

УДК 004.032.26

АРХИТЕКТУРА И ИНСТРУМЕНТЫ МЕДИЦИНСКОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ИНТЕГРАЦИИ С МОДЕЛЯМИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: СРАВНИТЕЛЬНЫЙ АНАЛИЗ СОВРЕМЕННЫХ РЕШЕНИЙ

В. В. БЕРЕЗОВСКИЙ¹, Н. В. ВЫГОВСКАЯ², М. В. ПАШКЕВИЧ²

¹Северный (Арктический) федеральный университет имени М. В. Ломоносова
Архангельск, Россия

²Белорусско-Российский университет
Могилев, Беларусь

Современные медицинские учреждения сталкиваются с проблемой высокой нагрузки на специалистов при анализе медицинских изображений, а также с ограниченной доступностью экспертной диагностики, что может приводить к задержкам в постановке диагноза и увеличению вероятности ошибок.

Разрабатываемое веб-приложение решает данную проблему за счет автоматизации анализа медицинских изображений с использованием модели искусственного интеллекта, предоставляя пользователю диагностическое заключение в цифровом формате. В условиях практического применения подобного решения особое значение приобретает выбор архитектуры программной системы, поскольку от него зависят надежность обработки данных, безопасность медицинской информации, воспроизводимость результатов инференса и возможность масштабирования приложения, что обуславливает необходимость обоснованного архитектурного анализа.

Основой разработки являются следующие аспекты.

1. Выбор наиболее оптимальной архитектуры.
2. Выбор фреймворка для реализации данной архитектуры.

В рамках разработки были рассмотрены монолитная и микросервисная архитектуры, представляющие различные подходы к построению системы.

Монолитная архитектура предполагает объединение интерфейса, серверной логики и модулей ИИ в едином программном комплексе с общим хранилищем данных. Такой подход минимизирует требования к инфраструктуре и исключает риски, связанные с сетевым взаимодействием между компонентами.

Для медицинских систем монолитная структура обладает рядом преимуществ: она упрощает контроль целостности, централизует управление версиями кода и моделей, а также облегчает валидацию диагностических результатов. В отличие от микросервисной архитектуры, требующей сложных инструментов оркестрации и значительных затрат на сопровождение, монолит обеспечивает предсказуемость системы и высокую воспроизводимость инференса.

Учитывая цели разработки – надежность и прозрачность формирования диагностических выводов в условиях ограниченных ресурсов, – выбор монолитной архитектуры является наиболее обоснованным. Это позволяет снизить техническую сложность проекта, упростить аудит логики принятия решений и гарантировать стабильную работу приложения в клинической практике.

Важным этапом разработки медицинского веб-приложения с использованием искусственного интеллекта является выбор веб-фреймворка для реализации серверной части системы. В рамках анализа были рассмотрены Flask, Django и FastAPI.

Flask является лёгким и гибким фреймворком, удобным для прототипирования и небольших сервисов, однако отсутствие встроенных механизмов безопасности, аутентификации и валидации данных усложняет его использование в сложных медицинских системах.

Django представляет собой полнофункциональное решение с широким набором встроенных инструментов, что делает его подходящим для крупных медицинских информационных систем, но избыточная функциональность и сложная архитектура ограничивают его применение в высоконагруженных сервисах инференса.

FastAPI ориентирован на разработку высокопроизводительных API, поддерживает асинхронную обработку запросов, встроенную валидацию данных и автоматическую генерацию документации, что делает его наиболее подходящим для микросервисной архитектуры.

На основании проведенного анализа было реализовано медицинское приложение для диагностики сколиоза и спондилолистеза, сочетающее монолитную архитектуру, фреймворк FastAPI и нейросетевую модель ResNet-50. После обработки рентгеновского изображения позвоночника модель выполняет предсказание диагноза, возвращая класс заболевания и вероятность (уверенность) предсказания. Результаты отправляются пользователю в виде JSON-ответа [1].

Для создания удобного пользовательского интерфейса был использован фреймворк React JS, имеющий богатую экосистему библиотек и инструментов для управления маршрутизацией, состоянием и стилизацией компонентов. Данный технологический стек обеспечил необходимую надежность и высокую скорость обработки данных. Результаты тестирования подтвердили эффективность системы и целесообразность её внедрения в клиническую практику для снижения нагрузки на профильных специалистов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Прототип системы искусственного интеллекта для диагностики заболеваний позвоночника с интеграцией в веб-приложение и телеграм-бот / В. В. Березовский, Н. В. Выговская, Р. В. Милевский, М. В. Пашкевич // Врач и информационные технологии. – 2025. – № 4. – С. 56–71.