

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**РАЗРАБОТКА .NET ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ
ОБРАБОТКИ АУДИОФАЙЛОВ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Никитиной Вероники Дмитриевны

Научный руководитель:

Старший преподаватель кафедры ИиП _____ М.С. Портенко

подпись, дата

Зав. кафедрой:

к.ф.-м.н., доцент _____ М.В. Огнева

подпись, дата

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. Актуальность темы обусловлена растущими потребностями музыкальной индустрии в автоматизации рутинных процессов. Звукорежиссеры и музыканты тратят значительное время на обработку дорожек отдельных инструментов для соответствия звучания современному продакшену. Существующие решения либо требуют глубоких знаний в области саунд-дизайна, либо не предоставляют целевой автоматизации под конкретный инструмент и жанр. Разработка специализированного веб-приложения, использующего современные backend технологии и сервисы машинного обучения, позволяет значительно повысить эффективность труда специалистов и сделать профессиональную обработку звука более доступной. Проведенный анализ литературных источников и рынка программных продуктов показал, что задача автоматической обработки аудио является активно развивающейся областью. Однако существующие коммерческие решения, такие как Audo Studio и LALAL.AI, сосредоточены на решении узкоспециализированных задач. Прямых аналогов, представляющих собой приложение для адаптации звучания заранее записанных инструментов с использованием ML-сервиса (machine learning) выявлено не было. Данный факт подтверждает новизну предлагаемого подхода и определяет нишу для разрабатываемого проекта.

Цель бакалаврской работы – разработать веб-приложение, с интеграцией сервиса с заранее обученными нейросетевыми моделями для автоматической обработки аудиофайлов, которое предоставит звукорежиссёрам инструмент для обработки дорожек различных инструментов.

Поставленная цель определила **следующие задачи:**

1. Провести анализ предметной области, существующих аналогов и сформулировать требования к функционалу веб-приложения.
2. Обосновать выбор технологического стека и спроектировать архитектуру приложения.

3. Реализовать функционал backend-приложения, обеспечивающий загрузку аудиофайлов, управление задачами обработки и взаимодействие с микросервисом worker, с ML-сервисом и нейросетевыми моделями.
4. Обеспечить интеграцию backend-приложения с внешним ML-сервисом для передачи задач на обработку и получения результатов.

Методологические основы разработки .NET веб-приложения для автоматизации обработки аудиофайлов представлены в работах Уваров А. М. [1], Охотников Д. В. [2], Филиппов П. С. [3], Тарасов А. В. [4], Скотт Д. [5], Кейн Ш. [6], Арора Г. [7], Смит Дж. П. [8].

Практическая значимость бакалаврской работы заключается в создании полнофункционального веб-приложения, которое может быть использовано звукорежиссерами, музыкальными продюсерами и независимыми музыкантами для профессиональной обработки аудио. Система устраняет необходимость в дорогостоящем специализированном программном обеспечении, предоставляя веб-интерфейс для работы с нейросетевыми моделями обработки звука для различных музыкальных инструментов.

Структура и объём работы. Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и двенадцати приложений. Общий объем работы – 112 страниц, из них 62 страниц – основное содержание, включая 16 рисунков, цифровой носитель в качестве приложения, список использованных источников информации – 22 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ предметной области и постановка задачи» посвящен исследованию теоретических основ обработки аудиосигналов с применением методов искусственного интеллекта, анализу существующих решений в области автоматизации аудиообработки, формированию требований к разрабатываемому веб-приложению и обоснованию выбора

технологического стека. В разделе рассмотрены современные подходы к цифровой обработке звука, включая использование нейросетевых моделей архитектуры encoder-decoder с оконным преобразованием для анализа и восстановления аудиосигнала во временно-частотной области. Проведённый сравнительный анализ коммерческих и исследовательских проектов, таких как Audo Studio, LALAL.AI, LANDR, а также отечественных разработок в области генерации музыкальных композиций и музыкальных стриминговых сервисов, выявил отсутствие прямых аналогов, представляющих собой универсальную платформу для адаптации звучания заранее записанных инструментов с использованием ML-сервиса. Большинство рассмотренных решений ориентированы на решение узкоспециализированных задач и обладают ограничениями в области масштабируемости из-за использования монолитной архитектуры или отсутствия механизмов асинхронной обработки задач. На основе методологии MoSCoW были сформулированы функциональные требования, включающие загрузку аудиофайлов через веб-интерфейс, выбор музыкального инструмента для обработки, отслеживание статуса выполнения задачи в реальном времени и получение результата в виде обработанного файла, а также нефункциональные требования к производительности, надёжности и масштабируемости системы. Обоснование выбора технологического стека определило использование .NET 9 и ASP.NET Core Web API для backend-модуля, PostgreSQL с Entity Framework Core для хранения метаданных, MinIO для объектного хранилища аудиофайлов, Apache Kafka в качестве брокера сообщений для организации асинхронной обработки задач, SignalR для real-time уведомлений клиента, Vue 3 с TypeScript для frontend-приложения и Docker для контейнеризации всех компонентов системы. Архитектурное решение построено на принципах слабосвязанных микросервисов с использованием событийно-ориентированной коммуникации через четыре специализированных Kafka-топика, что обеспечивает декомпозицию процесса обработки аудио на независимые этапы, горизонтальную масштабируемость и отказоустойчивость системы.

Проведённый анализ предметной области и существующих аналогов подтвердил новизну предлагаемого подхода и определил нишу для разрабатываемого проекта. Сформулированные требования и выбранная микросервисная архитектура с событийно-ориентированной коммуникацией на базе Apache Kafka создают теоретический фундамент для последующей практической реализации системы обработки аудиоданных.

Второй раздел «Процесс разработки» посвящен реализации всех компонентов системы, включая контейнеризацию и оркестрацию сервисов через Docker Compose, разработку backend API на платформе .NET 9 с использованием паттерна CQRS и библиотеки MediatR для разделения ответственности между операциями чтения и записи, реализацию worker-сервиса для оркестрации задач обработки через Apache Kafka, интеграцию с ML-сервисом на Python с фреймворком FastAPI и нейросетевой моделью UNet, а также создание frontend-приложения на Vue 3 с централизованным управлением состоянием через Pinia. В рамках контейнеризации был разработан файл docker-compose.yml, определяющий десять сервисов: backend-API, worker-сервис, ML-сервис, frontend-приложение, база данных PostgreSQL, объектное хранилище MinIO, три брокера Kafka с коэффициентом репликации 3 для отказоустойчивости, Zookeeper для координации кластера, сервис инициализации топиков Kafka и веб-интерфейс управления Kafka-UI. Многоэтапная сборка Docker-образов для backend, worker и ML-сервисов позволила минимизировать размер финальных образов и обеспечить эффективное кэширование слоёв. Backend API реализован по принципу тонкого контроллера, где вся бизнес-логика вынесена в обработчики MediatR, а контроллеры выполняют исключительно функции маршрутизации запросов и форматирования ответов. Разработаны контроллеры FilesController для загрузки и скачивания файлов, TracksController для управления метаданными аудиодорожек, ProcessController для инициации обработки аудио, JobsController для отслеживания статуса задач, CommonController для предоставления справочных данных и HealthController для проверки

работоспособности сервиса. Инфраструктурный слой включает сервис MinioService для взаимодействия с объектным хранилищем через официальный .NET SDK с методами загрузки и извлечения файлов потоком для эффективной работы с большими аудиофайлами, контекст данных AppDbContext для доступа к таблицам Tracks и Jobs базы данных PostgreSQL через Entity Framework Core с поддержкой миграций для эволюции схемы базы данных, а также репозитории TracksRepository и JobsRepository, инкапсулирующие CRUD-операции и изолирующие бизнес-логику от деталей реализации доступа к данным. Worker-сервис реализован как Background Service, потребляющий сообщения из топика job.created, подготавливающий данные для ML-обработки и публикующий события в топик job.prepared, при этом используется стратегия гарантированной доставки сообщений at-least-once с ручным коммитом смещений для обеспечения надёжности обработки задач. ML-сервис, разработанный Мыльниковой А., реализован на Python 3.10 с FastAPI для health check endpoints и асинхронным потреблением сообщений из Kafka, модельный менеджер обеспечивает динамическую загрузку и кэширование процессоров для различных музыкальных инструментов, а двойное уведомление backend через Kafka и HTTP PUT-запрос гарантирует доставку результатов обработки даже при временных сбоях отдельных каналов связи. Frontend-приложение эволюционировало от минимально работающего прототипа с базовой функциональностью загрузки, обработки и прослушивания результатов до полноценного интерфейса с компонентной архитектурой, системой модальных окон, типизированной обработкой ошибок API, валидацией файлов на стороне клиента и расширенной функциональностью экспорта проектов в ZIP-архиве через библиотеки JSZip и FileSaver. В результате проведённой разработки была создана полнофункциональная микросервисная система обработки аудиоданных, включающая контейнеризованную инфраструктуру из десяти сервисов, backend API с реализацией паттерна CQRS и MediatR для слабой связанности компонентов, worker-сервис для надёжной оркестрации задач через Apache

Kafka с гарантией доставки сообщений, ML-сервис для нейросетевого инференса и современный frontend на Vue 3 с централизованным управлением состоянием. Архитектура системы обеспечивает масштабируемость, отказоустойчивость и возможность независимого развёртывания отдельных компонентов.

Третий раздел «Перспективы развития и расширение функциональности системы» посвящен определению основных направлений бизнес-развития и технического совершенствования разработанной платформы для автоматизации обработки аудиофайлов. Ключевым направлением является расширение перечня поддерживаемых музыкальных инструментов за счёт обучения новых нейросетевых моделей для струнных, духовых и этнических инструментов, что реализуется благодаря модульной архитектуре ML-сервиса с динамической загрузкой процессоров через словарь INSTRUMENTS без существенной модификации основных компонентов системы. Перспективным направлением является поддержка жанровой параметризации обработки, позволяющая адаптировать алгоритмы разделения микса под специфические требования различных музыкальных направлений с учётом тембральных характеристик и особенностей аранжировок. Внедрение системы аутентификации и авторизации пользователей на базе ASP.NET Core Identity с JWT-токенами откроет возможности для долгосрочного хранения проектов на стороне сервера в базе данных PostgreSQL и обеспечения доступа к данным с различных устройств, устраняя текущее ограничение хранения информации исключительно в клиентском хранилище Pinia. Развитие многопользовательского режима с механизмами совместного доступа к проектам позволит командам звукорежиссеров и музыкальных продюсеров работать над общими проектами синхронно или асинхронно, при этом сущность ProjectEntity с полем user_id уже создаёт основу для разграничения прав доступа, а интеграция SignalR может обеспечить совместное редактирование параметров и мгновенное уведомление участников команды

об изменениях. Сбор обратной связи от профессиональных звукорежиссеров и музыкальных продюсеров через организацию бета-тестирования, создание пользовательских форумов и каналов обратной связи станет основой для формирования сообщества пользователей и генерации новых идей для развития проекта, поскольку практический опыт использования системы в реальных производственных условиях позволит выявить узкие места текущего интерфейса и сформировать приоритеты для будущих итераций разработки. Разработанная система представляет собой не просто законченный продукт с текущим набором функций, а гибкую платформу для дальнейшего развития и адаптации под изменяющиеся потребности рынка музыкальных технологий; модульная архитектура, событийно-ориентированная коммуникация через Apache Kafka и контейнеризация сервисов создают прочный технический фундамент для реализации описанных направлений развития, а сбор профессиональной обратной связи обеспечит востребованность будущих улучшений и превращение проекта в полноценный коммерческий продукт для профессиональной аудиоинженерии.

ЗАКЛЮЧЕНИЕ

В ходе данной работы были подробно изучены технологии для создания веб-приложения по автоматизации обработки аудиофайлов. Целью работы было разработать веб-приложение для автоматизации обработки аудиофайлов, которое предоставит звукорежиссерам инструмент для усовершенствования звучания дорожек различных инструментов через интеграцию с нейросетевыми моделями. Для достижения данной цели были рассмотрены различные возможности современных backend-технологий и архитектурных подходов к построению распределенных систем. Кроме того, были изучены существующие коммерческие и исследовательские решения в области обработки аудиоданных, их архитектурные особенности и технологические стеки. Был проведен анализ технологий, сравнение микросервисной и монолитной архитектур, а также оценка эффективности

различных подходов к организации асинхронной обработки задач. В результате проведенной работы были выполнены поставленные задачи, а именно: изучение, сравнение и выбор подходящих технологий, проектирование архитектуры веб-приложения на основе микросервисного подхода, реализация функционала backend-микросервиса на платформе .NET 9 с использованием ASP.NET Core Web API, обеспечение загрузки аудиофайлов, управления задачами обработки, интеграции с сервисами через брокер сообщений Apache Kafka и интеграцией с моделями нейросетей, которые были получены в ходе дипломной работы Мыльниковой Анастасии. Практическая значимость разработанной системы заключается в предоставлении доступного инструмента, который может быть использован звукорежиссерами, музыкальными продюсерами и независимыми музыкантами для профессиональной обработки аудио. Система устраняет необходимость в дорогостоящем специализированном программном обеспечении, предоставляя веб-интерфейс для работы с нейросетевыми моделями.

Основные источники информации:

1. Разработка приложения с пользовательским интерфейсом для музыкального стримингового сервиса / А. М. Уваров, И. Д. Борисов, А. М. Гоголев, Д. А. Шубенкин // Технологии 2022: основные проблемы и направления развития : Сборник статей II Международной научно-практической конференции, Пенза, 30 июня 2022 года. – Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2022. – С. 45-48.
2. Охотников, Д. В. Разработка музыкального сервиса / Д. В. Охотников // Вузовская наука в современных условиях : Сборник материалов 56-й научно технической конференции. В 2-х частях, Ульяновск, 24–29 января 2022 года. Том Часть 2. – Ульяновск: Ульяновский государственный технический университет, 2022. – С. 64-67.
3. Филиппов, П. С. Информационная система для обнаружения и классификации дефектов металлопрокатной продукции на базе

- искусственного интеллекта [Электронный ресурс] / Филиппов П. С., Греченева А. В., Никаноров М. С. // Известия ТулГУ. Технические науки. 2024. №3. — URL: <https://cyberleninka.ru/article/n/informatsionnaya-sistema-dlya-obnaruzheniya-i-klassifikatsii-defektov-metalloprokatnoy-produktsii-na-baze-iskusstvennogo-intellekta> (дата обращения: 18.11.2025). — Загл. с экрана. — Яз. рус.
4. Тарасов, А. В. Разработка веб-сервиса для облегчения доступа к генеративным нейронным сетям [Электронный ресурс] / А. В. Тарасов // Международный журнал гуманитарных и естественных наук. 2024. №5-4 (92). URL: <https://cyberleninka.ru/article/n/razrabotka-veb-servisa-dlya-oblegcheniya-dostupa-k-generativnym-neuronnyim-setyam> (дата обращения: 18.11.2025). — Загл. с экрана. — Яз. рус.
 5. Скотт, Д. Kafka в действии : практическое руководство / Д. Скотт, В. Гамов, Д. Клейн ; пер. с англ. А. Н. Киселева. - Москва : ДМК Пресс, 2022. - 310 с. - ISBN 978-5-93700-118-4.
 6. Кейн, Ш. П. Docker. Вводный курс. - 3-е изд.: Пер. с англ. / Ш. П. Кейн, К. Матриас. Астана: АЛИСТ, 2024. - 352 с.
 7. Арораа, Г. Паттерны проектирования для C# и платформы .NET Core. / Ч. Джеффри — СПб.: Питер, 2021. — 352 с.
 8. Смит Дж. П. Entity Framework Core в действии / пер. с англ. Д. А. Беликова. — М.: ДМК Пресс, 2022. — 690 с.