**Аннотация**

**на дипломный проект**

АСОИ «Медицинская информационная система обработки изображений».

Нейронная сеть и клиентская часть.

Структура и объём проекта

Дипломный проект состоит из пояснительной записки на 65 страницы 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 вопросы охраны труда.

В седьмом разделе описываются вопросы энерго- и ресурсосбережения.

В заключении производится анализ степени выполнения требований к проектируемой системе.

**Содержание**

[Введение 7](#_Toc169592295)

[1 Анализ предметной области 8](#_Toc169592296)

[1.1 Описание объекта автоматизации 8](#_Toc169592297)

[1.2 Анализ существующей ИС 9](#_Toc169592298)

[1.3 Краткое описание, классификация и анализ существующих ИС, аналогичных разрабатываемой. 11](#_Toc169592299)

[1.4 Постановка задачи на дипломное проектирование 12](#_Toc169592300)

[2 Проектирование структуры информационной системы 15](#_Toc169592301)

[2.1 Организационно-экономическая сущность задачи 15](#_Toc169592302)

[2.2 Структура ИС 16](#_Toc169592303)

[2.3 Информационное обеспечение 19](#_Toc169592304)

[2.4 Техническое и системное программное обеспечение 21](#_Toc169592305)

[2.5 Эргономическое обеспечение 22](#_Toc169592306)

[3 Политика информационной безопасности 23](#_Toc169592307)

[3.1 Цель, принципы и задачи защиты информации в АСОИ 23](#_Toc169592308)

[3.2 Методы и средства обеспечения защиты информационных ресурсов 23](#_Toc169592309)

[3.3 Средства защиты информации и информационных ресурсов 24](#_Toc169592310)

[4 Программная реализация информационной системы 25](#_Toc169592311)

[4.1 Выбор программных средств реализации автоматизированной системы 25](#_Toc169592312)

[4.2 Разработка программного кода ИС 26](#_Toc169592313)

[4.3 Сбор и подготовка данных 28](#_Toc169592314)

[4.4 Компиляция и обучение нейронной сети 29](#_Toc169592315)

[4.5 Руководство пользователя 31](#_Toc169592316)

[5 Расчет экономической эффективности технических решений 36](#_Toc169592317)

[5.1 Общая постановка к технико-экономическому обоснованию 36](#_Toc169592318)

[5.2 Расчет трудоемкости (производительности) 39](#_Toc169592319)

[5.3 Расчет единовременных затрат 39](#_Toc169592320)

[5.4 Расчет годовых текущих издержек 47](#_Toc169592321)

[5.5 Расчет показателей экономической эффективности 52](#_Toc169592322)

[5.6 Организация внедрения системы 53](#_Toc169592323)

[5.7 Заключение по разделу 54](#_Toc169592324)

[6 Охрана труда 56](#_Toc169592325)

[6.1 Система управления охраной труда в учреждении здравоохранения «Могилевская областная клиническая больница» 56](#_Toc169592326)

[6.2 Анализ выполнения мероприятий по обеспечению безопасной эвакуации людей в учреждении здравоохранения «Могилевская областная клиническая больница» 59](#_Toc169592327)

[6.3 Выводы и предложения 60](#_Toc169592328)

[7 Энерго- и ресурсосбережение 62](#_Toc169592329)

[7.1 Энергоэффективные технологии в здравоохранении 62](#_Toc169592330)

[Заключение 64](#_Toc169592331)

[Список использованных источников 65](#_Toc169592332)

[Ведомость документов к дипломному проекту 66](#_Toc169592333)

**Введение**

В медицинской практике существует острая потребность в точном анализе рентгеновских снимков для диагностики и лечения пациентов. Однако многие медицинские учреждения еще не оценили преимуществ автоматизированных систем анализа рентгенограмм. Внедрение систем искусственного интеллекта для анализа рентгеновских изображений становится все более актуальным, так как это позволяет оптимизировать диагностические процессы, повышая их точность и скорость.

Исследование бизнес-процессов в учреждении здравоохранения «Могилёвская областная клиническая больница» выявило неэффективности и необходимость дополнительной автоматизации в работе с медицинскими изображениями. Внедрение таких систем приведет к следующим положительным эффектам:

− Повышение эффективности диагностики и управления лечением пациентов, что улучшит качество медицинского обслуживания и удовлетворенность пациентов;

− Сокращение времени на обучение медицинского персонала и упрощение анализа рентгеновских снимков для врачей и специалистов.

Важным аспектом успешного внедрения автоматизированной системы анализа рентгеновских изображений является удобный и интуитивно понятный пользовательский интерфейс (UI). Для врачей и медицинского персонала крайне важно иметь доступ к инструментам, которые легко использовать и которые не требуют длительного обучения. Хорошо спроектированный UI позволяет медицинским специалистам быстро и эффективно взаимодействовать с системой, получать необходимые данные и проводить анализ без лишних затруднений. Это особенно важно в условиях высокой нагрузки и необходимости оперативного принятия решений.

На основании этих фактов было принято решение о разработке АСОИ «Медицинская информационная система обработки изображений» в виде веб-приложения. Для реализации веб-приложения были созданы два серверных приложения на ASP.NET Core и Flask, используя языки программирования C# и Python соответственно. База данных была создана с помощью свободной объектно-реляционной системы управления PostgreSQL. Для клиентского приложения использовались язык программирования JavaScript и библиотека React.

**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.

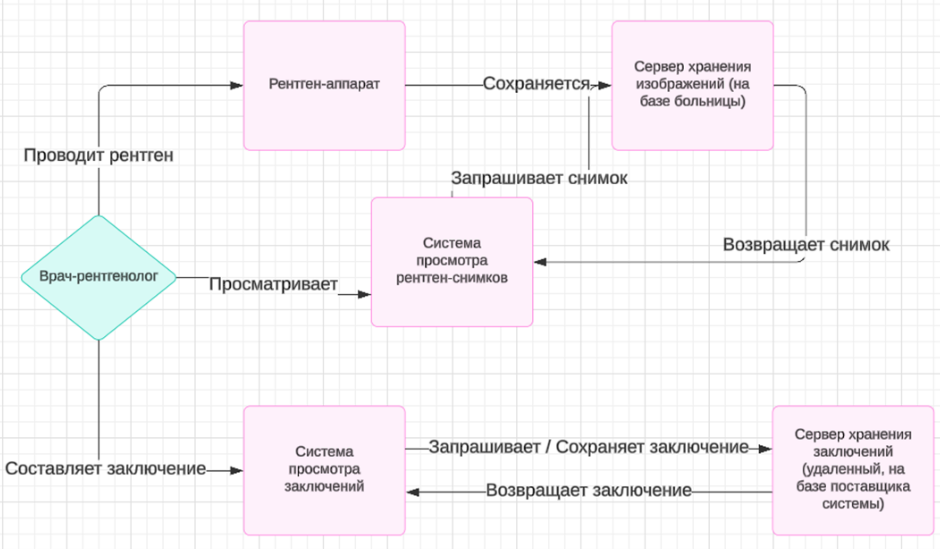


Рисунок 1.1 − Контекстная диаграмма процесса анализа и составления заключений врачом-рентгенологом

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.

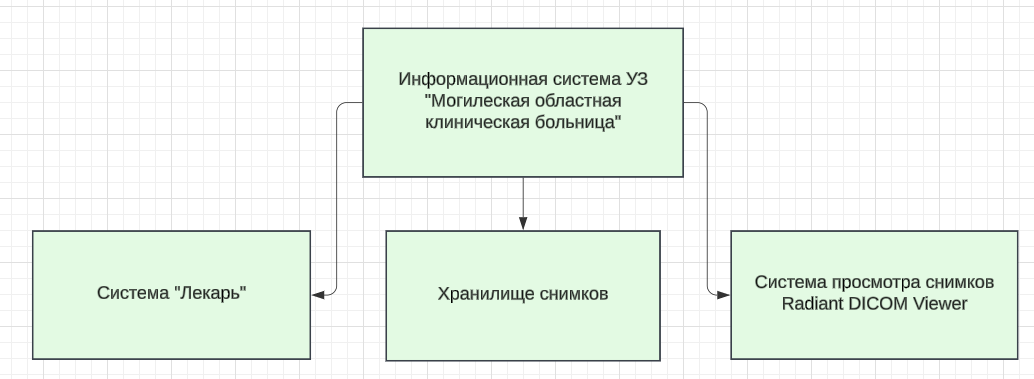


Рисунок 1.2 − Схема информационной системы больницы

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.

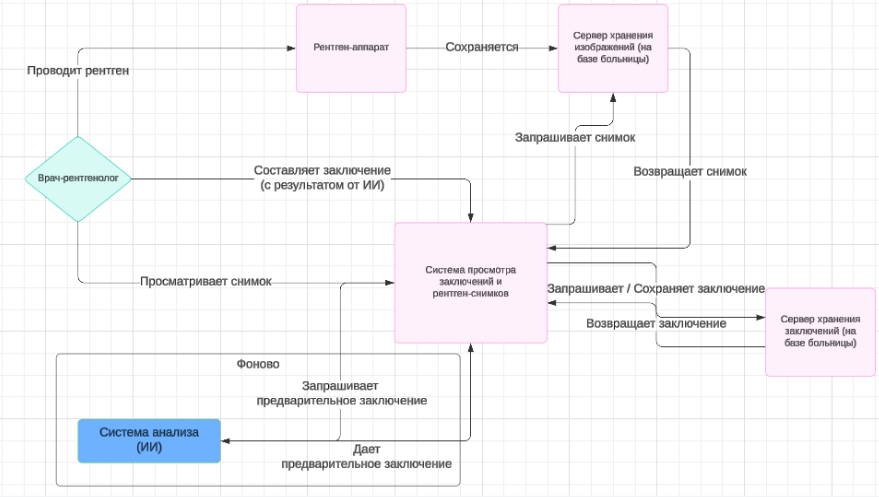


Рисунок 1.3 − Контекстная диаграмма процесса анализа и составления заключений с вводом новой системы

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 клиентской части»:

– разработать удобный веб-клиент для работы с системой (главный критерий удобности в данном случае – это отсутствие необходимости перехода по нескольким программам для работы с заключением и снимком);

– разработать механизм периодического переобучения нейронной сети на основании полученных снимков и заключений;

– 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, JavaScript, React, C#, ASP.NET, Python, Flask.

На рисунке 3.1 представлена схема вариантов использования автоматизированной системы.

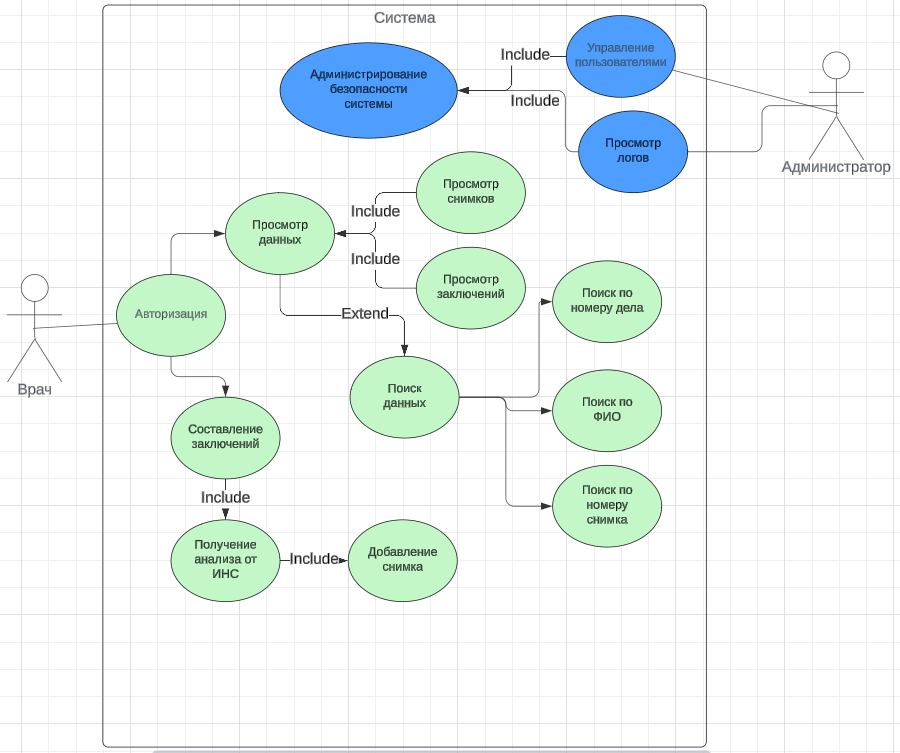


Рисунок 2.1 − Схема вариантов использования автоматизированной системы

Диаграмма вариантов использования представлена в графической части (301.1-53 01 02.10031225.007.001 ПД).

Система выполняет следующие функции:

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

– 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 систему.

**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.

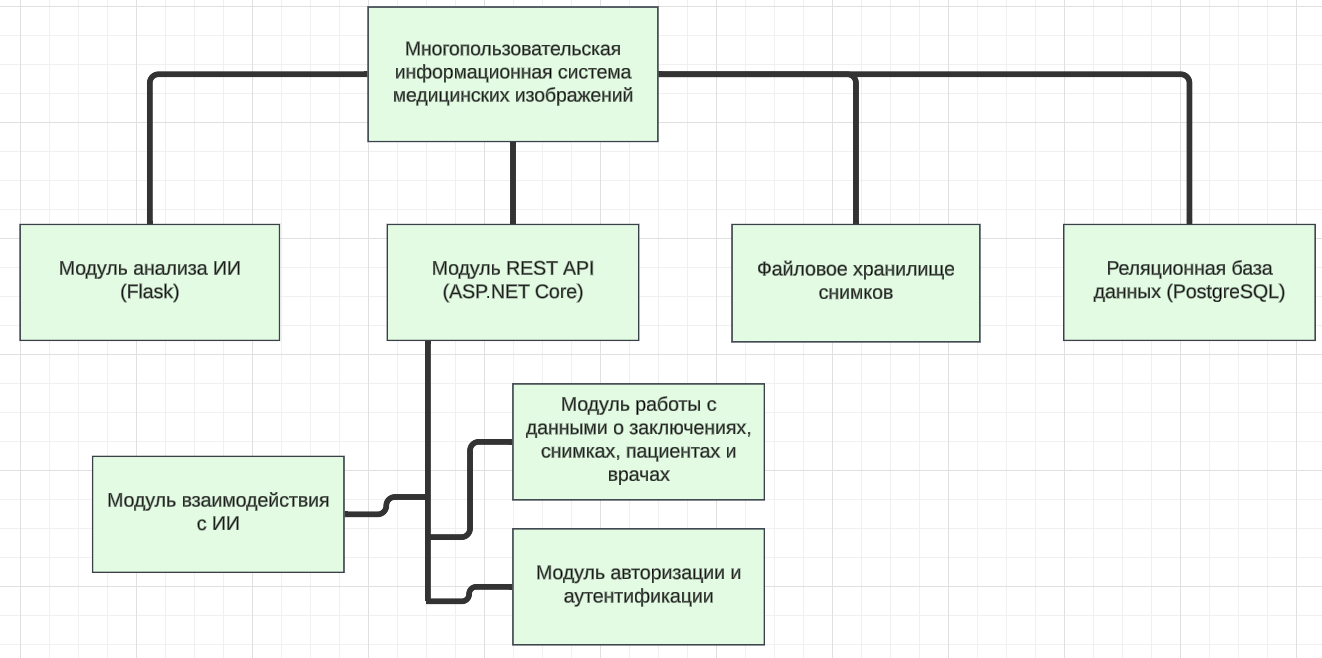


Рисунок 2.2 − Схема модулей информационной системы.

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.

Клиентская часть веб-приложения разработана с использованием JavaScript-библиотеки React, которая предназначена для создания динамичных и отзывчивых пользовательских интерфейсов. React позволяет эффективно управлять состоянием приложения и обновлять пользовательский интерфейс в ответ на изменения данных. React построен на концепции компонентов, что позволяет разделить интерфейс на независимые, переиспользуемые части. Это значительно упрощает разработку и поддержку кода, так как каждый компонент отвечает за свою часть интерфейса. React использует виртуальный DOM для оптимизации обновлений интерфейса. Виртуальный DOM представляет собой легковесную копию реального DOM. При изменении состояния приложения React обновляет виртуальный DOM, сравнивает его с предыдущей версией и вносит минимально необходимые изменения в реальный DOM, что повышает производительность приложения.

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.

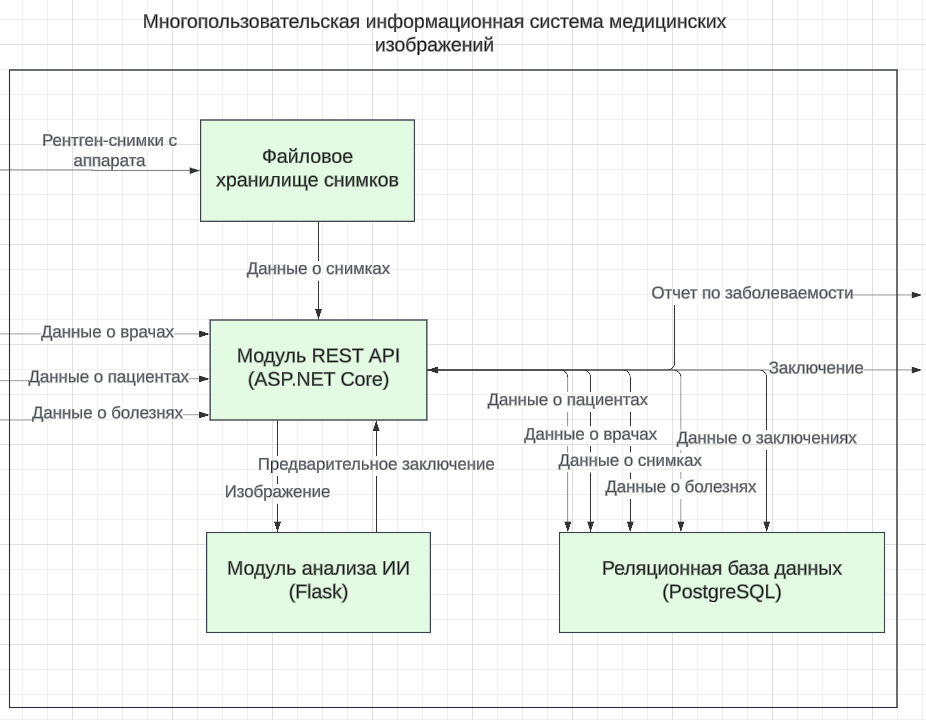


Рисунок 2.3 − Схема потоков данных

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;

На основании этого выделяются следующие рекомендации:

– все системные сообщения должны быть лаконичными и понятными, чтобы пользователь 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: авторизированные 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 хэшам. 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 пароля.

**4 Программная реализация информационной системы**

**4.1 Выбор программных средств реализации автоматизированной  
 системы**

Автоматизированная система разрабатывается c использованием языков программирования JavaScript, библиотеки React, и Python, фреймворков Flask, Tensorflow и Keras, в качестве базы данных выступает PostgreSQL.

React – это библиотека для создания динамичных и интерактивных пользовательских интерфейсов, которая используется для разработки клиентской части системы на языке программирования JavaScript. React предоставляет возможность разрабатывать компонентно-ориентированные приложения, что упрощает управление состоянием и повторное использование кода.

Дополнительно, в системе применяются следующие технологии:

– axios: библиотека для выполнения HTTP-запросов, позволяющая взаимодействовать с серверной частью, получать и отправлять данные, а также обрабатывать ответы сервера.

– styled-components: библиотека для стилизации компонентов в React, которая позволяет писать стили на уровне компонентов, обеспечивая модульность и гибкость в разработке пользовательского интерфейса.

– react router: библиотека для маршрутизации в приложениях на React, которая позволяет создавать одностраничные приложения с поддержкой навигации.

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.

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 взаимодействуют 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 Разработка программного кода ИС**

Компонент React состоит из трех основных частей: JSX (шаблон), CSS (стили) и JavaScript (логика). Например, App.js, App.css и дополнительные модули вместе составляют App компонент. Эта структура объединяет логику, представление и стили, что делает приложение более масштабируемым и удобным в обслуживании.

Компонент App является входной точкой в приложение React. Этот компонент конфигурирует приложение, настраивает библиотеки, которые приложение будет использовать, устанавливает компоненты для обработки навигации и стилей приложения, а также контролирует состояние и взаимодействие с пользователем.



Рисунок 4.1 – Пример кода компонента App

В архитектуре React центральное место занимают компоненты. Когда компонент получает данные (props), он обрабатывает их и возвращает результат в виде JSX, который затем отображается в пользовательском интерфейсе. Для управления состоянием и логикой компонентов используются хуки, такие как useState и useEffect.

Для обеспечения непрерывности пользовательского опыта и безопасности информационной системы был реализован модуль авторизации с использованием LocalStorage. Этот метод позволяет сохранять данные аутентификации пользователя непосредственно в браузере, что исключает необходимость повторного ввода учетных данных при каждом новом посещении сайта. Процесс авторизации включает следующие ключевые этапы:

– при успешном входе в систему, данные пользователя, такие как токен доступа, сохраняются в LocalStorage, что позволяет приложению запоминать пользователя даже после закрытия браузера;

– создание функций, которые управляют процессом сохранения и удаления данных из LocalStorage;

– разработка механизма проверки наличия данных аутентификации в LocalStorage при инициализации приложения для автоматического входа пользователя.



Рисунок 4.2 – Метод авторизации пользователя

Интеграция с нейронной сетью осуществляется через API, предоставляемый серверной частью. Клиент отправляет POST-запросы с данными пользователя для обработки нейронной сетью, которая, в свою очередь, возвращает результаты, используемые для улучшения пользовательского опыта.

Процесс включает в себя:

– определение сервиса для взаимодействия с API нейронной сети;

– реализация функций для отправки данных на сервер и обработки полученных результатов;

– интеграция сервиса в компоненты React, где необходимо использовать результаты работы нейронной сети.



Рисунок 4.3 – Метод отправки изображения нейронной сети

**4.3 Сбор и подготовка данных**

Для начального обучения сети был взят свободно распространяемый набор данных от Иорданского Университета Науки и Технологий из 260 снимков спины со сколиозом и без.

Данные были распределены по соответствующим классам, и приведены к общему виду и размеру (224, 224).

**4.4 Компиляция и обучение 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.

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

Компиляция модели в TensorFlow/Keras - это процесс настройки нескольких параметров модели перед обучением. Этот шаг является одним из ключевых этапов в построении нейронных сетей и включает в себя следующие ключевые компоненты:

Оптимизатор решает задачу оптимизации параметров модели, чтобы минимизировать функцию потерь. Он определяет метод обновления весов модели на каждом шаге обучения. Для обучение данной сети используется оптимизатор Adam.

Adam (Adaptive Moment Estimation) - это метод оптимизации, который является одним из наиболее популярных и эффективных алгоритмов для обучения нейронных сетей. Adam комбинирует две основные идеи из других методов оптимизации: момент (momentum) и адаптивный шаг градиента (RMSprop). Момент (momentum) помогает ускорить обучение и стабилизировать процесс градиентного спуска, добавляя к текущему обновлению весов часть предыдущего обновления. RMSprop адаптивно настраивает скорость обучения для каждого параметра, учитывая среднеквадратичное значение градиента.

Функция потерь определяет меру ошибки между предсказанными значениями модели и истинными значениями целевой переменной. Она является ключевым оптимизируемым критерием. Модель использует функцию Бинарной кросс-энтропии. Бинарная кросс-энтропия – это функция потерь, которая часто используется в задачах бинарной классификации, когда выход модели должен предсказывать два класса (например, 0 или 1, да или нет). Эта функция потерь оценивает разницу между предсказанными вероятностями классов и истинными метками (0 или 1). Она вычисляет, насколько предсказанные вероятности отличаются от истинных меток и генерирует числовое значение ошибки.

Метрики используются для оценки производительности модели во время обучения и тестирования. Например, точность (accuracy) – это распространенная метрика для задач классификации, которая измеряет долю правильных предсказаний модели. Модель использует бинарную точность.

После компиляции модель обучается в течении 20 эпох (эпоха – цикл обхода одного обучающего примера, после которого модель обновляет свои веса). Далее идет процесс, называемый тонкой настройкой (Fine-Tuning). Для него размораживаются исходные веса базовой модели, и она снова переобучается, но с гораздо более сниженной скоростью и меньшим количеством эпох. Так можно сохранить классифицирующие способности исходной модели, но при этом более тонко подогнать их под конкретную проблему.

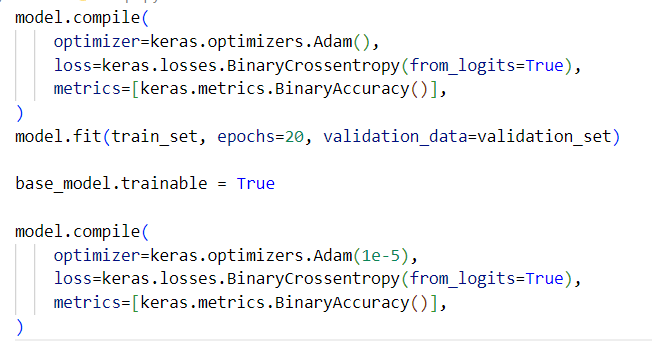


Рисунок 4.4 – Компиляция и обучение сети

На рисунках 4.5 и 4.6 указаны полученные метрики точности и функции потерь соответственно.

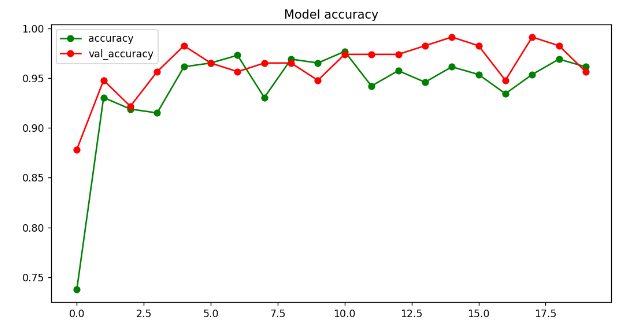


Рисунок 4.5 – Метрика точности

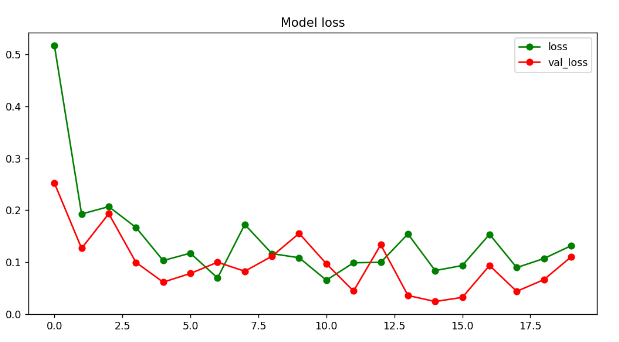


Рисунок 4.6 – Метрика потерь

**4.5 Руководство пользователя**

Это руководство предназначено для обеспечения плавной и эффективной работы с медицинским приложением, учитывая различные уровни доступа и функциональные возможности каждой роли пользователя.

Клиентская часть информационной системы состоит из следующих страниц:

страница «Администратор»;

страница «Доктор»;

страница «Логин»;

страница «Пациент»;

страница «Анализ» снимка;

Все пользователи должны войти в систему через **страницу** «**Логин**», используя свои учетные данные (логин и пароль). После входа в систему, в зависимости от роли пользователя, откроется соответствующий интерфейс.



Рисунок 4.7 – Страница авторизации

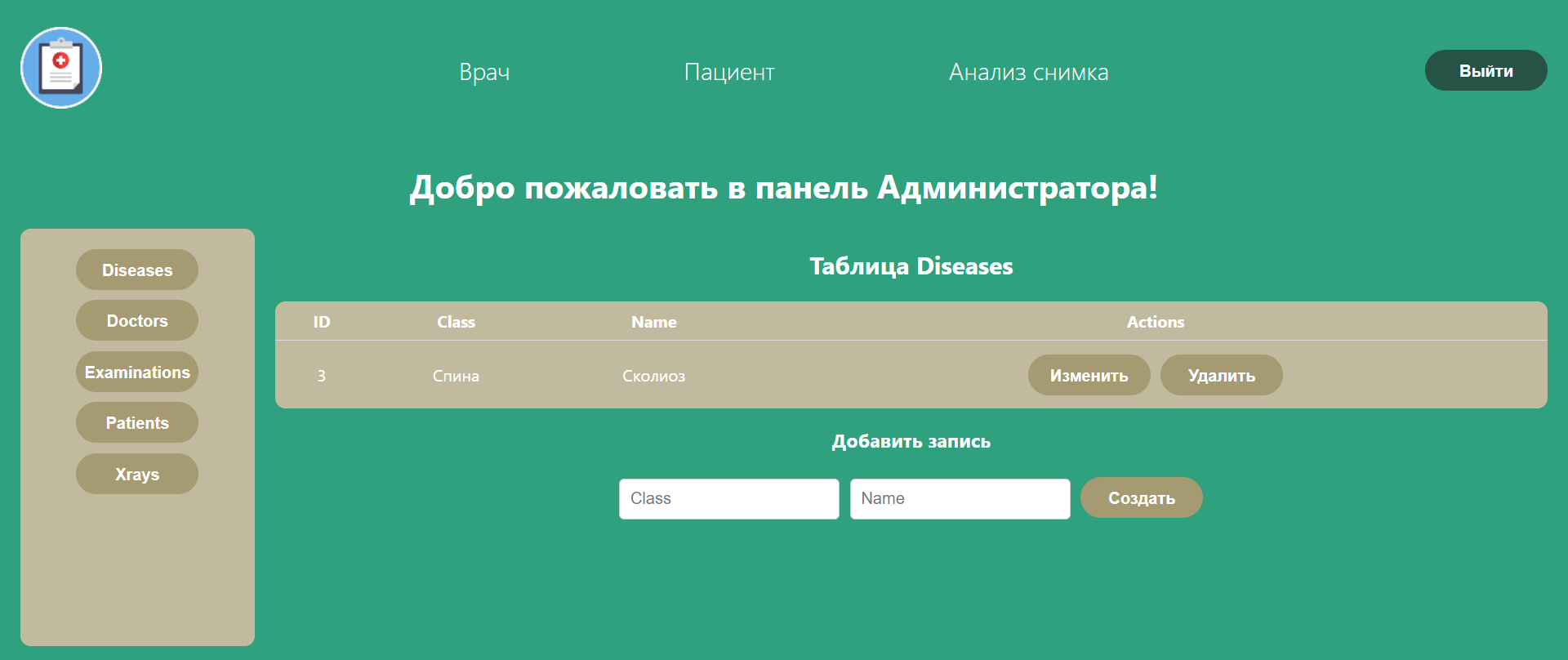


Рисунок 4.8 – Страница администратора

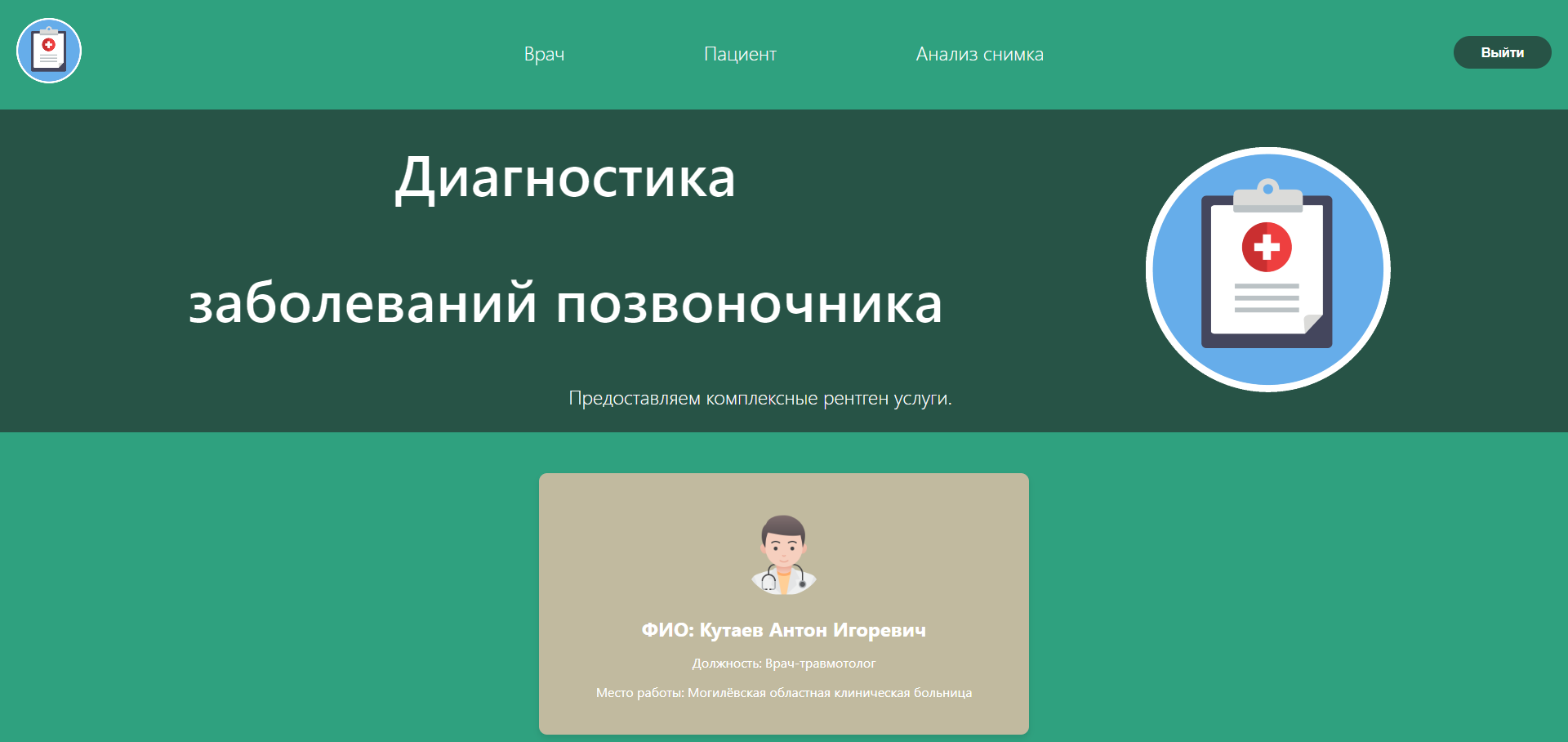


Рисунок 4.9 – Страница доктора

Администраторы могут добавлять, редактировать или удалять данные докторов, пациентов, заключений, заключений.

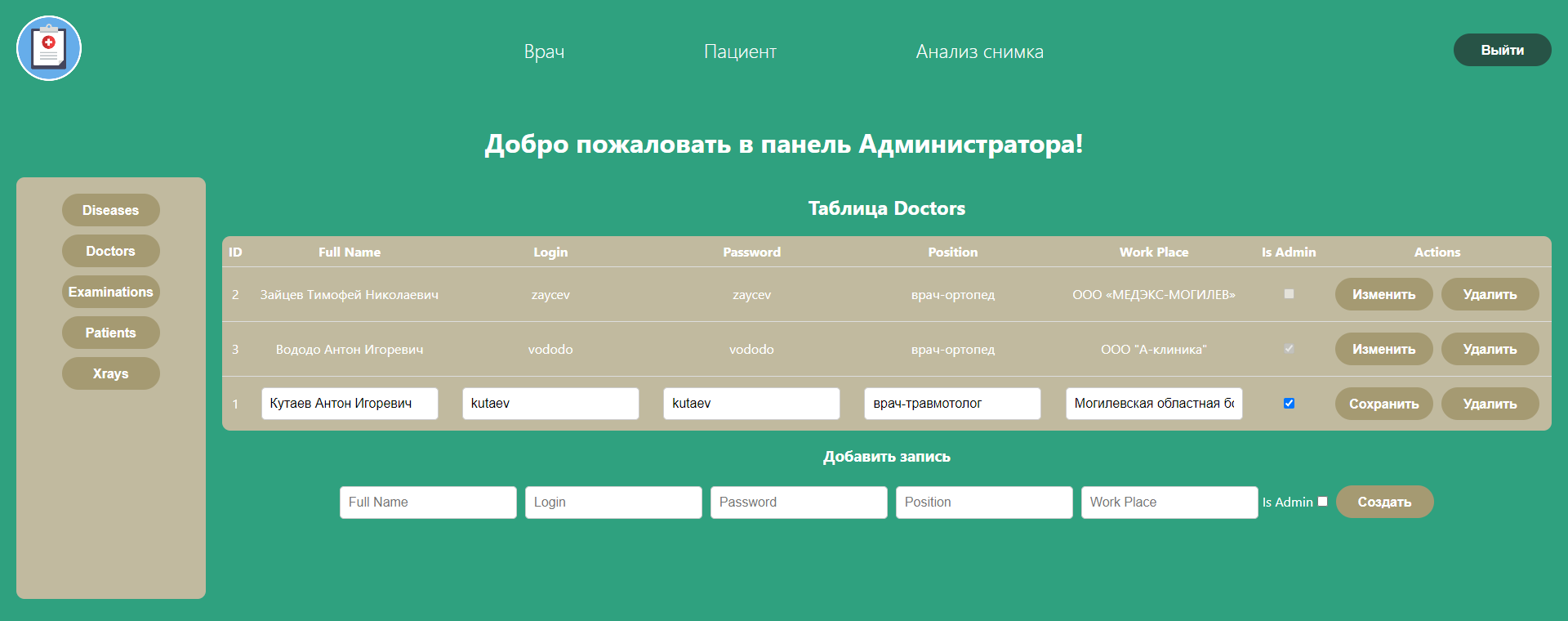


Рисунок 4.10 – Изменение данных доктора

Также у администратора есть весь функционал, который доступен доктору без прав администратора и будет описан ниже.

Страница «Врач» предназначена для и является стартовой страницей приложения, где доктор видит название системы и данные о себе.

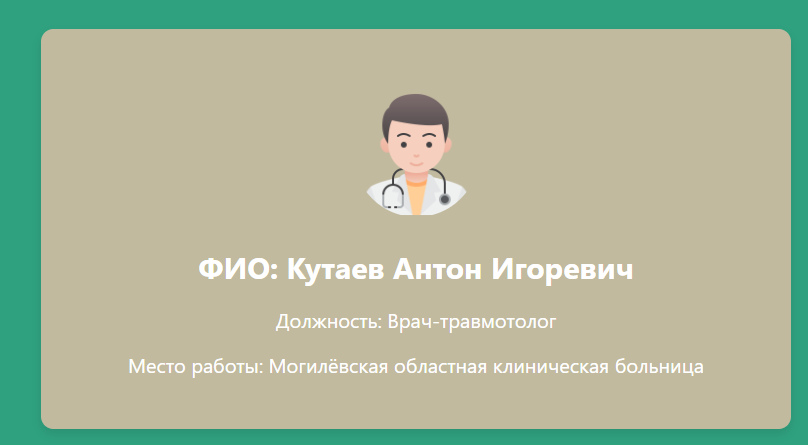


Рисунок 4.11 – Карточка с информацией о докторе

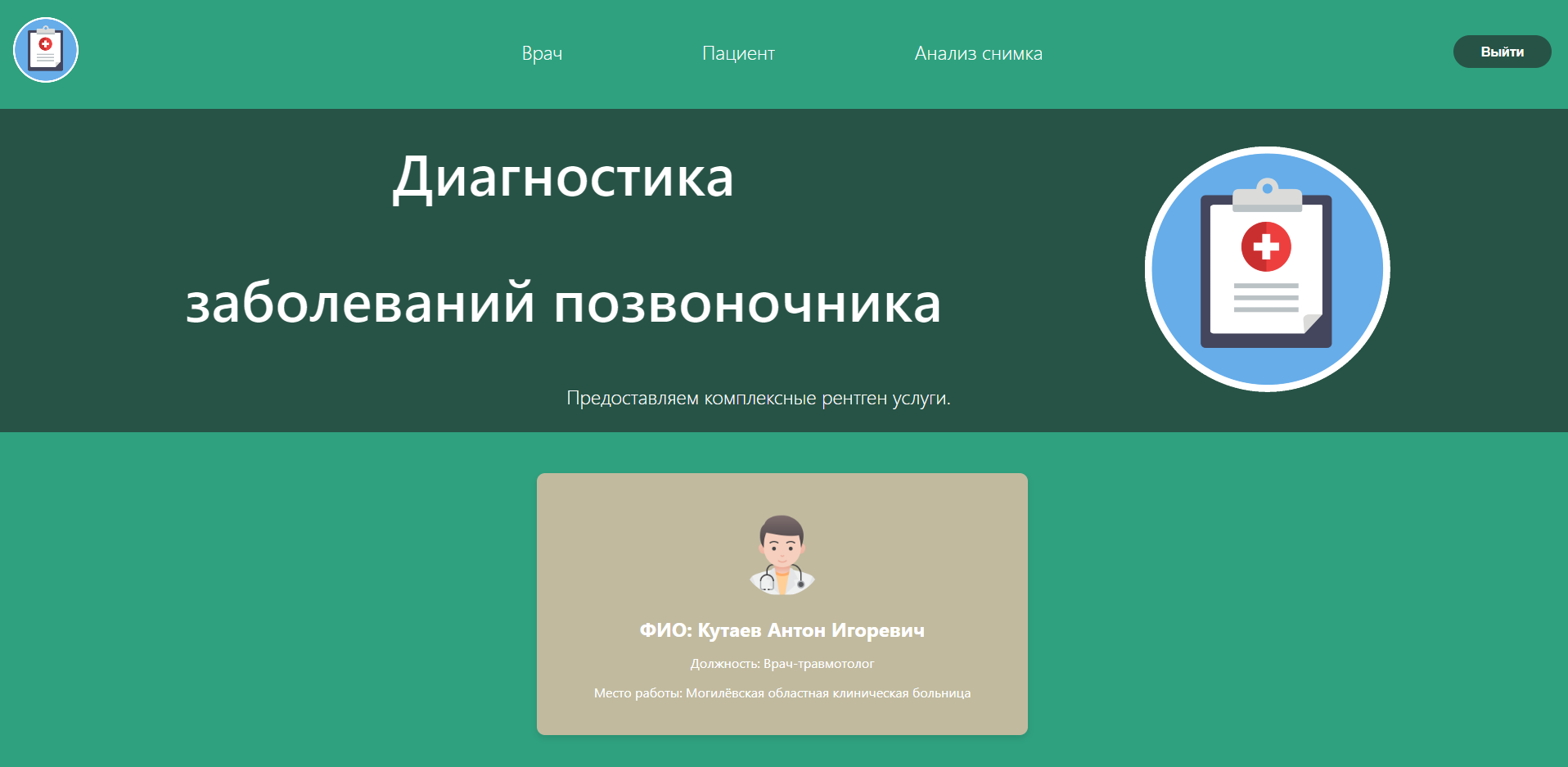


Рисунок 4.12 – Страница «Врач»

На странице Пациент доктор может просматривать данные о пациентах, искать определенного пациента.

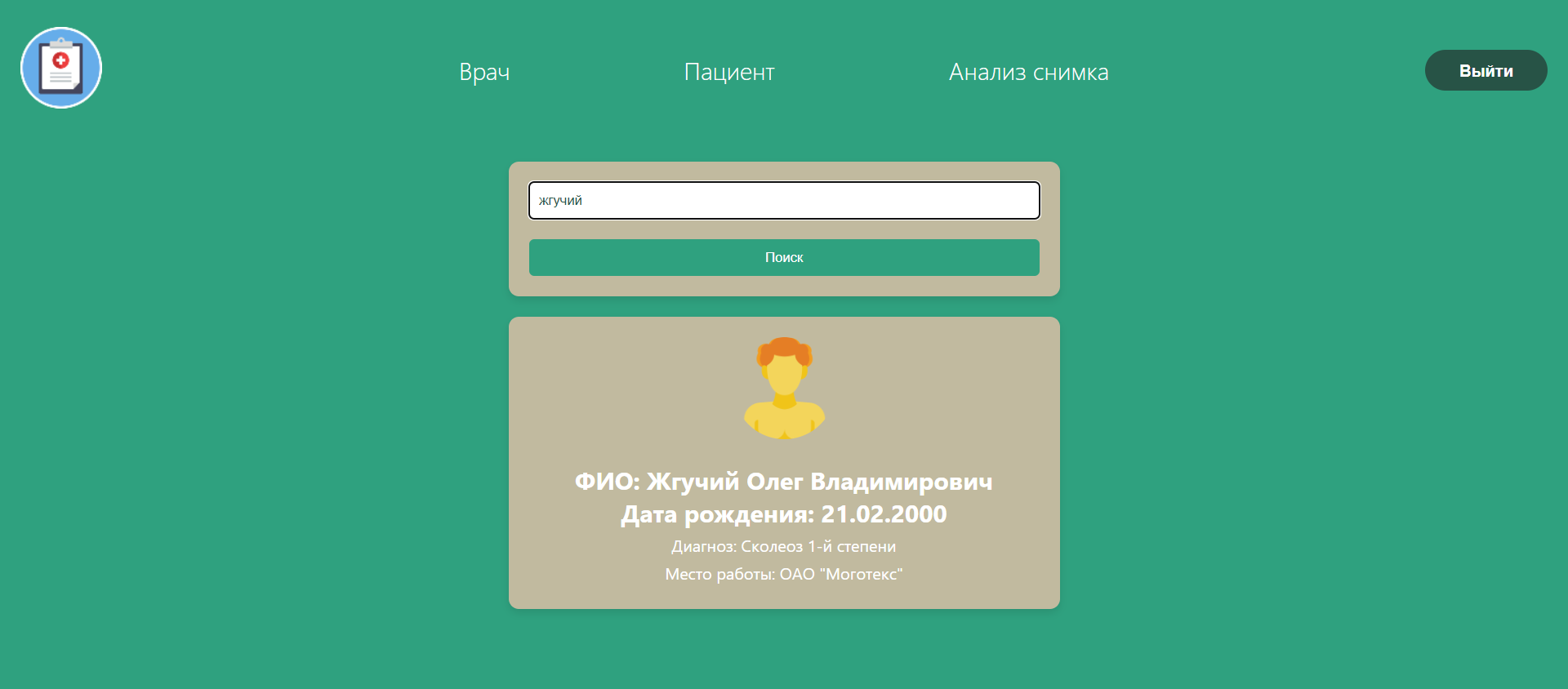


Рисунок 4.13 – Пример поиска пациента по ФИО

На странице «Анализ» снимка доктор имеет возможность загружать и анализировать медицинские изображения.

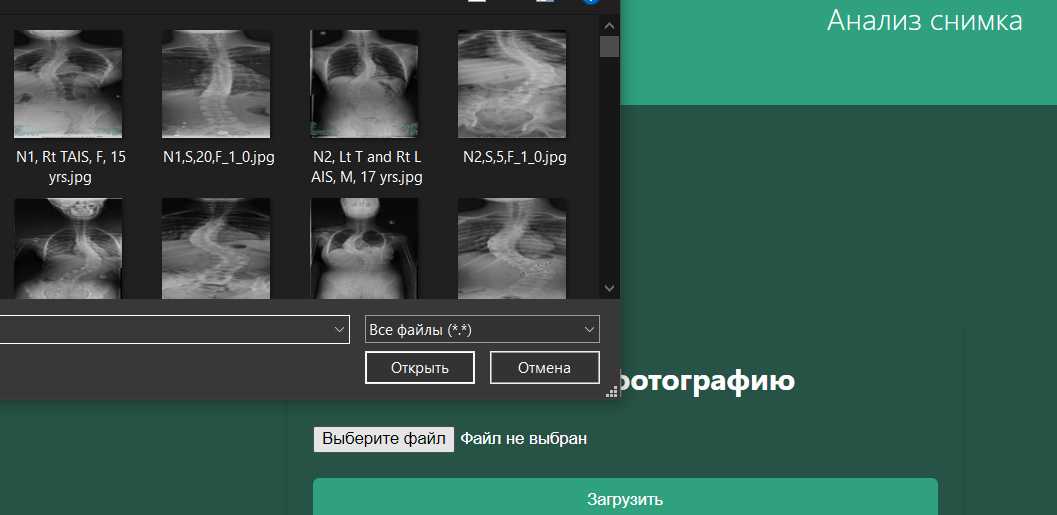


Рисунок 4.14 – Пример выбора снимка для анализа

Выбрав снимок, и нажав на кнопку «Загрузить», запустится процесс анализа изображения, после завершения анализа нейронная сеть выдаст результат.

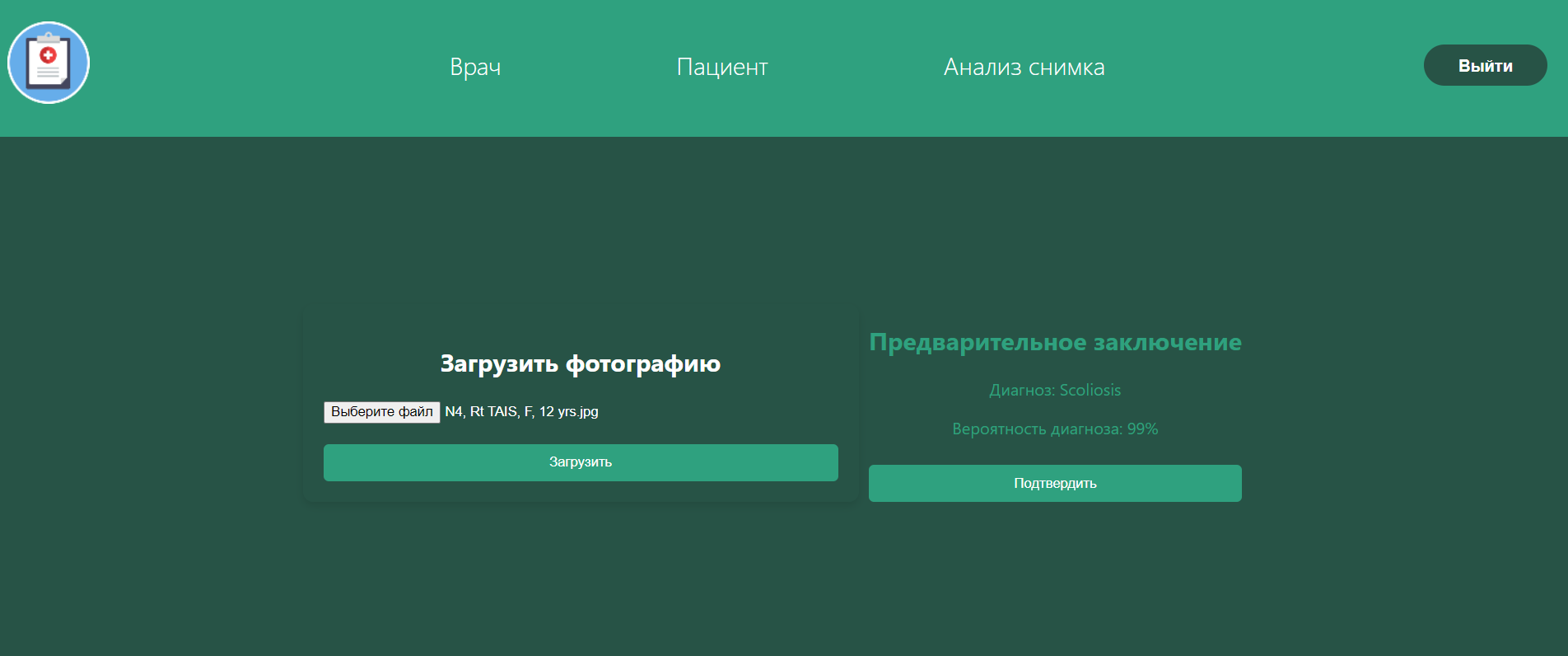


Рисунок 4.15 – Результат работы нейронной сети

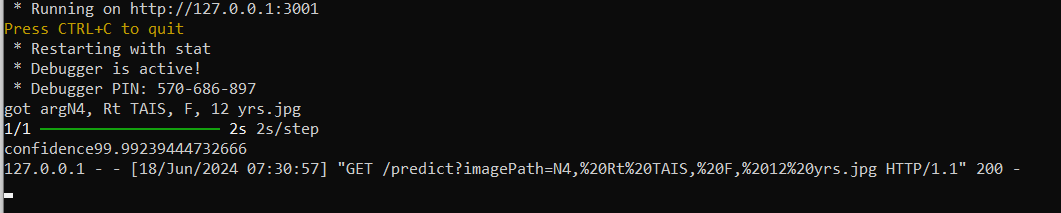


Рисунок 4.16 – Терминал нейронной сети в режиме диагностики

После, при нажатии кнопки «Подтвердить», откроется модальное окно, в котором доктор может указать пациента, и свое окончательное заключение.

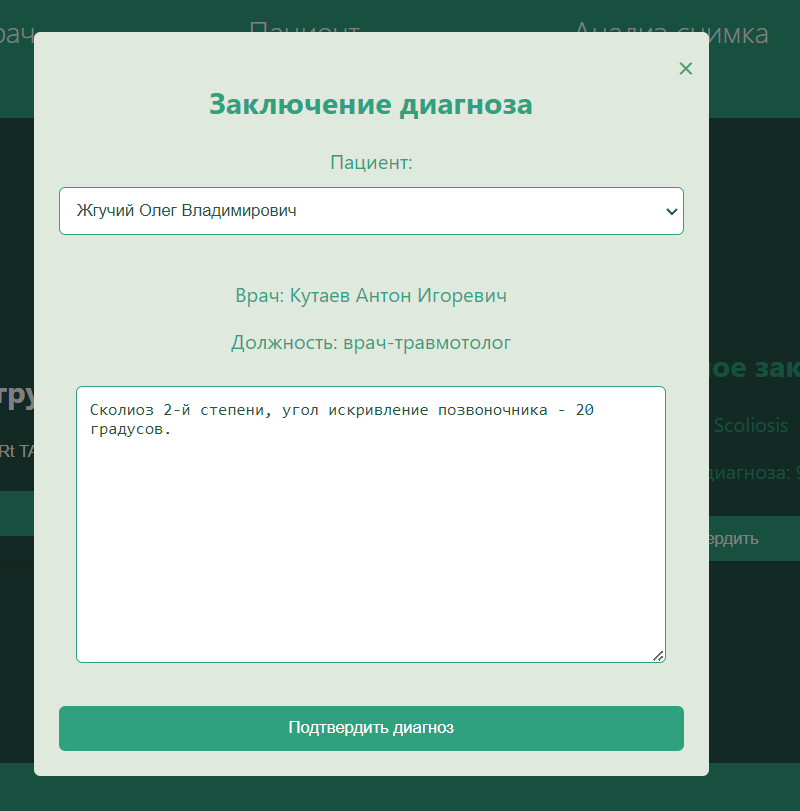


Рисунок 4.17 – Модальное окно итогового заключения

После нажатия на кнопку «Подтвердить диагноз», заключение отправится на сервер для дальнейшего хранения.

**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 клиентской 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.

Дипломный проект направлен на создание системы, которая обеспечивает автоматизированную обработку и архивацию медицинских снимков. Основные задачи включают интеграцию систем для работы с изображениями и медицинскими заключениями, упрощение процедур создания, регистрации и поиска медицинских данных, разработку интерфейса для пользователей, а также создание искусственного интеллекта, который может автоматически формировать первичные диагнозы на основе рентгеновских снимков позвоночника.

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 C, 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 | 840 строк программного кода, 400 строк автогенерируемого кода |
| Исходный язык | Объектно-ориентированные (JavaScript, 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:

где tПЗ – подготовительно-заключительное время на партию решаемых задач;

nП – количество последовательно решаемых задач за один прогон;

tОП – оперативное время выполнения задачи (сумма основного и вспомогательного не перекрываемого времени);

tОБ – время обслуживания рабочего места;

tОТЛ – время на отдых и личные надобности.

Время tОБ и tОТЛ чаще определяется косвенно как доля от оперативного времени tОП 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 (tшк ), мин | |
| --- | --- | --- |
| базовый | проектный |
| Заключения по результатам обработки рентгеновских снимков | | |
| Подготовительно-заключительное время, tПЗ | 2,00 | 2,00 |
| Оперативное время, tОП | 220 | 80,0 |
| Время обслуживания, tОБ | 26,4 | 9,6 |
| Время на отдых и личные надобности, tОТЛ | 33 | 12 |
| Итого на задачу | 281,4 | 103,6 |

Годовая программа Аг по задаче проведения медицинских исследований в виде рентгена и компьютерной томографии принята на уровне среднего количества 50 пациентов, принимаемых в кабинете рентгена в мес яц (Аг = 580 исследований в год).

**5.3 Расчет единовременных затрат**

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

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

где KOБ – стоимость запасов в оборотные средства, р.;

где KЗД – стоимость потребной площади здания, р.;

где KПР– затраты на проектирование, р.

Определение стоимости единовременных затрат в оборудование:

где принятое число единиц i-го оборудования (Nпi ≥ NРi – округляется до целого), шт;

РОi - цена приобретения i-го оборудования по варианту, р.;

αТi, αМi – коэффициенты, учитывающие величину транспортно-заготовительных расходов (αТi = 0,05), величину затрат на монтаж и отладку (αМi = 0,05);

dЗ - доля занятости принятых рабочих мест на решение задачи по варианту, dЗ = NР / NП.

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

где tшт- норма штучного времени;

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

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

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

где FСМ – номинальный сменный фонд работы, ч (8 часов);

где КСМ – коэффициент сменности – число смен работы в течение рабочего дня (2 смены);

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

где КПР – коэффициент, учитывающий долю времени простоев в плановых ремонтах, КПР = 0,05.

Получим расчетное количество рабочих мест:

Определим принятое количество рабочих мест путем округления их расчетной величины Nр до ближайшего целого числа в большую сторону:

, = 1.

Соответственно, доля занятости принятых рабочих мест:

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 5I12400D16SD24VGALW50 | 1 | 1 306,00 | 1 306,00 |
| Монитор Philips 220V8LL/62 | 1 | 331,20 | 331,20 |
| Принтер Deli P2500DN | 1 | 399,00 | 399,00 |
| Итого | - | - | 2036,20 |

Определим стоимость единовременных затрат на оборудование.

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

где РМj – цена приобретения j-го материала, используемого при решении задачи по варианту, р.;

ZМj - 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 «Xerox Performer», формат А4 (500 листов) | 16,60 | 4 | 16,60 | 2 |
| 2. Тонер НР 1.1 1020, 1022 (©2612А) | 62,00 | 1 | 62,00 | 1 |

Определена стоимость оборотных средств по вариантам

,

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

где  норматив площади производственной (6-10 м2 на одно рабочее место), служебно-бытовой (5 – 7 м2 на одного оператора);

 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.

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.

Затраты на проектирование определяются как сметная стоимость работ (постановка задачи и ее моделирование, программирование, создание информационного обеспечения длительного пользования, отладка и внедрение разработанной системы) по формуле:

где РПР – сметная ставка одного человеко-месяца проектирования, р.;

ТПР  – трудоемкость проектирования, чел.-мес.;

Дi и Дi+1 – величина дефектности для исходного уровня качества: по базовому варианту – i σ и проектируемому – (i+1)σ;

КВД и КНД – коэффициент уровня трудовых затрат на устранение выявленных и не выявленных дефектов;

– уровень выявления дефектов в программном изделии в процессе проведения тестирования.

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

где ЗТ – месячная тарифная ставка первого разряда, ЗТ =260 р.;

КТ – тарифный коэффициент проектировщика, КТ =2.65;

КП – коэффициент премирования, КП = 1,4;

КД – коэффициент, учитывающий дополнительную заработную плату, КД= 1,1;

КСС – коэффициент, учитывающий отчисления на социальные нужды, КСС= 1,346;

КНР – коэффициент, учитывающий накладные расходы, КНР = 1,15.

р.

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



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

;

KLOC – количество тысяч строк в программном продукте без учета числа строк, полученных в результате автоматического генерирования кодов, KLOC = 0,840 тыс. строк.;

МР – 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 Mi по таблице 5.8);

Tauto – затраты на автоматически генерируемый программный код, чел.-мес.



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

AT – процент автоматически генерируемого кода AT = 48,13%;

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

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

Таблица 5.6 – Характеристика масштабных факторов

| Масштабный фактор Wi | Оценка |
| --- | --- |
| Предсказуемость PREC | 2 |
| Гибкость разработки FLEX | 1 |
| Разрешение архитектуры риска RESL | 2 |
| Связанность группы TEAM | 3 |
| Зрелость процесса PMAT | 1 |
| Итого | 9 |

Коэффициент B на основании экспертных оценок равен:

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

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

Определим трудоемкость проектирования ПИ по формуле

Таблица 5.7 – Факторы затрат конструктивной модели стоимости

| Название Мi – 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 ≤ D / P < 1000 | 1,00 |
| 3. Сложность модуля в зависимости от области применения – CPLX | Низкий | Несложная вложенность структурированных операторов. Простые предикаты. | 0,9 |
| 4 Требуемая повторная используемость – RUSE | Низкий | Нет | 0,93 |
| 5. Документирование требований жизненного цикла (ЖЦ) – DOCU | Номинальный | Оптимизированы к требования жизненного цикла | 1 |
| 6. Ограничение времени выполнения платформы – TIME | Номинальный | Использование ≤ 50 % возможного времени | 1 |
| 7. Ограничение оперативной памяти платформы - STOP | Номинальный | Использование ≤ 50 % доступной памяти | 1 |
| 8. Изменчивость платформы – PVOL | Низкий | Значительные изменения – каждые 12 мес., незначительные – каждый мес. | 0,85 |
| 9. Возможность аналитика - ACAP | Номинальный | 55% | 0,88 |
| 10. Возможность программиста – PCAP | Низкий | 35% | 1,12 |
| 11. Опыт работы с приложениями – AEXP | Номинальный | 1 год | 1 |

Продолжение таблицы 5.7

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 12. Опыт работы с платформой – PEXP | Номинальный | 1 год | 1,12 |
| 13.Опыт работы с языком и утилитами – LTEX | Номинальный | 1 год | 1 |
| 14. Использование программных утилит – TOOL | Очень низкий | Редактирование, кодирование, отладка | 1,22 |
| 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 изделиях, 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 КП, (КП = 2.5) по формуле:

Качество разрабатываемого ПИ с позиций требований потребителя оценивается из условия, что распределение вероятностей строк кода размером в KAELOC, содержащих дефекты и принятых за случайные величины, подчиняются нормальному закону распределения. Соотношение поля допуска с полем разброса (в «сигмах») связывают с числом дефектов на единицу объема ПИ размером KAELOC (Дi). В данном случае уровень качества в базовом варианте – 4σ (Дi = 6,210), а проектируемом – 5σ (Дi+1 = 0,233).

В соответствии с объемом строк KAELOC (до 8) в ПИ определен КВД=1.5, КН.Д=3.5, КУД =0,75.

Определим затраты на проектирование:

Таблица 5.8 – Единовременные затраты по вариантам

|  |  |  |
| --- | --- | --- |
| Наименование элементов единовременных затрат | Величина по элементам, р. | |
| Базовый | Проектный |
| Стоимость комплекта оборудования | 1870,25 | 687,62 |
| Стоимость запасов в оборотные средства | 128,40 | 95,2 |
| Стоимость потребной площади здания | 6903,78 | 2538,28 |
| Затраты на проектирование | - | 4766,46 |
| Итого единовременных затрат К | 8902,43 | 8087,56 |

Единовременные затраты внедрения программного продукта в проектном варианте ниже на 814,87 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:

где ИЗП – годовые затраты на заработную плату операторов (системных администраторов) с начислениями, р.;

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

ИЭ – годовые затраты на силовую электроэнергию, р.;

ИРО – годовые затраты на ремонт и содержание оборудования, р.;

ИРЗ – годовые затраты на ремонт и содержание зданий, р.;

ИНР – 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



где tШКi - норма штучного времени по i-ой операции, ч;

ЗТ - часовая тарифная ставка первого, ЗТ =1.55 р.;

КТi - тарифный коэффициент разряда по i-ой операции КТ10=2.48;

КПi – коэффициент премирования по i-ой операции, КПi = 0,3;

КД – коэффициент, учитывающий дополнительную заработную плату, КД  = 0,1;

КСС – коэффициент, учитывающий отчисления на социальные нужды, КСС = 0,346;

КН – коэффициент, учитывающий налоги на заработную плату, КН = 0,1.

Годовые затраты на материалы рассчитываются по формуле:



где РМj и РОТj – цена приобретения, используемого j-го материала и реализуемых отходов, р.;

НМj и НOTj – норма расхода i-ых видов материала (листы бумаги, формы документов, картриджи и т.д.) и реализуемых отходов на решаемую задачу, шт. (кг);

АГ – 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 НР 1.1 1020, 1022 (©2612А), 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 22,59 р. и 16,31 р.

Определим годовые затраты на материалы:

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



где FД – годовой действительный фонд работы единицы оборудования, ч;

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

PЭ – цена (тариф) за один киловатт-час потребляемой электроэнергии (РЭ = 0,33734 р./кВт · ч.

Таблица 5.9 – Потребляемая мощность оборудования

|  |  |
| --- | --- |
| Оборудование | Потребляемая мощность, кВт |
| Системный блок Jet Office 3i10100D8SD24VGALW50 | 0.650 |
| Монитор HKC MB20S1FS | 0.130 |
| Принтер Pantum P2516 | 0.070 |
| Итого | 0.850 |

Определим годовые издержки на потребляемую электроэнергию:

,

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

где КО – стоимость используемого оборудования, р.

Определены годовые издержки на ремонт и содержание оборудования:

Годовые затраты на ремонт и содержание зданий:

где НРЗ – норматив на ремонт и содержание здания НРЗ = 2,5 %;

КЗД – стоимость используемых зданий, р.

р.,

р.

Годовые накладные расходы складываются из следующих статей затрат: на управление и обслуживание производства ИУ, освещение ИОС, воду на бытовые нужды ИБВ, тепловой энергии на горячую воду ИГВТЭ, отопление ИОТТЭ, вентиляцию ИВТТЭ:

Годовые расходы на управление и обслуживание производства определяются по формуле:

где ККУ – коэффициент, учитывающий косвенные расходы по управлению ККУ = 0,2;

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



где WS – норма освещенности WS = 0,03, кВт/м2;

S – площадь зданий производственных и служебно-бытовых, S=14м2;

FO  – годовой осветительный фонд времени FO = 2400 ч при двусменной.

Годовые затраты воды на бытовые нужды:

где – цена воды на бытовые нужды = 4,4086 р./ м3;

НБВ – норма расхода воды на бытовые нужды за сутки на одного работникаНБВ = 0,025 м3;

ЧР – численность работников (врачей – ренгенологов), ЧР = 1 чел.

= 4,4086 ∙ 0,025 ∙ 252 = 27,78 р.

Годовые затраты тепловой энергии на горячую воду рассчитываются по формуле:

где РТЭ – цена (тариф) за теплоэнергию РТЭ = 151,1356 р./Гкал;

qВТХ – удельная тепловая характеристика воды qВТХ = 1, ккал/(м3 ч оС);

VВГ – объем потребления горячей воды за час VВГ = 3 л;

tВГ, tВХ – температура воды горячей в системе tГВ = +65 оС, воды холодной - tВХ = +5 оС;

FВГ – период теплоснабжения водой горячей:

=

Годовые затраты тепловой энергии на отопление рассчитываются по формуле:

,

где qЗДТХ – удельная тепловая характеристика здания qЗДТХ = 0,40 ккал/(м3 ч оС);

VЗД – объем помещения здания по наружному обмеру VЗД = SH, Н= 3,5м;

tЗДВН, tЗДН – температура воздуха внутри помещения и снаружи соответственно tЗДВН = + 20, tЗДН = - 10, оС;

FОТ – отопительный период за год FОТ = 4320 ч, ч.

Годовые затраты тепловой энергии на вентиляцию рассчитываются по формуле:

=

где – удельная тепловая характеристика вентиляции здания = 0,15 ккал/(м3 ч оС);

– температура воздуха вытяжного и снаружи соответственно = +20 оС, = - 1,5 оС;

FВТ – период работы вентиляционной системы за год FВТ =1400 ч;

– коэффициент, учитывающий потери теплоэнергии, = 1,18.

Определены годовые накладные расходы:

р.,

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 | 20126,36 | 7409,70 |
| Затраты на материалы | 145,99 | 108,51 |
| Затраты на силовую электроэнергию | 971.26 | 303,47 |
| Затраты на ремонт и содержание оборудования | 358,57 | 154,72 |
| Затраты на ремонт и содержание зданий | 172,59 | 63,46 |
| Накладные расходы | 2059,81 | 1248,38 |
| Итого годовых текущих издержек И | 23834,58 | 9179,73 |

Годовые текущие издержки в результате внедрения программного продукта снизились на 14654,85 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:

=,

где ЗБГ, ЗПГ – годовые приведенные затраты по базовому и проектному варианту.

Величина приведенных затрат по сравниваемым вариантам определяется по формуле:

где EH – нормативный коэффициент эффективности, EH = 0,1;

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

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



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 tСЛ  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:

р.,

.,

Определен годовой результирующий эффект:

По результатам расчетов в данном дипломном проекте единовременные затраты по проектному варианту ниже затрат по базовому, соответственно расчет срока окупаемости дополнительных единовременных затрат не производятся. 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 – Технико-экономические показатели по сравниваемым  
 вариантам

|  |  |  |
| --- | --- | --- |
| Наименование показателей | Величина по вариантам | |
| Базовый | Проектный |
| 1 Годовое количество решаемых задач | | |
| 1.1 Среднее количество пациентов | 580 | 580 |
| 2 Норма времени решения задачи, мин | | |
| 2.1 Обработка медицинского исследования | 281,4 | 103,6 |
| 3 Уровень качества программного изделия, i σ | 4 | 5 |
| 4 Потребляемая мощность вычислительных средств, кВт | 0.850 | 0.850 |
| 5 Единовременные затраты, р. | 8902,43 | 8087,56 |
| 6 Годовые текущие издержки, р. | 23834,58 | 9179,73 |
| 7 Годовые приведенные затраты, р. | 25065,08 | 11043,46 |
| 8 Годовой результирующий эффект, р. | - | 14021.62 |
| 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 и единовременных затрат.

В результате годовой результирующий эффект по приведенным затратам составил 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-З (ред. от 17.07.2023 г. №300-З).

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.

Основными задачами 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, Руководители 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 охранен 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 безопасной эвакуации людей в учреждении здравоохранения «Могилевская областная клиническая больница»**

Безопасная эвакуация людей в случае чрезвычайных ситуаций – ключевой элемент обеспечения безопасности в медицинских учреждениях. Эффективность эвакуационных мероприятий напрямую влияет на способность больницы защитить жизни пациентов и персонала.

В учреждении здравоохранения «Могилевская областная клиническая больница»:

* имеются планы эвакуации, учитывающие особенности здания больницы.
* регулярно проводятся инструктажи с персоналом для повышения их готовности к действиям в экстренных ситуациях, обучение персонала.
* доступны эвакуационные выходы, заметны указатели, имеются схемы эвакуации и других средств, обеспечивающих быструю и организованную эвакуацию.
* 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
* двери эвакуационных выходов оборудованы замками, которые обеспечивают свободное открывание изнутри без использования ключа, что позволяет оперативно покинуть здание в случае необходимости
* размещение в палатах строго регламентируется, минимальное расстояние между койками 0,8 м, а проходы не менее 1,2 м
* здание соответствует требованиям пожарной безопасности, присутствует аварийное освещение, эвакуационные пути свободны для прохода.
* учитывается наличие пациентов с ограниченными возможностями и имеется специальное оборудование для их перемещения.

Для каждого этажа зданий и сооружений, где одновременно могут находиться более десяти человек, необходимо составить планы, обеспечивающие эвакуацию людей в случае возникновения пожара.

Двери эвакуационных выходов не допускается закрывать на замки. При необходимости допускается использование внутренних легкооткрывающихся (без ключа) замков и запоров.

Ковры, ковровые покрытия (дорожки) и иные рулонные покрытия на путях эвакуации жестко прикреплены к полу.

Кровати, оборудованные для транспортировки (эвакуации) пациентов с тяжелыми заболеваниями или ограниченными возможностями передвижения, должны быть в наличии. В случае их отсутствия, предусмотрено наличие носилок в количестве одни на каждые пять человек для эвакуации во время пожара. Эти носилки хранятся в местах, специально предназначенных для этого.

В помещениях с массовым пребыванием людей, запрещается установка глухих решеток на окнах, а также заделывать оконные проемы  
и загромождать подступы к ним. Если в помещении расположены постоянные рабочие места в количестве от 5 до 50, то допускается оснащение глухими решетками не более 50% окон.

Не допускается блокирование в открытом состоянии дверей, обеспечивающих противопожарную и дымозащитную функцию, включая двери лестничных клеток, за исключением случаев использования специальных устройств, которые автоматически закрывают двери в случае пожара. Также запрещается демонтаж таких дверей.

Для обеспечения безопасной эвакуации не допускается:

* уменьшать минимальную эвакуационную ширину и высоту, а также загромождать проходы, выходы, двери на путях эвакуации, эвакуационные выходы на кровлю (Минимальную эвакуационную ширину коридоров, проходов, галерей, эстакад и выходов в зданиях следует определять по таблице 2 ТКП 45-2.02-279-2013 «Здания и сооружения. Эвакуация людей при пожаре. Строительные нормы проектирования» либо расчетным путем по методикам, установленным в действующих ТНПА);
* изменять направление открывания дверей на препятствующее выходу из зданий и помещений;
* устраивать на путях эвакуации имитацию дверей, устанавливать витражи, зеркала, турникеты, выставочные стенды, торговые лотки, мебель, цветы, растения и другое имущество, препятствующее безопасной эвакуации;
* использовать лифты, подъемники, эскалаторы для эвакуации людей при пожаре. При возникновении пожара эскалаторы необходимо выключать и блокировать;
* размещать под маршами эвакуационных лестничных клеток горючие материалы и различные помещения, за исключением узлов управления центрального отопления и водомерных узлов.

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.

**6.3 Выводы и предложения**

По результатам анализа мероприятий по обеспечению безопасной эвакуации людей можно сделать вывод, что меры профилактики по обеспечению безопасной эвакуации людей в больнице соблюдены:

* таблички пожарной безопасности эвакуационных выходов расположены с соблюдением правил, схемы эвакуации созданы для каждого этажа в отдельности и для всего здания в целом;
* все эвакуационные пути снабжены указательными знаками в соответствии с ГОСТ 12.4.026-2015, которые находятся в исправном состоянии и хорошо видны.
* в больнице регулярно проводятся учебные эвакуации и инструктажи для персонала по действиям в случае пожара, что позволяет поддерживать высокий уровень готовности к экстренным ситуациям.
* инженерные системы больницы, включая вентиляционные и электроустановки, проходят регулярные проверки и техническое обслуживание, что снижает риск возникновения пожара и способствует безопасности эвакуации.

Для поддержания безопасной эвакуации предлагается:

* продолжение регулярного проведения плановых эвакуаций;
* проверка здания на соответствие различным нормам, в т.ч. пожарной безопасности
* подержание высокого уровня информированности и регулярное проведение инструкатажей не только среди персонала но и лечащихся
* введение дополнительного оборудования для эвакуации пациентов с ограниченными возможностями.

**7** **Энерго- и ресурсосбережение**

**7.1** **Энергоэффективные технологии в здравоохранении**

Целью данного дипломного проекта является разработка автоматизированной системы обработки изображений (АСОИ) "Медицинская информационная система обработки изображений", направленной на улучшение качества медицинского обслуживания и эффективное использование ресурсов. В условиях современной медицины одним из ключевых факторов является энергосбережение, которое позволяет не только снизить затраты на эксплуатацию медицинских учреждений, но и уменьшить экологический след. Поэтому разработка АСОИ для здравоохранения должна учитывать аспекты энергосбережения, интегрируя энергоэффективные технологии и процессы.

Современные медицинские учреждения оснащены разнообразным оборудованием, требующим больших объемов энергии для поддержания своей работы. Это включает в себя освещение, системы отопления и кондиционирования, медицинские аппараты и оборудование, а также информационные системы.

Внедрение энергосберегающих технологий в медицинских учреждениях включает в себя несколько ключевых направлений:

* Светодиодное освещение: переход на светодиодные лампы позволяет значительно сократить потребление электроэнергии на освещение, что особенно важно в круглосуточно работающих учреждениях.
* Системы управления энергопотреблением: современные системы автоматизации и управления позволяют оптимизировать энергопотребление, контролируя работу оборудования и освещения в зависимости от реальной потребности.
* Энергоэффективное оборудование: использование современного медицинского оборудования с низким энергопотреблением и высоким КПД (коэффициентом полезного действия) помогает уменьшить нагрузку на энергосистему учреждения.

Разработка и внедрение автоматизированной системы обработки изображений (АСОИ) может существенно способствовать энергосбережению в здравоохранении. Основные аспекты включают:

* Оптимизация процессов обработки данных: АСОИ позволяет автоматизировать и ускорить обработку медицинских изображений, снижая нагрузку на серверное оборудование и, соответственно, потребление энергии.
* Сокращение физического хранения данных: переход на электронные архивы и облачные технологии позволяет уменьшить количество физических серверов и систем хранения, что приводит к снижению энергозатрат на их содержание и охлаждение.
* Интеграция с энергоэффективными системами: АСОИ может быть интегрирована с системами управления зданием, что позволит более эффективно управлять энергопотреблением в зависимости от текущей нагрузки и потребностей учреждения.

Таким образом, энергосбережение в здравоохранении является важным аспектом, который необходимо учитывать при разработке и внедрении медицинских информационных систем. Автоматизированная система обработки изображений (АСОИ) "Медицинская информационная система обработки изображений" не только улучшит качество медицинских услуг, но и будет способствовать значительному сокращению энергозатрат. Интеграция энергоэффективных технологий и оптимизация процессов позволит медицинским учреждениям достичь устойчивого развития и эффективного использования ресурсов.

**Заключение**

Проект многопользовательской информационной системы для хранения и обработки медицинских изображений и заключений был успешно реализован, включая разработку клиентской части на React, что обеспечило динамичный и отзывчивый пользовательский интерфейс. Использование React позволило создать высокопроизводительную клиентскую часть, которая упрощает взаимодействие пользователей с системой, позволяя врачам и медицинским специалистам эффективно управлять медицинскими данными и мгновенно получать результаты анализа.

Разработка и сборка нейронной сети, реализованная с помощью TensorFlow и Keras, позволила автоматизировать процесс диагностики за счет предоставления предварительных медицинских заключений на основе анализа рентгеновских снимков. Это значительно ускоряет процесс принятия решений и повышает точность диагностических заключений, при этом уменьшая вероятность человеческой ошибки.

Интеграция клиентской части с серверной обеспечивает бесперебойную работу системы и плавное взаимодействие между различными компонентами. Поддержка множественных источников медицинских изображений делает систему универсальным инструментом для различных 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.

Таким образом, созданная система не только повышает качество медицинского обслуживания и скорость диагностики, но и сокращает затраты на хранение и обработку данных. Экономическая целесообразность проекта подтверждена, и система готова к внедрению в реальную медицинскую практику, предоставляя значительные преимущества для улучшения процессов 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. React. The library for web and native user interfaces   
   / [Электронный ресурс]. Режим доступа: <https://react.dev> – Дата доступа 10.06.2024
5. **Andersson, E.** Software Engineering for Internet Applications. / Greenspun P., Grumet A. – MIT Press, 2006. – 399 p.
6. **Fowler, M.** Patterns of Enterprise Application Architecture. – Addison-Wesley Professional, 2002. – 560 p.
7. Tensorflow / [Электронный ресурс]. – Режим доступа: https://tensorflow.org/ – Дата доступа 10.06.2024
8. Keras / [Электронный ресурс]. – Режим доступа: <https://keras.io/> – Дата доступа 10.06.2024
9. **Grinberg M.**, Flask Web Development. – O'Reilly Media, Inc., 2018 – 316 p.
10. Orthanc / [Электронный ресурс]. – Режим доступа: https://www.orthanc-server.com/ – Дата доступа 10.06.2024
11. **Goodfellow, I.** Deep Learning. / Bengio, Y., & Courville, A. – IFAC-PapersOnLine, 2015. – 800 p.
12. **Aggarwal C.** Neural Networks and Deep Learning: A Textbook. – MIT Press, 2018. – 497 p.
13. 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.
14. **Кривой, В. И.** Охрана труда: комментарий новейшего законодательства / В. И.Кривой. - Гомель: Ковалева Е. А., 2009. - 367 с. - (Библиотека практического работника).
15. **Кузнецов, И. Н.** Охрана труда: учебное пособие / И. Н. Кузнецов. - Минск: Четыре четверти, 2010. - 175 с.

**Ведомость документов к дипломному проекту**