**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Научный руководитель, преподаватель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Терлыч Н. А. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. | УТВЕРЖДАЮ  Академический руководитель  ОП «Программная инженерия», старший преподаватель департамента программной инженерии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Павлочев Н. А. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

|  |  |
| --- | --- |
| *Подп. и дата* |  |
| *Инв. № дубл.* |  |
| *Взам. инв. №* |  |
| *Подп. и дата* |  |
| *Инв. № подл* |  |

**Приложение для проверки оформления документации**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.12-01 81 01-1-ЛУ**

**Исполнитель:**  
студент группы БПИ222  
\_\_\_\_\_\_\_\_\_\_ / Проскурин Д. А. /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

УТВЕРЖДЕН

RU.17701729.05.12-01 81 01-1-ЛУ

|  |  |
| --- | --- |
| *Подп. и дата* |  |
| *Инв. № дубл.* |  |
| *Взам. инв. №* |  |
| *Подп. и дата* |  |
| *Инв. № подл* |  |

**Приложение для проверки оформления документации**

**Пояснительная записка**

**RU.17701729.05.12-01 81 01-1**

**Листов 21**

**Москва 2024**

Содержание

[ГЛОССАРИЙ 3](#_Toc162707608)

[1. ВВЕДЕНИЕ 4](#_Toc162707609)

[1.1. Наименование программы 4](#_Toc162707610)

[1.2. Документы, на основании которых ведется разработка 4](#_Toc162707611)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_Toc162707612)

[2.1. Назначение программы 5](#_Toc162707613)

[**2.1.1. Функциональное назначение 5**](#_Toc162707614)

[**2.1.2. Эксплуатационное назначение 5**](#_Toc162707615)

[2.2. Краткая характеристика области применения 5](#_Toc162707616)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc162707617)

[3.1. Постановка задачи на разработку программы 6](#_Toc162707618)

[3.2. Описание архитектуры 7](#_Toc162707619)

[**3.2.1. Архитектура python-пакета 7**](#_Toc162707620)

[**3.2.2. Архитектура веб-приложения 8**](#_Toc162707621)

[3.3. Описание алгоритма и функционирования программы 9](#_Toc162707622)

[**3.3.1. Алгоритм работы python-пакета 9**](#_Toc162707623)

[**3.3.2. Алгоритм взаимодействия клиента с сервером 11**](#_Toc162707624)

[3.4. Организация входных и выходных данных 12](#_Toc162707625)

[**3.4.1. Организация входных данных 12**](#_Toc162707626)

[**3.4.2. Организация выходных данных 13**](#_Toc162707627)

[**3.4.3. Хранение данных пользователя на стороне сервера 13**](#_Toc162707628)

[3.5. Описание и обоснование выбора технических и программных средств 14](#_Toc162707629)

[**3.5.1. Описание и обоснование выбора технических и программных средств python-пакета 14**](#_Toc162707630)

[**3.5.2. Описание и обоснование выбора серверного оборудования и программных средств веб-приложения 14**](#_Toc162707631)

[**3.5.3. Описание и обоснование технических и программных средств, необходимых для доступа к веб-приложению со стороны клиента 15**](#_Toc162707632)

[4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 16](#_Toc162707633)

[4.1. Предполагаемая потребность 16](#_Toc162707634)

[4.2. Ориентировочная экономическая эффективность 16](#_Toc162707635)

[4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами. 16](#_Toc162707636)

[ПРИЛОЖЕНИЕ 1 18](#_Toc162707637)

[ПРИЛОЖЕНИЕ 2 19](#_Toc162707638)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 20](#_Toc162707639)

# ГЛОССАРИЙ

* + - 1. docx – Microsoft Word Open XML Document. Формат файлов для хранения электронных документов пакетов офисных приложений. Формат представляет собой архив, содержащий текст, графику и другие данные.

1. Консольное приложение – это тип программы, предназначенный для работы через командную строку (консоль).
2. Бэкенд – это внутренняя часть продукта, которая находится на сервере и скрыта от пользователей.
3. Фронтенд – презентационная часть  [программной](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) системы, её [пользовательский интерфейс](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) и связанные с ним компоненты; применяется в соотношении с базисной частью системы, её внутренней реализацией, называемой в этом случае бэкендом.

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы – «Приложение для проверки оформления документации».

Наименование программы на английском языке – «Documentation Validation Application»

Краткое наименование программы – «Приложение для проверки документации».

## Документы, на основании которых ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем программы тема курсового проекта. А также техническое задание к текущему программному продукту RU.17701729.05.12-01 ТЗ 01-1.

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение программы

### Функциональное назначение

Продукт позволит автоматически проверять документацию на наличии ошибок оформления, указывая при этом на их местоположение в тексте документа, и на нормативно-правовой акт, в соответствии с которым эта ошибка была выделена.

### Эксплуатационное назначение

Программное обеспечение будет поставляться в двух видах: в виде пакета для языка программирования Python, в виде веб-приложения. Целевой аудиторией являются технические писатели, студенты образовательной программы «Программная инженерия».

Python-пакет может быть встроен в иной программный продукт, где необходима валидация технической документации. Его целевой аудиторией являются разработчики систем создания технической документации и работы с ней.

Сценарии эксплуатации представлены в [приложении 2](#_ПРИЛОЖЕНИЕ_2).

## Краткая характеристика области применения

«Приложение для проверки оформления документации» - приложение, позволяющее пользователю проверить техническую документацию на соответствии стандартам оформления и содержания.

Продукт может применяться как помощник для технического писателя или как средство для тестирования систем автоматической генерации документации.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## Постановка задачи на разработку программы

В рамках курсового проекта необходимо разработать python-пакет и веб-приложение. Пакет должен:

1. Иметь возможность быть встроенным в иные программные продукты, для чего обладать системой зависимостей, описанной с помощью пакетного менеджера poetry.
2. Быть исполняемым как утилита командной строки. Следовательно, должен реализовывать взаимодействие с пользователем в текстовом режиме.
3. Включать инструменты анализа входных файлов формата docx по ЕСПД.

Веб-приложение представляет собой клиент-серверный продукт, состоящий из одной страницы. Клиентская часть реализуется c использованием HTML5, CSS3 и JavaScript. Серверная часть написана на языке Python с использованием фреймворка Django.

Функционал клиентской части включает пункты:

1. Форма для загрузки файлов формата docx
2. Таблица для отображения выходных данных со столбцами:
   1. Описание ошибки
   2. Место в документе
   3. Стандарт, в соответствии с которым классифицирована ошибка

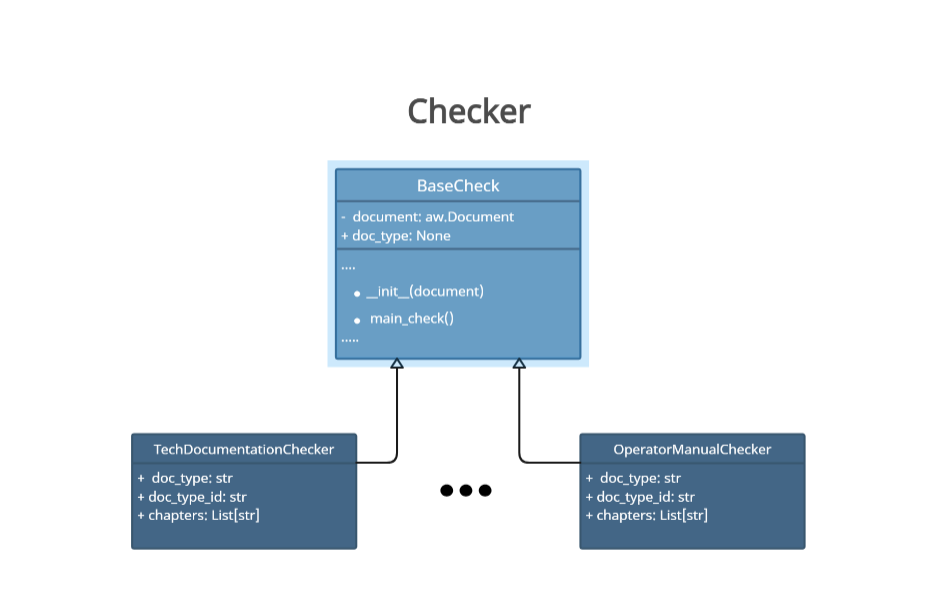
Серверная часть реализует следующие функции:

1. Обработку запросов пользователя с получением от него файлов
2. Обработка и возврат ошибок при некорректном обращении с формой
3. Обработка документа с использованием установленного python-пакета
4. Отправка пользователю на клиентскую часть веб-страницы с формой и результатами анализа.

## Описание архитектуры

### Архитектура python-пакета

Пакет состоит из трёх основных частей: исполняемый модуль (\_\_main\_\_), процедуры запуска анализа (модуль runners), анализаторы (модуль checker).

Рисунок 1. Диаграмма классов модуля Checker

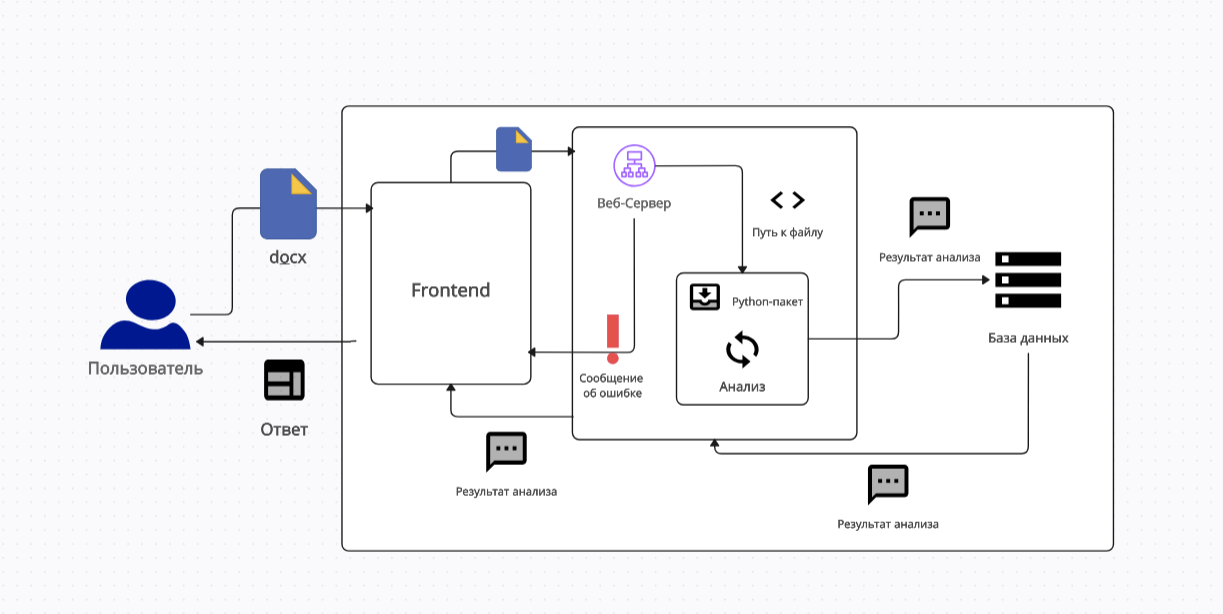
Исполняемый модуль взаимодействует с пользователем, обрабатывая поступающие от него входные данные (команды, файлы) и давая ему подсказки, если он что-либо делает не так. После валидации исполняемый модуль вызывает процедуру запуска анализа. Она выбирает подходящий анализатор и запускает анализ с помощью него. (подробнее – в пункте 3.3)

Основной элемент информации для обмена в программе – сообщения об ошибках в документе (класс Message в модуле utils), которые содержат информацию о месте, стандарте и описание ошибки, а также типе сообщения (ERROR/WARNING). Они собираются в вердикты (класс Verdict в модуле utils) по отдельным частям документа и сливаются в один.

### Архитектура веб-приложения

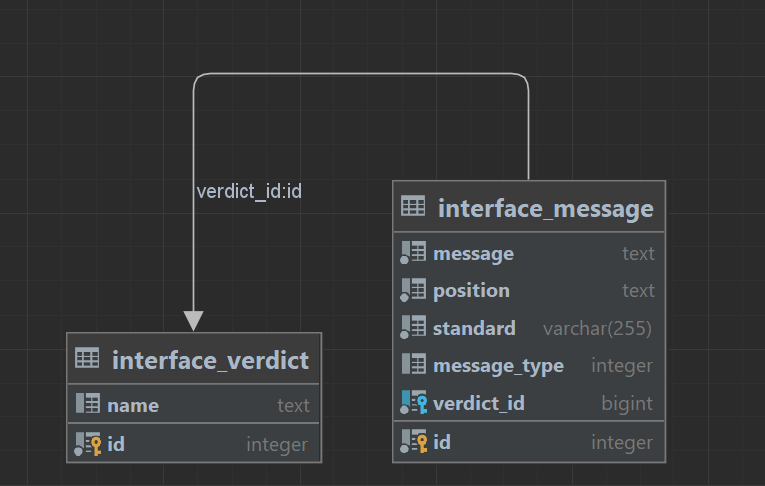
Проект придерживается принципов двухуровневой клиент-серверной архитектуры с одностраничной организацией. Основной фокус идет на обработке пользовательских файлов с использованием python-пакета. Клиентская часть (frontend) служит для создания визуального пользовательского интерфейса взаимодействия с сервером. Сервер обрабатывает запросы, формирует сообщения о возможных ошибках, получает файл и, сохраняя его, отправляет на обработку, затем конструирует клиентскую часть из шаблонов и данных и отправляет пользователю.

Построение системы осуществлялось в рамках шаблона Model View Controller. Бизнес-логика программы (обработка запросов от пользователя, сохранение файла, отправка его на анализ и получение результатов) выполняется на серверной части, клиентская часть отвечает за визуализацию данных, а сами данные хранятся на сервере с использованием СУБД SQLite, встроенной внутрь фреймворка Django.

Рисунок 2. Архитектура веб-приложения

Сервер взаимодействует с базой данных только для сохранения в ней нового анализа документа и получения из неё ранее сохраненного анализа. Готовый анализ хранится в виде сообщений и вердиктов в базе данных, что сочетается с архитектурой python-пакета.

Рисунок 3. Схема базы данных



Как видно, в отличие от пакета, в веб-приложении, вердикт служит лишь для агрегации сообщений вокруг одного анализа документа.

## Описание алгоритма и функционирования программы

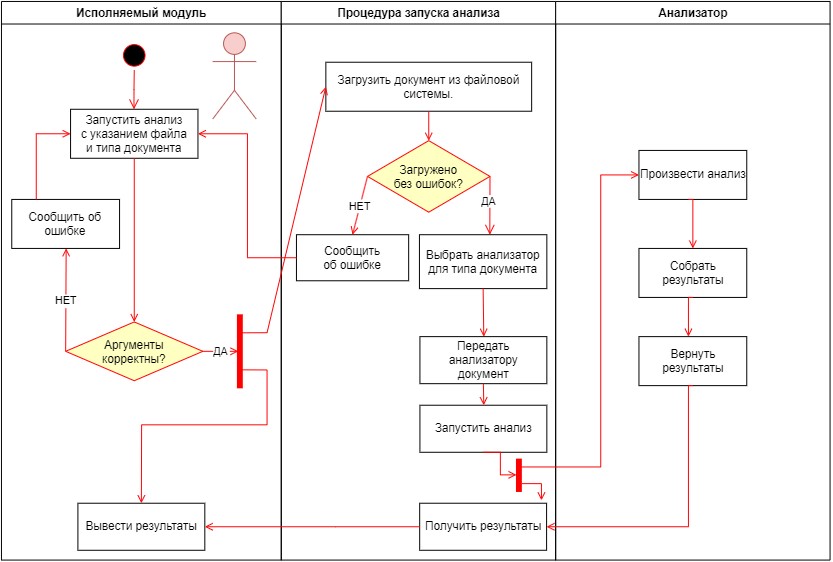
### Алгоритм работы python-пакета

#### Алгоритм взаимодействия с пользователем

Порядок действий режиме исполняемого модуля:

1. Пользователь запускает модуль, передавая ему путь к файлу и имя типа документа в аргументах командной строки.
2. Программа сообщает пользователю об ошибках и показывает ему краткую инструкцию по использованию.
3. Пользователь передаёт корректные аргументы.
4. Программа печатает результат анализа.

Рисунок 4. Диаграмма деятельности при взаимодействии с пользователем



#### Алгоритм анализа документа

Анализ документа производится одним из анализаторов в несколько этапов. Они разделяются на две группы: те, что требуют знаний о структуре документа, его содержании и остальные.

В первую группу входят:

1. Проверка общих стилей документа: настроек страницы, шрифтов, межстрочных интервалов.
2. Проверка листа утверждения.
3. Проверка титульного листа, извлечение идентификатора документа.
4. Проверка колонтитулов, сравнение идентификаторов в них с титульным листом.
5. Проверка списков и перечислений: нумерованы ли они, если нет, то подходящие ли используются маркеры.
6. Проверка содержания, работоспособности ссылок внутри документа и сбор информации о структуре разделов.

Во вторую группу входят:

1. Проверка форматирования заголовков, их иерархии в тексте (соответствия отступа слева уроню заголовка), с учётом информации о том, к разделу какого уровня они относятся. На этом этапе также проверяется расстояния до других заголовков и текста.
2. Проверка форматирования обычного текста, абзацных отступов, его выравнивания по левой стороне, с учётом информации о том, к разделу какого уровня он относятся.
3. Проверка наличия в содержании и тексте разделов, соответствующих заданному типу документа.

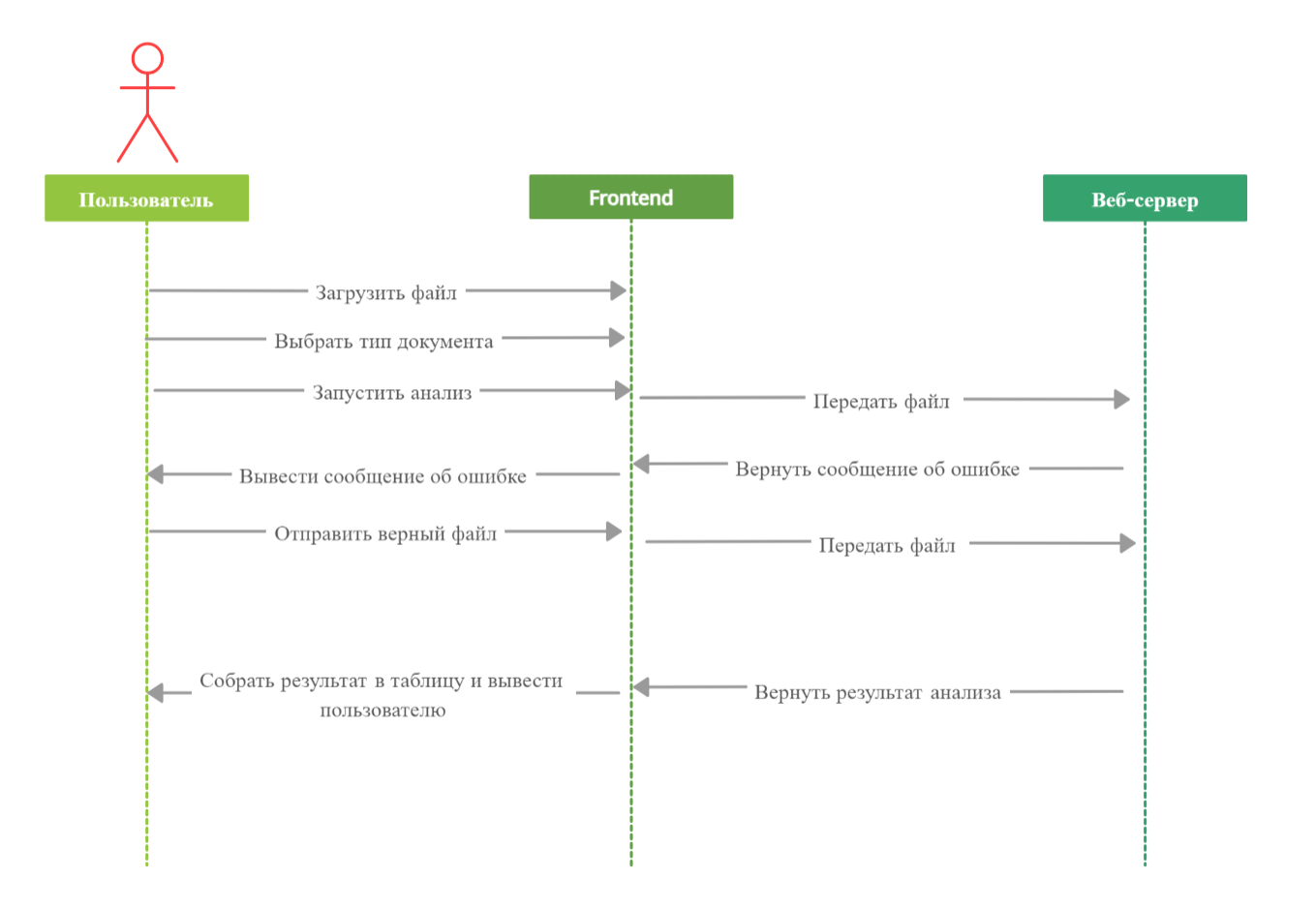
Если на этапе 6 были найдены ошибки, то проверки на этапах 7-9 не производятся.

### Алгоритм взаимодействия клиента с сервером

Общий порядок работы состоит в следующем:

1. Пользователь загружает файл.
2. Выбирает тип документа из доступных.
3. Запускает анализ.
4. Сервер проверяет, подходит ли файл для анализа. Если нет – сообщает пользователю об ошибке, он возвращается к пункту 1. Иначе – запускает анализ.
5. Сервер возвращает пользователю страницу с анализом.

Рисунок 5. Диаграмма последовательности действий веб-приложения



Выбор типа документа осуществляется путём нажатия на один из вариантов из предложенного списка.

При попытке перехода пользователя по несуществующему адресу или же внутренних сбоях клиентская часть веб-приложения демонстрирует страницу с сообщением об ошибке, информацией об ошибке.

## Организация входных и выходных данных

### Организация входных данных

Входными данными программного продукта являются файлы формата docx. Ограничений по размеру файла нет. Кроме файлов для проверки, входными данными является информация о типе документа. В случае веб-приложения эта информация задаётся в поле формы выбором из предложенных значений. В случае запуска исполняемого модуля python-пакета эта информация передаётся ему текстовой командой вида:

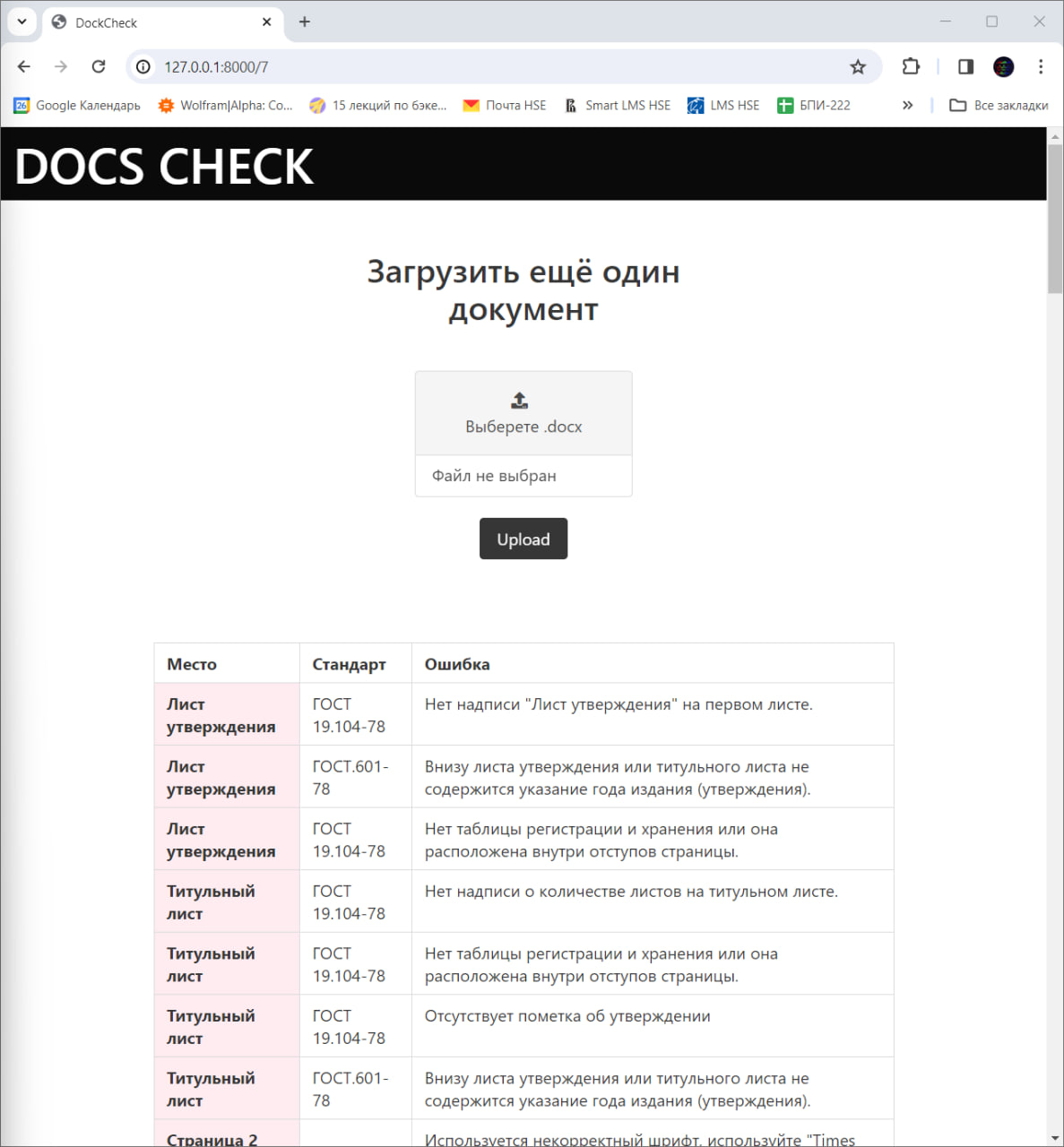
docsCheck <путь до файла> <тип документа>

### Организация выходных данных

В случае реализации в виде веб-приложения выходными данными является html-страница с таблицей результатов или, в зависимости от корректности переданного файла, таблица. Для исполняемого модуля python-пакета – текстовый вывод результатов анализа документа в консоль.

При выводе результата как в консоль, так и в таблицу, каждая строка имеет данные о позиции ошибке, стандарту (номеру ГОСТа), в соответствии с которым классифицирована ошибка и её текстовое описание.

Рисунок 6. Представление результата в виде таблицы.



### Хранение данных пользователя на стороне сервера

После отправки docx файла в виде формы POST-запросом файл временно сохраняется в локальном хранилище (директории media) и передаётся для python-пакету для анализа. По завершении анализа файлы могут быть удалены.

Никакие идентификационные данные пользователя, кроме самих файлов, в рамках проекта хранению не подлежат.

## Описание и обоснование выбора технических и программных средств

### Описание и обоснование выбора технических и программных средств python-пакета

Для работы python-пакета необходимы следующие программные средства:

1. Интерпретатор языка Python версии 3.9 и выше
2. Система управления пакетами PIP (package installer for Python), которая идёт в поставке с любой версией интерпретатора python или poetry.

При создании пакета была использована библиотека aspose-words версии. Она была выбрана постольку, поскольку единственная из существующих Python-библиотек предоставляет доступ к сложным инструментам текстового редактора, таким как расчёт расположения элементов документа в зависимости от настроек страницы. При установке пакета библиотека установится автоматически.

Специальных требований к клиентскому оборудованию не предъявляется. Достаточно, чтобы оно удовлетворяло требованиям, предъявляемым программными средствами, указанным выше.

### Описание и обоснование выбора серверного оборудования и программных средств веб-приложения

Требования к серверному оборудованию для корректной работы программы при штатной нагрузке:

1. 1 виртуальное ядро процессора Intel Xeon E5 2620 или AMD EPYC 7402P или выше;
2. 2 ГБ оперативной памяти;
3. 2 ГБ свободного дискового хранилища;
4. Скорость интернета не менее 100 Мбит/с.

Требования к программным средствам, используемым веб-приложением:

1. Интерпретатор языка Python версии 3.9 и выше
2. Система управления пакетами PIP (package installer for Python), которая идёт в поставке с любой версией интерпретатора python или poetry.
3. Python-пакет «docksCheck» версии 0.0.1 и выше
4. Веб-фреймворк Django версии 5
5. Библиотека django-simple-bulma версии 2.6.0 для поддержки CSS-фреймворка Bulma.

Фреймворк Django выбран как удобное средства для быстрого построения бэкенда веб-приложений, обладающих широким набором функций. Bulma выбран, поскольку предоставляет готовые и удобные инструменты стилизации веб-страниц и при этом является легковесным.

### Описание и обоснование технических и программных средств, необходимых для доступа к веб-приложению со стороны клиента

Для доступа к веб-сервису на клиентских устройствах, необходим следующий состав программных средств.

Для персональных стационарных или мобильных компьютеров:

1. Операционная система Windows 8 и выше;
2. Один из браузеров;

a) Google Chrome версии 105 или выше;

b) Microsoft Edge версии 105 или выше;

c) Mozilla Firefox версии 105 или выше;

d) Opera версии 90 или выше;

e) Safari версии 16.0 или выше.

Для смартфонов:

1. IOS 13.0 и выше;
2. Android 7.0 и выше.

При использовании иных программных средств корректное отображение элементов интерфейса и стабильная работа приложения не гарантируются.

Технические средства должны соответствовать требованиям к техническим средствам, предъявляемым выбранными программными средствами.

# ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Предполагаемая потребность

Анализ рынка показал, что существует множество программ, позволяющих генерировать документацию в различных форматах по настраиваемым или встроенным шаблонам. Но практически отсутствуют, программы с функцией анализа уже написанной документации (в основном это ограниченный дополнительный функционал в программах написания документации). К тому же, если при разработке ПО решают генерировать документацию по какому-либо стандарту автоматически (для более удобного контроля версий, то есть возможности поменять исходный код так, чтобы изменения отразились во всех видах документации), то отпадает надобность в проверке соответствия стандарту.

Однако за счёт фокусировки на ЕСПД и наличия дополнительного функционала расширения сторонними конфигурациями ориентировочный спрос на такое приложение для анализа технической документации должен быть достаточно высоким за счёт следующих факторов.

Во-первых, ЕСПД является широко используемым стандартом в области разработки и документирования цифровых продуктов. Многие компании в секторе информационных технологий России должны соответствовать этому стандарту при создании и выпуске своих продуктов. приложение, способное обрабатывать и анализировать документацию в соответствии с требованиями ЕСПД и иных стандартов, может значительно сэкономить время и затраты компаний.

Во-вторых, обладая реализацией в виде встраиваемого программного пакета, приложение может обрести высокий спрос как средство тестирования программ для создания документации.

## Ориентировочная экономическая эффективность

В рамках проекта расчет экономической эффективности программного продукта не производился.

## Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Для оценки преимуществ проекта было проведено сравнение функциональных характеристик нашего программного продукта с рядом аналогов. Результат сравнительного анализа представлен в таблице:

Таблица 1. Сравнительный анализ конкурентов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функционал | Параметры сравнения | Приложение для анализа технической документации | HelpnDoc | ClickHelp | «1С:Автоматизированная проверка конфигураций»  (непрямой аналог) |
| Поддержка платформы | Windows | **+** | **+** | **+** | **+** |
| Linux | **+** | **-** | **+** | **-** |
| IOS | **+** | **-** | **+** | **-** |
| Android | **+** | **-** | **+** | **-** |
| Базовый | Поддержка .docx | **+** | **+** | **+** | **-** |
| Контроль стилей текста | **+** | **+** | **+** | **+** |
| Контроль работоспособности гиперссылок | **+** | **+** | **+** | **+** |
| Дополнительный | Проверка структуры текста | **+** | **-** | **-** | **-** |
| Проверка орфографии | **-** | **+** | **-** | **+** |
| Расширение правил проверки | **+** | **-** | **-** | **+** |
| Поддержка ЕСПД | **+** | **-** | **-** | **-** |
| Возможность встроить в другое приложение | **+** | **-** | **-** | **-** |
| Режим доступа | | Свободный, бесплатно | Покупка,  299 евро | Подписка,  50$/мес. | Только с подпиской на ИТС «1С: Предприятие» |

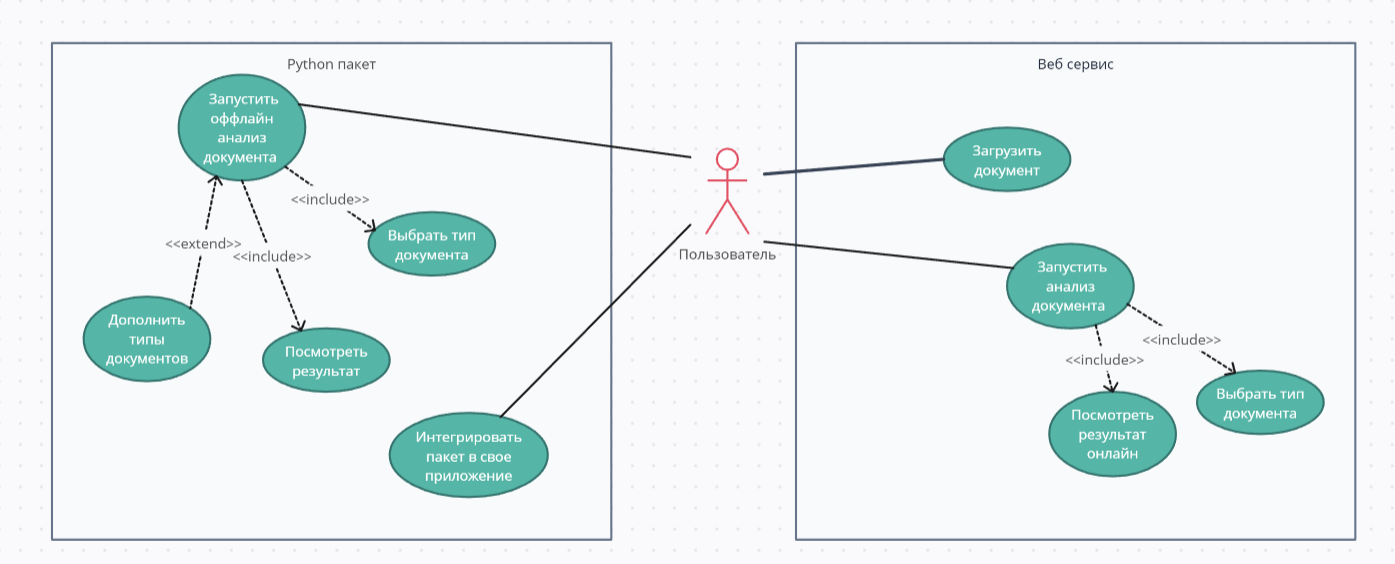
# ПРИЛОЖЕНИЕ 1

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. Продукт Python [Электронный ресурс] / Python. – Режим доступа: <https://www.python.org/>, свободный. (дата обращения: 23.01.2024)
3. Продукт HTML5 [Электронный ресурс] / HTML5. – Режим доступа: <https://html.spec.whatwg.org/>, свободный. (дата обращения: 23.01.2024)
4. Продукт CSS3 [Электронный ресурс] / CSS3. – Режим доступа: <https://www.w3.org/TR/css-syntax-3/>, свободный. (дата обращения: 23.01.2024)
5. Продукт JavaScript [Электронный ресурс] / JavaScript. – Режим доступа: <https://www.javascript.com/>, свободный. (дата обращения: 23.01.2024)
6. Продукт Django [Электронный ресурс] / Django. – Режим доступа: <https://www.djangoproject.com/>, свободный. (дата обращения: 23.01.2024)
7. Продукт PyPi (Python Package Index) [Электронный ресурс] / PyPi. – Режим доступа: <https://pypi.org/>, свободный. (дата обращения: 23.01.2024)
8. Django Templates. Документация. [Электронный ресурс] / Django. – Режим доступа: <https://docs.djangoproject.com/en/5.0/topics/templates/>, свободный. (дата обращения: 23.01.2024)
9. Продукт aspose-words [Электронный ресурс] / aspose-words. – Режим доступа: <https://pypi.org/project/aspose-words/>, свободный. (дата обращения: 23.01.2024)
10. Продукт Bulma [Электронный ресурс] / Bulma. – Режим доступа: <https://bulma.io/>, свободный. (дата обращения: 23.01.2024)
11. Продукт Poetry [Электронный ресурс] / Poetry. – Режим доступа: <https://python-poetry.org/>, свободный. (дата обращения: 23.01.2024)
12. Продукт pip [Электронный ресурс] / pip. – Режим доступа <https://pip.pypa.io/en/stable/installation/>, свободный. (дата обращения: 23.01.2024)
13. Продукт django-simple-bulma [Электронный ресурс] / django-simple-bulma. – Режим доступа: <https://pypi.org/project/django-simple-bulma/>, свободный. (дата обращения: 23.01.2024)

# ПРИЛОЖЕНИЕ 2

**UML ДИАГРАММА СЦЕНАРИЕВ ИСПОЛЬЗОВАНИЯ**



# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |