

## Аннотация на дипломный проект

АСОИ «Медицинская информационная система обработки изображений».  
Нейронная сеть и серверная часть

### Структура и объём проекта

Дипломный проект состоит из пояснительной записки на 73 страниц и 7 листов графической части. Пояснительная записка состоит из задания, аннотации, введения, семи разделов, заключения и списка литературы. Проект содержит 31 иллюстрацию и 21 таблицу. Список литературы включает 16 наименований.

### Содержание проекта

Во введении указаны цель выполнения проекта и актуальность темы.

В первом разделе проводится анализ предметной области.

Во втором разделе проводится проектирование структуры информационной системы.

В третьем разделе описывается политика информационной безопасности.

В четвертом разделе описывается программная реализация системы.

В пятом разделе рассматриваются организационно-экономические вопросы, производится технико-экономическое обоснование целесообразности разработки проекта.

В шестом разделе осуществляется рассмотрение вопросов охраны труда.

В седьмом разделе описываются вопросы энерго- и ресурсосбережения.

В заключении производится анализ степени выполнения требований к проектируемой системе.

					301.1-53 01 02.10031777.006 ПЗ			
Изм	Лист	№ докум.	Подпись	Дата				
Разраб.		Пантелеев			Многопользовательская информационная система медицинских изображений. Разработка нейронной сети и серверной части	Лит	Лист	Листов
Провер.		Выговская					4	73
						БРУ гр. АСОИ-201		
Н. Контр.		Плиско						
Утв.		Якимов						

## Содержание

Введение.....	7
1 Анализ предметной области.....	8
1.1 Описание объекта автоматизации .....	8
1.2 Анализ существующей ИС.....	9
1.3 Краткое описание, классификация и анализ существующих ИС, аналогичных разрабатываемой. ....	11
1.4 Постановка задачи на дипломное проектирование .....	12
2 Проектирование структуры информационной системы .....	15
2.1 Организационно-экономическая сущность задачи .....	15
2.2 Структура ИС .....	16
2.3 Информационное обеспечение .....	18
2.4 Техническое и системное программное обеспечение .....	21
2.5 Эргономическое обеспечение .....	22
3 Политика информационной безопасности .....	23
3.1 Цель, принципы и задачи защиты информации в АСОИ .....	23
3.2 Методы и средства обеспечения защиты информационных ресурсов. ....	23
3.3 Средства защиты информации и информационных ресурсов .....	24
4 Программная реализация информационной системы .....	25
4.1 Выбор программных средств реализации автоматизированной системы.....	25
4.2 Реализация базы данных.....	26
4.3 Реализация программного интерфейса серверной части .....	27
4.4 Реализация архитектуры нейронной сети.....	34
4.5 Реализация модуля анализа ИИ .....	37
4.6 Руководство пользователя.....	39
5 Расчет экономической эффективности технических решений .....	42
5.1 Общая постановка к технико-экономическому обоснованию .....	42
5.2 Расчет трудоемкости (производительности) .....	45
5.3 Расчет единовременных затрат.....	45
5.4 Расчет годовых текущих издержек .....	52
5.5 Расчет показателей экономической эффективности .....	58
5.6 Организация внедрения системы.....	59
5.7 Заключение по разделу .....	60

6 Охрана труда.....	61
6.1 Система управления охраной труда в учреждении здравоохранения «Могилевская областная клиническая больница».....	61
6.2 Анализ выполнения мероприятий по обеспечению пожарной безопасности в учреждении здравоохранения «Могилевская областная клиническая больница» .....	64
6.3 Выводы и предложения .....	67
7 Энерго- и ресурсосбережение.....	69
7.1 Энергоэффективные технологии в здравоохранении .....	69
Заключение .....	71

## Введение

В медицинской практике наблюдается необходимость в эффективном анализе рентген-снимков для точной диагностики и лечения пациентов. Однако многие медицинские учреждения еще не осознали преимущества автоматизированных систем анализа рентгенограмм. Внедрение системы искусственного интеллекта (ИИ) для анализа рентген-изображений становится все более актуальным, поскольку это позволяет оптимизировать процессы диагностики, повышая точность и скорость анализа.

Кроме того, исследование бизнес-процессов при работе с медицинскими изображениями в учреждении здравоохранения «Могилёвская областная клиническая больница» показало неэффективность и возможность доработки и дополнительной автоматизации в данной сфере деятельности.

Такие внедрения позволят добиться следующих положительных эффектов:

- повышение эффективности диагностики и управления лечением пациентов, что приведет к улучшению качества медицинского обслуживания и удовлетворенности пациентов;
- снижение времени, необходимого для обучения медицинского персонала и упрощение процесса анализа рентген-снимков для врачей и медицинских специалистов.

На основании вышеизложенного, принято решение о разработке АСОИ «Медицинская информационная система обработки изображений» в формате веб-приложения. Для реализации веб-приложения были написаны два серверных приложения на ASP.NET Core и Flask с использованием языков программирования C# и Python соответственно. Также создана база данных при помощи свободной объектно-реляционной системы управления PostgreSQL. Для клиентского приложения использовался язык JavaScript и веб-фреймворк React. Для создания модуля искусственного интеллекта была разработана и обучена нейронная сеть с использованием TensorFlow и Keras.

# 1 Анализ предметной области

## 1.1 Описание объекта автоматизации

Объектом автоматизации для дипломного проектирования по выбранной теме выделяется процесс анализа рентген-снимков спины для диагностики и составления предварительного заключения. В УЗ «Могилевская областная клиническая больница».

Анализом рентген-снимков занимается врач-рентгенолог. Его рабочее место включает в себя работу с рентгеновскими снимками и составление диагностических заключений на их основе. Врач анализирует рентген-снимки, выявляет патологии и формулирует заключения, которые затем используются для постановки диагноза и определения дальнейшего лечения. В настоящее время этот процесс осуществляется с использованием двух различных информационных систем: одной для анализа снимков и другой для составления и хранения заключений. Это создает дополнительные сложности и увеличивает время, необходимое для проведения анализа и формирования заключения. На рисунке 1.1 изображена контекстная диаграмма описанных процессов.

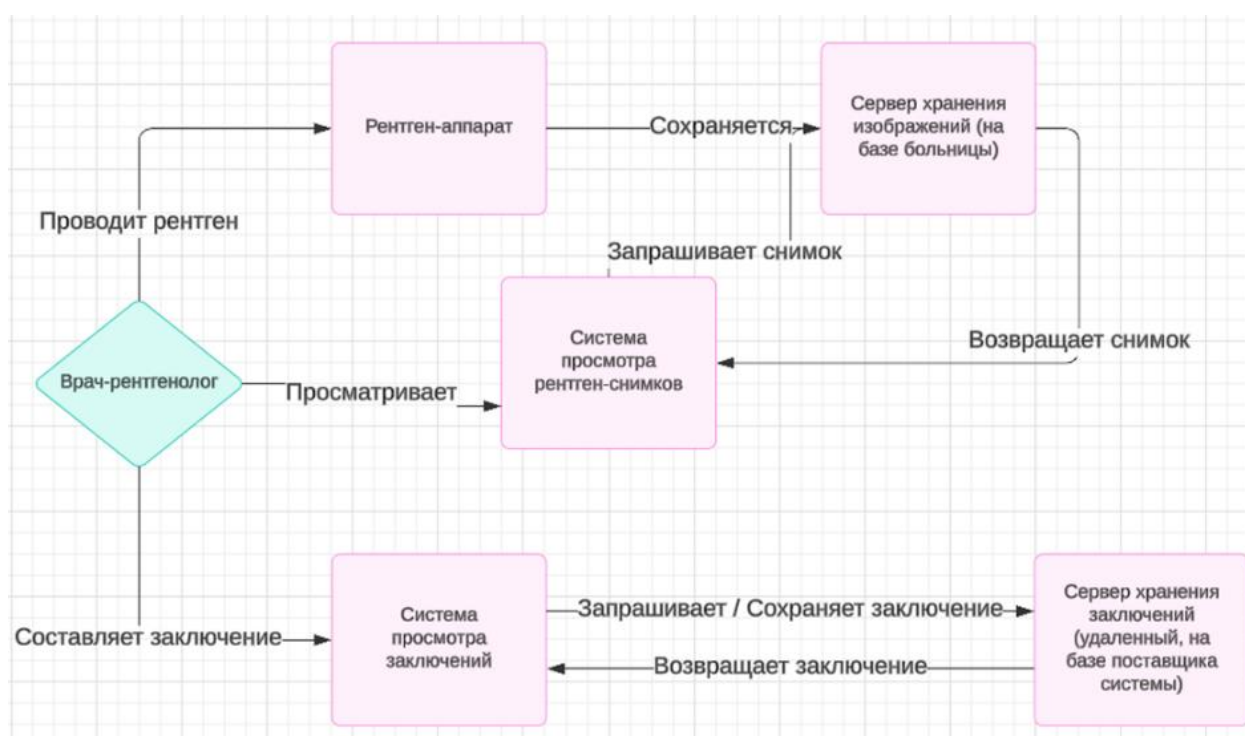


Рисунок 1.1 Контекстная диаграмма процесса анализа и составления заключений врачом-рентгенологом

Объект автоматизации можно увидеть на диаграмме – это звено между врачом и системой просмотра заключений: предлагается, что процесс анализа снимка составления заключения будет проводиться искусственной нейронной сетью.

Также можно увидеть отсутствие связи между системой просмотра снимков и просмотра заключений (в случае УЗ «Могилевская областная клиническая

больница» это Radiant Dicom Viewer и система «Лекарь» соответственно), именно это наблюдалось на базе больницы: снимок и заключение хранятся отдельно, доступ к ним осуществляется переключением между программами, например смотря ФИО пациента и дату приема находился подходящий по времени снимок. Такой процесс тратит много лишнего времени и труда, поэтому в объект дополнительно входит обеспечить объединение систем, возможную интеграцию с системой «Лекарь», более доступный и менее трудозатратный процесс доступа (хранения, поиска) к данным.

Кроме того, современные разработки в области искусственного интеллекта (ИИ) открывают новые возможности для медицинской диагностики, включая анализ рентгеновских снимков спины. ИИ может существенно повысить точность и скорость диагностики, автоматически выявляя патологии на изображениях, что особенно полезно при выявлении заболеваний позвоночника. Применение ИИ позволяет разгрузить врачей от рутинных задач, сосредоточив их внимание на более сложных случаях, а также способствует снижению вероятности ошибок, связанных с человеческим фактором.

## 1.2 Анализ существующей ИС

На данный момент в учреждении здравоохранения (УЗ) «Могилевская областная клиническая больница» используется специализированная система «Лекарь», внедренная в 2023 году. Ранее в больнице использовалась система, разработанная компанией 4Д в 2002 году, но она не получала крупных обновлений с 2010 года, что привело к её функциональному и техническому устареванию. «Лекарь» является более новой и технически лучше проработанной системой, позволяя работать с данными о пациентах, заключениях и др., однако функционально не отличается от ранее используемой 4D. Система сохраняет невозможность работы со снимками и интеграции внутреннего хранилища больницы, а хранилище заключений организовано на серверах организации-разработчика. Кроме того, система имеет закрытый код, что исключает возможность её модернизации.

В больнице также применяется программное обеспечение Radiant DICOM Viewer для работы с рентгеновскими снимками, однако оно никак не связано с системой «Лекарь». Хранилище снимков организовано на серверах больницы, а хранилище заключений находится на серверах системы «Лекарь». Текущая инфраструктура представлена в виде хранилища на серверах больницы. Сервера представлены несколькими виртуальными машинами с развернутыми на них серверами Orthanc. Получение снимков из хранилища организовано за счет файловой базы данных SQLite.

Схема информационной системы больницы приведена на рисунке 2.2.

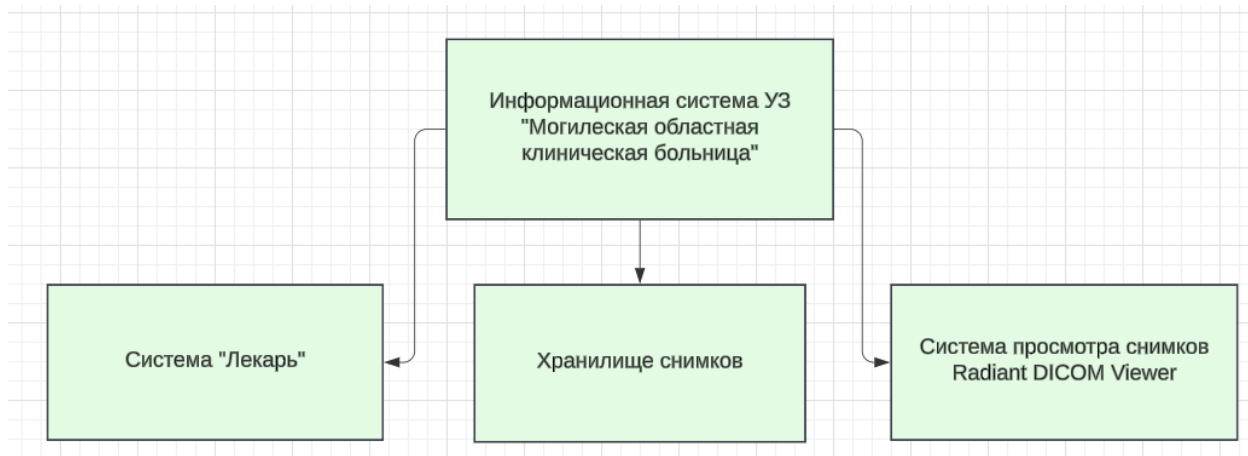


Рисунок 1.2 Схема информационной системы больницы

Таким образом, ссылаясь на описанную систему, а также описание объекта автоматизации, можно выделить следующие недостатки существующей ИС учреждения здравоохранения (УЗ) «Могилевская областная клиническая больница»:

- не позволяет работать со снимками и заключениями в рамках одной подсистемы;
- невозможность модификации;
- нет системы хранения заключений на базе больницы;
- есть вероятность потери результатов заключений;
- невозможность разделения хранилища и программы для просмотра результатов заключений.

Из-за отсутствия возможности модернизации существующей информационной системы, чтобы устранить недостатки существующей МИС, принято решение разработать новую систему, учитывающую вышеприведённые требования.

С вводом новой системы получится объединить системы просмотра и обработки заключений и снимков. Такая система позволит решить недостатки по необходимости наличия нескольких невзаимодействующих систем, а также иметь открытый код, давая возможность ее расширять и модифицировать. Кроме того, вводя новую систему откроется возможность организовать хранилище заключений на базе серверов больницы. На рисунке 1.3 приведена контекстная диаграмма процесса анализа и составления заключений с вводом новой системы.

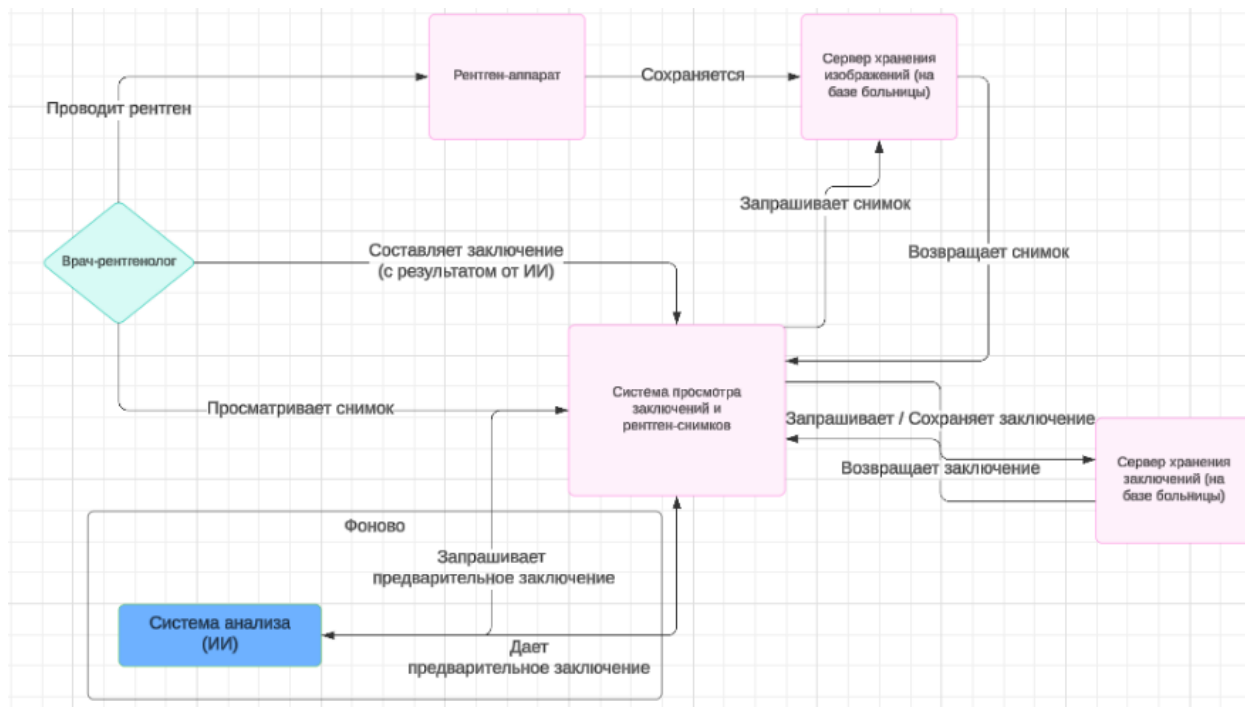


Рисунок 1.3 Контекстная диаграмма процесса анализа и составления заключений с вводом новой системы

Дополнительно предлагается ввести модуль анализа рентген-заключений с использованием искусственного интеллекта. Такой модуль возможен только с вводом новой системы. На рисунке 1.3 приведена контекстная диаграмма процесса анализа и составления заключений с вводом новой системы.

### 1.3 Краткое описание, классификация и анализ существующих ИС, аналогичных разрабатываемой.

На данный момент существуют различные виды программного обеспечения для медицинских учреждений, позволяющие работникам взаимодействовать с записями, заключениями и снимками пациентов.

**Archimed+ (Россия).** Archimed+ разработан для применения в частных медицинских центрах и стоматологических клиниках. Программа поддерживает операционные системы Microsoft Windows и является коммерческим продуктом. Основные функции программного обеспечения включают учет пациентов, ведение электронных медицинских карт, хранение и обработку рентгеновских снимков, а также создание и хранение медицинских заключений. Archimed+ позволяет вести учет пациентов, планировать приемы, контролировать звонки и управлять бизнес-процессами клиники. А также автоматизация регистратуры, IP-телефония, CRM-система, интеграция с внешними лабораториями, телемедицина и онлайн-платежи. Рекомендуемые параметры для Windows: разрядность 64-bit, процессор не хуже Intel Core i3, RAM 4 GB, 10 GB свободного дискового пространства.

**Medesk (Россия).** Medesk разработан для применения в частных медицинских центрах и поддерживает операционные системы Microsoft Windows и macOS. Программа является облачным сервисом и коммерческим продуктом. Основные функции программного обеспечения включают учет пациентов, ведение



электронных медицинских карт, хранение и обработку рентгеновских снимков, а также создание и хранение медицинских заключений. Medesk позволяет вести учет пациентов, проводить видеоконсультации, управлять расписанием и анализировать эффективность работы клиники. А также онлайн-запись, автоматизация регистратуры, интеграция с лабораториями, управление запасами, телемедицина и мобильное приложение для пациентов. Рекомендуемые параметры для Windows: разрядность 64-bit, процессор не хуже Intel Core i3, RAM 4 GB, 10 GB свободного дискового пространства.

Sycret (Россия). Sycret разработан для применения в частных медицинских центрах и поддерживает операционные системы Microsoft Windows. Программа является коммерческим продуктом. Основные функции программного обеспечения включают учет пациентов, ведение электронных медицинских карт, хранение и обработку рентгеновских снимков, а также создание и хранение медицинских заключений. Sycret позволяет вести учет пациентов, управлять расписанием, оформлять документы и контролировать выполнение назначений. А также автоматизация управления медицинскими процессами, учет лекарственных препаратов и подготовка отчетов. Рекомендуемые параметры для Windows: разрядность 64-bit, процессор не хуже Intel Core i3, RAM 4 GB, 10 GB свободного дискового пространства.

Данные технические средства представляют широкий набор возможностей, однако не имеют интеграции с ИИ (искусственным интеллектом). Его внедрение в создаваемую автоматизированную систему, а также интеграция с огромным набором снимков больницы для первичного обучения и ежедневное пополнение данного набора для дальнейшего переобучения системы дает перспективы в развитии как здравоохранения, так и разработки искусственного интеллекта как сферы деятельности. Данная причина легла в основу разработки нашей автоматизированной системы.

#### 1.4 Постановка задачи на дипломное проектирование

На основании приведенных выше описаний объекта автоматизации и существующей ИС ставятся следующие задачи на дипломное проектирование в рамках темы «Многопользовательская информационная система медицинских изображений. Разработка нейронной сети и серверной части»:

- разработать архитектуру искусственной нейронной сети (ИНС), способной составлять заключения по деформациям позвоночника (сколиоз) на основе рентген-снимка;
- разработать серверную часть для анализа, хранения и просмотра заключений и снимков, включая базу данных, соответствующие HTTP-эндпоинты (комбинация HTTP-метода и URL);
- разработать систему авторизации;
- разработать модуль для получения анализа от искусственной нейронной сети.

Задачи по хранению заключений и интеграции их в существующие системы больницы (или создание своих) добавит такие трудности, как необходимость составления договоров научно-технического сотрудника как с больницей, так и с разработчиками «Лекарь», а также необходимость регистрации системы в регистре информационных систем, информационных ресурсов и программного обеспечения, соответствующего требованиям HL7 FHIR. В связи с этим цели по автоматизации хранения, поиска и другого взаимодействия с данными пациентов в рамках проекта выделяются как проверка концепции.

Проверка концепции (от англ. Proof of concept, POC) — демонстрация практической осуществимости какого-либо метода, идеи, технологии, реализуемости с целью доказательства факта, что метод, идея или технология работают. В рамках демонстрации строится небольшой прототип, опытный образец, математическая или компьютерная модель, не обязательно являющиеся полноценными продуктами, но подтверждающие принципиальную возможность создания таковых.

Таким образом, доказав состоятельность идеи проекта, его потенциал, в дальнейшем можно будет вести работы по его внедрению и интеграции в бизнес-процессы как Могилевской областной больницы, так и других учреждений здравоохранения.

Кроме того, предъявляется ряд требований к информационной системе и операционной системе. Требования, предъявляемые к информационной системе:

- код программного продукта должен быть написан таким образом, чтобы дополнения и изменения могли вноситься без нарушения целостности системы. Расширяемость достигается за счет модульной структуры программы, при которой программа строится из набора отдельных модулей, взаимодействующих только через функциональный интерфейс;

- код программы должен легко переноситься с одной операционной системы на другую, с аппаратной платформы (которые различаются не только типом процессора, но и способом организации всей аппаратуры компьютера) одного типа на аппаратную платформу другого типа. Поэтому, программный продукт должен удовлетворять требованию переносимости;

- программный продукт должен обладать настолько хорошим быстродействием и временем реакции, на сколько это позволяет операционная система и аппаратная платформа. На производительность программы влияет ряд факторов, среди которых основными являются архитектура программы, многообразие функций, качество программирования кода, возможность работы с большими объемами данных;

- достоверность информации зависит как от пользователя, который проверяет вводимую информацию на соответствие типам данных, так и от самой программы, которая обеспечивает обработку введенной информации и хранение результатов на носителях данных.

Требования, предъявляемые к системному программному обеспечению:

- выполнение ею основных функций эффективного управления ресурсами и обеспечение удобного интерфейса для пользователя. Современная операционная система, как правило, поддерживает мультипрограммную обработку,

многооконный графический интерфейс пользователя, а также выполнять многие другие необходимые функции и услуги;

– переносимость. Код ОС должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы одного типа на платформу другого типа. Данное свойство ОС называют также кроссплатформенностью;

– совместимость. Существует несколько популярных систем (разновидности UNIX, OS/2, Windows NT), для которых наработана широкая номенклатура приложений. Некоторые из них пользуются широкой популярностью. Поэтому для пользователя, переходящего с одной ОС на другую, очень привлекательна возможность запуска в новой операционной системе привычного приложения;

– расширяемость. В то время как аппаратная часть компьютера устаревает за несколько лет, полезная жизнь операционных систем может измеряться десятилетиями. Поэтому операционные системы всегда изменяются эволюционно, и эти изменения более значимы, чем изменения аппаратных средств. Изменения ОС обычно заключаются в приобретении ею новых свойств, например поддержке новых типов внешних устройств или сетевых технологий. Расширяемость достигается за счет модульной структуры ОС.

## 2 Проектирование структуры информационной системы

### 2.1 Организационно-экономическая сущность задачи

Разработка информационной системы – управляемый процесс, состоящий из нескольких последовательных этапов, каждый из которых является в определённой степени независимым.

Разрабатываемая автоматизированная система нацелена на хранение и обработку результатов заключений, что и заложено в её функционал. Система работает с использованием таких технологий как: PostgreSQL, C#, ASP.NET, Python, Flask.

На рисунке 3.1 представлена схема вариантов использования автоматизированной системы.

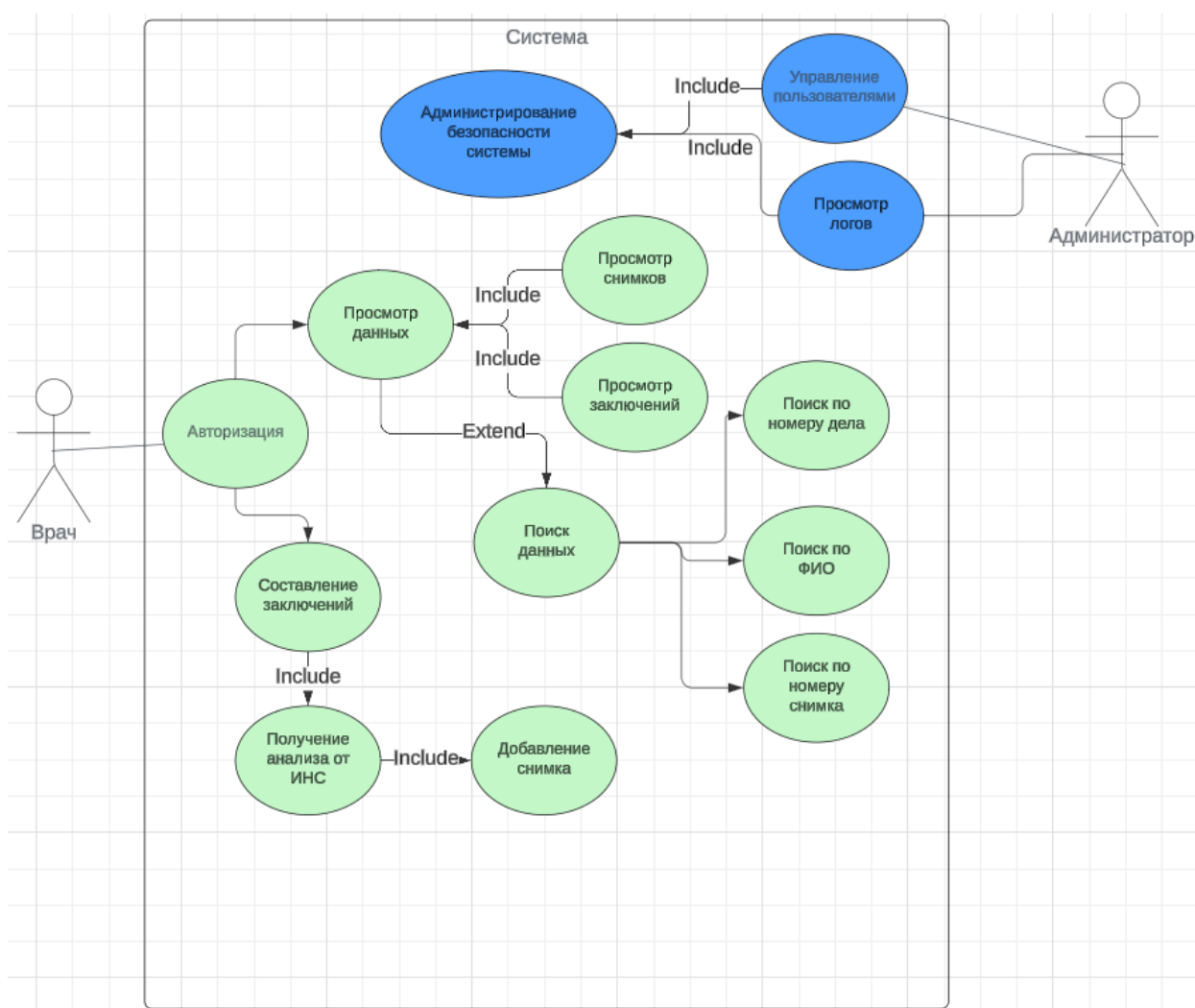


Рисунок 2.1 – Схема вариантов использования автоматизированной системы

Диаграмма вариантов использования представлена в графической части (301.1-53 01 02.10031777.006.001 ПД).

Система выполняет следующие функции:

– просмотр, добавление, изменение и удаление данных по заключениям;

Изм.	Лист	№ докум.	Подп.	Дата

301.1-53 01 02.10031777.006 ПЗ

Лист

15

- просмотр, добавление, изменение и удаление данных по снимкам;
- просмотр, добавление, изменение и удаление данных по пациентам;
- просмотр, добавление, изменение и удаление данных по врачам;
- поиск данных о заключениях либо пациентах;
- получение предварительного заключения от нейронной сети;
- авторизация и аутентификация в системе.

## 2.2 Структура ИС

В разрабатываемой информационной системе выделяются 4 модуля: модуль REST API, модуль анализа ИИ, реляционная база данных, файловое хранилище снимков. Схема модулей информационной системы представлена на рисунке 2.2.

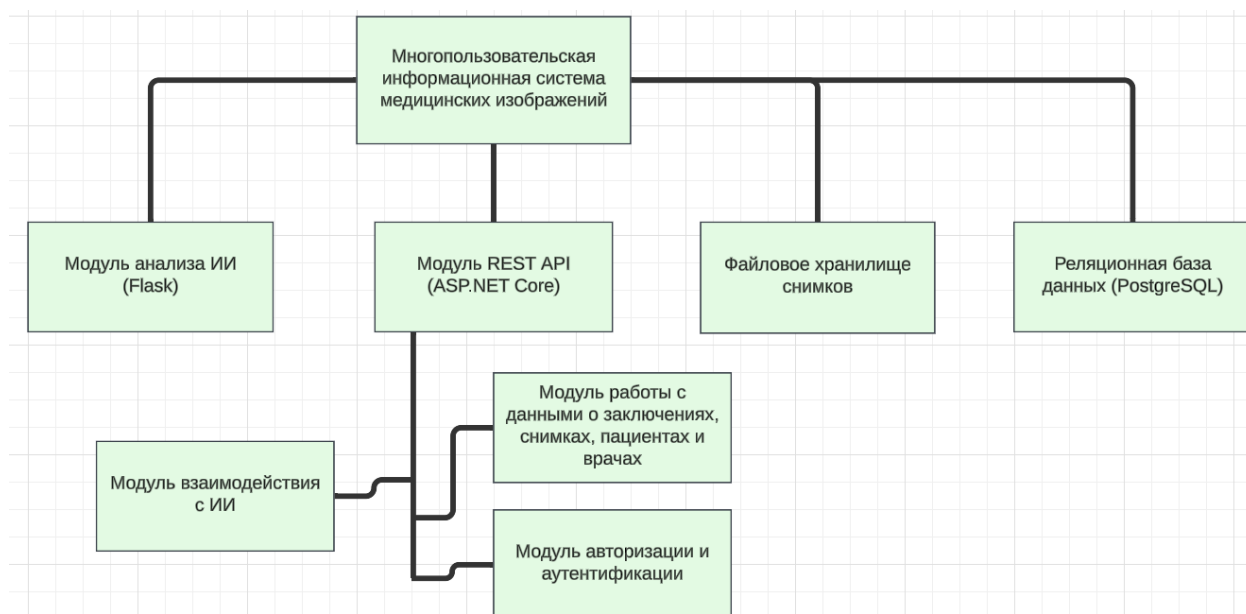


Рисунок 2.2 Схема модулей информационной системы.

Модуль REST API представляет собой Web API (англ. Application Programming Interface — программный интерфейс приложения, набор способов и правил, по которым различные программы общаются между собой и обмениваются данными), который обеспечивает функции по управлению и взаимодействию с данными о заключениях, снимках, пациентах. Для такого вида API общепринят и популярен подход REST. REST — архитектурный подход разработки Web API, основывающийся на 6 принципах:

- клиент-серверная модель: разделение клиента и сервера для упрощения разработки и масштабирования;
- отсутствие состояния (statelessness): каждый запрос от клиента к серверу должен содержать всю необходимую информацию для обработки запроса;
- кэширование (cacheability): ответы сервера могут быть кэшированы клиентом для повышения производительности;
- единообразие интерфейса (uniform interface): стандартизация взаимодействия между клиентом и сервером;

- многоуровневая система (layered system): возможность использования промежуточных серверов для улучшения масштабируемости и безопасности;
- код по требованию (code on demand): возможность передачи исполняемого кода от сервера к клиенту для расширения функциональности (необязательный принцип).

Программные интерфейсы, следующие вышеописанным принципам, называются RESTful. Для разработки такого интерфейса был выбран фреймворк (фреймворк — это набор взаимосвязанных классов и объектов, которые предоставляют архитектурный каркас для создания программного обеспечения. Он определяет структуру приложения и предоставляет готовые решения для типовых задач, что позволяет разработчикам сосредоточиться на специфических аспектах проекта) ASP.NET Core на основе платформы .NET 8.0 с использованием языка C# 12.

Модуль REST API должен включать в себя модуль работы с данными и модуль авторизации и аутентификации, работающие с файловой и реляционной базами данных, а также функционал для взаимодействия с модулем анализа от ИИ.

Модуль анализа ИИ является отдельной единицей, разрабатываемой с помощью фреймворка Flask на языке Python. Это объясняется спецификой написания нейронной сети: она создается с использованием библиотек Tensorflow и Keras. Tensorflow – это популярная библиотека машинного обучения с открытым исходным кодом, разработанная Google Brain. Она предоставляет различные инструменты и функции для создания, обучения и развертывания моделей машинного обучения. TensorFlow поддерживает как низкоуровневое программирование для детального контроля над процессом обучения, так и высокоуровневые API для более простого и быстрого создания моделей. Keras – это высокоуровневый API для создания и обучения нейронных сетей, который работает поверх различных библиотек глубокого обучения, включая TensorFlow. Keras была разработана для того, чтобы сделать машинное обучение доступным и легким в использовании, что позволяет быстро создавать прототипы моделей. TensorFlow и Keras могут быть использованы вместе для создания мощных и гибких моделей глубокого обучения. Keras предоставляет удобный интерфейс для быстрого создания и обучения моделей, в то время как TensorFlow обеспечивает мощную и гибкую основу для выполнения вычислений и оптимизации. Для сохранения модели, созданной с помощью Keras, используется формат HDF5 (Hierarchical Data Format), который позволяет сохранить структуру модели, ее веса и конфигурацию обучения. Сохраненную модель можно затем загрузить и использовать только в Python, причем первоначальная загрузка занимает от двух до десяти секунд. Такое время загрузки не отвечает поставленным требованиям, из-за чего для создания модуля анализа необходимо разработать программу, загружающую модель один раз и далее доступную в любое время. Программа также должна быть способна принимать изображение для анализа по протоколу или подгружая из локальной системы. В соответствии с такими требованиями было принято решение разработать еще один Web API используя Flask.

Flask был выбран для создания API, которое загружает модель Keras и принимает запросы на прогнозирование Flask – это легкий и гибкий веб-фреймворк

на Python, который позволяет быстро и легко создавать веб-сервисы. Он подходит для создания простых API, поскольку имеет минимальные накладные расходы и предоставляет необходимые инструменты для обработки HTTP-запросов. Используя Flask, можно легко интегрировать модель Keras, загружать её и обрабатывать запросы на прогнозирование в реальном времени, обеспечивая быстрый и эффективный отклик на запросы. Кроме того, Flask имеет хорошую документацию и сообщество, что облегчает разработку и поддержку приложения.

Файловое хранилище уже организовано на серверах больницы. Оно распределено на виртуальных машинах с использованием серверов Orthanc и базы данных SQLite для хранения и управления медицинскими изображениями. Orthanc – это сервер для управления медицинскими изображениями (например, DICOM-файлами), который обеспечивает возможности хранения, поиска и передачи изображений. Виртуальные машины позволяют легко масштабировать ресурсы и обеспечивают изоляцию и безопасность данных. Использование SQLite в качестве базы данных обеспечивает легкость и простоту настройки, так как это встраиваемая база данных, не требующая отдельного сервера. SQLite отлично подходит для небольших и средних объемов данных, обеспечивая быстрый доступ и минимальные накладные расходы на управление. Такая архитектура позволяет эффективно организовать централизованное хранилище медицинских изображений, обеспечивая высокую доступность и надежность данных, а также гибкость в настройке и развертывании системы.

Для хранения данных о пациентах, врачах, заключениях и др. необходима реляционная база данных. В качестве таковой выбрана PostgreSQL. Выбор обусловлен ее надежностью и мощными возможностями для работы с большими объемами данных. PostgreSQL поддерживает сложные запросы и транзакции, что важно для обеспечения целостности и консистентности медицинской информации. Кроме того, она обладает высокой производительностью и масштабируемостью, что позволяет эффективно обрабатывать и хранить большие объемы данных. Ее поддержка расширений и соответствие стандартам SQL делают PostgreSQL удобным и гибким инструментом для медицинских приложений, требующих сложных операций с данными.

## 2.3 Информационное обеспечение

В системе предполагается получение на вход данных о врачах (сотрудниках УЗ), пациентах, рентген-снимках. Данные о заключениях как могут поступать на вход, так и формироваться внутри системы. Внутри системы также формируется предварительное заключение от ИИ. На выход система дает заключения, отчет о заболеваемости. На рисунке 2.3 представлена схема потоков данных системы.

### Многопользовательская информационная система медицинских изображений

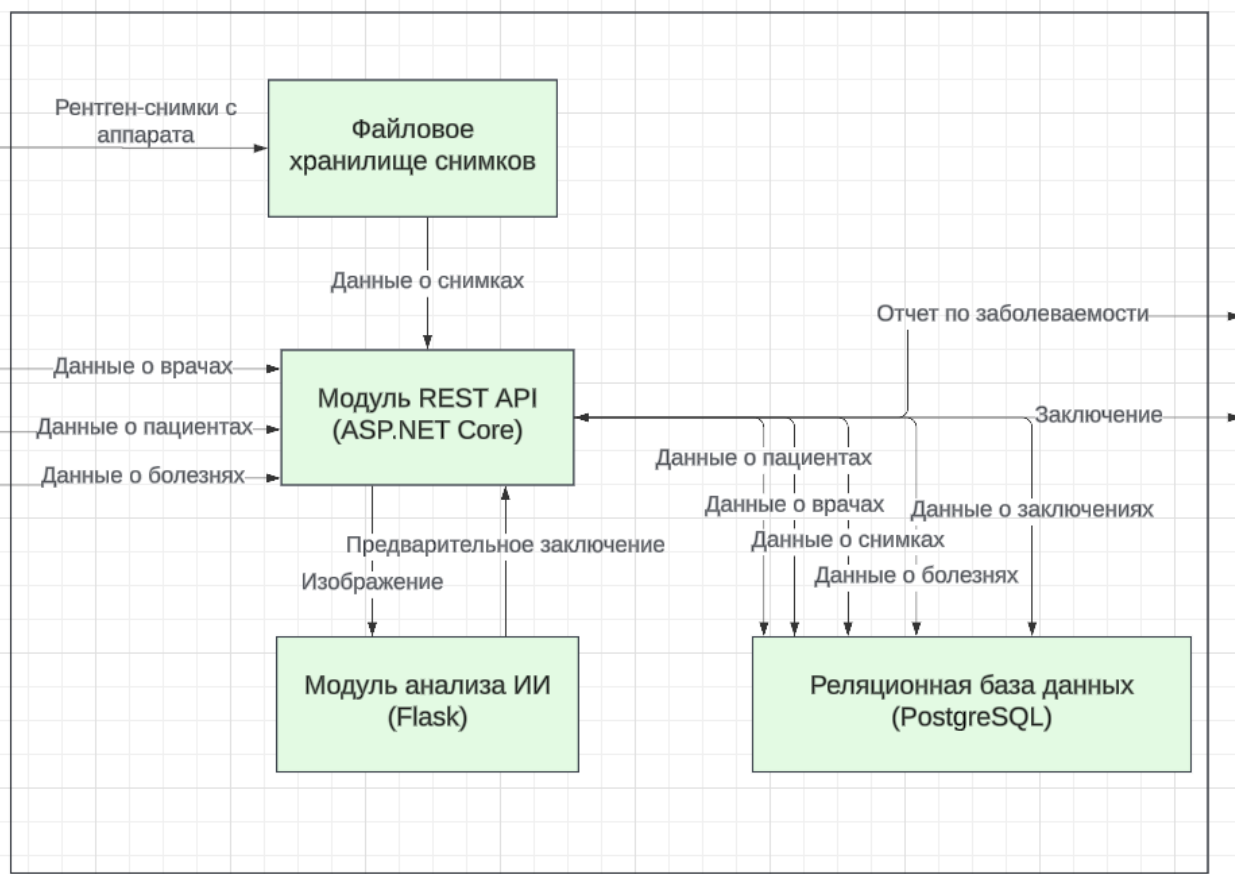


Рисунок 2.3 Схема потоков данных

Для системы проектируется как часть разрабатываемой ИС реляционная база данных. В таблице 2.1 приводится перечень проектируемых таблиц.

Таблица 2.1 Таблицы базы данных для информационной системы

Название	Содержимое	Обновление
Diseases	Информация о существующих болезнях	Врачом по мере необходимости
Doctors	Информация о врачах больницы	Администратором при необходимости доступа в систему нового врача или обновления данных уже добавленного
Examinations	Информация о заключениях	Врачом при добавлении заключения
Patients	Информация о пациентах	Врачом при добавлении заключения
Xrays	Информация о снимках	Врачом при добавлении заключения
ExaminationXrays	Для связи М:М заключений и снимков	При добавлении заключения либо снимка

Далее приведены описания структур таблиц.

В таблицах 2.2-2.7 приведены описания структур предлагаемых таблиц.



Таблица 2.2 Структура таблицы «Diseases»

Поле	Тип данных	Описание	Источник данных	Периодичность обновления
DiseaseId	Integer	Идентификатор	База данных, автоматически	При добавлении болезни в систему
Class	Varchar (50)	Класс заболевания	Международная классификация болезней (МКБ-10)	
Name	Varchar (200)	Название заболевания		

Таблица 2.3 Структура таблицы «Doctors»

Поле	Тип данных	Описание	Источник данных	Периодичность обновления
DoctorId	Integer	Идентификатор	База данных, автоматически	При добавлении сотрудника в систему
FullName	Varchar (100)	ФИО врача	Перечень сотрудников	
Position	Varchar (50)	Должность		
WorkPlace	Text	Место работы		
Login	Text	Логин для входа в систему	Администратор	
Password	Text	Пароль для входа в систему		
isAdmin	Boolean	Метка администратора		

Таблица 2.4 Структура таблицы «Patients»

Поле	Тип данных	Описание	Источник данных	Периодичность обновления
PatientId	Integer	Идентификатор	База данных, автоматически	При добавлении пациента в систему
FullName	Varchar (100)	ФИО пациента	Документ пациента	
BirthDate	Text	Дата рождения		
WorkPlace	Text	Место работы		

Таблица 2.5 Структура таблицы «ExaminationXrays»

Поле	Тип данных	Описание	Источник данных	Периодичность обновления
ExaminationId	Integer	Идентификатор заключения	Таблица заключений	При добавлении снимка или заключения
XrayId	Integer	Идентификатор снимка	Таблица снимков	

Таблица 2.6 Структура таблицы «Xrays»

Поле	Тип данных	Описание	Источник данных	Периодичность обновления
XrayId	Integer	Идентификатор	База данных, автоматически	При добавлении снимка в систему
XrayCode	Varchar (50)	Код снимка	Рентген-аппарат	
XrayFileIndex	Varchar (300)	Полное название файла	Файловое хранилище	

Таблица 2.7 Структура таблицы «Examinations»

Поле	Тип данных	Описание	Источник данных	Периодичность обновления
ExaminationId	Integer	Идентификатор	База данных, автоматически	При добавлении заключения в систему
PatientId	Integer	Код пациента	Таблица пациентов	
DoctorId	Integer	Код доктора	Таблица докторов	
DiseaseId	Integer	Код болезни	Таблица болезней	
Conclusion	Varchar(1000)	Диагноз	Врач	
ExaminationDate	DateTime	Дата заключения	Время, в которое сделано заключение	

Для таблиц разработаны связи. Связи описаны в таблице 2.8.

Таблица 2.8 Описание связей между таблицами

Название таблицы 1	Название таблицы 2	Описание
Examinations	Diseases	Связь один-ко-многим: одна болезнь может быть во многих заключениях
Examinations	Doctors	Связь один-ко-многим: у одного доктора может быть несколько заключений
Examinations	Patients	Связь один-ко-многим: у одного пациента может быть несколько заключений
Examinations	ExaminationXrays	Связь многие-ко-многим: один снимок может быть во многих заключениях, в одном заключении может быть несколько снимков
Xrays	ExaminationXrays	

В качестве выходных данных система позволяет сформировать отчет о заболеваемости с группировкой заключений по болезням за выбранный месяц года.

## 2.4 Техническое и системное программное обеспечение

Система представляет собой web-платформу с клиент-серверной архитектурой.

Для того, чтобы исправного функционирования серверной части, необходим персональный компьютер с следующими характеристиками:

Рекомендуемые системные требования:

- процессор: Intel Core i3 2600 1.2GHz
- оперативная память: 8 ГБ (для Windows 10, MacOS)
- видеокарта: GeForce GT 1030 (с 3 ГБ видеопамяти)
- интернет: 1 Мб/с

Кроме того, необходимо установить программные средства .NET 8, Python 3.12.

Клиентское приложение работает на любой ОС, на которой установлен современный браузер, поддерживающий JavaScript, наподобие Google Chrome или Microsoft Edge.

## 2.5 Эргономическое обеспечение

Разработка серверной части информационной системы медицинских изображений требует учета эргономических аспектов для обеспечения безопасной и удобной работы с системой. Как правило, такие аспекты включают:

- выбор цветов для элементов интерфейса разработанного программного средства;
- выбор шрифтов для сообщений, элементов меню и т. д.;
- решения по размещению элементов управления и данных;
- решения по выбору элементов управления;
- сообщения, обеспечивающие удобство работы с информационной системой;

Данный дипломный проект не затрагивает разработку клиентской части. Несмотря на это, необходимо обеспечить высокую надежность и удобство работы для администраторов и специалистов, которые будут взаимодействовать с серверной частью системы.

На основании этого выделяются следующие рекомендации:

- все системные сообщения должны быть лаконичными и понятными, чтобы администратор мог быстро понять суть проблемы или успешность выполненной операции.
- сообщения об ошибках должны содержать четкое описание проблемы и возможные пути её решения.
- логи системы должны быть структурированными и легко читаемыми, что позволит администраторам быстро анализировать и устранять неполадки.

## 3 Политика информационной безопасности

### 3.1 Цель, принципы и задачи защиты информации в АСОИ

Информационная безопасность – одна из самых важных задач, которые необходимо решить при разработке информационной системы. Информационная безопасность позволяет добиться целостности и защищенности разрабатываемой системы.

Основные цели защиты информации при разработке информационных систем:

- обеспечение правового режима использования информации и информационных ресурсов, обрабатываемых в АСОИ;
- обеспечение конфиденциальности, что позволяет закрыть пользователям доступ к информации, к которой они не должны иметь доступ;
- обеспечение доступности данных: авторизованные пользователи должны иметь возможность получить необходимые данные по запросу при условии того, что эти пользователи имеют достаточно прав на получение запрашиваемой информации;
- обеспечение целостности данных, что позволяет уберечь информацию от случайных ошибочных либо злонамеренных действий пользователей, которые могут привести к потере либо порче этой информации.

В ИС решена задача снижения риска несанкционированного воздействия на информацию, система работает с методами защиты данных, разработанными корпорацией Microsoft, они помогают избежать такого рода проблемы в приложении.

### 3.2 Методы и средства обеспечения защиты информационных ресурсов

Выделяются следующие классификации методов по обеспечению защиты информационных ресурсов:

- правовые;
- технологические;
- организационные;
- технические.

Для защиты паролей пользователей в разрабатываемой ИС используется алгоритм шифрования PBKDF2 с хэшем SHA256.

PBKDF2 – это функция получения ключа, разработанная RSA Laboratories, используемая для получения стойких ключей на основе хэша. Она работает путем применения псевдослучайной хэш-функции к строке, в данном случае – к паролю, вместе с солью (строка, используемая для усложнения хэша) и повторением этого процесса большое число раз. При использовании этого алгоритма хэшируется соль и открытый текст для получения первого хэша, затем в цикле тот же алгоритм используется для вычисления хэша от открытого текста и результата предыдущей итерации, после чего возвращается результат применения операции XOR ко всем

вычисленным хэшем. Таким образом пароль преобразуется в хэш, расшифровка которая может занять огромное количество времени. Даже если данных будут утеряны, расшифровать их будет достаточно сложно.

Так как разрабатываемая автоматизированная система – это веб-платформа, то еще один важный критерий – это защита информации, передаваемой по сети интернет. Например, когда пользователь регистрируется или авторизируется, то он вводит пароль, который необходимо передать по сети на сервер. Для того, чтобы данные, передаваемые по сети, не были перехвачены в момент отправки на сервере используется протокол HTTPS.

Протокол HTTPS – это защищенный протокол передачи данных, как и протокол HTTP. Главное отличие HTTPS от HTTP – его защищенность. Эта защищенность обеспечивается шифрованием передаваемых по сети данных. Таким образом, даже если данные были перехвачены, расшифровать их будет достаточно сложно.

### 3.3 Средства защиты информации и информационных ресурсов

Для решения защиты информации в разрабатываемой ИС были использованы следующие методы:

- идентификация и аутентификация пользователя;
- шифрование с помощью BitLocker;
- антивирусная защита информационных ресурсов.

При попытке авторизации пользователя в систему происходит идентификация и аутентификация данных пользователя. Данные операция выполняются вместе.

Идентификация – это операция распознавания субъекта по его идентификатору. В данной ИС идентификатором пользователя является его имя в системе. Для идентификатора установлены следующие ограничения:

- минимальная длина – 4 символа;
- уникальность.

Аутентификация – это операция проверки подлинности. Для проверки пользователя на подлинность используется пароль, устанавливаемый при регистрации.

Для аутентификатора установлены следующие ограничения:

- должен содержать как буквы, так и цифры;
- не должен совпадать с наиболее распространенными паролями.

## 4 Программная реализация информационной системы

### 4.1 Выбор программных средств реализации автоматизированной системы

Автоматизированная система разрабатывается с использованием языков программирования C# и Python, фреймворков ASP.NET Core, Flask, Tensorflow и Keras, в качестве базы данных выступает PostgreSQL.

PostgreSQL – это мощная, открытая реляционная система управления базами данных (СУБД), известная своей надежностью, производительностью и гибкостью. Она поддерживает широкий спектр типов данных, расширяемость и SQL-стандарты, что делает её идеальным выбором для различных приложений, от небольших проектов до крупных корпоративных систем. В рамках автоматизированной системы PostgreSQL используется для надежного хранения и управления медицинскими данными, включая изображения и заключения, обеспечивая их доступность и целостность.

ASP.NET Core – это кросс-платформенный, высокопроизводительный фреймворк для создания современных веб-приложений и API, разработанный компанией Microsoft. Он позволяет создавать масштабируемые и гибкие серверные приложения с использованием языка программирования C#. В системе ASP.NET Core используется для реализации серверной части, обеспечивая быстрый и безопасный доступ к данным, а также обработку запросов от различных компонентов системы. Для расширения функциональности и упрощения разработки используются дополнительные библиотеки, такие как iText7 для работы с PDF-документами, AutoMapper для автоматического отображения объектов, FluentValidation для валидации данных и EntityFrameworkCore для взаимодействия с базой данных.

Flask – это легковесный веб-фреймворк на языке Python, который позволяет быстро и просто создавать веб-приложения и API. Flask известен своей простотой, гибкостью и модульностью, что делает его идеальным для проектов, где важна быстрая разработка и легкость расширения функционала. В нашей системе Flask используется для создания API анализа ИИ, который взаимодействует с основным сервером, обеспечивая модульность и простоту управления различными компонентами приложения.

TensorFlow и Keras – это мощные библиотеки для машинного обучения и нейронных сетей, которые позволяют создавать, обучать и применять модели глубокого обучения. TensorFlow предоставляет комплексную экосистему инструментов и библиотек для построения и развертывания моделей машинного обучения, в то время как Keras предоставляет высокоуровневый интерфейс, упрощающий создание и настройку нейронных сетей. В нашей системе эти библиотеки используются для анализа рентгеновских снимков спины, что позволяет автоматически выявлять патологии и помогать врачам в диагностике. Их интеграция обеспечивает высокую точность и эффективность анализа медицинских изображений.

## 4.2 Реализация базы данных

Для решения поставленной задачи были созданы сущности базы данных. Структура БД представлена на рисунке 4.1. Более подробно структура БД изображена в графической части (301.1-53 01 02.10030563.019.002 ПД).

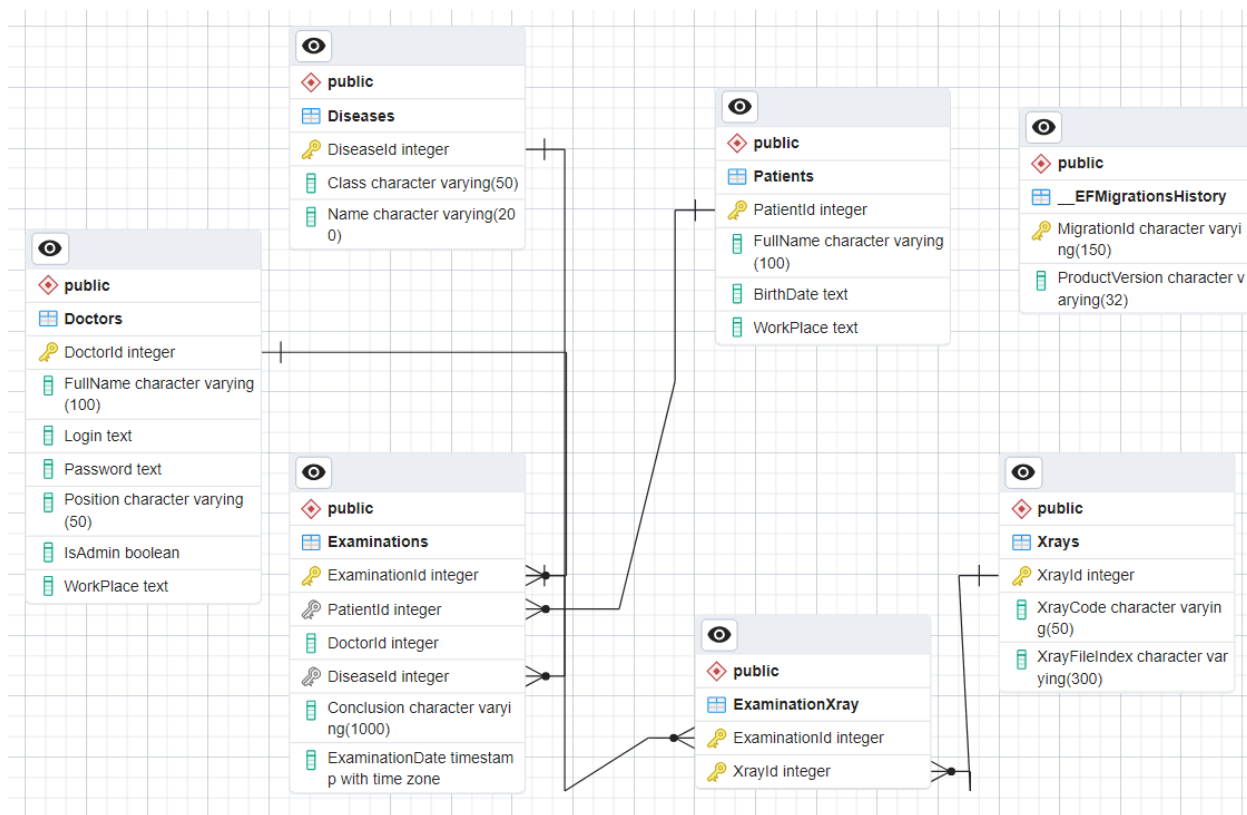


Рисунок 4.1 – Структура базы данных

Описания сущностей соответствуют описаниям структур таблиц из подраздела 2.3 за исключением таблицы \_EFMigrationsHistory. Данная таблица является технической и используется для отслеживания изменений в структуре базы данных при использовании подхода Code First.

Подход Code First в разработке баз данных предполагает, что сначала создаются классы сущностей в коде приложения, а затем на основе этих классов автоматически генерируется структура базы данных. Это позволяет разработчикам работать с объектно-ориентированными моделями, а не с SQL-скриптами, что ускоряет процесс разработки и делает его более интуитивным.

Одним из ключевых элементов подхода Code First являются миграции. Миграции позволяют управлять изменениями в структуре базы данных, отслеживая и применяя эти изменения последовательно. Когда разработчик изменяет модель сущности в коде, создается новая миграция, которая содержит инструкции для обновления схемы базы данных. Эти миграции могут включать добавление новых таблиц или столбцов, изменение типов данных, создание связей между таблицами и другие изменения.

Таблица \_EFMigrationsHistory используется для отслеживания примененных миграций. В этой таблице сохраняются идентификаторы миграций и версия


продукта, что позволяет системе знать, какие изменения были уже применены и избежать повторного применения миграций.


### 4.3 Реализация программного интерфейса серверной части

Разрабатываемый REST API можно разделить на 3 подмодуля: модуль CRUD операций, модуль аутентификации и авторизации, модуль взаимодействия с ИИ.

CRUD — акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание (англ. create), чтение (read), модификация (update), удаление (delete). Введен Джеймсом Мартином (англ. James Martin) в 1983 году как стандартная классификация функций по манипуляции данными.

Для работы с CRUD операциями над базой данных первоначально требуется определение моделей, над которыми операции будут выполняться. Данные модели соответствуют описанной структуре базы данных. При работе с EntityFrameworkCore (EF Core) модель определяется как класс, в котором обозначаются поля, соответствующие атрибутам таблицы, а связи между таблицами выделяются за счет добавления поля типа другой модели. Например, на рисунке 4.2 приведена модель для работы с сущностью «Patients». EF Core при создании миграций автоматически определит PatientId как первичный ключ, поля FullName, BirthDate и WorkPlace как обязательные (NOT NULL) атрибуты таблицы с типом text и наличие у модели пациента списка заключений как обозначение связи 1:M. В дальнейшем при работе с экземпляром данной класса, любые изменения полей помечаются и переносятся в базу данных при обновлении.

```
public class Patient
{
     1 usage
    public int PatientId { get; set; }

     2 usages
    public string FullName { get; set; } = null!;
    public string BirthDate { get; set; } = null!;
    public string WorkPlace { get; set; } = null!;


     2 usages
    public List<Examination> Examinations { get; set; } = null!;
}
```

Рисунок 4.2 Модель для сущности «Patients»

Базу данных в коде представляет объект класса DbContext. Он содержит позволяет работать с таблицами БД как с полями класса, а также определить для них дополнительную конфигурацию, например уточнить максимальную длину строки или минимальное значение числа. Определение таблиц БД как полей класса



DbContext представлено на рисунке 4.3, а конфигурация таблицы Patients на рисунке 4.4

```
public class HospitalDbContext(DbContextOptions<HospitalDbContext> options) : DbContext(options)
{
    [ 4 usages]
    public DbSet<Doctor> Doctors { get; set; } = null!;
    [ 3 usages]
    public DbSet<Xray> Xrays { get; set; } = null!;
    [ 4 usages]
    public DbSet<Examination> Examinations { get; set; } = null!;
    [ 3 usages]
    public DbSet<Disease> Diseases { get; set; } = null!;
    [ 3 usages]
    public DbSet<Patient> Patients { get; set; } = null!;
```

Рисунок 4.3 Определение таблиц как полей класса для работы с БД

```
modelBuilder.Entity<Patient>(builder =>
{
    builder.HasKey(p :Patient => p.PatientId);
    builder.Property(p :Patient => p.FullName).IsRequired()
        .HasMaxLength(100);

    builder.HasMany( navigationExpression: p :Patient => p.Examinations)
        .WithOne( navigationExpression: o :Examination => o.Patient)
        .HasForeignKey(o :Examination => o.PatientId);
});
```

Рисунок 4.4 Дополнительная конфигурация таблицы Patients

Создав модели, необходимо разработать методы, позволяющие совершать CRUD операции. Следуя архитектуре REST необходимо использовать стандартные HTTP-методы. Для операций Create, Read, Update, Delete это методы POST, GET, PUT и DELETE соответственно.

Фреймворк ASP.NET Core для инкапсуляции (инкапсуляция – это упаковка данных и функций в один компонент (например, класс) и последующий контроль доступа к этому компоненту) данных операций предполагает использование специальных классов – контроллеров.

В веб-API контроллеры – это классы, производные от ControllerBase. Контроллеры активируются и удаляются на основе каждого запроса. Контроллеры дают функционал для настройки маршрутизации, создания HTTP-эндпоинтов.

На рисунке 4.5 приведен пример объявления контроллера, а также GET (Read) метода для получения всех записей пациентов из базы данных.

```

[Route(template: "[controller]")]
[ApiController]
@ glebtyanov *
public class PatientsController(HospitalDbContext context, IMapper mapper) : ControllerBase
{
    private readonly PatientValidator _validator = new();

    [HttpGet]
    @ glebtyanov
    public async Task<IActionResult> GetAll()
    {
        var patients :List<Patient> = await context.Patients.ToListAsync();

        return Ok(patients);
    }
}

```

Рисунок 4.5 Пример объявления контроллера и GET (Read) метода.

Перед объявлением контроллера ему присваиваются атрибуты (в C# атрибуты – это специальные инструменты, которые позволяют встраивать в сборку дополнительные данные о классе) `Route` и `ApiController`. Атрибут `Route` определяет URL-путь к контроллеру по имени класса (в данном случае имя класса `Patients`, а URL получится `https://[адрес-сервера]/patients`). Атрибут `ApiController` позволяет фреймворку определить данный класс как класс-контроллер и взаимодействовать с ним соответствующим образом.

В объявлении контроллера указываются объекты, которые могут использоваться внутри него – контекст БД, маппер (класс, позволяющий автоматически привести объект одного класса к другому) и валидатор (класс, позволяющий проверить корректность данных). Присоединение к классу контекста и маппера производится механизмом внедрения зависимостей.

Внедрение зависимостей (Dependency Injection, DI) – это программный паттерн, который позволяет управлять зависимостями между объектами в приложении. DI способствует разделению ответственности и упрощает тестирование кода. При использовании DI объекты не создают свои зависимости напрямую, вместо этого они получают (внедряются) извне, чаще всего через конструктор или свойства.

В контексте проекта DI используется для подключения к контроллеру различных сервисов, таких как контекст базы данных, маппер и валидатор.

Метод GET запрашивает представление ресурса. Данным методом извлекаются данные. Метод GET может быть также использован с параметрами. Так этот метод не предполагает наличия тела запроса, параметры передаются через URL. Пример метода с параметром приведен на рисунке 4.6. В данном случае метод позволяет найти пациента по его уникальному идентификатору и возвращает запись о пациенте и 200 HTTP код (Ок) либо сообщение об ошибке и 404 HTTP код (Не найдено).

```

[HttpGet( template: "{id}")]
new *
public async Task<IActionResult> GetById(int id)
{
    var patient = await context.Patients.FindAsync(id);

    if (patient is null)
    {
        return NotFound("Object with given id was not found");
    }

    return Ok(patient);
}

```

Рисунок 4.6 Пример GET метода с параметром

Метод POST имеет строку и тело запроса. В теле запроса передается необходимая информация, например модель для записи в базу данных, как в примере на рисунке 4.7. Тело запроса передается как параметр метода. В случае с контроллером Patients в тело запроса записывается модель класса PatientAddDto.

DTO (Data Transfer Object) – модель данных, передаваемая из одного слоя приложения в другой. Например, добавляя объект в базу данных, она должна присвоить записи уникальный идентификатор, поэтому невозможно использовать для передачи ранее разработанные модели, так как они потребуют ручное заполнение такого идентификатора. Другой случай – это возвращение каких-либо данных на клиент из API. Например, запись класса Doctor хранит в себе его логин и пароль для входа в систему, которые не рекомендуется возвращать на клиент. В данном случае извне поступает модель. DTO разработаны для всех моделей и необходимы при работе с API.

Для того, чтобы не преобразовывать объект DTO класса в модель, с которой ведется работа на уровне API, используется маппер из библиотеки AutoMapper. Преобразовав модель, необходимо ее провалидировать, для чего используется класс-валидатор из библиотеки FluentValidation. В классе-валидаторе при его объявлении обозначаются правила, по которым будет проверяться корректность данных. Для модели Patient установлены 2 правила для поля FullName: обязательность и максимальная длина в 100 символов. Объявление класса PatientValidator приведено на рисунке 4.8.

```

[HttpPost]
public async Task<IActionResult> Create(PatientAddDto patientAdd)
{
    var patient = mapper.Map<Patient>(patientAdd);

    var validationResult = await _validator.ValidateAsync(patient);

    if (!validationResult.IsValid)
        return BadRequest(validationResult);

    await context.AddAsync(patient);
    await context.SaveChangesAsync();

    return Ok(mapper.Map<PatientGetDto>(patient));
}

```

Рисунок 4.7 POST запрос на добавление записи в БД

```

public class PatientValidator : AbstractValidator<Patient>
{
    2 usages glebtyanov
    public PatientValidator()
    {
        RuleFor(expression: p :Patient => p.FullName).NotEmpty().MaximumLength(100);
    }
}

```

Рисунок 4.8 Объявление класса-валидатора

Метод PUT схож с методом POST, однако его значение – это обновить данные. Пример объявления метода приведен на рисунке 4.9.

Метод DELETE, как и метод GET, не имеет тела запроса, а только строку. Однако метод DELETE не запрашивает никаких данных, только их удаление. Пример объявления DELETE метода приведен на рисунке 4.10.

Все перечисленные методы работают по асинхронной модели. Асинхронная модель позволяет выполнять операции без блокировки основного потока выполнения. В контексте веб-сервисов это означает, что сервер может обрабатывать запросы и выполнять операции, такие как доступ к базе данных или выполнение вычислений, без ожидания их завершения, что повышает производительность и масштабируемость системы.

```

[HttpPut]
@ glebtyanov
public async Task<IActionResult> Update(PatientAddDto patientUpdate)
{
    var patient = mapper.Map<Patient>(patientUpdate);

    var validationResult = await _validator.ValidateAsync(patient);

    if (!validationResult.IsValid)
        return BadRequest(validationResult);

    context.Update(patient);
    await context.SaveChangesAsync();

    return Ok(mapper.Map<PatientGetDto>(patient));
}

```

Рисунок 4.9 PUT запрос на объявление записи в БД

```

[HttpDelete]
@ glebtyanov *
public async Task<IActionResult> Delete(int id)
{
    var patient = await context.Patients.FindAsync(id);

    if (patient is null)
    {
        return NotFound("Object with given id was not found");
    }

    context.Patients.Remove(patient);
    await context.SaveChangesAsync();

    return NoContent();
}

```

Рисунок 4.10 DELETE запрос на удаление записи

Описанные классы-контроллеры разработаны для всех сущностей системы. Для того, чтобы система была многопользовательской разработан контроллер для аутентификации. Аутентификация – это процесс проверки личности пользователя. В данном случае, метод `LoginAsync` (представлен на рисунке 4.11) выполняет аутентификацию, проверяя логин и пароль врача против данных в базе. При успешной аутентификации генерируется JWT токен, который удостоверяет личность пользователя.

```

[HttpPost]
@ glebtyanov
public async Task<string> LoginAsync(LoginDto loginDto)
{
    var doctor = await context.Doctors // DbSet<Doctor>
        .Where(e:Doctor => e.Login == loginDto.Login && e.Password == loginDto.Password)
        .FirstOrDefaultAsync(); // Task<Doctor?>

    var token:string = GenerateJsonWebToken(doctor!);

    return token;
}

```

Рисунок 4.11 Метод, выполняющий аутентификацию пользователя

Второй аспект многопользовательской системы – авторизация. Авторизация – это процесс проверки прав пользователя на выполнение определенных действий. JWT токен, сгенерированный после аутентификации, включает информацию о пользователе, которая может быть использована для определения его прав доступа. Сервер может проверять этот токен при каждом запросе для определения, имеет ли пользователь право выполнять запрашиваемые действия. В системе определена политика авторизации для администратора. Пример ее использования – разрешить добавлять записи о врачах только администратору. В C# политики добавляются ранее разобранным механизмом атрибутов и могут применяться как к целым классам так и к отдельным методам. Пример добавления политики авторизации приведен на рисунке 4.12

```

[HttpPost]
[Authorize(Policy = "AdminOnly")]
@ glebtyanov *
public async Task<IActionResult> Create(DoctorAddDto doctorAdd)
{

```

Рисунок 4.12 Пример добавления политики авторизации

Последним аспектом REST API является модуль взаимодействия с ИИ. Данный модуль представлен еще одним классом-контроллером с эндпоинтом. Он принимает на вход изображение из формы, скачивает его и отправляет запрос на приложение Flask для проведения анализа, получает результат и возвращает его. Описанный метод приведен на рисунке 4.13. Кроме того, при запуске приложения ASP.NET Core также запускается и приложение Flask. Таким образом, приложение представляет собой единую точку входа в серверную часть системы.



```

[HttpPost(template: "predict")]
public async Task<IActionResult> Predict([FromForm] IFormFile? image)
{
    var imageFileName = "";
    try
    {
        imageFileName = await CopyImageToDirectory(image);
    }
    catch (ArgumentNullException)
    {
        return BadRequest(error: "No image was uploaded");
    };

    var predictionResponseMessage = await _httpClient.GetAsync(
        requestUri: $"http://{configuration["AiPredictApi:IP"]}:{configuration["AiPredictApi:Port"]}/predict?imagePath={imageFileName}"); // Task<HttpMessage>

    if (predictionResponseMessage.StatusCode == HttpStatusCode.BadRequest)
    {
        return BadRequest(error: "Server troubles. Try later.");
    }

    var prediction:string = await predictionResponseMessage.Content.ReadAsStringAsync();
    return Ok(prediction);
}

```

Рисунок 4.13 Метод эндпоинт для передачи изображения для анализа от ИИ

## 4.4 Реализация архитектуры нейронной сети

Создание нейронной сети включает в себя несколько ключевых этапов: проектирование архитектуры, подготовка данных, обучение сети и оценка ее производительности. Нейронная сеть состоит из множества связанных между собой нейронов, организованных в слои. Каждый слой выполняет специфические вычисления, которые преобразуют входные данные в выходные значения.

Архитектура нейронной сети определяет, как нейроны и слои связаны между собой. Существует несколько типов слоев, которые могут быть включены в архитектуру нейронной сети:

- входной слой: принимает исходные данные;
- скрытые слои: обрабатывают данные, применяя различные преобразования и фильтрации;
- выходной слой: производит окончательный результат.

Для решения задачи распознавания патологии выбрана архитектура сверточной нейронной сети. Сверточная нейронная сеть (англ. convolutional neural network, CNN) – специальная архитектура ИНС, нацеленная на эффективное распознавание образов. Идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв (англ. convolution layers) и субдискретизирующих слоёв (англ. pooling layers, слоёв подвыборки).

Для создания сети был выбран фреймворк TensorFlow с API от Keras. Данная связка является одной из самых используемых при обучении ИНС. Tensorflow обеспечивает высокопроизводительные библиотеки для обучения сети, а Keras предоставляет интерфейс для описания архитектуры сети.

При этом был выбран подход Transfer Learning, при котором модель создается не с нуля, а на основе уже обученной ранее модели с сохраненными весами, а после подгоняется под конкретную проблему.

В качестве базовой модели была выбрана сеть ResNet50V2. ResNet50V2 – это глубокая нейронная сеть, которая является вариацией оригинальной архитектуры ResNet. Она состоит из 50 слоев и является одной из самых популярных моделей глубокого обучения для классификации изображений. ResNet50V2 хорошо справляется с различными задачами компьютерного зрения благодаря своей глубине и эффективной архитектуре. Веса для модели подбираются из самой большой на текущий момент базы данных изображений ImageNet, таким образом, модель изначально умеет классифицировать изображения. Чтобы подвести ее под требования задачи из нее исключается верхний (последний слой), отвечающий за финальное определение класса, а веса замораживаются (т. е. не будут изменяться при обучении).

Далее определяются следующие слои поверх базовой модели, получая следующую архитектуру:

- слой масштабирования (Rescaling): этот слой масштабирует значения пикселей изображений в диапазоне от 0 до 1. Это важно для нейронной сети, так как большинство моделей ожидают входные данные в этом диапазоне. В данном случае, нейронная сеть ожидает входные изображения с пикселями в диапазоне от 0 до 1;

- базовая модель ResNet50V2: обрабатывает входные данные (масштабированные изображения) и извлекает признаки из них с помощью своих сверточных слоев. В данном случае, используется предварительно обученная модель ResNet50V2, чтобы использовать уже извлеченные признаки изображений из набора данных ImageNet.

- глобальный пулинг (GlobalAveragePooling2D): этот слой выполняет операцию усреднения значений признаков для каждого канала на каждом изображении. Он принимает выходные данные от базовой модели ResNet50V2 и создает вектор признаков фиксированной длины для каждого изображения. Это помогает сократить количество параметров модели и сгруппировать информацию о признаках изображения в более компактном представлении.

- слой регуляризации dropout: этот слой выполняет регуляризацию модели путем случайного выключения (обнуления) некоторых выходов во время обучения. Это помогает предотвратить переобучение модели, уменьшая зависимость между нейронами и повышая их устойчивость. В данном случае, мы устанавливаем коэффициент dropout в 0.2, что означает, что каждый нейрон будет с вероятностью 20% обнуляться во время обучения.

- полносвязный слой (Dense): этот слой представляет собой полносвязный слой, который принимает выходные данные от предыдущего слоя и преобразует их в одно значение. В данном случае, используется один нейрон с линейной активацией (по умолчанию), чтобы предсказать вероятность наличия сколиоза на рентгеновских изображениях.

Определение архитектуры приведено на рисунке 4.14, а ее изображено на рисунке 4.15.



```

base_model = keras.applications.ResNet50V2(
    weights="imagenet",
    input_shape=(224, 224, 3),
    include_top=False,
)

base_model.trainable = False

inputs = keras.Input(shape=(224, 224, 3))

scale_layer = keras.layers.Rescaling(scale=1)
x = scale_layer(inputs)

x = base_model(x, training=False)
x = keras.layers.GlobalAveragePooling2D()(x)
x = keras.layers.Dropout(0.2)(x)
outputs = keras.layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

```

Рисунок 4.14 Определение архитектуры сети

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
rescaling (Rescaling)	(None, 224, 224, 3)	0
resnet50v2 (Functional)	(None, 7, 7, 2048)	23,564,800
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dropout (Dropout)	(None, 2048)	0
dense (Dense)	(None, 1)	2,049

Рисунок 4.15 Итоговая архитектура сети

На рисунках 4.16 и 4.17 указаны полученные метрики точности и функции потерь соответственно.

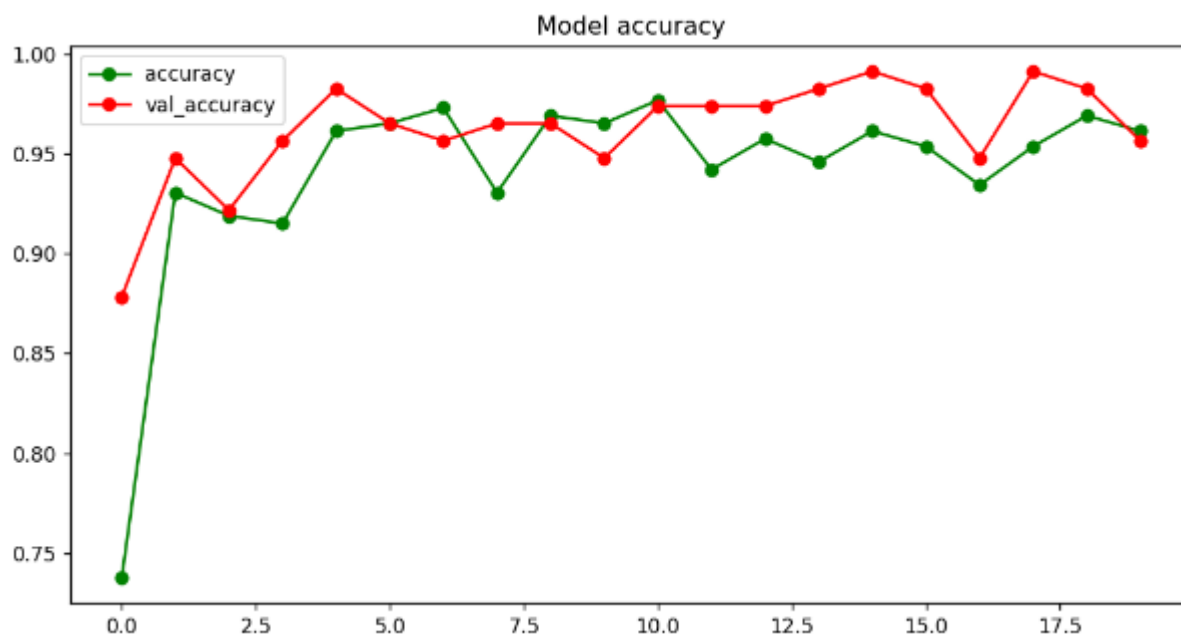


Рисунок 4.16 Метрика точности

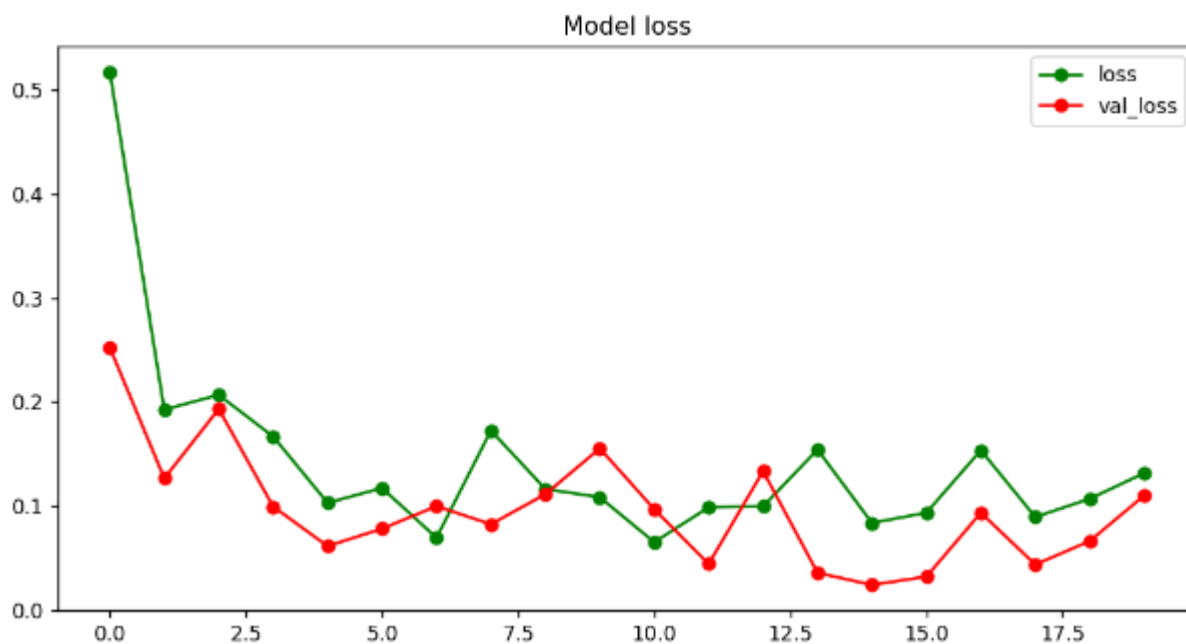


Рисунок 4.17 Метрика потерь

#### 4.5 Реализация модуля анализа ИИ

Модуль анализа ИИ предназначен для автоматического анализа рентгеновских снимков спины с целью выявления патологий. Этот модуль использует вышеописанную модель для обработки изображений и получения прогнозов. Модуль интегрирован в веб-приложение на базе Flask, что позволяет взаимодействовать с ним через HTTP-запросы.

В модуле выделяются 2 основные части:

- функция `predict_by_image`, которая принимает путь к изображению, загружает изображение с использованием библиотеки OpenCV, изменяет его

размер до 224x224 пикселей и преобразует в массив. Далее изображение нормализуется и передается в модель для получения предсказания и функция возвращает уверенность модели в виде процента. Функция представлена на рисунке 4.15

– маршрут /predict, который принимает POST-запрос с JSON-данными, содержащими входные данные для модели, преобразует входные данные в тензор (тензор – геометрический объект, который описывается многомерным массивом) и передает их в модель для получения предсказаний. Возвращает предсказание в формате JSON. Маршрут представлен на рисунке 4.16

```
def predict_by_image(image_path):
    image_path = str(image_path)

    prediction_image = cv2.imread(image_path)
    prediction_image = cv2.resize(prediction_image, (224, 224))
    prediction_image = tf.keras.utils.img_to_array(prediction_image)
    prediction_image = np.expand_dims(prediction_image, axis=0)
    prediction_image = prediction_image / 255

    prediction = model.predict(prediction_image)

    confidence = prediction[0] * 100

    return confidence
```

Рисунок 4.15 Функция для получения прогноза от ИНС

```
@app.route('/predict', methods=['POST'])
def predict():
    try:
        data = request.json
        input_data = data['input']
        input_data = tf.convert_to_tensor([input_data])

        predictions = model.predict(input_data)

        return jsonify({'predictions': predictions})
    except Exception as e:
        return jsonify({'error': str(e)}), 400
```

Рисунок 4.16 Маршрут для приема запроса и получения прогноза от ИНС

## 4.6 Руководство пользователя

Так как задачей проекта выделяется разработка серверной части, данное руководство не включает в себя клиентскую часть и рассматривается не как руководство для конечного пользователя (врача больницы), а руководство для разработчика клиентской части.

Для демонстрации функционала используется Swagger – инструмент, позволяющий документировать и отправлять запросы на API через веб-интерфейс. Swagger включен в приложение ASP.NET Core Web API по умолчанию по адресу /swagger/index.html. Инструмент позволяет увидеть возможные маршруты и нужные форматы запроса и ответа. Интерфейс Swagger приведен на рисунке 4.18.

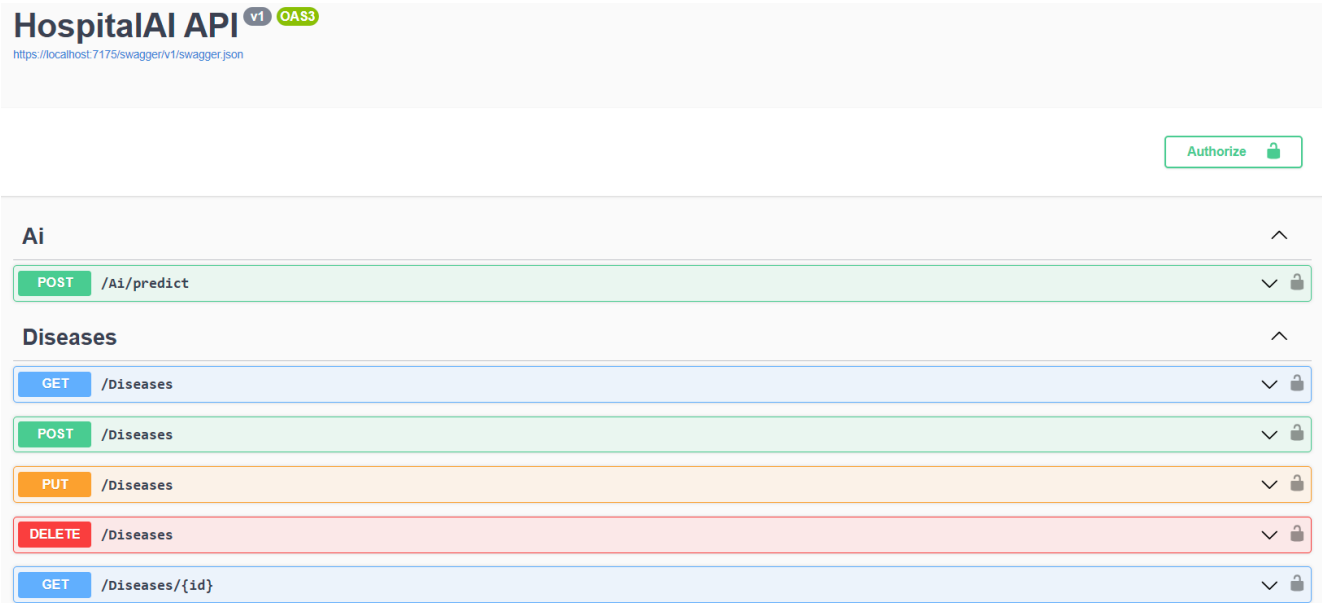


Рисунок 4.18 Интерфейс Swagger

Для аутентификации необходимо послать POST запрос по URL «/identity/login» с телом запроса, содержащим логин и пароль. В случае успеха ответом будет JWT-токен, который затем должен быть помещен в заголовок запроса для авторизации пользователя. На рисунке 4.19 приведен пример запроса, а на рисунке 4.20 расшифровка токена.

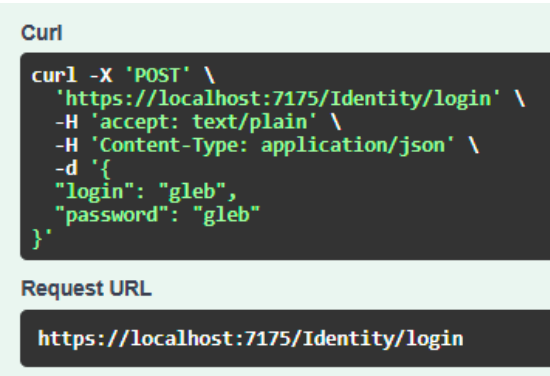


Рисунок 4.19 Пример запроса для аутентификации

```
{
  "Position": "Врач-рентгенолог",
  "FullName": "Пантелеев Глеб Викторович",
  "IsAdmin": "True",
  "nbf": 1718344520,
  "exp": 1718351720,
  "iat": 1718344520,
  "iss": "issuer",
  "aud": "issuer"
}
```

Рисунок 4.20 Расшифровка полученного JWT-токена

Для добавления данных также посылается POST запрос. Например, на рисунке 4.21 приведен пример запроса на добавление пациента.

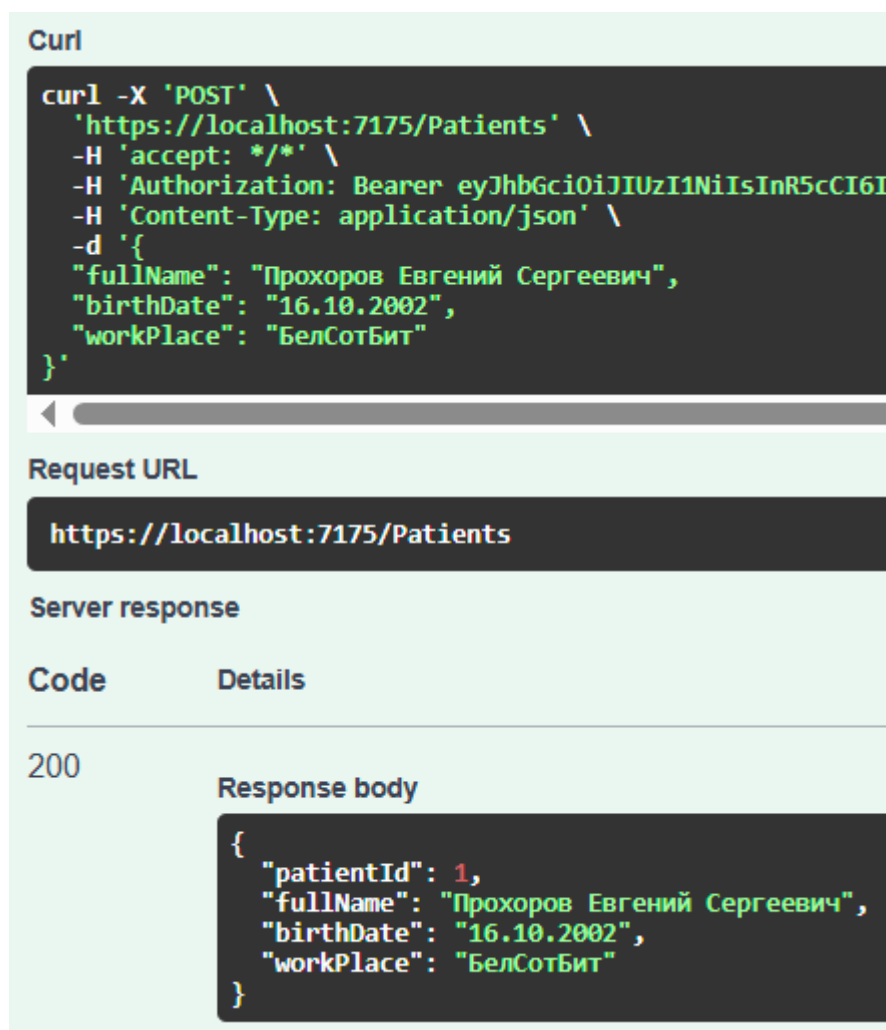


Рисунок 4.21 Добавление пациента

Для получения данных используется GET запрос. Например, на рисунке 4.22 продемонстрировано получение данных о пациентах.

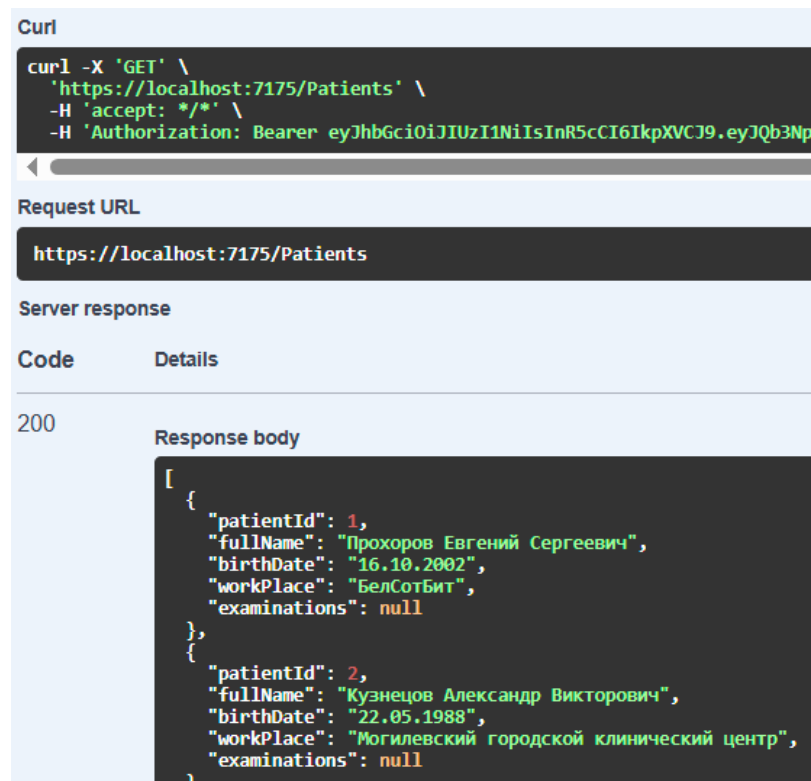


Рисунок 4.22 Получение данных о пациентах

Для получения прогноза от ИИ необходимо передать изображение через форму HTML. Пример такой формы представлен на рисунке 4.23.



Рисунок 4.23 Форма для отправки изображения для получения прогноза

## 5 Расчет экономической эффективности технических решений

### 5.1 Общая постановка к технико-экономическому обоснованию

Дипломный проект на тему «Многопользовательская информационная система медицинских изображений. Разработка нейронной сети и серверной части» выполнен для автоматизации оптимизации процессов обследования и составления заключения по результатам медицинских исследований, при помощи базы данных и веб-приложения, а также нейронной сети, составляющей предварительное заключение.

Целью дипломного проекта является автоматизация обработки и хранения медицинских изображений. Задачи автоматизации системы состоят в объединении систем взаимодействия с изображениями и заключениями, автоматизации процессов создания, регистрации, поиска медицинских исследований, разработка пользовательского интерфейса, разработка нейронной сети, способной составлять предварительные заключения по данному медицинскому изображению (рентгену) спины.

На данный момент в учреждении здравоохранения (УЗ) «Могилевская областная клиническая больница» выделяется специализированная система eDoctor, которая работает в web-клиенте, вся информация хранится на серверах компании-разработчика. Главная цель системы – автоматизация рабочего места. Помимо этого, система дает возможность ограничения прав доступа.

Для сбора и визуализации данных медицинских приборов (Dicom) используется RadiAnt DICOM Viewer – программа, способная открывать и отображать результаты исследований, полученные из различных модальностей медицинских изображений: цифровая рентгенография (ЦР), компьютерная томография (КТ), магнитно-резонансная томография (МРТ), ультразвуковая диагностика (УЗИ) и др.

Вышеупомянутые Dicom-изображения хранятся на серверах больницы. Хранилище представлено диском объемом 100 ТБ, организовано с помощью файловой базы данных SQLite. SQLite – однофайловая СУБД на языке С, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы. На сервере хранится один файл, позволяющий связать случайно названный файл снимка, расположенный в случайной папке с конкретный пациентом и вернуть его по запросу.

Главное наблюдение – две основные системы не связаны. Заполнение и обработка данных о пациенте из eDoctor и данных снимков из Dicom Viewer – два отдельных бизнес-процесса. Системы поставляются разными организациями, имеют отдельные клиенты. Такой подход предлагает область для улучшения эффективности и надежности, если удастся добиться автоматизированного взаимодействия двух систем. Это обеспечит гарант связанности данных о пациенте и снимке, а также ускорит работу сотрудника больницы.

Дополнительно, снимки пациентов могут быть классифицированы по различным патологиям и их степеням. При этом все они расположены в локальном

хранилище, к которому можно развернуть организованный доступ. Такие факторы позволяют создать автоматизированную систему с использованием технологий нейронных систем, предлагающую выдачу заключения по пациенту по анализу его снимка. Добившись интеграции данных о заключениях с данными о снимках с помощью первой предлагаемой системы, на ее базе можно развернуть обучение сети, причем масштабируемое с течением времени (получением новых снимков и заключений).

В соответствии с вышеописанным поставлены следующие задачи:

– создание автоматизированной информационной системы, позволяющей объединить функционал eDoctor как системы хранения данных о пациентах, RadiAnt DICOM Viewer как системы доступа и анализа снимков и хранилища на базе больницы. Система предлагает унифицированный доступ к снимку и данным как для других систем больницы, так и врачу за счет единого клиентского приложения;

– внедрение в объединенную АСОИ нейронной сети, использующей систему для создания базы знаний. На вход сеть будет принимать новый снимок пациента, на выход генерировать рекомендацию к заключению.

В рамках исследования рынка и консультации с врачами на базе больницы было выявлено, что на данный момент аналогичные введенные в производство системы отсутствуют, что делает проект новым, оригинальным и более востребованным.

Все основные параметры разработанной системы представлены в таблице 5.1.

Таблица 5.1 – Характеристика проектируемой информационной системы

Показатель	Параметр
1	2
Область прикладной деятельности	Автоматизированное рабочее место.
Цель автоматизации	Объединение процессов работы с изображениями и заключениями.
Функция программных средств	Хранение и обработка результатов медицинских исследований.
Уровень автоматизации	Автоматизированное рабочее место врача-рентгенолога
Порядок внедрения и использования	Документация и обеспечение ее качества; алгоритмы и программы, и соответствие их требованиям
Модель данных	Реляционная (табличная)
Прямая эффективность	Сокращение цикла обработки информации, экономия трудозатрат на обработку данных и экономия материальных затрат
Показатель	Параметр
Косвенная эффективность	Достоверность выходных данных



Продолжение таблицы 5.1

1	2
Режим эксплуатации обработки данных	В режиме реального времени поддержкой одновременной работы нескольких пользователей
Масштаб программных средств	920 строк – рукописный код, 500 строк – автогенерируемый код
Исходный язык	Объектно-ориентированные (C#, Python)
Класс пользователя	Специалист
Требуемые рабочие характеристики	Емкость памяти (средняя); длительность обработки (быстрая), производительность (средняя)
Требование защиты	Надежная защита от несанкционированного доступа к системе; защита данных
Требование надежности	Доступность, отказоустойчивость
Требования к вычислительным ресурсам	Процессор: Intel Core i3, 2-ядерный; Оперативная память: 6 Гб, DDR3 SO-DIMM; Жесткий диск: 500 Гб, HDD; Видеоадаптер: Intel HD Graphics, встроенный

В таблице 5.2 представлена общая характеристика сравниваемых вариантов. Предлагается программный модуль по поставленным задачам, так как базовый вариант имеет существенные недостатки, например отсутствие автоматизированного поиска данных, низкая скорость обработки.

Таблица 5.2 - Общая характеристика сравниваемых вариантов

Наименование показателей	Базовый	Проектный
Информационный процесс	Автоматизированное хранение данных и их обработка	
Средства информационного процесса		
Получение данных	Ручной поиск	Автоматизированный (запросы на PostgreSQL)
Хранение данных	Снимки компьютерной томографии и бумажные документы	На сервере (таблицы в БД на PostgreSQL)
Обработка данных, представление данных	В бумажном формате	В электронном форме
Исполнитель процесса	врач-рентгенолог	

Разработанная медицинская информационная система обработки изображений позволит в автоматизированном режиме выполнять ряд операций, что значительно сократит время на формирование медицинских документов, повысит оперативность и точность информации, а также обеспечит высокое качество обработки и объективность представляемых данных. Для определения эффективности разрабатываемой информационной системы произведено сравнение: традиционная система обработки данных врачом-рентгенологом (базовый вариант) и автоматизированная система обработки данных (проектируемый вариант). Такой подход позволяет определить целесообразность

создания специализированного программного обеспечения в системе управления медицинскими данными и обработки изображений.

## 5.2 Расчет трудоемкости (производительности)

Предварительно информационный процесс решения  $i$ -ой задачи по сравниваемым вариантам разбивается на последовательные стадии ( $j$ -ые операции). Функционально норма штучно-калькуляционного времени на решение задачи складывается из следующих элементов:

$$t_{\text{шк}} = \frac{t_{\text{пз}}}{n_{\text{п}}} + t_{\text{оп}} + t_{\text{об}} + t_{\text{отл}}$$

где  $t_{\text{пз}}$  – подготовительно-заключительное время на партию решаемых задач;  
 $n_{\text{п}}$  – количество последовательно решаемых задач за один прогон;  
 $t_{\text{оп}}$  – оперативное время выполнения задачи (сумма основного и вспомогательного не перекрываемого времени);  
 $t_{\text{об}}$  – время обслуживания рабочего места;  
 $t_{\text{отл}}$  – время на отдых и личные надобности.

Время  $t_{\text{об}}$  и  $t_{\text{отл}}$  чаще определяется косвенно как доля от оперативного времени  $t_{\text{оп}}$  в размере 0,12 – 0,16, в расчетах соответственно приняты значения: 0,12 и 0,15.

Результаты расчета трудоемкости произведены на основе нормативной трудоемкости, норма штучно-калькуляционного времени приведена в таблице 5.3.

Таблица 5.3 – Результаты расчета трудоемкости по вариантам

Наименование элементов нормы времени	Норма времени по вариантам ( $t_{\text{шк}}$ ), мин	
	базовый	проектный
Заключения по результатам обработки рентгеновских снимков		
Подготовительно-заключительное время, $t_{\text{пз}}$	2,00	2,00
Оперативное время, $t_{\text{оп}}$	210	70,0
Время обслуживания, $t_{\text{об}}$	25,2	8,4
Время на отдых и личные надобности, $t_{\text{отл}}$	31,5	10,5
Итого на задачу	268,7	90,9

Годовая программа  $A_{\text{г}}$  по задаче проведения медицинских исследований в виде рентгена и компьютерной томографии принята на уровне среднего количества 50 пациентов, принимаемых в кабинете рентгена в месяц ( $A_{\text{г}} = 600$  исследований в год).

## 5.3 Расчет единовременных затрат

По вариантам сравнения единовременные затраты (инвестиции) складываются из следующих основных элементов:

					301.1-53 01 02.10031777.006 ПЗ	Лист 45
Изм.	Лист	№ докум.	Подп.	Дата		

$$K = K_0 + K_{об} + K_{зд} + K_{пр}$$

где  $K_0$  – стоимость комплекта машин и оборудования с учетом необходимой мебели, р.;

$K_{об}$  – стоимость запасов в оборотные средства, р.;

$K_{зд}$  – стоимость потребной площади здания, р.;

$K_{пр}$  – затраты на проектирование, р.

Определение стоимости единовременных затрат в оборудование:

$$K_0 = \sum_{i=1}^n N_{ni} P_{oi} (1 + \alpha_{Ti} + \alpha_{Mi}) d_3,$$

где принятое число единиц  $i$ -го оборудования ( $N_{ni} \geq N_{Pi}$  – округляется до целого), шт;

$P_{oi}$  – цена приобретения  $i$ -го оборудования по варианту, р.;

$\alpha_{Ti}$ ,  $\alpha_{Mi}$  – коэффициенты, учитывающие величину транспортно-заготовительных расходов ( $\alpha_{Ti} = 0,05$ ), величину затрат на монтаж и отладку ( $\alpha_{Mi} = 0,05$ );

$d_3$  – доля занятости принятых рабочих мест на решение задачи по варианту,  $d_3 = N_P / N_{П}$ .

Расчетное количество машин (рабочих мест) на решение задачи по вариантам вычисляется следующим образом:

$$N_P = \frac{t_{шт} A_{г}}{F_d k_3}$$

где  $t_{шт}$  – норма штучного времени;

$F_d$  – годовой действительный фонд работы оборудования (рабочего места), ч;

$k_3$  – коэффициент запаса, учитывающий неравномерность поступления информации (для стабильных процессов  $k_3 = 0,90-0,95$ ; периодических –  $k_3 = 0,85-0,90$ ; нерегулярных –  $k_3 = 0,70-0,085$ ), принято  $k_3 = 0,85$ .

Величина годового действительного фонда рабочего места врача-рентгенолога определяется по следующей формуле:

$$F_d = F_{CM} K_{CM} D_P (1 - K_{пр})$$

где  $F_{CM}$  – номинальный сменный фонд работы, ч (8 часов);

$K_{CM}$  – коэффициент сменности – число смен работы в течение рабочего дня (2 смены);

$D_P$  – число рабочих дней в году (принято рабочих дней 253, из них 245 с полной продолжительностью и 8 с сокращенной продолжительностью);

$K_{\text{ПР}}$  – коэффициент, учитывающий долю времени простоев в плановых ремонтах,  $K_{\text{ПР}} = 0,05$ .

$$F_{\text{д}} = (245 \cdot 8 + 8 \cdot 7) \cdot 2 \cdot (1 - 0,05) = 3830,4 \text{ ч.}$$

Получим расчетное количество рабочих мест:

$$N_{\text{р}}^{\text{б}} = \frac{\frac{268,7}{60} \cdot 600}{0,85 \cdot 3830,4} = \frac{2687}{3255,84} = 0,826 \text{ р. м.}$$

$$N_{\text{р}}^{\text{п}} = \frac{\frac{90,90}{60} \cdot 600}{0,85 \cdot 3830,4} = \frac{909}{3255,84} = 0,279 \text{ р. м.}$$

Определим принятое количество рабочих мест путем округления их расчетной величины  $N_{\text{р}}$  до ближайшего целого числа в большую сторону:

$$N_{\text{п}}^{\text{б}} \geq N_{\text{р}}^{\text{б}} = 1, N_{\text{п}}^{\text{п}} \geq N_{\text{р}}^{\text{п}} = 1.$$

Соответственно, доля занятости принятых рабочих мест:

$$d_3^{\text{б}} = \frac{N_{\text{р}}^{\text{б}}}{N_{\text{п}}^{\text{б}}} = \frac{0,826}{1} = 0,826,$$

$$d_3^{\text{п}} = \frac{N_{\text{р}}^{\text{п}}}{N_{\text{п}}^{\text{п}}} = \frac{0,279}{1} = 0,279$$

В дипломном проекте для технического обеспечения программного модуля на рабочем месте врача-рентгенолога предусмотрено стандартное оборудование. Стоимость комплекта оборудования для автоматизированного места врача-рентгенолога представлена в таблице 5.4.

Таблица 5.4 – Комплект оборудования рабочего места врача-рентгенолога

Наименование оборудования	Количество	Цена, р.	Стоимость, р.
Системный блок Jet Office 3i10100D8SD24VGALW50	1	1 229,49	1 229,49
Монитор НКС MB20S1FS	1	267,58	267,58
Принтер Pantum P2516	1	299,00	299,00
Итого	-	-	1796,07

Определим стоимость единовременных затрат на оборудование.

$$K_0^{\text{б}} = 1796,07 \cdot 1,1 \cdot 0,826 = 1631,91 \text{ р.,}$$

$$K_0^{\Pi} = 1796,07 \cdot 1,1 \cdot 0,279 = 551,21 \text{ р.}$$

Стоимость оборотных средств, связанных с решением задачи по варианту, рассчитывается по формуле:

$$K_{\text{ОБ}} = \sum_{j=1}^m P_{Mj} Z_{Mj},$$

где  $P_{Mj}$  – цена приобретения  $j$ -го материала, используемого при решении задачи по варианту, р.;

$Z_{Mj}$  – средний запас  $j$ -го материала, используемого при решении задачи по варианту (принимается в размере 0,05-0,10 от годового расхода  $j$ -го материала в натуральном выражении)., принято 0.05.

Цены на расходные материалы представлены в таблице 5.5.

Таблица 5.5 – Используемые материалы по вариантам

Наименование материалов	Базовый		Проектный	
	Цена за единицу, рублей	Запас, шт.	Цена за единицу, рублей	Запас, шт.
1. Бумага офисная «Херох Performer», формат А4 (500 листов)	15,60	4	15,60	2
2. Тонер HP 1.1 1020, 1022 (©2612A)	61,00	1	61,00	1

Определена стоимость оборотных средств по вариантам

$$K_{\text{ОБ}}^{\text{Б}} = 15,6 \cdot 4 + 61 \cdot 1 = 123,40 \text{ р.}$$

,

$$K_{\text{ОБ}}^{\Pi} = 15,6 \cdot 2 + 61 \cdot 1 = 92,2 \text{ р.}$$

Единовременные затраты в стоимость потребной площади здания по вариантам определяются в рублях по формуле:

$$K_{\text{ЗД}} = (S_{\text{ЗД}}^{\text{ПР}} P_{\text{ЗД}}^{\text{ПР}} + S_{\text{ЗД}}^{\text{СБ}} P_{\text{ЗД}}^{\text{СБ}}) d_3 ,$$

где  $S_{\text{ЗД}}^{\text{ПР}}, S_{\text{ЗД}}^{\text{СБ}}$  – норматив площади производственной (6-10 м<sup>2</sup> на одно рабочее место), служебно-бытовой (5 – 7 м<sup>2</sup> на одного оператора);

$P_{\text{ЗД}}^{\text{ПР}}, P_{\text{ЗД}}^{\text{СБ}}$  – цена (стоимость) квадратного метра здания производственного 160 долл., служебно-бытового 220 долл. (по курсу Национального банка Республики Беларусь 3,18 р/долл.США на 01.06.2024).

Стоимость потребной площади здания включается в расчет единовременных затрат. Рассчитываем стоимость потребной площади здания для решения поставленной задачи.

$$K_{3Д}^Б = (8 \cdot 160 + 6 \cdot 220) \cdot 3,18 \cdot 0,826 = 6854,18 \text{ р.},$$

$$K_{3Д}^П = (8 \cdot 160 + 6 \cdot 220) \cdot 3,18 \cdot 0,279 = 2306,78 \text{ р.}$$

Затраты на проектирование для базового варианта не включается в расчет единовременных затрат. Произведен расчет затрат на проектирование программного продукта в рамках, решаемых задач по формированию документов.

$$K_{\text{ПР}} = \frac{P}{\text{ПР}} \cdot \frac{T}{\text{ПР}} \cdot (1 + 0,11 \cdot (1 - \frac{D_{i+1}}{D_i}) \cdot (K_{\text{ВД}} \cdot K_{\text{Д}}^Y + K_{\text{НД}} \cdot (1 - K_{\text{Д}}^Y))),$$

где  $P_{\text{ПР}}$  – сметная ставка одного человеко-месяца проектирования, р.;

$T_{\text{ПР}}$  – трудоемкость проектирования, чел.-мес.;

$D_i$  и  $D_{i+1}$  – величина дефектности для исходного уровня качества: по базовому варианту –  $i \sigma$  и проектируемому –  $(i+1)\sigma$ ;

$K_{\text{ВД}}$  и  $K_{\text{НД}}$  – коэффициент уровня трудовых затрат на устранение выявленных и не выявленных дефектов;

$K_{\text{Д}}^Y$  – уровень выявления дефектов в программном изделии в процессе проведения тестирования.

Сметная ставка одного человеко-месяца проектирования рассчитывается в рублях по формуле:

$$P_{\text{ПР}} = Z_T \cdot K_T \cdot K_{\text{П}} \cdot K_{\text{Д}} \cdot K_{\text{СС}} \cdot K_{\text{НР}},$$

где  $Z_T$  – месячная тарифная ставка первого разряда,  $Z_T = 250$  р.;

$K_T$  – тарифный коэффициент проектировщика,  $K_T = 2,65$ ;

$K_{\text{П}}$  – коэффициент премирования,  $K_{\text{П}} = 1,3$ ;

$K_{\text{Д}}$  – коэффициент, учитывающий дополнительную заработную плату,  $K_{\text{Д}} = 1,1$ ;

$K_{\text{СС}}$  – коэффициент, учитывающий отчисления на социальные нужды,  $K_{\text{СС}} = 1,346$ ;

$K_{\text{НР}}$  – коэффициент, учитывающий накладные расходы,  $K_{\text{НР}} = 1,2$ .

$$P_{\text{ПР}} = 250 \cdot 2,65 \cdot 1,3 \cdot 1,1 \cdot 1,346 \cdot 1,2 = 1530,2 \text{ р.}$$

Трудоемкость проектирования ПИ в человеко-месяцах в соответствии с конструктивной моделью стоимости рассчитывается по следующей формуле:

$$T_{\text{ПР}} = A_T KLOC^B M_P + T_{\text{auto}},$$

где  $A_T$ ,  $B$  – коэффициенты конструктивной модели стоимости по принятому типу проекта,  $B$  – величина коэффициента изменяется в диапазоне 1,01 – 1,26 и зависит от пяти масштабных факторов  $W_i$ . На основании экспертных оценок коэффициент вычисляется по формуле

$$B = 1,01 + 0,01 \sum_{i=1}^5 W_i;$$

$KLOC$  – количество тысяч строк в программном продукте без учета числа строк, полученных в результате автоматического генерирования кодов,  $KLOC = 0,920$  тыс. строк.;

$M_p$  – поправочный множитель (который зависит от 15 факторов затрат конструктивной модели стоимости на основании принятых характеристик факторов для проекта по таблице 5.7 и численных значений множителей  $M_i$  по таблице 5.8);

$T_{auto}$  – затраты на автоматически генерируемый программный код, чел.-мес.

$$T_{auto} = (KALOC \cdot AT / 100) ATPROD ,$$

где  $KALOC$  – количество строк автоматически генерируемого кода,  $KALOC = 0.5$  тыс. строк;

$AT$  – процент автоматически генерируемого кода  $AT = 54,08\%$ ;

$ATPROD$  – производительность автоматически генерируемого кода тысяч строк за месяц,  $ATPROD = 0,5$  тыс. строк.

$$T_{auto} = \left( 0,5 \cdot \frac{54,08}{100} \right) \cdot 0,5 = 0,1352$$

Для расчета коэффициента  $B$  составлена таблица 5.6, в которой приводится характеристика масштабных факторов.

Таблица 5.6 – Характеристика масштабных факторов

Масштабный фактор $W_i$	Оценка
Предсказуемость PREC	2
Гибкость разработки FLEX	3
Разрешение архитектуры риска RESL	3
Связанность группы TEAM	2
Зрелость процесса PMAT	2
Итого	12

Коэффициент  $B$  на основании экспертных оценок равен:

$$B = 1,01 + 0,01 \cdot 12 = 1.13$$

Тип модели: распространенная, соответственно коэффициенты определены в размере  $A_T=2,4$ ;  $B=1,13$ .

Факторы затрат конструктивной модели стоимости обобщены в таблице 5.7. Произведен расчет поправочного множителя  $M_p$ :

$$M_p = 1 \cdot 1,09 \cdot 0,88 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1 \cdot 1,16 \cdot 1 \cdot 1 \cdot 1 \cdot 1,24 \cdot 1 = 1,092$$

Определим трудоемкость проектирования ПИ по формуле

$$T_{\text{ПР}} = 2,4 \cdot 0,92^{1,13} \cdot 1,092 + 0,1352 = 2,756 \text{ чел. – мес.}$$

Таблица 5.7 – Факторы затрат конструктивной модели стоимости

Название $M_i$ – го фактора	Уровень фактора	Описание	Численное значение
1	2	3	4
1. Требуемая надежность ПО – RELY	Номинальный	Умеренные, легко восстанавливаемые потери	1
2. Размер базы данных – DATA	Высокий	$100 \leq D / P < 1000$	1,09
3. Сложность модуля в зависимости от области применения – CPLX	Низкий	Несложная вложенность структурированных операторов. Простые предикаты.	0,88
4 Требуемая повторная используемость – RUSE	Низкий	Нет	0,91
5. Документирование требований жизненного цикла (ЖЦ) – DOCU	Номинальный	Оптимизированы к требованиям жизненного цикла	1
6. Ограничение времени выполнения платформы – TIME	Номинальный	Использование $\leq 50\%$ возможного времени	1
7. Ограничение оперативной памяти платформы - STOP	Номинальный	Использование $\leq 50\%$ доступной памяти	1
8. Изменчивость платформы – RVOL	Низкий	Значительные изменения – каждые 12 мес., незначительные – каждый мес.	0,87
9. Возможность аналитика - ACAP	Номинальный	55%	1
10. Возможность программиста – PCAP	Низкий	35%	1,16
11. Опыт работы с приложениями – AEXP	Номинальный	1 год	1
12. Опыт работы с платформой – PEXP	Номинальный	1 год	1
13. Опыт работы с языком и утилитами – LTEX	Номинальный	1 год	1
14. Использование программных утилит – TOOL	Очень низкий	Редактирование, кодирование, отладка	1,24
15. Требуемый график разработки – SCED	Номинальный	100%	1



Реальный уровень качества программного изделия в процессе его эксплуатации оценивается количеством содержащихся в нем дефектов (ошибок). В целях соизмеримости программных изделий, которые разработаны на различных языках, плотность дефектов (дефектность) в таких случаях обычно рассчитывается на единицу размера программного кода «тысяча строк эквивалентного ассемблерного кода», KAELOC. Объем ПИ конкретного языка программирования в KLOC умножается на соответствующий коэффициент пересчета  $K_{\Pi}$ , ( $K_{\Pi} = 2.5$ ) по формуле:

$$KAELOC = 1 \cdot 2.5 = 2.5$$

Соотношение поля допуска с полем разброса (в «сигмах») связывают с числом дефектов на единицу объема ПИ размером KAELOC ( $D_i$ ). В данном случае уровень качества в базовом варианте –  $4\sigma$  ( $D_i = 6,210$ ), а проектируемом –  $5\sigma$  ( $D_{i+1} = 0,233$ ).

В соответствии с объемом строк KAELOC (до 8) в ПИ определен  $K_{ВД}=1.5$ ,  $K_{Н.Д}=3.5$ ,  $K^y_{Д}=0,75$ .

Определим затраты на проектирование:

$$K_{\Pi\Pi}^{\Pi} = 1530,2 \cdot 2.756 \cdot \left( 1 + 0,11 \cdot \left( 1 - \frac{0,233}{6,210} \right) \cdot (1,5 \cdot 0,75 + 3,5 \cdot (1 - 0,75)) \right) \\ = 5110,22 \text{ р}$$

Таблица 5.8 – Единовременные затраты по вариантам

Наименование элементов единовременных затрат	Величина по элементам, р.	
	Базовый	Проектный
Стоимость комплекта оборудования	1631,91	551,21
Стоимость запасов в оборотные средства	123,40	92,20
Стоимость потребной площади здания	6854,18	2306,78
Затраты на проектирование	-	5110,22
Итого единовременных затрат К	8609,49	8060,41

Единовременные затраты внедрения программного продукта в проектном варианте ниже на 549,08 р. не смотря на высокие затраты на проектирование, что связано со снижением трудоемкости выполнения поставленных задач.

#### 5.4 Расчет годовых текущих издержек

Годовые текущие издержки по вариантам сравнения рассчитываются по следующим статьям:

$$И = И_{ЗП} + И_{М} + И_{Э} + И_{РО} + И_{РЗ} + И_{НР},$$

где  $И_{ЗП}$  – годовые затраты на заработную плату операторов (системных администраторов) с начислениями, р.;

$I_M$  – годовые затраты на материалы за вычетом реализованных отходов, р.;

$I_{Э}$  – годовые затраты на силовую электроэнергию, р.;

$I_{PO}$  – годовые затраты на ремонт и содержание оборудования, р.;

$I_{PЗ}$  – годовые затраты на ремонт и содержание зданий, р.;

$I_{НР}$  – годовые накладные расходы по управлению и обслуживанию производства, р.

Годовые затраты на заработную плату операторам (системным администраторам) с начислениями по  $i$ -ым операциям (рабочим местам) рассчитываются по формуле

$$I_{ЗП} = \sum_{i=1}^n t_{ШКi} Z_T K_{Ti} (1 + K_{Pi}) (1 + K_D) (1 + K_{CC} + K_H) A_G,$$

где  $t_{ШКi}$  – норма штучного времени по  $i$ -ой операции, ч;

$Z_T$  – часовая тарифная ставка первого,  $Z_T = 1.60$  р.;

$K_{Ti}$  – тарифный коэффициент разряда по  $i$ -ой операции  $K_{T10} = 2.48$ ;

$K_{Pi}$  – коэффициент премирования по  $i$ -ой операции,  $K_{Pi} = 0.3$ ;

$K_D$  – коэффициент, учитывающий дополнительную заработную плату,  $K_D = 0.1$ ;

$K_{CC}$  – коэффициент, учитывающий отчисления на социальные нужды,  $K_{CC} = 0.346$ ;

$K_H$  – коэффициент, учитывающий налоги на заработную плату,  $K_H = 0.1$ .

$$I_{ЗП}^B = \frac{268,7}{60} \cdot 1.60 \cdot 2.48 \cdot 1.3 \cdot 1.1 \cdot 1.346 \cdot 600 = 20522,04 \text{ р.}$$

$$I_{ЗП}^П = \frac{90,9}{60} \cdot 1.60 \cdot 2.48 \cdot 1.3 \cdot 1.1 \cdot 1.346 \cdot 600 = 6942,51 \text{ р.}$$

Годовые затраты на материалы рассчитываются по формуле:

$$I_M = \sum_{j=1}^m (P_{Mj} H_{Mj} - P_{OTj} H_{OTj}) A_G,$$

где  $P_{Mj}$  и  $P_{OTj}$  – цена приобретения, используемого  $j$ -го материала и реализуемых отходов, р.;

$H_{Mj}$  и  $H_{OTj}$  – норма расхода  $i$ -ых видов материала (листы бумаги, формы документов, картриджи и т.д.) и реализуемых отходов на решаемую задачу, шт. (кг);

$A_G$  – годовое количество решаемых задач.

Определен расход материалов по решаемым задачам.

– расход офисной бумаги. В среднем для печати документации предусмотрено использование бумаги формата А4: расход в среднем для проектного варианта с учетом годовой программы решения задач составляет 960

листов (только для пациента); для базисного варианта потребность в бумаге в два раза выше, что определяет ее количество в размере 1920 листов (для врача и пациента). С учетом количества листов бумаги в пачке для базового варианта необходимо 4 пачки бумаги, для проектного варианта – 2 пачки бумаги;

– расходный материал по принтеру: Тонер HP 1.1 1020, 1022 (©2612A), 1000 г. Ресурс картриджа для используемого принтера составляет 2000 страниц, в соответствии с потребностью в бумаге (проектный - 960 листов; базовый – 1920 листов) для принтера необходима заправка в базовом и проектном вариантах 1 раз в год;

– в базовом и проектном вариантах предусмотрены канцтовары в размере 20% от стоимости бумаги и тонера. Стоимость канцтоваров соответственно по базовому и проектному вариантам составила 24,69 р. и 18,44 р.

Определим годовые затраты на материалы:

$$И_{\text{М}}^{\text{Б}} = 4 \cdot 15,6 + 1 \cdot 61 + 24,68 = 148,08 \text{ р.},$$

$$И_{\text{М}}^{\text{П}} = 2 \cdot 15,6 + 1 \cdot 61 + 18,44 = 110,64 \text{ р.}$$

Годовые издержки на потребляемую электроэнергию в рублях, если оборудование работает в режиме полной занятости в течение рабочего дня, рассчитываются по формуле

$$И_{\text{Э}} = \sum_{i=1}^n W_i F_{\text{Д}} K_3 P_{\text{Э}} d_3,$$

где  $F_{\text{Д}}$  – годовой действительный фонд работы единицы оборудования, ч;

$W_i$  – потребляемая мощность оборудования на  $i$ -ой операции (таблица 5.9), кВт;

$P_{\text{Э}}$  – цена (тариф) за один киловатт-час потребляемой электроэнергии ( $P_{\text{Э}} = 0,33734 \text{ р./кВт} \cdot \text{ч}$ ).

Таблица 5.9 – Потребляемая мощность оборудования

Оборудование	Потребляемая мощность, кВт
Системный блок Jet Office	0.550
3i10100D8SD24VGALW50	
Монитор НКС MB20S1FS	0.125
Принтер Pantum P2516	0.073
Итого	0.748

Определим годовые издержки на потребляемую электроэнергию:

$$И_{\text{Э}}^{\text{Б}} = 0,748 \cdot 3780,68 \cdot 0,9 \cdot 0,33734 \cdot 0,826 = 709,19 \text{ р.},$$

$$И_{\text{Э}}^{\text{П}} = 0,748 \cdot 3780,68 \cdot 0,9 \cdot 0,33734 \cdot 0,279 = 239,44 \text{ р.}$$

Годовые издержки на ремонт и содержание оборудования в рублях рассчитываются по формуле:

$$И_{PO} = 0,64 \cdot K_0^{0,84},$$

где  $K_0$  – стоимость используемого оборудования, р.

Определены годовые издержки на ремонт и содержание оборудования:

$$И_{PO}^B = 0,64 \cdot 1631,91^{0,84} = 319,78 \text{ р.},$$

$$И_{PO}^П = 0,64 \cdot 551,21^{0,84} = 128,50 \text{ р.}$$

Годовые затраты на ремонт и содержание зданий:

$$И_{PЗ} = K_{ЗД} \frac{Н_{PЗ}}{100},$$

где  $Н_{PЗ}$  – норматив на ремонт и содержание здания  $Н_{PЗ} = 2,5 \%$ ;

$K_{ЗД}$  – стоимость используемых зданий, р.

$$И_{PЗ}^B = 6854,18 \cdot 2,5/100 = 171,36 \text{ р.},$$

$$И_{PЗ}^П = 2306,78 \cdot 2,5/100 = 57,67 \text{ р.}$$

Годовые накладные расходы складываются из следующих статей затрат: на управление и обслуживание производства  $И_y$ , освещение  $И_{OC}$ , воду на бытовые нужды  $И_B^B$ , тепловой энергии на горячую воду  $И_{TЭ}^{ГВ}$ , отопление  $И_{TЭ}^{OT}$ , вентиляцию  $И_{TЭ}^{BT}$ :

$$И_{НР} = И_y + (И_{OC} + И_B^B + И_{TЭ}^{ГВ} + И_{TЭ}^{OT} + И_{TЭ}^{BT}) \cdot d_3$$

Годовые расходы на управление и обслуживание производства определяются по формуле:

$$И_y = И_{ЗП} K_{KY},$$

где  $K_{KY}$  – коэффициент, учитывающий косвенные расходы по управлению  $K_{KY} = 0,2$ ;

$$И_y^B = 20522,04 \cdot 0,2 = 4104,41 \text{ р.},$$

$$И_y^П = 6942,51 \cdot 0,2 = 1388,5 \text{ р.}$$

Для следующих статей затраты для базового и проектируемого варианта равны. Годовые затраты электроэнергии на освещение рассчитываются по формуле:

$$I_{OC} = P_{\Sigma} W_S S F_O,$$

где  $W_S$  – норма освещенности  $W_S = 0,03$ , кВт/м<sup>2</sup>;

$S$  – площадь зданий производственных и служебно-бытовых,  $S=14$  м<sup>2</sup>;

$F_O$  – годовой осветительный фонд времени  $F_O = 2400$  ч при двусменной.

$$I_{OC} = 0,33734 \cdot 0,03 \cdot 14 \cdot 2400 = 340,04 \text{ р.}$$

Годовые затраты воды на бытовые нужды:

$$I_B = P_B^B N_B^B \chi_P D_P,$$

где  $P_B^B$  – цена воды на бытовые нужды  $P_B^B = 4,4086$  р./ м<sup>3</sup>;

$N_B^B$  – норма расхода воды на бытовые нужды за сутки на одного работника  $N_B^B = 0,025$  м<sup>3</sup>;

$\chi_P$  – численность работников (врачей – рентгенологов),  $\chi_P = 1$  чел.

$$I_B = 4,4086 \cdot 0,025 \cdot 252 = 27,78 \text{ р.}$$

Годовые затраты тепловой энергии на горячую воду рассчитываются по формуле:

$$I_{TЭ}^{ГГ} = P_{TЭ} q_{ТХ}^B (t^{BG} - t^{BX}) 10^{-6} V^{BG} F^{BG},$$

где  $P_{TЭ}$  – цена (тариф) за теплоэнергию  $P_{TЭ} = 151,1356$  р./Гкал;

$q_{ТХ}^B$  – удельная тепловая характеристика воды  $q_{ТХ}^B = 1$ , ккал/(м<sup>3</sup> ч °С);

$V^{BG}$  – объем потребления горячей воды за час  $V^{BG} = 3$  л;

$t^{BG}, t^{BX}$  – температура воды горячей в системе  $t^{GB} = +65$  °С, воды холодной -  $t^{BX} = +5$  °С;

$F^{BG}$  – период теплоснабжения водой горячей:

$$F^{BG} = F_{CM} K_{CM} D_P,$$

$$I_{TЭ}^{GB} = 151,1356 \cdot (65 - 5) \cdot 10^{-6} \cdot 3 \cdot (245 \cdot 8 + 8 \cdot 7) = 54,85 \text{ р}$$

Годовые затраты тепловой энергии на отопление рассчитываются по формуле:

$$I_{TЭ}^{OT} = P_{TЭ} q_{ТХ}^{3Д} (t_{ВН}^{3Д} - t_{Н}^{3Д}) 10^{-6} V^{3Д} F^{OT},$$

где  $q^{3Д}_{ТХ}$  – удельная тепловая характеристика здания  $q^{3Д}_{ТХ} = 0,40$  ккал/(м<sup>3</sup> ч °С);

$V^{3Д}$  – объем помещения здания по наружному обмеру  $V^{3Д} = SH$ ,  $H = 3,5$ м;

$t^{3Д}_{ВН}$ ,  $t^{3Д}_{Н}$  – температура воздуха внутри помещения и снаружи соответственно  $t^{3Д}_{ВН} = + 20$ ,  $t^{3Д}_{Н} = - 10$ , °С;

$F^{OT}$  – отопительный период за год  $F^{OT} = 4320$  ч, ч.

$$И^{OT}_{ТЭ} = 151,1356 \cdot 0,4 \cdot ((20 - (-10)) \cdot 10^{-6} \cdot 14 \cdot 3,5 \cdot 4320 = 383,91 \text{ р.}$$

Годовые затраты тепловой энергии на вентиляцию рассчитываются:

$$И^{BT}_{ТЭ} = P_{ТЭ} q^{BT}_{ТХ} (t^{BT}_{ВН} - t^{BT}_{Н}) \cdot 10^{-6} V^{3Д} F^{BT} K^П_{ТЭ},$$

где  $q^{BT}_{ТХ}$  – удельная тепловая характеристика вентиляции здания  $q^{BT}_{ТХ} = 0,15$  ккал/(м<sup>3</sup> ч °С);

$t^{BT}_{ВН}$ ,  $t^{BT}_{Н}$  – температура воздуха вытяжного и снаружи соответственно  $t^{BT}_{ВН} = +20$  °С,  $t^{BT}_{Н} = - 1,5$  °С;

$F^{BT}$  – период работы вентиляционной системы за год  $F^{BT} = 1400$  ч;

$K^П_{ТЭ}$  – коэффициент, учитывающий потери теплоэнергии,  $K^П_{ТЭ} = 1,18$ .

$$И^{BT}_{ТЭ} = 151,1356 \cdot 0,15 \cdot ((20 - (-1,5)) \cdot 10^{-6} \cdot 14 \cdot 3,5 \cdot 1400 \cdot 1,18 = 33,95 \text{ р.}$$

Определены годовые накладные расходы:

$$И^Б_{НР} = 4104,41 + (325,4 + 0,506 + 52,35 + 367,37 + 32,48) \cdot 0,826 = 4747,13 \text{ р.,}$$

$$И^П_{НР} = 1388,5 + (325,4 + 0,506 + 52,35 + 367,37 + 32,48) \cdot 0,279 = 1605,60 \text{ р.}$$

Результаты расчетов за год по статьям текущих издержек сводятся в таблицу 5.10.

Таблица 5.10 - Годовые текущие издержки

Наименование статей издержек	Величина по вариантам, р.	
	базовый	проектный
Затраты на заработную плату системному администратору с начислениями	20522,04	6942,51
Затраты на материалы	148,08	110,64
Затраты на силовую электроэнергию	709,19	239,44
Затраты на ремонт и содержание оборудования	319,78	128,50
Затраты на ремонт и содержание зданий	171,36	57,67
Накладные расходы	4747,13	1605,60
Итого годовых текущих издержек И	26617,58	9084,36

Годовые текущие издержки в результате внедрения программного продукта снизились на 17533,22 р. по сравнению с базовым вариантом, что отвечает высокому уровню окупаемости затрат. Снижение текущих издержек при использовании программного модуля произошло за счет уменьшения трудоемкости решения задач, и, соответственно, уменьшения затрат по проведению медицинских исследований.

## 5.5 Расчет показателей экономической эффективности

Для технических решений в области совершенствования информационной системы, имеющих внутрипроизводственную значимость, годовой экономический эффект определяется по следующей формуле:

$$\mathcal{E}_Г = \mathcal{Z}_Г^Б - \mathcal{Z}_Г^П,$$

где  $\mathcal{Z}_Г^Б$ ,  $\mathcal{Z}_Г^П$  – годовые приведенные затраты по базовому и проектному варианту.

Величина приведенных затрат по сравниваемым вариантам определяется по формуле:

$$\mathcal{Z}_i = E_H K + \sum_i p_i K_i + И,$$

где  $E_H$  – нормативный коэффициент эффективности,  $E_H = 0,1$ ;

$K$ ,  $K_i$  – единовременные затраты (таблица 5.8) суммарные, по  $i$ -ым элементам, р.;

$p_i$  – норма реновации единовременных затрат, которая рассчитывается как обратная величина срока службы  $t_{СЛ}$  по  $i$ -ым элементам (средства вычислительной техники  $t_{СЛ} = 5$  лет; оборотные средств и затраты на проектирование  $t_{СЛ} = 4$  лет; здания и средства социально-культурной сферы  $t_{СЛ} = 50$  лет), а с учетом морального износа определяется по формуле

$$p_i = \frac{E_H}{(1 + E_H)^{t_{СЛ}} - 1};$$

$И$  – годовые текущие издержки (таблица 5.10), р.

В таблицу 5.11 внесены нормы реновации единовременных затрат по элементам в соответствии с выбранной величиной срока службы по  $i$ -м элементам.

Таблица 5.11 – Норма реновации элементов единовременных затрат

Наименование элементов единовременных затрат	Срок службы $t_{сл}$ по $i$ -м элементам	Норма реновации
Стоимость комплекта машин и оборудования с учетом необходимой мебели	5	0.164
Стоимость запасов в оборотные средства	4	0.215
Затраты на проектирование	4	0.215
Стоимость потребной площади здания	50	0.000859

Рассчитаем приведенные затраты:

$$З_{Г}^B = 0,1 \times 8609,49 + (0,164 \times 1631,91 + 0,215 \times 123,40 + 0,215 \times 0 + 0,000859 \times 6854,18) + 26617,58 = 27778,58 \text{ р.,}$$

$$З_{Г}^П = 0,1 \times 8060,41 + (0,164 \times 551,21 + 0,215 \times 92,20 + 0,215 \cdot 5110,22 + 0,000859 \times 2306,78) + 9084,36 = 10171,32 \text{ р.,}$$

Определен годовой результирующий эффект:

$$Э_{Г} = 27778,58 - 10171,32 = 17607,26 \text{ р.}$$

По результатам расчетов в данном дипломном проекте единовременные затраты по проектному варианту ниже затрат по базовому, соответственно расчет срока окупаемости дополнительных единовременных затрат не производится. Приведенные выше показатели сравнительной эффективности показывают высокий уровень выгодности внедрения разработанного программного продукта.

## 5.6 Организация внедрения системы

Дипломный проект на тему «Многопользовательская информационная система медицинских изображений. Разработка нейронной сети и серверной части» выполнен для автоматизации оптимизации процессов обследования и составления заключения по результатам медицинских исследований, при помощи базы данных и веб-приложения, а также нейронной сети, составляющей предварительное заключение. График внедрения программного продукта приведен в таблице 5.12.

Таблица 5.12 – План-график внедрения разработанного программного продукта

Наименование операции	Исполнитель	Время, дни
1 Установка необходимого ПО	Системный администратор или разработчик	2
2 Настройка	Системный администратор или разработчик	1
3 Тестирование	Разработчик и пользователь	1
4 Обучение пользователя	Разработчик	2
Итого	-	6



## 5.7 Заключение по разделу

Основные технико-экономические показатели дипломного проекта, которые определяют экономическую эффективность внедрения программного продукта, сведены в таблицу 5.13.

Таблица 5.13 – Техничко-экономические показатели по сравниваемым вариантам

Наименование показателей	Величина по вариантам	
	Базовый	Проектный
1 Годовое количество решаемых задач		
1.1 Среднее количество пациентов	600	600
2 Норма времени решения задачи, мин		
2.1 Обработка медицинского исследования	268,7	90,9
3 Уровень качества программного изделия, $i \sigma$	4	5
4 Потребляемая мощность вычислительных средств, кВт	0.748	0.748
5 Единовременные затраты, р.	8609,49	8060,41
6 Годовые текущие издержки, р.	20307,06	6949,55
7 Годовые приведенные затраты, р.	21468,07	8321,50
8 Годовой результирующий эффект, р.	-	13146,57
9 Продолжительность освоения ПИ, дней	-	6

Анализ технико-экономических показателей по проектному варианту относительно базового варианта позволил установить отсутствие капитальных вложений, так как единовременные затраты по проекту ниже базового варианта. При этом отмечено значительное снижение годовых текущих издержек на 13357,51 р. или на 66%, единовременных затрат на 549,08 р. или на 6% и приведенных затрат на 13146,57 р или на 61%.

В результате годовой результирующий эффект по приведенным затратам составил 13146,57 р. Внедрение программного продукта обеспечит повышение эффективности (снижение трудоемкости) при проведении медицинских исследований.

Автоматизированная технология для хранения обработки результатов медицинских исследований позволит снизить накладные расходы на единицу услуг на предприятии.

В ходе выполнения данного раздела дипломного проекта, обоснована экономическая целесообразность внедрения программного модуля относительно действующей в настоящее время технологии обработки информации.

## 6 Охрана труда

### 6.1 Система управления охраной труда в учреждении здравоохранения «Могилевская областная клиническая больница»

Охрана труда – система обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающая правовые, социально-экономические, санитарно-гигиенические, психо-физические, лечебно-профилактические, реабилитационные и иные мероприятия. (ст. 221, Трудовой Кодекс Республики Беларусь от 26.07.1999 № 296-3 (ред. от 17.07.2023 г. №300-3).

Для создания и поддержания безопасных условий труда, соответствия требованиям охраны труда, обеспечения безопасного состояния территорий, зданий, сооружений и лечебных процессов, предотвращения травм и профессиональных заболеваний в УЗ «Могилевская областная клиническая больница» внедрена система управления охраной труда (СУОТ), утвержденная главным врачом.

Главными целями СУОТ являются предотвращение производственных травм и профессиональных заболеваний, управление профессиональными рисками, их устранение или снижение до приемлемого уровня, а также постоянное улучшение условий и охраны труда.

Политика больницы в области охраны труда в соответствии с нормами законодательства:

- учитывает специфику учреждения здравоохранения и характер его деятельности;
- четко изложена, имеет дату и введена в действие главным врачом больницы;
- отражает обязательства руководства и работников соответствовать требованиям законодательства в области охраны труда;
- направлена на обеспечение конституционного права каждого работника на здоровые и безопасные условия труда;
- доведена до сведения и доступна работникам больницы;
- предусматривает постоянный анализ для совершенствования СУОТ;
- включает проведение обучения, инструктажа, стажировки и проверки знаний работников по вопросам охраны труда.

Матрица распределения ответственности структуры СУОТ УЗ «Могилевская областная клиническая больница» представлена в таблице 6.1.

Таблица 6.1. Матрица распределения ответственности

№ п/п	Задачи и функции в системе управления охраной труда	Ответственный	Исполнитель	Соисполнитель
1	2	3	4	5
1.	Определение, разработка и внедрение Политики в области охраны труда.	Руководитель, Управляющий по вопросам производственной и внутренней регуляции	Специалист по охране труда	

Продолжение таблицы 6.1

1	2	3	4	5
2.	Ознакомление работающих с Политикой.	Управляющий по вопросам производственной и внутренней регуляции	Специалист по охране труда, Кадровая служба	Руководители структурных подразделений
3.	Постановка целей и задач в области охраны труда.	Руководитель, Управляющий по вопросам производственной и внутренней регуляции	Специалист по охране труда	Руководители структурных подразделений
4.	Организация разработки и функционирования СУОТ.	Управляющий по вопросам производственной и внутренней регуляции	Специалист по охране труда	Руководители структурных подразделений
5.	Формирование программы управления охраной труда для достижения целей	Руководитель, Управляющий по вопросам производственной и внутренней регуляции	Специалист по охране труда	Руководители структурных подразделений
6.	Идентификация видов осуществляемой деятельности, выполняемых работ и услуг.	Руководитель, Управляющий по вопросам производственной и внутренней регуляции	Руководители структурных подразделений	Специалист по охране труда
7.	Формирование реестра и фонда законодательных и иных ОРД, содержащих обязательные требования в области охраны труда.	Руководитель, Управляющий по вопросам производственной и внутренней регуляции		
8.	Ознакомление с картами опасностей и рисков	Управляющий по вопросам производственной и внутренней регуляции	Руководители структурных подразделений	Специалист по охране труда
9.	Профессиональный подбор кадров	Управляющий по вопросам производственной и внутренней регуляции	Кадровая служба	Специалист по охране труда
10.	Определение медицинской пригодности при приеме на работу (первичные медосмотры)	Управляющий по вопросам производственной и внутренней регуляции	Доверенный врач, мед. комиссия	Кадровая служба

Продолжение таблицы 6.1

1	2	3	4	5
11.	Проведение периодических медицинских осмотров.	Управляющий по вопросам производственной и внутренней регуляции	Заместитель главного врача по медицинской части, главная медсестра, Руководитель структурных подразделений	Специалист по охране труда
12.	Страхование от несчастных случаев на производстве и профессиональных заболеваний.	Руководитель, Управляющий по вопросам производственной и внутренней регуляции	Главный бухгалтер, Кадровая служба	
13.	Нормативно - ресурсное обеспечение СУОТ.	Управляющий по вопросам производственной и внутренней регуляции	Планово-экономическая служба, Главный бухгалтер, Кадровая служба	Специалист по охране труда

В рамках реализации СУОТ в больнице разработаны:

- положение о службе охраны труда учреждения;
- инструкция о проведении контроля за соблюдением законодательства об охране труда в учреждении;
- приказ о назначении ответственных лиц за охрану труда и пожарную безопасность;
- приказы главного врача о назначении ответственных должностных лиц по соблюдению правил пожарной безопасности;
- приказы главного врача о создании системы обучения требованиям пожарной безопасности, утверждении программы пожарно-технического минимума для разных категорий работников;

Прием на работу новых работников производится после прохождения медосмотра, вводного инструктажа по охране труда и пожарной безопасности.

Во всех подразделениях больницы проводятся первичные, повторные, внеплановые и целевые инструктажи, что отражено в журналах регистрации инструктажа по охране труда.

## **6.2 Анализ выполнения мероприятий по обеспечению пожарной безопасности в учреждении здравоохранения «Могилевская областная клиническая больница»**

В комплексе противопожарных мер больницы включены разработка и выполнение требований пожарной безопасности, контроль правильной эксплуатации приборов и помещений, обеспечение требуемого состояния пожарной безопасности сооружений, разработка и выполнение противопожарных мероприятий.

Согласно инструкции противопожарные мероприятия в больнице должны соблюдаться следующим образом.

В палатных отделениях не допускается пользоваться утюгами, электроплитками и другими бытовыми электронагревательными приборами в палатах и помещениях проживания людей, размещать людей в период проведения ремонтов.

Для размещения тяжелобольных и лиц, лишенных возможности самостоятельного свободного передвижения, использовать кровати, позволяющие перевозить (эвакуировать) их при пожаре. При отсутствии таких кроватей для эвакуации людей при пожаре необходимо иметь носилки из расчета одни носилки на каждые пять человек. Носилки должны храниться в специально отведенных местах.

В помещении следует применять приборы и оборудование, допущенные к работе в среде с повышенным содержанием кислорода. Замена неисправных приборов допускается только на равноценные, имеющие соответствующие отметки о возможности их использования в среде с повышенным содержанием кислорода.

Расстояние от отопительных приборов и других источников тепла до барокамеры, баллонов с кислородом и закисью азота, стеллажей, шкафов из негорючих материалов для хранения рентген плёнки и легковоспламеняющихся лекарственных препаратов должно быть не менее 1 м.

Масса носителей диагностической информации (рентгено- и флюорограммы) в архиве отдела лучевой диагностики не должна превышать 100 кг. Рентген плёнка должна храниться в закрывающемся металлическом шкафу. В основных архивах рентген снимки должны храниться в шкафах или на стеллажах. Шкафы и стеллажи должны выполняться из негорючих материалов.

Легковоспламеняющиеся жидкости и горючие жидкости (далее – ЛВЖ и ГЖ соответственно) должны храниться отдельно от других медикаментов, при этом количество ЛВЖ и ГЖ, одновременно хранящихся в каждом рабочем помещении, не должно превышать 3 л. Хранить их необходимо в герметически закрытой таре в специальном шкафу или в металлическом ящике с крышкой.

На рабочем месте разрешается иметь огнеопасные вещества в количествах, необходимых для выполняемой в данный момент операции. Работники клинко-диагностических лабораторий (далее - лаборатории) обязаны знать пожарную опасность применяемых химических веществ, материалов, препаратов и соблюдать правила пожарной безопасности при работе с ними.

Все работы в лаборатории, связанные с возможностью выделения токсичных или пожаровзрывоопасных паров и газов, должны производиться только в вытяжных шкафах. Пользоваться вытяжными шкафами с разбитыми стеклами или неисправной вентиляцией не допускается.

Рабочие поверхности столов, стеллажей, вытяжных шкафов, предназначенных для работы с пожаровзрывоопасными веществами и материалами, должны иметь покрытие из негорючих, исключающих искрообразование при ударах, материалов, а при работе с кислотами, щелочами, ЛВЖ и ГЖ дополнительно оборудоваться бортиками из негорючего материала для исключения пролива жидкости за пределы шкафа или стола.

На столах и в вытяжных шкафах, где проводятся работы с открытым огнем и электронагревательными приборами, не допускается хранение ЛВЖ и ГЖ, переливание их и загрузка ими лабораторного оборудования.

Кислоты, щелочи и другие химически активные вещества в стеклянной таре разрешается переносить только в специальных ящиках.

В лабораториях не допускается:

- хранить на рабочем месте и в рабочих помещениях вещества и препараты с неизвестными пожароопасными свойствами;
- размещать емкости с ЛВЖ и ГЖ на расстоянии менее 1 м от нагревательных приборов, горелок и других источников огня;
- выливать отработанные ЛВЖ и ГЖ в канализацию;
- оставлять на рабочем месте промасленные ветошь и бумагу;
- оставлять без присмотра зажженные горелки и др. нагревательные приборы;
- применять лабораторное оборудование с неисправными системами охлаждения;
- нагревать на открытом огне, в электронагревательных приборах с открытыми нагревательными элементами сосуда, содержащие ЛВЖ и ГЖ, а также использовать водяные бани для обогрева сосудов, в которых находятся реагирующие с водой химические вещества и соединения.

На термостатах, холодильниках, сейфах, где хранятся инфицированные объекты, наносятся надписи: «Во время пожара не вскрывать!» и «Выносить запрещается!».

В отделениях и кабинетах физиотерапии не допускается применять для покрытия пола и изготовления занавесей процедурных кабин синтетические материалы, способные образовывать и накапливать статическое электричество.

Подогревать парафин и озокерит необходимо на подогревателях заводского изготовления или водяной бани. Стол, на котором устанавливаются подогреватели, должен быть покрыт термостойким материалом.

Вместимость кабинетов электросветолечения, лечения ультразвуком, тепло-, водо- и грязелечения, массажа должна соответствовать требованиям нормативных документов.

Все лечебные электроаппараты должны находиться в исправном состоянии, иметь надежное заземление, заводскую электрическую схему и технический паспорт. Неисправные аппараты к применению не допускаются.

Кабинеты для электросветолечения должны иметь самостоятельную питающую электрическую линию, идущую от распределительного щита, проложенную проводами (кабелем) необходимого по расчету сечения.

Обслуживание и ремонт электро- и светолечебных аппаратов и оборудования должны проводиться специалистами, прошедшими соответствующую подготовку и имеющими удостоверения. Результаты осмотра и ремонта должны фиксироваться в журнале установленной формы.

При проведении электролечебных процедур вне физиотерапевтического кабинета должны выполняться требования инструкций по эксплуатации оборудования. Не допускается оставлять включенными в электрическую сеть по окончании рабочего дня лечебные электроаппараты.

Анализ выполнения мероприятий по обеспечению пожарной безопасности в больнице показал следующее:

Хранение веществ и материалов осуществляется в соответствии с порядком хранения веществ и материалов, а также с учётом однородности средств их тушения.

Применение и хранение веществ и материалов неизвестного состава и с неизученными пожаровзрывоопасным и свойствами на территории больницы запрещено.

Здания, сооружения, помещения и установки используются по целевому назначению, определённом проектно-сметной документацией.

Для обеспечения пожарной безопасности при эксплуатации медицинского оборудования выполняются следующие необходимые требования:

- расстановка медицинского оборудования обеспечивает свободные проходы и подходы к нему;

- оборудование должно находиться в исправном состоянии.

Эксплуатация медицинского и специального оборудования с неисправной или отключенной автоматикой, обеспечивающей пожарную безопасность, в больнице запрещается. Не выполняются операции на медицинском оборудовании и установках с неисправностями, которые могут привести к пожарам, а также при отключении приборов, определяющих заданные режимы температуры, условия концентрации и других технологических параметров, газов, паров, жидкостей.

Пожароопасные вещества и материалы, получаемые больницей как правило имеют упаковку, соответствующую требованиям стандартов, технических условий с предупреждающими надписями маркировкой, знаками безопасности на этикетках и паспорт, в котором указывается наименование и марка вещества, дата его изготовления и номер сертификата, информация о пожаровзрывоопасных свойствах, а также меры пожарной безопасности при обращении с ним.

Для мойки, обезжиривания, антисептической обработки медицинских изделий применяются негорючие технические жидкости и составы, а также безопасные в пожарном отношении установки и способы.

Для технических помещений разработан план размещения веществ и материалов с указанием их наиболее характерных свойств (взрывопожароопасные, ядовитые, химически активные и т.п.).

Соблюдены расстояния и проходы в соответствии с противопожарными нормами.

Каждый вид газа хранится отдельно. Баллоны со сжатыми и сжиженными газами закреплены и размещены так, чтобы они не подвергались механическим воздействиям.

Пожарная безопасность при эксплуатации электроустановок обеспечена правильным выбором степени защиты электрооборудования, защитой электрических аппаратов и проводников от токов короткого замыкания и перегрузок, заземлением электроприемников, организационно-техническими мероприятиями, профилактическими ремонтами.

Все электроустановки обеспечены аппаратами защиты от токов короткого замыкания и других ненормальных режимов работы. После окончания работы все электроустановки в помещениях за исключением специального назначения отключают.

Для обеспечения пожарной безопасности при эксплуатации вентиляции и вентсистем предусмотрены меры пожарной безопасности инструкциями по их эксплуатации, согласно которым при возникновении пожара в производственном помещении, в вентиляционной камере, в воздуховодах или на любом участке вентиляционной системы следует немедленно выключить вентиляторы, сообщить о случившемся в пожарную службу и принять меры к ликвидации пожара.

Для соблюдения требуемых противопожарных требований имеется достаточное количество огнетушителей, инвентаря для тушения возгораний, сигнализаций, комплексов оповещения, телефонной сети.

Выполненные мероприятий пожарной безопасности указаны в журналах и отчетности по пожаробезопасности.

Штат работников, включая медицинский и инженерный персонал проходят вводный, первичный инструктаж при трудоустройстве, а вследствие проводятся периодические целевые и внеплановые инструктажи.

Проводятся совместные занятия с пожарным персоналом для выработки навыков по тушению возгораний и эвакуации сотрудников.

### 6.3 Выводы и предложения

По результатам анализа противопожарных мероприятий можно сделать вывод, что меры противопожарной профилактики в больнице соблюдены:

- приказом руководителя назначены ответственные за обеспечение пожарной безопасности, противопожарного режима, а также за содержание и эксплуатацию средств противопожарной защиты, инженерного оборудования, вентиляционных систем, электроустановок, средств связи, оповещения;
- на всех этажах и в помещениях расположены инструменты для тушения возгорания (огнетушители, пожарные краны, инвентарь);
- таблички пожарной безопасности эвакуационных выходов расположены с соблюдением правил, схемы эвакуации созданы для каждого этажа в отдельности и для всего здания в целом;
- запрещено применение электроприборов;



- в качестве отделки используются пожаробезопасные материалы;
- возле каждого телефона указан номер телефона вызова пожарной части;
- в кабинете рентгена разрешено хранить определенный объем пленки без металлических шкафов, пленку необходимо хранить в отдельных зданиях в удалении 15 м для минимизации рисков распространения возгораний;
- гардеробы и места хранения одежды не загромождены, остаются свободными;
- бумагу, промасленную ветошь в специальных контейнерах вывозят с территории больницы и утилизируют;
- размещение в палатах строго регламентируется, минимальное расстояние между койками 0,8 м, а проходы не менее 1,2 м.

При возникновении пожара производится оповещение пожарной охраны и эвакуация сотрудников и пациентов. В первую очередь эвакуации подлежат тяжелобольные, инвалиды и дети на кроватях или носилках. Пациенты, которые могут самостоятельно передвигаться, выходят из здания под контролем медперсонала. Их количество пересчитывается по спискам, переданным медперсоналом. Тушение пожара может начинать параллельно с эвакуацией людей или после нее, данное решение принимает главный врач.

Следует отметить, что специфика работы организаций здравоохранения оказывает непосредственное влияние на применяемые в больнице документы, содержащие требования по охране труда и пожарной безопасности. При этом зачастую, в использовании медицинских работников находится специальное медицинское электрооборудование с истекшим сроком службы или произведено в странах мира, осуществляющих продажу наиболее дешевого оборудования, которое может быть небезопасным в использовании.

Учитывая вышеизложенное, для повышения устойчивости к возникновению пожаров рекомендуется модернизация и своевременная замена оборудования, прошедшего нормативный срок службы, а также внедрение современного оборудования и средств для пожаротушения.

## 7 Энерго- и ресурсосбережение

### 7.1 Энергоэффективные технологии в здравоохранении

Целью дипломного проекта является автоматизация обработки и хранения медицинских изображений. Разрабатываемая многопользовательская информационная система обработки медицинских изображений применяется в здравоохранении.

Здравоохранение — сфера деятельности, к задачам которой относится обеспечение доступного медицинского обслуживания населения, сохранение и повышения уровня здоровья.

Рассматривая энергоэффективность медицинских учреждений, принимаются во внимание такие аспекты, как:

- непрерывность энергоснабжения. Это связано с необходимостью обеспечения жизни и здоровья пациентов и медицинского персонала;
- снижение затрат на модернизацию. Поскольку бюджет медицинских учреждений формируется из государственного, требуется обеспечить экономичное и быстро окупаемое решение;
- особые требования к электроснабжению дорогостоящего диагностического оборудования и других устройств.

Рассмотрев вышеописанные факторы, становится очевидным необходимость внедрения энергоэффективных технологий в учреждениях здравоохранения. Предлагается выделить и рассмотреть следующие категории таких технологий:

1. Светодиодное (LED) освещение играет ключевую роль в повышении энергоэффективности медицинских учреждений. Это освещение обладает множеством преимуществ по сравнению с традиционными источниками света, такими как лампы накаливания, люминесцентные и галогенные лампы. Положительные стороны такого освещения включают:

- экологическая, пожарная, электрическая безопасность;
- высокая световая отдача;
- минимальные потери потребленной энергии;
- длительный срок эксплуатации с минимальной потребностью в техническом обслуживании;
- хорошая цветопередача и возможность создания светового потока с требуемой цветовой температурой;
- возможность многократного включения и выключения, а также плавного изменения освещенности с применением дистанционных управляющих систем.

Такое освещение может найти применение в:

- операционных и процедурных кабинетах – светодиоды обеспечивают яркий и точный свет, который улучшает видимость и помогает медицинскому персоналу работать более эффективно и точно;
- палатах для пациентов – светодиодное освещение может быть настроено на различные уровни яркости, обеспечивая комфортные условия для отдыха и восстановления пациентов;

– общих и административных зонах – в коридорах, приемных, офисах и других административных зонах светодиодное освещение обеспечивает яркий и равномерный свет, улучшая видимость и безопасность, а автоматические системы управления освещением, такие как датчики движения, помогают дополнительно экономить энергию;

– экстренных и интенсивных отделений – светодиоды обеспечивают надежное и стабильное освещение, что критически важно в условиях экстренных медицинских ситуаций.

## 2. Энергоэффективное управление климатом в медицинских учреждениях

Эффективное управление климатом в больницах и клиниках имеет огромное значение для обеспечения комфортных условий для пациентов и медицинского персонала, а также для поддержания оптимальных условий для хранения и использования медицинского оборудования.

Современные системы отопления, вентиляции и кондиционирования воздуха (HVAC) используют меньше энергии по сравнению с традиционными системами, обеспечивают стабильное поддержание комфортной температуры и влажности в помещениях, что особенно важно для пациентов, включая новорожденных, пожилых людей и людей с ослабленным иммунитетом и работают тише.

### Примеры технологий и решений HVAC:

– Инверторные компрессоры. Инверторные технологии позволяют компрессорам работать на различных скоростях, что снижает энергопотребление и улучшает контроль температуры.

– Теплообменники с рекуперацией тепла. Эти устройства позволяют утилизировать тепло, выделяемое внутри здания, и использовать его для подогрева свежего воздуха, что снижает затраты на отопление.

– Системы управления климатом на базе IoT. Использование датчиков и интеллектуальных систем управления позволяет автоматически регулировать параметры, адаптируясь к изменениям температуры и влажности внутри и снаружи здания.

## 3. Энергоэффективная изоляция и строительные материалы

Энергоэффективная изоляция и использование современных строительных материалов помогают снизить теплопотери, уменьшить эксплуатационные расходы.

Современные теплоизоляционные материалы эффективно предотвращают утечку тепла зимой и сохраняют прохладу летом, что значительно уменьшает нагрузку на системы отопления и кондиционирования воздуха, обеспечивают стабильную температуру внутри помещений, что также способствует созданию комфортных условий для пациентов и персонала.

Автоматизированная система, разработанная в данном дипломном проекте, и использование медицинских данных в электронном виде поможет врачу быстро и оперативно получить необходимую информацию о пациенте, ускорит процесс принятия решения для постановки диагноза и в выборе эффективного метода лечения, что в свою очередь также увеличит энерго- и ресурсосбережение в данной области.

## Заключение

В процессе выполнения технического задания была спроектирована и разработана многопользовательская информационная система для хранения и обработки медицинских изображений и заключений, с возможностью получения предварительного заключения от нейронной сети.

Серверная часть системы, реализованная с использованием современных технологий, обеспечивает надежное хранение и обработку данных. Использование базы данных PostgreSQL позволило создать эффективное и масштабируемое хранилище медицинских изображений и заключений, обеспечивая быстрый доступ и обработку больших объемов данных.

Для анализа медицинских изображений была разработана и обучена нейронная сеть, использующая библиотеки TensorFlow и Keras. Эта нейронная сеть способна автоматически анализировать рентгеновские снимки и предоставлять предварительные заключения, что значительно ускоряет процесс диагностики и повышает точность медицинских решений.

Созданная информационная система позволяет врачам эффективно управлять медицинскими данными, загружать новые снимки и получать результаты анализа в едином интерфейсе. Система поддерживает интеграцию с различными источниками медицинских изображений, что делает ее универсальным инструментом для медицинских учреждений.

Внедрение данной автоматизированной системы позволит улучшить качество медицинского обслуживания, повысить скорость и точность диагностики, а также сократить затраты на хранение и обработку данных. Разработка показала свою экономическую целесообразность и готова к внедрению в реальную медицинскую практику.

Все этапы разработки системы были успешно завершены, включая реализацию серверной части, базы данных и нейронной сети. Система готова к эксплуатации и может значительно повысить эффективность работы медицинских учреждений, улучшая процессы диагностики и ухода за пациентами.

## Список использованных источников

- 1 Archimed+ / [Электронный ресурс]. – Режим доступа: <https://archimed.pro/> – Дата доступа 10.06.2024
- 2 Medesk / [Электронный ресурс]. – Режим доступа: <https://medesk.net/lp/ru/healthcare-crm.html/> – Дата доступа 10.06.2024
- 3 Sycret / [Электронный ресурс]. – Режим доступа: <https://sycret.ru/> – Дата доступа 10.06.2024
- 4 Masse, M. REST API Design Rulebook. – O'Reilly Media, Inc., 2011. – 116 p.
- 5 Andersson, E. Software Engineering for Internet Applications. / Greenspun P., Grumet A. – MIT Press, 2006. – 399 p.
- 6 Subramanian, H. Hands-On RESTful API Design Patterns and Best Practices. / Pethuru R. – Packt Publishing, 2019. – 378 p.
- 7 Fowler, M. Patterns of Enterprise Application Architecture. – Addison-Wesley Professional, 2002. – 560 p.
- 8 Tensorflow / [Электронный ресурс]. – Режим доступа: <https://tensorflow.org/> – Дата доступа 10.06.2024
- 9 Keras / [Электронный ресурс]. – Режим доступа: <https://keras.io/> – Дата доступа 10.06.2024
- 10 Grinberg M., Flask Web Development. – O'Reilly Media, Inc., 2018 – 316 p.
- 11 Orthanc / [Электронный ресурс]. – Режим доступа: <https://www.orthanc-server.com/> – Дата доступа 10.06.2024
- 12 Goodfellow, I. Deep Learning. / Bengio, Y., & Courville, A. – IFAC-PapersOnLine, 2015. – 800 p.
- 13 Aggarwal, C. Neural Networks and Deep Learning: A Textbook. – MIT Press, 2018. – 497 p.
- 14 Экономика и организация производства в дипломных проектах: Учеб. пособие для машиностроительных вузов / К.М. Великанов, Э.Г. Васильева, В.Д. Власов и др.; Под общ. ред. К.М. Великанова. - 4-е изд. перераб. и доп. - Л.: Машиностроение, 1986. – 285 с.
- 15 Кривой, В. И. Охрана труда: комментарий новейшего законодательства / В. И.Кривой. - Гомель: Ковалева Е. А., 2009. - 367 с. - (Библиотека практического работника).
- 16 Кузнецов, И. Н. Охрана труда: учебное пособие / И. Н. Кузнецов. - Минск: Четыре четверти, 2010. - 175 с.