

Функциональные характеристики

ВЫШКА Регламент • версия 0.1.0, выпуск R2

Документ РГЛ-R2-ФХ-01. Редакция от 09.10.2026. ООО «НЬЮ-ССТ», ИНН 7733311994.

1. Назначение

Программа преобразует русскоязычный текст регламента, технического задания или инструкции в карточки кандидатов в требования. Пользователь проверяет исходный фрагмент, уточняет исполнение, фиксирует состояние, доказательство и проверяющего, сохраняет рабочий проект и выгружает отчёт.

Целевой пользователь - специалист, самостоятельно читающий исходный документ и принимающий решения. Программа формирует предложения; полнота выявления требований, юридическая квалификация и фактическое исполнение обязательств требуют проверки человеком.

2. Входные данные и разбор

Вводятся название и обычный текст; интерфейс принимает UTF-8 TXT/MD. Прямое распознавание PDF, DOCX, изображений и OCR отсутствует. До анализа такие материалы переводятся в текст вне программы, с проверкой переноса пользователем.

Сервер делит текст по строкам и пунктуации. Распознанные нейтральные заголовки и вводные метаданные консервативно исключаются; неоднозначные фразы могут остаться кандидатами. Для каждого фрагмента формируется карточка со ссылкой на источник. Срок и ответственный извлекаются только при явном указании; иначе значение отсутствует (null). Периодичность хранится текстом, без автоматического календарного расчёта.

3. Основные функции

1. Выделение кандидатов в требования и тематическая классификация по семи категориям: безопасность, документы, поставка/выполнение, сопровождение, качество, финансы, прочее.
2. Прослеживаемость: точная цитата, номер строки, координаты фрагмента в неизменённом источнике и SHA-256 исходного документа. Координаты задаются полуинтервалом [start,end) в кодовых точках Unicode; строки нумеруются с 1. CRLF, CR и LF считаются одним переходом строки.
3. Проверка предложений: показ категории, уверенности, влиятельных слов и признака needs_review. Пользователь может изменить категорию, текст карточки, срок/периодичность, ответственного и критерий приёмки. Исходное предложение модели хранится отдельно.
4. Управление состояниями: «Предложено ИИ», «В работе», «На проверке», «Принято», «Отклонено». Принятие требует непустых полей доказательства и проверяющего. Изменение принятой карточки снимает приёмку и возвращает её на проверку. История входит в полный JSON-отчёт.
5. Поиск и фильтрация по категории и состоянию. Фильтры меняют представление карточек, а не состав полного экспортируемого отчёта.
6. Явное сохранение и восстановление проекта в localStorage текущего браузерного профиля и адреса. Автосохранение отсутствует. При смене адреса или профиля копия автоматически не переносится.

7. Экспорт полного JSON для восстановления и CSV для сводного просмотра; импорт собственного JSON с проверкой схемы, SHA-256, координат, строк, категорий, состояний и обязательных полей. Повреждённый импорт не заменяет текущий проект. CSV не является полной резервной копией.

4. Алгоритм и измерения

Классификация реализована собственной функцией Multinomial Naive Bayes на словесных и символьных n-граммах. Словесные признаки имеют вес 2 с ограничением частоты до трёх; символьные n-граммы длины 3 и 4 - вес 0,25. Словарь содержит 5154 признака, $\alpha=0,75$, $\text{temperature}=5$. Нормализация регистра и ё/е относится к признакам и не изменяет исходные цитаты.

Корпус содержит 175 синтетических примеров: 140 обучающих и 35 отложенных (по 20 и 5 на класс). Получено 27/35 верных классификаций: accuracy 0,771429, macro-F1 0,739394; majority baseline 0,142857. Recall класса security: $1/5 = 0,20$. Проверка мала и синтетична; эти показатели не измеряют качество на документах заказчиков, полноту извлечения, юридическую правильность или точность срока/ответственного.

needs_review устанавливается при confidence < 0,65, отрыве от второго класса < 0,15 либо менее двух известных слов. На отложенной выборке отмечены 33/35 примеров, включая все восемь ошибок. Confidence не откалибрована и не является вероятностью правильности. Отсутствие флага не отменяет проверки человеком.

Критерий приёмки предлагается по категории. LLM, внешние API моделей и дообучение на запросах пользователей не используются. Этот документ описывает фактический алгоритм и не является специальным экспертным заключением о признаке ИИ для реестра.

5. Архитектура и информационный обмен

Backend использует Python 3.12 и стандартную библиотеку. src/classifier.py реализует классификацию; src/train.py - обучение и оценку; src/engine.py - валидацию, сегментацию и извлечение явных полей; src/server.py - HTTP API и выдачу статических файлов. Интерфейс web/ содержит HTML/CSS/JavaScript/SVG без npm-пакетов и CDN. data/ содержит корпус, модель и результаты измерений.

Браузер передаёт {text,title} в POST /api/analyze своего origin. Сервер возвращает карточки и метаданные. Исходный текст в полный отчёт добавляется из снимка ввода браузера. GET /api/health сообщает состояние, GET /api/model - сведения о модели. Серверного API сохранения проектов и пользователей нет.

Схема отчёта: vyshka.reglament.report.v1. Document и model_suggestion отделены от редактируемых полей карточки. SHA-256 вычисляется по точным UTF-8 байтам исходника; он контролирует целостность текста, но не удостоверяет автора или историю изменений.

6. Ограничения и обработка данных

Лимиты: 60 000 кодовых точек Unicode для текста, 200 для названия, 262 144 байта HTTP-тела, 250 карточек. Переполнение возвращает ошибку, без успешного частичного отчёта. Импорт JSON ограничен 16 МиБ. Сервер ограничивает одновременные соединения (12), очередь (24), бездействие сокета (8 секунд), получение POST-тела (15 секунд) и POST-запросы (30 за 60 секунд на TCP peer). За обратным прокси лимит общий для peer.

Проверенная среда: Ubuntu 24.04.5 LTS, Python 3.12.3, 4 vCPU / 8 ГБ / 80 ГБ. Это фактическая конфигурация проверки; минимальная RAM, RPS, число пользователей и гарантированная доступность не измерены. Совместимость с двумя российскими ОС не подтверждена.

Тела запросов обрабатываются приложением в памяти и не журналируются приложением. Браузер и экспорт сохраняют данные по явному действию пользователя. Технические журналы, буферы, swap/core, прокси и резервные копии инфраструктуры регулируются отдельно. Для пилота предусмотрены синтетические или обезличенные материалы. Полный JSON содержит исходник и требует соответствующего режима доступа.

В программе нет учётных записей, ролевой модели, серверной базы пользовательских проектов, квалифицированной электронной подписи и защищённого серверного журнала приёма. Ввод имени проверяющего не удостоверяет личность. Состояние «Принято» не подтверждает внешнее исполнение обязательства.

7. Идентификация выпуска

Дистрибутив: vyshka-reglament-0.1.0-r2.tar.gz; SHA-256:

79c33e07c48f3f8708d3d8680c195db2c383dcd90c7aee14e3a030ef23b74f1f

Результаты R2: 52/52 Python-проверки, 14/14 проверки исправлений интерфейса, 15/15 JS-регрессии; изолированное восстановление приложения проверено отдельно. Программные проверки обработчиков не заменяют подтверждение каждой операции скачивания/импорта в конкретном браузере. Публичная доступность и включение в реестр этим документом не удостоверяются.