

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.01 – Информатика и вычислительная техника  
Направленность (профиль) образовательной программы Информатика и вычислительная техника

ДОПУСТИТЬ К ЗАЩИТЕ  
Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Разработка интеллектуальной системы поддержки пользователей для организации ООО «Амурагроцентр»

Исполнитель

студент группы 1103-об

\_\_\_\_\_

(подпись, дата)

С.А. Исаков

Руководитель

доцент, канд.техн.наук

\_\_\_\_\_

(подпись, дата)

С.Г. Самохвалова

Консультант

по безопасности и  
экологичности

\_\_\_\_\_

(подпись, дата)

А.Б. Булгаков

доцент, канд. техн. наук

Нормоконтроль

Инженер кафедры

\_\_\_\_\_

(подпись, дата)

В.Н. Адаменко

Благовещенск 2025

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов

« \_\_\_\_\_ » \_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

К выпускной квалификационной работе студента Исакова С.А.

1. Тема выпускной квалификационной работы: Разработка интеллектуальной системы поддержки пользователей для организации ООО «Амурагроцентр»

(утверждено приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025 г.

3. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): анализ предметной области проектирования, обоснование необходимости разработки, проектирование интеллектуальной системы, программная реализация системы, руководство пользователя и администратора, защита информации, безопасность и экологичность.

4. Перечень материалов приложения: техническое задание

5. Консультанты по выпускной квалификационной работе: Консультант по безопасности и экологичности: доцент, канд. техн. наук А.Б. Булгаков

6. Дата выдачи задания: 10.10.2024 г.

Руководитель выпускной квалификационной работы: \_\_\_\_\_  
доцент, канд. техн. наук С.Г. Самохвалова

(фамилия, имя, отчество, должность, уч. степень, уч. звание)

Задание принял к исполнению (10.10.2024): \_\_\_\_\_

(подпись студента)

## РЕФЕРАТ

Бакалаврская работа содержит 78 с., 13 рисунков, 10 таблиц, 1 приложение, 26 источников.

### ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, БАЗА ЗНАНИЙ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, БОЛЬШИЕ ЯЗЫКОВЫЕ МОДЕЛИ

Разработка интеллектуальной системы поддержки пользователей для организации ООО «Амурагроцентр».

Цель работы заключается в создании интеллектуальной системы, предназначенной для оптимизации процессов технической поддержки и управления корпоративной базой знаний в ООО «Амурагроцентр» путем интеграции технологий искусственного интеллекта, в частности, больших языковых моделей и внешних ИИ API. Система обеспечивает интеллектуальный поиск информации по запросам на естественном языке, автоматическую генерацию резюме статей и релевантных тегов, а также предоставляет инструменты для управления контентом базы знаний.

Разработанный программный продукт обладает интуитивно понятным интерфейсом, ориентированным на сотрудников организации. Результатом работы является готовая к внедрению интеллектуальная система поддержки, способствующая повышению эффективности работы пользователей и ИТ-специалистов ООО «Амурагроцентр».

## СОДЕРЖАНИЕ

Введение	8
1 Анализ предметной области проектирования	9
1.1 Краткая характеристика предприятия	9
1.2 Анализ существующих процессов технической поддержки	11
1.3 Выявление проблем и потребностей в интеллектуальной системе	13
2 Обоснование необходимости разработки	15
2.1 Анализ актуальности разработки интеллектуальной системы	15
2.2 Обзор и анализ методов интеллектуализации систем поддержки	17
2.3 Анализ и выбор внешних ИИ API	19
2.4 Анализ рынка на наличие схожих решений	22
2.5 Определение и утверждение требований к системе	23
2.6 Расчет экономической эффективности	25
3 Проектирование интеллектуальной системы	29
3.1 Разработка архитектуры программного решения	29
3.2 Проектирование модуля интеграции с внешними ИИ-сервисами	31
3.3 Проектирование многостадийного ИИ-поиска	33
3.4 Проектирование модуля управления статьями и тегами	34
3.5 Проектирование базы данных	36
3.6 Характеристика функциональных подсистем	45
3.7 Виды обеспечения системы	46
4 Реализация системы	47
4.1 Структура проекта	47
4.2 Реализация модулей системы	47
5 Руководство пользователя и администратора	49
5.1 Руководство пользователя	49
5.2 Руководство администратора	52
6 Защита информации	56
6.1 Модель угроз и основные векторы атак	56
6.2 Аутентификация и авторизация	57
6.3 Защита от веб-уязвимостей	58

6.4 Рекомендации по обеспечению безопасности	59
7 Безопасность и экологичность	61
7.1 Безопасность	61
7.2 Экологичность	65
7.3 Чрезвычайные ситуации	69
Заключение	74
Библиографические ссылки	75
Библиографический список	76
Приложение А	79

## НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе были использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 2.103-68 ЕСКД Стадии разработки;

ГОСТ 2.104-68 ЕСКД Основные надписи;

ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам;

ГОСТ 2.111-2013 ЕСКД. Нормоконтроль;

ГОСТ 7.1-2003 СИБИД. Библиографическая запись. Библиографическое описание. Общие требования и правила составления;

ГОСТ 8.417-2002 Государственная система обеспечения единства измерений (ГСИ). Единицы величин (с Поправками);

ГОСТ Р 12.2.143-2009 Система стандартов безопасности труда (ССБТ). Системы фотолюминесцентные эвакуационные. Требования и методы контроля (с Изменением N 1);

ГОСТ 12.4.026-2015 Система стандартов безопасности труда (ССБТ). Цвета сигнальные, знаки безопасности и разметка сигнальная. Назначение и правила применения. Общие технические требования и характеристики. Методы испытаний (с Поправками);

ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем;

ГОСТ Р ИСО 14915-1-2016 Эргономика мультимедийных пользовательских интерфейсов. Часть 1. Принципы проектирования и структура;

ГОСТ Р 50922-2006 Защита информации. Основные термины и определения;

ГОСТ Р 50948-2001. Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности. Введен в действие с 01.07.2002.

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

АСУ – Автоматизированная система управления

БД – База данных

БЯМ – Большие языковые модели

ИИ – Искусственный интеллект

ИТ – Информационные технологии

ОЕЯ – Обработка естественного языка

ПО – Программное обеспечение

СУБД – Система управления базами данных

API – (Application Programming Interface) Интерфейс прикладного программирования

CSS – (Cascading Style Sheets) Каскадные таблицы стилей

CSRF – (Cross-Site Request Forgery) Подделка межсайтовых запросов

HTML – (HyperText Markup Language) Язык гипертекстовой разметки

HTTP – (HyperText Transfer Protocol) Протокол передачи гипертекста

HTTPS – (HyperText Transfer Protocol Secure) Защищенный протокол передачи гипертекста

JS – Язык программирования JavaScript

JSON – (JavaScript Object Notation) Объектная нотация JavaScript

MVC – (Model-View-Controller) Модель-Представление-Контроллер

RBAC – (Role-Based Access Control) Управление доступом на основе ролей

SQL – (Structured Query Language) Язык структурированных запросов

XSS – (Cross-Site Scripting) Межсайтовый скриптинг

## ВВЕДЕНИЕ

Эффективность работы предприятия напрямую зависит от стабильности его ИТ-инфраструктуры. Рост объемов данных и усложнение программного обеспечения приводят к увеличению числа обращений в техподдержку.

Актуальность работы обусловлена необходимостью перехода от ручной поддержки к интеллектуальной, способной не только быстрее решать, но и предвидеть проблемы. Внедрение технологий искусственного интеллекта (ИИ), в частности обработки естественного языка и больших языковых моделей, позволяет создать систему, которая понимает запросы пользователей и автоматически находит релевантные решения в базе знаний [6].

Цель квалификационной работы: разработка интеллектуальной системы поддержки пользователей для ООО «Амурагроцентр» с интеграцией API систем искусственного интеллекта для улучшения процессов поиска информации и управления корпоративной базой знаний.

Объект исследования: процессы предоставления технической поддержки и управления корпоративными знаниями в ООО «Амурагроцентр».

Предмет исследования: методы и средства разработки веб-ориентированных систем управления базами знаний и автоматизированной генерации контента.

Задачи исследования:

- провести анализ предметной области и существующих процессов поддержки в ООО «Амурагроцентр»;
- обосновать необходимость разработки интеллектуальной системы;
- спроектировать архитектуру системы, структуру базы данных и пользовательский интерфейс;
- разработать программные модули системы, включая интеллектуальные функции.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОЕКТИРОВАНИЯ

## 1.1 Краткая характеристика предприятия

Общество с ограниченной ответственностью «Амурагроцентр», зарегистрированное 26 ноября 2001 года в городе Благовещенске Амурской области, представляет собой крупное агропромышленное предприятие. Основным направлением его деятельности является производство готовых кормов для сельскохозяйственных животных. Помимо этого, значительную долю в производственной деятельности занимают выпуск соевого масла, шрота, а также другие операции, связанные с переработкой сельскохозяйственной продукции. Предприятие располагает обширной производственной базой, которая включает в себя производственные комплексы, складские помещения, административные здания и сельскохозяйственные угодья.

Организационная структура ООО «Амурагроцентр» представлена совокупностью различных подразделений, обеспечивающих функционирование предприятия. К ним относятся управление производством, бухгалтерия, отдел кадров, отдел сбыта, отдел логистики, отдел маркетинга. Особое место занимает отдел автоматизированных систем управления (АСУ), на который возложены функции информационно-технического обеспечения всей деятельности предприятия. Организационная структура предприятия представлена на рисунке 1.

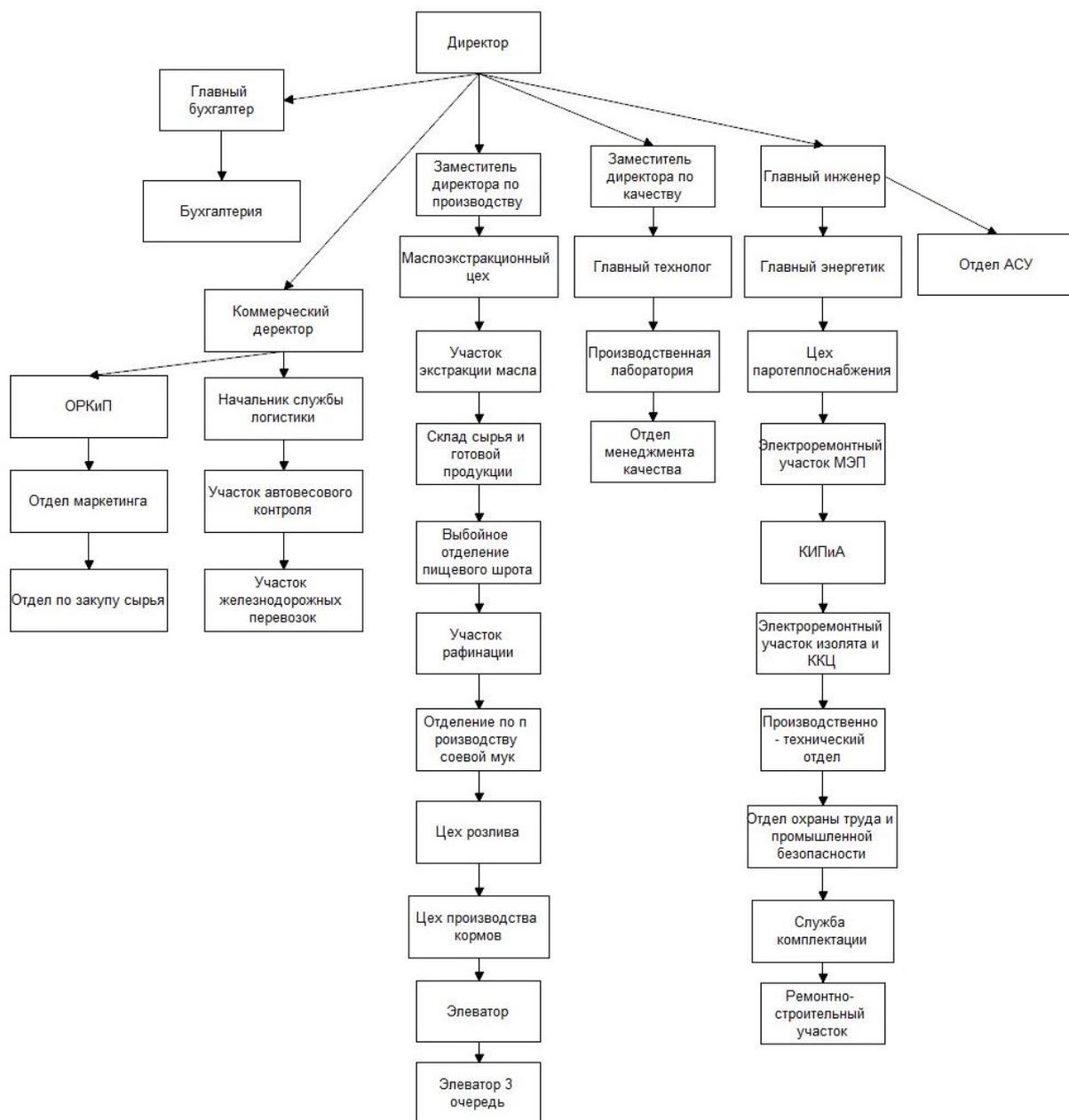


Рисунок 1 – Организационная структура предприятия

ИТ инфраструктура предприятия построена на основе современного серверного оборудования, включающего как физические, так и виртуальные серверы под управлением операционных систем семейства Windows Server. Данные серверы обеспечивают функционирование корпоративных сервисов, баз данных и прикладных программ. Сетевая инфраструктура реализована в виде локальной вычислительной сети, которая объединяет рабочие места сотрудников, серверы и периферийные устройства. Для подключения используются как проводные

(Ethernet), так и беспроводные (Wi-Fi) технологии, обеспечивая также доступ к глобальной сети Интернет.

Рабочие места пользователей оснащены персональными компьютерами и ноутбуками, функционирующими преимущественно под управлением операционных систем Windows 10/11. На них установлен стандартный набор офисного программного обеспечения, такой как Microsoft Office и веб-браузеры, а также специализированные программные продукты, включая «1С:Бухгалтерия» и программное обеспечение для управления производственными процессами. В организации применяются различные информационные системы для автоматизации ключевых бизнес-процессов, в том числе системы бухгалтерского учета, управления кадрами, складского учета и электронного документооборота (на платформе Битрикс24). ИТ-инфраструктура также включает разнообразные периферийные устройства, такие как принтеры, сканеры, многофункциональные устройства, сетевое оборудование (маршрутизаторы, коммутаторы).

Отдел АСУ несет ответственность за администрирование серверов и сети, установку и настройку программного обеспечения, обеспечение информационной безопасности и оказание технической поддержки пользователям. Именно для оптимизации работы данного отдела и повышения качества поддержки пользователей разрабатывается предлагаемая интеллектуальная система.

## **1.2 Анализ существующих процессов технической поддержки**

В настоящее время процесс оказания технической поддержки пользователям в ООО «Амурагроцентр» характеризуется как преимущественно централизованный, однако недостаточно автоматизированный. Основная нагрузка по обработке запросов и решению проблем ложится на сотрудников отдела АСУ.

Сотрудники предприятия могут инициировать обращение за технической поддержкой несколькими способами. Наиболее распространенными являются прямой телефонный звонок специалисту отдела АСУ, отправка электронного письма с описанием возникшей проблемы на общий адрес отдела или конкретному специалисту, личное обращение к специалисту в его рабочем кабинете, а

также использование системы электронного документооборота (Битрикс24) для постановки соответствующих задач.

Следует отметить, что единой системы регистрации и отслеживания заявок, такой как система управления ИТ-услугами, на данный момент в организации не внедрено. Запросы, поступающие по различным каналам связи, фиксируются специалистами отдела АСУ с использованием разрозненных инструментов. Это могут быть записи в личных блокнотах или рабочих журналах, электронные таблицы Excel, применяемые для отслеживания некоторых категорий задач, сохраненные электронные письма или задачи, зарегистрированные в системе Битрикс24. Подобный подход приводит к фрагментации информации, затрудняет получение целостной картины по всем обращениям и усложняет контроль за их выполнением. Процессы приоритизации заявок и назначения ответственного исполнителя происходят, как правило, неформально, основываясь на усмотрении специалиста, принявшего запрос, или решении начальника отдела АСУ.

Формализованная, централизованная и легкодоступная база знаний для пользователей или самих специалистов службы поддержки в организации отсутствует. Знания, необходимые для решения типовых проблем, существуют в виде разрозненных инструкций и документов, хранящихся на сетевых дисках или локальных компьютерах специалистов. Также информация содержится в личных записях сотрудников отдела АСУ и основывается на накопленном опыте отдельных специалистов, который передается преимущественно неформальным путем в процессе устного общения. Поиск необходимой информации для решения конкретной проблемы часто требует опроса коллег или ручного поиска по файлам и документам, что занимает значительное время и снижает оперативность реагирования. Актуальность и полнота существующих инструкций не всегда гарантированы из-за отсутствия централизованного управления контентом.

Характер обращений пользователей показывает ряд типичных категорий проблем. К ним относятся затруднения при работе с программным обеспечением, таким как Битрикс24, Word и Excel. Возникают вопросы, связанные с доступом к сетевым ресурсам, общим папкам и корпоративным порталам, а также

с подключением к сети Интернет. Распространены обращения по поводу неисправностей периферийной техники – принтеров, сканеров и других устройств. Пользователи также ищут руководства по установке нового или обновлению существующего программного обеспечения.

### **1.3 Выявление проблем и потребностей в интеллектуальной системе**

Существующая в ООО «Амурагроцентр» организация процессов оказания технической поддержки обладает рядом недостатков, которые негативно сказываются на эффективности работы как ИТ-отдела, так и пользователей, обращающихся за помощью. Отмечается высокая трудоемкость и низкая скорость обработки поступающих запросов. Существующие проблемы во многом обусловлены ручным характером регистрации обращений, а также необходимостью осуществлять поиск решений в разрозненных и несистематизированных источниках информации.

Отсутствие централизованной базы знаний осложняет поиск ответов на возникающие вопросы. Эффективность процесса поиска информации снижается из-за того, что поисковой запроса пользователя должен обладать конкретностью и точностью, что не всегда возможно, когда проблема описывается нетехническим языком.

Нехватка единой системы для регистрации заявок усложняет процесс контроля за их исполнением и создает риск потери части обращений. В сочетании с неразвитой базой знаний это приводит к нерациональному использованию рабочего времени специалистов, которые вынуждены повторно заниматься решением уже известных типовых проблем. Как следствие, существующий подход не обеспечивает возможности для эффективной оценки загрузки сотрудников службы поддержки, выявления наиболее частых и критически важных инцидентов, мониторинга сроков выполнения заявок и измерения степени удовлетворенности пользователей качеством оказываемой поддержки. Значительная доля времени ИТ-специалистов расходуется на решение рутинных, повторяющихся задач и поиск необходимой информации, отвлекая их от более сложных инцидентов и стратегических аспектов развития ИТ-инфраструктуры предприятия.

Обозначенные проблемы формируют перечень конкретных потребностей ООО «Амурагроцентр», на удовлетворение которых должна быть направлена разрабатываемая интеллектуальная система поддержки. Ключевой потребностью является централизация и автоматизация всего цикла приема и обработки пользовательских запросов, что предполагает создание единой точки входа и механизмов автоматической маршрутизации и первичной обработки заявок.

Не менее важным является формирование единой, хорошо структурированной и легкодоступной базы знаний, обеспечивающей эффективное накопление, систематизацию и оперативный поиск решений для типовых проблем и инструкций. Особое значение имеет реализация интеллектуального поиска информации, способного корректно интерпретировать запросы пользователей, сформулированные на естественном языке, и предоставлять наиболее релевантные статьи из базы знаний, даже при отсутствии в запросе точных ключевых слов или устоявшихся формулировок.

Кроме того, существует потребность в автоматизации процессов управления контентом самой базы знаний, включая инструменты для автоматического генерирования кратких аннотаций к статьям или предложения релевантных тегов для их категоризации. В конечном итоге, разрабатываемая система призвана снизить нагрузку на ИТ-специалистов, предоставляя пользователям действенные инструменты для самостоятельного разрешения возникающих проблем.

## 2 ОБОСНОВАНИЕ НЕОБХОДИМОСТИ РАЗРАБОТКИ

### 2.1 Анализ актуальности разработки интеллектуальной системы

Как было установлено ранее, существующие процессы оказания технической поддержки в ООО «Амурагроцентр» обладают рядом существенных недостатков, которые ограничивают их общую эффективность. В условиях постоянно возрастающей зависимости бизнес-процессов от стабильности и производительности ИТ-инфраструктуры, а также непрерывного увеличения объема цифровой информации, традиционные подходы к управлению запросами пользователей и корпоративными знаниями становятся недостаточными. Актуальность разработки именно интеллектуальной системы поддержки обусловлена действием нескольких ключевых факторов современного технологического и бизнес-контекста.

Прежде всего, в современных условиях существенно возросли требования пользователей к скорости и качеству предоставляемой технической поддержки. Сотрудники ожидают не только быстрого, но и точного решения возникающих у них проблем. Интеллектуальные системы, обладающие способностью понимать запросы, сформулированные на естественном языке, и оперативно находить релевантную информацию в базе знаний, могут значительно сократить время ответа и повысить общее качество обслуживания. Параллельно с этим наблюдается непрерывный рост объема корпоративных знаний, таких как инструкции, руководства, описания решений распространенных проблем. Этот рост делает традиционные методы поиска по ключевым словам все менее эффективными при работе с большими и зачастую слабо структурированными массивами данных. В данном контексте семантический поиск, основанный на понимании смыслового содержания запроса, а не только на лексических совпадениях, представляется гораздо более перспективным и эффективным подходом.

Разработка системы поддержки в значительной степени обусловлена существенным прогрессом, достигнутым в области технологий искусственного интеллекта (ИИ) в последние годы. Технологии обработки естественного языка (ОЕЯ)

и большие языковые модели (БЯМ) достигли уровня технологической зрелости, позволяющего эффективно применять их для решения широкого круга практических задач, включая анализ текстов, семантический поиск информации, автоматическую генерацию резюме и аннотаций, а также классификацию контента.

Доступность внешних интерфейсов прикладного программирования (API) от ведущих технологических компаний, таких как Яндекс, Сбер, NVIDIA, Cloudflare и других, представляет собой значительное преимущество для организаций, не обладающих собственными ресурсами для разработки сложных ИИ-моделей, поскольку использование таких API существенно упрощает интеграцию передовых ИИ-возможностей в корпоративные системы.

Наряду с этим, отмечается возрастающая потребность в автоматизации рутинных задач, связанных с процессами технической поддержки и управления знаниями. Интеллектуальные функции, например, автоматическая генерация кратких описаний для статей базы знаний или предложение релевантных тегов для их категоризации, способны существенно снизить трудозатраты ИТ-специалистов на ведение и актуализацию базы знаний. Это, в свою очередь, позволяет высвободить их время для решения более сложных технических проблем и выполнения стратегически важных задач по развитию ИТ-инфраструктуры предприятия.

Не менее значимым является аспект повышения самостоятельности пользователей при решении возникающих проблем. Предоставление им инструмента для интеллектуального поиска, способного понимать запросы, сформулированные на естественном языке, существенно повышает вероятность самостоятельного нахождения необходимого решения или инструкции в базе знаний, минимизируя необходимость прямого обращения в службу поддержки по простым или типовым вопросам.

Вследствие этого, разработка интеллектуальной системы поддержки для ООО «Амурагроцентр» рассматривается не как простое усовершенствование существующей системы обработки заявок, а как необходимый этап адаптации предприятия к современным требованиям эффективности, качества

обслуживания и управления корпоративными знаниями. Интеграция технологий искусственного интеллекта позволит создать систему, которая будет не только регистрировать и отслеживать запросы, но и активно содействовать в их решении, а также автоматизировать часть процессов управления контентом базы знаний.

## **2.2 Обзор и анализ методов интеллектуализации систем поддержки**

Интеллектуализация систем поддержки пользователей может быть достигнута за счет применения различных методов и технологий ИИ. В контексте разрабатываемой системы для ООО «Амурагроцентр» ключевыми являются методы, направленные на улучшение поиска информации и автоматизацию управления контентом корпоративной базы знаний.

### **2.2.1 Использование технологий обработки естественного языка и больших языковых моделей**

Технологии ОЕЯ и БЯМ предоставляют мощные инструменты для анализа и понимания текстовой информации, что является критически важным для эффективной работы систем поддержки. Ключевым аспектом применения данных технологий выступает улучшение понимания запросов пользователей. Методы ОЕЯ позволяют проводить лингвистический анализ текста запроса, выделять ключевые сущности, такие как название программного обеспечения, тип ошибки или наименование оборудования, определять основное намерение пользователя и преобразовывать исходный запрос в форму, более пригодную для последующего поиска в базе данных. Большие языковые модели способны выполнять эти задачи на более высоком семантическом уровне, понимая контекст запроса, даже если запрос сформулирован неточно, с использованием разговорной лексики или содержит грамматические ошибки.

Семантический поиск представляет собой значительное усовершенствование по сравнению с традиционным поиском по ключевым словам. В отличие от последнего, ориентированного на поиск документов, содержащих точные словесные совпадения с запросом, семантический поиск нацелен на нахождение документов, близких по смыслу к запросу пользователя. Это достигается за счет использования векторных представлений текста, которые кодируют

семантическое значение слов, фраз и целых текстов в виде числовых векторов в многомерном пространстве. Большие языковые модели могут генерировать такие векторные представления как для запросов пользователей, так и для статей базы знаний. Таким образом, поиск релевантных статей сводится к нахождению векторов статей, наиболее близких к вектору запроса, с использованием метрик сходства, например, косинусного расстояния.

Важным применением БЯМ в контексте систем поддержки также является автоматическое резюмирование текстов. Современные модели способны автоматически генерировать краткое содержание, такое как резюме или аннотация, для объемных текстов, например, для статей базы знаний или технических инструкций. Автоматическое тегирование на основе анализа содержания статьи предлагает список релевантных ключевых слов или тегов, что существенно упрощает процесс категоризации информации в базе знаний и улучшает возможности последующего поиска и фильтрации статей по тегам.

### 2.2.2 Описание концепции многостадийного ИИ-поиска

Для повышения точности и релевантности поиска информации в разрабатываемой системе предложена концепция многостадийного интеллектуального поиска, которая использует возможности БЯМ на нескольких последовательных этапах обработки запроса.

Первый этап многостадийного поиска включает анализ запроса пользователя и выбор релевантных тегов. Большой языковой модели передается на вход исходный запрос пользователя, сформулированный на естественном языке, а также список всех тегов, доступных в системе базы знаний. Задача ИИ на данном этапе заключается в анализе семантики запроса и выборе из предложенного списка тех тегов, которые наиболее точно соответствуют тематике запроса пользователя, что позволяет эффективно сузить область первоначального поиска.

На втором этапе осуществляется непосредственный поиск статей по тегам, выбранным на предыдущем этапе. Система выполняет стандартный поиск статей в базе данных, используя идентификаторы отобранных тегов, в результате чего формируется первичный список потенциально релевантных статей.

Третья стадия предусматривает семантический анализ кратких содержаний статей, найденных на втором этапе. Большой языковой модели предоставляется исходный запрос пользователя и краткие содержания статей из первичного списка. Задача ИИ на этом этапе состоит в анализе семантической близости каждого краткого содержания к запросу пользователя и оценке вероятности того, что данная статья содержит ответ на вопрос. Это позволяет эффективно отфильтровать менее релевантные статьи из первичного списка, основываясь не только на совпадении тегов, но и на смысловом соответствии контента.

Завершающий, четвертый этап, состоит в формировании подробного ответа или окончательном ранжировании статей. На основе анализа полного текста статей, отобранных на третьей стадии, или их более подробных резюме, БЯМ может либо сформировать сводный, синтезированный ответ на вопрос пользователя, объединяя информацию из нескольких наиболее релевантных источников, либо предоставить пользователю окончательно отранжированный список наиболее подходящих статей. Такой многостадийный подход позволяет последовательно уточнять область поиска и повышать релевантность конечных результатов, эффективно используя сильные стороны БЯМ на каждом этапе.

### **2.3 Анализ и выбор внешних ИИ API**

Ключевым элементом для реализации интеллектуальных функций разрабатываемой системы является использование внешних API для выполнения задач ОЕЯ и работы с БЯМ. В рамках проекта были рассмотрены и проанализированы бесплатные API, предоставляемые компаниями NVIDIA и Cloudflare.

#### **2.3.1 Сравнение API NVIDIA и Cloudflare**

API компании NVIDIA предоставляет доступ к широкому спектру современных БЯМ, включая популярные модели семейства Llama и собственные разработки NVIDIA, например, Nemotron, что позволяет гибко подходить к выбору модели для конкретной задачи. Данная система поддерживает детальную настройку параметров генерации, таких как температура, top-p и другие, обеспечивая контроль над характером ответа модели. Производительность и задержки при работе с API могут варьироваться, а модель ценообразования NVIDIA API

обычно основана на количестве обработанных токенов, при этом для коммерческого использования может потребоваться платный тариф. Интеграция с API осуществляется через стандартный REST интерфейс, для использования которого необходимо получить API-ключ.

В свою очередь, сервис Cloudflare предлагает доступ к набору популярных языковых моделей с открытым исходным кодом, таких как Llama, Mistral, Gemma, развернутых на глобальной инфраструктуре Cloudflare. Список доступных моделей может быть несколько более ограниченным по сравнению с NVIDIA, но включает эффективные модели для решения различных задач ОЕЯ. Использование распределенной инфраструктуры Cloudflare потенциально обеспечивает низкие задержки. В отношении стоимости Cloudflare предлагает достаточно обширный бесплатный уровень использования и конкурентоспособные цены на платное использование, где ценообразование также может основываться на количестве выполненных запросов или обработанных токенов. Интеграция осуществляется через Cloudflare Workers или AI Gateway, что требует предварительной настройки в панели управления Cloudflare, при этом существенным преимуществом является удобство интеграции для организаций, уже использующих инфраструктуру данной компании.

Принятие решения о выборе конкретного API для проекта основывалось на стремлении к гибкости. Модуль интеграции спроектирован таким образом, чтобы поддерживать работу с обоими провайдерами, позволяя переключаться между ними через конфигурационный файл. Такой подход обеспечивает возможность использования бесплатных уровней обоих сервисов, а также выбора наиболее производительного или экономически выгодного варианта в зависимости от текущих потребностей.

### 2.3.2 Принципы построения запросов

Эффективность работы БЯМ во многом зависит от качества и структуры подаваемых на вход запросов. Проектирование системы включало определение и применение ключевых принципов формирования запросов к ИИ-моделям.

Четкая постановка задачи имеет первостепенное значение: промпт должен ясно указывать, какое действие требуется от модели, например, «Создай краткое содержание для следующей статьи», «Сгенерируй 5 релевантных тегов для текста» или «Оцени релевантность следующих статей к данному запросу». Предоставление достаточного контекста также является важным аспектом; для задач, требующих анализа существующего текста, в промпт необходимо включать сам текст или его релевантные части. Структурирование входной информации с использованием разделителей и маркеров, например, «Заголовок», «Содержание», «Запрос пользователя», помогает модели лучше понять структуру данных. В некоторых случаях полезно явно указать формат ожидаемого ответа, например, «Верни список тегов, разделенных запятыми». Следует отметить, что создание эффективных промптов часто требует экспериментов и итеративной доработки.

В рамках проекта были разработаны шаблоны промптов для различных задач. Так, для генерации резюме используется формат, предписывающий создание очень краткого содержания статьи в 3-5 предложениях, с указанием заголовка статьи (представленного переменной {title}) и содержания статьи (переменной {content}). При генерации тегов применяется шаблон, в котором ставится задача сгенерировать релевантные теги для статьи, также с учетом ее заголовка и содержания, и дополнительно передается список существующих тегов в базе данных (переменная {tags\_list}), что способствует генерации более консистентных категорий. Для первой стадии ИИ-поиска используется промпт, направленный на анализ запроса пользователя (переменная {query}) и выбор наиболее релевантных тегов из списка доступных тегов (переменная {tags\_list}).

## **2.4 Анализ рынка на наличие схожих решений**

Рынок программных решений для поддержки пользователей и управления корпоративными знаниями предлагает множество продуктов. Часть из них уже интегрирует элементы ИИ. В рамках исследования были проанализированы несколько основных категорий таких систем.

Первая категория – это комплексные платформы управления ИТ-услугами и системы обработки заявок. К таким системам относятся Jira Service

Management, ServiceNow, Zendesk. Эти платформы часто включают модули баз знаний и начинают внедрять ИИ-функции, такие как чат-боты, автоматическая классификация заявок, предложение релевантных статей при создании заявки. Однако такие системы характеризуются высокой стоимостью, избыточной функциональностью для задач ООО «Амурагроцентр», а также сложностью внедрения и настройки.

Вторая категория – это специализированные системы управления знаниями. Примерами являются Confluence и Notion. Эти платформы ориентированы на создание и организацию контента. Некоторые, как Notion AI, активно внедряют БЯМ для генерации текста, резюмирования, ответов на вопросы на основе контента базы знаний. Преимуществами являются гибкость в организации контента и мощные ИИ-функции. Недостатками можно считать отсутствие доступа у пользователей из России.

Третья категория – это решения с открытым исходным кодом. Примерами являются Zammad или FreeScout. Это системы обработки заявок, которые могут быть расширены плагинами, в том числе для интеграции с ИИ. Их преимущества – бесплатность основного ПО и возможность доработки. Недостатками являются необходимость значительных усилий на настройку, интеграцию ИИ-функций и поддержку. Готовые ИИ-интеграции могут быть ограничены.

При сравнении с проектируемым решением, разрабатываемая система для ООО «Амурагроцентр» занимает нишу между простыми Wiki и комплексными коммерческими платформами. Она обладает рядом уникальных особенностей. Система характеризуется фокусом на конкретные ИИ-функции, такие как многостадийный ИИ-поиск, генерация резюме и тегов, используя внешние API. Отличительными чертами являются легковесность и простота. Важным преимуществом является возможность кастомизации, так как система может быть легко адаптирована под специфические нужды ООО «Амурагроцентр». Также следует отметить гибкость в выборе API, поскольку возможность переключения между провайдерами ИИ API дает контроль над затратами и доступом к технологиям.

В то время как коммерческие решения могут предлагать более широкий спектр функций, разрабатываемая система обеспечивает необходимый интеллектуальный функционал при меньшей сложности и потенциально меньших затратах, будучи точно настроенной под задачи предприятия.

## **2.5 Определение и утверждение требований к системе**

На основе анализа предметной области, выявленных потребностей и выбранных технологических подходов были сформулированы следующие требования к интеллектуальной системе поддержки пользователей ООО «Амурагроцентр».

Функциональные требования включают, во-первых, управление статьями. Система должна позволять пользователям с ролью «admin» или «editor» создавать, просматривать, редактировать и удалять статьи базы знаний. Редактор статей должен поддерживать форматирование текста с использованием синтаксиса Markdown. При отображении статьи Markdown-разметка должна корректно преобразовываться в HTML. Для каждой статьи должны храниться заголовок, полное содержание, дата создания и последнего обновления.

Во-вторых, требуется управление тегами. Система должна позволять при создании или редактировании статьи указывать список связанных тегов и обеспечивать связь «многие-ко-многим» между статьями и тегами. Пользователи должны иметь возможность просматривать все теги, используемые в системе, а также просматривать все статьи, связанные с конкретным тегом. Администраторы должны иметь возможность удалять теги, не связанные ни с одной статьей.

В-третьих, необходимо управление файлами. Пользователи с ролью «admin» или «editor» должны иметь возможность загружать файлы и прикреплять их к статьям. Пользователи должны иметь возможность скачивать файлы, прикрепленные к статьям. При удалении статьи связанные с ней файлы также должны удаляться.

В-четвертых, система должна осуществлять управление пользователями. Она должна поддерживать аутентификацию пользователей по логину и паролю. Пароли пользователей должны храниться в безопасном виде, то есть путем

хеширования с солью. Необходима поддержка как минимум двух ролей, а именно «admin» с полным доступом и «editor» с доступом к созданию и редактированию контента. Роль «client», подразумевающая только чтение, предназначена для неаутентифицированных пользователей. Администраторы должны иметь возможность создавать, редактировать и удалять учетные записи пользователей. Доступ к функциям должен быть ограничен в соответствии с ролями.

В-пятых, должен быть реализован поиск. Система должна предоставлять стандартный поиск по ключевым словам. Необходима реализация многостадийного интеллектуального поиска (ИИ-поиска), который принимает запрос пользователя на естественном языке, использует внешнее ИИ API для анализа запроса и выбора релевантных тегов, выполняет поиск статей по выбранным, использует внешнее ИИ API для анализа кратких содержаний найденных статей и их ранжирования по релевантности к запросу и представляет пользователю отранжированный список наиболее релевантных статей.

В-шестых, требуется интеллектуальная генерация контента. При создании или обновлении статьи система должна автоматически, по запросу пользователя-редактора, генерировать краткое резюме статьи с использованием внешнего ИИ API. Сгенерированное резюме должно сохраняться в базе данных. Также система должна автоматически, по запросу пользователя-редактора, генерировать список релевантных тегов с использованием внешнего ИИ API, предлагая их пользователю для добавления.

Нефункциональные требования к системе включают следующие аспекты. Производительность системы должна быть такова, что время отклика на стандартные операции не превышает 3 секунды. Время отклика на операции, использующие внешние ИИ API, не должно превышать 30 секунд в 90% случаев, без учета сетевых задержек до API. Безопасность системы подразумевает защиту от распространенных веб-уязвимостей, таких как CSRF и XSS. Должно быть реализовано безопасное хранение паролей и API-ключей. Доступ к функциям должен строго контролироваться на основе ролей. Надежность системы означает ее стабильное функционирование при штатной нагрузке и корректную обработку

ошибок, возникающих при взаимодействии с базой данных или внешними ИИ API. Должна быть предусмотрена возможность резервного копирования базы данных SQLite. Удобство использования предполагает интуитивно понятный и простой в освоении пользовательский интерфейс, а также логичную навигацию по сайту.

Детальные требования к системе оформлены в виде технического задания, которое приведено в приложении А.

## **2.6 Расчет экономической эффективности**

Оценка экономической эффективности внедрения разработанной интеллектуальной системы поддержки пользователей включает анализ предполагаемых затрат на ее создание и эксплуатацию, а также определение ожидаемого экономического эффекта от ее использования.

Рассмотрению подлежат затраты, связанные с разработкой программного продукта. Основной объем работ по созданию системы был реализован в процессе выполнения выпускной квалификационной работы. Как следствие отсутствие прямых денежных затрат на оплату труда разработчика в представленном экономическом расчете, поскольку выполнение работы осуществлялось в рамках образовательного процесса и не предполагало привлечения сторонних оплачиваемых специалистов или отвлечения штатного персонала от выполнения текущих производственных задач.

Следующей статьей расходов выступают затраты на использование внешних прикладных программных интерфейсов (API), необходимых для обеспечения функционирования интеллектуальных модулей системы. Архитектурная гибкость позволяет адаптировать систему под изменяющиеся условия рынка и новые технологии. Нейросетевой модуль обладает возможностями интеграции с разнообразными сервисами искусственного интеллекта, что предоставляет выбор между зарубежными платформами и отечественными аналогами. В качестве примера отечественного решения приведен YandexGPT API, стоимость использования которого в асинхронном режиме составляет 0,60 рубля за 1000 токенов, включая НДС.

При ориентировочной оценке умеренного объема использования, ежемесячные расходы на API могут достигнуть порядка 3000 рублей, что эквивалентно 36 000 рублей в год. Указанная сумма является предварительной и может изменяться в зависимости от реальной нагрузки на систему и выбранного провайдера API. Кроме того, имеется возможность использования сервисов Cloudflare Workers AI или NVIDIA в рамках предоставляемых бесплатных лимитов на начальных этапах эксплуатации, что потенциально способно снизить затраты до нулевого значения.

Затраты, связанные с развертыванием разработанной системы, в данном расчете не учитываются, так как предполагается задействование имеющейся серверной инфраструктуры предприятия. Такой подход исключает необходимость дополнительных капитальных инвестиций в приобретение нового оборудования.

Заключительным компонентом структуры затрат являются расходы на техническую поддержку и последующее сопровождение системы. Предполагается, что на выполнение задач администрирования, актуализацию базы знаний и осуществление мониторинга функционирования системы будет затрачиваться 4 часа рабочего времени квалифицированного специалиста в месяц. Исходя из ставки оплаты труда такого специалиста в размере 400 рублей в час, годовые издержки на поддержку и сопровождение оцениваются в 19 200 рублей (4 часа/месяц \* 400 руб./час \* 12 месяцев).

Таким образом, итоговые единовременные затраты на разработку и внедрение системы, с учетом бесплатного выполнения работ по программированию, стремятся к нулю. Итоговые ежегодные затраты на эксплуатацию будут варьироваться в зависимости от стоимости использования API и могут составлять от 19 200 рублей (в случае использования API в рамках бесплатного лимита) до 55 200 рублей (36 000 рублей за API + 19 200 рублей за поддержку).

Ожидаемый годовой экономический эффект от внедрения системы складывается из нескольких ключевых компонентов. Во-первых, прогнозируется существенная экономия времени сотрудников службы технической поддержки (отдела АСУ). Если предположить, что до внедрения системы два специалиста

тратили около 32 часов в месяц каждый на обработку типовых запросов, которые могут быть автоматизированы или решены с помощью интеллектуальной базы знаний, то при средней ставке специалиста в 600 рублей в час, годовые затраты на эту деятельность составляют 460 800 рублей (2 специалиста \* 32 часа/мес./чел. \* 12 мес. \* 600 руб./час). Ожидается, что внедрение интеллектуальной системы позволит сократить это время на 40%, что приведет к годовой экономии в размере 184 320 рублей (460 800 руб. \* 0.40).

Во-вторых, значимым фактором является экономия времени рядовых пользователей системы. Допустим, в организации насчитывается 50 пользователей, каждый из которых ранее тратил в среднем 1 час в месяц на ожидание ответа от технической поддержки или на самостоятельный поиск решений стандартных проблем. При средней часовой ставке пользователя в 350 рублей, годовые затраты времени пользователей на эти нужды оцениваются в 210 000 рублей (50 пользователей \* 1 час/мес./чел. \* 12 мес. \* 350 руб./час). Предполагается, что разработанная интеллектуальная система позволит ускорить поиск решений и сократить это время на 70%, что обеспечит годовую экономию времени пользователей в размере 147 000 рублей (210 000 руб. \* 0.70).

Помимо прямого экономического эффекта, внедрение системы принесет и качественные улучшения. К ним относятся повышение уровня удовлетворенности пользователей качеством и оперативностью технической поддержки, улучшение доступности и систематизации корпоративных знаний, снижение вероятности человеческих ошибок при решении типовых проблем, а также высвобождение времени высококвалифицированных ИТ-специалистов для концентрации на более сложных и стратегически важных задачах развития информационных систем предприятия.

Суммарный годовой экономический эффект от внедрения интеллектуальной системы поддержки пользователей составит 331 320 рублей (184 320 рублей от экономии времени специалистов поддержки + 147 000 рублей от экономии времени пользователей). Чистый годовой экономический эффект, рассчитанный как разница между суммарным годовым экономическим эффектом и

ежегодными затратами на эксплуатацию, будет находиться в диапазоне от 276 120 рублей (331 320 руб. - 55 200 руб.) до 312 120 рублей (331 320 руб. - 19 200 руб.).

Учитывая, что единовременные капитальные затраты на разработку системы сведены к нулю, традиционный расчет срока окупаемости инвестиций показывает, что проект является экономически выгодным с момента начала эксплуатации.

## 3 ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЕЙ

### 3.1 Разработка архитектуры программного решения

На основе сформулированных требований разработана архитектура интеллектуальной системы поддержки пользователей для ООО «Амурагроцентр». Она призвана обеспечить достижение поставленных целей в части функциональности, а также гарантировать гибкость, надежность и простоту последующего сопровождения разработанного программного продукта.

#### 3.1.1 Общее описание архитектуры системы

Программное решение основано на клиент-серверной модели взаимодействия. В качестве архитектурного паттерна был выбран Model-View-Controller (MVC), адаптированный для создания веб-приложений с использованием фреймворка Flask. Данный паттерн обеспечивает эффективное разделение ответственностей между компонентами системы, отвечающими за логику работы с данными (Model), их визуальное представление (View) и управление потоком обработки пользовательских запросов (Controller).

Компонент «Model» инкапсулирует логику взаимодействия с данными. Его реализация в рамках создаваемой системы предполагает использование прямых SQL-запросов к базе данных SQLite, осуществляемых посредством встроенного в язык Python модуля sqlite3.

Формирование пользовательского интерфейса и отображение информации выполняет компонент «View». Для его реализации применяется шаблонизатор Jinja2, интегрированный с фреймворком Flask, что позволяет динамически генерировать HTML-страницы на основе передаваемых данных.

Связующим звеном между моделью и представлением выступает компонент «Controller». Его функции включают обработку входящих HTTP-запросов, вызове соответствующих методов моделей для манипулирования данными и последующую передачу этих данных в представление для отображения.

Особенностью архитектуры является модуль интеграции с внешними API ИИ, такими как NVIDIA и Cloudflare. Этот модуль обеспечивает взаимодействие для выполнения задач ОЕЯ и работы с БЯМ, с возможностью переключения между провайдерами через конфигурацию.

Для реализации многостадийного ИИ-поиска разработан специализированный модуль, координирующий взаимодействие между базой данных, внешними API и пользовательским интерфейсом для последовательного выполнения этапов поиска.

Разработанная архитектура обеспечивает баланс между функциональностью, простотой развертывания и обслуживания, а также возможностью дальнейшего расширения системы. Структурная схема архитектуры представлена на рисунке 2.

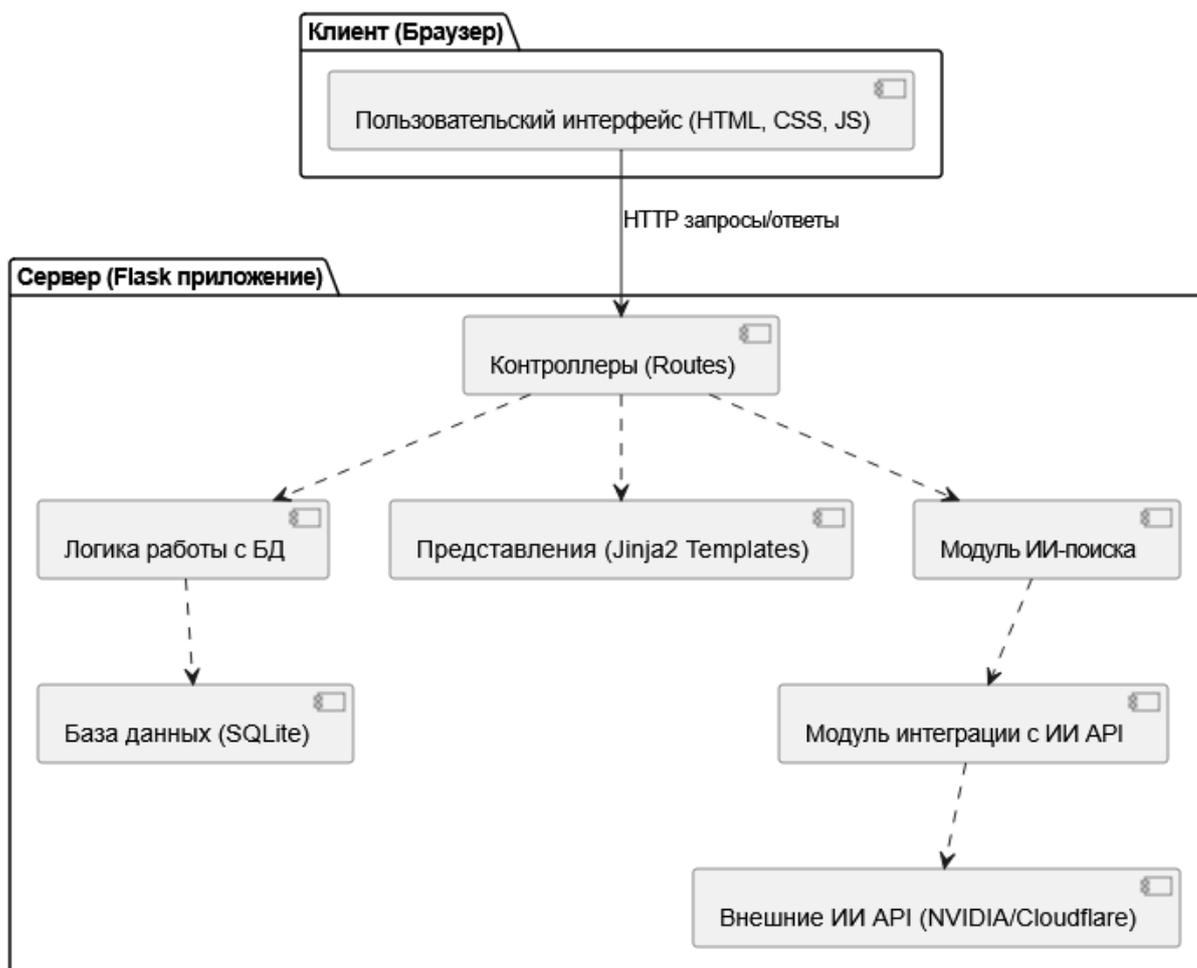


Рисунок 2 – Общая архитектура системы

### 3.1.2 Технологический стек

Для реализации архитектуры был выбран следующий технологический стек. В качестве языка программирования используется Python версии 3.9 и выше, что обусловлено его простотой и обширной экосистемой библиотек для веб-разработки и искусственного интеллекта [1]. Веб-фреймворком служит Flask, известный своей легковесностью, гибкостью и хорошей масштабируемостью для проектов различного уровня. В качестве системы управления базами данных выбрана SQLite, являющаяся встраиваемой СУБД, что упрощает развертывание и обслуживание и подходит для систем со средними нагрузками.

Фронтенд-часть системы реализуется с использованием HTML и CSS. Для обеспечения динамического обновления элементов страниц и выполнения AJAX-запросов используется нативный JS. Преобразование Markdown-разметки в HTML осуществляется с помощью регулярных выражений.

Взаимодействие с внешними API реализуется с использованием библиотеки Requests для выполнения HTTP-запросов и обработки JSON-ответов. Аутентификация пользователей и управление сессиями обеспечиваются расширением Flask-Login. Для работы с формами, их валидации и защиты от CSRF-атак применяется расширение Flask-WTF.

## **3.2 Проектирование модуля интеграции с внешними ИИ-сервисами**

Модуль интеграции с внешними ИИ-сервисами является ключевым компонентом, обеспечивающим интеллектуальные функции разрабатываемой системы. Он спроектирован таким образом, чтобы предоставлять унифицированный интерфейс для взаимодействия с различными API ИИ, абстрагируя детали реализации конкретных провайдеров и позволяя гибко переключаться между ними.

### 3.2.1 Архитектура модуля

Основным компонентом модуля интеграции является файл взаимодействия с API. Он содержит логику для взаимодействия с различными API ИИ. Центральная функция служит единой точкой входа для всех запросов к ИИ из

основного приложения. Эта функция инкапсулирует логику выбора конкретного провайдера ИИ на основе настроек, заданных в конфигурационном файле.

В разрабатываемой системе реализованы отдельные функции для вызова API каждого поддерживаемого провайдера. Эти функции отвечают за формирование HTTP-запросов в соответствии с требованиями конкретного API, включая установку необходимых заголовков, сериализацию данных запроса и отправку запроса. Они также обрабатывают ответы от API, извлекая полезные данные или информацию об ошибках.

Такой подход, где выбор конкретной реализации API инкапсулирован и управляется через конфигурацию, можно рассматривать как упрощенную реализацию паттерна «Стратегия», где центральная функция выступает в роли контекста, а взаимодействие с конкретными провайдерами – в роли конкретных стратегий. Это обеспечивает гибкость и упрощает добавление поддержки новых ИИ-провайдеров в будущем. Модуль также включает базовую обработку ошибок, возникающих при взаимодействии с API.

### 3.2.2 Проектирование промптов для ИИ-моделей

Качество взаимодействия с большими языковыми моделями (БЯМ) в значительной степени определяется качеством и структурой подаваемых на вход запросов, или промптов. При их проектировании для разрабатываемой системы были использованы следующие ключевые принципы.

Во-первых, это четкая постановка задачи: каждый промпт должен ясно и недвусмысленно указывать, какое действие или какой результат ожидается от модели. Для генерации краткого содержания статьи используется промпт, который включает указание на задачу, заголовок статьи и ее основное содержание.

Пример структуры промпта для этой задачи, реализованной в функции: «Задача: создать очень краткое содержание статьи в 3-5 предложений. Заголовок статьи: {заголовок}. Содержание статьи: {содержание}».

При формировании запроса (промпта) для генерации тегов к статье, помимо явной постановки задачи, в него включаются заголовок и содержание самой статьи. Дополнительно в контекст передается перечень тегов, уже

имеющихся в базе данных, а также специфицируются правила их генерации, которые могут включать инструкции по возврату результата, например, в виде тегов, перечисленных через запятую.

Все промпты подвергаются тщательной структуризации и тестированию на совместимость с API выбранных провайдеров, что является необходимым условием для обеспечения эффективного и предсказуемого взаимодействия с большими языковыми моделями.

### **3.3 Проектирование многостадийного ИИ-поиска**

Разработка многостадийного ИИ-поиска относится к созданию одной из ключевых интеллектуальных функций системы, которая предназначена для эффективного нахождения релевантной информации в базе знаний по запросам пользователей, сформулированным на естественном языке.

#### **3.3.1 Общий алгоритм многостадийного ИИ-поиска**

Процесс многостадийного ИИ-поиска основан на последовательном уточнении и фильтрации результатов на нескольких этапах, с использованием возможностей внешних ИИ API.

Первоначальный этап включает анализ запроса пользователя и выбор релевантных тегов. Исходный текстовый запрос пользователя вместе с полным списком всех тегов передается на обработку. Задача языковой модели на данном этапе заключается в анализе пользовательского запроса и выборе из предоставленного списка тех тегов, которые наиболее точно соответствуют тематике запроса. Выполнение этого шага позволяет сузить область первоначального поиска в базе данных. В случае если запрос не требует поиска по базе данных, нейросеть возвращает ответ пользователю, пропуская следующие этапы.

Второй этап включает поиск статей по тегам, отобранным на предыдущей стадии. Система выполняет SQL-запрос к базе данных для извлечения всех статей, связанных хотя бы с одним из выбранных тегов. Формируется набор релевантных статей.

Третий этап включает семантический анализ найденных статей. ИИ анализирует запрос пользователя и краткие содержания статей, выбирая из них

наиболее семантически близкие к запросу. Если статей мало (например, до трех), они могут быть выбраны все без дополнительного обращения к ИИ. Результатом этапа является уточненный список ID статей.

Четвертый этап – это представление результатов пользователю. Система извлекает полные тексты отобранных статей из базы данных. Затем, с помощью ИИ API, генерируется развернутый ответ на запрос пользователя, основанный на содержании этих статей и контексте диалога. Если на предыдущих этапах статьи не были выбраны (или ИИ на первом этапе решил, что они не нужны), ответ генерируется на основе общих знаний ИИ и истории чата. Конечный результат включает сгенерированный ИИ текстовый ответ и список статей, которые были использованы для его формирования (если таковые были).

### 3.3.2 Проектирование интерфейса поиска

Интерфейс ИИ-поиска представлен в виде чат-бота на главной странице приложения. Этот интерфейс спроектирован для интуитивного взаимодействия пользователя с системой.

Ключевым элементом является поле для ввода текстового запроса и кнопка для его отправки. История диалога отображается в специальном контейнере, где сообщения пользователя и ответы системы (ИИ-ассистента) визуально разделены.

Во время обработки запроса, для информирования пользователя о том, что система работает, отображается индикатор загрузки. После завершения всех этапов поиска, сгенерированный ИИ текстовый ответ добавляется в чат как сообщение от ассистента. Если ИИ использовал статьи из базы знаний для формирования ответа, их заголовки в виде ссылок на полные тексты статей выводятся под основным ответом в компактном виде.

## 3.4 Проектирование модуля управления статьями и тегами

Модуль управления статьями и тегами формирует ядро контентной части системы. Он обеспечивает создание, редактирование и организацию информационных материалов базы знаний, а также интегрирован с ИИ для автоматизации некоторых задач.

### 3.4.1 Архитектура модуля управления контентом

Управление контентом реализовано через набор функций-обработчиков маршрутов, которые отвечают за операции создания, чтения, обновления и удаления статей и тегов.

Создание и редактирование статей: при сохранении статьи происходит обработка ее заголовка, основного содержания и краткого содержания. Также обрабатывается строка тегов: теги разделяются, и для каждого тега проверяется его наличие в базе. При редактировании старые связи с тегами сначала удаляются, а затем создаются новые.

Управление файлами: при создании и редактировании статей предусмотрена загрузка файлов. Файлы сохраняются в директорию, указанную в конфигурационном файле, с уникальными именами. При редактировании статьи отображается список существующих вложений с возможностью их удаления. При удалении статьи все связанные с ней файлы также удаляются с сервера и из базы данных.

Интеграция с ИИ: на страницах создания и редактирования статей присутствуют кнопки «Сгенерировать краткое содержание» и «Автогенерация тегов». Эти кнопки инициируют обращения к ИИ API для получения сгенерированного текста. Полученные данные затем отображаются в соответствующих полях формы.

Удаление статей: обрабатывает удаление статьи, связанных с ней записей, а также физическое удаление файлов.

Управление тегами: помимо создания тегов при добавлении статей, система также включает функционал просмотра всех тегов с указанием количества связанных статей.

### 3.4.2 Проектирование интерфейса редактора статей

Интерфейс редактора статей спроектирован с целью обеспечения удобства и эффективности для пользователей, отвечающих за наполнение и актуализацию базы знаний.

Структурно интерфейс редактора представляет собой единую веб-страницу, логически разделенную на несколько секций для ввода различной информации о статье.

Основные поля для ввода данных включают поле для заголовка статьи, которое является обязательным для заполнения. Главным элементом является основное поле для редактирования содержания статьи. Секция управления тегами позволяет пользователю вводить теги, связанные со статьей.

В интерфейс редактора интегрирован ИИ-ассистент. Данный функционал представлен отдельной секцией с кнопками, такими как «Сгенерировать краткое содержание» и «Автогенерация тегов». При нажатии на эти кнопки система обращается к ИИ API для автоматической генерации соответствующего контента.

Секция управления вложениями содержит кнопку «Прикрепить файлы», позволяющую загружать файлы с компьютера пользователя. Поддерживается загрузка нескольких файлов одновременно.

Основная кнопка «Сохранить» отправляет форму на сервер. Также присутствует кнопка «Отмена», возвращающая пользователя на предыдущую страницу или на главную страницу.

### **3.5 Проектирование базы данных**

Проектирование базы данных является фундаментальным этапом разработки системы, определяющим структуру хранения данных и взаимосвязи между ними. Процесс проектирования включает три основных уровня: инфологический, логический и физический.

#### **3.5.1 Инфологическое проектирование**

На этапе инфологического проектирования определяются основные сущности предметной области и связи между ними.

##### **3.5.1.1 Формулировка сущностей**

На основе анализа требований к системе и ее функциональности были выделены следующие ключевые сущности:

– «Статьи (articles)» основная информационная единица системы, представляющая собой запись в базе знаний. Каждая статья содержит заголовок, основное содержание, опциональное краткое резюме и дату создания;

– «Теги (tags)» ключевые слова или фразы, используемые для категоризации и организации статей. Каждый тег имеет уникальное имя;

– «Пользователи (users)» учетные записи лиц, имеющих доступ к системе с различными уровнями прав. Каждый пользователь имеет уникальное имя (логин) и пароль;

– «Прикрепленные файлы (file\_attachments)» файлы различных форматов (документы, изображения), которые могут быть присоединены к статьям для предоставления дополнительной информации;

– «Связи статей и тегов (article\_tags)» сущность, реализующая связь «многие-ко-многим» между статьями и тегами, поскольку одна статья может иметь несколько тегов, и один тег может быть присвоен нескольким статьям.

### 3.5.1.2 Описание атрибутов

Для каждой выделенной сущности определены необходимые атрибуты, их характеристики и примеры значений, которые представлены в таблицах 1-5.

Таблица 1 – Спецификация атрибутов сущности «Статьи (articles)»

Название атрибута	Описание атрибута	Единица измерения	Диапазон значений	Пример атрибута
Код статьи (Id)	Уникальный идентификатор статьи	-	Целое, >0	101
Заголовок (Title)	Заголовок статьи	-	Текст, до 255 симв.	«Настройка Wi-Fi»
Содержимое (Content)	Полное содержание статьи (Markdown)	-	Текст	«Для подключения...»
Краткое содержание (Summary)	Краткое резюме статьи (сгенерировано ИИ или вручную)	-	Текст (опц.)	«Статья описывает...»

Дата создания (Created_at)	Дата и время создания/последнего обновления статьи	Дата/Время	-	2025-05-20 14:30:00
----------------------------	--	------------	---	------------------------

Таблица 2 – Спецификация атрибутов сущности «Теги (tags)»

Название атрибута	Описание атрибута	Единица измерения	Диапазон значений	Пример атрибута
Код тега (Id)	Уникальный идентификатор тега	-	Целое, >0	55
Название (Name)	Имя тега (уникальное)	-	Текст, до 50 симв.	«Сеть»

Таблица 3 – Спецификация атрибутов сущности «Пользователи (users)»

Название атрибута	Описание атрибута	Единица измерения	Диапазон значений	Пример атрибута
Код пользователя (Id)	Уникальный идентификатор пользователя	-	Целое, >0	1
Имя пользователя (Username)	Логин пользователя (уникальный)	-	Текст, до 50 симв.	«ivan»
Пароль (Password)	пароль	-	Текст	«Сложныйпароль123»
Роль (Role)	Роль пользователя в системе	-	Текст	«editor»

Таблица 4 – Спецификация атрибутов сущности «Прикрепленные файлы (file\_attachments)»

Название атрибута	Описание атрибута	Единица измерения	Диапазон значений	Пример атрибута
Код файла (Id)	Уникальный идентификатор файла	-	Целое, >0	203

## Продолжение таблицы 4

Код статьи (Article_id)	Идентификатор статьи, к которой прикреплен файл	-	Целое, >0	101
Оригинальное имя файла (Original_file- name)	Оригинальное имя файла, загруженного пользователем	-	Текст, до 255 симв.	«Инструк- ция.pdf»
Путь к файлу (Filepath)	Уникальное имя файла на сервере для хранения	-	Текст, до 255 симв.	«uuid_Ин- струк- ция.pdf»
Тип файла (Filetype)	Тип файла	-	Текст, до 50 симв.	«docx»
Размер файла (Filesize)	Размер файла в байтах	байты	Целое, >0	1572864
Дата загрузки (Uploaded_at)	Дата и время загрузки файла	Дата/Время	-	2025-05-20 14:35:00

Таблица 5 – Спецификация атрибутов сущности «Связи статей и тегов  
(article\_tags)»

Название атрибута	Описание атрибута	Единица измерения	Диапазон значений	Пример атрибута
Код статьи (Article_id)	Идентификатор статьи	-	Целое, >0	101
Код тега (Tag_id)	Идентификатор тега	-	Целое, >0	55

### 3.5.1.3 Определение связей между сущностями

На основе анализа предметной области и сформулированных сущностей определены следующие связи:

– Между сущностями «Статьи» и «Теги» существует связь типа «многие-ко-многим». Одна статья может быть помечена несколькими тегами, и один тег может применяться к нескольким статьям. Эта связь реализуется через промежуточную сущность «Связи статей и тегов».

– Между сущностью «Статьи» и сущностью «Прикрепленные файлы» существует связь типа «один-ко-многим». К одной статье может быть прикреплено несколько файлов, но каждый файл относится только к одной статье.

Сущность «Пользователи» напрямую не связана с сущностью «Статьи» на уровне внешних ключей, однако логическая связь подразумевается через права доступа на создание и редактирование статей. Концептуальная инфологическая модель базы данных, отображающая сущности и связи между ними, представлена на рисунке 3.

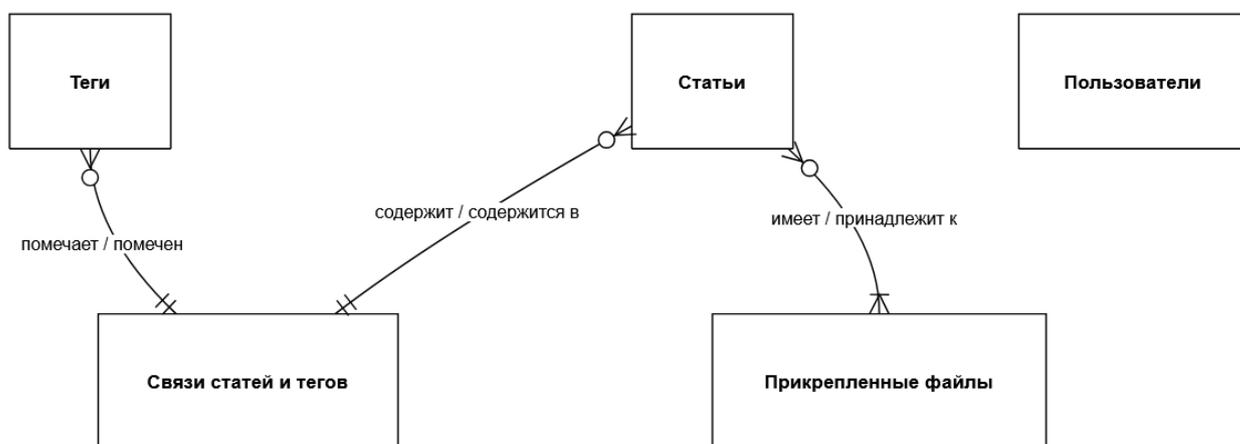


Рисунок 3 – Концептуальная инфологическая модель базы данных

### 3.5.2 Логическое проектирование

На этапе логического проектирования инфологическая модель преобразуется в реляционную модель данных, которая описывает структуру таблиц, их атрибуты, первичные и внешние ключи. Для реализации связи «многие-ко-

многим» между статьями и тегами создается промежуточная таблица связи статей и тегов. Эта таблица будет содержать два внешних ключа: код статьи, ссылающийся на первичный ключ таблицы статьи, и код тега, ссылающийся на первичный ключ таблицы теги. Комбинация этих двух ключей образует составной первичный ключ для таблицы связи статей и тегов. Таблица прикрепленные файлы связана с таблицей статьи отношением «один-ко-многим» через внешний ключ код статьи в таблице прикрепленные файлы, который ссылается на первичный ключ код статьи таблицы статьи.

Все проектируемые таблицы находятся как минимум в первой нормальной форме (1НФ), так как все их атрибуты являются атомарными. Таблицы статьи, теги, пользователи, прикрепленные файлы также находятся во второй нормальной форме (2НФ), поскольку все их неключевые атрибуты полностью функционально зависят от их простого первичного ключа. Таблица связи статей и тегов также находится во 2НФ, так как у нее отсутствуют неключевые атрибуты. Все таблицы удовлетворяют требованиям и третьей нормальной формы (3НФ), поскольку в них отсутствуют зависимости неключевых атрибутов от других неключевых атрибутов. Логическая схема данных представлена на рисунке 4.

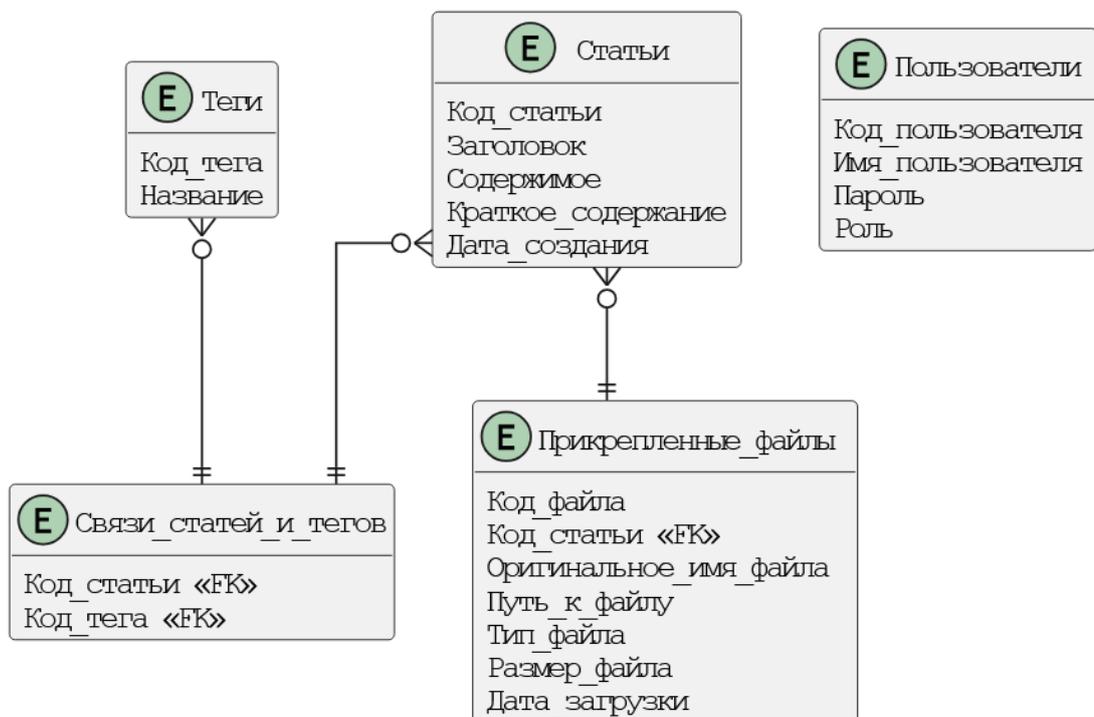


Рисунок 4 – Логическая схема данных

### 3.5.3 Физическое проектирование

База данных для разрабатываемой интеллектуальной системы поддержки пользователей реализована с использованием SQLite. SQLite представляет собой встраиваемую, файловую систему управления базами данных, что означает отсутствие необходимости в отдельном серверном процессе и упрощает развертывание приложения. Выбор данной СУБД обусловлен ее простотой, легковесностью, достаточной производительностью для предполагаемой нагрузки системы, а также удобством интеграции с веб-фреймворком Flask, используемым в проекте. Спроектированные таблицы базы данных, их структура и физические характеристики адаптированы под особенности SQLite и представлены ниже.

Таблица 6 – Структура таблицы «Статьи»

Наименование поля	Тип данных	Допустимость NULL	Индексация	Примечание
Код статьи (Id)	INTEGER	Нет	Да	Первичный ключ, автоинкремент
Заголовок (Title)	TEXT	Нет	Нет	Заголовок статьи
Содержание (Content)	TEXT	Нет	Нет	Полное содержание статьи в формате Markdown
Краткое содержание (Summary)	TEXT	Да	Нет	Краткое резюме статьи
Дата создания (Created_at)	TIMESTAMP	Нет	Нет	Дата и время создания, по умолчанию текущее время

Таблица 7 – Структура таблицы «Теги»

Наименование поля	Тип данных	Допустимость NULL	Индексация	Примечание
Код тега (Id)	INTEGER	Нет	Да	Первичный ключ, автоинкремент
Название (Name)	TEXT	Нет	Да	Имя тега, уникальное значение

Таблица 8 – Структура таблицы «Связи статей и тегов»

Наименование поля	Тип данных	Допустимость NULL	Индексация	Примечание
Код статьи (Article_id)	INTEGER	Нет	Да	Составной первичный ключ, внешний ключ на таблицу «Статьи»
Код тега (Tag_id)	INTEGER	Нет	Да	Составной первичный ключ, внешний ключ на таблицу «Теги»

Таблица 9 – Структура таблицы «Пользователи»

Наименование поля	Тип данных	Допустимость NULL	Индексация	Примечание
Код пользователя (Id)	INTEGER	Нет	Да	Первичный ключ, автоинкремент
Имя пользователя (Username)	TEXT	Нет	Да	Логин пользователя, уникальное значение

Пароль (Password)	ТЕХТ	Нет	Нет	Хеш пароля пользователя
Роль (Role)	ТЕХТ	Нет	Нет	Роль пользователя в системе

Таблица 10 – Структура таблицы «Прикрепленные файлы»

Наименование поля	Тип данных	Допустимость NULL	Индексация	Примечание
Код файла (Id)	INTEGER	Нет	Да	Первичный ключ, автоинкремент
Код статьи (Article_id)	INTEGER	Нет	Да	Внешний ключ на таблицу «Статьи»
Имя файла на сервере (Filename)	ТЕХТ	Нет	Нет	Уникальное имя файла, сохраненного на сервере
Оригинальное имя файла (Original_filename)	ТЕХТ	Нет	Нет	Оригинальное имя файла, загруженного пользователем
Дата загрузки (Created_at)	TIMESTAMP	Нет	Нет	Дата и время загрузки файла, по умолчанию текущее время

Физическая модель базы данных представлена на рисунке 5.

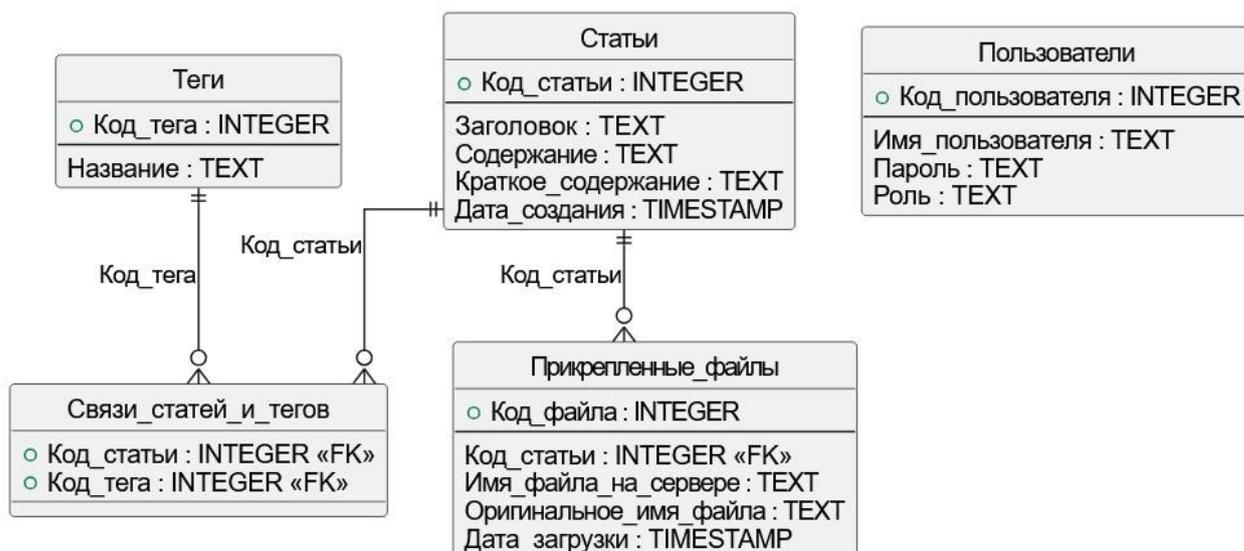


Рисунок 5 – Физическая модель базы данных

### 3.6 Характеристика функциональных подсистем

Разрабатываемая интеллектуальная система поддержки пользователей состоит из нескольких взаимосвязанных функциональных подсистем.

Подсистема ввода и взаимодействия с пользователем обеспечивает интерфейс для приема запросов на естественном языке через ИИ-чат, а также для управления контентом, включая статьи, теги и файлы, администраторами и редакторами. Далее, подсистема интеллектуальной обработки и поиска реализует ключевой функционал, такой как многостадийный ИИ-поиск с анализом запросов и семантической оценкой статей посредством внешних БЯМ, а также автоматическую генерацию резюме и тегов.

Данные, включая статьи базы знаний, теги и информацию о пользователях, структурированно хранятся и управляются подсистемой хранения данных на базе SQLite. Результаты поиска и контент представляются пользователю через подсистему отображения информации, которая формирует HTML-страницы. Подсистема администрирования и управления контентом предоставляет авторизованным пользователям инструменты для менеджмента учетных записей и информационного наполнения. Результаты интеллектуального поиска, найденные статьи базы знаний и другой информационный контент выводятся и

представляются пользователю посредством подсистемы вывода и отображения информации. Эта подсистема отвечает за динамическое формирование и корректное отображение HTML-страниц в веб-браузере пользователя, обеспечивая интуитивно понятное и удобное представление данных. Наконец, подсистема интеграции с внешними ИИ-сервисами отвечает за унифицированное взаимодействие с API различных провайдеров ИИ для выполнения задач обработки естественного языка.

### **3.7 Виды обеспечения системы**

Эффективное функционирование интеллектуальной системы поддержки пользователей базируется на комплексе различных видов обеспечения.

Алгоритмическое и математическое обеспечение включает совокупность алгоритмов и методов, лежащих в основе интеллектуальных функций. Сюда относятся алгоритмы многостадийного ИИ-поиска, включающие этапы анализа запроса и семантического ранжирования, а также методы обработки естественного языка, реализуемые через взаимодействие с БЯМ для понимания запросов и генерации контента, наряду с принципами построения эффективных промптов.

Лингвистическое обеспечение охватывает средства языковой поддержки взаимодействия, в частности поддержку запросов пользователей на естественном русском языке в ИИ-чате и использование русского языка в интерфейсе. Важным элементом являются специально разработанные шаблоны промптов на русском языке для БЯМ.

Программное обеспечение формирует ядро системы. Оно включает серверную часть на Python с использованием веб-фреймворка Flask, СУБД SQLite для хранения данных, клиентскую часть на HTML, CSS и JavaScript, а также необходимые библиотеки для интеграции с ИИ API и обеспечения безопасности.

Техническое обеспечение определяет комплекс аппаратных средств. Для серверной части необходим сервер под управлением ОС Linux или Windows с соответствующими ресурсами, в то время как клиентская часть предполагает наличие у пользователей устройств с доступом к сети и веб-браузером.

## 4 РЕАЛИЗАЦИЯ СИСТЕМЫ

### 4.1 Структура проекта

Программный проект имеет модульную структуру для обеспечения ясности и удобства сопровождения. Основной файл приложения `app.py` содержит логику веб-приложения на Flask, включая обработчики маршрутов. Модуль `api_provider.py` отвечает за унифицированное взаимодействие с внешними ИИ API (NVIDIA, Cloudflare). Конфигурационные параметры, такие как ключи API и настройки базы данных, хранятся в `config.py`. Утилиты для работы с паролями (`password_utils.py`, `common_passwords.py`) и схема базы данных (`schema.sql`) также вынесены в отдельные файлы. Пользовательский интерфейс представлен HTML-шаблонами в директории `templates` и статическими файлами (CSS, JS) в `static`. Загружаемые пользователями файлы размещаются в `static/uploads`, а файл базы данных SQLite – в `instance`.

В качестве технологического стека были выбраны: язык программирования Python, веб-фреймворк Flask, СУБД SQLite. Для клиентской части использовались HTML, CSS и нативный JavaScript. Шаблонизация осуществлялась с помощью Jinja2. Для взаимодействия с внешними API применялась библиотека Requests, для работы с веб-формами и защиты от CSRF – Flask-WTF, для парсинга HTML – BeautifulSoup, для работы с Markdown – python-markdown, для генерации уникальных идентификаторов – uuid.

### 4.2 Реализация модулей системы

Программная реализация охватывает серверную логику, интеграцию с ИИ-сервисами, интеллектуальные функции и пользовательский интерфейс.

#### 4.2.1 Серверная логика и взаимодействие с БД

Серверная часть на Flask инициализирует приложение, загружая конфигурацию из `config.py`. Маршрутизация связывает URL-адреса с функциями-обработчиками, отвечающими за отображение страниц и обработку AJAX-запросов. Взаимодействие с базой данных SQLite осуществляется через прямые SQL-запросы, инкапсулированные в специальных функциях для выполнения CRUD-

операций над статьями, тегами, пользователями и файлами. Система аутентификации использует хеширование паролей с солью и ролевую модель доступа, реализуемую через сессии Flask и декораторы для защиты маршрутов. Управление файлами включает их загрузку на сервер, проверку (тип, размер), сохранение с уникальным именем и удаление при необходимости.

#### 4.2.2 Интеграция с ИИ-сервисами

Модуль `api_provider.py` обеспечивает гибкое взаимодействие с API NVIDIA и Cloudflare. Центральная функция `call_api` служит единой точкой входа, выбирая провайдера на основе конфигурации. Для каждого провайдера реализованы отдельные функции: `call_nvidia_api`, `call_cloudflare_api`, которые формируют специфичные для API HTTP-запросы, включая заголовки авторизации и структуру тела запроса. Функция `format_messages_for_api` подготавливает сообщения для отправки в ИИ.

#### 4.2.3 Интеллектуальные функции

Ключевой интеллектуальной функцией является многостадийный ИИ-поиск, координируемый через эндпоинт `/api/chat-search`.

Система реализует автоматическую генерацию кратких резюме и тегов для статей. Эти функции инициируются AJAX-запросами со страниц создания/редактирования статей, формируют промпты для ИИ и вызывают `call_api` для получения результата.

#### 4.2.4 Пользовательский интерфейс

Пользовательский интерфейс создан с использованием HTML-шаблонов Jinja2 и стилизован с помощью CSS. Базовый шаблон `base.html` определяет общую структуру страниц. Клиентская логика на JavaScript обеспечивает интерактивность, особенно в ИИ-чате на главной странице (`index.html`), где управляется отображение сообщений, отправка запросов, обработка ответов и сохранение истории чата в `localStorage` [2]. На страницах управления контентом JavaScript используется для AJAX-запросов на генерацию резюме/тегов и управления файлами.

## 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ И АДМИНИСТРАТОРА

### 5.1 Руководство пользователя

Настоящее руководство предназначено для пользователей, использующих систему для получения информации и решения технических вопросов.

Доступ к системе осуществляется через веб-браузер по адресу, предоставленному администратором. Для просмотра статей базы знаний и использования ИИ-помощника регистрация и вход в систему не требуются. Главная страница системы (рисунок 6) предоставляет доступ к основным функциям: интеллектуальному поисковому ассистенту и быстрым ссылкам на разделы «База знаний» и «Теги» [5].

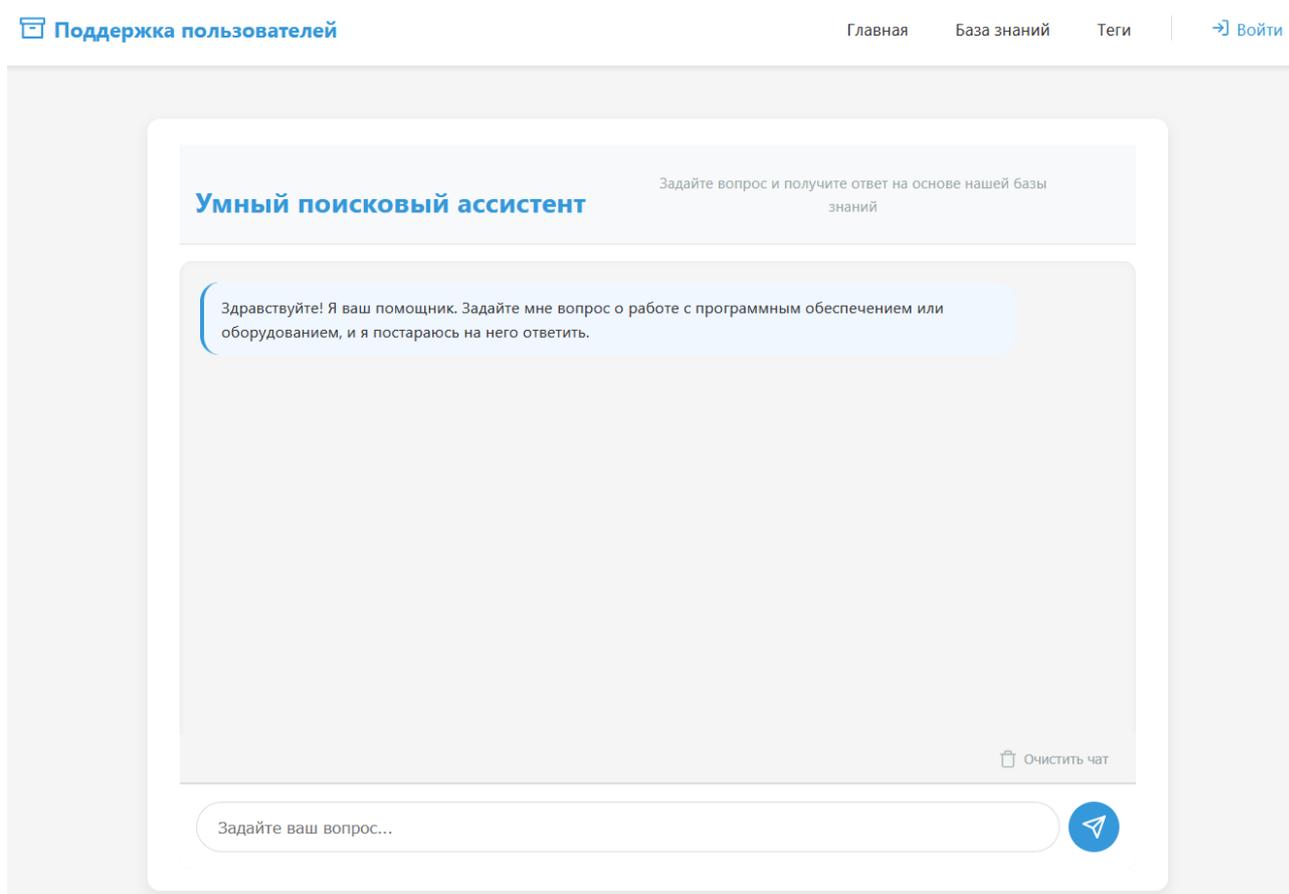
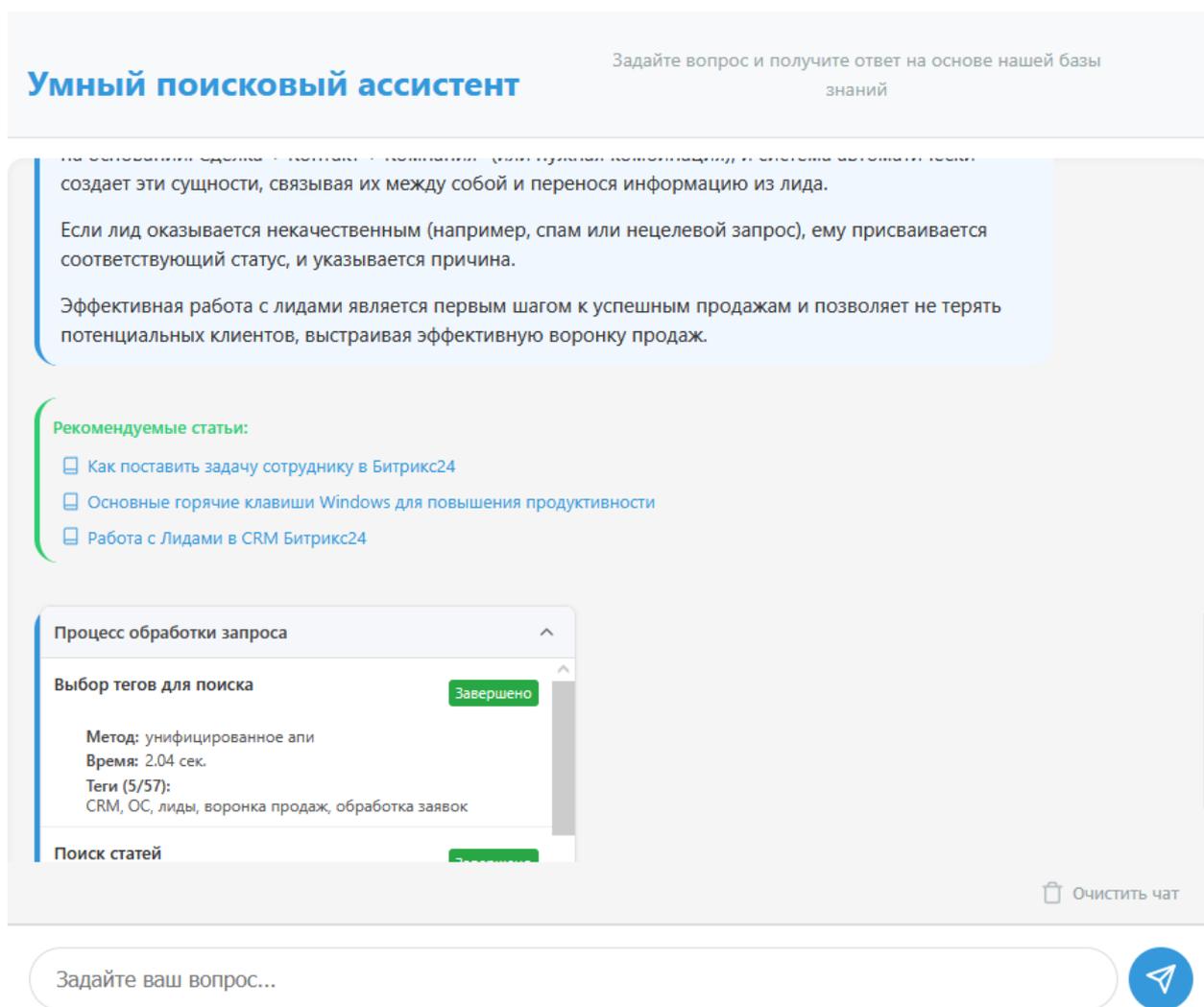


Рисунок 6 – Главная страница системы

Навигационное меню в верхней части страницы (рисунок 7) позволяет быстро переходить между разделами.

## Рисунок 7 – Навигационное меню

Интеллектуальный ассистент (ИИ-чат) предназначен для поиска информации в базе знаний с использованием запросов на естественном языке. Для начала работы с ассистентом пользователю необходимо ввести вопрос в текстовое поле в нижней части чата и нажать на кнопку отправки (рисунок 8).



## Рисунок 8 – Интерфейс ИИ-ассистента

Ассистент обработает запрос пользователя и предоставит ответ, основанный на статьях из базы знаний. Если релевантные статьи будут найдены, их

заголовки появятся под ответом в виде ссылок. История диалога сохраняется и доступна для просмотра. Кнопка «Очистить чат» позволяет удалить текущую историю переписки.

Раздел «База знаний» содержит список всех статей, доступных в системе, с возможностью постраничной навигации (рисунок 9). Для каждой статьи отображается заголовок, краткое превью и дата создания.

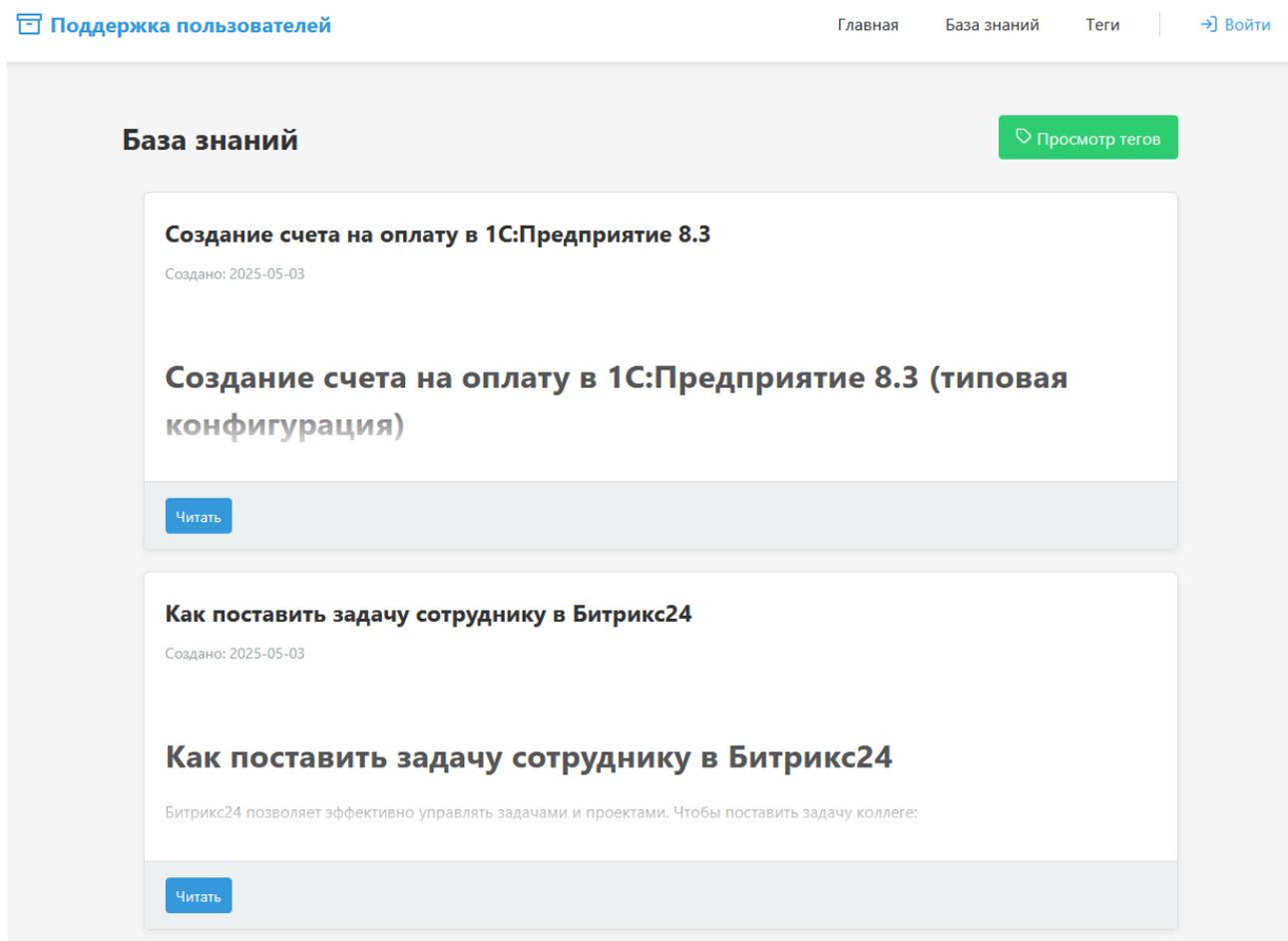


Рисунок 9 – Список статей в базе знаний

Для просмотра полного текста статьи, прикрепленных файлов и связанных тегов необходимо нажать на ее заголовок. Страница статьи (рисунок 10) отображает всю информацию.

## Как поставить задачу сотруднику в Битрикс24

Создано: 2025-05-03 10:25:57

### Краткое содержание

Инструкция по созданию и назначению задач сотрудникам в системе Битрикс24. Описаны шаги от ввода названия и описания до установки ответственного, крайнего срока и дополнительных параметров.

## Как поставить задачу сотруднику в Битрикс24

Битрикс24 позволяет эффективно управлять задачами и проектами. Чтобы поставить задачу коллеге:

1. **Найдите кнопку "Задача +"**: Обычно она находится в верхней панели портала или в левом меню в разделе "Задачи и Проекты".
2. **Введите название задачи**: Кратко и ясно опишите суть задачи.
3. **Добавьте описание**: Подробно опишите, что нужно сделать, какие ожидаются результаты. Можно использовать форматирование, списки, прикреплять файлы.
4. **Назначьте ответственного**: В поле "Ответственный" начните вводить имя сотрудника и выберите его из списка.
5. **Укажите постановщика**: По умолчанию это вы, но можно изменить при необходимости.
6. **Добавьте соисполнителей и наблюдателей (опционально)**:
  - *Соисполнители* также участвуют в выполнении задачи.
  - *Наблюдатели* получают уведомления о ходе выполнения задачи.
7. **Установите крайний срок**: Укажите дату и, при необходимости, время, к которому задача должна быть выполнена.
8. **Настройте дополнительные параметры (опционально)**:
  - **Проект**: Свяжите задачу с конкретным проектом или группой.

### Рисунок 10 – Страница просмотра статьи

Раздел «Теги» позволяет просмотреть все теги, используемые в системе. При нажатии на тег отобразится список всех статей, связанных с ним.

## 5.2 Руководство администратора

Данный раздел предназначен для пользователей с ролями «Редактор» и «Администратор», отвечающих за управление контентом и администрирование системы.

Для доступа к функциям управления контентом и администрирования необходимо войти в систему. Ссылка «Войти» находится в правом верхнем углу навигационного меню. На странице входа (рисунок 11) введите свои имя пользователя и пароль.

## Вход в систему

Имя пользователя

Пароль

Войти

[Отмена](#)

Если вы клиент и хотите только просматривать материалы, вход в систему не требуется.

Рисунок 11 – Страница входа в систему

После успешного входа пользователи с соответствующими правами получают доступ к созданию и редактированию статей.

Для создания новой статьи перейдите по ссылке «Создать статью» в навигационном меню. Откроется форма создания статьи (рисунок 12). Заполните заголовок, основное содержание, при необходимости сгенерируйте или введите краткое содержание и теги. Можно прикрепить файлы.

## Редактирование статьи

Заголовок

Содержание

# Создание счета на оплату в 1С:Предприятие 8.3 (типовая конфигурация)

Счет на оплату - один из основных документов в хозяйственной деятельности. Вот как его создать в 1С:

1. **\*\*Перейдите в раздел "Продажи"\*\*. В левом меню выберите "Продажи".**
2. **\*\*Откройте журнал "Счета покупателям"\*\*. В разделе "Продажи" найдите и кликните "Счета покупателям".**
3. **\*\*Нажмите кнопку "Создать"\*\*. Откроется форма нового документа "Счет покупателю".**
4. **\*\*Заполните основные поля:\*\***
  - \* **\*\*Контрагент:\*\*** Выберите покупателя из справочника. Если его нет, создайте новую карточку.
  - \* **\*\*Договор:\*\*** Выберите действующий договор с контрагентом.
  - \* **\*\*Организация:\*\*** Убедитесь, что выбрана ваша организация.

Краткое содержание

Пошаговая инструкция по созданию и печати документа "Счет на оплату" в программе 1С:Предприятие 8.3. Описаны основные этапы: выбор контрагента, заполнение товаров/услуг, проверка реквизитов и проведение документа.

Теги (через запятую)

Укажите теги через запятую, например: инструкция, ошибка, настройка

Рисунок 12 – Форма создания/редактирования статьи

Кнопки «Сгенерировать краткое содержание» и «Автогенерация тегов» позволяют использовать ИИ для автоматического заполнения этих полей на основе основного содержания статьи.

На странице просмотра статьи для пользователей с правами редактора или администратора доступна кнопка «Редактировать». Форма редактирования аналогична форме создания, но предварительно заполнена данными текущей статьи. Здесь также можно управлять прикрепленными файлами (удалять существующие или добавлять новые).

Администраторы могут удалять статьи со страницы просмотра статьи с помощью кнопки «Удалить». Перед удалением система запросит подтверждение.

Администраторы имеют доступ к разделу «Пользователи» через навигационное меню. На этой странице (рисунок 13) отображается список всех

зарегистрированных пользователей, их роли, а также доступны функции добавления новых пользователей (администраторов или редакторов) и удаления существующих.

Поддержка пользователей

Главная База знаний Теги Создать статью Пользователи admin (admin) Выйти

## Управление пользователями

Добавить администратора Добавить редактора

ID	Имя пользователя	Роль	Действия
1	admin	Администратор	Текущий пользователь
2	root	Администратор	Удалить

Вернуться на главную

Рисунок 13 – Страница управления пользователями

При создании нового пользователя необходимо указать имя, пароль (согласно требованиям безопасности) и роль.

## 6 ЗАЩИТА ИНФОРМАЦИИ

Обеспечение надежной защиты информации является неотъемлемой частью разработки и последующей эксплуатации интеллектуальной системы поддержки пользователей [7]. Это особенно актуально для производственного предприятия, такого как ООО «Амурагроцентр», где система может оперировать данными, касающимися ИТ-инфраструктуры, внутренних процессов и потенциально чувствительной коммерческой или технологической информации.

### 6.1 Модель угроз и основные векторы атак

При проектировании системы был проведен анализ потенциальных угроз и векторов атак, которые могут быть направлены на систему. Во-первых, это угрозы несанкционированного доступа. Сюда относятся попытки взлома учетных записей пользователей путем подбора пароля, использования методов социальной инженерии, а также эксплуатации уязвимостей приложения для обхода механизмов авторизации. Последствиями такого доступа могут стать просмотр, изменение или удаление конфиденциальных данных, а также создание вредоносного контента в базе знаний.

Во-вторых, актуальны распространенные веб-уязвимости. К ним относятся SQL-инъекции, позволяющие злоумышленнику выполнять произвольные запросы к базе данных системы; межсайтовый скриптинг (XSS), который заключается во внедрении вредоносного JavaScript-кода в страницы, просматриваемые другими пользователями; подделка межсайтовых запросов (CSRF), позволяющая выполнять определенные действия в системе от имени аутентифицированного пользователя без его ведома; а также риски, связанные с незащищенной загрузкой файлов, которые могут содержать вредоносный код.

В-третьих, существует угроза утечки конфиденциальной информации. Это может произойти из-за ошибок в логике контроля доступа, неправильной обработки исключительных ситуаций в коде, недостаточной защиты данных при их передаче по сети, а также из-за утечки чувствительных данных через промпты, передаваемые внешним ИИ API.

В-четвертых, система может быть подвержена атакам типа «отказ в обслуживании» (DoS/DDoS). Такие атаки направлены на перегрузку сервера большим количеством запросов, что приводит к исчерпанию его ресурсов (процессорного времени, памяти, пропускной способности сети) и, как следствие, к недоступности системы для легитимных пользователей. Атаки также могут быть направлены на внешние ИИ API, используемые системой, с целью исчерпания доступных квот или увеличения финансовых затрат.

В-пятых, существуют специфические угрозы, связанные с использованием ИИ API. К ним относятся инъекции промптов (prompt injection), когда злоумышленник пытается манипулировать поведением большой языковой модели путем подачи специально сформированных запросов. Также существуют финансовые риски, связанные с необоснованным или чрезмерным расходом ресурсов платных API, и риск генерации ИИ некорректных, неточных или даже вредоносных ответов.

Наконец, важной угрозой является компрометация API-ключей, необходимых для доступа к внешним ИИ-сервисам. Это может произойти из-за их небезопасного хранения, случайной утечки через систему контроля версий, логи сервера или другие каналы. Последствия компрометации ключей могут включать несанкционированный доступ к ИИ-сервисам от имени организации и связанные с этим финансовые и репутационные потери.

## **6.2 Аутентификация и авторизация**

Для обеспечения контроля доступа и идентификации пользователей в разработанной системе реализованы следующие механизмы аутентификации и авторизации.

Во-первых, применяется парольная защита учетных записей пользователей. При создании или изменении пароля используется функция `hash_password`, которая выполняет криптографическое хеширование пароля с добавлением индивидуальной случайной строки символов, известной как «соль». Это значительно усложняет восстановление исходного пароля даже в случае компрометации базы данных. Проверка введенного пользователем пароля при входе в

систему осуществляется функцией `verify_password`, которая выполняет аналогичное хеширование с использованием сохраненной соли и сравнивает результат с хешем, хранящимся в базе. Для повышения надежности паролей в системе реализована функция `validate_password_strength`, которая проверяет соответствие создаваемых паролей определенным критериям сложности (например, минимальная длина, наличие цифр, букв разного регистра и специальных символов).

Во-вторых, в системе внедрена ролевая модель доступа RBAC. Каждому пользователю системы при регистрации или администратором назначается определенная роль, например, `'admin'` (администратор с полными правами), `'editor'` (редактор с правами на управление контентом) или `'client'` (обычный пользователь с правами только на чтение). Доступ к различным функциям и разделам системы строго контролируется на уровне обработки HTTP-запросов (маршрутов) с помощью специальных декораторов, таких как `@login_required` (требует, чтобы пользователь был аутентифицирован), `@admin_required` (требует наличия роли администратора) и `@editor_required` (требует наличия роли редактора или администратора). Эти декораторы проверяют роль пользователя, информация о которой сохраняется в сессии после успешной аутентификации, и при необходимости могут запрашивать актуальную роль из базы данных для дополнительной проверки.

### **6.3 Защита от веб-уязвимостей**

В процессе разработки системы были приняты целенаправленные меры для минимизации рисков, связанных с распространенными веб-уязвимостями.

Для предотвращения SQL-инъекций, которые могут позволить злоумышленнику выполнить произвольный SQL-код в базе данных, при взаимодействии с СУБД SQLite через встроенный модуль `sqlite3` используется техника параметризованных запросов. Этот подход гарантирует, что данные, вводимые пользователем, всегда обрабатываются как данные, а не как часть исполняемого SQL-кода. Дополнительно, на серверной стороне проводится валидация всех входных данных, получаемых от пользователя, перед их использованием в запросах к базе данных или в другой логике приложения.

Для защиты от межсайтового скриптинга (XSS), который заключается во внедрении вредоносного JavaScript-кода в веб-страницы, просматриваемые другими пользователями, используется механизм автоматического экранирования пользовательского вывода в HTML-шаблонах. Это является стандартным поведением шаблонизатора Jinja2, используемого во Flask-приложении. На уровне веб-сервера рекомендуется настраивать заголовок Content-Security-Policy (CSP), который дополнительно ограничивает источники, из которых браузер может загружать и выполнять скрипты, стили и другие ресурсы.

Для противодействия атакам типа «подделка межсайтовых запросов» (CSRF), позволяющим злоумышленнику выполнять действия от имени аутентифицированного пользователя без его ведома, в системе используется библиотека Flask-WTF. Эта библиотека автоматически генерирует уникальные CSRF-токены для каждой сессии пользователя и встраивает их во все формы, отправляемые методом POST. При получении такого запроса сервер проверяет наличие и корректность CSRF-токена, и, если токен отсутствует или недействителен, запрос отклоняется.

Обеспечение безопасной загрузки файлов включает несколько мер. Во-первых, реализована проверка расширений загружаемых файлов, чтобы разрешать загрузку только файлов определенных, безопасных типов [8]. Во-вторых, загружаемые файлы сохраняются на сервере с уникальными именами, генерируемыми с помощью uuid, и размещаются вне корневой директории веб-сервера, чтобы предотвратить их прямое исполнение или доступ по URL.

#### **6.4 Рекомендации по обеспечению безопасности**

Для дальнейшего повышения уровня безопасности разработанной интеллектуальной системы поддержки при ее внедрении и в процессе эксплуатации рекомендуется придерживаться следующих практик и мер.

В области сетевой безопасности целесообразно развертывать приложение за межсетевым экраном, который будет контролировать и фильтровать входящий и исходящий сетевой трафик. Весь трафик к веб-приложению должен осуществляться исключительно по протоколу HTTPS, для чего необходимо

настроить веб-сервер на использование SSL/TLS сертификатов. Рассмотреть возможность сегментации сети, при которой сервер с приложением находится в отдельном защищенном сегменте.

Касательно серверной безопасности, важно обеспечить регулярное обновление операционной системы сервера и всего используемого программного обеспечения, включая веб-сервер, интерпретатор Python и все зависимости приложения, для своевременного устранения известных уязвимостей. Рекомендуется использовать технологии изоляции, такие как контейнеризация с помощью Docker, для ограничения потенциального ущерба в случае компрометации приложения. Необходимо отключать все неиспользуемые на сервере службы и порты для уменьшения поверхности атаки. Целесообразно настроить систему обнаружения вторжений и проводить регулярное сканирование сервера на наличие уязвимостей.

В части управления доступом следует внедрить строгую политику паролей для всех пользователей системы, особенно для администраторов и редакторов, требуя использования сложных и уникальных паролей. Необходимо проводить регулярный аудит учетных записей пользователей, удаляя неактивные или более не нужные аккаунты. Доступ к административным функциям системы можно дополнительно ограничить по IP-адресам или требовать подключения через частную виртуальную сеть.

Критически важным аспектом является резервное копирование. Необходимо настроить регулярное автоматическое резервное копирование файла базы данных SQLite и каталога с загруженными пользовательскими файлами. Резервные копии должны храниться в безопасном месте, отличном от основного сервера, и периодически проверяться на возможность восстановления.

Перечисленные меры в совокупности обеспечивают базовый, но достаточно надежный уровень защищенности разработанной интеллектуальной системы поддержки пользователей.

## 7 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

### 7.1 Безопасность

Обеспечение всесторонней безопасности при работе с разработанной интеллектуальной системой поддержки пользователей представляет собой многоаспектную и сложную задачу, охватывающую чрезвычайно широкий круг вопросов. Эти вопросы варьируются от тщательной эргономической организации рабочих мест операторов и пользователей системы, включая продуманность дизайна пользовательского интерфейса самой системы, и заканчивая строгим соблюдением установленных правил электробезопасности и пожарной безопасности на всех уровнях.

Создание максимально комфортных, производительных и, что наиболее важно, безопасных условий труда для сотрудников ООО «Амурагроцентр», которые будут ежедневно взаимодействовать с системой, является безусловным приоритетным направлением при ее внедрении и последующей эксплуатации.

#### 7.1.1 Эргономика и безопасность рабочего места пользователя системы

Эффективность трудовой деятельности и, что не менее важно, сохранение физического и психологического здоровья пользователей, работающих с интеллектуальной системой поддержки, в значительной степени зависят от правильной и продуманной организации их рабочих мест, а также от тщательного учета многочисленных эргономических требований, предъявляемых как к аппаратному обеспечению, так и к интерфейсу самой системы.

Прежде всего, необходимо уделить пристальное внимание вопросам детальной организации каждого рабочего места при непосредственном взаимодействии с системой. Условия труда должны не просто формально, а фактически полностью соответствовать действующим гигиеническим нормативам, в частности, положениям, подробно изложенным в СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [8].

Освещение рабочих мест должно быть достаточным по интенсивности и максимально равномерным по распределению светового потока, полностью исключая возможность слепящего действия прямых или отраженных лучей света и образование резких, контрастных теней, способных вызывать зрительное напряжение; при этом рекомендуется обеспечивать оптимальное сочетание естественного освещения, поступающего через оконные проемы, и искусственного освещения, создаваемого осветительными приборами.

Параметры микроклимата в рабочих помещениях, такие как температура воздуха, его относительная влажность и скорость движения воздушных масс, должны постоянно поддерживаться в оптимальных пределах, установленных соответствующими санитарными нормативами для различных категорий работ.

Используемая мебель, включая конструкцию рабочего стола и офисного кресла, должна в полной мере соответствовать антропометрическим данным конкретного пользователя и обеспечивать поддержание удобной, физиологически правильной рабочей позы на протяжении всего рабочего дня, что особо подчеркивается в Приказе Минтруда РФ от 29.10.2021 N 774Н «Об утверждении общих требований к организации безопасного рабочего места».

Рабочий стол должен иметь достаточную площадь поверхности для размещения необходимого оборудования и рабочих материалов, а также обеспечивать свободное пространство для ног. Кресло должно иметь регулируемую высоту сиденья и спинки, а также, желательно, подлокотники для снижения нагрузки на плечевой пояс.

Особые, повышенные требования предъявляются к видеодисплейным терминалам (ВДТ), посредством которых осуществляется непосредственный доступ к функционалу интеллектуальной системы и взаимодействие с ней. Характеристики используемых мониторов, включая такие параметры, как размер диагонали экрана, его разрешение, частота обновления кадров, а также отсутствие мерцания и бликов, должны быть подобраны таким образом, чтобы минимизировать зрительное утомление пользователей и предотвратить развитие профессиональных заболеваний органов зрения.

Следует отдавать явное предпочтение современным моделям мониторов, обладающим высоким качеством формируемого изображения, хорошей цветопередачей и предоставляющим пользователю возможность индивидуальной настройки параметров яркости, контрастности и цветовой температуры, что полностью соответствует общим принципам, изложенным в ГОСТ Р 50948-2001 «Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности».

Важное значение также придается вопросам эргономики графического пользовательского интерфейса самой разработанной системы. Читаемость используемых шрифтов, оптимальная контрастность применяемых цветовых схем, логичность и предсказуемость навигационных решений, а также интуитивная понятность всех элементов управления, как в интерфейсе базы знаний, так и в интерфейсе ИИ-чата, должны способствовать существенному снижению когнитивной нагрузки на пользователя и значительному уменьшению вероятности совершения им операторских ошибок. Принципы проектирования подобных пользовательских интерфейсов подробно отражены и регламентированы в ГОСТ Р ИСО 9241-161-2016 «Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса».

Также необходимо в полной мере учитывать требования доступности, изложенные в ГОСТ Р 52872-2019, с целью обеспечения возможности эффективного использования системы широким кругом пользователей, включая лиц с ограниченными возможностями здоровья и особыми потребностями. Это может включать поддержку масштабирования интерфейса, возможность управления с клавиатуры и совместимость со вспомогательными технологиями.

Немаловажным аспектом обеспечения работоспособности и здоровья пользователей является корректная организация режимов их труда и отдыха. Для сотрудников, которые активно и продолжительное время работают с интеллектуальной системой, необходимо предусматривать введение регулярных, регламентированных перерывов в работе. Во время таких перерывов настоятельно рекомендуется выполнять специально подобранный комплекс упражнений для глаз,

направленных на снятие зрительного напряжения, и легкую физическую разминку для всего тела, что способствует эффективной профилактике общего утомления и предотвращению развития профессиональных заболеваний, связанных с длительной статической нагрузкой и работой за компьютером. Периодичность и длительность таких перерывов должны устанавливаться с учетом характера и напряженности выполняемой работы.

#### 7.1.2 Электробезопасность и пожарная безопасность при эксплуатации ИТ-оборудования системы

Эксплуатация серверного оборудования, на котором развернута и функционирует интеллектуальная система поддержки, а также повседневное использование персональных компьютеров и периферийных устройств пользователями, неизбежно сопряжены с потенциальными рисками поражения электрическим током и возникновения пожароопасных ситуаций. В связи с этим, строгое и неукоснительное соблюдение установленных правил электробезопасности и пожарной безопасности является абсолютным и обязательным требованием для всех сотрудников предприятия.

Все используемое информационно-технологическое оборудование должно иметь соответствующие сертификаты соответствия национальным и международным стандартам безопасности и проходить регулярную проверку технического состояния. Электропроводка в помещениях, а также системы защитного заземления и зануления должны быть выполнены квалифицированными специалистами в строгом соответствии с действующими нормами и правилами устройства электроустановок.

Необходимо обеспечивать регулярное проведение плановых и внеплановых проверок состояния электрооборудования, кабельных линий и коммутационных устройств с привлечением аттестованного электротехнического персонала. Персонал, работающий с электрооборудованием, должен иметь соответствующую группу по электробезопасности и проходить периодические инструктажи.

Следует категорически избегать перегрузки электрических сетей путем подключения чрезмерного количества потребителей к одной розетке или

удлинителю, а также использования неисправного, поврежденного или не сертифицированного электрооборудования. Необходимо обеспечить свободный доступ к эвакуационным выходам и средствам пожаротушения.

## **7.2 Экологичность**

Процессы разработки, внедрения и последующей эксплуатации интеллектуальной системы поддержки пользователей, как и любой другой современной информационной системы, неизбежно оказывают определенное, хотя и различное по степени, воздействие на окружающую природную среду. Проведение тщательного анализа этого воздействия на всех этапах жизненного цикла системы и принятие своевременных и адекватных мер по его минимизации являются важными составляющими ответственного и дальновидного подхода к внедрению и использованию информационных технологий в деятельности ООО «Амурагроцентр».

### **7.2.1 Воздействие на окружающую среду при эксплуатации системы**

Основным и наиболее значимым фактором воздействия разработанной системы на окружающую среду в процессе ее повседневной эксплуатации является потребление электрической энергии. Данный аспект включает в себя прямое энергопотребление серверного оборудования, которое обеспечивает непрерывное функционирование веб-приложения, разработанного на базе фреймворка Flask, осуществляет работу с базой данных SQLite и обеспечивает взаимодействие с внешними программными интерфейсами (API) систем искусственного интеллекта.

Кроме того, сюда же относится и совокупное энергопотребление всех клиентских рабочих станций сотрудников ООО «Амурагроцентр», используемых для доступа к системе. Использование внешних ИИ API также опосредованно увеличивает общие энергозатраты, поскольку требует работы серверных мощностей компаний-провайдеров этих услуг.

Проведение точной количественной оценки суммарного энергопотребления системы требует детального учета специфических технических

характеристик конкретного используемого оборудования, режимов его работы и фактической интенсивности эксплуатации системы пользователями.

Однако, при объективном сравнении с традиционными, полностью ручными процессами технической поддержки и классическим бумажным документооборотом, внедрение интеллектуальной системы способно привести к некоторому ощутимому снижению общего потребления различных ресурсов, например, за счет значительного сокращения необходимости в регулярной распечатке объемных инструктивных материалов и потенциального уменьшения транспортных издержек, если значительная часть пользовательских запросов успешно решается в удаленном режиме без необходимости физического перемещения специалистов.

Внедрение современной интеллектуальной системы поддержки, основанной на полностью цифровой базе знаний и преимущественно электронном взаимодействии между пользователями и системой, способствует достижению значительного ресурсосбережения.

Прежде всего, это выражается в существенном и измеримом сокращении потребления бумаги, которая ранее была необходима для печати многочисленных инструкций, технологических регламентов, пользовательских руководств и отчетов по обработанным заявкам. Также наблюдается заметное снижение потребности в разнообразных канцелярских товарах (ручки, карандаши, папки и т.д.) и различных расходных материалах для оргтехники (картриджи для принтеров, тонеры и т.п.).

#### 7.2.2 Управление отходами, связанными с жизненным циклом системы

Полный жизненный цикл интеллектуальной системы поддержки включает в себя последовательные этапы проектирования, разработки, тестирования, внедрения, активной эксплуатации и, в конечном итоге, вывода из эксплуатации как программного обеспечения, так и соответствующей аппаратной платформы. Именно на последнем этапе возникает актуальная и сложная проблема эффективного управления образующимися электронными отходами.

Серверное оборудование (серверы, системы хранения данных), персональные компьютеры пользователей, мониторы, клавиатуры, мыши и другие периферийные устройства по истечении установленного срока их службы, при моральном устаревании или при физическом выходе из строя трансформируются в категорию отходов, требующих специальной, экологически безопасной процедуры утилизации.

ООО «Амурагроцентр» должен разработать и внедрить или строго следовать уже установленной на предприятии регламентированной процедуре обращения с такими электронными отходами, обеспечивая их своевременную и документально оформленную передачу специализированным лицензированным организациям, которые имеют официальное разрешение на осуществление деятельности по утилизации электронного и электротехнического оборудования.

Такой подход позволяет эффективно предотвратить попадание содержащихся в электронных компонентах вредных веществ, таких как тяжелые металлы (свинец, ртуть, кадмий), бромированные антипирены и различные виды токсичных пластиков, в окружающую среду, а также обеспечить возможность частичной переработки и повторного использования ценных компонентов (драгоценные металлы, медь, алюминий).

Требования к обращению с отходами производства и потребления, включая специфику электронных отходов, регламентируются, в том числе, обширными положениями СанПиН 2.1.3684-21 «Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпидемических (профилактических) мероприятий» [9].

В случае использования ртутьсодержащих люминесцентных ламп для освещения помещений, в которых эксплуатируется система или работают ее пользователи, их утилизация также требует особого, строго регламентированного подхода из-за высокой токсичности паров ртути. Необходимо организовать

их отдельный сбор в специальные контейнеры и последующую передачу на демеркуризацию специализированным предприятиям, имеющим соответствующую лицензию. Аналогичным образом, несмотря на общее прогнозируемое снижение потребления бумаги в результате внедрения системы, ее остатки (макулатура) также должны собираться отдельно от прочих отходов и по возможности передаваться на дальнейшую переработку для производства вторичной бумажной продукции.

### 7.2.3 Предложения по минимизации экологического воздействия

Для целенаправленного снижения общего экологического следа от эксплуатации интеллектуальной системы поддержки пользователей в ООО «Амурагроцентр» настоятельно рекомендуется планирование и реализация комплекса взаимодополняющих мер.

Прежде всего, при осуществлении закупки нового или при проведении модернизации существующего ИТ-оборудования следует отдавать безусловное предпочтение энергоэффективным моделям, которые соответствуют современным международным и национальным стандартам в области энергосбережения, таким как, например, стандарт Energy Star или его аналоги.

Кроме того, важным направлением деятельности является проведение регулярной оптимизации программного кода самой интеллектуальной системы с целью снижения вычислительной нагрузки на серверные ресурсы и, как прямое следствие, уменьшения общего объема энергопотребления. Это может включать рефакторинг кода, оптимизацию запросов к базе данных и использование более эффективных алгоритмов.

В дополнение к этим техническим мерам, представляется весьма целесообразным и перспективным внедрение на предприятии комплексной политики «зеленого офиса». Такая политика должна включать в себя конкретные меры по экономии электроэнергии (например, автоматическая настройка спящего режима для персональных компьютеров во время простоев, физическое выключение неиспользуемого периферийного оборудования по окончании рабочего дня), а также мероприятия, направленные на дальнейшее сокращение потребления

бумаги (например, поощрение двусторонней печати, использование электронного документооборота там, где это возможно) и совершенствование системы раздельного сбора различных видов отходов для их последующей переработки.

### **7.3 Чрезвычайные ситуации (ЧС)**

При повседневной эксплуатации интеллектуальной системы поддержки пользователей необходимо постоянно учитывать вероятность возникновения различных нештатных или даже чрезвычайных ситуаций, которые потенциально могут привести к серьезному нарушению ее штатного функционирования, необратимой потере критически важных корпоративных данных или создать непосредственную угрозу для жизни и здоровья персонала, работающего с системой.

7.3.1 Анализ потенциальных ЧС, влияющих на функционирование системы и безопасность пользователей

К потенциальным чрезвычайным ситуациям техногенного характера, способным оказать существенное негативное влияние на стабильную работу системы, относятся, в первую очередь, различные по своей природе и масштабу сбои в системе электропитания предприятия. Внезапное и непредвиденное отключение электроэнергии, даже кратковременное, может привести к аварийной остановке сервера, на котором развернуто и функционирует приложение, и, как следствие, к потере всех не сохраненных на тот момент данных, а также к полной невозможности доступа пользователей к системе до восстановления электропитания.

Отказы самого серверного оборудования, такие как физический выход из строя жестких дисков (появление сбойных секторов, отказ механики или контроллера), неисправность модулей оперативной памяти, перегрев или отказ центрального процессора, а также выход из строя других жизненно важных компонентов сервера, также неизбежно приведут к полной или частичной недоступности системы для пользователей.

Неисправности активного сетевого оборудования, включая коммутаторы, маршрутизаторы и межсетевые экраны, могут нарушить нормальную сетевую связность между клиентскими рабочими местами и центральным сервером

системы, а также ограничить или полностью заблокировать доступ к внешним ИИ API, необходимым для полноценной работы интеллектуальных функций.

Значительно более серьезной и опасной чрезвычайной ситуацией является возникновение пожара в офисном здании или непосредственно в серверном помещении, где размещено оборудование. Помимо прямой и неминуемой угрозы физическому уничтожению дорогостоящего оборудования и безвозвратной потере всех хранимых данных, пожар представляет серьезную опасность для жизни и здоровья сотрудников предприятия.

Для эффективного предотвращения пожаров, которые потенциально могут быть связаны с эксплуатацией ИТ-оборудования, серверные помещения, а также рабочие места пользователей должны быть в обязательном порядке оснащены современными и исправными средствами пожаротушения, тип и количество которых должны соответствовать классу возможного пожара (обычно класс Е для электрооборудования). Весь персонал, работающий с системой, должен проходить регулярное обучение правилам пожарной безопасности, включая знание порядка действий при обнаружении возгорания и правил использования первичных средств пожаротушения.

Специфическими для разработанной интеллектуальной системы являются чрезвычайные ситуации, непосредственно связанные с ее ключевыми программными и аппаратными компонентами. Продолжительные сбои в работе или полная недоступность внешних ИИ API, которые активно используются для реализации таких интеллектуальных функций системы, как семантический поиск информации, автоматическая генерация кратких резюме по текстам и присвоение тегов статьям, приведут к временной или полной неработоспособности указанных функций. При этом базовая функциональность системы, включая простой доступ к базе знаний и стандартный поиск по ключевым словам, может при определенных условиях сохраниться.

Особенно критической и потенциально катастрофической ситуацией является физическое повреждение или полная логическая потеря файла базы данных SQLite, в котором централизованно хранится вся информационная база системы,

включая тексты статей, присвоенные им теги, а также учетные данные пользователей. Подобное нежелательное событие может произойти вследствие разнообразных аппаратных сбоев (например, отказ дискового накопителя), критических ошибок в программном обеспечении самой системы или операционной системы сервера, или же в результате преднамеренных вредоносных действий злоумышленников (например, вирусная атака или несанкционированный доступ).

Чрезвычайные ситуации природного характера, такие как сильные ураганы с порывистым ветром, обширные наводнения, землетрясения различной интенсивности или экстремальные погодные условия (например, аномальные морозы или жара), могут также оказать разрушительное воздействие на физическую инфраструктуру предприятия (здания, линии электропередач, каналы связи) и, как прямое следствие, на стабильную работу интеллектуальной системы. Однако их конкретная вероятность и специфический характер проявления в значительной степени зависят от географического и климатического расположения ООО «Амурагроцентр».

7.3.2 Меры по предупреждению и ликвидации ЧС в ООО «Амурагроцентр» применительно к системе

Для обеспечения эффективного предупреждения и существенной минимизации возможных негативных последствий от возникновения чрезвычайных ситуаций в ООО «Амурагроцентр» должен быть заблаговременно разработан, утвержден и планомерно реализован комплекс взаимосвязанных и хорошо скоординированных организационных и технических мероприятий.

Организационные мероприятия в обязательном порядке должны включать проведение регулярных, не реже одного раза в квартал, инструктажей всего персонала по правилам поведения и конкретным действиям в условиях различных ЧС, а также разработку, утверждение и периодическую практическую отработку (в виде учений и тренировок) детальных планов эвакуации при возникновении пожара или других угроз.

Технические мероприятия должны предусматривать обязательное наличие и исправное состояние современных систем автоматического обнаружения и

тушения пожара в серверных помещениях и других критически важных зонах предприятия. С целью обеспечения гарантированной бесперебойной работы сервера, на котором функционирует интеллектуальная система, настоятельно рекомендуется использование мощных источников бесперебойного питания (ИБП), способных поддерживать работу оборудования в течение времени, достаточного для корректного завершения всех операций и штатного отключения, и, при наличии такой финансовой и технической возможности, установку резервных дизельных или бензиновых электрогенераторов. Для защиты от перепадов напряжения целесообразно использовать стабилизаторы напряжения.

Ключевой и наиболее важной мерой для обеспечения надежной защиты корпоративных данных, хранящихся в интеллектуальной системе, является внедрение и строгое соблюдение практики регулярного, автоматизированного резервного копирования файла базы данных SQLite, а также всех загруженных пользователями в систему файлов и документов.

Полученные резервные копии должны в обязательном порядке храниться в надежном и физически защищенном месте, предпочтительно географически удаленном от местоположения основного сервера (например, в облачном хранилище или на удаленном резервном сервере), и периодически проходить тщательную проверку на возможность успешного и полного восстановления данных.

Необходимо разработать, документально оформить и утвердить руководством предприятия детальный и пошаговый план восстановления работоспособности системы после возникновения ЧС. Этот план должен включать в себя четкие и однозначные процедуры восстановления данных из актуальных резервных копий, а также последовательность действий по проверке, настройке и вводу в эксплуатацию основного и резервного оборудования.

### 7.3.3 Действия пользователей и администраторов системы в случае ЧС

В случае фактического возникновения любой чрезвычайной ситуации персонал ООО «Амурагроцентр», включая как рядовых пользователей, так и администраторов системы, должен действовать строго, оперативно и слаженно в

соответствии с заранее утвержденными на предприятии инструкциями и планами действий в ЧС.

Рядовые пользователи системы при обнаружении ее полной недоступности, частичной неработоспособности или некорректной работы отдельных функций обязаны незамедлительно и максимально подробно уведомить об этом системного администратора или другого ответственного ИТ-специалиста, предоставив всю имеющуюся информацию о характере проблемы и времени ее возникновения.

Администратор системы, в свою очередь, при получении информации о произошедшем сбое или возникновении ЧС должен незамедлительно провести первичную диагностику для точного определения причины возникшей проблемы, оценить ее масштаб и возможные последствия, и затем незамедлительно приступить к выполнению всех необходимых мероприятий по восстановлению полной работоспособности системы согласно утвержденному плану действий в ЧС.

В случае выявления сбоев в работе внешних ИИ API, от которых зависит функционирование интеллектуальных возможностей системы, администратору следует оперативно проверить текущий статус соответствующих сервисов на сайтах компаний-провайдеров и, при необходимости и наличии такой технической возможности, временно переключиться на использование альтернативных (резервных) API, если такая опция предусмотрена конфигурацией системы.

В противном случае, он должен своевременно уведомить всех пользователей об имеющихся временных ограничениях функциональности системы и ориентировочных сроках их устранения. Любой инцидент, связанный с ЧС, должен быть тщательно задокументирован для последующего анализа и совершенствования мер безопасности.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы успешно разработан интеллектуальная система поддержки пользователей, предназначенная для внедрения в ООО «Амурагроцентр».

На начальном этапе проведен анализ предметной области, включая ИТ-инфраструктуру и существующие процессы поддержки в ООО «Амурагроцентр». Проанализированы методы интеллектуализации, выбраны технологии обработки естественного языка и больших языковых моделей, а также рассмотрены и выбраны внешние ИИ API для реализации интеллектуальных функций.

На этапе проектирования разработана архитектура системы. Спроектирована структура базы данных и детально проработаны алгоритмы многостадийного ИИ-поиска и взаимодействия с ИИ API, включая разработку эффективных запросов.

В процессе программной реализации созданы все основные модули системы: управление статьями, тегами, файловыми вложениями, пользователями и стандартный поиск. Ключевым результатом стала успешная интеграция интеллектуальных функций: многостадийного поиска по запросам на естественном языке, автоматической генерации резюме статей и предложений релевантных тегов.

Расчет экономической эффективности показал целесообразность внедрения системы со сроком окупаемости менее одного года. Разработанная система демонстрирует эффективное применение современных ИИ-технологий для решения задач предприятия.

## БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Воробьев Г. А. Основы программирования на Python : учебно-методическое пособие. Липецкий ГПУ, 2022. 89 с.
- 2 Гарретт Джесс. Веб-дизайн. Элементы опыта взаимодействия. М. Символ-Плюс, 2020. 285 с.
- 3 Кардаш Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ : учеб. пособие. АмГУ. ИФФ. Благовещенск. Изд-во Амур. гос. ун-та, 2002. 60 с.
- 4 Круг С. Не заставляйте меня думать. М. Эксмо, 2021. 256 с.
- 5 Любарский Ю. Я. Интеллектуальные информационные системы. М. Наука, 2019. 228 с.
- 6 Нестеров С. А. Основы информационной безопасности. 3-е изд., стер. СПб. Лань, 2024. – 324 с.
- 7 Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. М. Питер, 2023. 832 с.
- 8 СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания : утв. постановлением Главного государственного санитарного врача РФ от 28 января 2021 г. № 2 : введен в действие с 01 марта 2021 г.
- 9 СанПиН 2.1.3684-21. Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпидемических (профилактических) мероприятий : утв. постановлением Главного государственного санитарного врача РФ от 28 января 2021 г. № 3 : зарегистрировано в Минюсте России 29 января 2021 г. № 62297.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Акулов, О. А. Информатика: базовый курс : учебник / О. А. Акулов, Н. В. Медведев. – 4-е изд., стер. – М. : Омега-Л, 2021. – 560 с.
- 2 Бабич, А. В. Введение в UML / А. В. Бабич. — М. : НОУ «Институт», 2016. — 209 с.
- 3 Басаргин, А. А. Методы и средства проектирования информационных систем / А. А. Басаргин. – Новосибирск : СГУГИТ, 2015. – 236 с.
- 4 Боресков, А. В. Разработка и анализ требований к программному обеспечению / А. В. Боресков. – Москва : ДМК Пресс, 2020. – 256 с.
- 5 Бирюков, А. Н. Процессы управления информационными технологиями : учебное пособие / А. Н. Бирюков. – М. : Кнорус, 2019. – 208 с.
- 6 Данилин, Д. А. Анализ технологий веб-программирования для создания модулей визуализации и выгрузки данных информационных систем / Д. А. Данилин, Я. В. Зиновьев, К. М. Кузьмин // Вестник Пензенского государственного университета. — 2019. — № 3. — С. 76–79.
- 7 Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 : учебное пособие / А. В. Диков. — М. : Лань, 2019. — 188 с.
- 8 Илюшечкин, В. М. Основы использования и проектирования баз данных : учебник для вузов / В. М. Илюшечкин. – М. : Издательство Юрайт, 2024. – 213 с.
- 9 Исаков, С. А. Интеллектуальная система поддержки пользователей при работе с программным обеспечением / С. А. Исаков, С. Г. Самохвалова // Флагман науки : научный журнал. – 2025. – № 5(28). – СПб. : Изд.ГНИИ «Нацразвитие». – Режим доступа: [https://flagmannauki.ru/files/528-Isakov\\_Stepan\\_Alekseevich\\_3785.pdf](https://flagmannauki.ru/files/528-Isakov_Stepan_Alekseevich_3785.pdf) – 02.06.2025. – DOI: 10.37539/2949-1991.2025.28.5.001.
- 10 Калитвин, В. А. Введение в программирование на Python : учебное пособие / В. А. Калитвин. – Липецк : Липецкий ГПУ, 2023. — 84 с.

11 Карпова, Т. С. Базы данных. Модели, разработка, реализация / Т. С. Карпова. – 2-е изд. – М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 403 с.

12 Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — Москва : Издательство Юрайт, 2020. — 230 с.

13 Новиков, Б. Основы технологий баз данных : учебное пособие / Б. Новиков. – М. : ДМК Пресс, 2020. – 582 с.

14 Об утверждении общих требований к организации безопасного рабочего места : приказ Минтруда России от 29 октября 2021 г. № 774н : зарегистрирован в Минюсте РФ 25 ноября 2021 г. № 65987.

15 Рамальо, Л. Python. К вершинам мастерства / Л. Рамальо. – Москва : ДМК Пресс, 2021. – 880 с.

16 Савельев, А. О. Проектирование и разработка веб-приложений на основе технологий Microsoft : учебное пособие / А. О. Савельев, А. А. Алексеев. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 418 с.

17 Сидоренко, И. Дизайнер интерфейсов / И. Сидоренко. – М. : Олимп-Бизнес, 2019. – 255 с.

18 Советов, Б. Я. Базы данных : учебник для прикладного бакалавриата / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2019. — 420 с.

19 Стружкин, Н. П. Базы данных: проектирование / Н. П. Стружкин, В. В. Годин. – Москва : Издательство Юрайт, 2023. – 477 с.

20 Трудовой кодекс Российской Федерации : от 30 декабря 2001 г. № 197-ФЗ (ред. от 07 апреля 2025 г.).

21 Фрэйн, Б. Отзывчивый дизайн на HTML5 и CSS3 для любых устройств / Б. Фрэйн. – М. : Питер, 2021. – 336 с.

22 Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. – 2-е изд., перераб. и доп. – М. : Издательство Юрайт, 2024. – 349 с.

23 Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения / С. А. Чернышев. – Москва : Издательство Юрайт, 2023. – 176 с.

24 Шилдс, У. SQL: быстрое погружение / У. Шилдс. – М. : Питер, 2022. – 224 с.

25 Шумилин, В. К. ПЭВМ. Защита пользователя / В. К. Шумилин. – М. : Охрана труда и социальное страхование, 2001. – 214 с.

26 Ясенов, В. Н. Информационная безопасность : учебное пособие / В. Н. Ясенов, А. В. Дорожкин, А. Л. Сочков, О. В. Ясенов. – Нижний Новгород : Нижегородский госуниверситет им. Н. И. Лобачевского, 2017. – 198 с.

# ПРИЛОЖЕНИЕ А

## Техническое задание

### 1 ОБЩИЕ СВЕДЕНИЯ

#### 1.1 Полное наименование системы

Интеллектуальная система поддержки пользователей ООО «Амурагроцентр».

#### 1.2 Разработчик

Разработчик – студент 1103-об группы института компьютерных и инженерных наук Амурского государственного университета – Исаков Степан Алексеевич.

#### 1.3 Краткая характеристика области применения

Интеллектуальная система поддержки пользователей предназначена для автоматизированного предоставления ответов на запросы сотрудников ООО «Амурагроцентр» по вопросам технической поддержки, работы программного обеспечения и оборудования.

#### 1.4 Плановые сроки начала и окончания работы

Плановые сроки начала и окончания работ по созданию системы: начало разработки – 20.02.2025 г., окончание – 1.06.2025 г.

### 2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

#### 2.1 Назначение системы

Система разрабатывается с целью повышения эффективности работы службы технической поддержки ООО «Амурагроцентр» путем автоматизации процесса обработки запросов пользователей и предоставления решений на основе базы знаний.

#### 2.2 Цели создания системы

Основные цели создания системы:

- Сокращение времени обработки запросов пользователей;
- Снижение нагрузки на специалистов технической поддержки;
- Создание единой базы знаний по типовым проблемам;

## Продолжение Приложения А

### 3 ХАРАКТЕРИСТИКИ ОБЪЕКТА АВТОМАТИЗАЦИИ

Объектами автоматизации являются процессы:

- Обработка обращений пользователей по техническим вопросам;
- Поиск и предоставление решений на основе базы знаний;
- Анализ часто возникающих проблем;
- Формирование отчетности по работе службы поддержки;
- Управление базой знаний технических решений.

### 4 ТРЕБОВАНИЯ К СИСТЕМЕ

#### 4.1 Требования к системе в целом

##### 4.1.1 Требования к структуре и функционированию системы

Система должна быть реализована в виде веб-приложения с функциональными подсистемами, отвечающими за ввод и отображение информации, интеллектуальную обработку пользовательских запросов и поиск релевантных данных, управление контентом и администрирование системы, хранение всей необходимой информации, интеграцию с внешними сервисами искусственного интеллекта, а также формирование отчетности о работе системы. Комплексное функционирование должно обеспечиваться подсистемами информационного, лингвистического, программного, технического, алгоритмического, математического и организационно-методического обеспечения.

##### 4.1.2 Требования к численности и квалификации персонала

Для эксплуатации системы предусматривается наличие квалифицированного системного администратора и администратора базы знаний, при этом основными конечными пользователями являются сотрудники организации, обладающие базовыми навыками работы с веб-приложениями.

##### 4.1.3 Требования к режимам функционирования системы

Система должна функционировать в следующих режимах:

- Основной режим – получение ответов на запросы в реальном времени;

## Продолжение Приложения А

– Режим администрирования – для обновления базы знаний и управления системой;

### **4.2 Требования к функциям, выполняемым системой**

4.2.1 Подсистема обработки запросов пользователей должна обеспечивать:

- Прием и регистрацию запросов пользователей;
- Автоматический анализ текста запроса;
- Выделение ключевых слов и тематики запроса;
- Поиск релевантных решений в базе знаний;
- Ранжирование найденных решений по релевантности;
- Предоставление пользователю списка возможных решений.

4.2.2 Подсистема администрирования должна обеспечивать:

- Управление пользователями и их правами;
- Ведение справочников и классификаторов;
- Управление содержимым базы знаний;
- Мониторинг работы системы;

### **4.3 Требования к надежности**

4.3.1 Требования к обеспечению надежного функционирования программы

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении следующих внештатных ситуаций:

- Сбои в системе электроснабжения;
- Сбои операционной системы;
- Сбои в работе СУБД;
- Ошибки в работе сетевого оборудования;
- Сбои в работе или недоступность внешних ИИ API;
- Повреждение или нарушение целостности базы данных;
- Резкое увеличение нагрузки со стороны пользователей;
- Ошибки в работе программного обеспечения системы.

## Продолжение Приложения А

### **4.4 Требования к составу и параметрам технических средств**

Минимальный состав технических средств:

- Сервер приложений:
- Процессор: не менее 4 ядер с частотой не менее 2.5 ГГц;
- Оперативная память: не менее 8 ГБ;
- Жесткий диск: не менее 250 ГБ;
- Рабочие станции пользователей:
- Процессор: не менее 2 ядер с частотой не менее 2.0 ГГц;
- Оперативная память: не менее 8 ГБ;
- Жесткий диск: не менее 128 ГБ.

### **4.5 Требования к информационной и программной совместимости**

#### 4.5.1 Требования к информационным структурам и методам решения

Система должна быть реализована с использованием следующих технологий:

- Язык программирования: Python;
- Фреймворк: Flask;
- СУБД: SQLite;
- Клиентская часть: HTML5, CSS3, JavaScript.

#### 4.5.2 Требования к исходным кодам и языкам программирования

Исходные коды должны быть написаны на языке Python с использованием фреймворка Flask. Разработка должна вестись в среде PyCharm, Visual Studio Code или другой IDE с поддержкой Python.

#### 4.5.3 Требования к программным средствам, используемым программой

Для функционирования системы необходимо наличие следующего программного обеспечения:

- Операционная система: Windows Server 2019 или новее;
- СУБД: SQLite;

## Продолжение Приложения А

– Для клиентских рабочих мест: Windows 10 или новее, современный веб-браузер.

### 4.5.4 Требования к защите информации и программ

Система должна обеспечивать следующие меры безопасности:

- Идентификация и аутентификация пользователей;
- Разграничение прав доступа пользователей к функциям системы;
- Протоколирование действий пользователей;
- Резервное копирование базы данных;
- Защита от несанкционированного доступа;
- Защита от SQL-инъекций и других типов атак;
- Шифрование конфиденциальных данных;

### 4.6 Требования к маркировке и упаковке

Маркировка и упаковка программного обеспечения должна осуществляться в соответствии со следующими требованиями:

- Носители информации должны иметь маркировку с указанием версии ПО;
- Документация должна быть упакована в отдельные папки;
- Электронные версии документов должны быть записаны на отдельный носитель.

### 4.7 Требования к транспортированию и хранению

Требования к транспортированию и хранению носителей информации:

- Носители должны храниться в помещении при температуре от +5°C до +40°C;
- Относительная влажность воздуха при хранении не должна превышать 80%;
- Должна быть обеспечена защита от прямых солнечных лучей;
- Срок хранения носителей информации не менее 5 лет.

## Продолжение Приложения А

### 4.8 Специальные требования

Система должна обеспечивать следующие специальные требования:

- Возможность масштабирования системы;
- Поддержка одновременной работы не менее 10 пользователей;
- Время отклика системы не более 3 секунд;
- Доступность системы 24/7 с плановыми техническими перерывами;
- Возможность интеграции с существующими системами организации;

### 5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Документация должна разрабатываться в соответствии с требованиями следующих нормативных документов:

- ГОСТ 19.201-78 ЕСПД. Техническое задание;
- ГОСТ 19.404-79 ЕСПД. Пояснительная записка;
- ГОСТ 19.505-79 ЕСПД. Руководство оператора;
- ГОСТ 19.301-79 ЕСПД. Программа и методика испытаний;
- ГОСТ 34.602-89 Информационная технология. Техническое задание на создание автоматизированной системы;
- ГОСТ 34.601-90 Информационная технология. Автоматизированные системы. Стадии создания;
- ГОСТ Р 51188-98 Защита информации. Испытания программных средств;
- ГОСТ Р ИСО/МЭК 12207-2010 Процессы жизненного цикла программных средств.

В состав документации должны входить:

- Техническое задание;
- Пояснительная записка;
- Руководство пользователя;
- Руководство администратора;

## Продолжение Приложения А

### 6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Экономическая эффективность внедрения системы определяется следующими показателями:

- Сокращение времени обработки запросов на 60%;
- Снижение нагрузки на службу поддержки на 40%;
- Увеличение скорости решения типовых проблем на 70%;
- Повышение удовлетворенности пользователей на 50%;
- Сокращение затрат на техническую поддержку на 30%.
- Снижение времени простоя сотрудников на 25%;

### 7 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Разработка системы должна быть проведена в три стадии.

Разработка технического задания (1 месяц):

- Обследование объекта автоматизации;
- Разработка и утверждение технического задания.

Рабочее проектирование (3 месяца):

- Разработка проектных решений;
- Проектирование архитектуры системы;
- Разработка документации на систему;
- Разработка документации на поставку комплектующих изделий;
- Разработка базы данных;
- Разработка серверной части;
- Разработка пользовательского интерфейса;
- Разработка модуля анализа запросов.

Внедрение (2 месяца):

- Подготовка объекта автоматизации;
- Подготовка персонала;
- Комплектация системы поставляемыми изделиями;
- Проведение приемо-сдаточных испытаний;

## Продолжение Приложения А

- Проведение опытной эксплуатации.

### 8 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

Для проверки соответствия системы требованиям настоящего ТЗ проводятся следующие виды испытаний:

- Предварительные испытания;
- Опытная эксплуатация;
- Приемочные испытания.

Состав, объем и методы испытаний системы определяются программой и методикой испытаний системы, разрабатываемыми на стадии рабочего проектирования согласно ГОСТ 34.603-92 и согласованными с заказчиком.

В ходе проведения предварительных испытаний необходимо проверить:

- Комплектность поставки;
- Работоспособность в целом и функциональных элементов системы;
- Соответствие требованиям технического задания.

Опытная эксплуатация проводится для определения фактических значений количественных и качественных характеристик системы, выявления причин сбоев в работе, определения фактической эффективности.

Приемочные испытания проводятся для оценки соответствия технико-экономических показателей системы требованиям настоящего ТЗ. Для их проведения заказчиком формируется приемочная комиссия в соответствии с ГОСТ 34.603-92. В состав комиссии входят представители заказчика (руководство, начальник отдела АСУ), представители разработчика и независимые эксперты.

Целью испытаний является окончательная оценка соответствия разработанной системы всем требованиям технического задания, включая функциональные, нефункциональные и технико-экономические показатели. В ходе испытаний комиссия рассматривает представленную документацию и проводит демонстрацию работы системы с проверкой ее ключевых функций.

## Продолжение Приложения А

По результатам проведения приемо-сдаточных испытаний комиссия должна принять решение о соответствии системы требованиям ТЗ и возможности оформления документов об окончании работ.