**Введение**

В современном мире оформление текстовых документов является одним из ключевых аспектов деятельности организаций в самых различных направлениях. С каждым годом число текстовых документов увеличивается. При этом, каждый тип документа имеет свои собственные стандарты и правила оформления, а существующие текстовые редакторы не оснащены функционалом для полноценного анализа по всем необходимым требованиям.

Отдельно стоит отметить документы учебного характера, которые создаются учащимися разных организаций. Успешное прохождение нормоконтроля курсовых, выпускных и прочих работ является обязательным для сдачи. Проверка на соответствие документов правилам и нормам, установленным внутри учебного учреждения, и ГОСТам осуществляется преподавателями вручную. Однако такая проверка — трудоемкий процесс, требующий значительных усилий, а человеческий фактор приводит к пропуску ошибок. В случае нахождения несоответствий правилам, документ возвращается на доработку, что впоследствии приводит к увеличению временных затрат и ресурсов как у проверяющего, так и у автора [1].

Ошибки в оформлении могут носить как визуальный, так и логический характер. Контроль оформления — это не только исправление ошибок, проверка шрифта, межстрочных интервалов и отступов, но также обеспечение целостности и соответствия нормативам. От точности соблюдения форматов, стилей и структуры зависит не только восприятие и обработка информации, но и юридическая действительность документа.

Возникает необходимость в автоматизации процесса и создании системы, которая позволит сократить затраты на человеческие ресурсы, а также осуществит анализ с высокой точностью и обеспечит единообразие стилей документации.

Разработка программного средства, способного анализировать текстовые документы на соответствие заданным требованиям и стандартам, является не просто актуальной, но и необходимой.

Объект исследования: автоматизированные системы контроля оформления текстовых документов.

Предмет исследования: алгоритмы и методы, позволяющие осуществлять проверку текстовых документов формата DOCX.

Цель исследования: разработка программного средства автоматизации процесса контроля оформления текстовых документов.

Задачи исследования:

1. анализ теоретических материалов о различных системах проверки документации и существующих стандартов к оформлению документов;
2. анализ существующих методов и решений по проверке оформления текстовых документов;
3. проектирование программного средства, определение ресурсов для выполнения проекта;
4. программная реализация системы.

**1 Обзор предметной области**

**1.1 Общие сведения и понятие контроля оформления текстовых документов**

Оформление текстовых документов играет одну из главных ролей в восприятии. Важнейший фактор, влияющий на это — визуальная и структурная организация, которая достигается благодаря соблюдению установленным стандартам и требованиям.

В современном обществе во многих сферах деятельности текстовый документ должен строго соответствовать определенному ряду правил оформления, которые зависят от типа документа и сферы его применения. Соблюдение этих норм обеспечивает единообразие и упрощает восприятие информации. Однако ручная проверка оформления, особенно при больших объёмах документации, связана с высокой трудоемкостью и риском пропуска ошибок, что делает актуальной разработку автоматизированных решений.

Стандарты и требования могут быть определены как внутри одной организации, так и на государственном уровне. Их нарушение может привезти к отклонению документа, снижения оценки или юридической недействительности.ак и на требования могут быть определены как внутри одной организацииентов111111111111111111111111111111111111111111111111111

Системы контроля оформления текстовых документов представляют собой сложный программный комплекс, объединяющий алгоритмы, методы обработки данных, работа которых направлена на поиск и выявление несоответствий между оформлением созданного документа и заданными стандартами. Они позволяют предотвращать потенциальные ошибки, с помощью автоматических подсказок. Особую значимость такие системы приобретают в условиях увеличения объема документации и ужесточения требований к ее оформлению.

**1.2 Обзор существующих методов и инструментов**

Один из основных и часто встречающихся методов — визуальная проверка текстовых документов сотрудниками. Проверяющий должен вручную проверять их оформление на множество параметров, таких как шрифт, его размер, отступы, выравнивание, порядок нумерации, правильное использование шаблонов, наличие обязательных элементов, оформление ссылок, использованных ресурсов и т.д. Несмотря на широкую распространенность данного подхода, он имеет существенные недостатки и не гарантирует полного соответствия стандартам.

Анализ существующей практики выявляет несколько ключевых проблем ручной проверки текстовых документов:

* отсутствие единого подхода. Разные проверяющие могут по-разному интерпретировать требования стандартов;
* недостаточная надежность. Риск пропуска ошибок возрастает, когда объем документов увеличивается;
* трудоемкость и отсутствие гарантий. Даже особо тщательная проверка не гарантирует полного соответствия стандартам, а исправление ошибок требует дополнительных временных затрат.

Таким образом, система контроля оформления приобретает актуальность в следующих аспектах: соответствие нормативным требованиям, минимизация рисков признания документов недействительными, сокращение количества доработок, снижение затраченных ресурсов на ручную проверку и повышение общей эффективности работы.

**1.2.1 Автоматизированные системы: преимущества и примеры реализации**

Ручные методы постепенно вытесняются автоматизированным системами контроля оформления, которые представляют собой программные средства. Некоторые текстовые редакторы, представляющие собой простые системы предоставляют базовый функционал для проверки орфографии и грамматики, но этого недостаточно для особых требований к оформлению. Примерами реализации подобных систем могут послужить Microsoft Word, Google Docs или LibreOffice, но их возможности ограничены и не отвечают за проверку всех требований к оформлению.

Такие системы характеризуются стабильной производительностью и отсутствием «усталости» при проверке больших документов. Они появились с развитием компьютерных технологий и постепенно эволюционировали от простых проверочных модулей до комплексных решений с интеграцией искусственного интеллекта. На основе заданных алгоритмов и правил они способны анализировать параметры форматирования, обеспечивая стабильные и единообразные результаты.

Основными преимуществами автоматизированных систем являются:

* высокая скорость проверки. Современные системы способны проанализировать весь документ за считанные секунды или минуты, в то время как ручная проверка может занять часы или дни;
* стабильность и однородность результатов. Программа применяет одинаковые критерии проверки ко всем документам одного вида, исключая субъективность, связанную с человеческим фактором;
* масштабируемость. Автоматизированные системы с легкостью справляются с увеличением объема проверяемого документа без снижения качества контроля.

Существуют и более специализированные системы, которые помогают выявить неправильно оформленные заимствования в тексте, они косвенно влияют на качество оформления документа. Их называют плагиат-детекторами, но эти решения в основном ориентированы на частные задачи и не обеспечивают комплексной проверки на соответствие оформления нормативам.

Помимо универсальных текстовых редакторов, существуют специализированные системы, которые рассчитаны на проверку текстов. Одним из таких примеров может стать Антиплагиат, который направлен она определение уникальности. Он позволяет пользователям проверить оригинальность, предоставляя подробные отчеты об выявленных совпадениях и заимствований из других источников. Также, при проведении анализа документа, система проверяет наличие структурных элементов, таких как титульные листы, содержание, введение и остальных, но не осуществляет проверку их оформления. Функционал Антиплагиата ограничен, он решает задачу контроля уникальности, но не охватывает полный спектр требований к оформлению документов, что делает его неприменимым при нормоконтроле.

Ещё одним примером может стать платформа chext.ru. Она помогает при проверке орфографии и стилистики, отслеживает наличие «висячих предлогов», двойных пробелов или отсутствие пунктуации. Этот инструмент нацелен на пользователей, которые стремятся улучшить читаемость и повысить качество текстового контента. Его основной целью является визуальное упрощение текста, поэтому он не может быть универсальным решением для нормоконтроля.

Особое внимание стоит уделить системе Antiplagiat Killer. Она включает в себя не только техническую обработку текста для повышения уникальности, но и модуль для анализа базовых параметров. Проверка отступов, межстрочного интервала, шрифта, начертания и прочего позволяет назвать эту систему более многофункциональной по сравнению с описанными выше аналогами. Однако ее возможности в области нормоконтроля остаются ограниченными: она не поддерживает проверку сложных требований, таких как форматирование изображений, таблиц, или их подписей по требуемым стандартам.

Таким образом, на сегодняшний день существует значительный разрыв между реальными требованиями к оформлению и уровнем автоматизации средств контроля. Существующие системы, несмотря на их преимущества, не обеспечивают полного соответствия реальным потребностям нормоконтроля. Это подчеркивает актуальность разработки более универсальных и интеллектуально гибких решений, способных объединить анализ содержания, стилистики и оформления в единую систему.

**1.3 Анализ требований к автоматизированной системе**

Автоматизированная система контроля оформления текстовых документов должна отвечать ряду ключевых требований, вытекающих как из стандартов документооборота, так и из практических задач пользователей. Эти требования можно разделить на функциональные и нефункциональные, каждая из которых играет важную роль в обеспечении эффективности системы.

Функциональные требования включают:

* универсальность, то есть способность системы адаптироваться к различным стандартам и шаблонам, будь то ГОСТ, внутренний корпоративный регламент или международный стиль;
* комплексный анализ. Система должна проверять широкий спектр параметров, включая шрифты, отступы, выравнивание, порядок нумерации заголовков, оформление таблиц и изображений, подписей, списки;
* автоматические рекомендации. Для повышения удобства, система должна не просто выявлять ошибки, но и предлагать конкретные исправления;
* интеграция с существующими решениями. Возможность взаимодействия с популярными текстовыми редакторами (Microsoft Office, Google Docs, Р7-Офис и др.) повышает практическую ценность системы, особенно в образовательных средах.

Помимо функциональных возможностей, обеспечивающих анализ и исправление оформления, система должна соответствовать нефункциональным требованиям, определяющим ее удобство, надежность и масштабируемость.

* интуитивно понятный интерфейс. Система должна быть доступна пользователям без специальных технических знаний, обеспечивая простоту загрузки документов и просмотра отчетов об ошибках;
* высокая точность. Алгоритмы должны минимизировать ложные срабатывания и обеспечить стабильные результаты, исключая субъективность, характерную для ручной проверки;
* скорость обработки. Анализ документа должен происходить за минимальное время, даже при больших объёмах данных;
* масштабируемость. Возможность обработки документов различного объема и сложности без потери качества является критически важной для применения в условиях интенсивного документооборота.

Значимость перечисленных требований подтверждается анализом временных затрат на нормоконтроль. Как отмечает Д.А. Романов в своем исследовании процессов согласования документов, только исправление ошибок оформления занимает более 30% общего времени, затрачиваемого на проверку и утверждение документа. Ручная проверка одной работы объемом 50-100 страниц может занимать от 20 минут до нескольких часов суммарно, в зависимости от опыта проверяющего и качества оформленной работы. Это делает автоматизацию не только удобным, но и оправданным решением [2].

Разработка универсальных и интеллектуально гибких систем, способных обеспечивать высокую точность, представляет собой перспективное направление, которое может улучшить процессы документооборота в учебной и профессиональной сфере. Внедрение таких решений особенно актуально в условиях цифровизации образования, где скорость и качество обработки документов напрямую влияют на продуктивность работы.

**Вывод**

Проведенный анализ предметной области выявил ряд ключевых закономерностей и проблем, определяющих актуальность разработки автоматизированных решений для контроля оформления текстовых документов.

Традиционный метод контроля, основанный на визуальной проверке специалистами, несмотря на распространенность, демонстрирует существенные ограничения: субъективность оценки, высокая трудоемкость, неустойчивость к увеличению объема документации и отсутствие гарантий полного соответствия требованиям.

Обзор существующих автоматизированных систем выявил, что большинство из них фокусируется на отдельных задачах, не предлагая комплексного решения для полноценного нормоконтроля.

**2 Анализ и проектирование системы**

Анализ и проектирование системы представляют собой ключевой этап разработки автоматизированного решения контроля текстовых документов, направленный на создание эффективной и универсальной архитектуры. Данный раздел определяет функциональные и технические элементы системы, включая ее структуру, выбор инструментов и подходов к реализации.

**2.1 Структура разрабатываемой системы**

Структура разрабатываемой системы представляет собой совокупность взаимосвязанных компонентов, обеспечивающих выполнение функциональных требований. Комплекс взаимодействующих модулей обеспечивает полный цикл анализа и загрузки документа с сводным отчетом. Архитектура системы будет разработана с учетом основных принципов проектирования приложений [3].

Для обеспечения эффективного взаимодействия между компонентами системы была разработана модульная архитектура, представленная на рисунке 1. Она включает взаимосвязанные модули, каждый из которых отвечает за выполнение специализированных функций, таких как загрузка документа, анализ его структуры и формирование отчета об ошибках. Такой подход обеспечивает гибкость системы и возможность ее дальнейшего расширения.

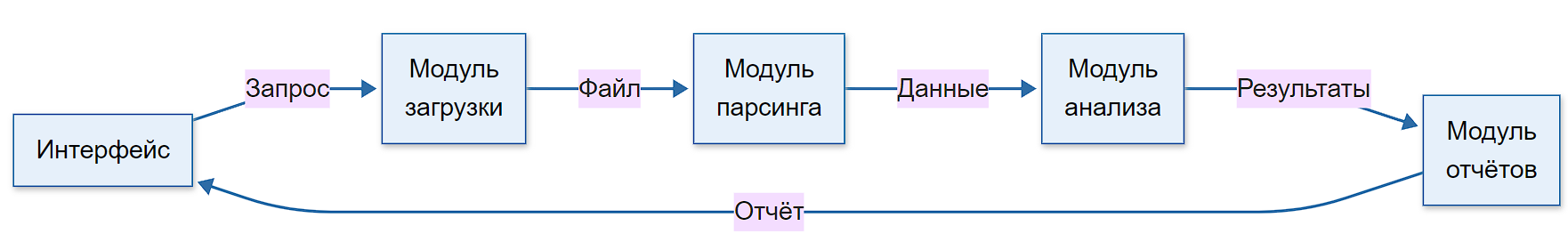


Рисунок 1 – Архитектура будущей системы

Каждый модуль — загрузка и парсинг документов, анализ форматирования, формирование отчетов и пользовательский интерфейс — выполняет строго определенный набор задач с минимальным пересечением функциональности.

Все функции работают отдельно, и каждая отвечает только за одну задачу. Изменение компонентов в данном случае не затрагивает другие. Такой подход соответствует принципу единственности ответственности.

Компоненты разрабатываемой системы должны взаимодействовать только с минимальным набором данных и объектов. Необходимо использовать инкапсулированные классы, для обработки списков, объединяя данные и методы.

Принцип избегания дублирования будет реализован путем сосредоточения функциональности и повторного использования кода.

На начальном этапе система разрабатывается итеративно только с минимальным количеством необходимых компонентов.

Описанные принципы обеспечат разрабатываемую систему в создании качественной и устойчивой архитектуры.

Применение инкрементной модели станет основой для построения структуры. Ее использование ускорит создание системы и минимизирует некоторые риски, связанные с некорректным проектированием. Поэтапная разработка облегчит адаптацию к изменяющимся требованиям [4].

Кроме того, такой подход позволит постепенно тестировать и дорабатывать модуль, снижая вероятность критических ошибок на этапе интеграции.

Процесс итеративной разработки включает регулярное тестирование каждого инкремента. На начальных этапах тестирование проводится на модельных документах с заранее известными ошибками, что позволяет оценить точность алгоритма анализа.

Все этапы модели наглядно продемонстрированы на рисунке 2.

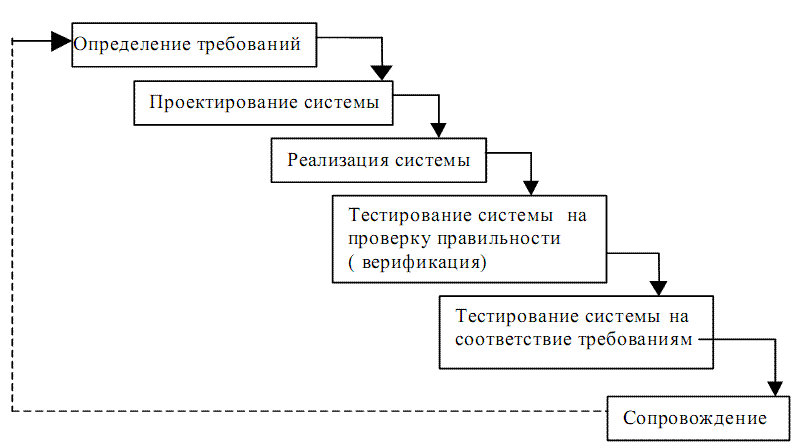


Рисунок 2 ­– Схема инкрементной модели

Такая архитектура, реализованная поэтапно, создаст основу для дальнейшего расширения системы. Она позволит на каждом этапе равномерно распределить задачи, тестировать функционал и своевременно вносить корректировки. Данный подход обеспечивает высокую адаптивность системы к будущим изменениям требований и стабильность работы всего комплекса в целом.

**2.2 Обзор библиотек для работы с текстовыми документами**

Выбор инструментов в рамках разработки автоматизированной системы контроля оформления стал первоочередной задачей. Анализ существующих библиотек, предназначенных для обработки и модификации текстовых документов, необходим для достижения поставленной цели. Было решено провести сравнительный анализ среди более популярных решений, ключевые характеристики которых отображены в таблице 1.

В настоящее время имеется достаточно большой количество современных библиотек, предоставляющих широкий функционал, который имеет свои преимущества и ограничения.

Таблица 1 – Анализ библиотек для работы с текстовыми документами

|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека | Язык | Основные возможности | Ограничения |
| Python-docx | Python | изменение текста, стилей, заголовков, работа с таблицами и изображениями | не поддерживает работу с комментариями |
| docx2python | Python | извлечение текста с сохранением структуры | ограниченные возможности редактирования, нет возможности создания документа |
| docx2txt | Python | извлечение чистого текста из документа | не сохраняет форматирование, не поддерживает редактирование |
| docxcompose | Python | объединение нескольких файлов, сохранение стилей | не поддерживает редактирование |

Библиотека python-docx представляет собой наиболее комплексное решение для работы с DOCX форматом. Ее API позволяет быстро реализовывать функции для проверки шрифтов, таблиц, изображений и других элементов. Также полнота и качество документации позволяет без особых усилий с ней работать. Единственное ограничение — отсутствие встроенной поддержки работы с комментариями, что необходимо для успешного выполнения задач, поставленных в проекте. Для решения данной проблемы было решено использовать надстройку docx-bayoo, которая легко интегрируется в архитектуру python-docx через механизм наследования классов и расширяет функциональность библиотеки, позволяя добавлять комментарии к параграфам, таблицам, изображениям и другим элементам.

Библиотека docx2python предназначена для извлечения содержимого из DOCX файла с сохранением структуры. Она фокусируется на парсинге документа, представляя содержимое в виде вложенных словарей и списков. Однако она не предоставляет доступ к стилям текста или низкоуровневым XML-элементамм, не поддерживает редактирование документов или их создание и имеет ограничения на работу с разными версиями python [5].

Библиотека docx2txt также предназначена для извлечения текста из документов, но она не поддерживает анализ стилей, таблиц, изображений и остальных элементов и редактирование файлов, что категорически необходимо для реализации автоматизированной системы контроля оформления текстовых документов.

Docxcompose специализируется на объединение нескольких файлов с сохранением стилей и форматирования, но не позволяет работать с содержимым документа и редактировать его.

По результатам анализа было выявлено, что библиотека python-docx является самым оптимальным вариантом для реализации функционала автоматизированной системы благодаря своим возможностям, простоте и совместимости с python.

**2.3 Определение функциональных требований и алгоритм работы**

Проектирование функциональности начинается с анализа конечных задач пользователя. Многоуровневый подход требует разбивки системы на специализированные модули, работающие последовательно или параллельно. Он обеспечит правильную программную реализацию и соответствие документа требованиям ГОСТ или внутренним регламентам образовательного учреждения. Функциональные требования к разрабатываемой системе можно структурировать по нескольким направлениям, каждое из которых отвечает за определенную группу проверок и операций. Основной целью автоматизированной системы контроля оформления текстовых документов является минимизация человеческого фактора и обеспечение соответствия документов установленным стандартам.

**2.3.1 Предварительная обработка документа**

Модуль первоначальной обработки выполняет подготовку документа для последующего анализа. Обеспечить безопасность и извлечь данные для передачи на другие этапы — основная задача. Этот модуль является связующим звеном между входными данными и последующей проверкой.

Алгоритм его работы следующий:

* проверка расширения файла. Система должна обрабатывать **широкий спектр форматов документов**. На первом этапе предпочтительно сосредоточиться на DOCX — наиболее распространённом в деловой и образовательной практике. В случае распознавания другого формата документа происходит выход из обработки;
* создание копии файла для предотвращения случайных изменений в исходном документе, что может повлиять на последующие проверки;
* загрузка файла для последующей работы с параграфами и другими элементами.

Полученные данные передаются в следующие модули, обеспечивая основу для анализа и поэтапную обработку.

**2.3.2 Анализ структуры**

Данный модуль отвечает за проверку оформления установленным требованиям ключевых компонентов и идентификацию обязательных структурных элементов, таких как введение, заключение, перечень информационных ресурсов и приложения.

Функциональные требования для контроля приложения предполагают комплексный подход к проверке их оформления. Каждое приложение должно быть четко обозначено с использованием последовательной алфавитной нумерации, начиная с «Приложения А» и возможным заголовком, выполненным шрифтом Times New Roman размером 16 pt с выравниванием по центру страницы, без отступов. Требуется контроль за последовательностью, чтобы исключить пропуски или дублирование букв.

Для листингов программного кода необходима проверка подписей, у которых есть установленный формат в виде «Листинг Х – Название». Они также должны быть набраны шрифтом Times New Roman размером 14 pt, без абзацного отступа. Сам код необходимо оформлять шрифтом Courier New размером 12 pt с одинарным межстрочным интервалом.

Алгоритм работы включает несколько последовательных этапов:

* распознавание структурных элементов, с помощью перебора каждого параграфа документа, используя регулярные выражения и ключевые слова, выявляя заголовки необходимых разделов;
* проверка существующих элементов и выведение ошибки при фиксации их отсутствия;
* обнаружение приложений и проверка на соответствие последовательности используемых букв в названии;
* проверка формата подписи листинга и оформления кода.

Результатом работы модуля является список структурированных ошибок, которые передаются на следующие этапы анализа.

**2.3.3 Списки**

Списки являются одним из важных элементов в документе, а их оформление должно строго соответствовать установленным правилам.

Нумерованные списки должны демонстрировать строгую последовательность номеров, в которых могут быть использованы арабские цифры или буквы русского алфавита, без пропусков или дублирования. В маркированный списках должны использоваться одинаковые знаки табуляции, которые также разрешены стандартами.

Алгоритм работы включает шаги:

* извлечение списков, анализ текстовых префиксов;
* проверка формата нумерации на допустимые префиксы и маркеры;
* анализ последовательности элементов с помощью древовидного подхода;
* проверка наличия и остальных правил пунктуации;
* проверка форматирования, включающая анализ шрифта, отступов, выравнивания и межстрочного интервала для обеспечения единообразия оформления документа.

Список информационных ресурсов дополнительно проверяется на соответствие допустимой структуре библиографических записей. Алгоритм позволяет выявлять ошибки, такие как пропущенные или дублированные знаки препинания, некорректное оформление каждого параметра, наличие обязательных элементов. Этот функционал поможет пользователям повысить качество библиографического оформления и снизит вероятность ошибок при повторной проверке.

Модуль анализирует нумерованные и маркированные списки, включая перечень использованных информационных ресурсов и все заголовки основной части, а также обеспечивает проверку порядка нумерации, форматирования и пунктуации, что особенно важно в соответствии с необходимыми нормативными требованиями.

Данный модуль обеспечит комплексный подход к контролю списков любого уровня вложенности, включая их взаимодействие с заголовками документа.

**2.3.4 Таблицы и изображения**

В процессе анализа документа особое внимание уделяется корректности оформления таблиц и изображений. Особое значение придается правилам переноса таблиц между страницами и соблюдению установленных отступов. Также графический материал в документе должен сопровождаться подписями, соответствующими шаблонам [6].

Проверка происходит по следующим критериям:

* наличие и корректность оформления подписей согласно требованиям;
* соблюдение единого стиля обозначений и нумерации;
* точность расположения относительно остального текста в документе;
* проверка форматирования подписей на необходимые параметры.

Данный модуль не только эффективно обрабатывает сами таблицы и изображения, но и уделяет особое внимание таким сложным случаям, как наличие соответствующих пометок о продолжении и окончании таблиц, если они располагаются на нескольких страницах.

**2.3.5 Форматирование абзацев**

Основной текст, не относящийся к графическим материалам, заголовкам и спискам, также должен проходить детальный анализ на соответствие установленным стандартам. Некорректное форматирование абзацев не только нарушает визуальную целостностью текста, но и смысловое восприятие информации, а также снижает его профессиональную ценность.

При контроле базовых параметров форматирования необходимо проверять такие характеристики, как тип и размер шрифта, межстрочные интервалы, двойные пробелы, величина абзацных отступов и выравнивание. Особое внимание стоит уделить выявлению отклонения от стандартов, например, использование нестандартных шрифтов или некорректных отступов. Алгоритмы должны применяться для каждого параграфа, обеспечивая единообразие.

**2.3.6 Формирование отчета**

При завершении обработки документа все ошибки, которые были выявлены предыдущими модулями, объединяются в список словарей, структурированный по типу ошибки. Результаты сохраняются в JSON-файл для дальнейшей работы.

Для повышения практической ценности разрабатываемого решения, стоит реализовать рекомендательную систему, которая будет не просто выявлять ошибки, но и предлагать способы их исправления.

Таким образом, комплекс функциональных требований к автоматизированной системе контроля оформления текстовых документов представляет собой многоуровневую структуру взаимосвязанных компонентов, обеспечивающих всесторонний анализ выпускных квалификационных и курсовых работ на соответствие установленным стандартам ГОСТ. Каждый модуль решает специфическую задачу, оптимизируя процесс нормоконтроля. Реализация этих требований позволит создать универсальный инструмент, который значительно повысит качество оформления документации в образовательной сфере.

**2.4 Выбор архитектуры**

При проектировании автоматизированной системы контроля оформления текстовых документов рассматривалось два варианта потенциальной архитектуры: автономное приложение и клиент-серверная модель. Каждый из данных подходов характеризуется уникальным набором преимуществ и ограничений.

Автономное приложение представляет собой программное решение и предполагает установку полного функционала на персональный компьютер (ПК) пользователя. Такой подход обладает неоспоримым достоинством — возможность работы без постоянного подключения к сети, что особенно ценно в ситуациях, где доступ к интернету ограничен или отсутствует. Стоит отметить, что такой вариант архитектуры предпочтителен для офлайн-работы и случаев, где безопасность конфиденциальных данных критична. Однако автономная система имеет существенное ограничение. Обновление правил оформления, таких как новые версии стандартов ГОСТ или внутренние регламенты, требует ручной установки обновлений на каждом ПК или другом устройстве, что усложняет поддержку системы. Также, ограничение возможности одновременной работы нескольких пользователей над документами, например, при проверке файлов преподавателем или студентом.

Альтернативный подход представлен клиент-серверной архитектурой. Эта модель предполагает разделение системы на клиентскую часть, которая отвечает за пользовательский интерфейс и взаимодействие, и серверную, где выполняется основная нагрузка, состоящая из анализа документа и его форматирования. Такой вариант позволить в будущем обновлять базу правил с актуальными стандартами ГОСТ и внутренними регламентами образовательных учреждений. Эти модернизации, которые будут внесены на сервер, станут автоматически доступны для всех пользователей, что устраняет необходимость в ручной настройке. Также эта модель поддерживает одновременную работу множества пользователей, что делает ее подходящей для коллективного использования в образовательной среде.

Такие системы, построенные на архитектуре клиент-серверной модели, отличаются высокой гибкостью и масштабируемостью, позволяя адаптировать ресурсы под увеличивающийся объем документооборота, а также обеспечивают интеграцию с другими внешними платформами и сервисами, которые могут быть также необходимы для проверки или работы с документами.

В сравнении с автономным приложением клиент-серверная модель требует стабильное интернет-соединение, что может быть ограничением для некоторых пользователей. Однако проведенный анализ показывает, что многочисленные преимущества данной архитектуры существенно перевешивают это временный недостаток.

На рисунке 3 представлена схема взаимодействия клиентской и серверной частей системы, включающая веб-интерфейс для пользователей, сервер обработки запросов и базу данных для хранения правил и результатов анализа.

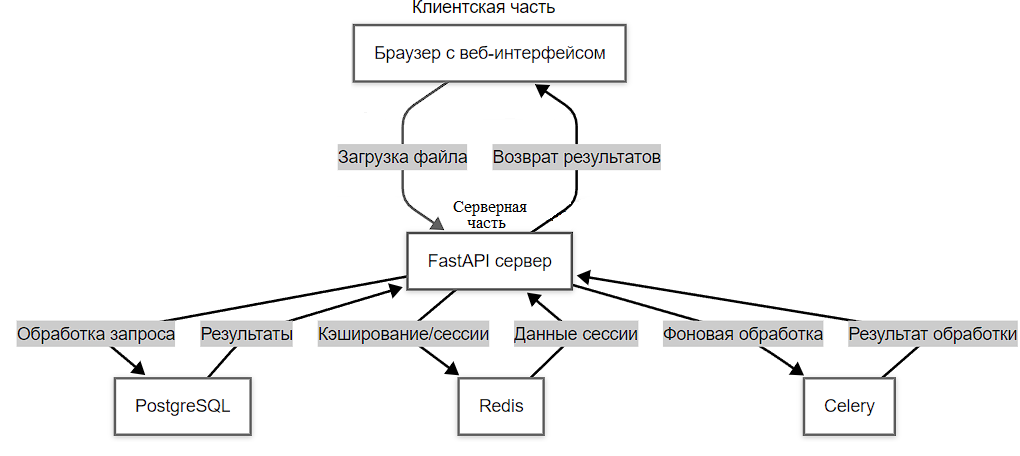


Рисунок 3 – Схема клиент-серверной архитектуры

Проведенный анализ позволил обосновать выбор клиент-серверной модели как наиболее подходящий вариант для архитектуры программного средства, учитывая потребности пользователей и дальнейшее развитие системы.

**2.5 Проектирование пользовательского интерфейса**

Пользовательский интерфейс является критически важным элементом автоматизированной системы контроля оформления документов, поскольку он непосредственно влияет на восприятие системы пользователями и эффективность ее использования. В данном разделе рассматриваются структура интерфейса, принципы его разработки, ключевые компоненты с акцентом на достижение баланса между функциональностью и визуальной частью.

Создание интуитивного пользовательским интерфейсом — один из важнейших аспектов функциональных требований. В данном случае он выступает не просто оболочкой системы, а важнейшим инструментом взаимодействия между системой анализа и пользователем. Интерфейс станет связующим звеном между пользователем и серверной частью, предоставляя доступ к функциональности через веб-платформу. Основной задачей является предоставление простого и понятного пути от загрузки документа до получения результатов анализа для эффективного взаимодействия с системой.

Проектирование интерфейса опирается на несколько современных базовых принципов User Experience (UX) и User Interface (UI) дизайна, направленных на повышение удобства использования, снижения когнитивной нагрузки и обеспечение четкой обратной связи [7].

Интерфейс системы реализован как веб-приложение, что соответствует клиент-серверной архитектуре. Главная страница включает область для загрузки документа с поддержкой механизма drag-and-drop, что позволяет пользователю интуитивно перетаскивать файлы в обозначенную зону. Для визуализации макета главной страницы был разработан прототип, представленный на рисунке 4, который демонстрирует расположение области загрузки и другие элементы навигации.

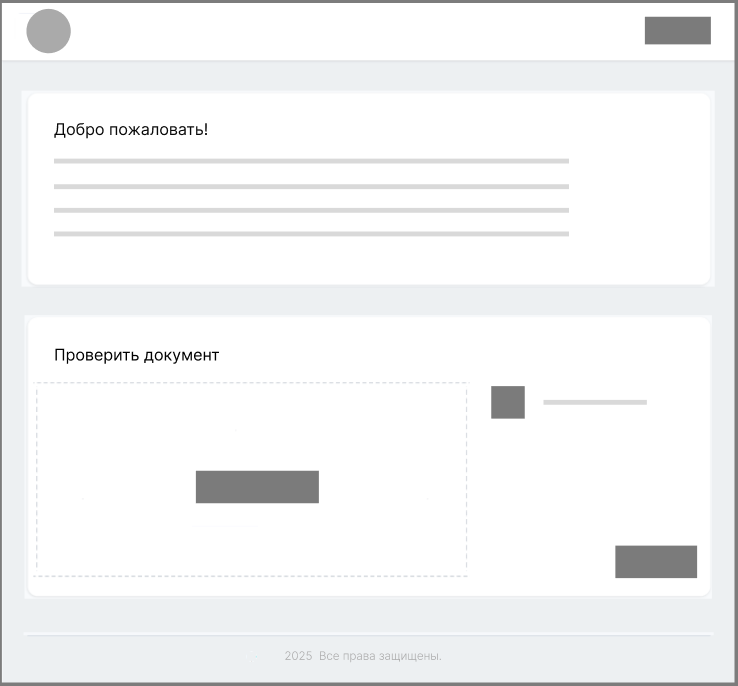


Рисунок 4 – Макет главной страницы

Разработанный макет главной страницы демонстрирует ключевой элемент интерфейса, который организован в соответствии с принципами визуальной иерархии. Центральное расположение области загрузки документов акцентирует внимание пользователя на основной функционал системы.

Использование нейтральной цветовой палитры с акцентными элементами, такими как кнопки контрастного оттенка, повысит интуитивность взаимодействия и снизит когнитивную нагрузку.

Интерфейс системы должен обеспечивать максимально простой и интуитивный путь от загрузки документа до получения детализированного отчёта о несоответствиях.

Архитектура пользовательского интерфейса автоматизированной системы включает два ключевых компонента: панель загрузки и управления документом, а также панель результатов проверки. Панель управления документами представляет собой многофункциональный модуль, обеспечивающий управление рабочим пространством пользователя. Она включает область загрузки документов, которая поддерживает не только перетаскивание, но и выбор через диалоговое окно. Далее список загруженных документов отображается в виде таблицы, где для каждого файла указаны название, статус с визуальным цветовым кодированием, дата загрузки, количество найденных ошибок. Также есть инструменты для управления обработанных файлов, которые предоставляют контекстно-зависимые операции, например, удаление или скачивание проверенного документа.

Для визуализации этих панелей был разработан прототип, представленный на рисунке 5, который демонстрирует дизайн панели загрузки и управления документом и таблицы результатов проверки.

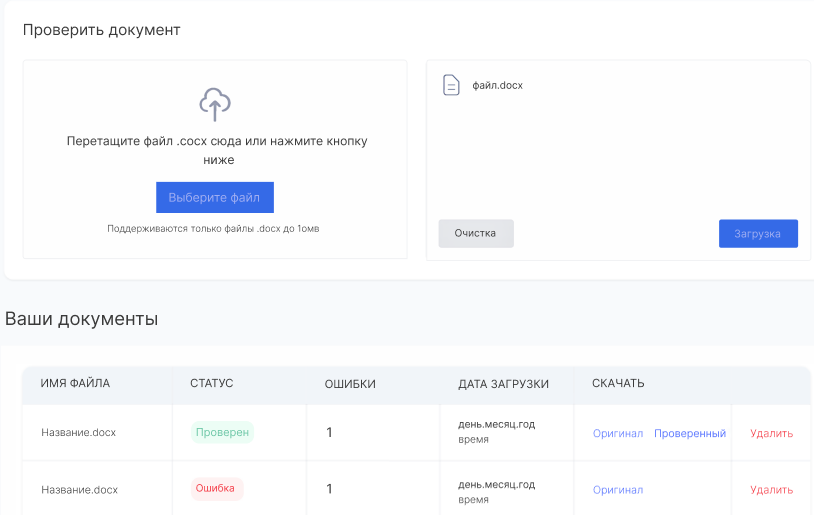


Рисунок 5 – Дизайн панели управления документом и таблицей результатов

Таблица результатов проверки предоставляет собой некую сводку, визуализирующую выявленные несоответствия. Она предоставляет структурированный сводный отчет, отображающий количественные характеристики ошибок по категориям. Это позволяет пользователю моментально оценить общее качество оформления документа. Пример такой таблицы можно посмотреть на рисунке 6.

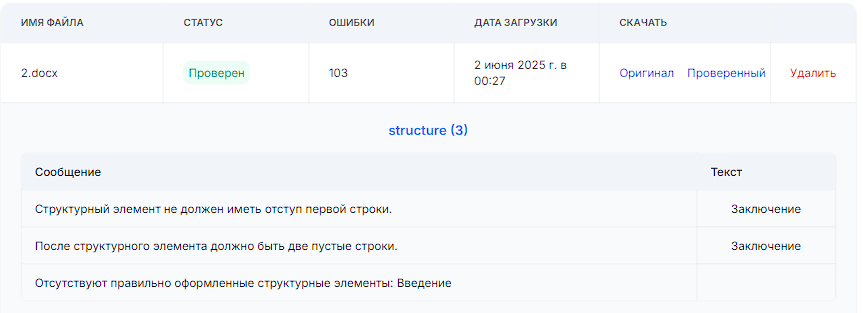


Рисунок 6 – Таблица со структурированным списком найденных ошибок

Детализированный список представлен в виде таблицы, где каждая ошибка сопровождается указанием конкретного параграфа или элемента и текстом самой ошибки с рекомендацией по исправлению.

После обработки документа пользователю предоставляется возможность сохранить отчет в формате DOCX, где несоответствия выделены цветной заливкой и снабжены примечаниями с подробными пояснениями. Это позволяет пользователям быстро понять, какие изменения необходимо внести в документ.

Разработанный пользовательский интерфейс веб-приложения представляет собой эффективное решение, обеспечивающее гармоничное сочетание функциональности и простоты использования. Благодаря соблюдению современных принципов UX/UI-дизайна, таким как четкая визуальная иерархия и минимизация когнитивной нагрузки, получилось создать интуитивно понятный путь от загрузки документа до получения сводного отчета, который требует минимального участия пользователя и предоставляет точную диагностику ошибок.

**Вывод**

В данной главе сформировалась целостная концепция автоматизированной системы контроля оформления текстовых документов, которая соответствует современным тенденциям в области автоматизации документооборота.

Проведенный анализ позволил сформировать прочную основу и комплексное представление о структуре, функциональных возможностях и технологических основах разрабатываемого решения. Итеративный подход к проектированию, подкрепленный инкрементной моделью разработки, предоставил возможность минимизировать риски и заложить потенциал для дальнейшего расширения функциональности.

Сравнение программных библиотек для работы с текстовыми документами выявило оптимальное решение, благодаря широким возможностям анализа и редактирования DOCX-файлов.

**3 Программное конструирование**

Создание автоматизированной системы контроля оформления текстовых документов потребовало комплексного подхода к разработке программного обеспечения, объединяющего множество технологических решений в единый комплекс.

Необходимо определить, какой язык программирования лучшего всего подойдёт для решения поставленных задач, а также уделить внимание используемым библиотекам и другим элементам технологического стека.

**3.1 Язык программирования и среда разработки**

Основным и первоначальным этапом работы над проектом стало определение оптимального стека технологий для программной реализации.

Определение языка программирования был обусловлен необходимостью сочетания высокой производительности, гибкости и поддержки кода. Выбор языка устанавливает не только эффективность кодирования, но и последующие возможности системы по интеграции и масштабированию. Предпочтение было отдано языку Python, благодаря его простоте синтаксиса и обширной экосистемой библиотек, покрывающей большой спектр задач в данном проекте — от обработки текстов в документе, до построения веб-сервиса. Важным критерием также была поддержка асинхронного программирования, которые позволили создать серверную часть. Python позволил сократить время написания кода за счет лаконичных конструкций, обеспечил высокую читаемость программных модулей и упростил процесс отладки.

В качестве интегрированной среды разработки был выбран PyCharm, которая предоставляет мощный инструментарий для работы с Python-проектами. Выбор обусловлен наличием специализированных инструментов для работы с фреймворками и базами данных, а также широкой поддержкой в сообществе, как и Python, что гарантирует доступ к актуальной документации. Ещё одним преимуществом данной среды разработки является статистический анализ кода и его интеллектуальное автодополнение.

Кроме того, для обеспечения воспроизводимости окружения на разных этапах разработки проекта использовалась технология контейнеризации Docker. Создание изолированный контейнеров для различных компонентов исключило влияние внешних факторов на стабильность системы.

**3.2 Используемые инструменты и библиотеки**

Функциональность и разработка автоматизированной системы опирается на тщательно подобранный комплекс взаимосвязанных программных библиотек, каждая из которых решает определенный спектр задач. Выбор технологий был обусловлен необходимостью эффективной обработки текстовых документов. Далее представлен обзор ключевых компонентов, которые были использованы при реализации программного средства.

Python-docx с надстройкой bayoo-docx используется для анализа и модификации документов в формате DOCX. Применяются для извлечения всех необходимых элементов из документа, а также валидации форматирования шрифтов, стилей, отступов и межстрочных интервалов. Надстройка расширяет базовую функциональность и позволяет добавлять примечания в документ.

Passlib — специализированна библиотека, которая используется для безопасного хэширования паролей и проверки их подлинности [8].

Xml — библиотека для анализа и обработки XML-структуры документов формата DOCX.

Re — стандартный модуль Python, предназначенный для работы с регулярными выражениями. Предоставляет инструменты для поиска, замены и разбиения текста по заданным шаблонам.

Центральное место в серверной архитектуре системы занял фреймворк FastAPI — современное решение для создания веб-приложений на Python. В сочетании с uvicorn обеспечивает скорость обработки на 30-50% выше по сравнению с аналогичными решениями на Flask или Django, что особенно важно при работе с объемными файлами. Предоставляет RESTful API для взаимодействия между клиентской частью и сервером, а также поддерживает асинхронную обработку запросов и автоматически генерирует документацию с помощью Swagger. В основном используется для описания эндопинтов.

Redis — NoSQL хранилище, работающее с данными типа «ключ-значение». Используется для кэширования и хранения сессий пользователей.

Celery — асинхронная очередь задач, которая выполняется параллельно. Используется для распределенной обработки сообщений, позволяет выполнять задачи в фоновом режиме.

MailHog — тестовый почтовый сервер для отправки электронных писем. Он не требует дополнительных настроек и позволяет видеть вложения.

Клиентская часть системы представлена в виде веб-интерфейса, разработанного с применением стандартного набора современных технологий — HTML5 и CSS3, которые обеспечивают универсальный доступ без конкретной привязки к конкретной операционной системе. За счёт применений асинхронных функций JavaScript реализовано обновление динамических элементов и взаимодействие с серверной частью без перезагрузки страницы. Передача данных между клиентской и серверной частями осуществляется с использованием файлов формата JSON, что обеспечивает высокую скорость и надежность взаимодействия.

**3.2.1 Организация базы данных**

Важным компонентом является система хранения данных, которая отвечает за структурированное и безопасное управление информацией.

В качестве такой системы была выбрана реляционная система управления базами данных (СУБД) PostgreSQL, обладающая рядом преимуществ. Это одна из наиболее стабильных СУБД с открытым исходным кодом. Она обеспечивает высокую производительность при работе с большим объемом данных, что критически важно для системы, которая ориентирована на интенсивный документооборот. Также PostgreSQL соответствует принципам ACID, гарантируя целостность данных при параллельной обработке множества запросов и сбоях оборудования или программного обеспечения. Еще одним преимуществом выбранной СУБД является поддержка работы с JSONB типами, что позволило хранить результаты анализа документов.

В проекте база данных используется для сохранения и контроля введенных данных пользователей при регистрации, параметры загруженных документов и результаты проверок. Основу структуры, схема которой изображена на рисунке 7, составляют две ключевые таблицы users и documetns, связанные между собой.

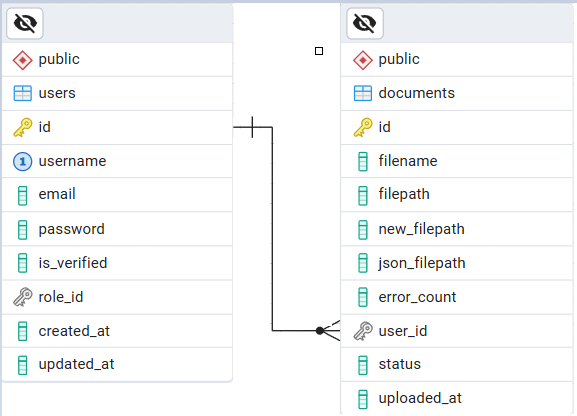


Рисунок 7 – Схема базы данных системы

Таблица users содержит аутентификационные данные, включая зашифрованные пароли, которые обрабатываются с помощью криптографического алгоритма bcrypt. Он обеспечивает безопасность и повышает устойчивость к подбору за счет добавления сгенерированной уникальной «соли» к введенному пользователем паролю перед хэшированием, предотвращая таким образом атаки с использованием радужных таблиц и исключая возможность восстановления записи даже при полном доступе к базе данных.

Таблица documents предназначена для хранения информации о загруженных документах, например, дату загрузки, статус анализа, количество ошибок, и результаты их проверки в формате JSONB. Работа с файлами JSON позволила сохранять итоги анализа в виде иерархической структуры, где каждый уровень соответствует определенному типу ошибок, после чего предоставила возможность для формирования сводных отчетов.

Связь между таблицами позволяет отслеживать принадлежность документов к каждому пользователю, что особенно важно при работе с множеством файлов. Для оптимизации производительности используется кэширование часто запрашиваемой информации в Redis, например, активные сессии пользователей и результаты анализа, что сокращает время обработки запросов и снижает нагрузку на основную базу данных.

Для управления подключениями и выполнения сложных запросов без блокировки основного потока задач используется асинхронная версия библиотеки SQLAlchemy. Взаимодействует с СУБД через python-объекты, именуемые модулями.

Выбор PostgreSQL в качестве основы системы управления базами данных для автоматизированной системы контроля оформления текстовых документов обеспечил надежную архитектуру с высокой производительностью, гибкостью и надежностью в сочетании с Redis и SQLAlchemy.

**3.3 Программная реализация**

Разработка автоматизированной системы потребовала создания программного комплекса, обеспечивающий анализ структуры и форматирования текстов в соответствии с заданными стандартами. В основе логики лежит пошаговый разбор документа. В данном разделе представлен детальный анализ созданных программных компонентов, образующих целостную систему автоматизированного нормоконтроля.

Вспомогательная функция set\_red\_background(run) добавляет красную заливку к фрагментам текста документа. С помощью этого получается визуально выделить нарушения в оформлении исходного файла.

Функция check\_structural\_elements(doc) выполняет следующие шаги:

* идентификация структурных элементов. Определяется наличие обязательных структурных элементов документа и корректность оформления;
* обработка приложений. Анализируется наличие и корректность нумерации приложений, используя буквы русского алфавита, а также проверяется соответствие их оформления требованиям;
* обнаружение листингов. С помощью регулярного выражения проверяется формат подписи листингов. Далее функция выделяет абзацы, где содержится программный код, проверяет их на форматирование, и исключает их из обработки как элементы, не подлежащие стандартной проверке.

Модуль проверки списков выполняет анализ нумерованных и маркированных списков, а также перечень информационных ресурсов. Он включает в себя следующие функции:

* extract\_list\_items(doc) парсит документ, выделяя списки и разделяя их на обычные списки и библиографические списки. Определяет формат нумерации и проверяет его соответствие стандартам;
* validate\_prefix\_format(doc, prefix, fmt, errors, full\_text, paragraph) аналзирует корректность префиксов списков, выявляет ошибки, включая недопустимую вложенность или некорректные символы;
* validate\_resource\_list(doc, resource\_groups, excluded\_ paragraphs) контролирует оформление перечня информационных ресурсов, проверяет его на соответствию формату, включая правильность пунктуации, шрифта и отступов.

Функция проверки заголовков check\_headings\_formatting(doc, structural\_paragraph, appendix\_paragraphs, listing\_paragraphs, table\_captions, image\_captions) отвечает за анализ иерархии заголовков и их оформления. Она обеспечивает соответствие заголовков заданным стандартам, включая нумерацию, шрифт, размер и выравнивание. Алгоритм работы функции следующий:

* проверка нумерации заголовков. Анализируется нумерация, исключая ошибки, такие как пропуск номер или несоответствие уровней вложенности;
* контроль форматирования. Проверяется использования шрифта, размера 16 pt для заголовков первого уровня и 14 pt для остальных, а также отсутствие курсива, подчеркивания или других отклонений от стандарта.

Функция check\_general\_formatting(doc, caprion\_ paragraphs, excluded\_ paragraphs, code\_ paragraphs) реализована для проверки форматирования общего текста документа, исключает абзацы, прошедшие проверку ранее. Для этого был создан механизм поэтапного сканирования документа. Основные этапы:

* проверка шрифта и размера. Функция контролирует использования шрифта Times New Roman и размера 14 pt для всех текстовых абзацев;
* контроль отступов и выравнивания. Проверяется наличие отступа первой строки, отсутствие отступов справа и слева, а также выравнивание по ширине;
* анализ межстрочного интервала. Функция обеспечивает соблюдение межстрочного интервала 1,5 см для всех абзацев;
* выявление дублированных пробелов. Происходит обнаружение двойных пробелов.

Важным моментом реализации данной функции является механизм эффективной идентификации параграфов с нестандартным форматированием. Алгоритм позволяет выявлять отклонения от заданных правил даже при наличии смешанного форматирования внутри абзаца, что достигается путем анализа на уровне отдельных фрагментов текста runs.

Далее модуль проверки таблиц и изображений, который отвечает за анализ подписей и расположения этих элементов. Функции, содержащиеся в модуле:

* check\_table\_formatting(doc) проверяет таблицы и формат подписей к ним, контролирует шрифт и его размер, отсутствие пустой строки после;
* check\_image\_formatting(doc) находит изображения и анализирует корректность подписей к ним. Проверка аналогична таблицам, отлична тем, что подпись выравнивается по центру, и должна находиться строго под изображением.

Обе функции отслеживают корректность нумерации в контексте разделов или приложений, выявляя нарушения последовательности.

Модуль интеграции и обработки ошибок обеспечивает целостность анализа и удобство пользователя. Содержит несколько функций:

* add\_comments\_to\_document(doc, errors, output\_path, author, initials) добавляет комментарии. Интегрирует выявленные ошибки в документ, добавляя комментарии к соответствующим абзацам, таблицам или изображениям. Это позволяет пользователю быстро обнаружить проблемные места;
* check\_document\_formatting(file\_path, new\_file\_path, json\_file\_path) сохраняет результаты проверки в JSON-файл, содержащий перечень ошибок с указанием их типа, текста абзаца, индекса и сообщения. Новый документ сохраняется отдельно, не изменяя исходный файл;
* remove\_duplicate\_errors(errors) исключает повторяющиеся ошибки, обеспечивая компактность сводного отчета.

Этот набор функций формирует основу программной логики автоматизированной системы контроля оформления текстовых документов. Разработанный алгоритм демонстрирует высокую производительность даже при обработке объемных документов. Каждая функция решает четко поставленную задачу и обеспечивает высокую читаемость кода. Они повышают удобство использования системы, обеспечивая пользователю четкую информацию о выявленных несоответствиях и способах их устранения.

**3.3.1 Работа системы**

Для оценки функциональности был подготовлен текстовый документ — курсовая работа объемом 30 страниц в формате DOCX. Файл содержал намеренно введенные ошибки, характерные для учебных работ. Пример работы системы можно рассмотреть на рисунке 8, где продемонстрирован размеченный документ, после обработки системой.

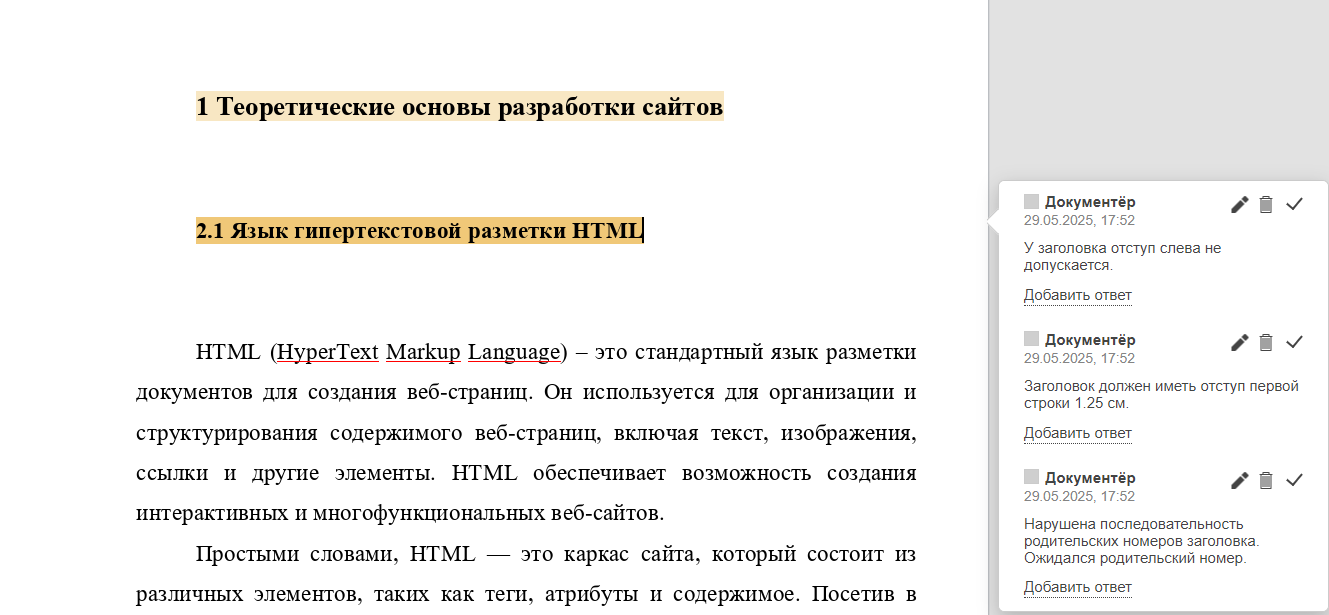


Рисунок 8 – Демонстрация добавленных комментариев об ошибках

Практическая проверка системы на тестовом документе подтвердила ее способность эффективно выявлять ошибки оформления, включая сложные случаи, такие как несоответствие нумерации заголовков или форматирование графических элементов, что подчеркивает ее применимость для нормоконтроля выпускных квалификационных и курсовых работ, обеспечивая точность, оперативность и единообразие результатов.

Также, на этапе практической оценки функциональности система была протестирована на реальных созданных текстовых документах различного объема и стилистической сложности. Для оценки эффективности системы были проанализированы ключевые показатели, включающие общее количество выявленных ошибок оформления, время обработки документа и объем проверяемого материала. Результаты тестирования представлены в таблице 2. Полученные данные позволяют оценить производительность системы и ее способности справляться с задачами нормоконтроля в условиях реального документооборота.

Таблица 2 – Характеристики работы системы на тестовых документах

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Количество страниц | Количество выявленных ошибок | Время обработки, сек |
| 1 | 35 | 287 | 8,1 |
| 2 | 47 | 375 | 9 |
| 3 | 30 | 229 | 7 |
| 4 | 20 | 102 | 3,5 |
| 5 | 66 | 266 | 2,8 |
| 6 | 45 | 98 | 4,2 |

Основной целью экспериментов являлась проверка способности системы выявлять широкий спектр нарушений в форматировании документов. Проведенные тесты подтвердили, что система успешно справляется с обработкой документов объемом до 100 страниц. Среднее время анализа файла составляет 5-7 секунд, что в 15-20 раз быстрее ручной проверки. Данные из таблицы свидетельствуют о высокой производительности системы при обработке различного объема и сложности и подтверждают возможность практического применения разработанной системы в условиях реальной эксплуатации.

Таким образом, полученные результаты подтвердили эффективность разработанной системы в решении задач автоматизированного контроля оформления документов.

**Вывод**

Разработка программного средства была основана на тщательном выборе архитектурных решений, инструментов и библиотек. Для реализации был выбран язык программирования Python и среда разработки PyCharm. Ключевым компонентом стала логика проверки документов, реализованная в виде набора функций, каждая из которых отвечает за проверку определенного аспекта.

Проведенные тесты продемонстрировали высокую степень автоматизации проверки оформления текстовых документов, что позволяет ее использовать как средство поддержки нормоконтроля в образовательных учреждениях.

**4 Безопасность и экологичность проекта**

Безопасность жизнедеятельности (БЖД) — область знаний, которая направлена на изучение и обеспечение безопасного взаимодействия человека с окружающей его средой.

В современном мире, вопросы безопасности приобретают большую актуальность, так как технологии очень стремительно развиваются и их воздействие на природу увеличивается. Чтобы предотвратить рост числа различных аварий и чрезвычайных ситуаций на производстве, необходимо обязать специалистов в неотъемлемом изучении и подготовке правил БЖД. Знания и навыки, касающиеся вопросов безопасности, помогут вовремя идентифицировать потенциальные опасности и осуществить комплекс мер по их предотвращению.

**4.1 Анализ вредных и опасных факторов на рабочем месте**

Рабочая среда связана с рядом факторов, способных негативно повлиять на здоровье и работоспособность специалиста. Ключевые из них можно разделить на несколько групп согласно ГОСТу 12.0.003-74 «ССБТ. Опасные и вредные производственные факторы».

Разработка автоматизированной системы контроля оформления текстовых документов предполагает длительную работу за компьютером, в связи с этим, рассмотрим потенциальные психофизические угрозы.

Неподвижное положение тела, перенапряжение мышц шеи и спины, повторяющиеся движения рук, фиксированное положение глаз — все это относится к физическим перегрузкам, которые могут вызвать не только дискомфорт, боли и проблемы с опорно-двигательным аппаратом у человека, но и повлечь за собой последствия.

Для предотвращения и минимизации таких рисков рекомендуется применять комплекс мер:

* регулярные перерывы каждые 45-60 минут работы для выполнения физических упражнений и снятия напряжения с глаз;
* обеспечение комфортного рабочего места для нормального функционирования опорно-двигательной системы и кровообращения;
* установка монитора на уровне глаз, использование специальных компьютерных очков.

Соблюдение указанных правил поможет снизить негативное влияние и физические перегрузки на организм разработчика.

Также, важно соблюдение и обеспечением оптимального микроклимата в рабочем помещении. В современном мире человек проводит большую часть жизни в закрытых пространствах, поэтому соблюдение сбалансированных показателей температуры и влажности воздуха является критически важным фактором для поддержания работоспособности и предотвращению различных заболеваний.

Нарушение температурного баланса может влиять на функциональные расстройства со стороны нервной и сердечно-сосудистой системы, а также других органов.

Пониженная влажность вызывает у человека обезвоживание, пересыхание слизистых оболочек и кожных покровов, что существенно снижает барьерные функции и повышает риск к заболеваниям различными инфекциями.

Повышенная влажность в теплую погоду усиливает теплоотдачу, что повышает вероятность перегрева, появления сердечно-сосудистых нарушений и головной боли, которая снижает работоспособность и увеличивает шанс получения теплового удара.

Для предотвращения всех возможных рисков для здоровья человека, следует соблюдать нормативы, установленные СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [9].

Следует отметить, что требования к параметрам микроклимата могут меняться, в зависимости от функционального назначения помещения, характера трудовой деятельности, регионально климатических особенностей и времени года.

Для обеспечения оптимальных параметров микроклимата и минимизации неблагоприятных воздействий на организм работница, следует соблюдать следующие меры:

1. проводить регламентированные перерывы для обогрева или охлаждения помещений;
2. обеспечивать средствами индивидуальной защиты;
3. соблюдать санитарно-эпидемиологические нормы;
4. проводить плановые медосмотры и профилактику инфекций.

Также, проблемы, связанные с электрооборудованием, относят к одним из самых опасных факторов. Основные опасности при работе с электричеством:

* пожар или взрыв,
* короткое замыкание,
* перегрузка электрической сети,
* повышение напряжения,
* удар током.

Нарушение правил эксплуатации оборудования или превышение установленных стандартов для параметров тока могут привести к серьезным типам повреждений. Они происходит из-за разных причин, но чаще всего это происходит из-за увеличения напряжения в электрической сети.

Отсутствие заземления в электрических системах значительно повышает риск поражения током. При повреждении корпуса или возникновении утечки тока, незаземленный прибор может оказаться под опасным напряжением. Это, в свою очередь, может привести человека к серьезным травмам [10].

Также, отсутствие заземления увеличивает вероятность возникновения пожара. Электрический ток может вызвать перегрев оборудования или даже возгорание в помещении при коротком замыкании или перегрузке сети.

Такие риски предоставляют жизни человека угрозу получения травм или даже к летальному исходу. Чтобы избежать пожара в помещении, оно всегда должно быть оснащено исправными средствами пожаротушения и пожарной сигнализацией.

Еще одной причиной потенциальных угроз является излишняя влажность воздуха в помещении. Повышенная влажность способствует появлению плесени на оборудовании и ускоренному износу изоляции проводов. Также высокая влажность вызывает ржавчину, что снижает производительность и срок службы электрооборудования.

Высокая температура тоже может влиять на работу устройств. Она может приводить к перегреву и отключению оборудования [11].

Для защиты специалиста в таких ситуациях необходимо:

1. обеспечить заземление электропроводки,
2. обеспечить помещение системой пожаротушения,
3. качественно изолировать электропроводку,
4. использовать особое устройство, которое контролирует напряжение,
5. использовать системы контроля температуры и влажности помещения.

Для всех перечисленных опасных и вредных факторов рабочей среды были проведены замеры, которые можно рассмотреть в таблице 3. Полученные результаты позволяют выявить существующие нарушения нормативных требований, а также определить меры для их устранения и обеспечения безопасных условиях труда. Реализация предложенных мер способствует снижение рисков для здоровья специалистов и повышению надежности эксплуатации оборудования.

Таблица 3 – Результаты замеров факторов рабочей среды

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Фактор | Допустимые значения | Полученные значения | Превышение | Рекомендации |
| Температура | 23-25°C | 25°C | Нет | Улучшение  системы кондиционирования |
| Влажность | 40-60% | 42% | Нет | Установка регулятора влажности |
| Шум | ≤ 50 дБ | 59 дБ | Да | Обеспечение звукоизоляции |
| Напряжение | 220-230 В | 220 В | Нет | Обновление проводки, дополнительная изоляция |

Таблица демонстрирует необходимость внесения изменений параметров рабочей среды. Для минимизации и устранения всех потенциальных опасностей по отношению разработчику и окружающих его людей, необходимо всегда соблюдать установленные требования.

**4.2 Расчет параметров шума, как негативного производственного фактора**

Следующим производственным фактором является шум. Он представляет собой нежелательные звуковые колебания, которые воспринимаются человеком как акустические раздражители. Появиться такое «загрязнение» может из-за абсолютно любых объектов, которые совершают периодические движения в пространстве.

Источником шума в помещениях ДГТУ может стать компьютерное оборудование, кондиционеры, деятельность в соседних аудиториях или уличный трафик. Шум может относиться к нескольким категориям:

1. постоянный. Уровень звука изменяется за день не более чем на 5 дБ;
2. непостоянный. Уровень звука изменяется более чем нам 5 дБ;
3. импульсный. Звуковые сигналы с длительностью менее 1 секунды.

Согласно установленным нормативам СанПиН 1.2.3685-21, допустимый уровень шума в рабочем помещении не должен превышать 50 дБ.

Результаты, полученные после проведения замеров в нескольких аудиториях при работающем оборудовании можно рассмотреть в таблице 4.

Таблица 4 – Параметры источников шума в учебных помещениях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Источник шума | Уровень шума, дБ | Расстояние до источника, м | Характеристика | Превышение нормы |
| Системные блоки ПК | 40 | 1-2 | Постоянный | Нет |
| Фоновый шум из коридора | 45 | 3-5 | Непостоянный | Нет |
| Кондиционер | 52 | 2-3 | Постоянный | Да |
| Уличный трафик | 58 | 15-20 | Непостоянный | Да |

Для расчёта уровня шума и получения оценки необходимо провести следующие вычисления согласно принципу энергетического суммирования по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

где *Li* – уровень шума от каждого источника в дБ.

|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |

Рассчитанное значение уровня шума превышает норму на 9 дБ, что указывает на необходимость мер по снижению шума в рабочих помещениях ДГТУ.

Постоянный и длительный производственный шум негативно влияет на человека, вызывая побочные действия, которые затрагивают различные функции организма. Головная боль, нарушение сна, снижение концентрации и ухудшение слуха являются лишь малой частью возможных последствий.

Чтобы минимизировать воздействие шума на рабочих местах, необходимо обеспечивать помещения звукоизоляцией, использовать современное техническое оборудование с наименьшим уровнем звука и проводить регламентированные перерывы.

**4.3 Экологическая безопасность рабочей среды**

Разработка программного средства тесно связана с воздействием на окружающую среду. Во время работы формируется несколько видов экологических проблем, для их решения необходимо предпринимать комплексные меры.

Качество воды в районе университета соответствует нормативам, но жесткость и наличие загрязнений требует дополнительной фильтрации и контроля, так как это может послужить поводом образования проблем со здоровьем сотрудников и ускорить процесс износа оборудования. Установка современных систем очистки позволит обеспечить сотрудникам и студентам высококачественную питьевую воду, а оптимизация водопотребления снизить нагрузку на местную систему водоснабжения.

Утилизация отходов, образующихся во время разработки программного средства, представляет собой особую экологическую проблему и требует соблюдать не мало условий. Устаревшая или вышедшая из строя техника должна проходить специализированную обработку. Для реализации правильной утилизации картриджей, батарей, аккумуляторов и прочих технических устройств согласно Федеральному закону от 24.06.1998 №89-ФЗ «Об отходах производства и потребления», рекомендуется создать систему сбора отходов, которая будет проводить ее в несколько необходимых этапов.

Также стоит отметить, что введение электронного документооборота поможет снизить воздействие на экологическую среду за счет снижения использования бумаги, чернил и других расходных материалов. Хранение данных в цифровых хранилищах позволят сократить использование бумаги на 55-60%.

Предложенные меры обеспечивают комплексное решение экологических проблем в районе ДГТУ, снижая воздействие разрабатываемого программного средства на окружающую среду.

**4.4 Вопросы безопасности**

Обеспечение пожарной безопасности при разработке автоматизированной системы контроля оформления текстовых документов является важным аспектом, соответствующим нормативно-правовым требованиям. Основу регулирования составляют Федеральный закон от 21.12.1994 № 69-ФЗ "О пожарной безопасности", Федеральный закон от 22.07.2008 № 123-ФЗ "Технический регламент о требованиях пожарной безопасности" и Правила противопожарного режима в Российской Федерации (утв. Постановлением Правительства РФ от 25.04.2012 № 390). Эти документы устанавливают стандарты эксплуатации электрооборудования и организации помещений в образовательных учреждениях [12,13,14].

Серверы и рабочие станции системы размещаются с учетом требований к вентиляции и расстоянию между устройствами для предотвращения перегрева. Электропроводка проверяется на соответствие нормам, исключающим риск короткого замыкания. В помещениях предусматриваются огнетушители и системы оповещения. Регулярные инструктажи персонала по мерам пожарной безопасности обязательны. Дополнительно применяются отраслевые стандарты, адаптированные к типу учреждения. Риски возгорания минимизируются за счет использования сертифицированного оборудования. Таким образом, система разрабатывается с учетом строгих требований пожарной безопасности.

**Вывод**

Разработка автоматизированной системы контроля оформления текстовых документов требует комплексного подхода к обеспечению безопасности и экологичности, учитывающего как защиту здоровья сотрудников, так и минимизацию воздействия на окружающую среду. Анализ вредных и опасных факторов на рабочем месте выявил необходимость соблюдения нормативов по микроклимату, шуму и электрооборудованию, включая внедрение мер по звукоизоляции, заземлению и контролю влажности. Пожарная безопасность обеспечивается строгим соответствием требованиям Федеральных законов и Правил противопожарного режима, что минимизирует риски возгорания за счет сертифицированного оборудования и систем пожаротушения. Экологическая безопасность достигается через оптимизацию водопотребления, правильную утилизацию отходов и переход к электронному документообороту, сокращающему использование бумаги на 55–60%. Реализация предложенных мер позволяет создать безопасную и экологически ответственную рабочую среду, соответствующую нормативным требованиям и способствующую устойчивому функционированию системы в образовательных учреждениях.

**Заключение**

В ходе исследования была последовательно проанализирована предметная область, спроектирована архитектура программного решения и выполнена реализация.

Разработка автоматизированной системы контроля оформления текстовых документов направлена на решение актуальной проблемы обеспечения единообразия и соответствия документации установленным стандартам. Анализ текущей ситуации в сфере нормоконтроля показал наличие системных проблем, связанных с трудоемкостью ручной проверки, низкой точностью, субъективности и ограниченностью существующих решений. Это подтвердило необходимость создания специализированного программного решения, способного повысить точность обработки документов и сократить временные затраты.

Проектирование системы осуществлялось с учетом всех ключевых функциональных и нефункциональных требований, что позволило сформировать модульную архитектуру. Для каждой группы требований были разработаны специализированные модули, которые действуют независимо, но взаимодействуют в рамках общей логики. Итеративный подход к разработке, реализованный с помощью инкрементной модели, способствовал поэтапному внедрению функциональности, минимизации рисков и своевременной корректировке проектных решений.

Алгоритмы, реализованный в рамках проекта обладают способностью к точному обнаружению нарушений, в том числе в сложных случаях. Разработанная система демонстрирует высокую эффективность автоматизации контроля оформления текстовых документов. Она обеспечивает выявление даже незначительных отклонений от стандартов, которые сложно заметить при ручной проверке.

В текущей версии программного средства реализована работа с форматом файлов .docx. Расширение поддержки других форматов позволит применять систему во многих других сферах и организациях, где точность оформления также важна. Перспективы дальнейшего развития включают и внедрение методов машинного обучения для интеллектуального анализа документов. Применение некоторых алгоритмов позволит автоматических распознавать новые типы ошибок без явного программирования правил проверки и адаптироваться к обновленным требованиям.

Внедрение такой системы в профессиональной и образовательной сферах позволит обеспечить повышение качества оформления документов, сократить затрачиваемое время и снизить нагрузку на специалиста, осуществляющего нормоконтроль.

В результате проделанной работы было разработано комплексное решение, позволяющее автоматизировать процесс контроля оформления текстовых документов.

**Перечень использованных информационных ресурсов**

1. Стариченко, Б. Е. Программа автоматизации контроля оформления текстовых документов / Б. Е. Стариченко, М. А. Устинов. – Текст : непосредственный // Педагогическое образование в России. – 2018. – №8. – С. 163-168.
2. Романов Д. А. Системы аналитической обработки текстовой информации / Д.А. Романов. // SL SOFT : сайт. – URL: https://slsoft.ru/products/preferentum/ (дата обращения 3.05.2025). – Текст: электронный.
3. Руководство по архитектуре приложений Microsoft / Microsoft .– Изд. 2-е, 2009. – 529 с. – ISBN: 978-0-7356-2710-9.
4. Инкрементная модель жизненного цикла разработки программного обеспечения. // NLT : сайт. – URL: https://newline.tech/inkrementnaya-model-zhiznennogo-cikla-razrabotki-programmnogo-obespecheniya-ru/ (дата обращения: 04.05.2025). – Текст: электронный.
5. docx2python 3.5.0 // PyPi : сайт. – URL: https://pypi.org/project/docx2python/ (дата обращения: 04.05.2025). – Текст: электронный.
6. ГОСТ Р 2.105-2019. Единая система конструкторской документации. Общие требования к текстовым документам : национальный стандарт Российской Федерации : издание официальное: утвержден и введен в действие Приказом Федерального агенства по техническому регулированию и метрологии от 8 апреля 2019 г. № 127-ст. – Введ. 2020-01-01. – Москва. : Стандартинформ, 2019. – Текст : непосредственный.
7. Агеева А.Д. Петросян Л.Э. Дизайн интерфейсов (UI) и пользовательских опыт (UX) / А.Д. Агеева, Л.Э. Петросян. – Текст: непосредственный // Вестник науки и образования. – 2024. – № 6 (75). – Т. 1. – С. 1354-1366.
8. Казарян К.К. Безопасность базы данных / К.К. Казарян. – Ростов-на дону : StudNet. – 2022. – №1. – С. 497-508.
9. СанПиН 1.2.3685-21. Гигиенические нормативны физического воздействия на человека: шум, вибрация, электромагнитные поля. – Введ. 01.03.2021. – Москва : Федеральная служба по надзору в сфере защиты прав потребителей и благополучия человека, 2021. – 25 с. – Текст: непосредственный.
10. Электробезопасность: 6 основных факторов. // HENGKO : сайт. – URL: https://schneider-rozetki.ru/ (дата обращения 07.09.2025). – Текст : электронный.
11. Влияние температуры и влажности на работу электрооборудования. // Schneider Electric : сайт. – 2022. – URL: https://www.hengko.com/ru/news/the-effect-of-temperature-and-humidity-on-electrical-equipment/ (дата обращения 08.05.2025). – Текст : электронный.
12. О пожарной безопасности: Федеральный закон от 21.12.1994 № 69-ФЗ : [в ред. от 29.12.2022 г.] // Собрание законодательства Российской Федерации. – 1994. – № 35. – 3649 с. – Текст : непосредственный.
13. Технический регламент о требованиях пожарной безопасности : Федеральный закон от 22.07.2008 № 123-ФЗ [в ред. от 14.07.2022] // Собрание законодательства Российской Федерации. – 2008. – № 30 (ч. 1). – 3579 с. – Текст : непосредственный.
14. Правила противопожарного режима в Российской Федерации : утв. Постановлением Правительства РФ от 25.04.2012 г. № 390 : [в ред. от 28.12.2023 г.] // Собрание законодательства Российской Федерации. – 2012. – № 19. – 2415 с. – Текст : непосредственный.