Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИУК «Информатика и управление»**

КАФЕДРА ***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему:

Разработка сервиса обработки мультимедиа

по дисциплине *Компьютерные сети*

Студент гр. ИУК4-72Б _____ (Карельский М.К.)
(подпись) (Ф.И.О.)

Руководитель _____ (Красавин Е.В.)
(подпись) (Ф.И.О.)

Оценка руководителя _____ баллов _____
30-50 (дата)

Оценка защиты _____ баллов _____
30-50 (дата)

Оценка проекта _____ баллов _____
(оценка по пятибалльной шкале)

Комиссия: _____ (Красавин Е.В.)
(подпись) (Ф.И.О.)

_____ (Гагарин Ю.Е.)
(подпись) (Ф.И.О.)

_____ (Белов Ю.С.)
(подпись) (Ф.И.О.)

Калуга, 2023

Калужский филиал
федерального государственного бюджетного образовательного учреждения высшего образования

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой _____

_____ (_____)

«____» _____ 2023г.

ЗАДАНИЕ
на выполнение курсового проекта

по дисциплине **Компьютерные сети**

Студент Карельский М.К., ИУК4-72Б
(фамилия, инициалы, индекс группы)

Руководитель Красавин Е.В.
(фамилия, инициалы)

График выполнения работы: 25% к 6 нед., 50% к 9 нед., 75% к 11 нед., 100% к 14 нед.

1. Тема курсового проекта
Разработка сервиса обработки мультимедиа

2. Техническое задание
Спроектировать сервис обработки мультимедиа

3. Оформление курсового проекта
3.1. Расчетно-пояснительная записка на _ листе формата А4.
3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)
1. *Архитектура приложения*
2. *Структура БД*
3. *Демонстрационный чертеж*

Дата выдачи задания «*09*» *сентября 2023*г.

Руководитель курсовой работы _____ / Красавин Е.В.
(подпись) (Ф.И.О.)

Задание получил _____ / Карельский М.К. / «*09*» *сентября* 20*23* г.
(подпись) (Ф.И.О.)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью производственной практики является разработка серверной части сервиса обработки изображений, видео и аудио с помощью реализации брокера сообщений.

Для достижения поставленной цели решаются следующие **задачи**:

1. Реализовать алгоритмы обработки;
2. Создать базу данных;
3. Реализовать API сервиса;
4. Интегрировать брокер сообщений;
5. Провести контейнеризацию.

1. Основные требования к разрабатываемой информационной системе

1.1. Описание предметной области

Множество пользователей ежедневно сталкивается с необходимостью быстрого изменения своих файлов, например, изменение формата изображения или извлечение фрагмента из видео. Для решения подобных задач применяется различное настольное программное обеспечение. Однако описанные задачи могут возникать перед пользователем не так часто, чтобы у того появлялась необходимость в установке на свой ПК дополнительных программ, особенно если сама задача не является сложной, комплексной. В таком случае гораздо удобнее воспользоваться специальным онлайн-сервисом, предоставляющим возможность загрузки личных файлов мультимедиа, их последующей требуемой обработки и скачивания готового результата.

Для реализации данного механизма предполагается использование двух программных модулей и брокера сообщений. Задача первого модуля – принимать запросы на обработку файлов, предоставлять статус выполнения задач, а также получать, хранить и возвращать сами файлы. Второй модуль представляет из себя несколько обработчиков, ожидающих поступления новых задач и выполняющих необходимые преобразования и параллельным изменением статуса самой задачи. Для обеспечения асинхронного взаимодействия описанных выше модулей необходимо использование брокера сообщений – специального программного узла, который принимает от первого модуля задачи на обработку, а затем хранит и возвращает их второму модулю-обработчику.

1.2. Требования к функциям

Представляя из себя серверную часть сервиса, данная система должна предоставлять различный API обработки мультимедиа. Среди этого можно выделить следующие опции:

1. Обработка изображений:
    - Двукратное увеличение разрешения. Данный вид обработки можно построить на основе алгоритма, с помощью которого анализируется окружение каждого пикселя, а затем каждый из них разбивается на 4 других, окрашиваясь в соответствующий полученной информации цвет. Задачу определения степени влияния окружающих пикселей на итоговый цвет (по сути, их веса) можно решить с помощью применения нейронной сети;
    - Удаление шума. Данная задача может быть решена так же с помощью анализа окружения пикселя и установления степени влияния этого окружения посредством нейронной сети;
    - Изменение формата;
    - Кадрирование;
    - Поворот по часовой или против часовой стрелки;
    - Отражение по горизонтали или вертикали;
    - Сжатие;
    - Добавление водяных знаков.
2. Обработка видео:
    - Изменение формата;
    - Изменение продолжительности;
    - Объединение;
    - Изменение цвета на черно-белое;
    - Кадрирование;

- Добавление простых переходов;
- Поворот по часовой или против часовой стрелки;
- Отражение по горизонтали или вертикали;
- Ускорение и замедление;
- Обратное воспроизведение;
- Изменение громкости;
- Извлечение аудио.

3. Обработка аудио:
    - Изменение формата;
    - Перевод в моно;
    - Определение темпа (BPM);
    - Изменение продолжительности;
    - Объединение;
    - Ускорение и замедление;
    - Обратное воспроизведение;
    - Изменение громкости;
    - Панорамирование.

2. Анализ аналогов и прототипов

2.1. Рассмотрение аналогов

ILoveIMG.com

ILoveIMG.com предоставляет различные инструменты для редактирования изображений. В их число входит сжатие, кадрирование, поворот, добавление водяных знаков и преобразование в различные форматы.

Основной функционал доступен бесплатно. Имеется платная подписка стоимостью 234 руб./месяц, предоставляющая более широкие возможности и неограниченную обработку. Также предлагаются индивидуальные планы для бизнеса.

Convertio.co

Convertio.co – онлайн-сервис, предоставляющий возможность преобразовывать изображения, видео и аудио во множество различных форматов.

Бесплатная версия имеет ограничение максимального размера загружаемого файла – 100 МБ. Сервис предоставляет 3 уровня подписки: за 6$, 9$ и 16$ в месяц. Более высокий уровень предоставляет возможность загрузки файлов большего размера, большего количества одновременных конвертаций, а также более высокий приоритет в очереди обработки.

123apps.com

Предоставляет широкий спектр веб-приложений для обработки видео и аудио. Среди них: объединение, сокращение, кадрирование, поворот, отражение, изменение громкости и скорости видео, объединение, сокращение, изменение громкости и скорости аудио. Также доступны конвертеры изображений, видео и аудио.

Бесплатная версия накладывает ограничение на максимальный размер файла в 500 МБ и количество доступных обработок в день – 5. Предлагается подписка за 300 руб./месяц, снимающая ограничения на количество обработок в день и увеличивающая максимальный размер загружаемого файла до 4 ГБ.

2.2. Обоснование выбора средств разработки

2.2.1. Язык программирования

PHP

PHP используют около 80% всех сайтов, а самому языку уже более 25 лет. Также обладает широким сообществом, благодаря чему язык прост для обучения и получает постоянные обновления. Его достаточно легко установить и настроить.

Java

Java – один из самых популярных языков программирования. Используется уже более 20 лет и имеет широкую поддержку сообществом. Также обладает хорошей универсальностью, которая обеспечивается с помощью виртуальной машиной Java (JVM).

C#

C# является широко известным и используемым объектно-ориентированным языком программирования. Для реализации серверной части используется достаточно популярный, стабильный и надежный фреймворк ASP.NET. Поддержку ASP.NET 6.0 обеспечивает среда разработки Visual Studio 2022.

Python

Python обладает большой популярностью среди различных сфер разработки, включая разработку серверов. Наиболее популярным фреймворком для этого является Flask, обладающий большим сообществом и позволяющий написать серверную часть достаточно просто. Помимо этого, Python может применяться для написания самих обработчиков задач, так как имеет для этого множество полезных и удобных библиотек, например, Pillow.

2.2.2. Брокер сообщений

RabbitMQ

RabbitMQ – это брокер распределенных сообщений, который собирает потоковые данные из нескольких источников и маршрутизирует их в разные пункты назначения для обработки. RabbitMQ обеспечивает легкость разработки, так как имеет библиотеки для множества языков, простое администрирование тонкую настройку.

Kafka

Представляет из себя распределённый программный брокер сообщений с открытым исходным кодом. Ядром функциональности является запись данных, хранение их в течение заданного времени и выдача этих данных по запросу. Предоставляет большую пропускную способность, возможность читать множество сообщений за раз, а также позволяет перечитывать ранее прочитанные сообщения.

SQS

SQS – сервис от Amazon, принимающий очереди сообщений для хранения. Весьма популярен за рубежом, предоставляет высокую безопасность и интеграцию со всеми сервисами компании. Однако в связи с полной

зависимостью от Amazon возникают проблемы при появлении необходимости в переходе на другого поставщика.

2.2.3. База данных

OracleDB

OracleDB – объектно-реляционная система управления базами данных компании Oracle. В связи с поддержкой крупной компании имеет высокую надежность. Обеспечивает хорошую масштабируемость, безопасность и скорость. Однако также имеет высокую стоимость.

MySQL

MySQL – реляционная система управления базами данных с открытым исходным кодом. Имеет достаточно широкую популярность и сообщество. Предлагает простую установку и использование. Хорошо работает с данными на базовом уровне, однако в процессе масштабирования может потребоваться дополнительная поддержка, также имеющая высокую стоимость.

PostgreSQL

PostgreSQL – реляционная база данных с открытым кодом, является одной из наиболее известных среди всех существующих реляционных баз данных. Предоставляет хорошую масштабируемость, дополнительную защиту посредством использования ролей и прав. Полностью бесплатна, поддерживается сообществом, хорошо справляется с большими нагрузками.

MongoDB

MongoDB – документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы

и схему базы данных. Написана на языке C++, что позволяет легко портировать ее на различные платформы.

2.2.4. Веб-сервер

Apache

Apache – это программное обеспечение с открытым исходным кодом, разработанное и поддерживаемое открытым сообществом разработчиков и работающее в самых разных операционных системах. Более прост в использовании, поддерживается всеми операционными системами, предоставляет возможность добавления модулей.

Nginx

Nginx – это программное обеспечение с открытым исходным кодом, которое позволяет создавать веб-сервер. Обеспечивает хорошую производительность для статического и динамического контента, высокую безопасность, имеет хорошую поддержку.

2.2.5. Контейнеризатор

Docker

Docker – это платформа для контейнеризации приложений с открытым исходным кодом. Она позволяет упаковывать приложения с их окружениями и зависимостями в контейнеры, а затем предоставляет возможность с помощью встроенных команд управлять ими. Предоставляет изолированное окружение для каждого контейнера, каждый из которых можно легко переносить между различными средами. Обеспечивает эффективное использование ресурсов хостовой системы и возможность быстрого развертывания приложений.

Podman

Podman – это контейнеризатор с открытым исходным кодом, представляет собой утилиту командной строки. Позволяет запускать и управлять контейнерами без наличия демона. Обеспечивает изоляцию каждого контейнера с более высокой безопасностью.

Containerd

Containerd – бывшая часть Docker, реализует исполняемую среду для запуска контейнеров. Предоставляет минимальный набор функций для управления образами, а также для запуска и остановки контейнеров. Предлагает высокую производительность, стабильность и широкую поддержку сообщества.

2.2.6. Итоговый выбор

В качестве языка программирования был выбран Python, используемый в написании как серверной части посредством фреймворка Flask, так и самих обработчиков задач. Для распределения сообщений был выбран RabbitMQ, предоставляющий легкую интеграцию в проект. Хранение данных производится в PostgreSQL, обладающий полным необходимым функционалом. Веб-сервером выступает Nginx, предоставляющий такие возможности, как, например, настройку обратного прокси. Для контейнеризации используется наиболее распространенный Docker.

2.3. Структура системы

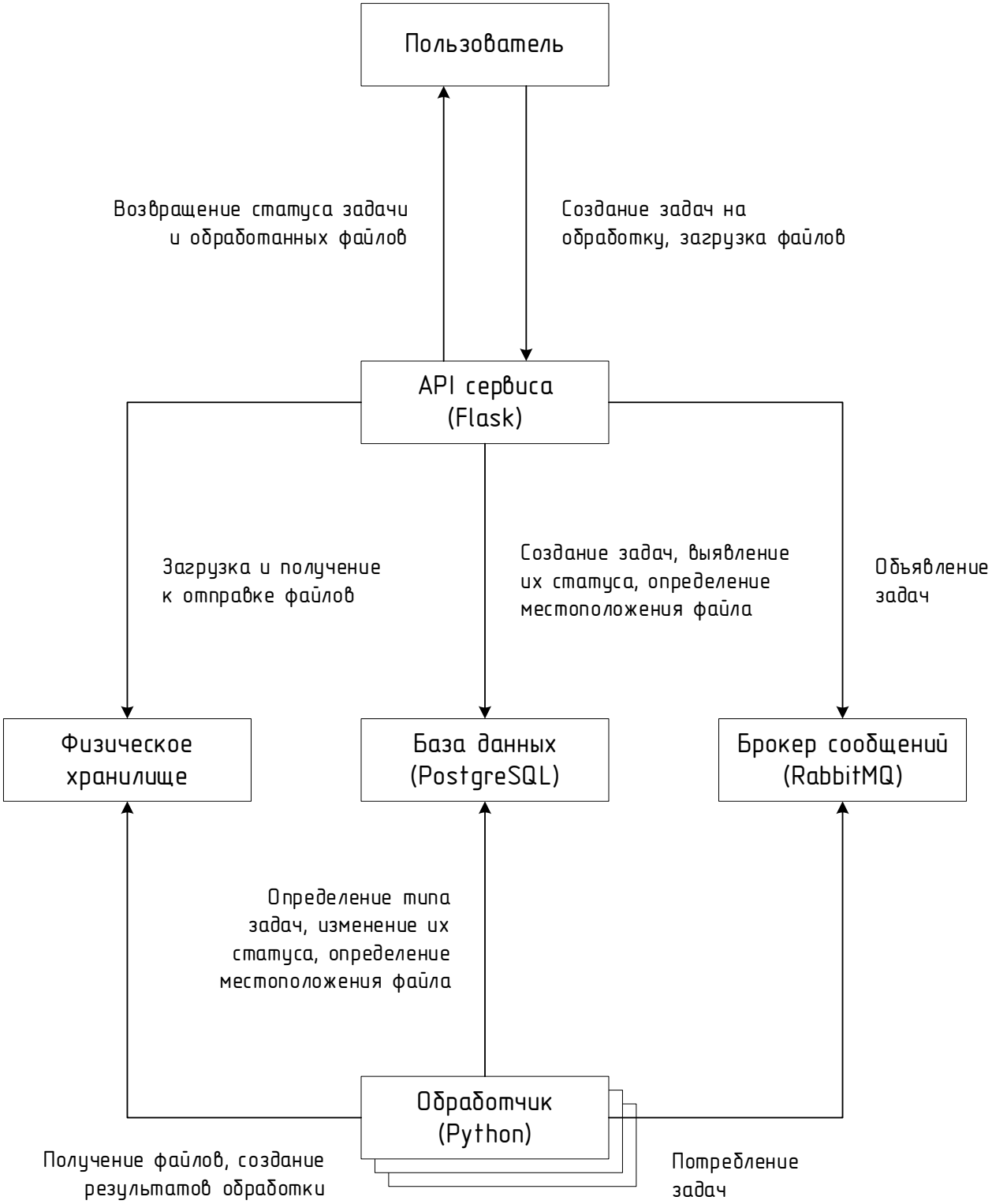

Рис. 1. Структура системы

Так как система представляет из себя серверную часть, взаимодействие с ней пользователем, например, клиентским приложением, предполагается вести через API. Обращаясь к нужным методам, пользователь может загружать файлы с указанием требуемого типа обработки и соответствующих параметров, проверять статус обработки и скачивать готовый результат.

Сервер системы, отвечающий за обработку API, при получении новой задачи сохраняет загруженный файл на физическом хранилище, создает соответствующую задаче запись в базе данных, где также указывается местоположение исходного файла, и заносит сообщение о задаче в очередь брокера сообщений.

Чтобы определить статус выполнения задачи, сервер делает обращение в базу данных. Если задача оказывается завершенной, то появляется возможность возвращения готового результата. Местоположение обработанного файла можно получить из записи соответствующей задачи в базе данных.

Предполагается запуск сразу нескольких обработчиков. При необходимости увеличения мощности работы имеется возможность простого увеличения числа обработчиков. Изначально обработчики производят прослушивание очереди брокера сообщений. Во время появления новой задачи один из обработчиков забирает данную задачу себе. Затем через обращение к базе данных определяется тип алгоритма обработки и необходимые параметры, а также местоположение исходного файла. Завершив непосредственную обработку, обработчик сохраняет результат на физическом хранилище и указывает его в базе данных, а затем возвращается к прослушиванию очереди брокера сообщений.

Кроме того, на протяжении всех этапов работы с задачей обработчик изменяет ее статус в базе данных. Изначально все задачи при своем создании получают статус «pending». Когда задача попадает к обработчику, тот обновляет ее статус до «processing». После успешного завершения обработки устанавливается статус «finished». В случае возникновения ошибок обработчик

указывает статус «error», прекращает обработку задачи и возвращается к брокеру сообщений.

Для избегания возникновения конфликтов имен при сохранении файлов на физическом хранилище предполагается выдавать им уникальные имена. Исходные и итоговые же имена будут храниться в соответствующей записи в базе данных.

Таким образом, структура базы данных имеет вид:

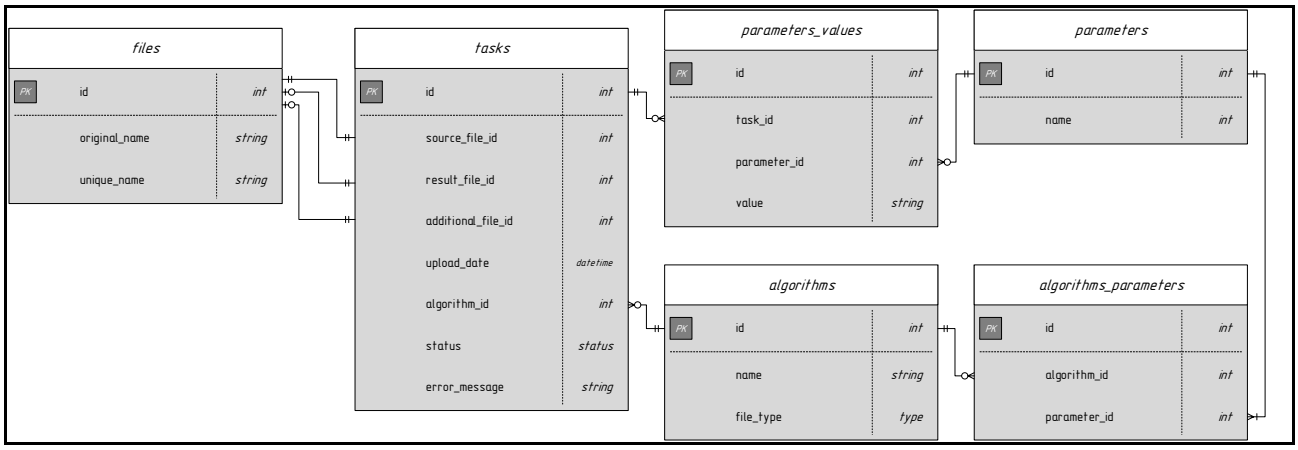

Рис. 2. Структура базы данных

База данных состоит из следующих таблиц (см. рис. 2):

- files – таблица загруженных и готовых обработанных файлов;
- tasks – таблица, хранящая созданные пользователями задачи на обработку файлов;
- algorithms – таблица доступных алгоритмов;
- parameters – таблица параметров, необходимых для работы алгоритмов;
- algorithms_parameters – таблица реализации связи «многое-ко-многим» между таблицами algorithms и parameters;
- parameters_values – таблица, устанавливающая соответствие между задачами на обработку и заданными параметрами;
- status – предоставляет этапы обработки задачи:
    - pending – ожидает выполнения,
    - processing – обрабатывается,
    - finished – выполнено,

- error – закончилось ошибкой;

- type – предоставляет доступные для обработки типа мультимедиа:
  - image – изображения,
  - audio – аудио,
  - video – видео.

Таблицы имеют следующие столбцы (см. табл. 1):

Таблица	Название	Тип	NN	U	PK	FK	Default
files	id	integer	+	+	+		
	Идентификатор						
	original_name	string	+				
	Исходное имя загруженного файла						
	unique_name	string	+				
	Уникальное имя файла, находящегося на физическом хранилище						
tasks	id	integer	+	+	+		
	Идентификатор						
	source_file_id	integer	+			files.id	
	Исходный загруженный файл						
	result_file_id	integer				files.id	
	Итоговый обработанный файл						
	additional_file_id	integer				files.id	
	Дополнительный файл, используемый в некоторых алгоритмах						
	upload_date	datetime	+				
	Дата создания задачи						
	algorithm_id	integer	+			algorithms.id	
	Алгоритм обработки файла						
	status	status	+				pending
	Этап выполнения задачи						
	error_message	string					
	Сообщение ошибки, произошедшей во время выполнения алгоритма						
algorithms	id	integer	+	+	+		
	Идентификатор						
	name	string	+				
	Название алгоритма						
	file_type	type	+				
	Тип обрабатываемого файла						

algorithms_parameters	id	integer	+	+	+		
	Идентификатор						
	algorithm_id	integer	+			algorithms.id	
	Алгоритм						
	parameter_id	integer	+			parameters.id	
	Параметр, необходимый для выполнения алгоритма						
parameters	id	integer	+	+	+		
	Идентификатор						
	name	string	+				
	Название параметра						
parameters_values	id	integer	+	+	+		
	Идентификатор						
	task_id	integer	+			tasks.id	
	Задача						
	parameter_id	integer	+			parameters.id	
	Параметр алгоритма						
	value	string	+				
	Значение параметра						

Табл. 1. Столбцы таблиц базы данных

ЗАКЛЮЧЕНИЕ

В результате прохождения производственно-технологической практики было разработано техническое задание для системы онлайн-сервиса обработки изображений, видео и аудио. Были определены возможные алгоритмы обработки мультимедиа и их теоретическая реализация, рассмотрены основные конкуренты, предлагаемый ими функционал и тарифные планы. Был произведен выбор из списка возможных языков программирования, брокеров сообщений, баз данных, веб-серверов и платформ контейнеризации. Также была составлена структура будущей системы с описанием ее элементов и их взаимодействия. Созданная система позволит множеству пользователей без особых усилий и больших затрат быстро и легко производить изменение своих файлов мультимедиа без необходимости установки на свой ПК дополнительного программного обеспечения.