



Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет  
им. И. Раззакова

Институт информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

ОТЧЕТ

На тему «Инструмент мультиконвертер — веб-сайт с базой данных SQLite и генерацией изображений через DALL·E 3 (размер 1024×1024)»

Выполнили: студенты группы ПИ-2-21

Турдахунов Алмир,  
Ордобаев Амирхан

Проверила: Сабаева Кундуз Кубанычбековна

Бишкек 2025

## Оглавление

- Введение – 3
- 1.1. Актуальность работы – 4
- 1.2. Цель работы – 4
- 1.3. Спецификация проблемы – 4
- 1.4.1. Назначение – 5
- 1.4.2. Цели – 5
- 1.5. Требования к системе – 6
- 1.5.1. Функциональные требования – 6
- 1.5.2. Требования к пользовательскому интерфейсу – 7
- 1.5.3. Требования по безопасности – 7
- 1.5.4. Требования к надежности и масштабируемости – 8
- 1.5.5. Требования к лингвистическому обеспечению – 8
- 1.5.6. Требования к программному и информационному обеспечению – 8
- 1.5.7. Требования к клиентской части – 9
- 1.5.8. Требования к серверной части – 9
- 1.5.9. Требования к организационному обеспечению – 9
- 1. Проектная часть – 10
- 2.1. Диаграмма классов – 10
- Порядок контроля и разработки системы – 11
- Заключение – 12

# Введение

Инструмент мультиконвертер – это веб-сайт, предназначенный для предоставления пользователям единой платформы для конвертации различных типов данных и файлов. Данный проект направлен на создание универсального сервиса, который объединяет в себе функции конвертации изображений, аудио, видео, документов, архивов, а также дополнительные возможности, такие как генерация изображений с помощью искусственного интеллекта и перевод текстов. Благодаря интеграции современных технологий (например, GPT API для перевода и DALL·E 3 для генерации изображений), мультиконвертер позволит пользователям быстро и удобно выполнять разнообразные задачи, связанные с преобразованием информации, без необходимости использовать несколько отдельных приложений или сервисов.

Проект мультиконвертера ориентирован на широкую аудиторию – от студентов и офисных работников, которым требуется конвертировать документы или форматировать изображения, до контент-создателей и IT-специалистов, которым может понадобиться преобразование медиафайлов или текстов. Веб-сайт обеспечивает доступность сервиса из любого современного браузера, без потребности в установке специального программного обеспечения. Интуитивно понятный интерфейс с поддержкой drag-and-drop загрузки файлов и гибкими настройками позволит пользователям с любым уровнем технической подготовки легко воспользоваться всеми возможностями платформы.

Особенностью предлагаемого решения является объединение разнообразных функций конвертации и связанных сервисов в одном месте. Пользователь сможет, например, загрузить изображение и конвертировать его в нужный формат, тут же распознать на нем текст (OCR), перевести полученный текст на другой язык с помощью GPT, либо сгенерировать новое изображение по текстовому описанию через DALL·E 3. Такой комплексный подход повысит эффективность работы пользователей, сэкономит их время и минимизирует необходимость переключаться между разными инструментами. Кроме того, в системе предусмотрена регистрация и авторизация пользователей, что дает возможность вести личный кабинет с историей конвертаций, настройками (например, выбор темы оформления) и другими персонализированными функциями.

Проект направлен на создание надежного, масштабируемого и безопасного веб-приложения, способного обработать значительные объемы данных. Внедрение мультиконвертера будет способствовать повышению удобства работы с информацией, снижению временных затрат на преобразование файлов, а также продемонстрирует практическое применение современных технологий (искусственного интеллекта и облачных сервисов) для решения повседневных задач пользователей.

## Глава 1. Анализ и разработка требований

### 1.1. Актуальность работы

В современной цифровой среде пользователям регулярно требуется преобразовывать файлы из одного формата в другой: изображения, аудио, видео, документы и другие типы данных. Обычно для каждой категории файлов используются отдельные программы или онлайн-сервисы, что неудобно и отнимает время. Например, для конвертации PDF-файла в DOCX требуется один инструмент, для преобразования изображения из PNG в JPEG – другой, а для скачивания видео с социальных платформ – третий. Отсутствие единой универсальной системы приводит к тому, что

пользователи вынуждены искать и применять разрозненные решения, сталкиваясь с ограничениями бесплатных сервисов, проблемами совместимости и безопасности.

Актуальность разработки инструмента мультиконвертера обусловлена потребностью в унифицированном подходе к конвертации данных. Такой веб-сервис позволит значительно упростить рабочий процесс при работе с разнородными файлами. Вместо множества специализированных приложений будет достаточно одного веб-приложения, доступного на любых устройствах. Кроме того, интеграция возможностей перевода текста с помощью моделей искусственного интеллекта (GPT) и генерации изображений (DALL·E 3) расширяет сферу применения системы, делая ее не только конвертером форматов, но и интеллектуальным ассистентом для творчества и работы. В условиях быстрого роста объемов данных и разнообразия форматов универсальный мультиконвертер становится востребованным решением, способным повысить эффективность и продуктивность пользователей.

## 1.2. Цель работы

Целью данной работы является разработка и реализация веб-приложения «Инструмент мультиконвертер», объединяющего в себе множество функций по преобразованию данных. Основная задача проекта – создать многофункциональную платформу, которая позволит пользователям:

- Конвертировать изображения между различными форматами (PNG, JPEG, WebP и др.);
- Конвертировать аудиофайлы в популярные форматы (MP3, WAV, AAC и др.);
- Конвертировать видеофайлы (MP4, AVI, MOV и т.д.) с возможностью изменения разрешения или качества;
- Архивировать и разархивировать файлы (поддержка форматов ZIP, RAR, 7z);
- Конвертировать офисные документы и таблицы (DOCX, PDF, XLSX и др.) для обеспечения совместимости;
- Выполнять перевод текстов на различные языки с использованием GPT API;
- Загружать видео по ссылкам с YouTube, Instagram, TikTok для последующего офлайн-просмотра или конвертации;
- Распознавать текст на изображениях (OCR) с сохранением результата в редактируемом формате;
- Генерировать изображения по текстовому описанию посредством DALL·E 3 (разрешение 1024×1024 пикселей).

Реализация всех перечисленных возможностей в единой системе призвана упростить пользователям работу с информацией. Проект также ставит целью обеспечить удобный и безопасный сервис: предоставить пользователям персональные аккаунты, защиту данных и стабильную работу даже при высокой нагрузке.

## 1.3. Спецификация проблемы

Анализ существующих решений и потребностей пользователей выявил следующие проблемы и ограничения, которые предполагается решить в рамках проекта мультиконвертера:

- **Фрагментация инструментов:** Для разных типов конвертации (медиа, документы, текст) требуется использование отдельных программ или веб-сервисов. Это усложняет рабочий процесс, заставляя пользователей переключаться между приложениями и веб-сайтами.

- **Ограниченная функциональность отдельных сервисов:** Многие бесплатные онлайн-конвертеры имеют ограничения по размеру файлов, скорости работы или навязчивую рекламу. Кроме того, они редко предлагают более одного-двух видов конвертации, не покрывая полный спектр потребностей.
- **Отсутствие интеграции с ИИ-сервисами:** Существующие конвертеры практически не объединяют традиционную смену форматов с возможностями AI, такими как перевод на естественные языки или генерация контента. Пользователям приходится отдельно обращаться к переводчикам и генераторам изображений.
- **Проблемы с безопасностью и конфиденциальностью:** Загрузка файлов на сторонние сервисы конвертации может нести риски утечки конфиденциальных данных, если сервис не гарантирует защиту информации. Также нет единой системы контроля доступа – пользователи не имеют личных аккаунтов на большинстве простых конверторов, из-за чего невозможна персонализация или сохранение истории операций.
- **Недостаточная автоматизация и удобство:** Отсутствует возможность пакетной обработки (batch-конвертации) или постановки задач в очередь на многих сервисах. Пользователь вынужден вручную ждать завершения каждой операции. Также зачастую отсутствуют уведомления об окончании процесса, что неудобно при обработке больших файлов.

Перечисленные проблемы снижают эффективность работы с файлами и данными. Проектируемый мультиконвертер нацелен на преодоление этих недостатков путем создания универсального, безопасного и интегрированного решения.

### 1.4.1. Назначение

Разрабатываемая система «Инструмент мультиконвертер» предназначена для обеспечения пользователей единым центром конвертации и обработки данных. Система позволит в режиме «одного окна» выполнять все необходимые преобразования файлов и информации, избавляя от необходимости использовать множество разрозненных средств. Мультиконвертер будет полезен в различных сценариях:

- Обычным пользователям и офисным работникам – для конвертации документов в нужные форматы, сжатия файлов для пересылки, изменения формата изображений и т.д.;
- Студентам и преподавателям – для перевода текстов на иностранные языки, подготовки материалов в требуемых форматах, извлечения текста из изображений (например, из отсканированных документов);
- Контент-мейкерам – для обработки видео и аудио (конвертация под нужные устройства или платформы), скачивания видеоконтента с социальных сетей для дальнейшего монтажа, генерации иллюстраций по описанию;
- IT-специалистам – для быстрого доступа к инструментам преобразования данных (например, конвертации JSON/CSV таблиц, подготовке данных для различных систем) в удобном веб-интерфейсе.

Таким образом, назначение системы заключается в предоставлении широкого спектра сервисов конвертации и связанных функций на одной платформе, доступной круглосуточно через Интернет. Это позволит повысить эффективность работы пользователей, обеспечить сохранность их данных (благодаря локальному развертыванию базы данных SQLite и контролю доступа) и продемонстрировать преимущества комплексного подхода к обработке информации.

## 1.4.2. Цели

Для реализации назначения системы сформулированы следующие основные цели разработки мультиконвертера:

- **Унификация процессов конвертации** – создать единое веб-приложение, способное обрабатывать изображения, аудио, видео, документы и другие файлы, обеспечивая конвертацию между распространенными форматами.
- **Интеграция ИИ-технологий** – встроить в систему современные API, такие как GPT для интеллектуального перевода текстов и DALL·E 3 для генерации изображений по описанию, расширяя функциональность сервиса за пределы простого изменения форматов.
- **Повышение удобства и эффективности** – разработать удобный пользовательский интерфейс с поддержкой drag-and-drop, системы уведомлений (оповещение пользователя по завершении конвертации или при возникновении ошибок), а также возможность настройки параметров (например, выбор темной или светлой темы оформления, язык интерфейса).
- **Безопасность и персонализация** – реализовать механизм регистрации и авторизации пользователей с хранением учетных данных и истории операций в защищенной базе данных. Обеспечить контроль доступа к личному кабинету, где пользователь сможет просматривать выполненные конвертации, повторно скачивать результаты, настраивать предпочтения.
- **Масштабируемость и надежность** – спроектировать систему таким образом, чтобы она могла обрабатывать несколько запросов одновременно, используя очередь задач для крупных файлов или длительных операций. Обеспечить стабильную работу сервера (на базе Python Flask и Kotlin Vert.x) при увеличении числа пользователей, предусмотрев возможности горизонтального или вертикального масштабирования.

Достижение перечисленных целей гарантирует, что созданный мультиконвертер будет отвечать потребностям пользователей, обеспечит широкий функционал и высокое качество сервиса.

## 1.5. Требования к системе

На основе проведенного анализа требований и целей проекта сформулированы следующие требования к разрабатываемой системе.

### 1.5.1. Функциональные требования

Система должна предоставлять следующий функционал для конечных пользователей:

- **Регистрация и авторизация пользователей:** Возможность создания новой учетной записи (регистрация) и входа в систему (авторизация) для персонального доступа к сервису. После входа пользователь получает доступ к своему личному кабинету и истории конвертаций.
- **Конвертация изображений:** Инструмент для загрузки изображения (форматы входных файлов: PNG, JPEG, WebP, BMP и др.) и сохранения его в другом формате, выбранном пользователем. Пользователь должен иметь возможность настроить базовые параметры (например, качество JPEG, разрешение) при конвертации.
- **Конвертация аудио:** Инструмент для преобразования аудиофайлов между распространенными форматами (MP3, WAV, AAC, FLAC и др.). Пользователь загружает

аудио, выбирает целевой формат, при необходимости может указать параметры (битрейт, стерео/моно).

- **Конвертация видео:** Средство для конвертации видеофайлов (MP4, AVI, MOV, MKV и т.д.) в другие форматы. Опционально предусматривается возможность выбора разрешения выходного видео, качества (битрейт) и извлечения аудиодорожки. При обработке больших видео реализуется отображение прогресса.
- **Конвертация архивов:** Функционал создания и распаковки архивов. Пользователь может загрузить архив (ZIP, RAR, 7z) для извлечения содержимого либо загрузить несколько файлов/папок и получить на выходе архив выбранного формата. Система должна корректно обрабатывать паролированные архивы (запрашивать пароль у пользователя) и поддерживать различные методы сжатия.
- **Конвертация документов и таблиц:** Возможность преобразования офисных файлов – например, DOCX в PDF и наоборот, XLSX (Excel) в CSV, PPTX (презентации) в PDF, и т.п. Система обеспечивает корректное сохранение содержимого (текст, изображения, форматирование) при конвертации. Для реализации данного функционала могут использоваться внешние библиотеки или сервисы, совместимые с Python (например, библиотеки *python-docx*, *PyPDF2* или интеграция с LibreOffice в серверном режиме).
- **Перевод текста:** Интеграция с GPT API для перевода введенного пользователем текста. Пользователь указывает исходный текст (или загружает текстовый файл) и выбирает целевой язык перевода. Система отправляет запрос к языковой модели (GPT) и выводит полученный перевод. Важно предусмотреть ограничения на размер переводимого текста и отобразить примерное время выполнения операции, если текст большой.
- **Загрузка видео по ссылке:** Функция, позволяющая пользователю вставить ссылку на видео из YouTube, Instagram, TikTok (или других поддерживаемых платформ) и получить возможность скачать это видео на свое устройство либо сразу передать его в модуль конвертации (например, для извлечения аудио или изменения формата). Должна быть реализована проверка валидности ссылки и вывод ошибок (например, если видео недоступно или защищено).
- **Оптическое распознавание символов (OCR):** Средство для распознавания текста на загруженном изображении. Пользователь выбирает изображение (скан документа, фотография текста и т.д.), система выполняет OCR и выдает пользователю извлеченный текст (в окне вывода или в виде файла .txt). Желательно поддерживать распознавание русского и английского языков как минимум, используя OCR-движок (например, Tesseract OCR) с соответствующими языковыми пакетами.
- **Генерация изображений:** Интеграция с моделью DALL-E 3 для генерации изображения по описанию. Пользователь вводит текстовый запрос (prompt) на русском или английском языке, система передает его в API генерации изображений и получает сгенерированное изображение разрешением 1024×1024 пикселей. Полученное изображение отображается пользователю с возможностью скачать. Необходимо ограничить количество подобных запросов в минуту для одного пользователя (во избежание перегрузки или злоупотребления API).

Кроме перечисленного основного функционала, система должна предоставлять пользователю **личный кабинет**, где будут доступны следующие возможности:

- Просмотр профиля и редактирование личных данных (имя, email, пароль);
- Просмотр истории выполненных операций (список последних конвертаций, переводы, OCR и т.д. с указанием даты/времени и результата, возможно с прямой ссылкой на полученные файлы);
- Настройки интерфейса (переключение темы оформления – светлая/темная, выбор языка интерфейса, если будет реализована мультиязычность);

- Выход из аккаунта (logout) и удаление аккаунта при необходимости.

### 1.5.2. Требования к пользовательскому интерфейсу

Пользовательский интерфейс веб-приложения должен быть интуитивно понятным, современным и отзывчивым. Основные требования к UI:

- **Единый портал для всех функций:** Главная страница после авторизации должна предоставлять доступ ко всем типам конвертации и сервисов (через меню или панель инструментов). Переход между разделами (изображения, видео, документы, перевод, и т.д.) – быстрый и не перегружающий пользователя лишней информацией.
- **Drag-and-drop загрузка:** Реализовать поддержку перетаскивания файлов в область загрузки для всех модулей конвертера. При перетаскивании файлов соответствующий модуль (например, "Конвертация изображений") должен подсвечиваться, и файлы автоматически подгружаются.
- **Отображение прогресса и уведомления:** В интерфейсе необходимо отображать индикатор выполнения для операций, которые занимают заметное время (например, конвертация видео, генерация изображения через ИИ). По завершении операции пользователь должен получать понятное уведомление (например, всплывающее сообщение или отметку рядом с соответствующим модулем) об успехе или ошибке. Также нужна система уведомлений/оповещений, информирующая, например, об окончании конвертации, если пользователь переключился на другой раздел или вкладку.
- **Адаптивность:** Интерфейс должен корректно отображаться на различных устройствах и экранах. Необходимо обеспечить адаптивную верстку, чтобы пользователи могли пользоваться мультikonвертером как с компьютера, так и с планшета или смартфона. Меню и элементы управления должны подстраиваться под размер экрана (в мобильной версии – сворачиваться в выпадающее меню, кнопки — укрупняться для сенсорного ввода и т.д.).
- **Настраиваемость внешнего вида:** Предусмотреть возможность переключения между светлой и темной темой оформления интерфейса. Выбранная тема сохраняется в настройках пользователя и применяется при последующих входах. Цветовая схема и шрифты должны обеспечивать комфортное восприятие информации, контрастный текст и иконки на любом фоне.
- **Локализация на русском языке:** Все элементы интерфейса (надписи на кнопках, подсказки, сообщения об ошибках и успехе) по умолчанию отображаются на русском языке, с использованием корректной технической терминологии, понятной целевой аудитории. Должна быть проверена читаемость шрифтов для кириллицы. В дальнейшем, при необходимости, интерфейс может поддерживать и другие языки, но русский – основной по требованиям проекта.

### 1.5.3. Требования по безопасности

Система должна обеспечивать надежную защиту данных пользователей и результатов их конвертаций, а также безопасное выполнение всех операций. Основные положения:

- **Контроль доступа и авторизация:** Весь расширенный функционал (кроме, возможно, самых простых конвертаций) доступен только авторизованным пользователям. Неавторизованные пользователи могут видеть главную страницу, но чтобы воспользоваться сервисом, им необходимо войти или зарегистрироваться. Таким образом обеспечивается персональный доступ и отслеживание действий в рамках аккаунта.
- **Хранение данных:** Учетные данные (логины, пароли, email) хранятся в базе данных SQLite. Пароли должны храниться в зашифрованном виде (хеширование с солью,

используя современные алгоритмы, например, bcrypt). Доступ к самой базе из вне (из Интернета) должен быть закрыт – взаимодействие идет только через серверное приложение.

- **Шифрование связи:** Рекомендуется разворачивать веб-приложение на HTTPS (SSL-сертификат) для шифрования трафика, особенно при передаче учетных данных, файлов для конвертации и результатов. Это предотвратит перехват конфиденциальной информации на пути от клиента к серверу.
- **Ограничение доступа к файлам:** Файлы, загруженные пользователями для конвертации, хранятся на сервере временно (на время обработки) и удаляются после выдачи результата. Доступ к загруженным файлам и результатам должен иметь только владелец (инициировавший конвертацию пользователь) – например, посредством уникальных временных ссылок или через защищенный личный кабинет.
- **Защита от вредоносных данных:** Система должна проверять типы и размеры загружаемых файлов (белый список расширений, ограничения на максимальный размер файла) и отклонять потенциально опасные файлы (например, исполняемые файлы, замаскированные под другой формат). Это минимизирует риск выполнения вредоносного кода на стороне сервера при попытке обработки непредусмотренных форматов.
- **Логирование и мониторинг:** На серверной стороне необходимо вести ограниченное логирование ключевых событий (входы в систему, попытки доступа, ошибки конвертации). Это поможет в обнаружении и предотвращении подозрительной активности (например, множественных неудачных попыток входа – потенциал brute force атаки, или аномально большого числа запросов от одного пользователя – потенциальный DoS или злоупотребление API).

#### 1.5.4. Требования к надежности и масштабируемости

Разрабатываемое приложение должно сохранять работоспособность и целостность данных при различных внештатных ситуациях, а также быть готовым к увеличению нагрузки:

- **Отказоустойчивость:** В случае сбоя оборудования или программного обеспечения (например, перезагрузки сервера, сбоя веб-сервера Vert.x или приложения Flask) система должна автоматически восстанавливаться при повторном запуске. Данные пользователей и история операций, находящиеся в базе SQLite, не должны повреждаться из-за некорректного завершения работы приложения. Рекомендуется периодическое резервное копирование базы данных.
- **Обработка сбоев конвертации:** Если при конвертации файла происходит ошибка (например, поврежденный входной файл, потеря связи с внешним API, внутренний сбой библиотеки), система должна корректно обработать исключение, уведомить пользователя о неудаче, не завершая работу всего приложения. Отказ одного модуля (скажем, сервис OCR) не должен влиять на работу других функций.
- **Производительность:** Система должна эффективно работать при одновременном обращении нескольких пользователей. За счет использования Vert.x (который обеспечивает асинхронность и высокую пропускную способность) и многопоточности Python (например, через Celery или встроенные возможности Flask для асинхронной обработки) следует добиться минимального времени отклика для небольших файлов и линейного масштабирования времени обработки для больших файлов. Время конвертации крупных видео или документов информирует пользователя индикатором прогресса, чтобы избежать ощущения "зависания".
- **Масштабируемость:** Архитектура системы должна допускать увеличение ресурсов по мере роста числа пользователей и объема данных. Это может быть реализовано путём развертывания нескольких экземпляров серверной части (Flask) за балансировщиком



нагрузки или распределения различных типов задач по разным сервисам (микросервисный подход, например, выделить отдельный сервис для тяжелых видео-конвертаций). Выбор Kotlin + Vert.x для веб-сервера обеспечивает потенциальную горизонтальную масштабируемость при необходимости. Также SQLite при усложнении задач может быть заменена на более мощную СУБД (например, PostgreSQL) без серьезных изменений в логике приложения, что следует учесть как опцию.

### 1.5.5. Требования к лингвистическому обеспечению

Интерфейс и взаимодействие с пользователем в системе должны осуществляться на русском языке:

- Все меню, кнопки, уведомления, сообщения об ошибках – на русском, с использованием общепринятых терминов (например, "Загрузить файл", "Конвертировать", "Ошибка: недопустимый формат" и т.п.). Текст должен быть грамотно составлен и легко понятен пользователю.
- Поддержка ввода текста на русском языке (например, для названий файлов, текстовых данных на перевод или генерацию изображений). Система должна корректно передавать русскоязычный текст в API (особенно важно для GPT перевода и DALL-E генерации, чтобы специальным образом кодировать или обрабатывать строку, если требуется).
- Документация пользователя (как воспользоваться сервисом) и справочная информация, если она будет доступна в интерфейсе, также представляются на русском языке.

*(Примечание: В дальнейшем, по мере развития проекта, можно предусмотреть мультиязычный интерфейс, добавив английский и другие языки, однако в рамках текущих требований достаточно реализации на русском.)*

### 1.5.6. Требования к программному и информационному обеспечению

При разработке системы должны использоваться современные технологии и инструменты, которые являются общедоступными и хорошо зарекомендовали себя:

- **Среда разработки и язык программирования (сервер):** Основная логика веб-приложения создается на языке Python с использованием микро-фреймворка Flask. Это обеспечивает быструю разработку и наличие множества библиотек для работы с различными форматами файлов и API. Для обеспечения высокопроизводительной обработки запросов на уровне веб-сервера используется технология Vert.x (с использованием языка Kotlin). Kotlin, будучи платформой JVM, хорошо сочетается с Vert.x, а использование Vert.x позволит эффективно обслуживать большое число одновременных подключений.
- **СУБД:** В качестве системы управления базой данных используется SQLite (не ниже версии 3). SQLite – легковесная встроенная база данных, которая не требует развертывания отдельного сервера и подходит для учебного проекта и начальных этапов разработки. Она будет хранить информацию о пользователях, их настройках и истории операций. В дальнейшем допускается переход на PostgreSQL или другую реляционную СУБД при росте требований к масштабируемости.
- **Внешние API и библиотеки:**
  - Для реализации перевода текста – API GPT (например, OpenAI GPT-4 или аналог) через соответствующий SDK или HTTP-запросы.
  - Для генерации изображений – API DALL-E 3 (предоставляется OpenAI), который будет вызываться с заданным prompt и возвращать изображение.

- Для OCR – библиотека Tesseract OCR (с русским языковым пакетом) через обертку, например *pytesseract*.
- Для конвертации медиа (аудио/видео) – можно использовать библиотеку *FFmpeg* (с оберткой *ffmpeg-python* или вызывать утилиту из Flask), либо специальные библиотеки как *moviepy* для видео, *pydub* для аудио.
- Для документов – *python-docx*, *xlsx2csv*, *PyPDF2* и другие по необходимости.
- Для скачивания видео по ссылкам – библиотека *youtube\_dl* или ее обновление *yt-dlp*, позволяющая программно загружать видео/аудио из многих источников.
- **Фронтенд:** Интерфейс реализуется с использованием стандартных веб-технологий – HTML5 для структуры страниц, CSS3 (возможно с использованием фреймворка типа Bootstrap для ускорения адаптивной верстки) для оформления и JavaScript (Vanilla JS или с использованием небольших библиотек вроде jQuery) для динамического поведения. Также могут применяться современные фреймворки (React, Vue, Angular) по усмотрению разработчиков, однако для учебного проекта достаточно и классического подхода с Flask-шаблонами или простыми JS-скриптами.
- **Контроль версий:** Рекомендуется использовать систему контроля версий (например, Git) во время разработки для отслеживания изменений в коде и совместной работы участников команды.
- **Средства разработки:** Visual Studio Code, PyCharm или IntelliJ IDEA (для Kotlin) могут использоваться как основные инструменты IDE. Эти среды широко распространены и обладают всеми необходимыми возможностями для написания и отладки кода.

### 1.5.7. Требования к клиентской части

Клиентская часть системы – это веб-интерфейс, доступный через браузер. Требования:

- **Поддерживаемые браузеры:** Веб-приложение должно корректно работать в актуальных версиях популярных браузеров: Yandex.Browser, Google Chrome, Mozilla Firefox, Opera, Microsoft Edge. Также желательно протестировать работоспособность в Safari для обеспечения поддержки пользователей устройств Apple.
- **Кроссплатформенность:** Так как фронтенд – это веб-страницы, все функции должны быть доступны независимо от операционной системы устройства (Windows, Linux, macOS, Android, iOS и др.), ограничение только одно – наличие современного браузера. Вёрстка и скрипты должны учитывать особенности разных движков (Blink, WebKit, Gecko) и использовать стандартизованные функции для максимальной совместимости.
- **Отсутствие необходимости плагинов:** Для работы с приложением не должны требоваться какие-либо устаревшие плагины (Flash, Java-апплеты). Все реализовано на современных веб-технологиях, встроенных в браузер.
- **Ресурсоёмкость:** Сайт должен загружаться и функционировать приемлемо быстро даже на устройствах со скромными ресурсами (например, смартфонах нескольких лет давности). Это требует оптимизации объема передаваемых данных (минимизация и сжатие скриптов, изображений интерфейса и т.п.).

### 1.5.8. Требования к серверной части

Серверная часть включает в себя backend-логику на Flask (Python) и веб-сервер на Vert.x (Kotlin), а также базу данных:

- **Совместимость ОС:** Серверное ПО должно быть кроссплатформенным. Выбранные технологии позволяют запускать систему как на Windows, так и на Unix-подобных системах (Linux, macOS). Для развёртывания в продакшне предпочтительной является Linux-среда (например, Ubuntu Server), но для разработки может применяться Windows.

- **Запуск и обслуживание:** Веб-сервер Vert.x, написанный на Kotlin, функционирует по принципу event loop (событийный цикл) и способен обрабатывать многочисленные подключения. Он слушает HTTP-запросы от клиентов и перенаправляет их к обработчикам (либо статика, либо вызывает Flask-приложение для получения динамического результата). Сервер Flask выполняет непосредственную обработку конвертации. Важно, чтобы серверная часть была запущена постоянно (в виде службы/демона) и автоматически перезапускалась в случае сбоя.
- **Производительность и параллелизм:** Vert.x обеспечивает неблокирующую обработку запросов, а Flask может быть запущен с поддержкой нескольких рабочих процессов (через WSGI-сервер, например Gunicorn, или используя многопоточные возможности). Задачи, требующие значительного времени (видео-конверсия, запрос к AI API), могут исполняться асинхронно (с помощью очереди задач, например Celery + Redis, или встроенных механизмов Vert.x Worker). Это предотвращает блокировку основного потока обработки запросов и повышает общую производительность.
- **Хранение файлов:** Сервер должен иметь достаточно места на диске для временного хранения загружаемых файлов и результатов конвертации. Рекомендуется настроить автоматическую очистку временных файлов после завершения операций, чтобы не переполнить хранилище. Также нужно учитывать ограничение SQLite – хранить большие файлы непосредственно в базе не следует; лучше хранить только пути/ссылки, а файлы – в файловой системе.
- **Безопасность на сервере:** Открытые порты – только необходимые (порт веб-сервера). Доступ к базе SQLite – локальный. Регулярное обновление зависимостей (особенно библиотек, обрабатывающих входящие файлы) для устранения уязвимостей.
- **Масштабирование:** При росте нагрузки серверная часть может быть запущена в распределенной конфигурации: несколько экземпляров бекенда за общим балансировщиком. SQLite в этом случае может быть заменена на сетевую СУБД (PostgreSQL/MySQL), которая позволит конкурентный доступ. Код приложения должен быть написан с учетом возможности такой миграции (например, использовать ORM, поддерживающее разные СУБД).

### 1.5.9. Требования к организационному обеспечению

Эффективная эксплуатация и сопровождение системы требуют соблюдения определенных организационных мер:

- **Роли персонала:** Для поддержки работы веб-приложения должен быть назначен администратор системы (или группа администраторов). Их задачи – мониторинг работоспособности сервера, управление учетными записями пользователей при необходимости (например, сброс пароля по запросу, блокировка злоумышленников) и модерация контента (если в будущем появится возможность публичного обмена результатами).
- **Регламент обслуживания:** Должны быть определены ответственные лица за резервное копирование данных (выгрузка SQLite базы с определенной периодичностью), обновление системы (развертывание новых версий, установка обновлений безопасности) и обработку обращений пользователей (например, через электронную почту поддержки).
- **Документация:** К системе следует подготовить пользовательскую инструкцию (описание основных шагов по конвертации, переводу и другим функциям) и техническую документацию для администраторов (описание установки, конфигурации сервера, структуры базы данных). Эти документы облегчат ввод в эксплуатацию и дальнейшее сопровождение.

- **Приватность и соглашения:** При внедрении сервиса в реальную эксплуатацию необходимо разработать политику конфиденциальности и пользовательское соглашение, так как система обрабатывает пользовательские файлы и возможно хранит их личные данные. Пользователи должны быть проинформированы об условиях использования сервиса, ответственности сторон и ограничениях (например, запрет на использование сервиса для нелегального контента).
- **Обратная связь от пользователей:** Организация канала для предложений и отчетов о проблемах (например, форма обратной связи на сайте или форум) поможет в дальнейшем улучшать систему, обнаруживать скрытые ошибки и добавлять востребованные функции.

## Глава 2. Проектная часть

### 2.1. Диаграмма классов

Диаграмма классов для системы мультиконвертера отражает основные сущности приложения и связи между ними. В центре находится класс пользователя (User), связанный с сущностями, представляющими задачи конвертации (например, класс ConversionTask или отдельные классы для каждого типа задачи: ImageConversion, AudioConversion и т.д.). Также присутствуют классы для управления файлами (UploadedFile, ConvertedFile), класс для взаимодействия с внешними API (например, AIService для GPT/DALL-E) и другие вспомогательные сущности (например, класс Notification для управления уведомлениями). На диаграмме показаны отношения ассоциации между пользователем и его задачами, а также связи между задачами и результирующими файлами в базе данных.

*Рисунок 1. Диаграмма классов.*

## Порядок контроля и разработки системы

Для реализации проекта «Инструмент мультиконвертер» разработан план-график, разбитый на 5 этапов. Каждый этап сопровождается контрольными точками (процент завершения) и имеет ответственных исполнителей (двое разработчиков). Ниже представлено распределение задач по этапам, процент выполнения и сроки.

- 1. Этап 1 (20%, до 15.10.2025)** – Выполнение базовых работ по пользовательской части. Реализованы регистрация и авторизация пользователей (Турдахунов Алмир), а также разработан личный кабинет пользователя с возможностями редактирования профиля и просмотра истории операций (Ордобаев Амирхан). На этом этапе заложена структура базы данных (таблица пользователей, базовые таблицы для хранения задач).
- 2. Этап 2 (40%, до 30.10.2025)** – Реализация основных функций конвертации для медиа. Выполнена разработка модуля конвертации изображений между форматами (Турдахунов) и модуля конвертации аудио файлов (Ордобаев). Интерфейс дополнен соответствующими разделами, обеспечен вывод результатов конвертации и возможность скачивания. Начата интеграция библиотеки для работы с медиа (FFmpeg) и тестирование на примерах файлов.
- 3. Этап 3 (60%, до 15.11.2025)** – Разработка функций для работы с видео и документами. Реализован модуль конвертации видеофайлов (Турдахунов), включая изменение формата и базовые настройки качества. Параллельно разработан модуль конвертации офисных документов (Ордобаев) – например, преобразование DOCX↔PDF, XLSX→CSV. Проведено тестирование этих функций на различных типах файлов. Дополнительно на этом этапе

внедрена очередь задач на сервере для обработки долгих операций (в частности, видео-конвертации) во избежание блокировки основных потоков.

4. **Этап 4 (80%, до 30.11.2025)** – Интеграция интеллектуальных сервисов и OCR. Добавлена возможность перевода текста через GPT API (Турдахунов) – настроен обмен данными с внешним API, создан интерфейс для ввода текста и вывода перевода. Также внедрён OCR функционал (Ордобаев) с использованием Tesseract: пользователи могут загружать изображения с текстом и получать распознанный текст. Оба новых модуля интегрированы в веб-интерфейс. На этом этапе также реализована система уведомлений: пользователю показываются всплывающие сообщения об окончании длительных процессов или возникших ошибках.
5. **Этап 5 (100%, до 15.12.2025)** – Финальная отладка, дополнительные сервисы и улучшения UI/UX. Реализована функция скачивания видео по ссылке (Ордобаев) – интеграция с *youtube-dl/yt-dlp* для поддержки YouTube, Instagram, TikTok. Выполнена интеграция генератора изображений DALL-E 3 (Турдахунов) – пользователи могут вводить описание и получать сгенерированное изображение. Проведена оптимизация интерфейса: добавлена возможность переключения темы (тёмная/светлая), обеспечена полноценная адаптивность для мобильных устройств. Система протестирована комплексно всеми участниками команды, исправлены выявленные баги. Подготовлен пользовательский гид и финальная презентация проекта.

## Заключение

В ходе выполнения данного учебного проекта было разработано техническое задание на создание веб-приложения «Инструмент мультиконвертер». В документе проведен анализ актуальности задачи и существующих проблем, сформулированы цели и назначение системы, а также детально описаны функциональные и нефункциональные требования. Проект предусматривает объединение множества сервисов конвертации и современных технологий (таких как GPT-API и DALL-E 3) на одной платформе, что делает его инновационным и практически значимым.

Разработка мультиконвертера, согласно представленному плану-графику, охватывает все этапы – от создания пользовательской инфраструктуры (аккаунты, интерфейс) до интеграции специализированных модулей для каждого вида конвертации. Особое внимание уделяется удобству использования (drag-and-drop, уведомления, адаптивность), безопасности данных пользователей и возможности масштабирования системы под растущие нагрузки.

Подготовленное техническое задание может служить основой для непосредственной реализации проекта. Выполнение поставленных требований позволит создать востребованный веб-сервис, способный существенно упростить работу пользователей с разнородными форматами данных. В заключение отметим, что предложенное решение не только сочетает уже известные инструменты, но и демонстрирует интеграцию искусственного интеллекта в прикладное веб-приложение, отражая современный тренд развития информационных технологий. Завершение проекта согласно намеченному графику даст участникам ценный опыт командной разработки и проектирования комплексных систем, а полученный мультиконвертер сможет занять свое место среди полезных сервисов, облегчающих жизнь широкому кругу пользователей.