

```
    }),
    new DeflateCompressorFactory(new DeflateCompressionSettings
    {
        Level = CompressionLevel.Optimal
    }),
    new GZipCompressorFactory(new GZipCompressionSettings
    {
        Level = CompressionLevel.Optimal
    })
};
}); // Производим подключение и настройку минификации файлов

_Services.AddHttpsRedirection(_Options =>
{
    _Options.RedirectStatusCode =
StatusCodes.Status308PermanentRedirect;
    _Options.HttpsPort = 443;
}); // Делаем подключение и настройку SSL
_Services.AddDetection();
Configurator.ConfigureServices(_Services, Configuration);
if (Env.IsDevelopment())
_Services.AddControllersWithViews().AddRazorRuntimeCompilation(); //
Производим компиляцию в райнтайме
else _Services.AddControllersWithViews();
_Services.AddSession();
}

public void Configure(IApplicationBuilder _App, IWebHostEnvironment
_Env, IHostApplicationLifetime _ApplicationLifetime)
```


Продолжение Приложения А

Программный код

```
{
// Настраиваем отлов ошибок
if (_Env.IsDevelopment())
{
    _App.UseDeveloperExceptionPage();
}
else
{
    _App.UseExceptionHandler("/Home/Error");
    _App.UseStatusCodePagesWithReExecute("/Home/HandleError/{0}");
    _App.UseHsts();
}
_App.UseHttpsRedirection(); // Делаем редирект на Htpps
_App.UseStaticFiles(new StaticFileOptions
{
    OnPrepareResponse = _Ctx =>
    {
        const int durationInSeconds = 60 * 60 * 24 * 7;
        _Ctx.Context.Response.Headers[HeaderNames.CacheControl] =
            "public,max-age=" + durationInSeconds;
    }
}); // Делаем кеширование статичных файлов
_App.UseResponseCaching(); // Кеширование
_App.UseWebMarkupMin(); // Минификация файлов
_App.UseDetection();
Configurator.Configure(_App);
_App.UseRouting(); // Маршрутизация
_App.UseAuthorization(); // Авторизация
```

Продолжение Приложения А

Программный код

```
_App.UseEndpoints(_Endpoints =>
{
    _Endpoints.MapControllerRoute(null,
        "{controller=Admin}/{action=Index}/{_Id?}",
        new { controller = "Admin", action = "Index" }
    );
    _Endpoints.MapControllerRoute(null,
        "{_Domain}/{controller=Admin}/{action=Index}/{_Id?}",
        new { controller = "Admin", action = "Index" }
    );
}); // Делаем маршруты страниц
_App.UseSession();
}
}
}
```

"DM-

Фрагмент файла AdminController.cs

```
public ActionResult AddDoctor()
{
    ViewData["DoctorSpecs"] = _db.DoctorSpecs.AsNoTracking().ToArray();
    // Инициализируем данные для отображения по ключу
    ViewData["MedAreas"] = _db.MedAreas.AsNoTracking().ToArray();
    return View();
}
[HttpPost]
public async Task<ActionResult> AddDoctor(Doctor _Model, int _Cabinet,
int _From, int _To, bool[] _Days)
{
    var _Shedule = new DoctorShedule()
```

Продолжение Приложения А

Программный код

```
{
    Cabinet = _Cabinet,
    Days = _Days,
    From = _From,
    To = _To
};
await _db.AddModel(_Shedule); // Добавляем запись в базу
_Model.DoctorSheduleId = _Shedule.Id; // Получаем ключ добавленной
записи
await _db.AddModel(_Model); // Добавляем запись в базу
return RedirectToAction("Doctors");
}
public ActionResult EditDoctor(int _Id)
{
    var _Doctor = _db.Doctors.AsNoTracking().FirstOrDefault(_X => _X.Id
== _Id); // Пробуем получить запись по ключу
    if (_Doctor == null) return new NotFoundResult(); // Вывод если в базе
нет записи с запрошенным ключом
    ViewData["DoctorSpecs"] = _db.DoctorSpecs.AsNoTracking().ToArray();
    ViewData["MedAreas"] = _db.MedAreas.AsNoTracking().ToArray();
    return View(_Doctor);
}
[HttpPost]
public async Task<ActionResult> EditDoctor(Doctor _Model)
{
    await _db.EditModel(_Model); // Сохраняем изменения
```

Продолжение Приложения А

Программный код

```
return RedirectToAction("Doctors");
}

public async Task<ActionResult> DelDoctor(int _Id)
{
    var _Model = _db.Doctors.First(_X => _X.Id == _Id);
    var _Model2 = _db.DoctorShedules.First(_X => _X.Id ==
_Model.DoctorSheduleId);

    await _db.DeleteModel(_Model);
    await _db.DeleteModel(_Model2);

    return RedirectToAction("Doctors");
}
```

Приложение Б

Декомпозиция процесса администрирования приёма пациентов

