МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра информационных технологий в атомной энергетике (ИТАЭ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ 2

по дисциплине «Автоматизированные системы управления элементами промышленных и административных объектов»

АСУ П по подбору и продаже мультимедиа контента

Студент группы *ИКБО-50-23, Враженко Д.О.* _____
(подпись)

Преподаватель *Щербаков К.В.*

_____
(подпись)

Отчет представлен «___»_____202__г.

Москва 2026 г.

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

№	Функциональное требование	Обоснование (зачем это нужно)	Примеры реализации / Источники
Подсистема взаимодействия			
1.	Прием текстовых запросов на естественном языке в свободной форме.	Пользователь должен иметь возможность общаться с системой так же, как с человеком-менеджером, без необходимости изучения сложного синтаксиса поиска.	Реализовано в поисковых системах (Google, Яндекс), чат-ботах (ChatGPT). [Источник: Анализ интерфейсов современных поисковиков]
2.	Поддержка голосового ввода запросов.	Обеспечение удобства для пользователей мобильных устройств и людей с ограниченными возможностями, расширение каналов ввода.	Реализовано в голосовых помощниках ("Алиса", "Маруся"), приложениях (Shazam). [Источник: Требования к современным мобильным приложениям]
3.	Отображение результатов поиска в виде структурированного списка (карточки контента).	Обеспечение наглядности и удобства восприятия информации. Пользователь должен видеть название, тип, автора, уровень сложности и краткое описание.	Реализовано в интернет-магазинах (Ozon, Wildberries), онлайн-кинотеатрах (Кинопоиск). [Источник: Анализ UI/UX крупных маркетплейсов]
4.	Фильтрация и сортировка выдачи по различным параметрам (тип контента, автор, дата, рейтинг, уровень сложности).	Предоставление пользователю возможности самостоятельно уточнить результаты, если автоматический подбор оказался недостаточно точен.	Реализовано в библиотечных каталогах, системах электронного обучения (Coursera). [Источник: Функциональные требования к ЭБС]
5.	Ведение и отображение истории диалога и предыдущих запросов пользователя в личном кабинете.	Обеспечение контекста для уточняющих запросов, возможность вернуться к ранее найденным материалам без повторного поиска.	Реализовано в мессенджерах (Telegram) и поддержке пользователей (Zendesk). [Источник: CRM-системы, управление взаимодействием с клиентами]
6.	Сохранение выбранного контента в "Избранное" или "Мои материалы".	Реализация персонализации и возможности формирования пользователем собственной библиотеки для последующего доступа.	Стандартная функция для всех медиа-платформ (Spotify, YouTube, ЛитРес). [Источник: Анализ функционала стриминговых сервисов]

№	Функциональное требование	Обоснование (зачем это нужно)	Примеры реализации / Источники
Подсистема семантического анализа на базе корпоративной нейросети			
7.	Автоматический семантический анализ и распознавание сущностей из текста запроса.	Преобразование неструктурированной фразы ("хочу изучить Python для начинающих") в структурированные данные: {Тема: "Python", Уровень: "для начинающих", Цель: "изучить"}.	Реализовано в системах обработки естественного языка (BERT, GPT). [Источник: Научные статьи по NLP, материалы конференций по ИИ]
8.	Определение интента (намерения) пользователя на основе запроса и контекста.	Понимание цели запроса: купить, посмотреть, скачать, изучить, прослушать. Это критически важно для корректного подбора типа контента и действий системы.	Реализовано в чат-ботах технической поддержки и виртуальных ассистентах. [Источник: Документация по проектированию диалоговых систем]
9.	Учет контекста предыдущих запросов из истории диалога при анализе текущего запроса.	Обработка уточнений ("нет, мне нужен продвинутый уровень по Go"), когда система должна помнить тему ("Go") и изменить только параметр уровня.	Реализовано в диалоговых системах (Google Duplex). [Источник: Исследования в области многосессионных диалогов]
10.	Обеспечение круглосуточной доступности (24/7) сервиса семантического анализа.	Корпоративная нейросеть работает всегда, в отличие от человека-менеджера. Это главное преимущество автоматизации, обеспечивающее мгновенный ответ в любое время.	Стандартное требование к серверным решениям и облачным сервисам (SLA 99.9%). [Источник: ГОСТ Р ИСО/МЭК 20000-1, требования к доступности ИТ-услуг]
Подсистема управления контентом и поиска			
11.	Полнотекстовый поиск по метаданным и содержимому внутренней базы контента.	Обеспечение быстрого и релевантного поиска по всему массиву имеющихся материалов на основе формализованных параметров от нейросети.	Реализовано в системах управления базами данных с поддержкой полнотекстового индекса (Elasticsearch). [Источник: Техническая документация СУБД]
12.	Мультиформатный поиск: одновременный поиск по всем типам контента (книги, видео, аудио).	Пользователю не нужно указывать тип контента, система сама может предложить разные форматы по одной теме, предоставляя выбор.	Реализовано на агрегаторах контента (Google Книги, iTunes). [Источник: Обзор мультимедийных платформ]
13.	Автоматическое определение недостаточности имеющегося контента для удовлетворения запроса.	Если по запросу "Продвинутое машинное обучение на TensorFlow" во внутренней базе ничего нет, система должна это понять и инициировать поиск вовне.	Реализовано в системах рекомендаций для пополнения каталога (Amazon, Netflix). [Источник: Описание работы рекомендательных систем]

№	Функциональное требование	Обоснование (зачем это нужно)	Примеры реализации / Источники
14.	Автоматическое формирование поискового запроса и обращение к API внешних источников (партнерские библиотеки, магазины контента).	Автоматизация процесса поиска нового, еще не приобретенного контента, для расширения базы под конкретную потребность пользователя.	Реализовано в системах сравнения цен и агрегаторах товаров (Яндекс.Маркет). [Источник: Принципы работы API-интеграций]
15.	Сбор и нормализация метаданных о внешнем контенте из различных источников.	Приведение информации из разных внешних систем (разные форматы данных) к единому внутреннему формату для удобного отображения пользователю.	Реализовано в ETL-процессах (Extract, Transform, Load) при интеграции данных. [Источник: Курсы по интеграции информационных систем]
Подсистема поддержки принятия решений и управления			
16.	Ранжирование (сортировка) результатов поиска по степени релевантности запросу.	Наиболее подходящие материалы должны быть в начале списка, чтобы пользователь не пролистывал сотни нерелевантных позиций.	Реализовано во всех поисковых системах (на основе PageRank, TF-IDF и др.). [Источник: Алгоритмы машинного обучения в информационном поиске]
17.	Персонализация результатов на основе профиля и истории пользователя.	Пользователь, который постоянно слушает подкасты, должен видеть в выдаче подкасты выше, чем длинные видео, даже если формально они равны по релевантности.	Реализовано в рекомендательных системах (Spotify, Netflix). [Источник: Патенты и публикации компаний о персонализации]
18.	Автоматическое создание задачи контент-менеджеру на рассмотрение возможности приобретения нового контента из внешних источников.	Вовлечение человека в процесс на финальной стадии принятия решения о закупке, но без его участия в рутинном поиске и сборе информации.	Реализовано в системах Service Desk и workflow-системах (Jira, Trello). [Источник: Функциональные требования к системам автоматизации бизнес-процессов]
19.	Учет прав доступа и лицензий на контент при выдаче результатов.	Система не должна предлагать пользователю контент, на который у организации нет прав для его предоставления данному пользователю.	Реализовано в системах управления цифровыми правами (DRM) и корпоративных библиотеках. [Источник: Требования информационной безопасности]
20.	Логирование всех действий системы и запросов пользователей для последующего анализа и дообучения нейросети.	Сбор данных о том, как пользователи формулируют запросы, на что они кликают, какие результаты считают хорошими. Это необходимо для улучшения работы нейросети.	Реализовано в системах сбора логов и аналитики (ELK stack, Splunk). [Источник: Методологии Data-Driven Development]

НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

№	Нефункциональное требование (категория)	Обоснование (зачем это нужно)	Примеры реализации / Источники
Производительность и надежность			
1.	Время обработки типового запроса и выдачи результата не должно превышать 2 секунд.	Пользователи ожидают мгновенного ответа от автоматизированной системы. Длительное ожидание приведет к негативному опыту и отказу от использования.	Стандарты производительности веб-приложений (Google Core Web Vitals). [Источник: Исследования юзабилити, стандарты производительности]
2.	Коэффициент готовности системы (доступность) должен быть не ниже 99.5% (не более 3.5 часов простоя в месяц).	Система является корпоративным сервисом и должна быть доступна практически всегда, заменяя человека 24/7. Длительные простои недопустимы.	Стандарт для коммерческих SaaS-решений (SLA). [Источник: ГОСТ Р ИСО/МЭК 20000-1]
3.	Система должна обеспечивать одновременную работу не менее 1000 активных пользователей без деградации производительности.	На начальном этапе эксплуатации необходимо заложить запас прочности, чтобы выдержать пиковые нагрузки и рост числа пользователей.	Требования к нагрузочному тестированию. [Источник: Типовые требования к высоконагруженным системам]
4.	Время восстановления после сбоя (RTO - Recovery Time Objective) не должно превышать 1 часа.	В случае аварии (падение сервера, ошибка в коде) система должна быть быстро возвращена в строй, чтобы минимизировать ущерб от простоя.	Стандарты в области обеспечения непрерывности бизнеса (BCM). [Источник: ISO 22301]
5.	Система должна корректно обрабатывать пиковые нагрузки (например, начало учебного года) без потери запросов (должна быть предусмотрена автоматическая масштабируемость).	Обеспечение стабильной работы в периоды, когда количество запросов резко возрастает.	Принципы построения облачных архитектур (AWS Auto Scaling). [Источник: Архитектуры распределенных систем]
Безопасность и конфиденциальность			
6.	Передача всех данных между клиентом и сервером должна осуществляться по протоколу HTTPS с шифрованием TLS 1.3.	Защита персональных данных пользователей и истории их запросов от перехвата в публичных сетях.	Закон "О персональных данных" (ФЗ-152), стандарты безопасности ПДн. [Источник: ФЗ-152, ГОСТ 34.10]
7.	Доступ к модели нейросети и данным для ее обучения должен быть строго ограничен на уровне инфраструктуры.	Корпоративная нейросеть является интеллектуальной собственностью компании. Необходимо защитить ее от кражи или несанкционированного доступа.	Стандарты управления доступом (RBAC - Role-Based Access Control). [Источник: ИСО/МЭК 27001]

№	Нефункциональное требование (категория)	Обоснование (зачем это нужно)	Примеры реализации / Источники
8.	Система должна обеспечивать разграничение прав доступа для разных ролей пользователей (пользователь, контент-менеджер, администратор).	Предотвращение случайных или намеренных изменений в системе неавторизованными лицами. Контент-менеджер не должен иметь доступ к настройке нейросети.	Стандартный подход к проектированию корпоративных систем. [Источник: Требования к АСУ различного уровня]
9.	Аутентификация пользователей должна быть надежной (например, с поддержкой двухфакторной аутентификации для сотрудников).	Снижение риска компрометации учетных записей пользователей и получения доступа к истории их запросов или (для сотрудников) к управлению системой.	Рекомендации ФСТЭК и Минцифры по безопасности веб-приложений. [Источник: Методики оценки угроз безопасности]
10.	Система должна обеспечивать невозможность восстановления персональных данных пользователя из обезличенных логов для аналитики.	Выполнение требований по обезличиванию ПДн при их обработке для улучшения качества сервиса.	[Источник: Разъяснения Роскомнадзора по обезличиванию данных]
Удобство использования			
11.	Интерфейс системы должен быть интуитивно понятным и не требовать от пользователя предварительного обучения.	Система должна быть доступна широкому кругу пользователей с разным уровнем цифровой грамотности.	Эвристики Якоба Нильсена. [Источник: Книги и статьи по юзабилити]
12.	Интерфейс должен быть адаптивным (responsive) и корректно отображаться на всех типах устройств (ПК, планшет, смартфон).	Пользователи могут обращаться к системе с любых устройств, интерфейс должен подстраиваться под размер экрана.	Требования к современной веб-разработке. [Источник: Best practices frontend-разработки]
13.	Система должна предоставлять пользователю понятные сообщения об ошибках (например, "Ничего не найдено по вашему запросу, попробуйте изменить формулировку").	Сообщения типа "Error 500" пугают и дезориентируют пользователя. Система должна "человеческим" языком объяснять, что пошло не так.	[Источник: Стандарты проектирования пользовательского опыта]
Технологические и архитектурные			
14.	Модель нейросети должна быть развернута в контейнеризированной среде (например, Docker) для обеспечения воспроизводимости и простоты обновления.	Упрощение процесса развертывания новых версий модели, обеспечение независимости от окружения на сервере.	Современные практики DevOps и MLOps. [Источник: Документация Docker, Kubernetes]

№	Нефункциональное требование (категория)	Обоснование (зачем это нужно)	Примеры реализации / Источники
15.	Система должна быть построена по микросервисной архитектуре для обеспечения независимости компонентов (анализ, поиск, интеграция).	Отказ одного микросервиса (например, внешнего поиска) не должен приводить к остановке всей системы (внутренний поиск продолжит работать). Легче масштабировать и обновлять компоненты по отдельности.	[Источник: "Микросервисы" Мартина Фаулера, паттерны архитектуры ПО]
16.	API для взаимодействия с внешними источниками контента должны быть стандартизированы (REST/gRPC).	Упрощение процесса подключения новых внешних источников в будущем.	[Источник: Принципы проектирования Web API]
17.	Система должна вести подробное логирование всех внутренних операций для последующего анализа инцидентов и производительности.	Невозможно отладить работу сложной системы без детальных логов. Это требование для эксплуатации.	[Источник: Практики Site Reliability Engineering (SRE)]
Правовые и нормативные			
18.	Хранение и обработка персональных данных пользователей (ФИО, email, история запросов) должны осуществляться на серверах, расположенных на территории РФ.	Выполнение требования федерального закона № 242-ФЗ о локализации баз данных персональных данных.	[Источник: Федеральный закон №242-ФЗ]
19.	Весь контент, предоставляемый пользователям, должен распространяться в соответствии с лицензионными договорами, система должна вести их учет.	Соблюдение авторских прав, предотвращение судебных исков и претензий от правообладателей.	[Источник: Гражданский кодекс РФ, Часть 4]
Эргономические			
20.	Цветовая гамма и элементы интерфейса не должны вызывать быстрого утомления при длительной работе; рекомендуется использование светлого и темного режимов.	Обеспечение комфорта пользователей, которые могут проводить в системе много времени (например, студенты за подбором материалов для курсовой).	СанПиН 1.2.3685-21 "Гигиенические нормативы...", стандарты WCAG. [Источник: СанПиН 1.2.3685-21]